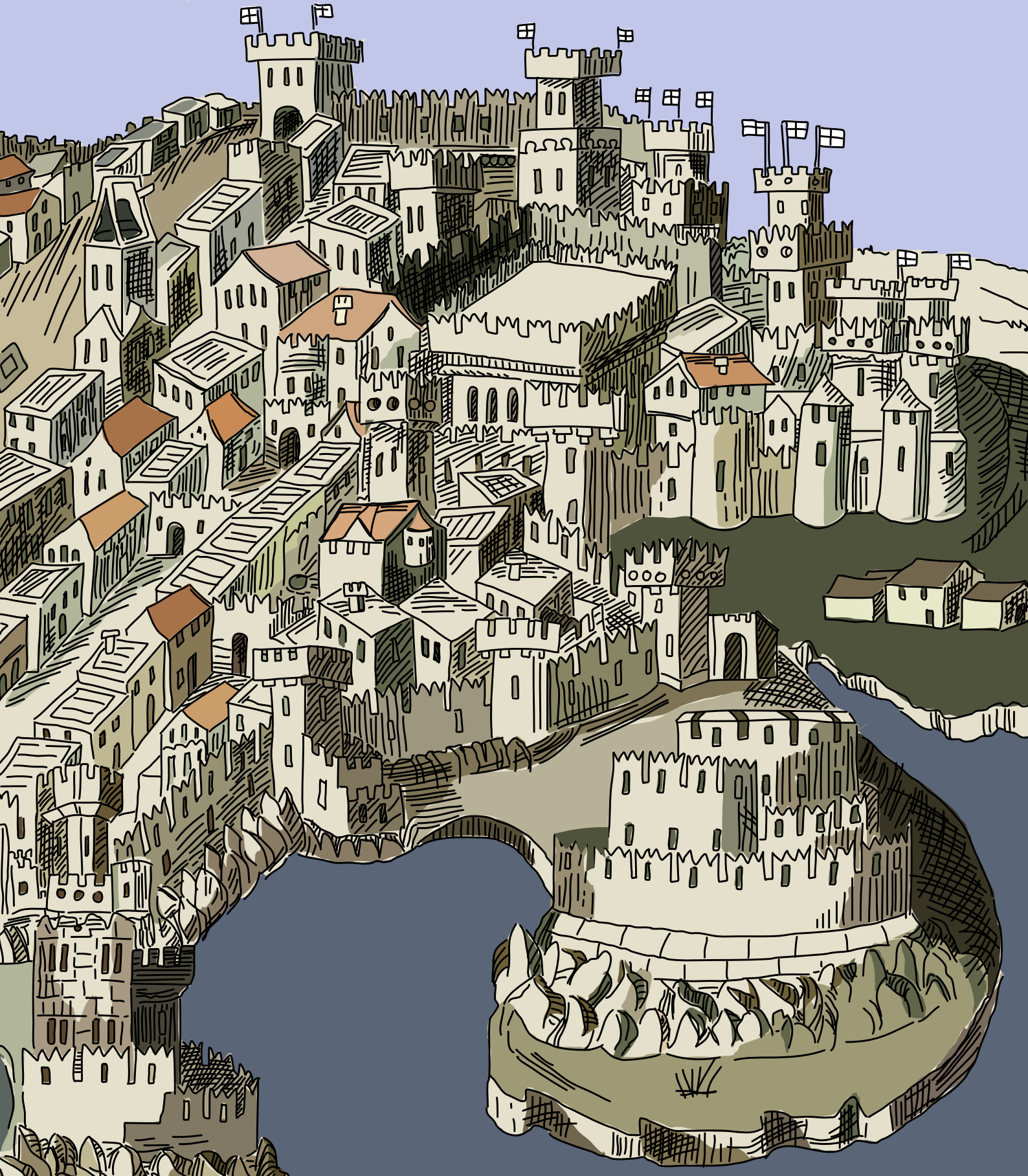


Θέματα Ασφάλειας Υπολογιστών και Δικτύων

Μιλτιάδης Αναγνώστου

- Έκδοση 0.80: 7 Δεκεμβρίου 2021, 16:36.

- Η εικόνα του εξωφύλλου δείχνει την οχυρωμένη Ρόδο περί το 1487. Είναι βασισμένη σε μια εικόνα από το χειρόγραφο με τίτλο *Περιγραφή του ταξιδιού από την Κωνσταντία στην Ιερουσαλήμ*, του Konrad von Grünenberg (Beschreibung der Reise von Konstanz nach Jerusalem) που βρίσκεται στην Badische Landesbibliothek Karlsruhe (`https://digital.blb-karlsruhe.de/blbhs/content/titleinfo/7061`, άδεια Creative Commons CC BY 4.0).

Πρόλογος

Αυτό το κείμενο προορίζεται για το μάθημα της ασφάλειας στο όγδοο εξάμηνο σπουδών της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών και για το μάθημα της ασφάλειας με έμφαση στα δεδομένα στο διατμηματικό πρόγραμμα ΕΔΕΜΜ. Δεν είναι ένα γενικό βιβλίο ασφάλειας υπολογιστών και δικτύων επειδή καλύπτει μόνο θέματα που διδάσκονται στα παραπάνω μαθήματα. Είναι μια «προέκδοση», δηλαδή δεν είναι ολοκληρωμένο, ενώ αργότερα μπορεί να υπάρξουν επόμενες εκδόσεις.

Γι᾽ αυτό ο αναγνώστης που θέλει οπωσδήποτε να τυπώσει πριν διαβάσει καλό θα ήταν να περιοριστεί στην εκτύπωση μόνο όποιου κεφαλαίου διαβάζει. Ιδανικά όμως δεν θα τύπωνε καθόλου το κείμενο, αφενός για να ωφελήσει το περιβάλλον, αφετέρου για να μπορεί να το ανανεώνει εύκολα. Εξ άλλου είναι γραμμένο κυρίως ως ηλεκτρονικό βιβλίο, έχει δηλαδή εσωτερικούς και εξωτερικούς συνδέσμους, παραπομπές κ.λπ., ενώ ο αναγνώστης θα μπορούσε να χρησιμοποιήσει την αναζήτηση όποιου pdf viewer χρησιμοποιεί. Παρ᾽ όλα αυτά στο τέλος του κειμένου έχει προστεθεί ένα αλφαβητικό ευρετήριο.

Ο αναγνώστης παραπέμπεται συχνά στη βιβλιογραφία. Πολλές φορές είναι καλό να διαβάσει το πρωτότυπο άρθρο για βαθύτερη κατανόηση. Μερικά άρθρα είναι ιστορικής πλέον σημασίας.

Είναι αυτονόητο πως είναι καλοδεχούμενα σχόλια, υποδείξεις για διορθώσεις, αναφορές σφαλμάτων κ.λπ.

Οκτώβριος 2021

Περιεχόμενα

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή στην ασφάλεια

Η ασφάλεια είναι μια περιοχή που μελετάει τρόπους άμυνας απέναντι σε επιθέσεις, δηλαδή σε προβλήματα που δημιουργεί στη λειτουργία ενός συστήματος ένας αντίπαλος εκ προθέσεως.

Τα πιο συνηθισμένα ίσως προβλήματα είναι αυτά που δημιουργούνται επειδή ένα σύστημα λειτουργεί σε ένα δύσκολο περιβάλλον και ο σχεδιασμός του δεν έχει αρθεί στο ύψος των περιστάσεων, δηλαδή δεν έχει προβλέψει αντοχές σε όλες τις συνθήκες και απέναντι σε όλα τα πιθανά περιστατικά. Τυπικό παράδειγμα είναι πολυάριθμοι πύραυλοι και άλλα διαστημικά οχήματα που εξ αιτίας ενός μικρού λάθους στο σχεδιασμό ή στην υλοποίηση κατέληξαν σε μια μπάλα φωτιάς. Στον σχεδιασμό συνήθως εμπλέκονταν οι καλύτεροι επιστήμονες στην περιοχή, αλλά το διάστημα δεν συγχωρεί τα λάθη. Ωστόσο κανείς δεν θα μπορούσε να αποδώσει στο διάστημα πρόθεση.

Ο Norbert Wiener είχε κάνει έναν παραστατικό διαχωρισμό ανάμεσα στην πολυπλοκότητα που καλούμαστε να αντιμετωπίσουμε σε ένα δύσκολο περιβάλλον και σε κείνη που μας δημιουργεί ένας ευφυής αντίπαλος. Στην περιοχή της *Κυβερνητικής* που ο ίδιος είχε ορίσει διέκρινε ανάμεσα στον *Αυγουστινιανό διάβολο* και τον *Μανιχαϊστικό διάβολο* [Wie65; Gal94]. Ο πρώτος είναι αθώος αλλά περίπλοκος, είναι μια μεταφορά για την πολυπλοκότητα ενός συστήματος που καλείται να αντιμετωπίσει ένας επιστήμονας, π.χ. ο φυσικός επιστήμονας πρέπει να ανακαλύψει τα μυστικά της φύσης. Η φύση όμως δεν προσπαθεί να παραπλανήσει τον επιστήμονα για να τον κάνει να φτάσει σε μια αποτυχία. Ο δεύτερος είναι ακριβώς ο αντίπαλος που θα χρησιμοποιήσει μεθόδους παραπλάνησης.

Οι σχεδιαστές οποιωνδήποτε συστημάτων έχουν ανακαλύψει εδώ και αιώνες διάφορες άμυνες για να αντιμετωπίζουν τον Αυγουστινιανό διάβολο. Ανάμεσα σ᾽ αυτές είναι να χρησιμοποιούν μοντέλα και αυστηρές σχεδιαστικές μεθόδους και πρακτικές υλοποίησης, καθώς και το να κάνουν δοκιμές πριν την τελικά χρήση. Σχετικά πρόσφατα έχει επίσης αναπτυχθεί η μελέτη συστημάτων με ανοχή και αντοχή σε σφάλματα (fault tolerance), δηλαδή συστημάτων που έχουν τη δυνατότητα να αποφεύγουν μια καταστροφή ακόμη και αν ορισμένα υποσυστήματά τους αστοχήσουν, επειδή διαθέτουν εναλλακτικές λύσεις. Για παράδειγμα ένα αεροπλάνο με δύο κινητήρες, καθένας από τους οποίους είναι σε θέση να το κρατήσει στον αέρα μόνος του, είναι πιο ανθεκτική λύση από ένα αεροπλάνο με ένα και μοναδικό κινητήρα.

Τέτοιου είδους αστοχίες που είναι πολύ δύσκολο να προβλεφθούν, π.χ. αν ένα σμήνος πουλιών θα πέσει πάνω στον κινητήρα του αεροπλάνου, οι σχεδιαστές τις θεωρούν «τυχαία» περιστατικά και τυχαία σφάλματα και συχνά κάνουν εκτιμήσεις για την πιθανότητα να συμβούν. Αντίθετα, μια επίθεση κίνησης σε ένα εξυπηρετητή (server) από ένα botnet δεν θεωρείται τυχαίο περιστατικό.

Ένα ακόμη ενδιαφέρον παράδειγμα για τη διάκριση ανάμεσα στις δύο αυτές κατηγορίες προβλημάτων, τα τυχαία και τα εμπρόθετα, μας δίνει το θεώρημα του Shannon για τη χωρητικότητα καναλιού. Ο Shannon μας λέει τι συμβαίνει όταν μια πηγή στέλνει bits σε ένα προορισμό μέσα από ένα κανάλι. Με την υπόθεση ότι κάθε bit μπορεί να φτάσει αλλοιωμένο με κάποια πιθανότητα, το θεώρημα εγγυάται ότι υπάρχει τρόπος να περάσουν C bits ανά sec χωρίς λάθη. Το C είναι η χωρητικότητα του καναλιού και το σημαντικό εδώ είναι οποιοσδήποτε ρυθμός μετάδοσης κάτω από C εξασφαλίζει επιτυχία (αν και η επιτυχία μπορεί να απαιτεί τη χρήση προχωρημένου ελέγχου σφαλμάτων). Η απόδειξη του θεωρήματος βασίζεται ακριβώς στο ότι τα λάθη συμβαίνουν τυχαία. Αν στο κανάλι παρεμβληθεί ένας αντίπαλος που αλλοιώνει τα bits όπως νομίζει, το θεώρημα πλέον δεν ισχύει. Αν την πρώτη περίπτωση την εξετάζει η θεωρία πληροφορίας, την δεύτερη την εξετάζει η ασφάλεια επικοινωνιών.

Συχνά η ίδια η έννοια της «αστοχίας» ή του «προβλήματος» ή της «βλάβης» πρέπει να επανακαθορισθούν στην περιοχή της ασφάλειας και να αποκτήσουν μια πολύ ευρεία σημασία. Υπάρχουν περιπτώσεις που ένα παρείσακτο πρόγραμμα το μόνο που κάνει σε έναν υπολογιστή είναι να υποκλέπτει στοιχεία. Η λειτουργία του υπολογιστή ελάχιστα παραβλάπτεται, όμως ο χρήστης του υπολογιστή μπορεί να ζημιωθεί αργότερα αν ο τραπεζικός του λογαριασμός εμφανίσει ένα έλλειμμα. Σε άλλες περιπτώσεις οι πληροφορίες μπορεί να καταλήξουν σε ένα αρχείο πληροφοριών, που θα χρησιμοποιήσει π.χ. ένα κόμμα για να επηρεάσει την ψήφο του ή μπορεί και να μη χρησιμοποιηθούν ποτέ.

ΚΕΦΑΛΑΙΟ 2

Συμμετρική κρυπτογραφία

2.1 Εισαγωγή

Η συμμετρική κρυπτογραφία είναι ίσως το πιο βασικό εργαλείο της περιοχής της ασφάλειας, τουλάχιστον στις περιοχές της επικοινωνίας και της αποθήκευσης δεδομένων. Επιτρέπει σε δύο πλευρές που επικοινωνούν (συνήθως τις ονομάζουμε Bob & Alice) να ανταλλάσσουν μηνύματα που δεν μπορεί να διαβάσει ένα τρίτο μέρος (που συνήθως ονομάζεται Trudy) και που επίσης έχει (ατυχώς) πρόσβαση στο κανάλι επικοινωνίας, επειδή τα μηνύματα έχουν μετασχηματισθεί σε μορφή ακατανόητη για το τελευταίο μέρος - είναι όπως λέμε «κρυπτογραφημένα».

Σε μια απλουστευμένη διατύπωση η κρυπτογραφία επιτρέπει στις δύο πλευρές να ανταλλάσσουν μυστικά μηνύματα μέσα από ένα φανερό κανάλι (δηλαδή κανάλι που το περιεχόμενό του δεν είναι αόρατο σε τρίτους). Εάν οι δύο πλευρές έχουν στη διάθεσή τους ένα κανάλι, στο οποίο δεν έχει πρόσβαση ένα τρίτο ανεπιθύμητο μέρος, μπορούν να παραιτηθούν από το να στείλουν τα μηνύματά τους «κρυπτογραφημένα».[1]

Ο Bob μετασχηματίζει κάθε κείμενο (string) x με συγκεκριμένη συνάρτηση, έστω f, και στέλνει στην Alice το $y = f(x)$. Για παράδειγμα ο Ιούλιος Καίσαρ χρησιμοποιούσε ως f την κυκλική ολίσθηση των γραμμάτων κατά 3 γράμματα με τη σειρά που είναι στο αλφάβητο, δηλαδή αντί του A θα έγραφε D και αντί του `venividivici` θα έγραφε `zhqlzlglzlfl`.

Στη συνέχεια η Alice εφαρμόζει την αντίστροφη συνάρτηση f^{-1} πάνω στο y για να ανακτήσει το x. Η μυστικότητα της επικοινωνίας μπορεί προφανώς να διαφυλαχθεί (από την περιέργεια της Trudy) μόνον εφόσον η συνάρτηση f (και η αντίστροφή της) είναι γνωστή μόνο στις δύο πλευρές (Bob & Alice). Η f είναι συνήθως γνωστή ως *αλγόριθμος κρυπτογράφησης* και η f^{-1} ως *αλγόριθμος αποκρυπτογράφησης*. Στο παράδειγμα της κρυπτογράφησης του Καίσαρα η f^{-1} είναι η κυκλική ολίσθηση κατά τρία γράμματα προς τα πίσω στο αλφάβητο.

Για πολλά χρόνια η κρυπτογραφία βασιζόταν στη μυστικότητα του αλγορίθμου κρυπτογράφησης. Ωστόσο η πιθανότητα διαρροής του αλγορίθμου οδήγησε αρκετά νωρίς στην παραμετροποίησή του ως προς μια μυστική παράμετρο που λέγεται «κλειδί»,

[1] Για παράδειγμα οι τεχνικές *απλωμένου φάσματος* (spread spectrum) εμποδίζουν τον αντίπαλο να διαβάζει την πληροφορία που επικοινωνείται.

δηλαδή οδήγησε στη χρήση μιας συνάρτησης f_k, όπου όλες οι συναρτήσεις της οικογένειας ακολουθούν την ίδια αλγοριθμική λογική, αλλά διαφοροποιούνται ως προς την τιμή του k, την οποία γνωρίζουν μόνο ο Bob και η Alice. Στο παράδειγμα του αλγορίθμου του Καίσαρα το k θα μπορούσε να εκφράζει το μήκος της ολίσθησης, εν προκειμένω 3. Ο διάδοχός του ο Οκταβιανός χρησιμοποίησε $k = 1$. Με μήκος αλφαβήτου ίσο με 25 υπάρχουν 24 διαφορετικά διαθέσιμα κλειδιά $k = 1, 2, \ldots, 24$ (ενώ για $k = 25$ δεν υπάρχει μεταβολή από το αρχικό στο τελικό κείμενο).

Παρά την χρήση κλειδιών στην κρυπτογραφία, όσοι την χρησιμοποιούσαν προσπαθούσαν να κρατήσουν κρυφό και τον ίδιο τον αλγόριθμο μέχρι πριν λίγες δεκαετίες (πρακτική που λέγεται "security by obscurity"). Για τον λόγο αυτόν δεν υπήρχαν και βιβλία κρυπτογραφίας.

Ωστόσο στη συνέχεια άλλαξε η αντίληψη σε σχέση με την μυστικότητα του αλγορίθμου. Βλέποντας η κοινότητα των κρυπτογράφων ότι συχνά βρίσκονταν αδυναμίες στους αλγορίθμους, θεώρησε ότι δημοσιοποιώντας τους θα εξασφαλίσει την μεγαλύτερη δυνατή δοκιμασία και οποιαδήποτε αδυναμία θα διαπιστωθεί το ταχύτερο δυνατό. Εστίασε δηλαδή τις προσπάθειές της στο να δημιουργεί καλύτερες και δοκιμασμένες μεθόδους κρυπτογράφησης, αντί να υπολογίζει στη μυστικότητά τους. Επομένως ο χρήστης μιας μεθόδου κρυπτογράφησης σήμερα μπορεί να υπολογίζει μόνο στη μυστικότητα του κλειδιού.

Ωστόσο η χρήση μυστικών κλειδιών θέτει στους χρήστες των κρυπτογραφικών μεθόδων προβλήματα διαχείρισης των κλειδιών. Το πρώτο απ᾽ αυτά είναι η επιλογή κατάλληλων κλειδιών. Το μέγεθος του συνόλου δυνατών κλειδιών είναι ζήτημα αποφασιστικής σημασίας: Όταν ο Bob στείλει στην Alice ένα κρυπτογραφημένο μήνυμα $y = f_k(x)$, η Trudy (εφόσον αναχαιτίσει το y στο κανάλι) έχει τη δυνατότητα να δοκιμάσει να κάνει τον υπολογισμό $f^{-1}(k)$ με διαφορετικά k και να δει αν το αποτέλεσμα βγάζει νόημα. Προκειμένου ο Bob και η Alice να δυσκολέψουν την Trudy είναι καλό να επιλέξουν το k μέσα από ένα αρκετά μεγάλο σύνολο τιμών. Αν το κλειδί είναι π.χ. ένας φυσικός αριθμός, ο αλγόριθμος πρέπει να δέχεται ως κλειδί έναν αριθμό ικανού μήκους.

Αν το κλειδί το διάλεξε ο Bob πρέπει να βρει έναν ασφαλή τρόπο να το παραδώσει στην Alice. Αν ο Bob και η Alice είναι ήδη αρκετά μακριά, πρέπει να βρει ένα κανάλι επικοινωνίας διαφορετικό από εκείνο στο οποίο σκοπεύουν να ανταλλάσσουν στη συνέχεια κρυπτογραφημένα μηνύματα. Συχνά ένα τέτοιο κανάλι είναι δύσκολο να βρεθεί και μπορεί επίσης να προσβληθεί από την Trudy. Περαιτέρω μέρος της διαχείρισης κλειδιών μπορεί να είναι η ασφαλής φύλαξή του, που γίνεται τόσο πιο δύσκολη όσο πιο μακρύ και απρόβλεπτο είναι το κλειδί. Θα δούμε αργότερα ότι η διαχείριση των κλειδιών είναι ιδιαίτερα απαιτητική διαδικασία στα κρυπτονομίσματα.

Στα επόμενα μια συνάρτηση κρυπτογράφησης με κλειδί K θα συμβολίζεται με e_K, ενώ η αντίστοιχη συνάρτηση αποκρυπτογράφησης θα συμβολίζεται με d_K.

2.2 Κώδικες υποκατάστασης

Πριν εξετάσουμε κώδικες που εμπίπτουν σε αυτήν την κατηγορία δυο λόγια για την ταξινόμησή τους και την σχετική ορολογία: Ένας *κώδικας υποκατάστασης* (substitution cipher) κωδικοποιεί το ακρυπτογράφητο κείμενο κατά τμήματα σύμφωνα με ένα σταθερό κανόνα. Αν αντικαθιστά κάθε γράμμα χωριστά λέγεται *απλός κώδικας υποκατάστασης*, αν τα τμήματα είναι μεγαλύτερα λέγεται *πολυγραφικός*. Ένας *μονοαλφαβητικός* κώδικας κάνει την υποκατάσταση ενός συγκεκριμένου τμήματος με τον ίδιο τρόπο σε όλο το μήκος του μηνύματος, ενώ ένας *πολυαλφαβητικός* κώδικας χρη-

σιμοποιεί διαφορετικές υποκαταστάσεις του ίδιου τμήματος σε διαφορετικές θέσεις του μηνύματος.

Οι επόμενοι λίγοι κώδικες αφορούν σε κείμενα γραμμένα σε μια φυσική γλώσσα, π.χ. γραμμένα με το λατινικό αλφάβητο (με 26 σύμβολα, χωρίς να ενδιαφερόμαστε να ξεχωρίσουμε τα κεφαλαία από τα μικρά και χωρίς να παριστάνουμε τα κενά ανάμεσα στις λέξεις). Στον συμβολισμό που θα χρησιμοποιεί για την περιγραφή των κωδίκων το x συμβολίζει ένα γράμμα του κειμένου (δηλ. είναι ένα string μήκους 1). Για να ξεχωρίζει το ακρυπτογράφητο κείμενο (plaintext) από το κρυπτογραφημένο το μεν πρώτο καταγράφεται με μικρά γράμματα, το δε δεύτερο με κεφαλαία. Αυτή είναι μια δημοφιλής σύμβαση στη βιβλιογραφία.

Συχνά αντί των 26 γραμμάτων $a, b, c, \ldots, z$ χρησιμοποιούμε τους αντίστοιχους αριθμούς $\{0, 1, 2, \ldots, 25\}$. Είναι χρήσιμο να ορίσουμε το σύνολο

$$\mathbb{Z}_m = \{0, 1, \ldots, m-1\},$$

δηλαδή το σύνολο των m πρώτων μη αρνητικών ακεραίων, εφοδιασμένο με τις πράξεις της πρόσθεσης και του πολλαπλασιασμού modulo m (ομάδα).

Μονοαλφαβητικοί κώδικες

Στον πρώτο από τους επόμενους δύο κώδικες γίνεται ολίσθηση κάθε γράμματος κατά ένα σταθερό αριθμό θέσεων, π.χ. τριών θέσεων στον κώδικα του Καίσαρα. Με τον τρόπο αυτόν είμαστε πάντοτε σίγουροι ότι δύο διαφορετικά γράμματα δεν θα πέσουν πάνω στο ίδιο κωδικοποιημένο γράμμα. Αυτό όμως μπορεί να γίνει αντιστοιχίζοντας τα 26 γράμματα σε 26 γράμματα με άλλη σειρά, δηλαδή σε μια μετάθεσή τους, επομένως η κρυπτογράφηση βασίζεται σε ένα πίνακα αντιστοίχισης. Αυτό γίνεται στον δεύτερο κώδικα.

Κώδικας ολίσθησης

Στον Κώδικα Ολίσθησης (Shift Cipher), που αποτελεί γενίκευση του κώδικα του Καίσαρα, ένα γράμμα που έχει αντίστοιχο αριθμό τον $x \in \mathbb{Z}_{26}$ κρυπτογραφείται βάσει της

$$e_K(x) = (x + K) \mod 26,$$

όπου $e_K(x)$ είναι ο αντίστοιχος αριθμός του κρυπτογραφημένου γράμματος και αποκρυπτογραφείται αντίστοιχα βάσει της

$$d_K(x) = (x - K) \mod 26$$

Για $K = 3$ έχουμε τον κλασσικό κώδικα του Καίσαρα και για $= 1$ τον κώδικα του Οκταβιανού. Στο εξής όταν αναφερόμαστε σε ένα γράμμα θα υπονοούμε τον αντίστοιχο αριθμό του χωρίς ιδιαίτερη αναφορά στο γεγονός, αν τουλάχιστον δεν προκύπτει παρεξήγηση.

Στη συνέχεια ορίζουμε ένα κώδικα υποκατάστασης, αφού πρώτα ορίσουμε το σύνολο $\mathcal{K}$ ως το σύνολο των δυνατών κλειδιών ενός κώδικα.

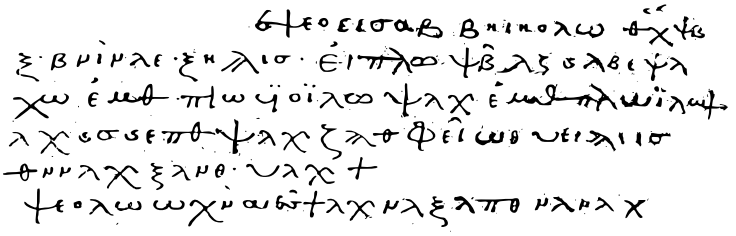
Κώδικας μετάθεσης (permutation cipher)

Έστω ότι το $\mathcal{K}$ περιέχει όλες τις δυνατές μεταθέσεις του συνόλου $\{0, 1, \ldots, 25\}$. Αν $\mu = (\mu_0, \mu_1, \ldots, \mu_{25}) \in \mathcal{K}$ είναι μια από τις μεταθέσεις (όπου καθένα από τα μ_i

είναι ένα από τα στοιχεία του $\{0, 1, \ldots, 25\}$ και όλα είναι διαφορετικά μεταξύ τους), τότε η κρυπτογράφηση με κλειδί $K = \mu$ γίνεται ως

$$e_\mu(x) = \mu_x$$

και η αποκρυπτογράφηση με την αντίστροφη συνάρτηση.



Σχήμα 2.1: Κρυπτογράφημα από βυζαντινό χειρόγραφο.

Παράδειγμα 1. Το απόσπασμα του Σχ. 2.1 βρίσκεται σε βυζαντινό χειρόγραφο[2] του έτους 1103. Η κρυπτογράφηση έχει γίνει με τον εξής πίνακα αντικατάστασης:

α	β	γ	δ	ε	ζ	η	θ
θ	η	ζ	ς	ε	γ	β	α
ι	κ	λ	μ	ν	ξ	ο	π
ϟ	π	ο	ξ	ω	μ	λ	κ
ρ	σ	τ	υ	φ	χ	ψ	ω
ϡ	ω	ψ	χ	φ	υ	τ	σ

Είναι κώδικας μετάθεσης, με μια μικρή τροποποίηση: Το αλφάβητο εξόδου είναι αριθμητικά ελληνικά σύμβολα, αλλ' επί της ουσίας διαφέρει από το αλφάβητο του μηνύματος μόνο στα σύμβολα , , . Ο αναγνώστης μπορεί να προσπαθήσει να βρει μνημονικό κανόνα για την αντικατάσταση. Το μήνυμα που έχει κρυπτογραφηθεί στο κείμενο του Σχ. 2.1 είναι το εξής: *ετελειωθη η βιβλοσ αυτη μηνι νοεμβριω εικοστη ογδοη ετουσ εξακισχιλιοστου εξακοσιοστου δωδεκατου γραφεισα χειρι ιωαννου μοναχου + τελοσ συν θεω του νομοκανονου.* □

Συγγενής κώδικας (affine cipher)

Ο επόμενος κώδικας αποτελεί γενίκευση του κώδικα μετάθεσης και λέγεται affine cipher, συγγενής κώδικας ή γραμμικός κώδικας.

Αν $x \in \mathbb{Z}_{26}$, η κρυπτογράφηση ανά γράμμα x γίνεται ως

$$e_{a,b}(x) = (ax + b) \mod 26$$

Η αποκρυπτογράφηση μπορεί να γίνει καταρτίζοντας τον σχετικό πίνακα. Μια πιο προσεκτική μαθηματική ανάλυση δείχνει ότι η αποκρυπτογράφηση μπορεί να γίνει ως

$$d_{a,b}(y) = a^{-1}(y - b) \mod 26$$

[2]Το κρυπτογραφημένο κείμενο βρίσκεται στο χειρόγραφο με αρ. 266 της μονής Βατοπεδίου του Αγ. Όρους. Το χειρόγραφο περιέχει το έργο του πατριάρχη Φωτίου (810-893) με τίτλο *Νομοκάνων*, μια συλλογή κανόνων εκκλησιαστικού δικαίου.

Στον παραπάνω τύπο a^{-1} είναι ο πολλαπλασιαστικός αντίστροφος του $a \mod 26$.[3]

Για να πέφτει κάθε γράμμα μέσω της κωδικοποίησης σε διαφορετικό γράμμα αποδεικνύεται ότι αναγκαία και ικανή συνθήκη είναι να ισχύει ότι μκδ$(a, 26) = 1$ (όπου μκδ(x, y) είναι ο μέγιστος κοινός διαιρέτης των x, y, δηλαδή οι a, 26 να είναι πρώτοι μεταξύ τους) (βλ. απόδειξη στο [SP18]). Επομένως οι δυνατές τιμές του a είναι 3, 5, 7, 9, 11, 15, 17, 21, 23 και 25 (12 το πλήθος). Το b μπορεί να πάρει οποιαδήποτε τιμή μεταξύ 0 και 25 (μεγαλύτερες τιμές δίνουν τους ίδιους κώδικες λόγω του modulo 26). Άρα τα δυνατά κλειδιά $K = (a, b)$ είναι μόνο $12 \times 26 = 312$.

Σε ένα γενικευμένο κώδικα με m σύμβολα, δηλαδή που χρησιμοποιεί αντί του $\mathbb{Z}_{26}$ το $\mathbb{Z}_m$, αποδεικνύεται ότι το πλήθος των δυνατών κλειδιών είναι $m\phi(m)$, όπου $\phi(m)$ είναι η *συνάρτηση Euler* (Euler phi function), της οποίας η τιμή ισούται με το πλήθος των αριθμών στο $\mathbb{Z}_m$ που είναι πρώτοι προς το m.

Πολυαλφαβητικοί κώδικες

Σε ένα πολυαλφαβητικό σύστημα κρυπτογράφησης κάθε γράμμα μπορεί να αντιστοιχηθεί με περισσότερα από ένα γράμματα. Ένας τρόπος να γίνει αυτό είναι με τον κώδικα που είναι γνωστός ως Vigenère[4].

Vigenère

Χρησιμοποιούμε ως κλειδί K μια λέξη μήκους m (που λέγεται λέξη-κλειδί, keyword). Χωρίζουμε το κείμενο σε ομάδες των m γραμμάτων και έστω ότι $(x_1, x_2, \ldots, x_m)$ είναι οι αριθμοί του $\mathbb{Z}_{26}$ που αντιστοιχούν στα γράμματα της ομάδας. Έστω επίσης ότι $(k_1, k_2, \ldots, k_m)$ είναι οι αντίστοιχοι αριθμοί για την λέξη κλειδί. Τότε η κρυπτογραφημένη λέξη που αντιστοιχεί στην $(x_1, x_2, \ldots, x_m)$ είναι αυτή που αντιστοιχεί στους αριθμούς $(y_1, y_2, \ldots, y_m)$, όπου $y_i = x_i + k_i$ $(i = 1, \ldots, m)$. Κατόπιν εφαρμόζουμε τον ίδιο αλγόριθμο στην επόμενη ομάδα από m γράμματα κ.ο.κ. μέχρις εξαντλήσεως του κειμένου.

Παράδειγμα 2. Για ακρυπτογράφητο κείμενο `attackatdawn`[5] δίνεται η λέξη κλειδί `lemon` που σε αριθμητική μορφή είναι

$$(k_1, k_2, k_3, k_4, k_5) = (12, 4, 12, 14, 13).$$

Το πρώτο γράμμα `a` του κειμένου αντιστοιχεί στο $x_1 = 0$ και δεδομένου ότι $k_1 = 12$, για το `l` του κλειδιού, μετασχηματίζεται στο ίδιο το `L` (και γενικώς όλα τα `a` θα μετατραπούν στο υποκείμενο γράμμα του κλειδιού). Στη συνέχεια $x_2 = 19$ (για το `t`), ενώ $k_2 = 5$ για το `e` του κλειδιού, οπότε $y_2 = 24$, άρα το δεύτερο γράμμα που προκύπτει είναι το `x`. Μετά το 5ο γράμμα ο αλγόριθμος αρχίζει πάλι από το `l` του `lemon`. Άρα η μετατροπή έχει ως εξής:

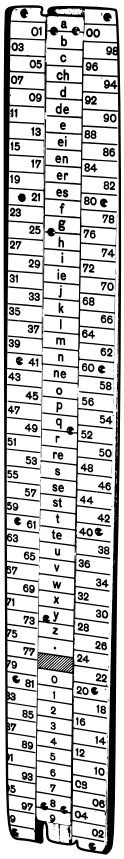
```
attackatdawn
lemonlemonle
LXFOPVEFRNHR □
```

Το κλειδί δεν είναι απαραίτητο να είναι υπαρκτή λέξη στη γλώσσα και είναι πιο ασφαλής η κρυπτογράφηση αν δεν είναι, αν δηλαδή είναι μια αυθαίρετη ακολουθία συμβόλων. Τότε υπάρχουν 26^m δυνατά κλειδιά. Στο παραπάνω παράδειγμα $26^5 = 11881376$.

[3] Ο πολλαπλασιαστικός αντίστροφος του $a \mod m$ είναι ένας αριθμός $a' \in \mathbb{Z}_m$ τέτοιος, ώστε να ισχύει $aa' = 1 \mod m$.

[4] Από τον διπλωμάτη, κρυπτογράφο και αλχημιστή Blaise de Vigenère (1523 – 1596), παρ' όλο που δεν ήταν ο πρώτος που εισηγήθηκε ένα τέτοιο κώδικα.

[5] Βλ. και Wikipedia, `https://en.wikipedia.org/wiki/Vigenère_cipher`



Η πρακτική χρήση ενός τέτοιου αλγόριθμου μπορεί στην πράξη να χρησιμοποιεί πίνακες και άλλα εργαλεία, που σε ορισμένες περιπτώσεις προσθέτουν στοιχεία στην πολυπλοκότητα του συστήματος.

Παράδειγμα 3. Στο περιθώριο φαίνεται μια *swiss reglette* (κρυπτογραφικός κανόνας ολίσθησης) που χρησιμοποιήθηκε στον ελβετικό στρατό μεταξύ 1914 και 1940.[6]

Η μεσαία λωρίδα ολισθαίνει μέσα σε ένα πλαίσιο που αποτελείται από την δεξιά και την αριστερή λωρίδα. Οι τελευταίες περιλαμβάνουν 100 σύμβολα, δηλαδή τους αριθμούς από 0 ως 99, από τους οποίους συντίθεται το κρυπτογραφημένο κείμενο. Η μεσαία κινούμενη λωρίδα περιλαμβάνει 50 σύμβολα, ήτοι 48 αλφαριθμητικά σύμβολα, μερικά από τα οποία είναι ζευγάρια γραμμάτων που απαντώνται στη φυσική γλώσσα, μαζί με ένα κενό και μια τελεία.

Ο κανόνας αυτός χρησιμοποιείται σε κρυπτογράφηση ως εξής: Το κλειδί αποτελείται από μια λέξη και έναν αριθμό από 0 ως 99 (για κείμενα μεγάλου μήκους δίνονται περισσότεροι αριθμοί). Π.χ. έστω ότι το κλειδί είναι `drehzahl,37` και ότι το κείμενο προς κρυπτογράφηση είναι `else`. Για να κρυπτογραφηθεί το πρώτο γράμμα `e` ευθυγραμμίζεται το πρώτο γράμμα του κλειδιού, εν προκειμένω το `d`, με τον αριθμό του κλειδιού, δηλ. το 37. Στη συνέχεια καταγράφεται ο αριθμός που βρίσκεται απέναντι από το `e`, δηλαδή το 60. Στη συνέχεια ευθυγραμμίζεται το δεύτερο γράμμα του κλειδιού, δηλ. το `r`, με το 37 και καταγράφεται ο αριθμός δίπλα στο `l` ($= \ell$), δηλαδή το 78. Για τα δυο επόμενα γράμματα `se` προκύπτουν οι αριθμοί 22 78. Ο συνολικός αριθμός 60782278 χωρίζεται σε πεντάδες, δηλαδή 60782 και 278.

Δείτε ότι εκ πρώτης όψεως χρησιμοποιείται ένα διπλό κλειδί (λέξη και αριθμός) ενώ στον κλασσικό Vigenère αρκεί μια λέξη. Ωστόσο σε μια πιο προσεκτική θεώρηση του συστήματος αυτού διαπιστώνουμε ότι κάθε ζευγάρι λέξης και αριθμού ισοδυναμεί με άλλα 99 ζευγάρια λέξεων-αριθμών, ένα ζευγάρι για κάθε αριθμό από το 0 ως το 99. Άρα ο αριθμός των δυνατών κλειδιών τελικά έχει να κάνει μόνο με το πρώτο συνθετικό κάθε κλειδιού, τη λέξη. Θεωρητικά θα μπορούσαμε να οριστικοποιήσουμε τον αριθμό, π.χ. να χρησιμοποιούμε πάντοτε 29, και να δίνουμε κάθε φορά μόνο μια άλλη λέξη. Π.χ. αν κρατήσουμε το 29, η λέξη που παράγει το ισοδύναμο αποτέλεσμα με το συνδυασμό `drehzahl,37` είναι η λέξη `a(ne)c(er)v6i`. Παρ' όλα αυτά πρακτικά η χρήση ενός «διπλού» κλειδιού βοηθάει στο εξής: Επιτρέπει στις λέξεις να είναι λέξεις της φυσικής γλώσσας (όπως παραπάνω η λέξη `drehzahl` = ταχύτητα περιστροφής), οι οποίες με τη βοήθεια ενός αριθμού μετατρέπονται σε σχεδόν τυχαίες ισοδύναμες λέξεις κλειδιά.

Να σημειώσετε όμως επίσης ότι κατά περίπτωση μακρύτερα κείμενα μπορούσαν να κρυπτογραφούνται κατά τμήματα (με κάποιο προσυνεννοημένο μήκος) χρησιμοποιώντας στο κλειδί περισσότερους από ένα διψήφιους στο αριθμητικό μέρος, αλλάζοντας το κλειδί από τμήμα σε τμήμα. Επίσης δείτε ότι το κρυπτογραφημένο κείμενο είναι αποκλειστικά αριθμητικό. Σε μια περαιτέρω απόπειρα να αυξηθεί η πολυπλοκότητα, η δεξιά πλευρά έχει τους αριθμούς σε αντίστροφη σειρά. Επίσης, ενώ τα προς κρυπτογράφηση σύμβολα είναι όσα φιλοξενούνται στην μεσαία λωρίδα, άρα 50, δεξιά υπάρχουν 100 αριθμοί. Όμως για κάθε κλειδί με συγκεκριμένο διψήφιο αριθμό προκύπτουν μόνο άρτιοι ή μόνο περιττοί αριθμοί στην δεξιά λωρίδα, εκτός αν γίνει κρυπτογράφηση με διαφορετικούς αριθμούς κατά τμήματα, αλλά και τότε διατηρείται η ιδιότητα αρτίων ή περιττών εσωτερικά σε κάθε τμήμα. Η ομαδοποίηση του τελικού αριθμητικού string σε πεντάδες βοηθάει στο να μη γίνεται αμέσως αντιληπτή αυτή η ιδιότητα. □

Κώδικας Vernam

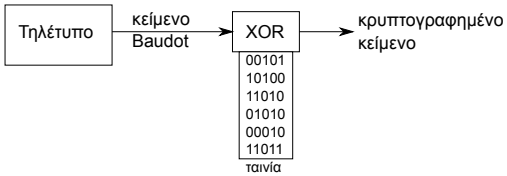
Ο Gilbert S. Vernam (1890-1960) ήταν μηχανικός στην AT&T (Americal Telephone and Telegraph Company). Το 1917 σχεδίασε μια τηλετυπική μηχανή με δυνατότητα να προφυλάσσει κρυπτογραφικά τα μηνύματα. Η μηχανή χρησιμοποιούσε μια εκδοχή

[6] Photograph by Rama, Wikimedia Commons, Cc-by-sa-2.0-fr / CC BY-SA 2.0 FR, `https://commons.wikimedia.org/wiki/File:Cryptographic_sliding_rule-IMG_0533.jpg`. Για την λειτουργία βλ. και `https://www.cryptomuseum.com/crypto/swiss/reglette/index.htm`, καθώς και στην [Rit02].

Binary	Decimal	Hex	Octal	Letter	Figure
00000	0	0	0	Blank	Blank
00001	1	1	1	T	5
00010	2	2	2	CR	CR
00011	3	3	3	O	9
00100	4	4	4	Space	Space
00101	5	5	5	H	
00110	6	6	6	N	,
00111	7	7	7	M	.
01000	8	8	10	Line Feed	Line Feed
01001	9	9	11	L	)
01010	10	A	12	R	4
01011	11	B	13	G	&
01100	12	C	14	I	8
01101	13	D	15	P	0
01110	14	E	16	C	:
01111	15	F	17	V	;
10000	16	10	20	E	3
10001	17	11	21	Z	"
10010	18	12	22	D	$
10011	19	13	23	B	?
10100	20	14	24	S	BEL
10101	21	15	25	Y	6
10110	22	16	26	F	!
10111	23	17	27	X	/
11000	24	18	30	A	-
11001	25	19	31	W	2
11010	26	1A	32	J	'
11011	27	1B	33	Figure Shift	
11100	28	1C	34	U	7
11101	29	1D	35	Q	1
11110	30	1E	36	K	(
11111	31	1F	37	Letter Shift	

Σχήμα 2.2: Ο κώδικας Baudot.

της 5-μπιτης κωδικοποίησης Baudot, που περιλαμβάνει αλφαριθμητικά και άλλα σύμβολα.[7] Σε κάθε τέτοια πεντάδα γινόταν δυαδική πρόσθεση μιας άλλης πεντάδας, η οποία ήταν γραμμένη σε μια κυκλικά επαναλαμβανόμενη ταινία, δηλαδή γινόταν XOR με μια σειρά από κυκλικά επαναλαμβανόμενες πεντάδες [Kon07].



Σχήμα 2.3: Η μηχανή του Vernam.

Στη συνέχεια ο Vernam βελτίωσε αυτή τη μηχανή προσθέτοντας περισσότερα του ενός διαδοχικά στάδια κρυπτογράφησης με ταινίες διαφορετικές μεταξύ τους ως προς τις πεντάδες που φιλοξενούσε κάθε μία, αλλά και διαφορετικού μήκους. Εν ολίγοις το αρχικό κείμενο περνούσε από XOR με ένα αρκετά μακρύ περιοδικά επαναλαμβανόμενο κλειδί. Στην περίπτωση των πολλαπλών διαδοχικών ταινιών μια προσεκτική

[7] Ο προσεκτικός αναγνώστης θα παρατηρήσει ότι με $2^5 = 32$ διαθέσιμους αριθμούς δεν μπορούν να κωδικοποιηθούν 26 αλφαβητικά σύμβολα και 10 αριθμητικά και μερικά άλλα σύμβολα. Αυτό ήταν κατορθωτό με τη χρήση δύο συμβόλων που είχαν την λειτουργία του `shift` για να περάσει κανείς από τα γράμματα στις «εικόνες» και αντίστροφα.

επιλογή των περιόδων των ταινιών μπορούσε να οδηγήσει σε περίοδο που αντιστοιχεί στο γινόμενο των περιόδων (δεν είναι όμως ισοδύναμη με μια ταινία αυτού του μήκους) [Tuc70].

Μηχανές με τροχούς

Στο μεσοπόλεμο (δηλαδή στο διάστημα ανάμεσα στον πρώτο και στον δεύτερο παγκόσμιο πόλεμο) αναπτύχθηκαν μηχανές βασισμένες σε ηλεκτρομηχανική τεχνολογία και εφοδιασμένες με μια σειρά περιστρεφόμενων τροχών (rotor machines). Οι μηχανές αυτές έπαιζαν με την ιδέα του συνεχώς μεταβαλλόμενου κλειδιού, όπως το one time pad, αλλά σε μια πιο πρακτική προσέγγιση. Η πιο διάσημη απ΄αυτές τις κρυπτογραφικές μηχανές είναι η γερμανική Enigma, για την οποία υποβλήθηκε πατέντα το 1918 (δηλαδή τον ίδιο καιρό που αναπτύχθηκε η μηχανή του Vernam) και κυκλοφόρησε για εμπορικές εφαρμογές με το όνομα Enigma από το 1923. Αντίστοιχες μηχανές ήταν η αγγλική Typex και η αμερικανική SIGABA. Η βασική ιδέα είναι για κάθε χαρακτήρα που πληκτρολογείται από το πληκτρολόγιο της μηχανής δημιουργείται ένα ηλεκτρικό κύκλωμα που φτάνει να ενεργοποιεί έναν άλλο χαρακτήρα σε ένα φωτιζόμενο αλφάβητο. Το φωτιζόμενο κάθε φορά γράμμα είναι το κωδικοποιημένο σύμβολο. Το κύκλωμα περνάει μέσα από τροχούς, καθένας από τους οποίους έχει 26 σημεία επαφών (pins) εισόδου και άλλα τόσα εξόδου με σταθερή καλωδίωση κάθε ζεύγους, αλλά η επιλογή των ζευγών εισόδου-εξόδου αποτελεί μέρος της κωδικοποίησης. Με κάθε πάτημα πλήκτρου ένας τουλάχιστον τροχός περιστρέφεται με αποτέλεσμα την μεταβολή του γράμματος που φωτίζεται ακόμη και όταν πατηθεί το ίδιο γράμμα εισόδου. Οι μηχανές αυτές διέφεραν ως προς τον αριθμό των τροχών μέσα στη μηχανή, το σύνολο των διαθέσιμων τροχών με τους οποίους μπορούσε να γίνει αντικατάσταση και άρα να αλλάξει η κωδικοποίηση, τον τρόπο περιστροφής των τροχών, καθώς και άλλες τροποποιήσεις με τις οποίες μπορούσαν να αυξηθούν οι δυνατοί συνδυασμοί κωδικοποίησης. Ο αναγνώστης μπορεί να διαβάσει περισσότερα για τις ιστορικές αυτές μηχανές στη βιβλιογραφία και στη Wikipedia.

Οι μηχανές με τροχούς, παρά την επιτυχία πολωνών αποκωδικοποιητών προπολεμικά στο να σπάσουν τις αρχικές εκδόσεις την Enigma και στη συνέχεια του Turing, γνώρισαν μεγάλη διάδοση στον δεύτερο παγκόσμιο πόλεμο. Ωστόσο μπαίνοντας στη μεταπολεμική είχαν μια σειρά μειονεκτημάτων: (α) ήταν αργές, ακόμη και μηχανοποίηση της εισόδου (π.χ. να διαβάζουν μια ταινία αντί να πληκτρολογείται το κείμενο εισόδου) δεν θα μπορούσαν να ξεπεράσουν λίγους χαρακτήρες ανά δευτερόλεπτο, (β) η καλωδίωση των τροχών έπρεπε να γίνεται από έμπιστο προσωπικό, (γ) η αρχική ρύθμιση των τροχών και άλλων στοιχείων έπρεπε κάθε φορά (συνήθως στην αρχή της μέρας) να μεταβάλλεται μέσα από ένα προκαταβολικά διανεμημένο κατάλογο ρυθμίσεων, (δ) η μηχανή ήταν δύσκολο να καταστραφεί σε περίπτωση που θα έπεφτε στα χέρια του εχθρού.

Κώδικας μετάθεσης

Ο κώδικας αυτός βασίζεται στην αναδιάταξη των γραμμάτων ενός κειμένου σύμφωνα με ένα προκαθορισμένο pattern, εφαρμοζόμενη σε τμήματα μήκους m. Έστω κείμενο μήκους m και μια μετάθεση $\mu = (\mu_0, \mu_1, \ldots, \mu_m)$ των στοιχείων του $\mathbb{Z}_m$. Η κρυπτογράφηση ενός κειμένου $(x_1, x_2, \ldots, x_m)$ σε κείμενο $(y_1, y_2, \ldots, y_m)$ γίνεται με τον τύπο

$$y_i = x_{\mu_i}, \quad i = 1, \ldots, m$$

Στη συνέχεια κρυπτογραφούνται με τον ίδιο τρόπο τα επόμενα m γράμματα του κειμένου ως την ολοκλήρωσή του. Εάν το μήκος του κειμένου δεν είναι πολλαπλάσιο του m χρειάζεται να γίνει προσθήκη γραμμάτων, π.χ. με κλήρωση. Η αποκρυπτογράφηση γίνεται με τον ίδιο τρόπο μέσω της αντίστροφης μετάθεσης μ^{-1}.

Παρατήρηση: Αυτός ο κώδικας μετάθεσης είναι διαφορετικός από τον κώδικα μετάθεσης της σελίδας 5, παρά την «συνωνυμία». Στον παρόντα κώδικα ανακατεύονται τα γράμματα κάθε m-άδας, χωρίς όμως να προκύπτουν νέα γράμματα εσωτερικά στη m-άδα. Αν π.χ. το κείμενο είναι `thelittledog` και $m = 5$, η πρώτη ομάδα μπορεί να κρυπτογραφηθεί σε `ihtel`, αλλά δεν μπορεί να υπάρξει εκεί ένα άλλο γράμμα εκτός από αυτά που υπάρχουν ήδη στην ομάδα, όπως το `a`. Το ίδιο γράμμα σε άλλη ομάδα μπορεί να κρυπτογραφηθεί σε άλλο γράμμα. Αντίθετα, στον κώδικα της σελίδας 5 κάθε γράμμα κρυπτογραφείται σταθερά σε ένα συγκεκριμένο γράμμα και μπορεί να πάρει οποιαδήποτε τιμή από τις υπόλοιπες 25.

Κώδικας Hill

Ο κώδικας επινοήθηκε το 1926 από τον αμερικανό μαθηματικό Lester S. Hill (1891–1961). Έστω x ένα διάνυσμα μήκους $m \geq 2$ με συνιστώσες από το $\mathbb{Z}_{26}$, οπότε $x \in (\mathbb{Z}_{26})^m$. Η κρυπτογράφηση προσδιορίζει ένα διάνυσμα $y \in (\mathbb{Z}_{26})^m$ ίδιου μήκους με το αρχικό μέσω του τύπου

$$y = e_K(x) = xK$$

όπου το «κλειδί» K είναι ένας τετραγωνικός πίνακας διαστάσεων $m \times m$ αναστρέψιμος στο $\mathbb{Z}_{26}$ και όλες οι πράξεις γίνονται εσωτερικά στο $\mathbb{Z}_{26}$. Η αποκρυπτογράφηση γίνεται ως

$$d_K(y) = yK^{-1}$$

Βασική προϋπόθεση για να μπορεί να αξιοποιηθεί ένας πίνακας K στον κώδικα Hill είναι να είναι αναστρέψιμος μέσα στο $\mathbb{Z}_{26}$ και για την ιδιότητα αυτή ισχύουν συγκεκριμένες συνθήκες. Ο πιο πάνω κώδικας μετάθεσης είναι ειδική περίπτωση του κώδικα Hill, αρκεί να κατασκευάσουμε ένα πίνακα K με μια μόνο μονάδα ανά σειρά και στήλη (και όλα τα άλλα στοιχεία μηδενικά) με τρόπο που να υλοποιείται η μετάθεση. Σε αυτήν την περίπτωση είναι επίσης εύκολη η κατασκευή του πίνακα της αποκρυπτογράφησης ακολουθώντας την αντίστροφη μετάθεση. Ο πίνακας αυτός αντιστοιχεί στον K^{-1}, επομένως ένας πίνακας K που υλοποιεί μια μετάθεση είναι πάντοτε αναστρέψιμος.

2.3 Τέλεια μυστικότητα

Οι κρυπτογράφοι ήδη πριν από τον Β' παγκόσμιο πόλεμο δέχονταν, έστω εμπειρικά, ότι το one time pad ήταν απαραβίαστο, δηλαδή η κρυπτογράφηση με τυχαίο κλειδί αρκετά μακρύ όσο να καλύπτει το κείμενο. Το 1949 ο Claude Shannon δημοσίευσε ένα άρθρο [Sha49], στο οποίο εξήγησε πότε ένας κώδικας είναι απαραβίαστος απέναντι σε έναν αντίπαλο που διαθέτει μόνο δείγματα από το κρυπτογραφημένο κείμενο. Η θεωρία του Shannon δείχνει για παράδειγμα ότι ο κώδικας Vigenère είναι απαραβίαστος αν το κλειδί με το οποίο κρυπτογραφείται ένα κείμενο χρησιμοποιείται μόνο μια φορά και έχει μήκος (τουλάχιστον) ίσο με το κείμενο.

Έστω τυχαία μεταβλητή X που παίρνει ως τιμή ένα κείμενο x προς κρυπτογράφηση και τυχαία μεταβλητή K που που παίρνει ως τιμή ένα κλειδί k. Για τις δύο αυτές τ.μ. θα υποθέτουμε στη συνέχεια ότι είναι μεταξύ τους ανεξάρτητες και ότι

παίρνουν τιμές από τα σύνολα $\mathcal{P}$ και $\mathcal{K}$ αντίστοιχα. Θεωρούμε επίσης τις συναρτήσεις κρυπτογράφησης και αποκρυπτογράφησης $e_k(x), d_k(x)$ με την γνωστή ιδιότητα αντιστροφής $d_k(e_k(x)) = x$ και περαιτέρω ορίζεται το σύνολο $\mathcal{C}$ των δυνατών κρυπτογραφημένων κειμένων. Έστω ότι $C(k)$ είναι το υποσύνολο των κρυπτογραφημένων κειμένων που μπορούν να παραχθούν με ένα συγκεκριμένο κλειδί k, δηλαδή $C(k) = \{e_k(x) : x \in \mathcal{P}$. Δεδομένων των κατανομών (συναρτήσεων μάζας πιθανότητας) των X και K η υπό συνθήκη κατανομή της τ.μ. του κρυπτογραφημένου κειμένου Y μπορεί να υπολογισθεί ως

$$\Pr[Y = y|X = x] = \sum_{k:y\in C(k)} \Pr[K = k]$$

και η κατανομή της Y ως

$$\Pr[Y = y] = \sum_{k:y\in C(k)} \Pr[K = k] \Pr[X = d_k(y)]$$

Μπορούμε τώρα να βρούμε την αντίστροφη υπό συνθήκη πιθανότητα με τον νόμο του Bayes:

$$\begin{aligned} \Pr[X = x|Y = y] &= \frac{\Pr[X = x, Y = y]}{\Pr[Y = y]} = \frac{\Pr[X = x] \Pr[Y = y|X = x]}{\Pr[Y = y]} \\ &= \frac{\Pr[X = x] \sum_{k:y\in C(k)} \Pr[K = k]}{\sum_{k:y\in C(k)} \Pr[K = k] \Pr[X = d_k(y)]} \end{aligned}$$

Εάν ένα κρυπτογραφημένο δείγμα δεν προσθέτει πληροφορία για το ποιο μπορεί να είναι το αντίστοιχο ακρυπτογράφητο κείμενο, τότε θεωρείται (κατά Shannon, εξ ορισμού) ότι υπάρχει τέλεια μυστικότητα:

Ορισμός 1. Ένα κρυπτοσύστημα έχει «τέλεια μυστικότητα» αν

$$\Pr\{X = x|Y = y\} = \Pr\{X = x\}. \quad \square$$

Εφαρμογή στον Κώδικα Ολίσθησης

Για τον *κώδικα ολίσθησης* (shift cipher, βλ. σελ. 5) έστω ότι χρησιμοποιούμε ένα από τα δυνατά 26 κλειδιά ισοπίθανα, δηλ. $\Pr[K = k] = 1/26$. Η κατανομή του κρυπτογραφημένου κειμένου υπολογίζεται ως

$$\begin{aligned} \Pr[Y = y] &= \sum_{k\in\mathbb{Z}_{26}} \Pr[K = k] \Pr[X = d_k(y)] \\ &= \frac{1}{26} \sum_{k\in\mathbb{Z}_{26}} \Pr[X = y - k] \\ &= \frac{1}{26} \sum_{k\in\mathbb{Z}_{26}} \Pr[X = k] = \frac{1}{26} \end{aligned}$$

Επίσης για την υπό συνθήκη πιθανότητα της εξόδου ως προς την είσοδο ισχύει

$$\Pr[Y = y|X = x] = \Pr[K = y - x \mod 26] = \frac{1}{26}$$

οπότε χρησιμοποιώντας την αντιστροφή Bayes

$$\Pr[X = x|Y = y] = \frac{\Pr[X = x]}{\Pr[Y = y|X = x]}\Pr[Y = y] = \frac{\Pr[X = x]\frac{1}{26}}{\frac{1}{26}} = \Pr[X = x]$$

Επομένως ο Κώδικας Ολίσθησης (shift cipher) έχει τέλεια μυστικότητα, δηλαδή είναι απαραβίαστος εφόσον για κάθε νέο χαρακτήρα του αρχικού κειμένου χρησιμοποιείται ένα νέο τυχαίο κλειδί.

Γιατί μια εξαντλητική αναζήτηση του κλειδιού δεν θα καταφέρει να αποκρυπτογραφήσει το κείμενο; Ας υποτεθεί ότι το κείμενο έχει μήκος n. Κατά συνέπεια τα δυνατά κλειδιά για όλο μαζί το κείμενο είναι 26^n. Όταν αυτά δοκιμασθούν θα προκύψουν όλα τα δυνατά κείμενα μήκους n. Το εξής γενικότερο θεώρημα αποδεικνύεται εύκολα με την ίδια λογική όπως παραπάνω:

Θεώρημα 1. Έστω ότι σε ένα κρυπτοσύστημα το σύνολο κλειδιών $\mathcal{K}$, το σύνολο των ακρυπτογράφητων κειμένων $\mathcal{P}$ και το σύνολο των κρυπτογραφημένων κειμένων $\mathcal{C}$ έχουν ίσους πληθικούς αριθμούς. Εάν το κλειδί επιλέγεται με ίδια πιθανότητα $1/|\mathcal{K}|$ και εάν για κάθε ζεύγος $(x, y) \in \mathcal{P} \times \mathcal{C}$ υπάρχει ένα μοναδικό κλειδί k τέτοιο ώστε $e_k(x) = y$, το κρυπτοσύστημα παρέχει τέλεια μυστικότητα. □

Το παραπάνω θεώρημα εξηγεί γιατί είναι απαραβίαστο το *One Time Pad* (σημειωματάριο μιας χρήσης), ένας αλγόριθμος κρυπτογράφησης που ήταν γνωστός από το 1917 και είχε προταθεί από τον Vernam.

Σημειωματάριο μιας χρήσης (One Time Pad)

Έστω ότι σε ένα κρυπτοσύστημα το σύνολο κλειδιών $\mathcal{K}$, το σύνολο των ακρυπτογράφητων κειμένων $\mathcal{P}$ και το σύνολο των κρυπτογραφημένων κειμένων $\mathcal{C}$ είναι όλα ίσα με το $(\mathbb{Z}_{26})^n$ (δηλαδή αποτελούνται από όλα τα δυνατά δυαδικά strings μήκους n). Η κρυπτογράφηση του string εισόδου x με το κλειδί k γίνεται ως $x \oplus k$ και η αποκρυπτογράφηση ως $y \oplus k$.

Την ιδέα για το One Time Pad είχε μερικές δεκαετίες πιο πριν ο Frank Miller, τραπεζίτης στο Sacramento [Bel11]. Η μηχανή του Vernam (βλ. Σχ. 2.3) θα μπορούσε να θεωρηθεί υλοποίηση του One Time Pad αν υποτεθεί ότι η ταινία είναι μήκους μεγαλύτερου από το κείμενο και δεν επαναχρησιμοποιηθεί σε άλλο κείμενο. Ο στρατιωτικός Joseph Oswald Mauborgne (1881 – 1971) και συνεφευρέτης του One Time Pad πρότεινε αργότερα τη χρήση του με κείμενο.

2.4 Κώδικες ροής

Ας υποτεθεί ότι στην είσοδο ενός συστήματος κρυπτογράφησης εμφανίζεται ένα κείμενο, που παράγεται από μια πηγή που γεννάει συνεχώς νέα σύμβολα και πρέπει να μεταφέρονται κρυπτογραφημένα στον προορισμό τους (δηλαδή το κείμενο είναι κατά κάποιο τρόπο <απείρου» μήκους). Η προφανής λύση, με βάση όσα έχουμε δει ως τώρα, είναι η τμηματική κρυπτογράφηση, δηλαδή να χωρίζεται το ακρυπτογράφητο κείμενο (plaintext) σε ομάδες των n συμβόλων και να γίνεται κρυπτογράφηση κάθε ομάδας χωριστά με γνωστό αλγόριθμο σαν αυτούς που έχουμε δει ως τώρα. Ένα ζήτημα, του οποίου η σημασία ποικίλλει ανά εφαρμογή, είναι πως στην έξοδο παρουσιάζεται μια καθυστέρηση ίση με την απόσταση των χρόνων εισόδου μεταξύ πρώτου και τελευταίου συμβόλου μιας ομάδας. Η λύση είναι ίσως να χρησιμοποιούνται μικρότερες ομάδες, αλλά κάτι τέτοιο εν γένει μειώνει την ασφάλεια. Πιο σημαντικό σε κείμενα

πολύ μεγάλου μήκους ίσως είναι το ζήτημα της διαχείρισης των κλειδιών. Αν το κλειδί παραμένει το ίδιο, έστω K, σε όλες τις ομάδες, ένας πιθανός αντίπαλος συσσωρεύει όλο και περισσότερα στοιχεία για να επιτύχει τους σκοπούς του. Αν χρησιμοποιείται μια αλλαγή κλειδιών κάθε τόσο, πρέπει να φροντίσουν οι δύο πλευρές που επικοινωνούν να έχουν ανταλλάξει στο παρελθόν αρκετά κλειδιά ή να έχουν στη διάθεσή τους ένα τρόπο να συμφωνούν σε νέα κλειδιά στη διάρκεια της επικοινωνίας.

Στους λεγόμενους *κώδικες ροής* ή *ρεύματος* (stream codes) ο στόχος είναι για κάθε εισερχόμενο σύμβολο στο σύστημα κρυπτογράφησης (ή αποκρυπτογράφησης) να παράγεται ένα σύμβολο στην έξοδο. Αν $\mathbf{x} = x_1x_2\ldots$ είναι μια σειρά συμβόλων (string) που εμφανίζονται στην είσοδο, $\mathbf{y} = y_1y_2\ldots$ η αντίστοιχη σειρά στην έξοδο και $\mathbf{z} = z_1z_2\ldots$ είναι μια σειρά συμβόλων που χρησιμοποιείται ως κλειδί, ισχύει γενικά ότι

$$y_i = e_{z_1,z_2,\ldots,z_i}(x_1, x_2, \ldots, x_i) \tag{2.1}$$

αν τα κλειδιά $\mathbf{z}$ είναι ανεξάρτητα της εισόδου και της εξόδου. Ωστόσο μπορεί τα κλειδιά να δημιουργούνται από μια αρχική ακολουθία κλειδιών περιορισμένου μήκους $\mathbf{Z} = z_1z_2\ldots z_N$ και στη συνέχεια οι επόμενες τιμές να εξαρτώνται από την είσοδο ή και την έξοδο. Σ' αυτήν την περίπτωση

$$y_i = e_{\mathbf{Z}}(x_1, x_2, \ldots, x_i) \tag{2.2}$$

Μια συνηθισμένη μορφή αυτού του προβλήματος είναι τα σύμβολα να είναι δυαδικά, δηλαδή η πηγή να παράγει ένα ρεύμα από bits, και μια συνηθισμένη λύση είναι να έχει στη διάθεσή της ένα δεύτερο ρεύμα, γνωστό και στις δύο πλευρές, με το οποίο κάνει bit προς bit XOR το πρώτο ρεύμα. Ένας τρόπος να δημιουργηθεί το δεύτερο ρεύμα είναι να χρησιμοποιηθεί μια γεννήτρια τυχαίων αριθμών με μια γνωστή και στις δύο πλευρές τιμή αρχικοποίησης (seed) που λειτουργεί ως κλειδί.

Σε μια γενίκευση για σύμβολα παρμένα μέσα από ένα αλφάβητο με περισσότερα σύμβολα, ο κανόνας κωδικοποίησης για το i-οστό σύμβολο μπορεί να είναι $y_i = e_{z_i}(x_i)$, δηλαδή να εξαρτάται μόνο από την τρέχουσα τιμή του κλειδιού.

Έστω ότι

$$\mathbf{z} = k_1k_2\ldots k_mk_1k_2\ldots k_mk_1\ldots$$

δηλαδή το ρεύμα των κλειδιών παράγεται από την περιοδική επανάληψη των συμβόλων $k_1k_2\ldots k_m$ ενός κλειδιού μήκους m. Έστω επίσης ότι η κρυπτογράφηση γίνεται με τον κώδικα Vigenère. Στην περίπτωση αυτήν έχουμε ουσιαστικά ένα τμηματικό κώδικα μήκους m που χρησιμοποιεί τον κώδικα Vigenère ανά τμήμα. Επειδή όμως στον Vigenère η επεξεργασία κάθε συμβόλου γίνεται χωριστά, δεν υπάρχει καθυστέρηση τμήματος.

Μια λύση για να αποφευχθεί η μικρή περίοδος που εμφανίζεται στο κλειδί στο παραπάνω παράδειγμα είναι να χρησιμοποιηθεί em αναδρομή βαθμού m. Δηλαδή ξεκινώντας από ένα αρχικό κλειδί $k_1k_2\ldots k_m$ η ακολουθία κλειδιών z_i μπορεί να παράγεται με τον αναδρομικό τύπο

$$z_{i+m} = \sum_{j}^{m-1} c_j z_{i+j} \mod n$$

όπου n είναι ο πληθικός αριθμός του αλφαβήτου της πηγής και χρειάζονται επί πλέον οι σταθερές $c_0, \ldots, c_{m-1}$. Στα περισσότερα από τα επόμενα παραδείγματα τα σύμβολα θα είναι bits, δηλαδή $n = 2$.

Παράδειγμα 4. Αν το κλειδί είναι $K = 10010$ και $z_i = z_{i-2} + z_{i-4}$ και το κείμενο είναι 100010101001001, η κρυπτογράφηση έχει ως εξής

x =100010101001001
z =100101000101000
y =000111101100001 □

Ένας κώδικας που το ρεύμα κλειδιών βασίζεται γενικά σε προηγούμενες τιμές κλειδιών, εισόδου και εξόδου, λέγεται μη-σύγχρονος. Μια τέτοια περίπτωση είναι ο *Κώδικας Αυτόματου Κλειδιού* (Autokey Cipher) που βασίζεται στον Vigenère.

Κώδικας Αυτόματου Κλειδιού (Autokey Cipher)

Με αλφάβητο εισόδου, εξόδου και κλειδιών το $\mathbb{Z}_{26}$, αρχικά τίθεται $z_1 = K$ και στη συνέχεια $z_i = x_{i-1}$ για $i \geq 2$. Η κρυπτογράφηση γίνεται μέσω του τύπου

$$e_z(x) = (x + z) \mod 26$$

και η αποκωδικοποίηση ως

$$d_z(y) = (y - z) \mod 26$$

Παράδειγμα 5. Έστω $K = 4$ και το κείμενο εισόδου είναι attackatdawn (μετατρεπόμενο στους αριθμούς $0, 19, 19, 0, 2, 10, 0, 19, 3, 0, 22, 13$), οι τιμές της εισόδου, των κλειδιών και της εξόδου είναι:

x =00 19 19 00 02 10 00 19 03 00 22 13
z =04 00 19 19 00 02 10 00 19 03 00 22
y =04 19 12 19 02 12 10 19 22 03 22 09

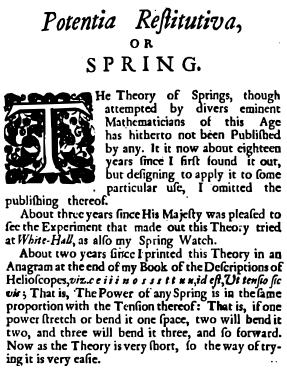
Άρα το κρυπτογραφημένο κείμενο είναι ETMTCMKTWDWJ. □

2.5 Δίκτυα υποκατάστασης-μετάθεσης (SPN)

Παρ' όλο που υπάρχουν κρυπτογραφικοί αλγόριθμοι που στηρίζονται στη μαθηματική θεωρία, όπως οι αλγόριθμοι RSA και Diffie-Hellmann στην περιοχή της μη συμμετρικής κρυπτογραφίας, στη συμμετρική κρυπτογραφία οι περισσότεροι αλγόριθμοι βασίζονται στον επανειλημμένη χρήση απλών σχετικά μεθόδων, αλλά σε μεγάλη ποσότητα επαναλήψεων. Κατά κάποιο τρόπο δηλαδή και η πλευρά της «άμυνας» στην ασφάλεια υιοθέτησε τα ίδια εργαλεία όπως η πλευρά της «επίθεσης», ήτοι την brute force. Οι σύγχρονοι κώδικες συμμετρικής κρυπτογράφησης δεν βασίζονται σε σπουδαία μαθηματικά θεωρήματα και μεθόδους. Χρησιμοποιούν την απλή ιδέα της επανάληψης της κρυπτογράφησης σε πολλούς διαδοχικούς γύρους, όπου σε κάθε γύρο ως είσοδος χρησιμοποιείται η έξοδος του προηγούμενου. Εσωτερικά σε ένα γύρο χρησιμοποιούνται γνωστές εδώ και αιώνες απλές μέθοδοι. Ήδη από την εποχή του Καίσαρα (αν όχι και παλιότερα) είναι γνωστή η ιδέα της αντικατάστασης ενός συμβόλου με ένα άλλο βάσει ενός κανόνα. Μια άλλη παλιά ιδέα είναι αυτή της διατήρησης των ίδιων συμβόλων του αρχικού κειμένου, αλλά της τοποθέτησής τους σε άλλη διάταξη.

Έχουμε δει ήδη τους κώδικες μετάθεσης, αλλά μια διάσημη χρήση μετάθεσης είναι γνωστή από το 1678 από τον Robert Hooke. Ο φυσικός που είναι διάσημος για τον νόμο που φέρει το όνομά του[8] είχε ανακαλύψει τη σχετική συμπεριφορά των ελατηρίων, αλλά δεν ήθελε να δώσει το συμπέρασμά του στη δημοσιότητα πριν να ολοκληρώσει τη μελέτη του. Το 1676 έγραψε στο τέλος ενός συγγράμματός του ότι είχε βρει

[8]Νόμος του Hooke: $F = kx$, όπου F είναι η δύναμη που εφαρμόζεται πάνω σε ένα ελατήριο και x είναι η επιμήκυνσή του, ενώ k είναι μια σταθερά.



Potentia Reſtitutiva,
OR
SPRING.

THe Theory of Springs, though attempted by divers eminent Mathematicians of this Age has hitherto not been Publiſhed by any. It it now about eighteen years ſince I firſt found it out, but deſigning to apply it to ſome particular uſe, I omitted the publiſhing thereof.

About three years ſince His Majeſty was pleaſed to ſee the Experiment that made out this Theory tried at *White-Hall*, as alſo my Spring Watch.

About two years ſince I printed this Theory in an Anagram at the end of my Book of the Deſcriptions of Helioſcopes, *viz. ceiiinosssttuu, id eſt, Ut tenſio ſic vis*; That is, The Power of any Spring is in the ſame proportion with the Tenſion thereof: That is, if one power ſtretch or bend it one ſpace, two will bend it two, and three will bend it three, and ſo forward. Now as the Theory is very ſhort, ſo the way of trying it is very eaſie.

ένα νέο νόμο, αλλά ότι για την ώρα μπορούσε μόνο να δώσει μια κωδική λέξη που στο μέλλον θα χρησίμευε ως απόδειξη ότι γνώριζε το νόμο. Η λέξη ήταν *ceiiinosssttuv*, που δυο χρόνια αργότερα προέκυψε ότι ήταν αναγραμματισμός της φράσης *ut tensio sic vis*, δηλαδή *όπως η επιμήκυνση, έτσι η δύναμη*, όταν δημοσιοποιήθηκε το σύγγραμμά του με τίτλο *Lectures de Potentia Restitutiva, Or of Spring Explaining the Power of Springing Bodies* [Hoo78], βλ. και σχετικό απόσπασμα από το πρωτότυπο στο περιθώριο.

Βέβαια στον αναγραμματισμό του Hooke η σειρά των συμβόλων προσδιορίζεται από τη θέση τους στο αλφάβητο, ενώ στην κρυπτογραφία χρησιμοποιείται ένας κανόνας μετάθεσης ανεξάρτητος από το κείμενο.

Επιστρέφοντας στο θέμα της χρήσης πολλών γύρων κρυπτογράφησης, η μέθοδος με την οποία το αρχικό κείμενο (string) x κρυπτογραφείται τελικά στο y μέσα από ένα σχήμα N γύρων θα μπορούσε να περιγραφεί με το ακόλουθο σχήμα

$$\begin{aligned} z^0 &= x \\ z^1 &= e_{K^1}(z^0) \\ z^2 &= e_{K^2}(z^1) \\ &\dots \\ z^N &= e_{K^N}(z^{N-1}) \\ y &= z^N \end{aligned} \tag{2.3}$$

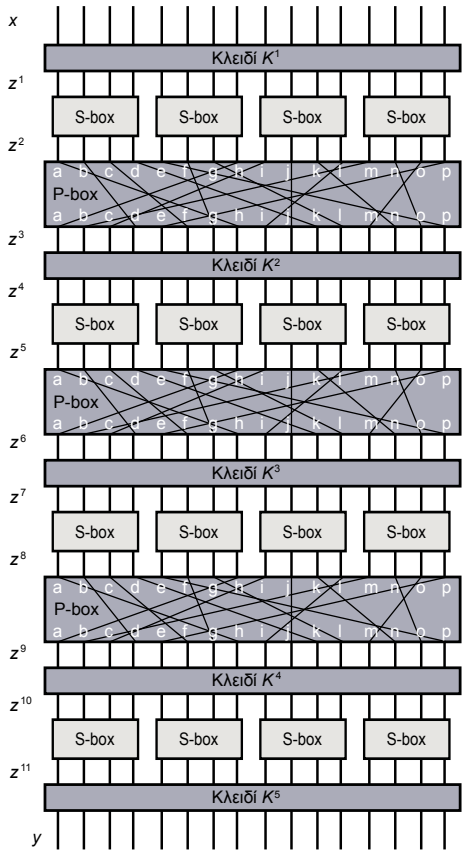
όπου $e_k(x)$ είναι μια συνάρτηση που κρυπτογραφεί ένα κείμενο x με ένα κλειδί k. Στο σχήμα αυτό υπεισέρχονται επίσης τα κλειδιά $K^1, K^2, \dots, ,$ τα οποία θα μπορούσαν μεν να δοθούν ως N ανεξάρτητα strings, αυξάνοντας έτσι το μήκος του συνολικού κλειδιού, αλλά πρακτικά συνήθως παράγονται με κάποια δημόσια μέθοδο από ένα αρχικό κλειδί K, πιθανώς μεγαλύτερο από καθένα από τα K_i, αλλά οπωσδήποτε μικρότερο από το άθροισμα των μηκών τους.

Η αποκωδικοποίηση μπορεί να γίνει πάλι με γύρους, αρκεί η συνάρτηση $e_k(x)$ να είναι αναστρέψιμη, δηλαδή να υπάρχει η $d_K(x)$ με την ιδιότητα αντιστροφής $d_K(e_k(x)) = x$. Τότε προφανώς μπορεί να χρησιμοποιηθεί το σχήμα:

$$\begin{aligned} z^N &= y \\ z^{N-1} &= d_{K^N}(z^{N-1}) \\ &\dots \\ z^0 &= d_{K^1}(z^1) \\ x &= z^0 \end{aligned}$$

Στην επόμενη ενότητα εξηγούμε πώς η συνάρτηση κρυπτογράφησης ενός γύρου μπορεί να κατασκευαστεί από ένα συνδυασμό συναρτήσεων υποκατάστασης και μετάθεσης. Κάθε μια από αυτές τις λειτουργίες μπορεί να ανατεθεί σε ένα υποσύστημα, ένα «κουτί», που θα τις υλοποιεί για ένα συγκεκριμένο μήκος εισόδου και εξόδου. Ένα σύστημα που αποτελείται από τέτοια δομικά στοιχεία λέγεται *δίκτυο υποκατάστασης-μετάθεσης* (substitution-permutation network, SPN). Ακολουθεί αμέσως ένα παράδειγμα τέτοιου δικτύου.

Παράδειγμα 6. Στο Σχ. 2.4 φαίνεται ένα SPN που κωδικοποιεί 16μπιτες ακολουθίες. Στο συγκεκριμένο δίκτυο το τμήμα υποκατάστασης (S-box), το οποίο δέχεται τετράδες από bits, αντικαθιστά τον αριθμό που αντιστοιχεί στην τετράδα με τον επόμενο mod 16, δηλαδή το 0000 με το 0001, το 0001 με το 0010, κ.λπ. και εν τέλει τον 1111 με 0000. Το τμήμα μετάθεσης (P-box) λειτουργεί με βάση τον πίνακα



Σχήμα 2.4: Παράδειγμα SPN.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
h	d	f	k	p	g	l	a	c	j	n	i	b	o	m	e

δηλαδή το bit που είναι στη θέση a μεταφέρεται στη θέση h κ.ο.κ.

Στο δίκτυο χρησιμοποιούνται 5 κλειδιά $K^1, \ldots, K^5$ που φαίνονται στον Πίνακα 2.1. Στην περίπτωση που κάποιος χρησιμοποιήσει αυτόν τον αλγόριθμο κρυπτογράφησης είτε θα χρειαστεί ένα συνολικό κλειδί μήκους $5 \times 16 = 80$ bits, είτε θα χρησιμοποιήσει ένα κλειδί μικρότερου μήκους, από το οποίο με κάποιον αλγόριθμο θα παραχθούν τα 5 κλειδιά. Στο παράδειγμα υποθέτουμε ότι το βήμα αυτό έχει γίνει ήδη.

Η εξέλιξη των σχετικών υπολογισμών φαίνεται στον Πίνακα 2.1. Το αρχικό string που είναι το

$$x^1 = 0000\,0000\,0000\,0000,$$

οδηγείται σε XOR με το κλειδί

$$K^1 = 0011\,1010\,1001\,0100$$

και προκύπτει το z^1. Στη συνέχεια αυτό το string εισάγεται ανά τετράδα από bits στα 4 κουτιά υποκατάστασης, καθένα από τα οποία λειτουργεί με τον ίδιο τρόπο, όπως έχει περιγραφεί παραπάνω. Π.χ. τα πρώτα από αριστερά 4 bits του z^1 είναι $(z_1^1, z_2^1, z_3^1, z_4^1) = (0, 0, 1, 1)$, δηλ. ο δυαδικός αριθμός 0011, ο οποίος αντικαθίσταται στο z^2 από τον επόμενο δυαδικό, δηλ. τον 0100, οπότε $(z_1^2, z_2^2, z_3^2, z_4^2) = (0, 1, 0, 0)$. Ομοίως σχηματίζονται οι λοιπές τρεις 4δες του z^2 από τις αντίστοιχες τετράδες του z^1. η 16άδα του z^3 σχηματίζεται από μετάθεση των bits του

x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K^1	0	0	1	1	1	0	1	0	1	0	0	1	0	1	0	0
z^1	0	0	1	1	1	0	1	0	1	0	0	1	0	1	0	0
z^2	0	1	0	0	1	0	1	1	1	0	1	0	0	1	0	1
z^3	1	0	1	1	1	0	0	0	0	0	0	1	0	1	1	1
K^2	1	0	1	0	1	0	0	1	0	1	0	0	1	1	0	1
z^4	0	0	0	1	0	0	0	1	0	1	0	1	1	0	1	0
z^5	0	0	1	0	0	0	1	0	0	1	1	0	1	0	1	1
z^6	0	1	0	0	1	1	0	0	0	1	0	1	1	1	0	0
K^3	1	0	0	1	0	1	0	0	1	1	0	1	0	1	1	0
z^7	1	1	0	1	1	0	0	0	1	0	0	0	1	0	1	0
z^8	1	1	1	0	1	0	0	1	1	0	0	1	1	0	1	1
z^9	1	1	1	1	1	1	0	1	1	0	0	0	1	0	0	1
K^4	0	1	0	0	1	1	0	1	0	1	1	0	0	0	1	1
z^{10}	1	0	1	1	0	0	0	0	1	1	1	0	1	0	1	0
z^{11}	1	1	0	0	0	0	0	1	1	1	1	1	1	0	1	1
K^5	1	1	0	1	0	1	1	0	0	0	1	1	1	1	1	1
y	0	0	0	1	0	1	1	1	1	1	0	0	0	1	0	0

Πίνακας 2.1: Εξέλιξη των υπολογισμών στο SPN του Σχ. 2.4.

z^2 σύμφωνα με τον πίνακα μετάθεσης που δόθηκε πιο πάνω, δηλαδή τα πρώτα από αριστερά bits που βγαίνουν από τις θέσεις εξόδου a, b, c, d προέρχονται από τις θέσεις εισόδου h, m, i, b. Με άλλα λόγια για την τετράδα αυτήν ισχύει $(z_1^3, z_2^3, z_3^3, z_4^3) = (z_8^2, z_{13}^2, z_9^2, z_2^2)$. Η συνέχεια είναι ανάλογη. Δείτε όμως ότι η 16άδα z^{10} σύμφωνα με το Σχ. 2.4 περνάει μεν από φάση υποκατάστασης, δεν ακολουθεί όμως άλλη φάση μετάθεσης. Η έξοδος οδηγείται απ' ευθείας σε ένα τελικό XOR με το 5ο κλειδί.

Είσοδος	Έξοδος	Hamming d.
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 1 0 1 1 1 1 1 0 0 0 1 0 0	
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1	0 1 1 0 1 1 1 1 1 1 0 0 1 1 1 0	6
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0	0 1 1 0 0 1 1 1 1 1 0 0 0 1 0 0	3
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1	1 1 0 1 1 0 1 1 1 1 1 0 1 1 1 0	8
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0	1 1 0 1 0 1 1 1 1 1 0 0 0 1 0 0	5
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1	0 0 1 0 1 1 1 1 1 1 0 0 1 1 1 0	7
0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0	0 0 1 0 0 1 1 1 1 1 0 0 0 1 0 0	3
0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1	0 0 0 1 1 1 1 1 1 1 0 0 1 1 1 0	5
0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0	0 0 0 1 0 0 1 1 1 1 0 0 0 1 0 0	5

Πίνακας 2.2: Έξοδος από διαφορετικές εισόδους στο SPN του Σχ. 2.4.

Ένα ενδιαφέρον ερώτημα είναι σε ποιο βαθμό η έξοδος διαφοροποιείται όταν αλλάξει η είσοδος. Είδαμε ήδη ότι το μηδενικό string x παράγει ένα αρκετά διαφορετικό string $y = 0001011111000100$. Στον Πίνακα 2.2 φαίνονται τα παραγόμενα αποτελέσματα όταν στην είσοδο εμφανίζονται οι δυαδικοί αριθμοί από 0 ως 8 και στην τελευταία στήλη υπολογίζεται κάθε φορά η απόσταση Hamming από την προηγούμενη έξοδο (δηλαδή ο αριθμός των bits που είναι διαφορετικά). □

2.6 Data Encryption Standard

Οι κώδικες του Horst Feistel

Ο κρυπτογράφος Horst Feistel γεννήθηκε στο Βερολίνο το 1915. Μετανάστευσε στις ΗΠΑ το 1934, πήρε πρώτο πτυχίο από το MIT και Master από το Harvard, αμφότερα στη φυσική. Τον καιρό του πολέμου ήταν προληπτικά σε κατ' οίκον περιορισμό εξ αιτίας της καταγωγής του. Όμως το Γενάρη του 1944 πήρε την αμερικανική υπηκοότητα και αμέσως μετά την εξουσιοδότηση να εμπλακεί σε θέματα ασφάλειας (security clearance) και να εργασθεί ως ερευνητής για την πολεμική αεροπορία. Αργότερα, το 1973 και ενώ εργαζόταν στην IBM, έγραψε ένα άρθρο για τη σημασία της κρυπτογραφίας [Fei73] σε εμπορικές εφαρμογές, παρουσιάζοντας κώδικες βασισμένους σε υποκατάσταση και μετάθεση και πιο συγκεκριμένα τον κώδικα *Lucifer*. Συνοψίζοντας το αποτέλεσμα έγραψε: *The basic cryptographic transformation amounts to no more than random addition of single bits. The method derives its strength solely from the fact that for each message digit we completely and randomly change the key. This is the only class that one can prove to be undecipherable in the absolute sense.* Στο σημείο αυτό ίσως θα έγραφε πως μια παρόμοια ιδέα είχε μπει σε εφαρμογή στην Enigma, αν δεν απέφευγε να θυμίζει τη γερμανική του καταγωγή.

Ωστόσο η ιδέα για την εφαρμογή της κρυπτογραφίας σε εμπορικές εφαρμογές και μάλιστα σε τραπεζικές ήταν ήδη γνωστή πριν από τον Β' παγκόσμιο πόλεμο. Η αρχική έκδοση της Enigma και η βρεττανική Typex είχαν δημιουργηθεί για τέτοιες εφαρμογές, παρ' όλο που μετά έγινε μετατροπή τους για στρατιωτικούς σκοπούς.

Η ιδέα για κρυπτογράφηση μέσω μιας τυχαίας ακολουθίας ήταν γνωστή πριν τον πόλεμο. Οι κρυπτογράφοι ήδη δέχονταν, έστω εμπειρικά, ότι το one time pad ήταν απαραβίαστο, δηλαδή η κρυπτογράφηση με τυχαίο κλειδί αρκετά μακρύ όσο να καλύπτει το κείμενο. Στη συνέχεια ήρθε και η απόδειξη από τον Shannon [Sha49]. Στο άρθρο του '74 ο Feistel αναφέρεται στον κώδικα Vernam σημειώνοντας ότι το μήκος κλειδιού τον κάνει μη πρακτικό και ότι κώδικες όπως αυτός που προτείνει είναι η καλύτερη δυνατή προσέγγιση.

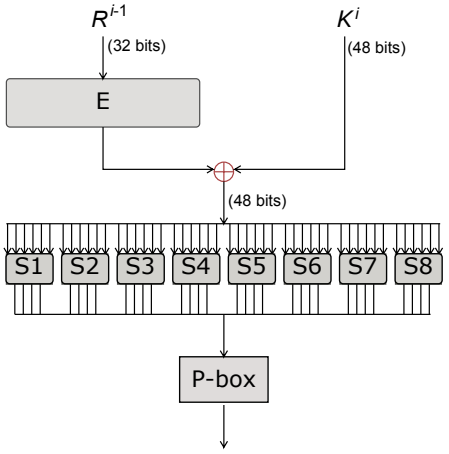
Η IBM υπέβαλε μια πιο εξελιγμένη εκδοχή του Lucifer για το Data Encryption Standard, η οποία και υιοθετήθηκε αφού έγιναν διάφορες μεταβολές, με κυριότερες τη μείωση του μήκους του κλειδιού και αλλαγές στα S-boxes. Κάποιες από αυτές τις αλλαγές, τις οποίες επέβαλε η NSA, θεωρήθηκαν ύποπτες από διακεκριμμένους κρυπτογράφους, όπως οι Diffie και Hellman, ενώ άλλοι, όπως ο Schneier, είπαν ότι δεν φαίνεται να υπάρχει συστημικό πρόβλημα.

Συνοπτική περιγραφή του DES

Το *Data Encryption Standard* (DES) περιέχεται στην Federal Information Processing Standards (FIPS) Publication 46 με ημερομηνία 15/1/1977 και είναι μια ειδική μορφή του κώδικα Feistel. Αποσύρθηκε το 2005.

Η κρυπτογράφηση γίνεται σε 16 γύρους όπως στην (2.3). Η συνάρτηση κρυπτογράφησης του ενός γύρου επεξεργάζεται το string z^i μήκους 64 bits. Το z^i υφίσταται διαφορετική επεξεργασία στα πρώτα 32 bits από αυτήν που γίνεται στα δεύτερα 32 bits, οπότε υποθέτουμε ότι το μισό αριστερό μέρος συμβολίζεται με L^i και το μισό δεξιό μέρος συμβολίζεται με R^i, δηλαδή $z^i = L^i \| R^i$ (όπου το σύμβολο $\|$ εδώ σημαίνει την συνένωση -concatenation- των δύο strings). Ο μετασχηματισμός που γίνεται στον γύρο i ($i = 1, 2, \ldots, 16$) έχει τότε ως εξής:

$$z^i \equiv L^i \| R^i = R^{i-1} \| L^{i-1} \oplus F_{K^i}(R^{i-1}) \tag{2.4}$$



Σχήμα 2.5: Η συνάρτηση F στο DES.

Κατά συνέπεια το δεξιό μέρος της εισόδου του γύρου έχει περάσει χωρίς μεταβολή στο αριστερό μέρος της εξόδου. Στον υπολογισμό του δεξιού μέρους εμπλέκεται η συνάρτηση F (Feistel), η οποία παίρνει ως είσοδο προφανώς 32 bits, αλλά η έξοδός της εξαρτάται και από το κλειδί K^i του γύρου με μήκος 48 bits.

Η συνάρτηση F αναλύεται περαιτέρω στο Σχ. 2.5. Στο block E τα 36 εισερχόμενα bits γίνονται 48 με κάποια μετάθεση και διπλασιασμό 16 bits. Στη συνέχεια γίνονται XOR με το κλειδί του γύρου και ανά 6 μπαίνουν στα block υποκατάστασης (S(ubstitution)-box). Κάθε S-box δέχεται ως είσοδο μια εξάδα και βγάζει μια τετράδα. Η συνένωση των 8 τετράδων μπαίνει στο τελικό P-box για μια ακόμη μετάθεση.

Και τώρα δυο λόγια για τον αρχικό γύρο. Το μη κρυπτογραφημένο κείμενο x υφίσταται μια αρχική μετάθεση IP (Initial Permutation) πριν οδηγηθεί στην είσοδο του πρώτου γύρου, δηλαδή αντί του πρώτου τύπου της 2.3 ισχύει η $z^0 = \text{IP}(x)$. Στη συνέχεια η $z^0 \equiv L^0 \| R^0$ οδηγείται στην (2.4) για $i = 1$. Ομοίως στο τέλος του 16ου γύρου, όταν πλέον έχει δημιουργηθεί το string z^{16}, εκτελείται η αντίστροφη της IP μετάθεση πάνω στην $z^{16} = L^{16} \| R^{16}$, αφού όμως πρώτα το δεξιό μέρος ανταλλαγεί με το αριστερό, δηλαδή τελικά $y = \text{IP}^{-1}(R^{16} \| L^{16})$.

Για την ακριβή λειτουργία των P/S-boxes, την μετάθεσης IP και την παραγωγή των κλειδιών από το αρχικό κλειδί μπορείτε να δείτε το βιβλίο του Alan Konheim [Kon07] (ο οποίος ήταν εκ των σχεδιαστών του DES) ή απ' ευθείας το σχετικό πρότυπο [FIP99].

Αναστρεψιμότητα Για να είναι ένας τέτοιος κώδικας αναστρέψιμος αρκεί να είναι κάθε γύρος αναστρέψιμος. Η αρχική και η τελική μετάθεση μέσω της IP είναι προφανώς αναστρέψιμη. Περαιτέρω για όσα συμβαίνουν σε κάθε γύρο δείτε ότι από την (2.4) είναι δυνατή η αναστροφή ανεξαρτήτως της F. Το δεξιό μέρος του z^{i-1} προκύπτει απ' ευθείας από το αριστερό μέρος του z^i, δηλαδή $R^{i-1} = L^i$. Το αριστερό μέρος του z^{i-1} προκύπτει ως εξής:

$$\begin{aligned} & R^i = L^{i-1} \oplus F_{K^i}(R^{i-1}) \\ \Rightarrow \quad & L^{i-1} = R^i \oplus F_{K^i}(R^{i-1}) = R^i \oplus F_{K^i}(L^i) \end{aligned}$$

Κλειδιά Είδαμε ότι στον αλγόριθμο υπεισέρχονται τα κλειδιά $K^1, K^2, \ldots, K^{16}$ και καθένα έχει μήκος 48 bits. Αυτό δεν σημαίνει ότι ο συνολικός αλγόριθμος χρειάζεται ένα κλειδί μήκους $16 \times 48 = 768$ bits. Το κλειδί K για τον DES έχει μήκος 56 bits. Εξ αυτών των 56 bits με επιλογή και μετάθεση δημιουργούνται τα 16 κλειδιά.

3DES

Το 1995 προκειμένου να παραταθεί η ζωή του DES και να αντιμετωπισθεί το μικρό μήκος κλειδιού προτάθηκε με την *RFC 1851* η εφαρμογή του τρεις φορές διαδοχικά, δηλαδή διοχετεύοντας την έξοδο της πρώτης εφαρμογής του DES στην είσοδο της δεύτερης και την έξοδο αυτής στην τρίτη. Το σύστημα αυτό ονομάζεται *τριπλό DES* (3DES) και ο αντίστοιχος αλγόριθμος *Triple Data Encryption Algorithm* (TDEA). Κατά τις τρεις φάσεις χρησιμοποιούνται τρία διαφορετικά κλειδιά εφόσον δοθεί ένα κλειδί τριπλού μήκους, δηλαδή αποτελούμενο από 168 bits. Μια δεύτερη δυνατότητα είναι η χρήση κλειδιού διπλού μήκους, 112 bits, όπου το πρώτο μισό χρησιμοποιείται στην πρώτη και στην τρίτη φάση. Τέλος για λόγους συμβατότητας με το παρελθόν μπορεί να χρησιμοποιηθεί και απλό κλειδί μήκους 56 bits, ίδιο και στις τρεις φάσεις. Το μήκος του block του κειμένου προφανώς παραμένει στα 64 bits, όπως στον DES. Η περιγραφή του 3DES ανανεώθηκε το 2017 μέσω της *NIST SP 800-67 Rev2* [Bar17].

2.7 Advanced Encryption Standard

Το Advanced Encryption Standard (AES) βασίζεται στον τμηματικό αλγόριθμο Rijndael των Βέλγων κρυπτογράφων Vincent Rijmen and Joan Daemen.[9] Ανακοινώθηκε στις ΗΠΑ από το NIST (National Institute of Standards and Technology) ως FIPS PUB 197 στις 26 Νοεμβρίου 2001. Έχει επίσης περιληφθεί στο διεθνές πρότυπο ISO/IEC 18033-3.

Το NIST δήλωσε στις αρχές του 1997 την πρόθεσή του να τυποποιήσει ένα νέο συμμετρικό αλγόριθμο κρυπτογράφησης ευαίσθητων κρατικών πληροφοριών. Ο αλγόριθμος θα έπρεπε να είναι διαθέσιμος διεθνώς και ατελώς. Ανάμεσα στις προδιαγραφές ήταν να δέχεται τμήματα μήκους 128 bits και κλειδιά με 128, 192 και 256 bits. Τον Αύγουστο του 1998 ανακοίνωσε ότι είχε κάνει μια προεπιλογή 15 υποψηφίων αλγορίθμων (που προέρχονταν από 15 χώρες) και τους έθεσε σε δημόσια διαβούλευση, ώστε να τεθούν σε δοκιμές από το ευρύτερο δυνατό κοινό, περιλαμβανομένης και της ακαδημαϊκής κοινότητας. Βάσει των αποτελεσμάτων που ήταν γνωστά ως τον Μάρτιο του 1999, ανακοίνωσε τον Αύγουστο την επιλογή των πέντε επικρατεστέρων. Όλοι ήταν αλγόριθμοι που βασίζονταν σε μετασχηματισμούς που επαναλαμβάνονταν σε έναν αριθμό γύρων και σε κάποια μέθοδο παραγωγής κλειδιών των γύρων από το αρχικό κλειδί. Οι 4 εξ αυτών χρησιμοποιούσαν S-boxes και οι 3 παραλλαγές της συνάρτησης Feistel. Ανάμεσα στους υποψήφιους αλγόριθμους ήταν ο MARS της IBM, που χρησιμοποιούσε μεταθέσεις, υποκαταστάσεις και παραλλαγές της Feistel, ο RC6 των RSA Laboratories, ο Serpent με 32 γύρους όπου γίνονται ανάλογοι μετασχηματισμοί, ο Twofish, βασισμένος σε ένα δίκτυο Feistel 16 γύρων, και φυσικά ο Reijndael,

[9] Ο Vincent Rijmen γεννήθηκε το 1970 στο Leuven του Βελγίου. Σπούδασε ηλεκτρονική στο Καθολικό Πανεπιστήμιο του Leuven (Katholieke Universiteit Leuven), εντάχθηκε στο ερευνητικό εργαστήριο *Computer Security and Industrial Cryptography* (COSIC) και το 1977 τελείωσε τη διδακτορική του διατριβή στην περιοχή της κρυπτογραφίας. Σήμερα είναι καθηγητής στο πανεπιστήμιο του Leuven και στο πανεπιστήμιο του Bergen στη Νορβηγία.

Ο Joan Daemen γεννήθηκε το 1965 στο Achel του Βελγίου. Σπούδασε ηλεκτρολόγος μηχανικός επίσης στο Καθολικό Πανεπιστήμιο του Leuven, όπου επίσης εντάχθηκε στο εργαστήριο COSIC και τελείωσε τη διδακτορική διατριβή του το 1995. Σήμερα είναι καθηγητής στο Radboud University Nijmegen.

τον οποίο θα περιγράψουμε στη συνέχεια. Δύο εξ αυτών, ο Reijndael και ο Twofish κάνουν χρήση των πεδίων Galois, όλοι βασίζονται σε XOR, όλοι κάνουν μεταθέσεις και περιστροφές, όλοι εκτός του RC6 χρησιμοποιούν έτοιμους πίνακες.

Για την επιλογή εκτός από το βασικό κριτήριο της ασφάλειας χρησιμοποιήθηκαν και κριτήρια κόστους, ταχύτητας, απαιτήσεων μνήμης, επεκτασιμότητας στο μέλλον και άλλων χαρακτηριστικών της υλοποίησης. Λεπτομέρειες για την επιλογή μπορούν να αναζητηθούν στην έκθεση [Nec+01] που συνοδεύει την τελική επιλογή και κατέληξε στον Rijndael. Ανάμεσα στα πλεονεκτήματά του ήταν διπλή ταχύτητα από τον επόμενο υποψήφιο και χαμηλή χρήση μνήμης.

Το Advanced Encryption Standard δέχεται block μήκους 128 bits και μπορεί να χρησιμοποιήσει κλειδιά των 128, 192 και 256 bits. Είναι και πάλι ένας κώδικας με πολλούς γύρους των οποίων ο αριθμός εξαρτάται από το μήκος κλειδιού, δηλαδή με $N = 10, 12$ και 14 γύρους αντίστοιχα με τα παραπάνω μήκη.

Κλειδιά γύρων

Κάθε μια από τις τρεις παραλλαγές του DEA χρησιμοποιεί $N + 1$ κλειδιά γύρων που παράγονται από το αρχικό κλειδί. Θα περιγράψουμε την παραγωγή κλειδιών για $N = 10$ από το κλειδί των 128 bits. Πρέπει να παραχθούν 11 κλειδιά γύρων, καθένα από τα οποία έχει μήκος 16 bytes (128 bits) ή 4 λέξεων (με 4 bytes εκάστη). Αν βάλουμε όλα τα κλειδιά σε ένα string $K^0 \| K^1 \| \dots K^{10}$ αποτελούνται από την συνένωση 4×11 λέξεων, δηλαδή είναι ένα string της μορφής $w_0 \| w_1 \| \dots w_{43}$, όπου w_{i-1} είναι η i-οστή λέξη ($i = 1, \dots, 44$). Έστω ότι το αρχικό κλειδί γραμμένο σε οκτάδες είναι $K = k_0 \| k_1 \| \dots \| k_{15}$. Υπολογίζουμε τις πρώτες 4 λέξεις w_0, w_1, w_2, w_3 με τον τύπο

$$w_i = k_{4i} \| k_{4i+1} \| k_{4i+2} \| k_{4i+3}, \quad i = 0, 1, 2, 3$$

οπότε ως τώρα έχουμε ήδη χρησιμοποιήσει όλο το αρχικό κλειδί και έχουμε μόλις δημιουργήσει το $K^0 = w_0 \| w_1 \| w_2 \| w_3$. Για τις επόμενες λέξεις χρειάζονται 10 σταθερές μήκους 32 bits εκάστη και είναι οι εξής γραμμένες ανά 4-άδα στο 16-δικό σύστημα:

$$\begin{aligned} r_1 &= 01000000 \\ r_2 &= 02000000 \\ r_3 &= 04000000 \\ r_4 &= 08000000 \\ r_5 &= 10000000 \\ r_6 &= 20000000 \\ r_7 &= 40000000 \\ r_8 &= 80000000 \\ r_9 &= 1B000000 \\ r_{10} &= 36000000 \end{aligned}$$

Στις σταθερές αυτές εμφανώς μεταβάλλεται μόνο το πρώτο byte.[10]. Χρειάζονται επίσης ένα P-box κι ένα S-Box που εκτελούν αντίστοιχα τους μετασχηματισμούς R και

[10] Το πρώτο byte αντιστοιχεί στους συντελεστές ενός πολυωνύμου του πεπερασμένου πεδίου (Galois field) $\mathrm{GF}(2)[x]/(x^8 + x^4 + x^3 + x + 1)$, βλ. παρακάτω.

S πάνω σε λέξεις, δηλαδή σε τετράδες από bytes. Ο μετασχηματισμός R από την τετράδα των bytes $b_0\|b_1\|b_2\|b_3$ κάνει μια κυκλική ολίσθηση αριστερά (rotation) κατά ένα byte, δηλαδή δίνει το string $b_1\|b_2\|b_3\|b_0$. Ο μετασχηματισμός S κάνει μια υποκατάσταση όπως το S-box του βασικού αλγόριθμου που θα δούμε παρακάτω. Με βάση αυτά οι υπόλοιπες λέξεις σχηματίζονται διαδοχικά ως εξής:

$$\begin{aligned} w_4 &= S(R(w_3)) \oplus w_0 \oplus r_1 \\ w_5 &= S(R(w_4)) \oplus w_1 \\ w_6 &= S(R(w_5)) \oplus w_2 \\ w_7 &= S(R(w_6)) \oplus w_3 \\ w_8 &= S(R(w_7)) \oplus w_4 \oplus r_2 \\ &\dots \end{aligned}$$

δηλαδή οι σταθερές προστίθενται μόνο στις λέξεις με δείκτη που είναι πολλαπλάσιος του 4.

Βασικός αλγόριθμος

Ο αλγόριθμος μπορεί να περιγραφεί σε ψηλό επίπεδο ως εξής:

1. Το αρχικό string εισόδου (plaintext) x (μήκους 128 bits) γίνεται XOR με το K^0, δηλαδή $z^0 = x \oplus K^0$.

2. Για τους επόμενους $N-1$ γύρους

   $$z_i = M(P(S(z_{i-1}))) \oplus K^i, \quad i = 1, 2, 3, \dots, N-1$$

   όπου S είναι υποκατάσταση, P είναι μετάθεση και M είναι μίξη στηλών. Οι λειτουργίες αυτές αναλύονται στις επόμενες ενότητες.

3. Στον τελευταίο γύρο

   $$z^N = P(S(z^{N-1})) \oplus K^N$$

   και το κρυπτογραφημένο κείμενο είναι $y = z^N$.

Στη συνέχεια περιγράφουμε καθένα από τους παραπάνω μετασχηματισμούς M, P και S.

Υποκατάσταση S (S-Box)

Οι πράξεις στο AES είναι byte oriented, δηλαδή εκτελούνται εν γένει πάνω σε bytes. Δεδομένου ότι το block είναι μήκους 128 bits, μπορεί να γραφεί με μορφή 16 bytes ως $x = x_0\|x_1\| \dots \|x_{15}$, όπου κάθε x_i παριστάνει ένα byte. Στο S-box του AES ένα byte υποκαθίσταται σταθερά από ένα άλλο byte σύμφωνα με τον Πίνακα 2.3 διπλής εισόδου, όπου η πρώτη στήλη δείχνει τον δεκαεξαδικό αριθμό που αποτελεί το πρώτο μισό byte και η πρώτη γραμμή το δεκαεξαδικό για το δεύτερο μισό byte (του εισερχόμενου byte). Για παράδειγμα αν το εισερχόμενο byte είναι E9, το εξερχόμενο από το S-box είναι 1E. Ωστόσο ο πίνακας υποκατάστασης δεν έχει σχηματισθεί αυθαίρετα, αλλά προέρχεται από τη χρήση της θεωρίας των πεπερασμένων πεδίων (Galois fields).

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Πίνακας 2.3: Ο πίνακας υποκατάστασης για το S-box του AES.

Υπολογισμός του S-box

Έστω ότι το εισερχόμενο byte αναλυόμενο στα επί μέρους bits είναι $a_7a_6a_5a_4a_3a_2a_1a_0$. Σχηματίζουμε το πολυώνυμο

$$\phi(x) = a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$$

όπου οι συντελεστές προφανώς είναι μόνο 0 ή 1. Στο πεδίο $\mathrm{GF}(2)[x]/(x^8 + x^4 + x^3 + x + 1)$ υπολογίζουμε το αντίστροφο πολυώνυμο (δείτε πιο κάτω με ποιο τρόπο γίνεται ο υπολογισμός) και έστω ότι είναι $\psi(x) = \sum_{i=0}^{7} b_i x^i$. Έστω επίσης το string $c_7c_6c_5c_4c_3c_2c_1c_0 = 01100011$. Στη συνέχεια θέτουμε

$$d_i = b_i + b_{i+4} + b_{i+5} + b_{i+6} + b_{i+7} + c_i \mod 2 \tag{2.5}$$

και θέτουμε την έξοδο του S-box ίση με $a_7a_6a_5a_4a_3a_2a_1a_0$

Τότε το εξερχόμενο byte είναι αντιστοίχως $d_7d_6d_5d_4d_3d_2d_1d_0$.

Μικρή εισαγωγή στα πεδία Galois

Τα *πεδία Galois* ή *πεπερασμένα πεδία* είναι μια επέκταση της έννοιας της ομάδας $\mathbb{Z}_m$ στα πολυώνυμα μιας μεταβλητής με ακέραιους συντελεστές και εκθέτες. Στην περίπτωση των ακεραίων το $\mathbb{Z}_m$ υπολογίζοντας το αποτέλεσμα των πράξεων $\mod m$ κλείνονται οι πράξεις μέσα στο σύνολο $\{0, 1, \ldots, m-1\}$, δηλαδή το αποτέλεσμα είναι πάντοτε μέσα σ' αυτό το σύνολο. Ένα πολυώνυμο χαρακτηρίζεται από τον εκθέτη του μεγιστοβάθμιου όρου και από τις τιμές των συντελεστών. Έστω ότι θεωρούμε πολυώνυμα το πολύ n-οστής δύναμης με συντελεστές από το $\mathbb{Z}_m$. Η πρόσθεση τέτοιων πολυωνύμων (και κατά συνέπεια και ο πολλαπλασιασμός) μπορεί να οδηγήσει εύκολα σε δυνάμεις μεγαλύτερες του n και σε συντελεστές εκτός του $\mathbb{Z}_m$. Η θεωρία των πεδίων Galois βασίζεται στο ότι οι συντελεστές υπολογίζονται πάντοτε $\mod m$ (για κάποιο ορισμένο θετικό ακέραιο m) και ο υποβιβασμός του βαθμού του πολυωνύμου γίνεται μέσω της διαίρεσης με ένα πολυώνυμο $\varphi(x)$ βαθμού $m+1$ και της λήψης του υπολοίπου. Σύμφωνα με τη θεωρία των πεδίων Galois κατάλληλο για την παραπάνω διαδικασία είναι ένα πολυώνυμο $\varphi(x)$ που είναι «ανάγωγο» (irreducible) μέσα στο πεδίο, δηλαδή δεν μπορεί να σπάσει σε γινόμενο δύο άλλων πολυωνύμων.

Στην περίπτωση του S-box του AES το πεδίο που χρησιμοποιείται είναι το $\mathrm{GF}(2)[x]/(x^8+x^4+x^3+x+1)$, δηλαδή ένα πεδίο με μέγιστο βαθμό $n=7$ και $m=2$, όπου το ανάγωγο πολυώνυμο είναι το $\varphi(x)=x^8+x^4+x^3+x+1$. Επομένως οι πράξεις πάνω στους συντελεστές γίνονται όλες $\bmod 2$ και όταν γίνεται πρόσθεση ή πολλαπλασιασμός των πολυωνύμων πρέπει στο τέλος να λαμβάνεται το υπόλοιπο της διαίρεσης με το $\varphi(x)$. Πρακτικά αυτό σημαίνει ότι κατά την εκτέλεση των πράξεων όροι που είναι πολλαπλάσια του $\varphi(x)$ διαγράφονται και συντελεστές διορθώνονται $\bmod 2$.

Έστω τώρα ότι δύο πολυώνυμα του πεδίου $\mathrm{GF}(m)[x]/\varphi(x)$ (όπου το m όπως πιο πάνω υποδηλώνει τις τιμές που παίρνουν οι συντελεστές, ενώ το $\varphi(x)$ είναι ένα ανάγωγο πολυώνυμο) δύο πολυώνυμα έχουν την ιδιότητα $\psi_1(x)\times\psi_2(x)=1$. Τότε το ένα λέγεται αντίστροφο του άλλου. Προφανώς αν οι πράξεις δεν γίνονταν μέσα στο πεδίο το αποτέλεσμα θα ήταν ένα πολυώνυμο βαθμού ίσου με το άθροισμα των βαθμών των πολυωνύμων, αλλά μετά την απλοποίηση των συντελεστών στο $\mathbb{Z}_m$ καταλήγει να είναι της μορφής $\psi_3(x)\varphi(x)+1$, οπότε μετά τη διαίρεση με το $\varphi(x)$ γίνεται ίσο με 1 μέσα στο πεδίο.

Η αντιστροφή μπορεί να γίνει προσαρμόζοντας κατάλληλα τη μακρά διαίρεση πολυωνύμων, αλλά πρακτικά προσφέρουν πράξεις σε πεδία πολλά υπολογιστικά πακέτα, όπως Maxima, Mathematica κ.α., περιλαμβανομένης και της αντιστροφής.

Παράδειγμα 7. Θα κάνουμε τη μετατροπή για το S-box του string εισόδου DD. Το δεκαεξαδικό D είναι ο δεκαδικός αριθμός 13 ή δυαδικός 1101, άρα σχηματίζουμε το πολυώνυμο με συντελεστές 11011101, δηλαδή το $x^7+x^6+x^4+x^3+x^2+1$.

Θα χρησιμοποιήσουμε το πακέτο Maxima για την αντιστροφή του πολυωνύμου.[11] Η αντιστροφή δίνει $x^7+x^6+x^5+x^4+x^3$, άρα με τον παραπάνω συμβολισμό $b_7b_6b_5b_4b_3b_2b_1b_0=11111000$. Εφαρμόζοντας την 2.5 το αποτέλεσμα είναι 11000001, στο δεκαεξαδικό C1.

Στη συνέχεια μπορείτε για άσκηση να πολλαπλασιάσετε τα δύο πολυώνυμα και κάνοντας τις απλοποιήσεις στους συντελεστές των πολυωνύμων και στα πολλαπλάσια του $\varphi(x)$ να καταλήξετε στη μονάδα. □

Μετάθεση P

Όπως είπαμε πιο πριν, το block είναι μήκους 128 bits και μπορεί να γραφεί με μορφή 16 bytes ως $x=x_0\|x_1\|\ldots\|x_{15}$, όπου κάθε x_i παριστάνει ένα byte. Η μετάθεση M μπορεί να περιγραφεί καλύτερα αν από το αρχικό block κάθε byte εισαχθεί σε ένα πίνακα X διαστάσεων 4×4 σχηματίζοντας πρώτα τις στήλες. Τότε ο μετασχηματισμός μετάθεσης P αναδιατάσσει τα στοιχεία του X και δημιουργεί τον X' ως εξής:

$$X=\begin{bmatrix} x_0 & x_4 & x_8 & x_{12}\\ x_1 & x_5 & x_9 & x_{13}\\ x_2 & x_6 & x_{10} & x_{14}\\ x_3 & x_7 & x_{11} & x_{15}\end{bmatrix}\xrightarrow{P} X'=\begin{bmatrix} x_0 & x_4 & x_8 & x_{12}\\ x_5 & x_9 & x_{13} & x_1\\ x_{10} & x_{14} & x_2 & x_6\\ x_{15} & x_3 & x_7 & x_{11}\end{bmatrix}$$

Μίξη στήλης M - mix columns

Έστω $[s_0,s_1,s_2,s_3]^T$ (όπου το T σημαίνει transpose, δηλαδή κάνει τη γραμμή στήλη και τη στήλη γραμμή) μια οποιαδήποτε από τις 4 στήλες του X. Ο μετασχηματισμός που θα περιγράψουμε εφαρμόζεται και στις 4 στήλες, αλλά σε κάθε μια χωριστά. Έστω ότι η συνάρτηση g μετατρέπει σε πολυώνυμο του x το πολύ εβδόμου βαθμού ένα

[11] Οι πράξεις στο συγκεκριμένο πεπερασμένο πεδίο ενεργοποιείται μέσω της εντολής `gf_set_data(2,`$x^8+x^4+x^3+x+1$`)` και η αντιστροφή γίνεται μέσω της συνάρτησης `gf_inv`.

byte, όπως είδαμε σε προηγούμενη ενότητα (δηλ. το πρώτο από αριστερά bit είναι ο συντελεστής του όρου x^7), και η συνάρτηση g^{-1} μετατρέπει αντίστροφα το πολυώνυμο σε ένα byte. Έστω επίσης ότι ο πολλαπλασιασμός πολυωνύμων στο γνωστό μας πεδίο Galois παριστάνεται με $\odot$, ενώ το σύμβολο $\oplus$ παριστάνει την πρόσθεση πολυωνύμων στο πεδίο, πράγμα που σημαίνει XOR των συντελεστών της αντίστοιχης συνιστώσας. Τότε η στήλη αυτή μετασχηματίζεται σε μια στήλη $[s'_0, s'_1, s'_2, s'_3]^T$ ως εξής:

$$\begin{aligned}
s'_0 &= g^{-1}[[g(s_0) \odot x] \oplus [g(s_1) \odot (x+1)] \oplus g(s_2) \oplus g(s_3)] \\
s'_1 &= g^{-1}[[g(s_1) \odot x] \oplus [g(s_2) \odot (x+1)] \oplus g(s_3) \oplus g(s_0)] \\
s'_2 &= g^{-1}[[g(s_2) \odot x] \oplus [g(s_3) \odot (x+1)] \oplus g(s_0) \oplus g(s_1)] \\
s'_3 &= g^{-1}[[g(s_3) \odot x] \oplus [g(s_0) \odot (x+1)] \oplus g(s_1) \oplus g(s_2)]
\end{aligned}$$

Παράδειγμα 8. Ας υποτεθεί ότι μια από τις στήλες είναι

$$[s_0, s_1, s_2, s_3]^T = \begin{bmatrix} 10001011 \\ 00100111 \\ 01000100 \\ 00101010 \end{bmatrix}$$

Επομένως τα αντίστοιχα πολυώνυμα είναι

$$\begin{aligned}
g(s_0) &= x^7 + x^3 + x + 1 \\
g(s_1) &= x^5 + x^2 + x + 1 \\
g(s_2) &= x^6 + x^2 \\
g(s_3) &= x^5 + x^3 + x
\end{aligned}$$

Θα υπολογίσουμε μόνο το s'_0.

Πολλαπλασιάζουμε το $g(s_0)$ με το x, αρχικά

$$g(s_0)x = x^8 + x^4 + x^2 + x$$

Για να απαλλαγούμε από τον όρο x^8 προσθέτουμε το ανάγωγο πολυώνυμο $\varphi(x) = x^8 + x^4 + x^3 + x + 1$ του πεδίου (προσθέτοντας τους αντίστοιχους συντελεστές $\mod 2$, δηλαδή ουσιαστικά αφαιρούνται οι ίδιοι όροι), οπότε προκύπτει $g(s_0) \odot x = x^3 + x^2 + 1$.

Στη συνέχεια πρέπει να γίνει ο υπολογισμός $g(s_0) \odot (x+1)$. Πολλαπλασιάζουμε το $g(s_0)$ με το $x+1$, αρχικά

$$g(s_1)(x+1) = (x^5 + x^2 + x + 1)(x+1) = x^6 + x^3 + x^2 + x + x^5 + x^2 + x + 1$$

αλλά οι όροι $2x^2$ και $2x$ μηδενίζονται εφόσον οι συντελεστές τους $\mod 2$ είναι μηδέν. Αυτό που μένει δεν διαιρείται προφανώς με το $\varphi(x)$ εφόσον είναι κατώτερου βαθμού, οπότε

$$g(s_1) \odot (x+1) = x^6 + x^5 + x^3 + 1$$

Τα δύο παραπάνω πολυώνυμα $g(s_0) \odot x$ και $g(s_1) \odot (x+1)$ και τα $g(s_2) = x^6 + x^2$ και $g(s_3) = x^5 + x^3 + x$ τα προσθέτω όλα μαζί, δηλαδή προσθέτω $\mod 2$ τους αντίστοιχους συντελεστές των όρων ίσου βαθμού των τεσσάρων πολυωνύμων και προκύπτει $x^3 + x$. Επομένως το byte $s_0 = 10001011$ μετασχηματίζεται στο $s'_0 = 00001010$. □

ΚΕΦΑΛΑΙΟ 3

Μη συμμετρική κρυπτογραφία

Η *μη συμμετρική κρυπτογραφία* και η *κρυπτογραφία δημόσιου κλειδιού* χρησιμοποιούνται ως όροι συνώνυμοι. Η μη συμμετρική κρυπτογραφία υποδηλώνει ένα σύστημα όπου το κλειδί κρυπτογράφησης δεν είναι ίδιο με το κλειδί αποκρυπτογράφησης, δηλαδή η κρυπτογράφηση γίνεται ως $e_{K_1}(x)$ και η αποκρυπτογράφηση ως $d_{K_2}(x)$ και $d_{K_2}(e_{K_1}(x)) = x$, αλλά εν γένει με $K_1 \neq K_2$. Εκ πρώτης όψεως αυτή είναι μια μικρή διαφορά, η οποία όμως ανοίγει πληθώρα νέων δυνατοτήτων και καταλήγει να αξιοποιείται σε πολυάριθμα συστήματα και εφαρμογές.

Παρ' όλο που κανείς δεν μπορεί να απαγορεύσει στον Bob και την Alice να δημιουργήσουν δύο τέτοια κλειδιά K_1, K_2 και να τα κρατήσουν αμφότερα μυστικά, αυτό πρακτικά θα ήταν ανορθόδοξο για μια σειρά από λόγους, ανάμεσα στους οποίους είναι και ότι οι μη συμμετρικές μέθοδοι είναι εν γένει πιο περίπλοκες από τις συμμετρικές. Η συνήθης πρακτική είναι το ένα από τα δύο κλειδιά να δημοσιοποιείται, ενώ το άλλο παραμένει μυστικό.

Δυο χαρακτηριστικά παραδείγματα τέτοιας χρήσης είναι τα ακόλουθα:

1. Αρχικά ο Bob δημιουργεί ένα ζεύγος κλειδιών (K_1, K_2) και δημοσιοποιεί το K_1 ενώ κρατάει κρυφό το K_2. Η Alice θέλει να στείλει ένα μυστικό μήνυμα x στον Bob. Παίρνει το δημόσιο κλειδί του Bob, δηλαδή το K_1, και του στέλνει το κρυπτογραφημένο $y = e_{K_1}(x)$. Ο Bob και μόνο αυτός μπορεί να το αποκρυπτογραφήσει ως $d_{K_2}(y)$. Με τον ίδιο τρόπο οποιοσδήποτε μπορεί να στείλει ένα μυστικό μήνυμα στον Bob και μόνο ο Bob θα μπορέσει να το διαβάσει.

2. Ο Bob δημιουργεί ένα ζεύγος κλειδιών (K_1, K_2) και δημοσιοποιεί το K_2 ενώ κρατάει κρυφό το K_1. Ο Bob θέλει να στείλει ένα φανερό μήνυμα x στην Alice, αλλά να υπάρχει εγγύηση ότι το μήνυμα προέρχεται από αυτόν και όχι από κάποιον άλλον. Παίρνει το ιδιωτικό του κλειδί, δηλαδή το K_1, και στέλνει στην Alice το κρυπτογραφημένο $y = e_{K_1}(x)$. Η Alice παίρνει το δημόσιο κλειδί του Bob, υπολογίζει το $x = d_{K_2}(y)$ και έτσι είναι σίγουρη ότι το μήνυμα προέρχεται από τον Bob, που είναι ο μόνος που θα μπορούσε να το κρυπτογραφήσει. Βεβαίως το μήνυμα μπορεί να το διαβάσει οποιοσδήποτε.

Όταν ο Whitfield Diffie ρωτήθηκε μετά την απονομή του βραβείου Turing 2015 σε συνέντευξη[1] για τη σημασία της μη συμμετρικής κρυπτογραφίας αναφέρθηκε στο θέμα της διαχείρισης κλειδιών δίνοντας ένα παράδειγμα που δείχνει το μέγεθος του προβλήματος διαχείρισης κλειδιών όταν αυτή πρέπει να γίνει μαζικά. Είπε ότι η διαχείριση κλειδιών στις αμερικανικές στρατιωτικές και μυστικές υπηρεσίες γίνεται κεντρικά από την NSA, η οποία τα κατασκευάζει και τα μοιράζει σε όλους τους βαθμούς της ιεραρχίας. Για μια εποχή όπου η διανομή γινόταν κάρτες και ταινίες με βάση το χαρτί έπρεπε να τα εκτυπώνει σε τεράστια μονάδα εκτύπωσης. Στη συνέχεια αναφέρθηκε στην εποχή του Internet, όπου κάτι αντίστοιχο δεν θα μπορούσε να δημιουργηθεί για τους χρήστες του, αν μη τι άλλο γιατί δεν ανήκουν σε κάποιου είδους ενιαίο οργανισμό που θα αναλάμβανε τη διανομή κλειδιών. Αντίθετα μπορεί να είναι παντελώς άγνωστοι μεταξύ τους και να πρέπει να συνεννοηθούν χωρίς να έχουν συναντηθεί ποτέ προηγουμένως.

3.1 Ο αλγόριθμος RSA

Ένας από τους πρώτους αλγόριθμους δημόσιου κλειδιού είναι ο αλγόριθμος των Ron Rivest, Adi Shamir και Len Adleman, γνωστός ως RSA [RSA78]. Ο RSA επεξεργάζεται ένα μήνυμα κατά τμήματα. Τόσο το αρχικό κείμενο M, όσο και το κρυπτογραφημένο C είναι ακέραιοι μεταξύ 0 και $n-1$ (ένα κείμενο που έχει μετατραπεί σε bit string μήκους m είναι αριθμός μεταξύ 0 και 2^m-1). Η κρυπτογράφηση γίνεται μέσω του τύπου

$$C = M^e \bmod n$$

Η αποκρυπτογράφηση γίνεται ως

$$M = C^d \bmod n$$

Η γενική παράμετρος n είναι δημόσια γνωστή, καθώς και το ένα εκ των d, e, όποιο εξ αυτών είναι δημόσιο κλειδί.

Πριν εξηγήσουμε πώς σχετίζονται μεταξύ τους οι διάφορες παράμετροι και πώς υπολογίζονται οι τιμές τους είναι σκόπιμες ορισμένες διευκρινίσεις σχετικές με την ακρίβεια των πράξεων. Η χρήση ενός κλειδιού μήκους 1024 bits που εφαρμόζεται σε τμήματα των 86 bytes συνεπάγεται υπολογισμούς με πολύ μεγάλους ακέραιους. Η μέγιστη τιμή του κλειδιού είναι $2^{1024}-1 \simeq 1.798 \times 10^{308}$ και η μέγιστη τιμή του κειμένου είναι $2^{86\times 8} \simeq 1.284 \times 10^{207}$, κατά συνέπεια η πλήρης γραφή των παραπάνω ως ακεραίων χρειάζεται (στο δεκαδικό σύστημα) 309 και 208 ψηφία αντίστοιχα. Η συνήθης ακρίβεια ενός calculator στην αναγραφή και επεξεργασία αριθμών είναι μερικές δεκάδες σημαντικών ψηφίων, δηλαδή ένας αριθμός σε επιστημονική μορφή όπως ο 1.798×10^{308} περιλαμβάνει μετά την υποδιαστολή το πολύ λίγες δεκάδες δεκαδικών ψηφίων. Κατά συνέπεια ένας αριθμός όπως ο 2^{1024} στρογγυλοποιείται (προς τα κάτω) στον ακέραιο που έχει μηδενικά μετά από τις πρώτες λίγες δεκάδες σημαντικών ψηφίων. Όταν στη συνέχεια αυτός ο αριθμός διαιρείται με κάποιον ακέραιο και λαμβάνεται το υπόλοιπο, το αποτέλεσμα σε ένα συνηθισμένο calculator αποκλείεται να είναι σωστό. Πράξεις επί ακεραίων με πολύ μεγάλο μήκος πρέπει εν γένει να γίνονται με ειδικούς αλγόριθμους (μακρύ πολλαπλασιασμό, μακρά διαίρεση κ.λπ.), παρόμοιους με αυτούς που χρησιμοποιούμε όταν κάνουμε πράξεις στο χαρτί. Ωστόσο στην περίπτωση πράξεων της μορφής $a^b \bmod c$ είναι δυνατή η σημαντική απλοποίηση

[1] Βλ. και `https://amturing.acm.org/`.

του υπολογισμού με εφαρμογή ιδιοτήτων από τη θεωρία αριθμών, όπως θα δούμε πιο κάτω.

Για να παράγεται ο ίδιος αριθμός μετά από κρυπτογράφηση και αποκρυπτογράφηση πρέπει να ισχύει η σχέση $M^{ed} \bmod n = M$, δηλαδή χρειάζεται ένας μη πρώτος αριθμός f (για να μπορεί να σπάσει σε γινόμενο δύο τουλάχιστον άλλων ακεραίων που θα είναι τα κλειδιά) με την ιδιότητα $M^f \bmod n = M$. Η λύση που πρότειναν οι Rivest, Shamir και Adleman είναι να αξιοποιηθεί το θεώρημα του Euler που παρατίθεται πιο κάτω.

Επιλογή παραμέτρων και κλειδιών

Η επιλογή των κατάλληλων ακεραίων n, e, d σύμφωνα με τη μέθοδο των Rivest, Shamir και Adleman [RSA78] γίνεται με τα εξής βήματα:

- Επιλέγεται ένας φυσικός αριθμός $n = pq$ όπου αμφότεροι οι p και q είναι πρώτοι αριθμοί. Ιδανικά επιλέγονται δυο πολύ μεγάλοι «τυχαίοι» πρώτοι αριθμοί.
- Επιλέγεται ένας ακέραιος d μικρότερος του n, αλλά πρώτος προς το γινόμενο $(p-1)(q-1)$, δηλαδή μκδ$(d, (p-1)(q-1)) = 1$.[2] Και ο d επιλέγεται να είναι πολύ μεγάλος «τυχαίος» αριθμός.
- Ο αριθμός e επιλέγεται να είναι πολλαπλασιαστικός αντίστροφος του d modulo $(p-1)(q-1)$, δηλαδή

$$ed \bmod (p-1)(q-1) = 1$$

Η παραπάνω επιλογή παραμέτρων βασίζεται στο εξής θεώρημα:

Θεώρημα 2. (*E*uler) Αν a ακέραιος και n θετικός ακέραιος και οι a, n είναι πρώτοι μεταξύ τους, ισχύει

$$a^{\phi(n)} \bmod n = 1, \tag{3.1}$$

όπου $\phi(n)$ είναι η *συνάρτηση Euler* (Euler's totient function). □

Η συνάρτηση Euler δίνει εξ ορισμού το πλήθος των φυσικών αριθμών, οι οποίοι είναι μικρότεροι ή ίσοι με το n και οι οποίοι είναι πρώτοι προς το n. Αν m είναι πρώτος αριθμός, ισχύει ότι $\phi(m) = m - 1$. Αν m, r είναι αμφότεροι πρώτοι αριθμοί, ισχύει $\phi(mr) = \phi(m)\phi(r)$ και κατά συνέπεια $\phi(mr) = (m-1)(r-1)$.

Δεδομένης της (3.1) εάν ένα μήνυμα M και το n είναι αριθμοί πρώτοι προς αλλήλους ισχύει

$$M^{\phi(n)} \bmod n = 1$$

Εφόσον οι p, q είναι πρώτοι αριθμοί, ισχύει

$$\phi(pq) = (p-1)(q-1)$$

Επειδή ο αριθμός d είναι πρώτος προς το $\phi(n)$, υπάρχει ο πολλαπλασιαστικός αντίστροφός του με την ιδιότητα

$$ed \bmod \phi(n) = 1$$

οπότε $ed = k\phi(n) + 1$ για κάποιο φυσικό αριθμό k.

[2] μκδ συμβολίζει τον μέγιστο κοινό διαιρέτη.

Το M αποκλείεται να διαιρεί το p, αφού το p είναι πρώτος αριθμός, εκτός αν $M = p$. Αφού το p είναι πρώτος, οι διαιρέτες του p είναι μόνο το 1 και το p, οπότε ο μέγιστος κοινός διαιρέτης του με το M είναι το p αν το p είναι διαιρέτης του M, ενώ αν το p δεν διαιρεί το M είναι μόνο το 1, οπότε μκδ$(p, M) = 1$, δηλ. οι p, M είναι πρώτοι προς αλλήλους και ικανοποιούν τις προϋποθέσεις για το θεώρημα Euler. Επομένως αν το p δεν διαιρεί το M ισχύει

$$M^{\phi(p)} \bmod p = M^{p-1} \bmod p = 1$$

Επειδή το $(p-1)(q-1) = \phi(n)$ ισχύει

$$M^{k\phi(n)} \bmod p = M^{k(q-1)(p-1)} \bmod p = (M^{p-1} \bmod p)^{k(q-1)} \bmod p = 1$$

(λόγω της ιδιότητας $a^b \bmod c = (a \bmod c)^b \bmod c$). Πολλαπλασιάζοντας και τις δύο πλευρές με M

$$M^{k\phi(n)+1} \bmod p = M \tag{3.2}$$

Αν όμως το p διαιρεί το M ισχύει η γενικότερη της (3.2)

$$M^{k\phi(n)+1} \bmod p = M \bmod p \tag{3.3}$$

με τετριμμένο τρόπο, επειδή αμφότερες οι πλευρές της είναι ίσες με μηδέν. Κατά συνέπεια η (3.3) ισχύει είτε το p διαιρεί το M είτε όχι. Ομοίως ισχύει η συμμετρική ιδιότητα

$$M^{k\phi(n)+1} \bmod q = M \bmod q.$$

Επομένως ο αριθμός $M^{k\phi(n)+1} - M$ διαιρείται και από το p και από το q, άρα διαιρείται και από το $pq = n$, οπότε (για $M \leq n$) ισχύει

$$M^{k\phi(n)+1} \bmod n = M$$

άρα τελικά

$$M^{ed} \bmod n = M$$

Παράδειγμα 9. Ας δούμε αρχικά ένα παράδειγμα που δείχνει ότι ισχύει το θεώρημα Euler. Για $n = 14$ οι αριθμοί που είναι πρώτοι προς τον αριθμό n είναι 1, 3, 5, 9, 11, 13, οπότε $\phi(n) = 6$. Εναλλακτικά μπορούμε να χρησιμοποιήσουμε την ιδιότητα $\phi(pq) = \phi(p)\phi(q) = (p-1)(q-1)$ αν p, q είναι πρώτοι αριθμοί και εν προκειμένω $n = 7 \times 2$, άρα $\phi(pq) = 6 \times 1 = 6$. Επομένως εδώ το θεώρημα λέει ότι $a^6 \bmod 14 = 1$. Έστω $a = 3$, άρα πρέπει να ισχύει $3^6 \bmod 14 = 1$ και όντως $3^6 = 729 = 52 \times 14 + 1$.

Ας υποθέσουμε τώρα ότι ψάχνουμε δύο αριθμούς των οποίων το γινόμενο θα είναι $k\phi(14) + 1 = 6k+1$. Οι αριθμοί $6k+1$ για τους λίγους πρώτους ακέραιους k είναι 7, 13, 19, 25, 31, 37, 43, 49, 55 κ.λπ. Εξ αυτών οι 7, 13, 19, 31, 37, 43 είναι πρώτοι και δεν μπορούν να δημιουργήσουν ένα γινόμενο $d \times e$ και απομένουν οι $25 = 5^2$, $49 = 7^2$, $55 = 5 \times 11$ και για να χρησιμοποιήσει κανείς ένα ζεύγος διαφορετικών d, e η πρώτη κατάλληλη λύση είναι $d = 5, e = 11$ ή αντίστροφα.

Ας υποτεθεί τώρα ότι χρησιμοποιείται το ζεύγος $e = 5, d = 11$ για κρυπτογράφηση - αποκρυπτογράφηση και ότι το κείμενο (που δεν πρέπει να υπερβαίνει το $n = 14$) είναι $M = 12$. Τότε η κρυπτογράφηση γίνεται ως $M^e \bmod n = 12^5 \bmod 14 = 248832 \bmod 14 = 17773 \times 14 + 10 \bmod 14 = 10$. Η αποκρυπτογράφηση θα γίνει ως $10^{11} \bmod 14 = 12$, διότι $10^{11} = 7142857142 \times 14 + 12$. □

Παράδειγμα 10. Στο παράδειγμα αυτό οι πράξεις έχουν γίνει με ένα μαθηματικό εργαλείο και δίνονται απ' ευθείας τα αποτελέσματα. Επιλέγονται ως p, q ο δέκατος και ο εικοστός πρώτος αριθμός, αντίστοιχα 29 και 71, οπότε

$$n = pq = 2059, \quad (p-1)(q-1) = 1960$$

Μετά από δοκιμές επιλέγεται $d = 1007$ ώστε μκδ$(d, (p-1)(q-1)) = 1$ και υπολογίζεται ο πολλαπλασιαστικός αντίστροφος $e = 1343$. Ας σημειωθεί ότι το k που υπεισέρχεται στον τύπο $ed = k\phi(n) + 1$ είναι ίσο με $\lfloor ed/(p-1)(q-1) \rfloor = 690$. 'Εστω ότι το κείμενο προς κρυπτογράφηση είναι $M = 1000$. Το κρυπτογραφημένο κείμενο είναι

$$C = 1000^{1343} \bmod 2059 = 1622$$

και η αποκρυπτογράφηση γίνεται ως

$$1622^{1007} \bmod 2059 = 1000. \quad \square$$

Μολονότι για να λειτουργήσει ο αλγόριθμος χρειάζονται οι παράμετροι n, e, d, όταν γίνεται αναφορά στο μήκος κλειδιού του RSA υπονοείται το μήκος του n και αυτό ορίζεται κατά κανόνα στα 1024 ή 2048 ή 4096 bits. Μαζί με το n ορίζονται τα d, e με μήκος περίπου το μισό του n. Το μέγεθος του block του μη κρυπτογραφημένου κειμένου πρέπει να είναι τέτοιο ώστε ο αριθμός M που προκύπτει να μην υπερβαίνει το n. Αν το μήκος του n σε bits είναι s, το μεγαλύτερο δυνατό μήκος block σε bytes είναι $\lfloor (s-1)/8 \rfloor$. Ωστόσο κάποιος μπορεί να διαλέξει μικρότερο μέγεθος προκειμένου να διευκολύνει τους υπολογισμούς. Αλγόριθμοι για την εκτέλεσή τους υπάρχουν και στο αρχικό άρθρο των Rivest, Shamir, Adleman. Εν γένει οι υπολογισμοί είναι αργοί σε σύγκριση με άλλες μεθόδους κρυπτογράφησης, γι' αυτό πιο συχνά ο RSA χρησιμοποιείται για την ασφαλή μεταφορά ενός κλειδιού για ένα συμμετρικό αλγόριθμο, που χρησιμοποιείται στη συνέχεια.

Υπολογισμός του αντιστρόφου

Για τον υπολογισμό του αντίστροφου κλειδιού χρειάζεται κάποια μέθοδος υπολογισμού του πολλαπλασιαστικού αντίστροφου (modular multiplicative inverse).

Υπολογισμός μέσω της συνάρτησης Euler

Ο υπολογισμός $x^{-1} \bmod k$ εφόσον οι x, k είναι πρώτοι μεταξύ τους μπορεί να γίνει μέσω του θεωρήματος του Euler:

$$x^{\phi(k)} \bmod k = 1 \Rightarrow x^{\phi(k)-1} \bmod k = x^{-1} \bmod k$$

Βέβαια η δυνατότητα αξιοποίησης αυτής της μεθόδου υπόκειται στον περιορισμό μκδ$(x, k) = 1$.

Παράδειγμα 11. Ας υποτεθεί ότι ψάχνουμε τον $x = 21^{-1} \bmod 323$. Οι 21, 323 είναι πρώτοι προς αλλήλους και ισχύει $\phi(323) = 288$. Επομένως μπορεί να υπολογισθεί ως $x = 21^{287} \bmod 323 = 200$. $\square$

Στον RSA πρέπει να υπολογισθεί ο $e = d^{-1} \bmod \phi(n)$, όπου ήδη ικανοποιείται η προϋπόθεση να είναι οι $d, \phi(n)$ πρώτοι μεταξύ τους, ωστόσο χρειάζεται να υπολογισθεί το $\phi(\phi(n)) = \phi[(p-1)(q-1)]$, όπου οι $p-1, q-1$ μπορεί να μην είναι πρώτοι.

Παράδειγμα 12. Έστω $n = pq$, όπου $p = 541$ (πρώτος) και $q = 1223$ (επίσης πρώτος), οπότε $n = 661643$ και $\phi(n) = (p-1)(q-1) = 659880$. Πρέπει να επιλέξουμε κατά RSA το d και δεδομένου ότι $p - 1 = 2^2 \times 3^3 \times 5$ και $q - 1 = 2 \times 13 \times 47$ μια ασφαλής επιλογή ώστε να ισχύει μκδ$((p-1)(q-1), d) = 1$ είναι $d = 7 \times 13^3 = 34391$. Επομένως το e υπολογίζεται ως $e = 34391^{-1} \bmod 659880$. Υπολογίζουμε το $\phi(659880) = 158976$ και κατόπιν το $34391^{158976-1} \bmod 659880 = 46031 = e$. Μπορείτε να επαληθεύσετε ότι το υπολογισμένο e είναι όντως ο αντίστροφος του $d \bmod \phi(n)$.

Υπολογισμός μέσω του επεκτεταμένου αλγορίθμου του Ευκλείδη

Ο αλγόριθμος του Ευκλείδη υπολογίζει τον μέγιστο κοινό διαιρέτη ανάμεσα σε δυο φυσικούς αριθμούς. Μια μετατροπή του βασικού αυτού πολυωνυμικού αλγόριθμου μπορεί να δώσει τον υπολογισμό του πολλαπλασιαστικού αντιστρόφου.

Αλγόριθμος του Ευκλείδη Ο αλγόριθμος του Ευκλείδη βασίζεται στην ιδιότητα πως αν $\delta|a$ και $\delta|b$, τότε $\delta|(a-b)$. Έστω ότι θέλουμε τον μ.κ.δ. των a, b όπου $a \geq b > 1$, τότε το πρόβλημα αυτό ανάγεται στο πρόβλημα του μ.κ.δ. των $a - \lfloor a/b \rfloor b, b$, δηλαδή μπορούμε να ελαττώσουμε τον a αφαιρώντας του όσα αντίγραφα του b είναι δυνατό να αφαιρεθούν και να παραμείνει το υπόλοιπο θετικό. Αυτό μπορεί να γίνει διαδοχικά μέχρι να φτάσει το αποτέλεσμα να γίνει μηδέν. Το υπόλοιπο του προηγούμενου βήματος είναι το ζητούμενο αποτέλεσμα.

Παράδειγμα 13. Ας δούμε πρώτα ένα αριθμητικό παράδειγμα. Έστω ότι $a = 731, b = 604$. Το 604 χωράει μόνο μια φορά στο 731, άρα το νέο ζευγάρι είναι $731 - 604 = 127$ και 604. Το 127 χωράει 4 φορές στο 604, άρα το νέο ζευγάρι είναι $604 - 4 \times 127 = 96$ και 127. Αφαιρώντας το 96 από το 127 μένει 31, άρα το νέο ζευγάρι είναι $96, 31$. Το επόμενο είναι $31, 96 - 3 \times 31 = 3$ και τέλος το 3, $31 - 10 \times 3 = 1$. Τέλος το 3, $3 - 3 \times 1 = 0$. Ο υπολογισμός αυτός μπορεί να γραφεί πιο τακτοποιημένα ως εξής:

$$\begin{aligned} 731 - 604 &= 127 \\ 604 - 4 \times 127 &= 96 \\ 127 - 96 &= 31 \\ 96 - 3 \times 31 &= 3 \\ 31 - 10 \times 3 &= 1 \\ 3 - 3 \times 1 &= 0 \end{aligned}$$

Αρα ο μέγιστος κοινός διαιρέτης είναι το 1. □

Επεκτεταμένος αλγόριθμος του Ευκλείδη Το παραπάνω υπολογιστικό σχήμα μπορεί να διατυπωθεί ως εξής, όπου οι αρχικοί αριθμοί είναι οι $a = r_0, b = r_1$ τέτοιοι ώστε $r_0 > r_1 > 0$:

$$\begin{aligned} r_2 &= r_0 - q_1 r_1 \quad 0 < r_2 < r_1 \\ r_3 &= r_1 - q_2 r_2 \quad 0 < r_3 < r_2 \\ &\dots \\ r_m &= r_{m-2} - q_{m-1} r_{m-1} \quad 0 < r_m < r_{m-1} \\ 0 &= r_{m-1} - q_m r_m \end{aligned}$$

Στη συνέχεια μπορεί κανείς να αποδείξει εύκολα [SP18] ότι οι παραπάνω συντελεστές r_i συνδέονται με τις αρχικές τιμές r_0, r_1 μέσω του τύπου

$$r_i = s_i r_0 + t_i r_1 \tag{3.4}$$

όπου

$$t_i = \begin{cases} 0 & \text{if } i = 0 \\ 1 & \text{if } i = 1 \\ t_{i-1} - q_{i-1}t_{i-1} & \text{if } i \geq 2 \end{cases}$$

και

$$s_i = \begin{cases} 1 & \text{if } i = 0 \\ 0 & \text{if } i = 1 \\ s_{i-1} - q_{i-1}s_{i-1} & \text{if } i \geq 2 \end{cases}$$

Η (3.4) δείχνει ότι με αυτόν τον υπολογισμό από δύο ακέραιους a, b μπορούν να υπολογισθούν ακέραιοι r, s, t τέτοιοι ώστε $r = \text{mkd}(a, b)$ και $r = sa + tb$. Ο σχετικός υπολογισμός αποτελεί τον *επεκτεταμένο αλγόριθμο του Ευκλείδη*. Στην περίπτωση που μκδ$(a, b) = 1$ παίρνοντας στην $r = sa + tb$ το υπόλοιπο ως προς a προκύπτει $tb \bmod a = 1$, δηλαδή $t = b^{-1} \bmod a$. Επομένως ας ισχύουν οι κατάλληλες προϋποθέσεις ο επεκτεταμένος αλγόριθμος του Ευκλείδη μπορεί να χρησιμοποιηθεί για την αντιστροφή ακεραίου.

Στον RSA το ένα εκ των d, e λαμβάνεται αυθαίρετα μεν, αλλά με την προϋπόθεση να είναι πρώτο προς τον $(p-1)(q-1)$, και υπολογίζεται το άλλο, δηλ. αν επιλεγεί το d, τότε υπολογίζεται το $e = d^{-1} \bmod (p-1)(q-1)$. Κατά συνέπεια υπάρχουν οι προϋποθέσεις χρήσης του αλγόριθμου. Στη συνέχεια δίνουμε ένα παράδειγμα εκτέλεσης του υπολογισμού.

Παράδειγμα 14. Ας υποτεθεί ότι ψάχνουμε τον $x = 21^{-1} \bmod 323$. Ο υπολογισμός φαίνεται στον ακόλουθο πίνακα. Αρχικά υπολογίζονται οι στήλες r_i και q_i όπως στον αλγόριθμο του Ευκλείδη. Στη συνέχεια μπορούν να υπολογισθούν τα t_i, ενώ τα s_i μολονότι μπορούν να υπολογισθούν δεν έχουν χρησιμότητα.

	r_i	q_i	t_i
0	323		0
1	21	15	1
2	8	2	-15
3	5	1	31
4	3	1	-46
5	2	1	77
6	1		-123

Επομένως $x = -123 \bmod 323 = 200$ □.

Επιθέσεις στον RSA

Το πρόβλημα αποκρυπτογράφησης ενός κρυπτογραφημένου κειμένου χωρίς γνώση του κλειδιού d συνίσταται στον υπολογισμό του M όταν είναι γνωστό το $C = M^e \bmod n$, καθώς και τα n, e. Κατά συνέπεια πρόκειται για τον υπολογισμό της e-οστής ρίζας ενός αριθμού modulo n.Ωστόσο δεν είναι γνωστός κάποιος αποδοτικός αλγόριθμος που θα μπορούσε να κάνει τον υπολογισμό. Γι' αυτό οι συνήθεις επιθέσεις περνούν μέσα από την ανακάλυψη του μυστικού κλειδιού d. Δείτε πρώτα το παρακάτω παράδειγμα.

Παράδειγμα 15. Ας υποτεθεί ότι ο Bob έχει στείλει το κρυπτογραφημένο μήνυμα $C = 1622$ του Παραδείγματος 10, όπου είναι γνωστές επίσης οι παράμετροι $n = 2059$ και $e = 1343$. Τα παραπάνω είναι όλα γνωστά και στην Trudy, η οποία μπορεί να υπολογίσει αρχικά το $\phi(2059) = 1960$ είτε απ' ευθείας είτε κάνοντας παραγοντοποίηση του n. Η τελευταία δίνει $2059 = 29 \times 71$, πράγμα που επιτρέπει τον υπολογισμό του $\phi(n) = 28 \times 70 = 1960$.

Στη συνέχεια μπορεί να υπολογίσει τον $1341^{-1} \bmod 1960 = 1007$ και να κάνει κανονικά τον υπολογισμό αποκρυπτογράφησης. □

Η μέθοδος της Trudy στο παράδειγμα φαίνεται απλή, αλλά στον πραγματικό κόσμο οι αριθμοί C, n, e είναι τόσο μεγάλοι που κάνουν πολύ δύσκολους τους υπολογισμούς. Ο πιο δύσκολος από αυτούς τους υπολογισμούς είναι αυτός του $\phi(n)$ από το n και ακόμη και αν δεν γίνει μέσω παραγοντοποίησης του n δεν είναι γνωστός κάποιος αλγόριθμος πιο εύκολος από αυτούς που είναι γνωστοί για την παραγοντοποίηση.

Το 1991 τα RSA Laboratories δημοσίευσαν μια σειρά ακεραίων με διαφορετικά μήκη, από 100 ως 2048 δεκαδικά ψηφία, δημιουργώντας έτσι τον διαγωνισμό *RSA Factoring Challenge* και προσφέροντας χρηματικές αμοιβές. Ο διαγωνισμός τυπικά τελείωσε το 2007, αλλά οι προσπάθειες συνεχίστηκαν. Ο μεγαλύτερος ακέραιος που έχει ως τώρα (2020) παραγοντοποιηθεί είναι μήκους 250 δεκαδικών ψηφίων ή 829 bits. Ο αριθμός αυτός πλησιάζει πλέον το μήκος κλειδιού των 1024 bits, γεγονός που ωθεί την επιλογή κλειδιών προς τα 2048 bits.

3.2 Ανταλλαγή κλειδιών κατά Diffie-Hellman

Το 1974 ο Ralph Merkle είχε επινοήσει μια μέθοδο που χρησιμοποιούσε μια σειρά από αινίγματα και επέτρεπε σε δυο πλευρές να ανταλλάξουν ένα μυστικό [Mer78].[3] Η ανταλλαγή κλειδιών με τη μέθοδο των Whitfield Diffie και Martin Hellman [DH76] ήρθε λίγο αργότερα και επιτρέπει σε δύο πλευρές να συμφωνήσουν σε μια αριθμητική τιμή. Η μέθοδος αυτή έχει ενσωματωθεί σε μια σειρά πρωτοκόλλων του Διαδικτύου ώστε να συμφωνούν στην τιμή ενός κλειδιού και στη συνέχεια οι δύο πλευρές που τα χρησιμοποιούν να μπορούν να κρυπτογραφούν τα μεταξύ τους μηνύματα με μια συμμετρική μέθοδο. Τέτοια πρωτόκολλα είναι το Hypertext Transport Protocol Secure (HTTPS), το Secure Shell (SSH), το Internet Protocol Security (IPsec) και το Simple Mail Transfer Protocol Secure (SMTPS).

Σε συνέντευξη για το ευρύ κοινό που έδωσε ο Hellman το 2017 περιέγραψε τον αλγόριθμο με απλά λόγια ως εξής: Ο αλγόριθμος απλοποιεί τη διαδικασία ανταλλαγής κλειδιών, επιτρέποντας στις δύο πλευρές να κάνουν αρχικά μια ανοιχτή σε όλους συνεννόηση για το πρωτόκολλο που θα χρησιμοποιήσουν, ύστερα να κάνει η κάθε πλευρά τις δικές της μαθηματικές πράξεις και στο τέλος αυτή η διαδικασία τους επιτρέπει να καταλήξουν σε μια πληροφορία που ένας τρίτος δεν μπορεί να αποκτήσει. Και συμπλήρωσε ότι οποιοσδήποτε τρίτος θα χρειαστεί υπερβολική υπολογιστική δύναμη για να αποκτήσει την ίδια πληροφορία.[4]

Σε σύγκριση με τον αλγόριθμο Diffie-Hellmann ο RSA εμφανίζεται να κάνει πιο «ολοκληρωμένη» δουλειά, με την έννοια ότι ο πρώτος επιτρέπει την ανταλλαγή ενός κλειδιού για την περαιτέρω συμμετρική κρυπτογράφηση ενός μηνύματος, ενώ ο δεύτερος μπορεί να κρυπτογραφήσει απ' ευθείας το μήνυμα. Στην πράξη όμως και οι δύο αλγόριθμοι χρησιμοποιούνται για την ανταλλαγή ενός συμμετρικού κλειδιού, επειδή η απ' ευθείας κρυπτογράφηση με τον RSA είναι υπολογιστικά πιο δύσκολη από τη συμμετρική κρυπτογράφηση.

[3] Το άρθρο του R. Merkle δημοσιεύθηκε το 1978, είχε όμως υποβληθεί το 1975. Το άρθρο των Diffie-Hellmann είχε υποβληθεί τον Ιούνιο του 1976 και δημοσιεύθηκε το Νοέμβρη του 1976.

[4] A.M. Turing Award, An Interview with Martin Hellman Recipient of the 2015 ACM Turing Award. Interviewer: Hugh Williams May 19, 2017, Palo Alto, California, `https://amturing.acm.org/pdf/HellmanTuringTranscript.pdf`.

Αρχική ρίζα (primitive root) πρώτου αριθμού

Δεδομένου ενός αριθμού n *αρχική ρίζα* (primitive root) a του ακέραιου n είναι ένας αριθμός τέτοιος ώστε κάθε αριθμός που είναι πρώτος προς τον n να συμπίπτει με κάποια δύναμη του a mod n.Ένας αριθμός n μπορεί να έχει περισσότερες από μια αρχικές ρίζες. Ωστόσο δεν υπάρχουν αρχικές ρίζες για όλους τους αριθμούς. Ενδεικτικά, οι πρώτοι είκοσι αριθμοί που έχουν αρχικές ρίζες είναι οι εξής: 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 13, 14, 17, 18, 19, 22, 23, 25, 26, 27. Από αυτόν τον περιορισμένο κατάλογο φαίνεται ότι το 8, το 12, το 15 και το 16 δεν έχουν αρχική ρίζα.

Παράδειγμα 16. Το 14 έχει δύο αρχικές ρίζες, το 3 και το 5. Οι αριθμοί 5^i mod 14 για τις πρώτες λίγες τιμές του i αρχίζοντας από $i = 1$ είναι οι εξής: 1, 5, 11, 13, 9, 3, 1, 5, 11, 13, 9, 3, 1, 5, 11, 13, 9, 3, . . ., δηλ. επαναλαμβάνονται περιοδικά. Δείτε ότι εμφανίζονται μόνο οι αριθμοί που είναι πρώτοι προς το 14, ενώ δεν εμφανίζονται ζυγοί (επειδή έχουν με το 14 κοινό διαιρέτη το 2), καθώς και το 7 (με κοινό διαιρέτη το 7). □

Στην ειδική περίπτωση που ο αριθμός n είναι πρώτος, (α) υπάρχει θεώρημα που εγγυάται ότι έχει μια τουλάχιστον αρχική ρίζα a και (β) όλοι οι αριθμοί από το 1 ως το $n - 1$ είναι πρώτοι προς τον n, άρα το σύνολο $\{a^i \bmod n \,|\, i = 1, 2, \ldots, n - 1\}$ ταυτίζεται με το σύνολο $\{1, 2, \ldots, n - 1\}$, δηλαδή οι αριθμοί a^i mod n είναι ίδιοι με τους αριθμούς $1, 2, \ldots, n - 1$, αλλά σε άλλη σειρά.

Επομένως για κάθε πρώτο αριθμό n και για κάθε αριθμό x μεταξύ 0 και $n - 1$ μπορεί κανείς να βρει ένα μοναδικό αριθμό i με την ιδιότητα

$$x = a^i \bmod n$$

Ο αριθμός i λέγεται *διακριτός λογάριθμος* (discrete logarithm) του x.

Παράδειγμα 17. Το 13 έχει αρχικές ρίζες τους αριθμούς $2, 6, 7, 11$. Οι αριθμοί 7^i mod 13 για $i = 1, 2, \ldots, 12$ είναι κατά σειρά οι εξής: $7, 10, 5, 9, 11, 12, 6, 3, 8, 4, 2, 1$. □

Αλγόριθμος Diffie-Hellman

Ο τρόπος με τον οποίο καταλήγουν σε ένα κοινό αριθμό, συνήθως ένα κοινό κλειδί, δύο πλευρές, ας υποτεθεί ότι είναι ο Bob και η Alice κατά τα συνήθη, έχει ως εξής:

1. Δίνονται (δημόσια) ένας πρώτος αριθμός p και μια αρχική του ρίζα a.
2. Ο Bob επιλέγει ως ιδιωτικό κλειδί ακέραιο $X_B < p$ και υπολογίζει το δημόσιο κλειδί του ως $Y_B = a^{X_B} \bmod p$.
3. Η Alice επιλέγει ως ιδιωτικό κλειδί ακέραιο $X_A < p$ και υπολογίζει το δημόσιο κλειδί της ως $Y_A = a^{X_A} \bmod p$.
4. Ο Bob υπολογίζει το $K_B = Y_A^{X_B} \bmod p$.
5. Η Alice υπολογίζει το $K_A = Y_B^{X_A} \bmod p$.
6. Όμως $K_B = K_A = K$, οπότε το K μπορεί να χρησιμοποιηθεί ως γνωστό και στους δύο μυστικό κλειδί για συμμετρική κρυπτογράφηση.

Η ισότητα $K_A = K_B$ εξασφαλίζεται από το γεγονός ότι

$$K_A = Y_B^{X_A} \bmod p = (a^{X_B} \bmod p)^{X_A} \bmod p == a^{X_A X_B} \bmod p$$

και ομοίως υπολογίζεται το $K_B = a^{X_A X_B} \bmod p$. Δεδομένου ότι η ισότητα $K_A = K_B$ εξασφαλίζεται για οποιαδήποτε τιμή του a, σε τι εξυπηρετεί το γεγονός ότι το a είναι αρχική ρίζα του p; Το αποτέλεσμα των πράξεων $a^{X_A} \bmod p$ ούτως ή άλλως καταλήγει μέσα στο σύνολο $\mathbb{Z}_p = \{0, \ldots, p-1\}$, όταν όμως το a δεν είναι αρχική ρίζα οι τιμές που προκύπτουν ανήκουν σε γνήσιο υποσύνολο του $\mathbb{Z}_p$ με αποτέλεσμα την μη ικανοποιητική διασπορά τους. Αντίθετα, αν το a είναι αρχική ρίζα, οι τιμές διασπείρονται σε όλο το $\mathbb{Z}_p$, επομένως εξαντλητικές δοκιμές κλειδιών από έναν αντίπαλο γίνονται πιο δύσκολες. Βασικά στην περίπτωση του αλγορίθμου Diffie-Hellmann η Trudy θα μπορούσε να αποπειραθεί να μάθει το ιδιωτικό κλειδί της Alice μέσω της αντιστροφής της $Y_A = a^{X_A} \bmod p$, δηλαδή υπολογίζοντας τον διακριτό λογάριθμο του Y_A με βάση το a modulo p.

ΚΕΦΑΛΑΙΟ 4

Συναρτήσεις κατακερματισμού

Συνάρτηση κατακερματισμού ή κατατεμαχισμού (hash function) είναι μια συνάρτηση που απεικονίζει μια σειρά συμβόλων (string) αυθαιρέτου μήκους σε μια σειρά με σταθερό μήκος (και όχι απαραίτητα από το ίδιο σύνολο συμβόλων). Συνήθως η δεύτερη σειρά είναι μικρότερου μήκους από την πρώτη. Σε πολλές εφαρμογές αμφότερες είναι ακολουθίες από bits (bit strings).

Υπάρχουν πολλές και διάφορες εφαρμογές των συναρτήσεων κατακερματισμού. Μερικές ενδεικτικές είναι οι εξής:

- Αν κάποιος θέλει να προστατεύσει ένα κείμενο ή αρχείο από αλλοιώσεις μπορεί να αποθηκεύσει (με ασφαλή τρόπο) μια τιμή κατακερματισμού και κάθε φορά που εγείρεται αμφιβολία για την ακεραιότητα του κειμένου να επανυπολογίζεται η τιμή και να συγκρίνεται με την αποθηκευμένη.[1]

- Σε μια βάση δεδομένων οι πληροφορίες μιας εγγραφής που χαρακτηρίζεται με ένα κλειδί x (π.χ. το ονοματεπώνυμο ενός ατόμου) εγγράφονται στη θέση $y = h(x)$, όπου h είναι μια συνάρτηση κατακερματισμού της οποίας οι τιμές είναι θέσεις της μνήμης. Σε μια τέτοια εφαρμογή πρέπει οι διαθέσιμες θέσεις της μνήμης να είναι όσες και οι δυνατές τιμές του y. Επειδή το y προκύπτει με ψευδοτυχαίο τρόπο από το x, για να αποφεύγονται οι συμπτώσεις διαφορετικών x στο ίδιο y πρέπει το σύνολο τιμών να έχει αρκετά μεγαλύτερο πληθικό αριθμό από το σύνολο τιμών του x (και επί πλέον να υπάρχει μηχανισμός επίλυσης των συγκρούσεων). Αυτό οδηγεί σε σπατάλη μνήμης.

- Σε ένα σύστημα ελέγχου πρόσβασης με όνομα (login name) και συνθηματικό (password) αντί να αποθηκεύεται για κάθε όνομα το αντίστοιχο συνθηματικό αποθηκεύεται σε αρχείο η τιμή κατακερματισμού (hash) του συνθηματικού. Όταν κάποιος ζητήσει πρόσβαση και δώσει το συνθηματικό, αυτό περνάει από την ίδια συνάρτηση κατακερματισμού και ελέγχεται αν η τιμή αντιστοιχεί στο όνομα σύμφωνα με το υπάρχον αρχείο.

- Σε ένα σύστημα anti-virus ευαίσθητα αρχεία που δεν μεταβάλλονται μπορούν να προστατευθούν με μια συνάρτηση κατακερματισμού. Η τιμή της συνάρτη-

[1] Μια περίπτωση τέτοιας χρήσης είχαμε δει για τον νόμο του Hooke στη σελ. 15.

σης για κάθε αρχείο επανυπολογίζεται κάθε τόσο και αν διαπιστωθεί απόκλιση λαμβάνονται μέτρα.

- Ευρεία χρήση των συναρτήσεων κατακερματισμού γίνεται στα κρυπτονομίσματα και στα κατανεμημένα κατάστιχα.

Η βασική λειτουργία μιας συνάρτησης κατακερματισμού συνίσταται στην προστασία της ακεραιότητας ενός κειμένου.[2] Τον σκοπό αυτόν θα μπορούσε να εξυπηρετήσει μια κρυπτογραφική λύση, μα συχνά μπορεί να είναι υπερβολική για τον επιδιωκόμενο σκοπό.

4.1 Κρυπτογραφικές λύσεις

Μια «απλή» λύση για την προστασία ενός κειμένου που θέλει να στείλει ο Bob στην Alice είναι η κρυπτογράφηση με το κοινό τους κλειδί, εφόσον τουλάχιστον ληφθούν προφυλάξεις ώστε να μην προκύπτουν ζητήματα παρόμοια με αυτά που περιγράφτηκαν στην προηγούμενη παράγραφο. Η κρυπτογραφική λύση όχι μόνο προφυλάσσει το κείμενο, αλλά ικανοποιεί και άλλες απαιτήσεις. Πρώτα πρώτα πείθει την Alice ότι προέρχεται από τον Bob. Αν ο Bob θέλει να χρονοσφραγίσει το κείμενο αρκεί να προσθέσει σ᾽ αυτό μια χρονική ένδειξη, την οποία η Trudy όχι μόνο δεν θα μπορεί να αλλοιώσει, αλλά ούτε καν να διαβάσει.

Σε ένα άλλο σενάριο ένα κείμενο πρέπει να φτάσει σε πολλούς αποδέκτες και να είναι αναγνώσιμο από όλους, ενώ πρέπει να διαπιστώνεται ότι είναι αυθεντικό. Για παράδειγμα, ένα μήνυμα του 112 πρέπει να είναι ορατό σε όλους, αλλά και να μπορεί να διαπιστωθεί ότι είναι αυθεντικό και δεν πρόκειται για φάρσα. Για τέτοιες περιπτώσεις μπορεί και πάλι να υιοθετηθεί μια κρυπτογραφική λύση, καταφεύγοντας αυτή τη φορά στη μη συμμετρική κρυπτογραφία. Ο αποστολέας μπορεί να το στείλει με το ιδιωτικό του κλειδί και οι παραλήπτες να το αποκρυπτογραφήσουν με το δημόσιο κλειδί.

Μια συμμετρική κρυπτογραφική λύση είναι επίσης δυνατή όταν μπορεί να προστατευθεί με άλλο τρόπο η τιμή κατακερματισμού. Ως συνάρτηση κατακερματισμού μπορεί να χρησιμοποιηθεί η κρυπτογράφηση με ένα φανερό κλειδί και να κρατηθούν επιλεγμένα n bits από το κρυπτογραφημένο κείμενο (στην πιο απλή περίπτωση τα n πρώτα).

Ωστόσο η κρυπτογραφική λύση απαιτεί συνήθως περίπλοκους υπολογισμούς και μπορεί σε σχέση με τον επιδιωκόμενο σκοπό να είναι υπερβολική από πλευράς πόρων που θα απαιτηθούν. Μια περίπτωση είναι να υπάρχει ανάγκη για δειγματοληπτικό μόνο έλεγχο ενός κειμένου. Εδώ είναι περιττό να υποχρεωθούν όλοι να κάνουν μια αποκρυπτογράφηση για να το διαβάσουν. Σε κάποιες περιπτώσεις ένας πληθυσμός τερματικών που πρέπει να παραλάβει ένα μήνυμα μπορεί να περιλαμβάνει και χαμηλών δυνατοτήτων τερματικά που δεν θα μπορούσαν να κάνουν ιδιαίτερους υπολογισμούς.

Οι συναρτήσεις κατακερματισμού δίνουν συχνά μια εύκολη απάντηση σε σχετικά απλές ανάγκες, αλλά εφόσον απαιτηθεί μπορούν επίσης να γίνουν αρκετά πολύπλοκες ώστε να μπορούν να καλύπτουν απαιτήσεις για αυξημένη ασφάλεια.

Ας σημειωθεί επίσης ότι οι κρυπτογραφικές λύσεις δεν είναι πάντοτε ασφαλείς. Η Trudy μπορεί σε κάποιες περιπτώσεις να αλλοιώσει με ελεγχόμενο τρόπο ένα κείμενο ακόμη και αν είναι κρυπτογραφημένο. Στην περίπτωση που ο Bob στέλνει στην Alice ένα κείμενο x το οποίο έχει κάνει XOR (bit προς bit) με ένα κείμενο z, με αποτέλεσμα

[2] Η λέξη *κείμενο* σ᾽ αυτό το κεφάλαιο, όπως εξ άλλου και στα κεφάλαια τα σχετικά με κρυπτογραφία, χρησιμοποιείται αντί της λέξης *συμβολοσειρά* ή *string* χωρίς ιδιαίτερη διάκριση.

την παραγωγή ενός κειμένου $y = x \oplus z$, η Trudy μπορεί αλλοιώσει συγκεκριμένα bits στο y, παράγοντας ένα κείμενο $y' = y \oplus r$ (όπου προφανώς το r έχει την τιμή 1 μόνο στις θέσεις αναστροφής). Όταν η Alice σχηματίσει το $y' \oplus z$ αυτό είναι ίσο με $x \oplus r$, δηλαδή οι αλλοιώσεις θα βρεθούν στις ίδιες ακριβώς θέσεις που τις έβαλε η Trudy [SP18]. Η επίθεση αυτή είναι γνωστή ως επίθεση αναστροφής των bits (bit flipping attack). Σε ένα γενικό κείμενο ίσως η Trudy να μη γνωρίζει ποια bits είναι σκόπιμο να αλλοιώσει, αλλά σε ένα κείμενο με σταθερή δομή, π.χ. σε μια επιταγή όπου το ποσό είναι σε συγκεκριμένη θέση, ίσως να γνωρίζει πού να επιτεθεί. Σε ένα ακόμη χειρότερο σενάριο ο Bob κρυπτογραφεί το κείμενο για να μπορεί να βεβαιώσει την Alice ότι αυτός είναι ο αποστολέας, ενώ αδιαφορεί για την μυστικότητα, οπότε η Trudy γνωρίζει το αρχικό κείμενο x και μπορεί να το μετασχηματίσει σε οποιοδήποτε άλλο κείμενο x' επιθυμεί (θέτοντας $r = x \oplus x'$). Στην περίπτωση αυτήν η Alice δεν είναι σε θέση να αντιληφθεί την αλλοίωση, ενώ παραμένει βέβαιη ότι το κείμενο το έχει στείλει ο Bob.

4.2 Απλές συναρτήσεις κατακερματισμού

Ως μια από τις πιο απλές συναρτήσεις κατακερματισμού θα μπορούσε να θεωρηθεί η μέθοδος του Hooke που ήδη έχει αναφερθεί στη σελίδα 15. Ο Hooke ξεκίνησε από τη φράση *ut tensio sic vis* (που περιγράφει συνοπτικά στη λατινική γλώσσα το νόμο του Hooke για τα ελατήρια, ότι δηλαδή η επιμήκυνση είναι ανάλογη της εφαρμοζόμενης δύναμης) και παρουσίασε τη «λέξη» *ceiiinosssttuv* βάζοντας απλώς και μόνο τα γράμματα σε αλφαβητική σειρά. Θα μπορούσαμε να πούμε ότι χρησιμοποίησε μια συνάρτηση κατακερματισμού h που βάζει τα γράμματα σε αλφαβητική σειρά. Ο σκοπός του Hooke ήταν να κατοχυρώσει τη γνώση του νόμου στον εαυτό του πριν τον ανακοινώσει δημόσια, πράγμα που έκανε δυο χρόνια αργότερα. Κατά κάποιο τρόπο χρησιμοποίησε τον κατακερματισμό στο ρόλο μιας χρονοσφραγίδας. Το εγχείρημά του ήταν ασφαλές στο βαθμό που δεν θα μπορούσε κάποιος άλλος να παρουσιάσει μια άλλη φράση που θα έβγαζε νόημα και θα κατέληγε στο ίδιο αποτέλεσμα *ceiiinosssttuv*, ώστε να ισχυριστεί ότι ο Hooke δεν είχε στο μυαλό του το *ut tensio sic vis* αλλά κάτι άλλο.

Το επόμενο παράδειγμα δείχνει ότι μπορούμε να επινοήσουμε απλές συναρτήσεις κατακερματισμού, αλλά είναι πιθανό να μπορούν να υποστούν εξ ίσου απλές επιθέσεις.

Παράδειγμα 18. Ένα κείμενο σε υπολογιστή, που μπορεί επομένως να θεωρηθεί ως ένα bit string αυθαιρέτου μήκους, θα μπορούσε να τεμαχισθεί σε τμήματα μήκους n bits και να σχηματισθούν n bits «ισοτιμίας» c_i αθροίζοντας τα i-οστά bits κάθε τμήματος. Δηλαδή αν $b_1, b_2, \ldots$ είναι τα bits του κειμένου, το i-οστό bit της τιμής κατακερματισμού είναι

$$c_i = b_i \oplus b_{i+n} \oplus b_{i+2n} \oplus \ldots \quad (i = 1, 2, \ldots, n)$$

Προφανώς η παραγόμενη τιμή κατακερματισμού έχει σταθερό μήκος n ανεξάρτητο από το μήκος του αρχικού κειμένου. Αυτή όμως η μέθοδος κατακερματισμού μπορεί εύκολα να παραβιασθεί από την Trudy, η οποία μπορεί να αλλοιώσει όσο θέλει το κείμενο, αρκεί να βρει μια προσθήκη μήκους n bits (ή και μεγαλύτερη) τέτοια ώστε τα παραγόμενα c_i να είναι τα ίδια. Αυτό μπορεί να έχει μια μικρή δυσκολία αν το κείμενο είναι γραμμένο σε φυσική γλώσσα, αλλά πιο εύκολο αν περιέχει αριθμούς και άλλα μέρη με μικρό βαθμό πλεονασμού. Επίσης, οποιαδήποτε αναδιάταξη των τμημάτων μήκους n bits οδηγεί στο ίδιο αποτέλεσμα για την τιμή του $c_1 c_2 \ldots c_n$. □

Κατά κανόνα το αποτέλεσμα μιας συνάρτησης κατακερματισμού είναι σταθερού και μικρού μήκους σε σύγκριση με το αρχικό κείμενο, το οποίο είναι μακρύτερο και

μπορεί να έχει οποιοδήποτε μήκος. Επομένως τα δυνατά κείμενα εισόδου είναι κατά τάξεις μεγέθους περισσότερα από τις δυνατές τιμές κατακερματισμού, πράγμα που έχει ως συνέπεια να υπάρχουν πολλά διαφορετικά κείμενα που καταλήγουν στην ίδια τιμή κατακερματισμού. Σπανίως όμως ένα αντίπαλος θα χρειαστεί να βρει πολλά τέτοια κείμενα ή όλα. Συνήθως του αρκούν ένα ή δύο και μια επίθεση με brute force αποσκοπεί στο να βρεθεί ένα κείμενο που καταλήγει σε δεδομένη τιμή κατακερματισμού (ή δύο κείμενα που καταλήγουν σε μια οποιαδήποτε ίδια τιμή). Ο αντίπαλος μπορεί να δοκιμάζει κείμενα μέχρι να καταλήξει σε κάποιο που βγάζει την επιθυμητή τιμή. Αν η τιμή κατακερματισμού είναι μήκους n bits όλες οι τιμές είναι 2^n και με ένα τέτοιο αριθμό δοκιμών είναι σχεδόν βέβαιο στατιστικά ότι θα επιτύχει το σκοπό του (αν και η πιθανότητα δεν θα γίνει αυστηρά ίση με 1 με οσοδήποτε μεγάλο αριθμό δοκιμών). Ακόμη και με τις μισές δοκιμές έχει πολύ σοβαρή πιθανότητα επιτυχίας που πλησιάζει το 50%. Τις πιθανότητες αυτές θα υπολογίσουμε παρακάτω, αυτό όμως που έχει σημασία να καταλάβουμε τώρα είναι ότι το μήκος της τιμής κατακερματισμού αποτελεί βασικό μέγεθος για την ασφάλειά της, αν τουλάχιστον δεν υπάρχουν άλλες αδυναμίες που μπορούν να οδηγήσουν σε μαθηματικές επιθέσεις.

Η βασική αδυναμία μιας συνάρτησης κατακερματισμού $h(x)$ είναι να είναι αναστρέψιμη και να είναι εφικτός πρακτικά ο υπολογισμός $h^{-1}(x)$.

Παράδειγμα 19. Αν θεωρήσουμε πως το αρχικό κείμενο x είναι ένα οποιοδήποτε bit string, ίσως η πρώτη συνάρτηση που θα μας ερχόταν στο μυαλό και που παράγει ένα string μήκους n είναι η

$$h(x) = x \mod 2^{n+1}$$

όπου το x και το $y = h(x)$ είναι προφανώς δυαδικοί αριθμοί. Αν

$$x = 11110100001001000000$$

(που είναι στο δεκαδικό σύστημα ίσο με 10^6) και χρειαζόμαστε τιμή με 8 bits, το υπόλοιπο της διαίρεσης του x με το $2^9 = 512$ είναι 64, άρα $h(x) = 01000000$. Ωστόσο αν δοθεί το $y = 64_{10} = 01000000_2$, οποιοσδήποτε αριθμός $y + k2^{n+1}$ ($k = 0, 1, 2, \ldots$) είναι κατάλληλος ως x. Οι 4 πρώτοι τέτοιοι αριθμοί είναι 1000000, 1001000000, 10001000000, 11001000000. □

4.3 Ιδιότητες των συναρτήσεων κατακερματισμού

Μολονότι οι επιθυμητές ιδιότητες μιας συνάρτησης κατακερματισμού διαφοροποιούνται εν μέρει ανάλογα με τη χρήση, μια γενικά επιθυμητή ιδιότητα είναι αυτή του μη αναστρέψιμου. Πρέπει δηλαδή να είναι σχετικά εύκολο να υπολογισθεί το $h(x)$ όταν δοθεί το x, αλλά να είναι εξαιρετικά δύσκολο να υπολογισθεί το x αν είναι γνωστό το $h(x)$. Η ιδιότητα αυτή εκφράζεται με διάφορους τρόπους που θα δούμε στη συνέχεια.

Μια άλλη ιδιότητα, που όμως σχετίζεται με την προηγούμενη, είναι να προκύπτουν τιμές της συνάρτησης κοντινές με αυτές που θα έβγαζε μια γεννήτρια τυχαίων αριθμών ορισμένη πάνω στο ίδιο σύνολο τιμών. Για παράδειγμα, αν οι τιμές κατακερματισμού είναι δυαδικές με μήκος n bits, αυτές οι τιμές να δίνουν την εντύπωση ότι είναι ομοιόμορφα τυχαίες στο διάστημα των ακεραίων από 0 ως $2^n - 1$. Μια ανάλογη απαίτηση διατυπώνεται και για το κρυπτογραφημένο κείμενο ενός κώδικα, δηλαδή να μοιάζει με μια σειρά από τυχαία bits.

Όταν δοθεί μια τιμή y και αναζητείται ένα (τουλάχιστον) x τέτοιο ώστε $y = h(x)$ και δεν υπάρχει κανένας αποδοτικός τρόπος υπολογισμού της $h^{-1}(y)$, η λύση που απομένει στην Trudy είναι να χρησιμοποιήσει μια σειρά από τυχαίες τιμές του x' και να υπολογίσει για κάθε μια το $h(x')$ μήπως και πέσει πάνω στο y. Εννοείται ότι η λύση

στο πρόβλημα της αντιστροφής κατά κανόνα δεν είναι μοναδική, εφόσον οι δυνατές τιμές εξόδου είναι συνήθως λιγότερες από τις δυνατές τιμές εισόδου (εκτός ίσως από την περίπτωση του hash table σε μια βάση δεδομένων). Η Trudy θεωρητικά έχει την δυνατότητα να αρχίσει μια εξαντλητική αναζήτηση, δηλαδή να προσπαθήσει να υπολογίσει το $h(x)$ για όλες τις δυνατές τιμές του x. Αν όμως υποτεθεί ότι οι τιμές εξόδου y ακολουθούν ένα περίπου τυχαίο pattern ανάμεσα σε N δυνατές τιμές εξόδου, αρκεί κατά μέσο όρο να κάνει $N/2$ προσπάθειες είτε αυτές γίνουν με τυχαία σειρά των x είτε με κάποια προκαθορισμένη διάταξη.

Προβλήματα αντιστροφής

Προκειμένου να διατυπώσουμε τις επιθυμητές ιδιότητες μιας συνάρτησης κατακερματισμού δίνονται στη συνέχεια ορισμένες παραλλαγές του προβλήματος αντιστροφής. Τα προβλήματα αυτά είναι καλό να μην έχουν εύκολη λύση, επειδή είναι προβλήματα που επιθυμεί να λύσει ένας αντίπαλος.

Στους ορισμούς των προβλημάτων το σύνολο των δυνατών κειμένων (ή strings) παριστάνεται με το $\mathcal{X}$, ενώ το σύνολο των δυνατών τιμών κατακερματισμού παριστάνεται με το $\mathcal{Y}$. Τα στοιχεία του $\mathcal{X}$ είναι κείμενα οποιουδήποτε μήκους, παρμένα μέσα από κάποιο αλφάβητο, ενώ οι τιμές κατακερματισμού, δηλαδή τα στοιχεία του $\mathcal{Y}$, μπορεί εν γένει να είναι από άλλο αλφάβητο. Στην πράξη οι τιμές κατακερματισμού έχουν σταθερό μήκος, αλλά η ιδιότητα αυτή δεν είναι απαραίτητη για τους παρακάτω ορισμούς προβλημάτων. Στη συνήθη περίπτωση που η τιμή κατακερματισμού είναι ένα bit string μήκους n, ο πληθικός αριθμός του $\mathcal{Y}$ είναι προφανώς $|\mathcal{Y}| = 2^n$. Επίσης, μολονότι υποτίθεται ότι το $\mathcal{X}$ είναι πεπερασμένο σύνολο ($|\mathcal{X}| < \infty$), θα δούμε σε επόμενη ενότητα ότι μπορούμε να κατασκευάσουμε μια συνάρτηση κατακερματισμού για κείμενα χωρίς περιορισμό μήκους.

Πρόβλημα 1. [Προ-εικόνα (pre-image)]
Δίνεται συνάρτηση κατακερματισμού $h : \mathcal{X} \to \mathcal{Y}$ και $y \in \mathcal{Y}$. Να υπολογισθεί $x \in \mathcal{X}$ τέτοιο ώστε $h(x) = y$. □

Πρόβλημα 2. [Δεύτερη προ-εικόνα (second pre-image)]
Δίνεται συνάρτηση κατακερματισμού $h : \mathcal{X} \to \mathcal{Y}$ και $x \in \mathcal{X}$. Να υπολογισθεί $x' \in \mathcal{X}$ τέτοιο ώστε $x' \neq x$ και $h(x') = h(x)$. □

Πρόβλημα 3. [Σύγκρουση (collision)]
Δίνεται συνάρτηση κατακερματισμού $h : \mathcal{X} \to \mathcal{Y}$. Να υπολογισθεί $(x, x') \in \mathcal{X}^2$ τέτοιο ώστε $x' \neq x$ και $h(x') = h(x)$. □

Αντίστοιχα με το κατά πόσο επιδέχεται αποδοτική λύση καθένα από τα παραπάνω προβλήματα ταξινομούνται οι συναρτήσεις κατακερματισμού:

- Μια συνάρτηση h για την οποία δεν επιλύεται με αποδοτικό τρόπο το πρόβλημα προ-εικόνας λέγεται ανθεκτική σε προεικόνα (pre-image resistant).
- Μια συνάρτηση h για την οποία δεν επιλύεται με αποδοτικό τρόπο το πρόβλημα δεύτερης προ-εικόνας λέγεται ανθεκτική σε δεύτερη προεικόνα (second pre-image resistant).
- Μια συνάρτηση h για την οποία δεν επιλύεται με αποδοτικό τρόπο το πρόβλημα σύγκρουσης λέγεται ανθεκτική σε σύγκρουση (collision resistant).

Είναι φανερό ότι η συνάρτηση h του Παραδείγματος 19 δεν έχει καμμιά από τις τρεις αυτές ιδιότητες. Στη συνέχεια εξετάζουμε τον σχετικό βαθμό δυσκολίας των παραπάνω προβλημάτων.

Εάν υπάρχει αλγόριθμος A επίλυσης του προβλήματος προ-εικόνας, μπορεί να επιλυθεί και το πρόβλημα δεύτερης προ-εικόνας: Αρκεί να υπολογισθεί το $h(x') = y$ και στη συνέχεια να χρησιμοποιηθεί ο αλγόριθμος A για τον υπολογισμό του $h^{-1}(y)$. Κατά συνέπεια το πρόβλημα προ-εικόνας είναι όσο δύσκολο είναι το πρόβλημα δεύτερης προ-εικόνας ή και πιο δύσκολο (με την έννοια της υπολογιστικής πολυπλοκότητας).

Αν έχουμε έναν αλγόριθμο B που επιλύει το πρόβλημα δεύτερης προ-εικόνας, μπορούμε να επιλύσουμε το πρόβλημα σύγκρουσης: Επιλέγουμε ένα οποιοδήποτε x και υπολογίζουμε μέσω του B το x' για το οποίο $h(x') = h(x)$. Τότε το ζεύγος (x, x') είναι ζεύγος σύγκρουσης. Κατά συνέπεια το πρόβλημα δεύτερης προ-εικόνας είναι όσο δύσκολο είναι το πρόβλημα σύγκρουσης ή και πιο δύσκολο. Άρα το τελευταίο από τα τρία παραπάνω προβλήματα είναι το «πιο εύκολο» και το πρώτο είναι το «πιο δύσκολο».

Συναρτήσεις για τις οποίες είναι υπολογιστικά δύσκολη η επίλυση όλων των παραπάνω προβλημάτων λέγονται *κρυπτογραφικές συναρτήσεις κατακερματισμού* (cryptographic hash functions). Στην επόμενη ενότητα εκτίθεται η έννοια της ιδανικής συνάρτησης κατακερματισμού, που αποτελεί ένα μοντέλο για υπολογισμούς σχετικούς με κρυπτογραφικές συναρτήσεις.

Ιδανικές συναρτήσεις κατακερματισμού

Ένας τρόπος να γίνει μια εκτίμηση του κινδύνου επίλυσης των παραπάνω προβλημάτων αντιστροφής είναι να χρησιμοποιηθεί το λεγόμενο *τυχαίο μαντείο* (random oracle) [BR93; CGH04]. Σύμφωνα με αυτό το μοντέλο για κάθε x η αντίστοιχη τιμή $h(x)$ έχει δημιουργηθεί με τυχαίο τρόπο και έχει καταγραφεί σε ένα κατάλογο. Ο μόνος τρόπος να μάθει κανείς το $h(x)$ για ένα δοσμένο x είναι να ρωτήσει το μαντείο, δηλαδή να μάθει τι είναι γραμμένο στον κατάλογο.

Μια εναλλακτική διατύπωση για το τυχαίο μαντείο είναι ότι παίρνει ως είσοδο ένα string οποιουδήποτε μήκους και παράγει ένα string απείρου μήκους του οποίου τα bits είναι ομοιόμορφα κατανεμημένα και ανεξάρτητα [Ber+07]. Μια συνάρτηση κατακερματισμού με μήκους εξόδου n ιδανικά συμπεριφέρεται ως τυχαίο μαντείο με έξοδο κομμένη στα πρώτα n bits.

Αν το πλήθος των δυνατών τιμών του y είναι N (δηλαδή $y \in \mathcal{Y}$ και $|\mathcal{Y}| = N$) και δοθεί ένα x, για το οποίο δεν έχει ακόμη ερωτηθεί το μαντείο, η πιθανότητα να ισχύει $y = h(x)$ είναι ίση με $1/N$ (και είναι ανεξάρτητη από τις προηγούμενες απαντήσεις).

Έστω ότι κάποιος δοκιμάζει K τιμές του x μήπως λύνουν το *πρόβλημα της προεικόνας* εφόσον έχει δοθεί ένα επιθυμητό y. Δηλαδή σχηματίζει ένα σύνολο $\mathcal{X}_0$ με πληθικό αριθμό K και για όλα τα $x \in \mathcal{X}_0$ ρωτάει το μαντείο και μαθαίνει την τιμή $h(x)$, σχηματίζοντας έτσι το σύνολο απαντήσεων $\mathcal{Y}_0 = \{y | y = h(x) \wedge x \in \mathcal{X}_0\}$. Τότε

$$\Pr\{y \in \mathcal{Y}_0\} = 1 - \left(1 - \frac{1}{N}\right)^K$$

Παρόμοιο είναι και το αποτέλεσμα για το *πρόβλημα της δεύτερης προ-εικόνας*, δεδομένου ότι δοθέντος ενός x υπολογίζεται το αντίστοιχο $y = h(x)$ και κατόπιν επιλύεται το προηγούμενο πρόβλημα για το y.

Το πρόβλημα σύγκρουσης όμως επιλύεται ως εξής: Επιλέγεται σύνολο $\mathcal{X}_0$ με πληθικό αριθμό K. Στη συνέχεια υπολογίζεται για όλα τα $x \in \mathcal{X}_0$ το $h(x)$, οπότε και πάλι

σχηματίζεται το $\mathcal{Y}_0$ όπως πριν, ενώ κάθε στοιχείο που προστίθεται συγκρίνεται με όλα τα προηγούμενα. Στην περίπτωση αυτήν σύμφωνα με το *θεώρημα των γενεθλίων* η πιθανότητα επιτυχίας p είναι

$$p = 1 - \left(\frac{N-1}{N}\right)\left(\frac{N-2}{N}\right)\ldots\left(\frac{N-K+1}{N}\right) \tag{4.1}$$

Το θεώρημα των γενεθλίων δίνει την παραπάνω απάντηση στο εξής πρόβλημα:

Πρόβλημα 4. [Πρόβλημα των γενεθλίων]
Σε μια αίθουσα υπάρχουν άτομα. Δεδομένου ότι το έτος έχει N μέρες, ποια είναι η πιθανότητα p να έχουν γενέθλια την ίδια μέρα δύο τουλάχιστον άτομα; □

Λύση: Θα υπολογίσουμε την πιθανότητα $1-p$ να έχουν όλοι γενέθλια σε διαφορετικές μέρες. Επιλέγεται ένα άτομο από τα K. Κατόπιν επιλέγεται δεύτερο άτομο. Η πιθανότητα να έχει γενέθλια σε διαφορετική μέρα από το πρώτο είναι $(N-1)/N$. Επιλέγεται τρίτο άτομο και η πιθανότητα να μην έχει την ίδια μέρα γενέθλια με τα προηγούμενα δύο δεδομένου ότι αυτά δεν έχουν την ίδια μέρα είναι $(N-2)/N$. Επιλέγεται 4ο άτομο και η πιθ. να μην έχει γενέθλια την ίδια μέρα με τα άλλα τρία δεδομένου ότι αυτά ήδη έχουν γενέθλια σε διαφορετικές μέρες είναι $(N-3)/N$. Στο K-οστό άτομο η πιθανότητα αυτή είναι $(N-K+1)/N$. Η πιθ. $1-p$ να μην υπάρχει κοινή μέρα γενεθλίων είναι το γινόμενο όλων αυτών των υπό συνθήκη πιθανοτήτων. □

Δεδομένης της έκφρασης (4.1) πόσο μεγάλο πρέπει να είναι το πλήθος των ερωτήσεων K προς το μαντείο για να υπάρχει πιθανότητα επιτυχίας p μεγαλύτερη από $1/2$; Η επίλυση της (4.1) ως προς K δεν δίνει κλειστό τύπο, αλλά μπορεί κανείς να αυξήσει το K σταδιακά μέχρι να περάσει η πιθανότητα p την επιθυμητή τιμή $1/2$. Ενδεικτικά για $N = 365$ η πιθανότητα επιτυχίας γίνεται μεγαλύτερη από $1/2$ για $K = 23$ άτομα.

Στη συνέχεια θα χρησιμοποιήσουμε μια προσέγγιση. Αν $K \ll N$ το κλάσμα K/N είναι μικρό και μπορούμε να χρησιμοποιήσουμε την προσέγγιση $e^{-x} \simeq 1-x$, δηλαδή να θέσουμε

$$\frac{N-i}{N} = 1 - \frac{i}{N} = e^{-i/N}$$

οπότε

$$1 - p \simeq \prod_{i=1}^{K-1} e^{-i/N} = e^{-\sum_{i=1}^{K-1} i/N} = e^{-\frac{K(K-1)}{2N}}$$

Επιλύοντας ως προς $K(K-1)$

$$K(K-1) \simeq 2N \ln \frac{1}{1-p}$$

και αμελώντας τον όρο K (σε σύγκριση με το K^2)

$$K \simeq \left(2N \ln \frac{1}{1-p}\right)^{1/2}$$

Για $p = 1/2$ προκύπτει $K \simeq 1.177\sqrt{N}$. Για $N = 365$ ο προσεγγιστικός τύπος δίνει $K = 22.5$.

Αυτή η προσέγγιση δείχνει ότι η Trudy για να λύσει το πρόβλημα της σύγκρουσης χρειάζεται αριθμό δοκιμών της τάξης του $\sqrt{N}$ αντί του $N/2$ που χρειάζεται για τα δύο προβλήματα προ-εικόνας. Στην περίπτωση που η συνάρτηση κατακερματισμού h παράγει ένα bit string μήκους n, τα προβλήματα προ-εικόνας λύνονται με περίπου $2^n/2 = 2^{n-1}$ δοκιμές, ενώ το πρόβλημα της σύγκρουσης με περίπου $(2^n)^{1/2} = 2^{n/2}$ δοκιμές.

Συναρτήσεις με και χωρίς κλειδί

Εάν ο Bob και η Alice, ή και τρίτοι, ενδιαφέρονται να διαπιστώσουν ότι ένα κείμενο παραμένει εγγυημένα αναλλοίωτο αρκεί να χρησιμοποιήσουν μια «καλή» συνάρτηση κατακερματισμού και να βρουν ένα τρόπο να διατηρούν με ασφάλεια την τιμή κατακερματισμού. Από μια άποψη αυτό μοιάζει με μια απλή μετάθεση του προβλήματος, δηλαδή αντί να βρουν ένα τρόπο να μην αλλοιωθεί το κείμενο πρέπει να βρουν ένα τρόπο να μην αλλοιωθεί η αντίστοιχη τιμή κατακερματισμού. Σε μια άλλη διατύπωση το πρόβλημα της φύλαξης ενός μεγάλου μήκους string ανάγεται στο πρόβλημα φύλαξης ενός string με μικρότερο μήκος. Αν και αυτή η αναγωγή φαίνεται να αποτελεί μια απλή ποσοτική διαφορά, δεν πρέπει να υποτιμάται η αξία της. Αν όλοι οι νόμοι ενός κράτους είναι γραμμένοι σε μακροσκελή κείμενα, η τιμή κατακερματισμού που θα έδινε εγγύηση για τη μη αλλοίωση των κειμένων θα μπορούσε απλώς να χαραχτεί πάνω σ' ένα μάρμαρο στην είσοδο του ανώτατου δικαστηρίου. Είδαμε επίσης ότι η σχετικά αργή μέθοδος Diffie-Hellman επιτρέπει σε δυο πλευρές να ανταλλάξουν ένα μικρό σχετικά string και να το χρησιμοποιήσουν περαιτέρω ως κλειδί για μια πιο γρήγορη συμμετρική κρυπτογράφηση που θα επιτρέψει την ασφαλή ανταλλαγή μεγαλύτερων κειμένων. Σε ένα άλλο παράδειγμα στην περιοχή των κρυπτονομισμάτων θα δούμε ότι μια τιμή κατακερματισμού μπορεί να τυπωθεί με ανεξίτηλο τρόπο και να εγγυάται τα δικαιώματα κάποιου πάνω σε μια σειρά από συναλλαγές. Με δυο λόγια υπάρχουν εφαρμογές, στις οποίες η ακεραιότητα δεδομένων μπορεί να στηριχθεί στην ασφαλή φύλαξη μιας τιμής κατακερματισμού και αντίστοιχα κινδυνεύει στο βαθμό που η φύλαξη μπορεί να υποστεί επιθέσεις.

Το ίδιο συμπέρασμα θα μπορούσε να διατυπωθεί και για περιπτώσεις όπου μεσολαβεί επικοινωνία, εφόσον διαφυλαχθεί η ακεραιότητα της επικοινωνίας. Ο Bob μπορεί να στείλει στην Alice ένα μακρύ κείμενο και να είναι αμφότεροι βέβαιοι ότι δεν έχει αλλοιωθεί αν ο Bob έχει έναν ασφαλή τρόπο να της στείλει την τιμή κατακερματισμού. Σε κάθε περίπτωση όμως η επικοινωνία αυξάνει το βαθμό δυσκολίας του προβλήματος.

Εναλλακτικά ο Bob και η Alice θα μπορούσαν να ανταλλάσσουν μηνύματα με εγγυημένη αυθεντικότητα (προέλευση) και ακεραιότητα αν χρησιμοποιούσαν μια συνάρτηση κατακερματισμού που την ξέρουν μόνο αυτοί. Τότε η Trudy δεν θα μπορούσε να αλλοιώσει το κείμενο επειδή δεν θα ήξερε πώς να αλλάξει αντίστοιχα την τιμή κατακερματισμού. Όπως όμως είναι σκόπιμο δυο πλευρές σε επικοινωνία να μη χρησιμοποιούν ένα μυστικό αλγόριθμο κρυπτογράφησης, ομοίως είναι σκόπιμο να μη χρησιμοποιούν μια μυστική συνάρτηση κατακερματισμού, επειδή και στις δύο περιπτώσεις ο αλγόριθμος μπορεί να διαρρεύσει. Κατά συνέπεια γίνεται γενικά η παραδοχή ότι οι συναρτήσεις κατακερματισμού είναι γενικά δημοσιοποιημένες (και συχνά τυποποιημένες). Αυτό που μπορεί να γίνει είναι να χρησιμοποιηθεί μια συνάρτηση κατακερματισμού που εξαρτάται από μια πρόσθετη παράμετρο, έστω K, και να χρησιμοποιηθεί το K όπως ένα κλειδί στην κρυπτογραφία.

Και πάλι το κλειδί πρέπει να ανταλλαγεί με μυστικό και ασφαλή τρόπο, αλλά η διαφορά σε σχέση με την ανταλλαγή μιας τιμής κατακερματισμού είναι ότι το κλειδί μπορεί να ανταλλαγεί μόνο μια φορά και εκ των προτέρων. Στη συνέχεια το ίδιο κλειδί θα μπορεί να χρησιμοποιείται για να είναι εγγυημένη η ακεραιότητα όσων κειμένων επιθυμούν να ανταλλάξουν οι δύο πλευρές. Μια συνάρτηση κατακερματισμού με κλειδί K θα την συμβολίζουμε με $h_K(x)$. Συνήθως όταν η τιμή κατακερματισμού προέρχεται από μια συνάρτηση με κλειδί ονομάζεται κωδικός κατακερματισμού (hash code).

Αν πάλι ο Bob θέλει να χρησιμοποιήσει μια απλή συνάρτηση κατακερματισμού (δηλαδή μια συνάρτηση που δεν έχει κλειδί σύμφωνα με δημόσια -ή και τυποποιημένη-

περιγραφή της) μαζί με ένα κλειδί K που έχει ανταλλάξει με την Alice, μπορεί να χρησιμοποιήσει την απλή λύση να στείλει μαζί με το κείμενο x την τιμή κατακερματισμού $h(K\|x)$.[3] Μια συνήθης πρακτική είναι η αποστολή του $h(K\|x\|K)$, δηλαδή να προστίθεται το κλειδί στην αρχή και στο τέλος του κειμένου, αλλά θα δούμε ότι υπάρχουν και πιο «αξιοπρεπείς» τρόποι κατασκευής μιας συνάρτησης κατακερματισμού με κλειδί χρησιμοποιώντας ως βάση μια απλή συνάρτηση.

Γιατί να θέλει ο Bob να χρησιμοποιήσει κάτι από τα παραπάνω; Οι αλγόριθμοι κατακερματισμού, όπως επίσης και οι αλγόριθμοι κρυπτογράφησης, συχνά ενσωματώνονται σε συστήματα, εφαρμογές, πρωτόκολλα κ.λπ. και προσφέρονται με πληρωμή δικαιωμάτων. Ίσως ο Bob ή αυτός που ετοιμάζει ένα σύστημα να θέλει να αποφύγει μια τέτοια πληρωμή. Ίσως να έχει υλοποιήσει ήδη μια απλή συνάρτηση κατακερματισμού και να θέλει να επαναχρησιμοποιήσει τον κώδικα. Ίσως να μη θέλει να επιβαρύνει μια συσκευή με περιορισμένες δυνατότητες με τους υπολογισμούς μιας προχωρημένης συνάρτησης κατακερματισμού. Υπάρχει επομένως μια σειρά από λόγους για την παράλληλη ανάπτυξη και χρήση λύσεων που εκ πρώτης όψεως φαίνονται να επικαλύπτουν η μια την άλλη.

Όπως θα δούμε και στη συνέχεια υπάρχει μια σειρά από λύσεις στο οπλοστάσιο της ασφάλειας, άλλες εκ των οποίων ξεκινούν από σχετικά απλές συναρτήσεις κατακερματισμού πάνω στις οποίες χτίζονται πιο πολύπλοκες και άλλες ξεκινούν την άλλη άκρη της πολυπλοκότητας, δηλαδή από σύνθετους κρυπτογραφικούς κώδικες για να φτάσουν σε ένα κωδικό κατακερματισμού μέσω απλοποίησης.

Εφαρμογή σε κείμενα απεριόριστου μήκους

Οι συνήθεις συναρτήσεις κατακερματισμού παίρνουν ως είσοδο ένα block, δηλαδή ένα κείμενο με σταθερό μήκος m, και παράγουν μια «σύνοψη» (μια τιμή κατακερματισμού) με επίσης σταθερό μήκος n, εν γένει μικρότερο από το μήκος του κειμένου. Αν το κείμενο έχει μήκος μικρότερο από m η προφανής λύση είναι να συμπληρωθεί με κενά. Αν είναι bit string μπορεί να συμπληρωθεί με μηδενικά. Ωστόσο για να διευκολύνεται λιγότερο ο πιθανός αντίπαλος είναι σκόπιμο να συμπληρώνεται με κάτι λιγότερο προφανές, π.χ. με μια ένδειξη μήκους, ίσως ακόμη με μια χρονική ένδειξη και με κάποιο αυθαίρετο ή τυχαίο string.

Αν το κείμενο έχει μήκος μεγαλύτερο από n η επίσης προφανής λύση είναι ο τεμαχισμός σε κομμάτια μήκους n. Επειδή όμως αυτό θα μπορούσε να το εκμεταλλευθεί αμέσως ο αντίπαλος κάνοντας μια επίθεση αναδιάταξης, η παρακάτω μέθοδος Merkle–Damgård συνδέει την τιμή κατακερματισμού για κάθε κομμάτι (block) με το περιεχόμενο των προηγούμενων κομματιών. Η μέθοδος αυτή χρησιμοποιείται σε δημοφιλείς αλγόριθμους κατακερματισμού, όπως οι MD5, SHA-1 και SHA-2.

Το σχήμα Merkle–Damgård

Ο Ralph Charles Merkle (γενν. το 1952 στο Berkeley, California) έκανε τη διδακτορική διατριβή του [Mer79] στο Stanford από το 1974 ως το 1979. Περιέγραψε μια μέθοδο κατασκευής μιας κρυπτογραφικής συνάρτησης κατακερματισμού h', δηλαδή έναν αλγόριθμο που διατηρεί την ιδιότητα αντίστασης στη σύγκρουση όταν δοθεί μια συνάρτηση συμπίεσης h (για είσοδο σταθερού μήκους και έξοδο σταθερού μήκους) που έχει αυτήν την ιδιότητα. Ο Ivan Bjerre Damgård είναι ένας Δανός μαθηματικός (Master και PhD από το παν. του Aarhus), τώρα καθηγητής στο παν. του Aarhus, και

[3] Όπως και αλλού στο κείμενο ο συμβολισμός $a\|b$ υποδηλώνει την ένωση των string a, b.

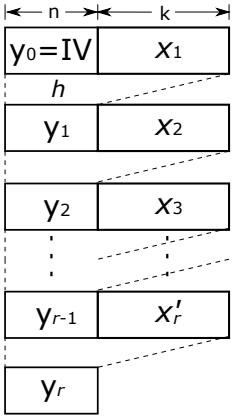
επινόησε τον ίδιο αλγόριθμο ανεξάρτητα από τον Merkle. Το πλεονέκτημα του σχήματος Merkle–Damgård είναι ότι η h' έχει αποδεδειγμένη αντίσταση σε σύγκρουση αν αυτό ισχύει για την συνάρτηση h.

Έστω ότι η κρυπτογραφική συνάρτηση h' παράγει ένα string y μήκους n από ένα string εισόδου x μήκους m.Έστω επίσης ότι η h δέχεται είσοδο μήκους $n+k$, και δίνει έξοδο μήκους n.

Διαιρούμε το x σε τμήματα μήκους k, δηλαδή

$$x = x_1 \| x_2 \| \ldots \| x_{r-1} \| x_r, \quad r = \left\lceil \frac{m}{k} \right\rceil$$

όπου $|x_1| = |x_2| = \ldots = |x_{r-1}| = k$, αλλά εφόσον εν γένει το m δεν θα είναι ακριβώς πολλαπλάσιο του k το τελευταίο θα έχει μήκος $|x_r| = k - d$ $(0 \leq d \leq k-1)$. Από το x κατασκευάζουμε το x' που είναι μήκους ακριβώς rk πρόσθέτοντας στο τέλος d μηδενικά. Δηλαδή το x' αποτελείται από strings $x'_i = x_i$ για $i = 1, \ldots, r-1$ και καθένα έχει μήκος k, ενώ το τελευταίο $x'_r = x_r \| 0_d$, όπου 0_d είναι ένα string αποτελούμενο από d μηδενικά και έχει επίσης μήκος k.



Σχήμα 4.1: Το σχήμα Merkle–Damgård.

Δίνουμε μια προκαθορισμένη αρχική τιμή (IV) στο string y_0 μήκους n και υπολογίζουμε το

$$y_1 = h(y_0 \| x_1)$$

και στη συνέχεια

$$y_i = h(y_{i-1} \| x_i), \quad i = 2, \ldots, r-1$$

και τέλος

$$y_r = h(y_{r-1} \| x'_r).$$

Η έξοδος της συνολικής συνάρτησης $h'(x)$ είναι το $y = y_r$.

4.4 Τυποποιημένες συναρτήσεις

Στη δεκαετία του '90 ο Rivest (της ομάδας RSA) πρότεινε μια σειρά αλγορίθμων που υπολογίζουν μια σύνοψη κατάλληλη για εφαρμογές ψηφιακής υπογραφής. Τέτοιοι αλγόριθμοι είναι οι MD2, MD4 και MD5 που παράγουν σύνοψη μήκους 128 bits και περιγράφονται στις RFC 1319 ως 1321.

Το 1990 ο Rivest ανέπτυξε τον αλγόριθμο MD4 βασισμένο στο σχήμα Damgård-Merkle και στη συνέχεια τον MD5, που αποτελεί βελτίωση του MD4.

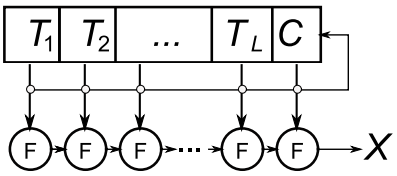
MD2 Message Digest Algorithm

Η RFC 1319 [Kal92] του 1992 περιγράφει τον αλγόριθμο MD2 με τον οποίο παράγεται μια σύνοψη των 128 bits όταν δοθεί ένα κείμενο αυθαιρέτου μήκους. Ο αλγόριθμος αναπτύχθηκε από το Rivest το 1989.

Η γενική ιδέα είναι όπως στο Σχ. 4.2 [Mul04]. Η επεξεργασία του κειμένου γίνεται σε blocks των 16 bytes (128 bits). Το αρχικό κείμενο επεκτείνεται ώστε να αποτελείται από ακέραιο πλήθος blocks, δηλαδή να μπορεί να παρασταθεί ως

$$T = T_1 \| T_2 \| \dots \| T_L$$

και συμπληρώνεται με ένα block C από bits ελέγχου (checkbits). Ακολουθεί η κύρια διαδικασία κατακερματισμού, που γίνεται από την συνάρτηση F στο σχήμα, ενώ από block σε block μεταφέρεται ένα string X μήκους 48 bytes και η τελική του τιμή εξάγεται ως τιμή κατακερματισμού του T.



Σχήμα 4.2: Γενικό σχήμα του MD2.

Η είσοδος ας υποτεθεί ότι είναι μήκους b bytes. Το εισερχόμενο κείμενο μπορεί να γραφεί ως $m_0 \| m_1 \| \dots \| m_{b-1}$, όπου m_i είναι το $(i+1)$-οστό byte του κειμένου.

1. Το κείμενο συμπληρώνεται ώστε να αποκτήσει μήκος

$$N = \begin{cases} 16 \lceil b/16 \rceil & \text{αν } b \bmod 16 > 0 \\ b + 16 & \text{αν } b \bmod 16 = 0 \end{cases}$$

   δηλαδή επεκτείνεται ώστε το μήκος του να είναι πολλαπλάσιο των 16 bytes. Στην περίπτωση που αυτό ήδη συμβαίνει προστίθενται ακέραια 16 bytes.

   Η προσθήκη γίνεται προσθέτοντας i bytes με την τιμή i. Το επεκτεταμένο κείμενο είναι της μορφής

$$M_0 \| M_1 \| \dots \| M_{N-1}$$

   όπου M_i είναι το $(i+1)$-οστό byte και το N είναι πολλαπλάσιο του 16, δηλαδή σε σχέση με το Σχ. 4.2 είναι $N = 16L$ και $T_1 = M_0 \| M_1 \| \dots \| M_{15}$, $T_2 = M_{16} \| M_{17} \| \dots \| M_{31}$ κ.ο.κ.

2. Προστίθεται ακόμη μια 16-άδα από bytes $C_0 \| C_1 \| \dots \| C_{15}$ που σχηματίζονται ως άθροισμα ελέγχου (checksum) του ως τώρα κειμένου. Χρησιμοποιείται για τον σκοπό αυτόν ένα διάνυσμα S με προκαθορισμένες τιμές από το σύνολο $\{0, 1, 2, \dots, 255\}$, με συνιστώσες $S(i)$ $(i = 0, 1, 2, \dots, 255)$ που αποτελούν μετάθεση των παραπάνω τιμών κατασκευασμένη με οδηγό τα ψηφία του π. Ο πίνακας περιλαμβάνεται σε παράρτημα της σύστασης.

Τα 16 bytes ελέγχου σχηματίζονται από κάθε block χρησιμοποιώντας και τις τιμές που έχουν σχηματισθεί από το προηγούμενο block. Για το πρώτο block:

$$\begin{aligned} C_0 &:= S(M_0) \\ C_1 &:= S(M_1 \oplus C_0) \\ &\vdots \\ C_{15} &:= S(M_{15} \oplus C_{14}) \end{aligned}$$

Για το δεύτερο block:

$$\begin{aligned} C_0 &:= S(M_{16} \oplus C_{15}) \\ C_1 &:= S(M_{17} \oplus C_0) \\ &\vdots \\ C_{15} &:= S(M_{31} \oplus C_{14}) \end{aligned}$$

Δεδομένου ότι το N είναι πολλαπλάσιο του 16 ο τελευταίος υπολογισμός είναι

$$C_{15} := S(M_N \oplus C_{14})$$

Οι υπολογισμοί αυτοί φαίνονται και στο Σχ. 4.3. Η έξοδος αυτού του υπολογισμού είναι προφανώς τα τελευταία 16 bytes $C_0 \| C_1 \| \ldots \| C_{15}$. Αυτά προστίθενται στο αρχικό μήνυμα, το οποίο τώρα παίρνει τη μορφή

$$M_0 \| M_1 \| \ldots \| M_N \| C_0 \| \ldots \| C_{15} = M_0 \| M_1 \| \ldots \| M_{N'}$$

όπου $N' = N + 16$.

3. Το βήμα αυτό χρησιμοποιεί ένα καταχωρητή των 48 bytes. Τα bytes του καταχωρητή παριστάνονται ως $X_0, X_1, \ldots, X_{47}$ και όλα τίθενται αρχικά ίσα με μηδέν. Η επεξεργασία του κειμένου $M_0 \| \ldots \| M_{N'}$ γίνεται ανά block των 16 bytes.

   Το πρώτο block εισέρχεται στον υπολογισμό ως εξής:

   $$X_{16+j} := M_j, \quad X_{32+j} := X_{16+j} \oplus X_j, \quad \text{για } j = 0, 1, \ldots, 15$$

   Αρχικοποιείται μια πρόσθετη μεταβλητή t στο μηδέν και γίνονται 18 γύροι που ο καθένας έχει ως εξής:

   $$X_k := X_k \oplus S(t), \quad t := X_k \oplus S(t), \quad \text{για } k = 0, 1, \ldots, 47$$

   Στο τέλος του j γύρου ($j = 0, 1, \ldots 17$) τίθεται $t := t + j \bmod 256$.

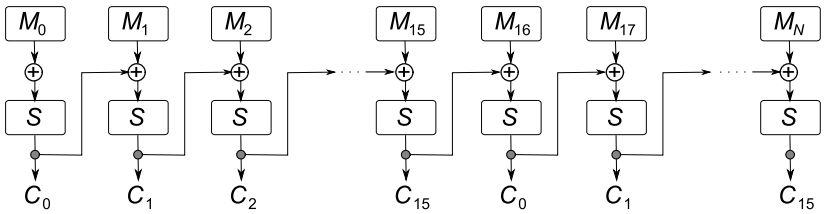
   Με τις τιμές των $X_0, \ldots, X_{47}$ που έχουν υπολογισθεί επαναλαμβάνεται ο προηγούμενος υπολογισμός για το επόμενο block, δηλαδή με το byte M_{16} στη θέση του M_0, το M_{17} στη θέση του M_1 κ.ο.κ. ως το M_{31}.

   Στη συνέχεια γίνεται το ίδιο με όλες τις υπόλοιπες 16-δες ως το N'.

4. Ως τελική σύνοψη του αρχικού μηνύματος εξάγονται τα 16 bytes

   $$X_0, \ldots, X_{15}$$

Επιθέσεις στη βασική συνάρτηση έχουν δημοσιευθεί από τους Rogier και Chauvaud [RC95; RC97] το 1995 και το 1997. Στο άρθρο του Frédéric Muller του 2004 [Mul04] περιγράφονται επιθέσεις προ-εικόνας ενάντια στον MD2 με πολυπλοκότητα 2^{104} αντί του 2^{128}. Επίσης μπορεί κανείς να βρει στο ίδιο άρθρο μια πιο αναλυτική περιγραφή του παραπάνω βήματος 3.



Σχήμα 4.3: Υπολογισμός των $C_0\|C_1\|\ldots\|C_{15}$ στο MD2.

MD4 Message Digest Algorithm

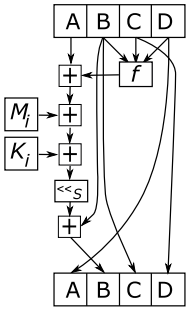
Το μήνυμα προσαυξάνεται έτσι ώστε μαζί με 448 bits να είναι πολλαπλάσιο των 512 bits. Ο αλγόριθμος επεξεργάζεται το μήνυμα σε τμήματα των 512 bits σύμφωνα με το σχήμα Damgård-Merkle και το περνάει από τρεις γύρους. Η έξοδος έχει μήκος 128 bits. Ωστόσο βρέθηκαν αδυναμίες πολύ γρήγορα, π.χ. ο Dobbertin το 1998 δημοσίευσε [Dob98] μια μέθοδο εύρεσης συγκρούσεων μέσα σε δευτερόλεπτα από ένα PC της εποχής. Κατά συνέπεια ο MD4 θεωρήθηκε «σπασμένος» πολύ γρήγορα.

MD5 Message Digest Algorithm

Ο MD5 αποτελεί βελτίωση του ήδη πολύ ασθενούς MD4. Ο MD5 έχει 4 γύρους (ανά block, βλ. και πιο κάτω) αντί τριών του MD4, χρησιμοποιεί «τυχαιοποιημένες» ομάδες από σταθερές και σε κάθε βήμα προσθέτει το αποτέλεσμα του προηγούμενου βήματος.

Τα του μεγέθους του μηνύματος, του τμήματος και του μήκους εξόδου είναι όπως στον MD4 και χρησιμοποιείται το σχήμα Damgård-Merkle για την δημιουργία της τελικής εξόδου από την έξοδο κάθε τμήματος.

Η επαύξηση για να γίνει το μήκος του πολλαπλάσιο των 512 bits γίνεται ως εξής: Προστίθεται αρχικά το bit 1 στο τέλος του μηνύματος και κατόπιν μηδενικά μέχρι το μήκος να φτάσει να είναι $k \times 512 - 64$. Τέλος, τα 64 bits γεμίζουν με έναν αριθμό που παριστάνει το μήκος του αρχικού μηνύματος.



Σχήμα 4.4: Ένα μέρος του υπολογισμού του MD5.

Ο αλγόριθμος διατηρεί και επεξεργάζεται συνεχώς ένα string από 128 bits χωρισμένο σε 4 ομάδες των 32 bits (δηλ. είναι κατασκευασμένος για 32-μπιτους επεξεργαστές). Η επεξεργασία ενός block γίνεται σε 4 γύρους παρόμοιας λογικής. Ας υποτεθεί ότι τα 4 strings ονομάζονται A, B, C, D. Δεδομένου ότι ένα block των 512 bits μπορεί

να διαιρεθεί σε 16 ομάδες των 32 bits, οι οποίες παριστάνονται ως M_i ($i = 1, \ldots, 16$), η επεξεργασία ενός block γίνεται σε 16 υπο-γύρους, καθένας από τους οποίους εμπλέκει διαδοχικά ένα M_i και μια σταθερά (των 32 bits) K_i.

Στο Σχ. 4.4 φαίνεται με ποιο τρόπο ένα από τα M_i επιδρά πάνω στο $A\|B\|C\|D$ μέσα σε ένα υπο-γύρο. Τα σημεία πρόσθεσης στο σχήμα υποδηλώνουν πρόσθεση των αντίστοιχων 32-μπιτων δυαδικών αριθμών, δηλαδή πρόσθεση $\bmod 2^{32}$. S είναι μια αριστερή κυκλική περιστροφή κατά S bits, όπου το S μεταβάλλεται. Στον πρώτο γύρο η συνάρτηση f είναι

$$f_1(B, C, D) = (B \text{ AND } C) \text{ OR } (\text{ NOT}(B) \text{ AND } D)$$

ή με συμβολική μορφή των λογικών πράξεων

$$f_1(B, C, D) = (B \wedge C) \vee (\neg B \wedge D)$$

(όπου η παραπάνω πράξη γίνεται ανά bit). Στους υπόλοιπους τρεις γύρους η f έχει ως εξής:

$$\begin{aligned} f_2(B, C, D) &= (B \wedge C) \vee (B \wedge \neg D) \\ f_3(B, C, D) &= B \oplus C \oplus D \\ f_4(B, C, D) &= C \oplus (B \vee \neg D) \end{aligned}$$

όπου το $\oplus$ είναι ως συνήθως το XOR (ενώ το $+$ στο σχήμα είναι πρόσθεση των 32-μπιτων ακεραίων $\bmod 2^{32}$).

Στο τέλος του 4ου γύρου έχει σχηματισθεί το string $A\|B\|C\|D$ που έχει λάβει υπόψη το συγκεκριμένο block των 512 bits και τροφοδοτείται ως αρχική τιμή του $A\|B\|C\|D$ για το επόμενο block. Για το πρώτο block έχει μια συγκεκριμένη αρχική τιμή. Όταν γίνει η επεξεργασία και του τελευταίου block το $A\|B\|C\|D$ αποτελεί την τιμή εξόδου του MD5.

Ως προς τις τιμές των S και K_i ισχύουν τα εξής: Ένα block χρειάζεται $4 \times 16 = 64$ υπο-γύρους όπως στο σχήμα. Δίνονται επομένως 64 τιμές ολίσθησης για το S

$$\begin{aligned} &7, 12, 17, 22, 7, 12, 17, 22, 7, 12, 17, 22, 7, 12, 17, 22 \\ &5, 9, 14, 20, 5, 9, 14, 20, 5, 9, 14, 20, 5, 9, 14, 20 \\ &4, 11, 16, 23, 4, 11, 16, 23, 4, 11, 16, 23, 4, 11, 16, 23 \\ &6, 10, 15, 21, 6, 10, 15, 21, 6, 10, 15, 21, 6, 10, 15, 21 \end{aligned}$$

και 64 τιμές για τα K_i σε 16δική μορφή (όπου τα πρώτα 4 είναι στην πρώτη γραμμή):

$d76aa478$	$e8c7b756$	$242070db$	$c1bdceee$
$f57c0faf$	$4787c62a$	$a8304613$	$fd469501$
$698098d8$	$8b44f7af$	$ffff5bb1$	$895cd7be$
$6b901122$	$fd987193$	$a679438e$	$49b40821$
$f61e2562$	$c040b340$	$265e5a51$	$e9b6c7aa$
$d62f105d$	02441453	$d8a1e681$	$e7d3fbc8$
$21e1cde6$	$c33707d6$	$f4d50d87$	$455a14ed$
$a9e3e905$	$fcefa3f8$	$676f02d9$	$8d2a4c8a$
$fffa3942$	$8771f681$	$6d9d6122$	$fde5380c$
$a4beea44$	$4bdecfa9$	$f6bb4b60$	$bebfbc70$
$289b7ec6$	$eaa127fa$	$d4ef3085$	$04881d05$
$d9d4d039$	$e6db99e5$	$1fa27cf8$	$c4ac5665$
$f4292244$	$432aff97$	$ab9423a7$	$fc93a039$
$655b59c3$	$8f0ccc92$	$ffeff47d$	$85845dd1$
$6fa87e4f$	$fe2ce6e0$	$a3014314$	$4e0811a1$
$f7537e82$	$bd3af235$	$2ad7d2bb$	$eb86d391$

Επίσης δίνονται οι εξής αρχικές τιμές των A, B, C, D σε 16δική μορφή:

$$\begin{aligned} A &= 67452301 \\ B &= efcdab89 \\ C &= 98badcfe \\ D &= 10325476 \end{aligned}$$

Ο αλγόριθμος MD5 έχει χρησιμοποιηθεί για την παραγωγή ψηφιακών πιστοποιητικών κατά X.509. Ωστόσο η πρόοδος στην ανίχνευση συγκρούσεων έκανε δυνατή την δημιουργία πλαστών πιστοποιητικών [SLD07; Sot+08]. Στην διάσημη περίπτωση του κακόβουλου λογισμικού *Flame* [Ben+12] χρησιμοποιήθηκε ένα πλαστό πιστοποιητικό γνησιότητας λογισμικού της Microsoft. Ο MD5 θεωρείται πλέον παρωχημένος. Προς το τέλος του 2008 ο Rivest ανέπτυξε τον αλγόριθμο MD6 [Riv+08] που έχει ιδιότητες παραλληλισμού του υπολογισμού για να μπορεί να προσφέρει κατακερματισμό σε ρεύμα δεδομένων μεγάλης ταχύτητας (περισσότερο από 1 GBps με 16 παράλληλες CPU το έτος 2008). Ο MD6 προοριζόταν ως υποψήφιος για τον διαγωνισμό του SHA-3, αλλά δεν πέρασε στο δεύτερο γύρο.

SHA-1

Ο SHA-1 (Secure Hash Algorithm -1) περιγράφεται στην σύσταση FIPS[4] 180-1 [Sta95] του NIST με τον τίτλο *Secure Hash Standard* (SHS). Ο SHA αναπτύχθηκε στα πλαίσια ενός διαρκούς κρατικού προγράμματος των ΗΠΑ για την ανάπτυξη κρυπτογραφικών αλγορίθμων, του προγράμματος Capstone που διαχειριζόταν η NSA και το NIST και έχει πλέον εγκαταλειφθεί. Αρχικά εμφανίσθηκε το *Secure Hash Standard* μέσω της σύστασης FIPS PUB 180 του 1993 και αναθεωρήθηκε μέσω της FIPS PUB 180-1 το 1995. Ο αλγόριθμος της FIPS PUB 180 είναι γνωστός ως SHA-0. Οι δύο αλγόριθμοι διαφέρουν ελάχιστα, κατά μία κυκλική μετάθεση.

Ο SHA-1 έχει χρησιμοποιηθεί ως βάση των πρωτοκόλλων TLS, SSL, PGP, SSH, S/MIME και IPsec, καθώς και του Digital Signature Standard (DSA). Ωστόσο ήδη από

[4] U.S. Federal Information Processing Standard

το 2010 το αμερικανικό κράτος συνιστά τη χρήση του SHA-2, όπως παρωχημένο είναι και το DSA.

Οι γνωστότεροι φυλλομετρητές (web browsers) έπαψαν να χρησιμοποιούν τον SHA-1 από τις αρχές του 2017. Το 2017 δημοσιεύθηκε επίθεση για τον πλήρη SHA-1 [Ste+17] με $2^{63.1}$. Η επίθεση αυτή είναι πάνω από 5 τάξεις μεγέθους ταχύτερη από μια επίθεση με brute force. Ωστόσο ο SHA-1 χρησιμοποιείται ακόμη και θεωρείται ασφαλής ως μέρος του HMAC (keyed-hash message authentication code).

Ο αλγόριθμος είναι της ίδιας γενικής λογικής με τον MD5, αλλά με ορισμένες βελτιώσεις, όπως περισσότερους γύρους, περισσότερες σταθερές και μακρύτερη σύνοψη (των 160 bits αντί των 128).

Επαύξηση - padding

Η επαύξηση για να γίνει το μήκος του πολλαπλάσιο των 512 bits γίνεται ως εξής: Προστίθεται αρχικά το bit 1 στο τέλος του μηνύματος και κατόπιν μηδενικά μέχρι το μήκος να φτάσει να είναι $k \times 512 - 64$. Τέλος, τα 64 bits γεμίζουν με έναν αριθμό που παριστάνει το μήκος του αρχικού μηνύματος (πριν μπει το bit 1).

Παράδειγμα 20. Το παράδειγμα αυτό προέρχεται από τη σύσταση. Έστω το αρχικό μήνυμα[5]

$$01100001 \quad 01100010 \quad 01100011 \quad 01100100 \quad 01100101$$

Το μήκος του είναι προφανώς $\ell = 5 \times 8 = 40$ και το ίδιο string σε δεκαεξαδική μορφή είναι

$$61 \quad 62 \quad 63 \quad 64 \quad 65$$

Αρχικά προστίθεται το 1:

$$01100001 \quad 01100010 \quad 01100011 \quad 01100101 \quad 1$$

Το νέο μήκος είναι 41, άρα πρέπει να συμπληρωθεί με $512 - 41 - 64 = 407$ μηδενικά (και με 64 bits του μήκους ℓ. Τα πρώτα 3 μηδενικά μετά το 1 σχηματίζουν το δεκαεξαδικό 8, επομένως το μήνυμα πριν την προσθήκη των τελευταίων 64 bits με το μήκος είναι στο δεκαεξαδικό:

$$\begin{array}{cccc} 61626364 & 65800000 & 00000000 & 00000000 \\ 00000000 & 00000000 & 00000000 & 00000000 \\ 00000000 & 00000000 & 00000000 & 00000000 \\ 00000000 & 00000000 & & \end{array}$$

Το μήκος 40 στο δυαδικό είναι 101000 και προηγούνται άλλα $64-6 = 58$ μηδενικά. Το 101000 μαζί με τα προηγούμενά του δύο μηδενικά σχηματίζει τα δύο δεκαεξαδικά ψηφία 28, άρα μέχρι τώρα δεκαεξαδικός επεκτείνεται με το string $00000000\ 00000028$ και το τελικό string μετά την επέκταση είναι:

$$\begin{array}{cccc} 61626364 & 65800000 & 00000000 & 00000000 \\ 00000000 & 00000000 & 00000000 & 00000000 \\ 00000000 & 00000000 & 00000000 & 00000000 \\ 00000000 & 00000000 & 00000000 & 00000028 \end{array}$$

Με αυτόν τον τρόπο το string έχει τώρα μήκος $16 \times 8 \times 4 = 512$ bits. □

[5] Εννοείται ότι τα κενά είναι μόνο για τη διευκόλυνση του αναγνώστη.

Συναρτήσεις f, σταθερές, αρχικές τιμές

Οι πράξεις στον SHA-1 γίνονται σε γύρους με την ίδια γενική λογική όπως στον MD5, αλλά πάνω σε ένα συνεχώς μεταβαλλόμενο string μήκους 160 bits, το οποίο χωρίζεται σε 5 λέξεις των 32 bits που ονομάζονται A, B, C, D, E. Πριν περιγράψουμε αυτήν την επεξεργασία δίνουμε μια σειρά από συναρτήσεις και σταθερές που εμπλέκονται σ᾽ αυτήν.

Ο SHA-1 χρησιμοποιεί μια σειρά 80 συναρτήσεων f_i ($i = 0, \ldots, 79$) ως εξής:

$$\begin{aligned} f_t(B,C,D) &= (B \wedge C) \vee (\neg B \wedge D) \quad (0 \le t \le 19) \\ f_t(B,C,D) &= B \oplus C \oplus D \quad (20 \le t \le 39, 60 \le t \le 79) \\ f_t(B,C,D) &= (B \wedge C) \vee (B \wedge D) \vee (C \wedge D) \quad (50 \le t \le 59). \end{aligned}$$

Επίσης χρειάζονται 80 σταθερές των 32 bits, που διαφοροποιούνται ανά εικοσάδα, δηλαδή ορίζονται ως εξής:

$$\begin{aligned} K_t &= \text{5A827999} \quad (0 \le t \le 19) \\ K_t &= \text{6ED9EBA1} \quad (20 \le t \le 39) \\ K_t &= \text{8F1BBCDC} \quad (40 \le t \le 59) \\ K_t &= \text{CA62C1D6} \quad (60 \le t \le 79) \end{aligned}$$

Οι 5 λέξεις των 32 bits A, B, C, D, E χρειάζονται 5 σταθερές αρχικοποίησης αντίστοιχου μήκους που είναι οι εξής:

$$\begin{aligned} H_0 &= \text{67452301} \\ H_1 &= \text{EFCDAB89} \\ H_2 &= \text{98BADCFE} \\ H_3 &= \text{10325476} \\ H_5 &= \text{C3D2E1F0} \end{aligned} \tag{4.2}$$

Όλα τα παραπάνω αποτελούν το σταθερό «σώμα» του αλγόριθμου.

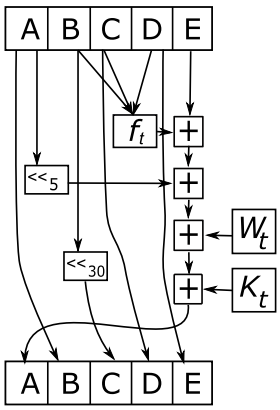
Επεξεργασία ενός block

Πρώτα δίνουμε την επεξεργασία της πρώτης 512-άδας των bits του μηνύματος, δηλαδή του block M_0.

Αρχικά το block διαιρείται σε 16 (32-μπιτες) λέξεις $W_0, \ldots, W_{15}$ (όπου W_0 είναι η πρώτη από αριστερά) και σχηματίζονται άλλες 64 λέξεις θέτοντας για $t = 16, \ldots, 79$

$$W_t = S^1(W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16})$$

όπου S^i είναι αριστερή ολίσθηση κατά i bits. Με τον τρόπο αυτόν έχουν σχηματισθεί από το αρχικό block M_0 (μήκους 512 bits) 80 λέξεις των 32 bits (συνολικού μήκους 2560 bits, δηλαδή πενταπλάσιου μήκους). Κάθε μια από αυτές μπαίνει διαδοχικά σε ένα σχήμα 80 υπο-γύρων, όπως αυτό που φαίνεται στο Σχ. 4.5. Στο σχήμα τα δύο κουτιά με τις αριθμητικές ενδείξεις 5 και 30 υποδηλώνουν αριστερή κυκλική μετάθεση κατά 5 και 30 bits αντίστοιχα. Δεξιά κάτω από τη λέξη E γίνονται διαδοχικές προσθέσεις ακεραίων $\bmod 2^{32}$, ωστόσο η σειρά των προσθέσεων είναι αδιάφορη. Αρχικά τα A, B, C, D, E τροφοδοτούνται με τις αρχικές τιμές H_0, H_1, H_2, H_3, H_4 όπως στην (4.2). Στον πρώτο υπογύρο εισέρχεται το W_0 μαζί με τη σταθερά K_0 και δημιουργείται η νέα πεντάδα A, B, C, D, E. Στον δεύτερο υπο-γύρο μαζί με τις νέες τιμές των A, B, C, D, E εισέρχεται το W_1 μαζί με τη σταθερά K_1 κ.ο.κ. μέχρι να συμπληρωθούν



Σχήμα 4.5: Το σχήμα υπο-γύρου του SHA-1.

80 υπο-γύροι με την επεξεργασία του W_{79}. Στο σημείο αυτό ολοκληρώνεται ένας γύρος επεξεργασίας, ο οποίος έχει λάβει υπόψη του όλο το block M_0. Οι τιμές των H_i ανανεώνονται ως

$$H_0 := H_0 + A, H_1 := H_1 + B, H_2 := H_2 + C, H_3 := H_3 + D, H_4 := H_4 + E$$

και αν δεν υπάρχει άλλο block το string $H_0 \| H_1 \| \ldots \| H_4$ είναι η σύνοψη του μηνύματος. Διαφορετικά επαναλαμβάνεται ο γύρος με τις νέες τιμές των H_i και το block M_1 κ.ο.κ. μέχρις εξαντλήσεως όλων των blocks.

SHA-2

Το 2002 με την σύσταση FIPS 180-2 εισήχθη μια οικογένεια αλγορίθμων που περιλαμβάνει τον SHA-1, αλλά και τους νέους αλγόριθμους SHA-256, SHA-384 και SHA-512. Η περιγραφή αναβαθμίσθηκε το 2015 με την FIPS 180-4, διατηρώντας τον τίτλο *Secure Hash Standard* (SHS). Η συνολική οικογένεια περιλαμβάνει τους αλγόριθμους του Πίνακα 4.1 που περιλαμβάνει το μέγιστο μήκος κειμένου και τα μεγέθη block, λέξης και σύνοψης. Στη συνέχεια περιγράφεται κυρίως ο αλγόριθμος SHA-256.

Αλγόριθμος	μέγ. μηνύματος	μέγ. block	μέγ. λέξης	μέγ. σύνοψης
SHA-1	$< 2^{64}$	512	32	160
SHA-224	$< 2^{64}$	512	32	224
SHA-256	$< 2^{64}$	512	32	256
SHA-384	$< 2^{128}$	1024	64	256
SHA-512	$< 2^{128}$	1024	64	512
SHA-512/224	$< 2^{128}$	1024	64	224
SHA-512/256	$< 2^{128}$	1024	64	256

Πίνακας 4.1: Αλγόριθμοι του SHS.

Σταθερές και προεπεξεργασία του μηνύματος

Για τον SHA-256, όπως και για τον SHA-224, χρησιμοποιούνται 64 σταθερές $K_0, \ldots, K_{63}$ μήκους 32 bits. Οι λέξεις αυτές αποτελούνται από τα πρώτα 32 bits του κλασματικού

μέρους της κυβικής ρίζας για τους πρώτους 64 πρώτους αριθμούς $2, 3, 5, \ldots, 311$. Ενδεικτικά οι πρώτες 4 είναι:

$$\begin{aligned} K_0 &= \text{428a2f98} \\ K_1 &= \text{71374491} \\ K_2 &= \text{b5c0fbcf} \\ K_3 &= \text{e9b5dba5} \end{aligned}$$

Για τους αλγόριθμους SHA-384, SHA-512, SHA-512/224 και SHA-512/256 χρησιμοποιούνται 80 σταθερές με 64 bits εκάστη, και πάλι με χρήση των κυβικών ριζών (κατά συνέπεια τα πρώτα 32 bits των πρώτων 64 σταθερών είναι ίδια με εκείνα των προηγουμένων).

Η επέκταση του μηνύματος προκειμένου το μήκος του να είναι πολλαπλάσιο του 512 γίνεται για τους SHA-224 και SHA-256 όπως για τον SHA-1. Για τους λοιπούς αλγόριθμους γίνεται με ανάλογο τρόπο προκειμένου να αποκτηθεί μήκος πολλαπλάσιο των 1024 bits.

Κατά την εκτέλεση των αλγόριθμων SHA-224 και SHA-256 συντηρείται ένα string με μήκος ίσο με 256 bits, χωρισμένο σε 8 λέξεις των 32 bits. Η αρχική τιμή αυτών των λέξεων για τον SHA-256 έχει δημιουργηθεί παίρνοντας τα πρώτα 32 bits από το κλασματικό μέρος της τετραγωνικής ρίζας των πρώτων 8 πρώτων αριθμών και έχει ως εξής:

$$\begin{aligned} H_0 &= \text{6a09e667} \\ H_1 &= \text{bb67ae85} \\ H_2 &= \text{3c6ef372} \\ H_3 &= \text{a54ff53a} \\ H_4 &= \text{510e527f} \\ H_5 &= \text{9b05688c} \\ H_6 &= \text{1f83d9ab} \\ H_7 &= \text{5be0cd19} \end{aligned}$$

Για τον SHA-224 είναι διαφορετικές, παρ' όλο που είναι ίδιου μήκους, ενώ για τους ανώτερης τάξης αλγόριθμους είναι διπλάσιου μήκους.

Επεξεργασία ενός block

Ο SHA-256 (καθώς επίσης και ο SHA-224) χρησιμοποιεί τις εξής συναρτήσεις:

$$\begin{aligned} \text{Ch}(x, y, z) &= (x \wedge y) \oplus (\neg x \wedge y) \\ \text{Maj}(x, y, z) &= (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z) \\ \Sigma_0^{\{256\}}(x) &= R^2(x) \oplus R^{13}(x) \oplus R^{22}(x) \\ \Sigma_1^{\{256\}}(x) &= R^6(x) \oplus R^{11}(x) \oplus R^{25}(x) \\ \sigma_0^{\{256\}}(x) &= R^7(x) \oplus R^{18}(x) \oplus r^3(x) \\ \sigma_1^{\{256\}}(x) &= R^{17}(x) \oplus R^{19}(x) \oplus r^{10}(x) \end{aligned}$$

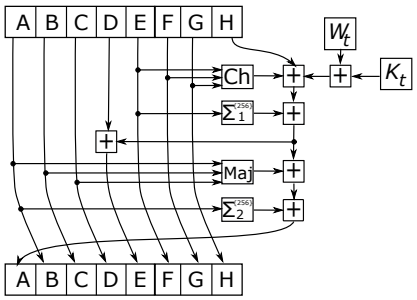
όπου $R^k = S^{-k}$ είναι δεξιά κυκλική μετάθεση κατά k και $r^k(x)$ είναι δεξιά ολίσθηση κατα k, όπου τα τελευταία δεξιά k bits χάνονται, ενώ τα πρώτα k αριστερά συμπληρώνονται με μηδενικά.

Πρώτα δίνουμε την επεξεργασία της πρώτης 512-άδας των bits του μηνύματος, δηλαδή του block M_0.

Αρχικά το block διαιρείται σε 16 (32-μπιτες) λέξεις $W_0, \ldots, W_{15}$ (όπου W_0 είναι η πρώτη από αριστερά). Στη συνέχεια σχηματίζονται άλλες 48 λέξεις για $t = 16, \ldots, 63$ ως εξής:

$$W_t = \sigma_1^{\{256\}}(W_{t-2}) + W_{t-7} + \sigma_0^{\{256\}}(W_{t-15}) + W_{t-16}$$

Αρχικοποιούνται τα $A, \ldots, H$ παίρνοντας τις αρχικές τιμές των $H_0, \ldots, H_7$ αντίστοιχα. Με τον τρόπο αυτόν έχουν σχηματισθεί από το αρχικό block M_0 (μήκους 512 bits) 64 λέξεις W_t των 32 bits. Κάθε μια από αυτές μπαίνει με τη σειρά (για $t = 0, 1, \ldots, 63$) σε ένα σχήμα 64 υπο-γύρων, όπως αυτό που φαίνεται στο Σχ. 4.6.



Σχήμα 4.6: Το σχήμα υπο-γύρου του SHA-256.

Μετά από αυτόν τον υπολογισμό οι τιμές των H_i ανανεώνονται ως

$$H_0 := H_0 + A, H_1 := H_1 + B, H_2 := H_2 + C, \ldots, H_4 := H_7 + H$$

Στη συνέχεια εισάγεται το επόμενο block M_1, σχηματίζονται εξ αυτού οι 64 λέξεις W_t, τα $A, B, C, \ldots, H$ παίρνουν τις τελευταίες τιμές των H_i και ακολουθούν οι 64 γύροι όπως πριν. Ο ίδιος γύρος επαναλαμβάνεται μέχρι και το τελευταίο block. Τελικά εξάγεται ως σύνοψη το string

$$H_0 \| H_1 \| \ldots \| H_7.$$

Με ανάλογο τρόπο λειτουργούν οι αλγόριθμοι SHA-384 και SHA-512 χρησιμοποιώντας 64-μπιτες λέξεις. Για τους αλγόριθμους SHA-512/224 και SHA-512/256 ο υπολογισμός γίνεται όπως στον SHA-512 (με μια διαφοροποίηση στην αρχικοποίηση) και στο τέλος χρησιμοποιούνται από τη σύνοψη μόνο τα πρώτα (αριστερά) 224/256 bits αντίστοιχα. Επίσης το πρότυπο έχει μια πρόβλεψη για την περίπτωση που χρειάζεται μήκος σύνοψης διαφορετικό από τα τυποποιημένα μεγέθη του Πίνακα 4.1 και επιτρέπει την επιλογή συγκεκομμένης σύνοψης σε όποιο μήκος είναι αναγκαίο. Παραπέμπει ωστόσο στην σύσταση SP 800-107 (*Recommendation for Applications Using Approved Hash Algorithms*) [Dan+12] για τις συνέπειες της χρήσης σύνοψης μειωμένου μήκους.

SHA-3

Ο αλγόριθμος SHA-3 που αποτελεί πρότυπο του NIST από το 2015 έχει ως βάση τον αλγόριθμο Keccak των Guido Bertoni, Joan Daemen, Michaël Peeters και Gilles Van Assche. Ο αλγόριθμος Keccak μπορεί από ένα αρχικό μήνυμα να δώσει έξοδο οποιουδήποτε μήκους. Έχει ενταχθεί στη σύσταση NIST FIPS 202 [Dwo] με διαφορετικές παραμετροποιήσεις για να προδιαγραφούν 4 συναρτήσεις κατακερματισμού ως συναρτήσεις SHA-3 με διαφορετικό μήκος εξόδου εκάστη (224, 256, 384 και 512 bits).

Η πορεία προς τον Keccak

Ήδη από το 1998 οι Joan Daemen και Craig Clapp είχαν προτείνει μια κρυπτογραφική συνάρτηση κατακερματισμού που την είχαν ονομάσει *Panama* [DC98]. Το 2006 οι Guido Bertoni, Joan Daemen και Gilles Van Assche πρότειναν το *RadioGatún* [Ber+06] σε μια προσπάθεια να ξεφύγουν από τις συναρτήσεις που αποτελούν απογόνους του MD4 (και οι οποίες περιλαμβάνουν τον SHA-1 και την οικογένεια SHA-2). Συγχρόνως οι MD4/5 και SHA-1 εθεωρούντο ήδη ασθενικές ως προς την αντίσταση σε συγκρούσεις, ενώ και για τον αλγόριθμο Panama είχαν προκύψει προβλήματα. Το RadioGatún ήταν μια προσπάθεια ανασχεδιασμού ενός αλγόριθμου ξεκινώντας από τον Panama. Στη συνέχεια δοκίμασαν τη χρήση των λεγόμενων συναρτήσεων *σπόγγου* (sponge function) [Ber+07; Ber+09] περνώντας στον αλγόριθμο *Keccak*.

Στο μεταξύ στην καθημερινή πράξη (π.χ. σε υπογραφές κειμένων και πιστοποιητικών TLS) κυριαρχούσαν οι απόγονοι του MD4, περιλαμβανομένων των SHA-1, SHA-2. Το 2005 δημοσιεύθηκε η πρώτη επίθεση σύγκρουσης για τον πλήρη SHA-1 [WYY05], αλλά παρέμεινε ως καθαρά θεωρητική δυνατότητα ως το 2017, οπότε και δημοσιεύθηκε από την Google σε συνεργασία με το *Centrum Wiskunde & Informatica* (CWI) του Amsterdam μια πρακτική υλοποίηση, συνοδευμένη από ένα ζευγάρι αρχείων pdf που έφεραν την ίδια σύνοψη (hash) [Ste+17]. Για ένα μεγάλο διάστημα επομένως ήταν γνωστές οι αδυναμίες του SHA-1, αλλά το γεγονός ότι δεν υπήρχαν ενδείξεις πρακτικής τους αξιοποίησης κράτησε τον αλγόριθμο σε χρήση. Παρ' όλα αυτά δημιουργήθηκε εξ αρχής ανησυχία για το μέλλον του αλγόριθμου και η ανησυχία αυτή επεκτεινόταν σε όλους τους αλγόριθμους της ίδιας οικογένειας, περιλαμβανομένων αυτών που υπάγονται στο SHA-2.

Η επιλογή αλγορίθμου με διαγωνισμό

Ήδη από το 2006 το NIST μπήκε στη διαδικασία επιλογής και τυποποίησης ενός σχήματος εναλλακτικού προς τον SHA-2, το οποίο θα λειτουργούσε κατά προτίμηση με αρκετά διαφορετική λογική από την οικογένεια MD/SHA. Το σχήμα αυτό θα αποτελούσε τον SHA-3 και οι δημιουργοί του Keccak είχαν την ευκαιρία να τον προτείνουν ως βάση του νέου σχήματος. Το NIST διοργάνωσε για την επιλογή του SHA-3 ένα διεθνή διαγωνισμό τύπου beauty contest, με κριτήρια σχετικά με την χρήση της συνάρτησης κατακερματισμού σε διαδικασίες ασφάλειας, το πλήθος των υπολογισμών, τις απαιτήσεις σε μνήμη κ.α.

Οι υποβολές υποψήφιων αλγορίθμων έφτασαν ως τον Οκτ. του 2008 τις 64. Εξ αυτών 51 έγιναν ως τον Δεκέμβρη δεκτές για κρίση στον πρώτο γύρο που κράτησε ένα εξάμηνο. Στη συνέχεια 14 πέρασαν στο δεύτερο γύρο τον Ιούλιο του 2009 και τον Δεκέμβρη του 2010 πέρασαν 5 στον τρίτο και τελευταίο γύρο, οι BLAKE, Grøstl, JH, Keccak και Skein [Reg+09]. Σε όλες τις υποψηφιότητες δόθηκε η ευκαιρία να κάνουν βελτιώσεις (tweeks) στη λειτουργία του αλγορίθμου.

Η κρυπτανάλυση του Keccak κατά την τότε αξιολόγηση έδειξε ότι ήταν εφικτός ο εντοπισμός σύγκρουσης μόνο στους 5 από τους 24 γύρους, δηλαδή σε ποσοστό 21% [Cha+12]. Το ποσοστό αυτό ήταν πιο χαμηλό από τα ποσοστά των υπολοίπων υποψηφίων αλγορίθμων. Το αμέσως καλύτερο ήταν του BLAKE με 29%, ενώ οι Grøstl και JH είχαν γύρω στο 60%. Το αντίστοιχο ποσοστό του SHA-2 ήταν 38% (24 γύροι στους 64). Ωστόσο η ανάλυση δεν είχε προχωρήσει στο ίδιο βάθος για όλους τους αλγορίθμους. Ο Keccak ήταν πιο γρήγορος και πιο οικονομικός σε hardware. Κάτι που επίσης έπαιξε ρόλο στην επιλογή ήταν το γεγονός ότι ήταν αρκετά διαφορετικής λογικής από τους ως τότε αλγόριθμους της οικογένειας SHA.

Σχεδόν δυο χρόνια χρειάστηκαν για την επιλογή του νικητή που ολοκληρώθηκε τον Οκτ. του 2012. Το 2014 ήταν έτοιμη μια πρόχειρη έκδοση του νέου προτύπου που ολοκληρώθηκε τον Αύγουστο του 2015 με την FIPS 202 [Dwo15]. Με δυο λόγια από την αρχική ιδέα ως την οριστικοποίηση του προτύπου χρειάστηκαν περί τα δέκα χρόνια.

Ο SHA-3 δεν αντικαθιστά τον SHA-2, ο οποίος παραμένει σε ισχύ όσο δεν έχει εμφανισθεί αποτελεσματική επίθεση εναντίον του. Συγχρόνως με την έκδοση του προτύπου για τον SHA-3 το NIST με την FIPS PUB 180-4 *Secure Hash Standard (SHS)* (Αύγ. 2015) επανέλαβε όλους τους αλγόριθμους της σειράς SHA χωρίς να καταργεί τον SHA-1 (για την ασφάλειά τους όμως παρέπεμψε στην SP 800-107 [Dan+12]). Ο SHA-3 αποτελεί μια ισχυρή εναλλακτική λύση προς το παρόν και τη διέξοδο σε περίπτωση που προκύψει πρόβλημα με τον SHA-2.

Ταυτόχρονα ο αλγόριθμος Keccak αποτελεί τη βάση των SHAKE128 και SHAKE-256 που προσφέρονται με το πρότυπο ως *επεκτάσιμες συναρτήσεις εξόδου* (extendable-output functions - XOFs), δηλαδή ως συναρτήσεις κατακερματισμού με οσοδήποτε μεγάλο μήκος εξόδου.

Γενική άποψη του αλγόριθμου

Ο αλγόριθμος Keccak συντηρεί (όπως και οι άλλοι) ένα string μήκους $r + c$, το οποίο εμπλουτίζεται διαδοχικά με input από κάθε επόμενο block p_i του μηνύματος, όπως φαίνεται στο Σχ. 4.7. Η προσθήκη ενός block γίνεται με XOR μόνο στα πρώτα r bits του συντηρούμενου string. Μετά την είσοδο ενός block γίνονται πάνω σε όλο το string διάφοροι μετασχηματισμοί από μια συνάρτηση f και προστίθεται το επόμενο block. Όταν ενσωματωθεί και το τελευταίο block μπορεί να εξαχθεί ένα πρώτο string, αλλά μπορούν να εξαχθούν και επόμενα, κάθε φορά μετά από ένα στάδιο μετασχηματισμού μέσω της f. Θεωρητικά η έξοδος μπορεί να είναι με αυτόν τον τρόπο απεριόριστου μήκους.

Η κατασκευή αυτή με την οποία ενσωματώνονται διαδοχικά blocks αρχικά χωρίς να παράγεται έξοδος, ενώ στη συνέχεια παράγεται έξοδος χωρίς περαιτέρω ενσωμάτωση, λέγεται *σπόγγος*. Η πρώτη φάση προσθήκης των blocks λέγεται φάση απορρόφησης, ενώ η δεύτερη φάση στιψίματος.

Εσωτερικά η συνάρτηση f περιλαμβάνει επεξεργασία του string σε έναν αριθμό από γύρους, 24 για τον Keccak 1600. Το string οργανώνεται σε μορφή ενός πίνακα τριών διαστάσεων, με την έννοια ότι οι μετασχηματισμοί που προκαλούνται από την f σε κάθε γύρο αφορούν σε μεταθέσεις των στηλών σε διάφορες διαστάσεις ή μεταθέσεις των στοιχείων εντός των στηλών.

Ο σπόγγος

Ο σπόγγος [Ber+07] παίρνει ως είσοδο strings αποτελούμενα από σύμβολα που είναι στοιχεία μιας ομάδας $\mathcal{A}$ που είναι το αλφάβητο και παράγει έξοδο μέσα από το ίδιο σύνολο. Η πράξη της ομάδας παριστάνεται με $a+$ και το ουδέτερο στοιχείο με 0.Το $\mathcal{A}$ μπορεί να είναι bits, χαρακτήρες ή ακόμη και blocks από n bits. Επίσης διαθέτει ένα σύνολο $\mathcal{C}$ που χρησιμοποιείται εσωτερικά στον σπόγγο. Το $\mathcal{C}$ περιέχει ένα «μηδενικό» στοιχείο που θα αποτελέσει μέρος της αρχικής τιμής της κατάστασης του σπόγγου.

Έστω $p(m)$ μια αμφιμονοσήμαντη απεικόνιση από το σύνολο των μηνυμάτων σε strings από σύμβολα του $\mathcal{A}$ τέτοια ώστε $|p(m)| \geq 1$ και δεν λήγει σε 0. Για παράδειγμα, έστω ότι $\mathcal{A} = \mathbb{Z}_2^n$. Τότε η $p(m)$ θα μπορούσε να είναι ένα αναστρέψιμο πα-

ραγέμισμα (padding) όπως έγινε στους αλγόριθμους SHA-1 και 2, δηλαδή προσθήκη ψηφίων ώστε το μήκος να γίνει πολλαπλάσιο του n. Το τελευταίο είναι 1.

Η είσοδος του σπόγγου είναι ένα string p αποτελούμενο από σύμβολα του $\mathcal{A}$. Ο i-οστός χαρακτήρας του p θα συμβολίζεται με p_i ($i = 1, \ldots, |p|$).

Η εκάστοτε κατάσταση του σπόγγου είναι ένα στοιχείο του $\mathcal{A} \times \mathcal{C}$, δηλαδή ένα ζεύγος $S = (S_\mathcal{A}, S_\mathcal{C})$ με αρχική τιμή $(0, 0)$ (όπου τα δύο αυτά «μηδενικά» έχουν οριστεί πιο πάνω για το $\mathcal{A}$ και το $\mathcal{C}$). Η λειτουργία του σπόγγου αποτελείται από δύο φάσεις ως εξής:

Απορρόφηση Για κάθε χαρακτήρα p_i η κατάσταση $S = (S_\mathcal{A}, S_\mathcal{C})$ του σπόγγου μεταβάλλεται ως εξής:

$$(S_\mathcal{A}, S_\mathcal{C}) := f(S_\mathcal{A} + p_i, S_\mathcal{C})$$

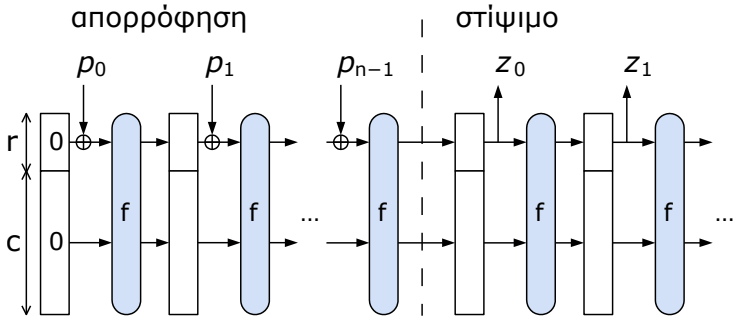
Στίψιμο Παράγεται ένα απείρου μήκους string z, του οποίου ο επόμενος κάθε φορά χαρακτήρας προκύπτει ως

$$z_j = S_\mathcal{A}$$

και η κατάσταση ανανεώνεται ως

$$S := f(S)$$

Η διαδικασία αυτή απεικονίζεται στο Σχ. 4.7.[6]



Σχήμα 4.7: Ο σπόγγος.

Ορίζονται τα εξής δύο μεγέθη:

- Ο *ρυθμός* του σπόγγου
$$r = \log_2 |\mathcal{A}|$$
- Η *χωρητικότητα* του σπόγγου
$$c = \log_2 |\mathcal{C}|$$

Παραγέμισμα

Το παραγέμισμα στον SHA-3 γίνεται με προσθήκη ενός αρχικού bit 1, μιας σειράς από μηδενικά και ενός ακόμη τελικού bit 1, ώστε το συνολικό μήκος να είναι πολλαπλάσιο του r. Η προσθήκη ενός ακόμη block γίνεται ακόμη και αν το αρχικό μήκος του μηνύματος είναι $kr + r - 1$ ή kr.

[6] Αρχείο χρησιμοποιούμενο με άδεια Creative Commons Attribution 3.0 Unported και τροποποιήσεις στο κείμενο. Προέλευση `http://sponge.noekeon.org`.

Ο πίνακας κατάστασης

Το συνολικό μήκος b του string S που διατηρείται κατά την εκτέλεση του αλγόριθμου και αποτυπώνει την τρέχουσα κατάσταση του συστήματος μπορεί να πάρει οποιαδήποτε από τις τιμές

$$b = 25 \times 2^{\ell}, \quad (\ell = 0, 1, \ldots, 6) \tag{4.3}$$

Η μεγαλύτερη και πρακτικά πιο ενδιαφέρουσα τιμή είναι $b = 1600$. Αν συμβολίσουμε με S_i ($i = 0, \ldots, b-1$) τα b bits του S, δηλαδή

$$S = S_0 \| S_1 \| \ldots \| S_{b-1}$$

Έστω $w = b/25$. Κατασκευάζουμε ένα πίνακα A τριών διαστάσεων μεγέθους $5 \times 5 \times w$ και τα στοιχεία του τα ονομάζουμε $A_{x,y,z}$. Προφανώς $x = 0, 1, 2, 3, 4$, $y = 0, 1, 2, 3, 4$ και $z = 0, 1, 2, \ldots, w-1$. Πέντε στοιχεία κατά μήκος της διάστασης x (δηλ. που διαφέρουν κατά το x, ενώ έχουν ίδια y, z) λέγονται *γραμμή* (row) του πίνακα, πέντε στοιχεία κατά μήκος της διάστασης y λέγονται *στήλη* (column) και 25 στοιχεία κατά μήκος της διάστασης z λέγονται *λωρίδα* (lane):

$$\text{Lane}_{x,y} = A_{x,y,0} \| \ldots \| A_{x,y,w-1}$$

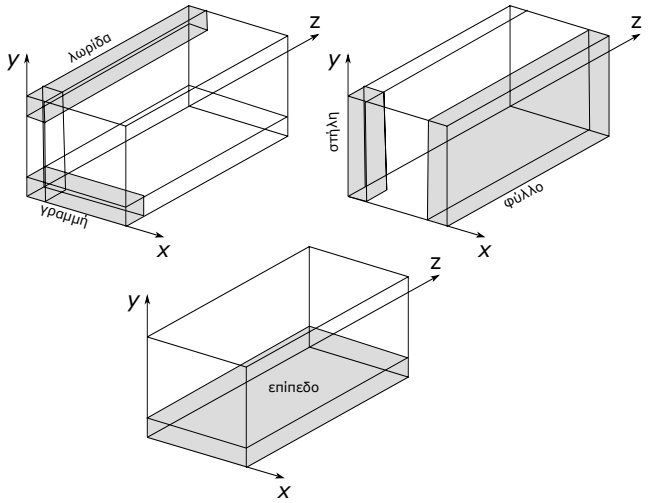
Επίσης $5w$ στοιχεία στο ίδιο επίπεδο x, z (δηλ. έχουν το ίδιο y) λέγονται *επίπεδο* (plane):

$$\text{Plane}_y = \text{Lane}_{0,y} \| \ldots \| \text{Lane}_{4,y}$$

Πέντε επίπεδα μαζί είναι όλο το string S:

$$S = \text{Plane}_0 \| \ldots \| \text{Plane}_4$$

Ένα κατακόρυφο επίπεδο αποτελούμενο από λωρίδες λέγεται *φύλλο* (sheet), δηλαδή αποτελείται από $5w$ στοιχεία με το ίδιο το ίδιο x. Οι ομαδοποιήσεις αυτές των bits φαίνονται στο Σχ. 4.8.



Σχήμα 4.8: Ο πίνακας A.

Τα b bits του string S εισάγονται στον πίνακα ως εξής:

$$A_{x,y,z} = S_{w(5y+x)+z}$$

δηλαδή πρώτα γεμίζει η λωρίδα με $(x, y) = (0, 0)$ και μετά οι διπλανές της στο ίδιο επίπεδο (με αύξηση του x) και στη συνέχεια προστίθενται με τον ίδιο τρόπο επίπεδα στην διεύθυνση του y. Κατά κάποιο τρόπο τα x, z, y είναι αντίστοιχα πλάτος, βάθος και ύψος του πίνακα και τα στοιχεία συμπληρώνονται πρώτα σε κολώνες κατά βάθος, μετά προστίθενται κολώνες κατά πλάτος και στη συνέχεια επίπεδα καθ' ύψος. Για $b = 1600$ κάθε λωρίδα έχει $w = 64$ στοιχεία και είναι κατάλληλη για επεξεργασία από ένα 64-μπιτο επεξεργαστή.

Συναρτήσεις εντός της f

Στη συνέχεια θα δούμε τι συμβαίνει μέσα στην f του Σχ. 4.7. Χρησιμοποιείται μια σειρά από n_r γύρους, όπου $n_r = 12 + 2\ell$. Το ℓ συνδέεται με το μήκος b του string S μέσω της (4.3), για $b = 1600$ το ℓ είναι 6, οπότε $n_r = 24$. Καθένας από τους γύρους επιφέρει στον πίνακα κατάστασης διαδοχικές μεταβολές που περιγράφονται από μια σειρά συναρτήσεων.

Οι συναρτήσεις χρησιμοποιούν μια αρίθμηση των στοιχείων του A διαφορετική από την προφανή που είδαμε ως τώρα. Για την διάσταση x η αρίθμηση γίνεται ως $3, 4, 0, 1, 2$ (αντί της $0, 1, 2, 3, 4$), δηλαδή το φύλλο με $x = 0$ είναι το κεντρικό φύλλο του πίνακα, ενώ το πρώτο είναι το φύλλο με $x = 3$. Το ίδιο συμβαίνει με την διάσταση y, όπου πάλι η αρίθμηση είναι $3, 4, 0, 1, 2$, άρα το επίπεδο με $y = 0$ είναι το κεντρικό επίπεδο. Προσέξτε ότι πρόκειται για μια κυκλική μετάθεση των δεικτών που φέρνει για τους δύο δείκτες x, y την αρχή των αξόνων στο κέντρο.

Συνάρτηση θ Κάθε στοιχείο $A_{x,y,z}$ του A το κάνει XOR με το bit ισοτιμίας καθεμιάς από δύο στήλες, αυτής που είναι δίπλα αριστερά από τη θέση x, y, z, δηλαδή την στήλη $(x - 1, z)$ και την στήλη $(x + 1, z - 1)$ που βρίσκεται διαγωνίως προς τη θέση x, y, z (όπου όλες οι προσθαφαιρέσεις δεικτών γίνονται mod5, δηλ. κυκλικά, και η επαναρίθμηση x, y δεν έχει επίδραση). Πιο συγκεκριμένα, υπολογίζεται πρώτα το bit ισοτιμίας κάθε στήλης μέσω της

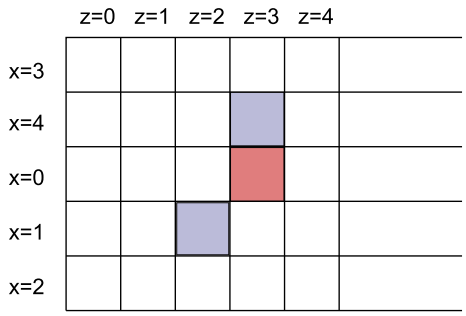
$$C[x, z] = \bigoplus_{y=0}^{4} A[x, y, z] = \sum_{y=0}^{4} A[x, y, z] \bmod 2$$

και στη συνέχεια μεταβάλλονται όλα τα στοιχεία του A βάσει της

$$A_{x,y,z} := A_{x,y,z} \oplus C[x - 1, z] \oplus C[x + 1, z - 1]$$

Προσέξτε ότι τα $C[x, z]$ υπολογίζονται όλα πριν να αρχίσει η μεταβολή των $A_{x,y,z}$, οπότε δεν επηρεάζονται από αυτήν. Τελικό αποτέλεσμα της θ είναι ότι κάθε στήλη είτε μένει ως έχει είτε αναστρέφεται ολόκληρη.

Παράδειγμα: Έστω ότι επεξεργαζόμαστε ένα οποιοδήποτε στοιχείο $(0, y, 3)$ του πίνακα. Στο Σχ. 4.9 βλέπουμε τον πίνακα A «από πάνω», οπότε το στοιχείο αυτό βρίσκεται μέσα στην κόκκινη στήλη. Το εν λόγω στοιχείο θα γίνει XOR με τα bits ισοτιμίας των δύο γαλάζιων στηλών. Όλα τα στοιχεία της στήλης θα γίνουν XOR εν τέλει με το ίδιο bit.



Σχήμα 4.9: Επεξεργασία με τη συνάρτηση θ.

Συνάρτηση ρ Κάνει κυκλικές δεξιές ολισθήσεις ανά λωρίδα (lane) σύμφωνα με ένα ψευδοτυχαίο αριθμό διαφορετικό για κάθε λωρίδα. Ο κάθε αριθμός για τη λωρίδα των στοιχείων $(x, y, 0), \ldots (x, y, 23)$ φαίνεται στον Πίνακα 4.2, όπου η ολίσθηση τελικά πρέπει να εκλαμβάνεται mod$w$ (64 στον Keccak 1600). Η παράμετρος t αφορά στον υπολογισμό του πίνακα.

t	x	y	ολίσθηση
0	1	0	1
1	0	2	3
2	2	1	6
3	1	2	10
4	2	3	15
5	3	3	21
6	3	0	28
7	0	1	36
8	1	3	45
9	3	1	55
10	1	4	66
11	4	4	78
12	4	0	91
13	0	3	105
14	3	4	120
15	4	3	136
16	3	2	153
17	2	2	171
18	2	0	190
19	0	4	210
20	4	2	231
21	2	4	253
22	4	1	276
23	1	1	300

Πίνακας 4.2: Υπολογισμός των τιμών ολίσθησης για την ρ.

Στόχος της εκτέλεσης της ρ είναι να δημιουργηθεί ο πίνακας A' από τις τιμές του A. Ο A' αποτελείται από ολισθημένες λωρίδες του A και υπολογισμός του γίνεται ως

εξής:

1. Η λωρίδα $(A_{0,0,z})_{z=0,\ldots,w-1}$ δεν μεταβάλλεται, δηλαδή θέτουμε $A'_{0,0,z} = A_{0,0,z}$ για $z = 0, \ldots, w-1$.

2. Θέτουμε ως αρχική συνθήκη $(x, y) = (1, 0)$.

3. Υπολογίζουμε επαναληπτικά για $t = 0$ ως 23:

   α) $A'[x, y, z] = A[x, y, z - (t+1)(t+2)/2 \bmod w]$ για $z = 0, \ldots, w-1$

   β) $(x, y) := (y, 2x + 3y \bmod 5)$

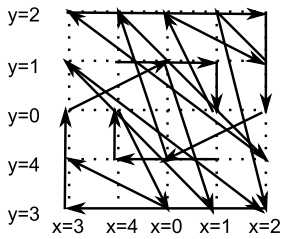
Αποτέλεσμα του βρόχου του βήματος 3 είναι ο υπολογισμός μιας ολίσθησης κατά αριθμό θέσεων που δίνει ο τύπος $(t+1)(t+2)/2$ για κάθε μία από τις λωρίδες, διότι τελικά παράγονται όλα τα δυνατά ζεύγη (x, y), αλλά με ανακατεμένες αυτές τις τιμές. Ο υπολογισμός αυτός έχει γίνει με τη σειρά του παραπάνω αλγόριθμου στον Πίν. 4.2 και οι τιμές της ολίσθησης που αναγράφονται είναι αυτές πριν την τελική αναγωγή $\bmod w$. Στο τέλος τίθεται $A = A'$.

Συνάρτηση π Μετακινεί όλες τις λωρίδες σε νέες θέσεις. Στον Πίνακα 4.3 δίνεται η νέα θέση (x', y') μιας λωρίδας που είναι πριν την εφαρμογή της π στη θέση (x, y). Η μετακίνηση (και ο υπολογισμός του πίνακα) δίνεται μέσω του τύπου:

$$A'_{x,y,z} = A_{x+3y \bmod 5, x, z} \quad (0 \le x \le 4, 0 \le y \le 4, 0 \le z \le w-1)$$

Για παράδειγμα $A'_{3,3,z} = A_{12 \bmod 5, 3, z} = A_{2,3}$, δηλαδή η λωρίδα με $(x, y) = (2, 3)$ πάει στη θέση $(x, y) = (3, 3)$.

Υπενθυμίζεται ότι ισχύει η αλλαγή της αρίθμησης του πίνακα που περιγράψαμε παραπάνω, δηλαδή η κεντρική λωρίδα είναι η $(0, 0)$. Ο τύπος μετακίνησης δείχνει ότι αυτή δεν αλλάζει θέση. Στο Σχήμα 4.10 κάθε βέλος δείχνει την αρχική και τελική θέση μιας λωρίδας (x, y). Μπορεί να διαπιστώσει κανείς ότι υπάρχει μια συμμετρία ως προς το κέντρο $(0, 0)$ στις μετακινήσεις, δηλαδή ζευγάρια από λωρίδες που αρχικά βρίσκονται σε μεταξύ τους συμμετρικές θέσεις ως προς το κέντρο καταλήγουν σε επίσης συμμετρικές θέσεις.



Σχήμα 4.10: Αρχική και τελική θέση λωρίδων κατά την εφαρμογή της π.

Εννοείται ότι όταν ολοκληρωθεί ο υπολογισμός του A' τίθεται $A = A'$.

Συνάρτηση χ Επεξεργάζεται κάθε σειρά του A ξεχωριστά, δηλαδή το αποτέλεσμα κάθε σειράς εξαρτάται μόνο από την δική της προηγούμενη τιμή και όχι από άλλα στοιχεία του A. Η βασική πράξη είναι να κάνει XOR κάθε bit με ένα μη γραμμικό

x'	y'	x	y
0	0	0	0
0	1	3	0
0	2	1	0
0	3	4	0
0	4	2	0
1	0	1	1
1	1	4	1
1	2	2	1
1	3	0	1
1	4	3	1
2	0	2	2
2	1	0	2
2	2	3	2
2	3	1	2
2	4	4	2
3	0	3	3
3	1	1	3
3	2	4	3
3	3	2	3
3	4	0	3
4	0	4	4
4	1	2	4
4	2	0	4
4	3	3	4
4	4	1	4

Πίνακας 4.3: Υπολογισμός των τιμών περιστροφής για την π.

συνδυασμό των δύο επόμενων bits της ίδιας σειράς, ως εξής: Για όλα τα (x, y, z) ο πίνακας A' υπολογίζεται μέσω της

$$A'_{x,y,z} = A_{x,y,z} \oplus [(A_{x+1 \bmod 5,y,z} \oplus 1) \odot A_{x+2 \bmod 5,y,z}]$$

όπου $\odot$ είναι πολλαπλασιασμός, που εδώ $\bmod 2$ είναι το ίδιο με το λογικό AND.

Συνάρτηση ι Η συνάρτηση ι (γιώτα) είναι η μοναδική από τις εσωτερικές της f συναρτήσεις που λαμβάνει υπόψη τον αύξοντα αριθμό του γύρου και γεννάει $\ell + 1$ bits για κάθε λωρίδα ως σταθερά RC (round constant). Σκοπός της είναι να τροποποιήσει ορισμένα bits της κεντρικής λωρίδας $(0, 0)$. Οι υπόλοιπες 24 λωρίδες δεν επηρεάζονται.

Η συνάρτηση χρησιμοποιεί μια άλλη συνάρτηση, την $\text{rc}(t)$, η οποία ορίζεται για ακέραιο t $\bmod 255$) και παράγει τις τιμές των round constants. Οι τιμές των πρώτων 24 round constants (που είναι οι μόνες που χρειάζονται στον SHA-3) δίνονται στον Πίνακα 4.4 σε δεκαεξαδική μορφή (big endian) για $w = 64$, ενώ για μικρότερα μήκη προκύπτουν κρατώντας μόνο τα πρώτα χαμηλότερης τάξης w bits. Όλες οι

Σε κάθε γύρο i ($i = 0, 1, \ldots, n_r - 1$) η συνάρτηση ι κάνει XOR το string της κεντρικής λωρίδας $(0, 0)$ με το αντίστοιχο $\text{RC}[i]$.

RC[0]	0000000000000001	RC[12]	000000008000808B
RC[1]	0000000000008082	RC[13]	800000000000008B
RC[2]	800000000000808A	RC[14]	8000000000008089
RC[3]	8000000080008000	RC[15]	8000000000008003
RC[4]	000000000000808B	RC[16]	8000000000008002
RC[5]	0000000080000001	RC[17]	8000000000000080
RC[6]	8000000080008081	RC[18]	000000000000800A
RC[7]	8000000000008009	RC[19]	800000008000000A
RC[8]	000000000000008A	RC[20]	8000000080008081
RC[9]	0000000000000088	RC[21]	8000000000008080
RC[10]	0000000080008009	RC[22]	0000000080000001
RC[11]	000000008000000A	RC[23]	8000000080008008

Πίνακας 4.4: Τιμές των πρώτων 24 round constants για την ι.

Ο αλγόριθμος Keccak μέσα στον σπόγγο

Η f μεταμορφώνει ένα πίνακα κατάστασης A εφαρμόζοντας για έναν αριθμό γύρων όλες τις προηγούμενες συναρτήσεις με τη σειρά σε κάθε γύρο.

Αν ως Rnd ορισθεί η συνάρτηση που εφαρμόζεται σε ένα γύρο, αυτή εκτός του πίνακα A χρησιμοποιεί ως παράμετρο και το i_r που υποδηλώνει τον δείκτη του γύρου. Τότε η $mboxRnd$ έχει ως εξής:

$$mboxRnd(A, i_r) = \iota(\chi(\pi(\rho(\theta(A)))), i_r)$$

Δηλαδή εφαρμόζονται οι συναρτήσεις με τη σειρά θ, ρ, π, χ που είναι ανεξάρτητες του τρέχοντος γύρου και τελικά εφαρμόζεται η ι που λαμβάνει υπόψη το i_r.

Ο αλγόριθμος Keccak-$p[b, n_r](S)$ που εφαρμόζεται πάνω σε μια σειρά από bits (string) S μήκους b για οποιονδήποτε επιθυμητό αριθμό γύρων n_r λειτουργεί ως εξής: Μετατρέπει το S σε πίνακα A όπως έχει περιγραφεί πιο πριν. Εφαρμόζει την παραπάνω R πάνω στον A για n_r γύρους με τιμές του i από $12+2\ell-n_r$ ως $12+2\ell-1$. Μετατρέπει πίσω σε S τον τελικό πίνακα A. Αυτή είναι και η έξοδος του αλγορίθμου.

Μέσω συγκεκριμένης παραμετροποίησης του Keccak-$p[b, n_r]$ ορίζεται η *οικογένεια συναρτήσεων σπόγγου Keccak* ως αυτή για την οποία το n_r παίρνει τις τιμές $12 + 2\ell$ και η επαύξηση του κειμένου γίνεται με ένα string της μορφής 10^*1 (δηλ. 1, μετά μηδενικά και μετά πάλι 1, όπως περιγράψαμε πιο πριν). Στην οικογένεια αυτή το $b = r+c$ παίρνει μια από τις τιμές του συνόλου $\{25, 50, 100, 200, 400, 800, 1600\}$, δηλαδή τις τιμές $25 \times 2^\ell$. Στην περίπτωση $b = 1600$ ($\ell = 6$) η υπο-οικογένεια που προσδιορίζεται χαρακτηρίζεται ως Keccak[c] και το κάθε μέλος της χαρακτηρίζεται από την επιλογή του c. Η οικογένεια Keccak[c] έχει προφανώς $n_r = 12 + 2 \times 6 = 24$ και είναι αυτή που χρησιμοποιείται για τον προσδιορισμό των συναρτήσεων κατακερματισμου του SHA-3 ως εξής:

- SHA3-224 με $c = 448$, άρα $r = 1152$.
- SHA3-256 με $c = 512$, άρα $r = 1088$.
- SHA3-384 με $c = 768$, άρα $r = 832$.
- SHA3-512 με $c = 1024$, άρα $r = 576$.

Οι συναρτήσεις SHAKE

Στο πρότυπο ορίζονται επιπροσθέτως οι δύο *επεκτάσιμες συναρτήσεις εξόδου* (extendable-output functions - XOFs) SHAKE128 και SHAKE256, οι οποίες έχουν αυθαιρέτου μήκους έξοδο χρησιμοποιώντας το σπόγγο στη φάση «στιψίματος» για όσο χρειαστεί. Η SHAKE128 ορίζεται μέσω του Keccak$[256]$ και η SHAKE 256 μέσω του Keccak$[512]$, με την διαφορά ότι το padding γίνεται με διαφορετικό string. Για οποιαδήποτε περαιτέρω λεπτομέρεια ο αναγνώστης παραπέμπεται στο κείμενο του προτύπου [Dwo].

ΚΕΦΑΛΑΙΟ 5

Κακόβουλο λογισμικό

> Αὐτὰρ ὅτ᾽ εἰς ἵππον κατεβαίνομεν, ὃν κάμ᾽ Ἐπειός, Ἀργείων οἱ ἄριστοι, ἐμοὶ δ᾽ ἐπὶ πάντα τέταλτο, ἠμὲν ἀνακλῖναι πυκινὸν λόχον ἠδ᾽ ἐπιθεῖναι
>
> Οδύσσεια, ραψ. λ᾽, 523-525.

5.1 Τι είναι το κακόβουλο λογισμικό

Με το *κακόβουλο λογισμικό* (malware) έρχεται σε επαφή αργά ή γρήγορα σήμερα ο καθένας, είτε το θέλει είτε όχι. Από τα αντικείμενα που εκτίθενται μέσα σ᾽ ένα βιβλίο ασφάλειας το κακόβουλο λογισμικό είναι εκείνο, από το οποίο καθένας έχει παραστάσεις και ενδεχομένως δυσάρεστες εμπειρίες, κατά κανόνα από ιούς και παραπλανητικά μηνύματα. Ορισμένοι πιο άτυχοι βιώνουν πιο δυσάρεστες καταστάσεις, όπως απώλειες δεδομένων, βλάβες υπολογιστών, εκβιασμούς, αφαίρεση ποσών από τραπεζικούς λογαριασμούς κ.α. Όλα τα προηγούμενα έχουν συνήθως την πηγή τους σε κάποιου είδους κακόβουλο λογισμικό. Την ίδια προέλευση έχουν και ζημιές είτε πιο σπάνιες, είτε λιγότερο αντιληπτές, είτε μη αντιληπτές ως σχετικές με την ασφάλεια (όπως π.χ. βλάβη στους φωτεινούς σηματοδότες, καθυστερημένα τρένα ή μια διακοπή ρεύματος).

Οι Grégio, Afonso κ.α. [Gré+15] ορίζουν το κακόβουλο λογισμικό ως *σύνολο εντολών που τρέχουν σε ένα σύστημα για να κάνει αυθαίρετες ενέργειες για λογαριασμό ενός επιτιθέμενου ή να ενεργήσει με (αυτόματο ή μη) τρόπο τέτοιο ώστε να απειλείται η ασφάλεια του αλωθέντος συστήματος, οι χρήστες του και σχετικά δεδομένα.*

Ο Éric Filiol έγραψε ένα σημαντικό βιβλίο περί ιών [Fil06] και προτιμάει τον εξής ορισμό για το malware: *Ένα πρόγραμμα που προσβάλλει υπολογιστές είναι ένα απλό ή ένα αυτοαναπαραγόμενο πρόγραμμα που εγκαθιστά διακριτικά τον εαυτό του σε ένα σύστημα επεξεργασίας δεδομένων, χωρίς να το γνωρίζουν ή να συναινούν σ᾽ αυτό οι χρήστες του, με σκοπό να θέσει σε κίνδυνο την ακεραιότητα και την εμπιστευτικότητα των δεδομένων, την διαθεσιμότητα του συστήματος ή προκειμένου να παγιδεύσει τους χρήστες του για την περαιτέρω διάπραξη εγκληματικών ενεργειών.*

Το νέο στοιχείο σ' αυτόν τον ορισμό βρίσκεται στην τελευταία φράση, όπου επισημαίνεται ότι η εγκατάσταση κακόβουλου λογισμικού μπορεί να εμπλέξει κάποιον αθώο χρήστη σε πράξεις τις οποίες εκείνος αγνοεί, μερικές φορές παρουσιάζοντας στοιχεία που τον εμφανίζουν ως προφανή ένοχο. Γενικά οι χρήστες αποτελούν τον ασθενέστερο κρίκο στην ασφάλεια ενός συστήματος, δεδομένου ότι είναι σχετικά εύκολο να παρασυρθούν με μεθόδους κοινωνικής μηχανικής και να διευκολύνουν την διάδοση του κακόβουλου λογισμικού.

Με μια έννοια στην περιοχή της ασφάλειας υπολογιστών και δικτύων σχεδόν όλες οι απειλές και οι ζημιές υλοποιούνται με κάποιου είδους λογισμικό, με δευτερεύουσα μόνο εμπλοκή του υλισμικού (hardware). Το λογισμικό είναι πιο εύκολο να δημιουργηθεί ή να αλλοιωθεί καθώς και να μεταφερθεί μέσα από τα σύγχρονα δίκτυα. Μπορεί να δώσει από μόνο του απτά αποτελέσματα στον επιτιθέμενο, ή περαιτέρω να χρησιμοποιηθεί για την αλλοίωση της συμπεριφοράς ή τη βλάβη του hardware ώστε να επιτευχθεί ένας σκοπός. Η τάση για μεταφορά λειτουργιών από το λογισμικό στο υλισμικό (όπως για παράδειγμα στα Software Defined Networks), η καθημερινή δημιουργία νέων εφαρμογών σε κάθε τομέα δραστηριότητας, καθώς και η γενίκευση της χρήσης λογισμικού σε παντός είδους συσκευές που μετατρέπονται σε «έξυπνες» (smart devices) χειροτερεύουν την κατάσταση από πλευράς ασφάλειας.

Σπανιότερες είναι οι επιθέσεις με ευθείες επεμβάσεις στο hardware, αν και όχι ανύπαρκτες. Στην εποχή της παγκοσμιοποίησης, όπου η παραγωγή του υλισμικού γίνεται από υποκατασκευαστές με εργοστάσια που βρίσκονται μακριά από σημείο σχεδιασμού, οι κίνδυνοι πολλαπλασιάζονται, π.χ. κακόβουλες μετατροπές σε συνιστώσες (components) [Bec+13]. Ακόμη και αν η παραγωγή γίνεται σε ελεγχόμενο περιβάλλον, αντικαταστάσεις γνήσιων μερών με αλλοιωμένα μπορούν να γίνουν πιο κάτω στην εφοδιαστική αλυσίδα. Σε ορισμένες περιπτώσεις τα κέρδη είναι άλλης κλίμακας: Αν κάποιος καταφέρει να εγκαταστήσει ένα back door σε ένα επεξεργαστή γενικής χρήσης, θα αποκτήσει αμέσως πρόσβαση σε εκατομμύρια μηχανών.

Το λογισμικό παραμένει το κύριο όχημα διεξαγωγής επιθέσεων και παντός είδους παρανομιών, εξ αιτίας της ευελιξίας του, της δυνατότητας για τοποθέτησή του χωρίς φυσική πρόσβαση και του χαμηλού κόστους.

Στην ελληνική γλώσσα ο όρος *malware* αποδίδεται συνήθως ως *κακόβουλο λογισμικό*, με την κατανόηση ότι πίσω απ' αυτό κρύβεται ένας κακόβουλος επιτιθέμενος, ενώ αυτό εγκαθίσταται σε σύστημα του θύματος.[1] Σε ορισμένες γκρίζες περιπτώσεις δεν είναι σαφές αν υπάρχουν αυτές οι προϋποθέσεις. Αν ο κάτοχος ενός κινητού εγκαταστήσει μια εφαρμογή καταγραφής της φωνής και των δύο πλευρών της τηλεφωνικής συνδιάλεξης και την χρησιμοποιεί παρανόμως χωρίς να ζητήσει την άδεια της άλλης πλευράς, η εφαρμογή πιθανότατα δεν θα χαρακτηρισθεί ως malware. Διαφορετική είναι η περίπτωση αν κάποιος εγκαταστήσει μια εφαρμογή καταγραφής στην κινητή συσκευή ενός άλλου χωρίς την άδειά του. Μεγάλο μέρος της σχετικής έρευνας και μεθοδολογίας γύρω από το malware αφιερώνεται στις μεθόδους της εισβολής, η οποία προφανώς δεν υφίσταται αν το λογισμικό έχει εγκατασταθεί σε ένα σύστημα από τον ιδιοκτήτη του συστήματος ή με την άδεια και τη συνέργειά του. Λογισμικό που εγκαθίσταται από αφελές θύμα εξαπατημένο με μεθόδους κοινωνικής μηχανικής και χωρίς αντίληψη της βλάβης πιθανώς θα περιληφθεί στο malware. Αν το θύμα γνωρίζει τη λειτουργία του, αλλά την έχει αποδεχτεί υποτιμώντας τις συνέπειες ή δεχόμενο κάποιο αντάλλαγμα, η περίπτωση αυτή πάλι πέφτει στη γκρίζα περιοχή και ο χαρακτηρισμός ως malware ή όχι μπορεί να αποτελέσει ζήτημα ερμηνείας, τρεχουσών αντιλήψεων, νομικού πλαισίου κ.λπ.

[1] Πιθανώς καλύτερος ελληνικός όρος θα ήταν *βλαπτικό λογισμικό* ή *βλαβερό λογισμικό*.

Το κακόβουλο λογισμικό βεβαίως έχει ως βασικό λόγο ύπαρξης να ζημιώσει το θύμα και να προσπορίσει όφελος στον επιτιθέμενο. Η εκτίμηση της ζημιάς δεν είναι πάντοτε τόσο εύκολη, συχνά υπάρχουν δευτερογενείς απώλειες για το θύμα απλωμένες σε μια εκτεταμένη χρονική περίοδο, ενώ σε άλλες περιπτώσεις η ζημιά είναι σχεδόν μηδενική. Επίσης, σε ορισμένες περιπτώσεις η ζημιά δεν γίνεται ποτέ αντιληπτή από το θύμα. Περαιτέρω η καταγραφή των ζημιών συχνά δεν προχωράει στην αντίστοιχη ανακοίνωση, είτε από το θύμα που δεν επιθυμεί να αμαυρώσει την καλή του φήμη, είτε από τις εταιρίες που οργανώνουν την άμυνα. Κατά συνέπεια οι επιθέσεις και ζημιές που δημοσιοποιούνται είναι κατώτερες σε πλήθος και όγκο της πραγματικότητας.

Συγχρόνως όμως το κακόβουλο λογισμικό χρειάζεται κάποιου είδους πρόσβαση σε πόρους του θύματος ή μια επιζήμια αλληλεπιδραση με αυτούς. Μια συνηθισμένη περίπτωση είναι να εγκατασταθεί σε ένα υπολογιστή του θύματος, αλλά για να φτάσει εκεί χρειάζεται να γίνει κάποιου είδους εισβολή, όπως αυτή την οποία αναλαμβάνει ένας ιός. Μια άλλη επίσης δημοφιλής επίθεση γίνεται με παρεμπόδιση της επικοινωνίας του θύματος με τον έξω κόσμο, π.χ. με μια επίθεση υπερβολικής κίνησης σε τηλεπικοινωνιακούς πόρους. Γενικά επομένως ο επιτιθέμενος πρέπει να λύσει δύο διακριτά προβλήματα σε σχέση με πόρους του θύματος:

1. Πώς θα αποκτήσει πρόσβαση σ' αυτούς ή θα έρθει σε αλληλεπίδραση με αυτούς και

2. αν και πώς θα τους βλάψει για να εξυπηρετήσει τους σκοπούς του.

Για παράδειγμα, ένας ιός μπορεί να είναι από αβλαβής ως ιδιαζόντως καταστροφικός σε καθένα από τα συστήματα που θα προσβάλει. Μπορεί να προσβάλλει με ευκολία πολλά συστήματα ή λίγα, τυχαία ή στοχευμένα. Η συνολική βλάβη που θα προκαλέσει εξαρτάται και από τις δύο σχετικές ιδιότητές του.

Κατά συνέπεια η μελέτη του κακόβουλου λογισμικού απλώνεται και στα δυο αυτά ζητήματα. Η μελέτη αυτή μπορεί να γίνει από την οπτική γωνία είτε του αμυνόμενου είτε του επιτιθέμενου.

Για πολλά χρόνια στην ανάλυση του κακόβουλου λογισμικού, όπως και στην κρυπτογραφία, επικρατούσε μια νοοτροπία γνωστή ως "security by obscurity" δηλαδή όσο λιγότερα είναι γνωστά, τόσο λιγότεροι είναι οι κίνδυνοι. Η σύγχρονη άποψη είναι ότι ασφαλέστερο είναι το σύστημα που είναι ανοιχτό και σκληρά δοκιμασμένο από επιθέσεις, γεγονός που κάνει απαραίτητη τη γνώση και των δύο πλευρών από τους ειδικούς της περιοχής.

5.2 Ταξινόμηση

Όπως σε όλες τις οντολογίες [CJB99] υπάρχουν διαφορετικοί τρόποι ταξινόμησης του malware. Μερικοί από τους πιο δημοφιλείς τρόπους ταξινόμησης στην περιοχή του κακόβουλου λογισμικού χρησιμοποιούν τα εξής κριτήρια:

- Τον τρόπο διάδοσης ή και αναπαραγωγής,
- το είδος της βλάβης που προκαλείται (ή απειλείται να προκληθεί) και ο τρόπος υλοποίησής της,
- το σκοπούμενο τελικό αποτέλεσμα (π.χ. ο εκβιασμός για να αποσπασθούν λύτρα ή η αφαίρεση χρημάτων από λογαριασμούς),
- τον τρόπο διαχείρισης από τον επιτιθέμενο,

- την κατηγορία συσκευών-στόχων (με διάφορα επί μέρους κριτήρια, π.χ. το λειτουργικό σύστημα ή την βασική τους εξειδικευμένη λειτουργία),
- την κατηγορία των θυμάτων (π.χ. συνηθισμένοι χρήστες email, καταθέτες τραπεζών, ταξιδιώτες),
- την κατηγορία του χρήσιμου λογισμικού ή διεργασίας με την οποία πιθανώς σχετίζονται.

Για να διαμορφωθούν «οντολογίες» από κάποια «όντα» πρέπει αυτά στην καλύτερη περίπτωση να έχουν ονόματα ή έστω κάποιες περιγραφές. Οι *γάτες* και οι *σκύλοι* υπάγονται στην κατηγορία των *θηλαστικών*, αλλά ήδη οι γάτες μπορούν να ταξινομούνται σε συγκεκριμένες ράτσες και κάτω κάτω σ' αυτόν τον γράφο τοπολογίας δέντρου εμφανίζονται ως φύλλα συγκεκριμένες γάτες.

Κατά βάση εδώ τα «φύλλα» στον γράφο της οντολογίας είναι ατομικά περιστατικά ασφάλειας, πιο απλά ή πιο σύνθετα. Καθένα από αυτά τα περιστατικά μπορεί να μοιάζει με προηγούμενα σε κάποιες από τις ιδιότητές του ή να είναι εντελώς νέο. Ένα περιστατικό μπορεί να προσδιοριστεί πάντοτε από μια σειρά χαρακτηριστικών (π.χ. τόπος, χρόνος, τύπος μηχανής που έχει προσβληθεί, είδος προσβολής κ.λπ.), αλλά κατηγορίες παρόμοιων αξιοπρόσεκτων επαναλαμβανόμενων περιστατικών (ή και μοναδικών σε ορισμένες περιπτώσεις) τείνουν να αποκτούν το δικό τους όνομα. Για παράδειγμα αν ένας ιός με τον ίδιο κώδικα έχει προσβάλει μερικές χιλιάδες υπολογιστών και έχει παντού σβήσει το boot record σε λειτουργικό Windows έκδοσης τάδε, ίσως πάρει το δικό του όνομα. Το ίδιο μπορεί να συμβεί με μια επαναλαμβανόμενη επίθεση κίνησης που γίνεται με συγκεκριμένη μέθοδο.

Σε γενικές γραμμές σε ανιχνευμένα σημαντικά περιστατικά ασφάλειας απονέμεται κατά κανόνα ένα όνομα. Με τα χρόνια έχει αναπτυχθεί μια όχι ιδιαίτερα συστηματική ταξινόμηση και ονοματολογία από διάφορες πηγές, κυρίως από εταιρίες που παράγουν λογισμικό προστασίας. Στην εργασία [Gré+15] (σελ. 2759, πίνακας 1) περιγράφεται το εξής πείραμα: Οι ερευνητές έστειλαν το ίδιο δείγμα κακόβουλου λογισμικού σε οκτώ διαφορετικές εταιρίες παραγωγής συστημάτων anti-virus και πήραν οκτώ διαφορετικές απαντήσεις για το όνομα ή την κατηγορία του. Σε τρεις από αυτές τις απαντήσεις αναγνωρίσθηκε ως ιός (virus), rootkit και δούρειος ίππος (Trojan). Το ουσιώδες επομένως ερώτημα (και μάλιστα για ένα κατασκευαστή antivirus) δεν είναι πώς θα ονομάσουμε τον κώδικα ή το φαινόμενο που προκαλεί, αλλά τι θα κάνουμε γι' αυτό [BH08].

Στο περίφημο βιβλίο του Filiol με τίτλο *Computer Viruses* [Fil06] προτείνεται η απλή διάκριση ανάμεσα σε *αυτο-αναπαραγόμενο* λογισμικό (ιοί και σκουλήκια) και απλό ή *επειακό* (*Epeian*, από τον Επειό[2] που κατασκεύασε τον Δούρειο Ίππο στον τρωικό πόλεμο).

Οι Weaver, Paxson κ.α. [Wea+03] έχουν προτείνει να χρησιμοποιηθούν τα εξής κριτήρια ταξινόμησης για σκουλήκια:

- Τρόπος ανακάλυψης του στόχου,
- μέθοδος διάδοσης,
- μέθοδος ενεργοποίησης,

[2] Η μετάφραση από τον Ζήσιμο Σιδέρη του κειμένου της Οδύσσειας που βρίσκεται στην αρχή αυτού του κεφαλαίου έχει ως εξής: *Και στ' άλογο όταν μπήκαμε των Αχαιών οι πρώτοι, που το 'χε φτιάσει ο Επειός κι εγώ όλη τη φροντίδα είχα ν' ανοίγω ή να σφαλώ τη στεριωμένη κρύφτρα* (λόγια που βάζει ο Όμηρος στο στόμα του Οδυσσέα).

- φορτίο (κώδικας προκαλεί τη βλάβη) και
- κίνητρα του επιτιθέμενου.

Ο Fred Cohen ασχολήθηκε κυρίως με το ζήτημα της αναπαραγωγής και διάδοσης του λογισμικού [Coh87], το οποίο μπορεί να είναι ή να μην είναι βλαπτικό. Ο Leonard Adleman, που υπήρξε και επιβλέπων της διατριβής του Cohen, έκανε μια ταξινόμηση του λογισμικού που λαμβάνει υπόψη και κατά πόσο αυτό είναι βλαπτικό. Προφανώς υπάρχουν 4 κατηγορίες ως προς τις δύο αυτές ιδιότητες της διάδοσης και της βλαπτικότητας ή *παθογένειας*: (α) Λογισμικό που δεν διαδίδεται και δεν είναι *παθογενές* (δεν προξενεί βλάβες), (β) λογισμικό που είναι παθογενές, αλλά δεν διαδίδεται, (γ) λογισμικό που διαδίδεται χωρίς να είναι παθογενές, (δ) λογισμικό που διαδίδεται και είναι παθογενές. Στην κατηγορία (β), που είναι γνωστή και ως *επειακή* (epeian, από τον Επειό, δημιουργό του Δούρειου Ίππου) υπάγονται π.χ. λογικές βόμβες, δούρειοι ίπποι κ.λπ. Στην κατηγορία (γ) υπάγονται ιοί που δεν έχουν *φορτίο*, δηλαδή εντολές με τις οποίες να προκαλούν ζημιές. Οι συνήθεις ιοί, σκουλήκια κ.λπ. προφανώς υπάγονται στην (δ).

Κίνητρα του επιτιθέμενου

Ο επιτιθέμενος μπορεί να έχει πολύ διαφορετικά κίνητρα [Wea+03] που συνδέονται με τη φύση του και κατά περίπτωση διαμορφώνουν τον τρόπο επίθεσης. Τέτοια κίνητρα μπορεί να είναι τα εξής (αρχίζοντας από τα πιο αθώα):

Περιέργεια και πειραματισμός Εδώ εντάσσονται δημιουργοί κακόβουλου λογισμικού από παιδιά που διασκεδάζουν (και αδιαφορούν για τις ζημιές που μπορεί να προκληθούν) ως ερευνητές που πειραματίζονται (και έχουν μια πιο υπεύθυνη στάση, αλλά μπορεί να χάσουν τον έλεγχο).

Ασκήσεις επίδειξης και δύναμης Περιλαμβάνονται συνήθως ανοργάνωτοι μη συστηματικοί επιτιθέμενοι που θέλουν να δείξουν τι είναι ικανοί να επιτύχουν.

Εμπορικό πλεονέκτημα Εντάσσονται επαγγελματίες ως και μεγάλες εταιρίες που ασκούν έτσι αθέμιτο ανταγωνισμό ή βιομηχανική κατασκοπεία. Εδώ η επίθεση είναι πιθανό να ανατεθεί σε εξωτερικούς επαγγελματίες (με κίνητρο των τελευταίων την αμοιβή).

Τυχαία διαμαρτυρία από κάποιον που θεωρεί εαυτόν αδικημένο, ή έχει απολυθεί από μια εταιρία ή είναι απλώς ανισόρροπος.

Πολιτική, ακτιβιστική, τρομοκρατική δράση από άτομα ή οργανωμένες ομάδες, που όμως δεν ανήκουν σε μυστικές ή στρατιωτικές υπηρεσίες.

Έγκλημα από μεμονωμένα άτομα ή και οργανωμένες εγκληματικές ομάδες με συνήθη σκοπό να αποσπασθούν χρηματικά ποσά.

Κυβερνοπόλεμος που ασκείται από ειδικές κρατικές μυστικές υπηρεσίες στα πλαίσια της αντιπαράθεσης μεταξύ κρατών.

Συχνά οι παραπάνω κατηγορίες δεν έχουν σαφή όρια και η διάκριση ανάμεσά τους είναι δύσκολη. Τον Ιούλιο του 2020 ο Άγγλος υπουργός James Brokenshire δήλωσε βεβαιότητα πάνω από 95% ότι ρώσοι εισβολείς υπέκλεψαν την τεχνολογία εμβολίων για τον Covid-19 από Αγγλία και Αμερική [Wal20]. Η δήλωση άραγε αυτή υπονοεί ιδιωτικούς εισβολείς ή μυστικές υπηρεσίες;

Τα κίνητρα συνδέονται προφανώς στενά με το ποιον των επιτιθεμένων, την εξειδίκευση, την εκπαίδευση, την εργασία, τις πεποιθήσεις τους. Συνδέονται περαιτέρω με τα μέσα και το είδος των επιθέσεων. Η σχετικά νεότερη γενιά επιθέσεων που καλούνται *προχωρημένες επίμονες απειλές* (advanced persistent threats, APTs) απαιτούν πόρους που μόνο κρατικές υπηρεσίες και μείζονες εγκληματικοί οργανισμοί διαθέτουν.

Φορτίο

Το φορτίο στην περίπτωση ενός ιού, σκουληκιού, δούρειου ίππου κ.λπ. είναι το μέρος του κώδικα που είναι υπεύθυνο για την ζημιά, υπεξαίρεση δεδομένων κ.λπ. Μερικές από τις κατηγορίες φορτίου που αναφέρονται στην κυρίως στο [Wea+03] είναι οι εξής:

Μηδενικό ή εξουδετερωμένο φορτίο που μπορεί να συμβεί είτε εκ προθέσεως είτε από λάθος. Παρ' όλα αυτά ένας ιός με ουδέτερο φορτίο μπορεί παρ' όλα αυτά να προκαλέσει κίνηση.

Διαφημίσεις Το *adware*, δηλαδή λογισμικό που δείχνει διαφημίσεις είτε με είτε χωρίς την προηγούμενη αποδοχή του χρήστη, είναι οριακή περίπτωση ως προς το αν μπορεί να θεωρηθεί βλαβερό. Συνήθως είναι απλώς ενοχλητικό. Παρ' όλα αυτά συχνά κατατάσσεται ως malware [QKC19].

Διαβίβαση spam δηλαδή διευκόλυνση αποστολής μηνυμάτων spam.

Διευκόλυνση επίθεσης άρνησης υπηρεσίας , ήτοι εγκαθίσταται λογισμικό που όταν έρθει η ώρα δημιουργεί υπερβολική κίνηση προς συγκεκριμένο στόχο.

Συλλογή δεδομένων τα οποία περαιτέρω είναι αξιοποιήσιμα για χρηματικό όφελος.

Καταστροφή δεδομένων είτε στοχευμένων, είτε μέχρι να αχρηστευθούν οι λειτουργίες ενός συστήματος.

Παρεμπόδιση της πρόσβασης του νόμιμου χρήστη σε δικά του δεδομένα και πόρους με σκοπό τον εκβιασμό και την απόσπαση λύτρων (ransomware).

Λογ. κατασκοπείας (spyware) παρακολουθεί ένα χρήστη και αποσπά ευαίσθητα ή άλλα σημαντικά δεδομένα.

Exploit εκμεταλλεύεται μια τρωτότητα για την απόκτηση πρόσβασης σε μια μηχανή.

Ζημιές σε μικτά συστήματα (cyber-physical systems), όπου τελικά αλλοιώνεται η συμπεριφορά του συνολικού συστήματος. Διάσημο παράδειγμα είναι το σκουλήκι Stuxnet που αχρήστευσε φυγοκεντρικά μηχανήματα εμπλουτισμού ουρανίου.

Διάδοση

Ο τρόπος διάδοσης έχει παραδοσιακά χρησιμοποιηθεί εκτεταμένα για τον χαρακτηρισμό και την ονομασία του κακόβουλου λογισμικού. Γνωστές συνηθισμένες κατηγορίες είναι οι εξής:

Ιός (virus) παρασιτικό λογισμικό με ικανότητα αναπαραγωγής.

Σκουλήκι (worm) αυτόνομο αναπαραγόμενο λογισμικό (που δεν χρειάζεται να αποτελέσει μέρος άλλου προγράμματος).

Δούρειος ίππος (Trojan) βλαβερό λογισμικό καλυμμένο μέσα σε χρήσιμο λογισμικό, προκειμένου να παρασυρθεί κάποιος να το κατεβάσει και να το ενεργοποιήσει (δηλαδή εδώ εν μέρει χρησιμοποιείται κοινωνική μηχανική).

Απ᾽ ευθείας πρόσβαση και τοποθέτηση του βλαβερού λογισμικού, είτε με πρόσβαση στην ίδια τη μηχανή είτε διαδικτυακά [Hea06].

Μια φημισμένη περίπτωση που δείχνει πόσο σύνθετη μπορεί να είναι η διαδικασία της απ᾽ ευθείας τοποθέτησης του λογισμικού είναι αυτή του σκανδάλου των ελληνικών υποκλοπών του 2005 [PS07; Bam15]. Σε πρώτη φάση έγινε με ενημέρωση λογισμικού η τοποθέτηση ενός ανενεργού συστήματος νόμιμων παρακολουθήσεων, του οποίου ο επίσημος σκοπός ήταν να ενεργοποιηθεί μετά από χρόνια με την άδεια του εισαγγελέα. Η βασική λειτουργία της υποκλοπής έγινε ενεργοποιώντας αυτό το σύστημα με την προσθήκη κώδικα που παρακάμπτει τη διαδικασία νόμιμης έγκρισης.

Πλατφόρμες και συσκευές

Η κλασσική περίπτωση ενός ιού είναι αυτή που μεταδίδεται σε υπολογιστές με λειτουργικό σύστημα Windows, το οποίο έχει γίνει πόλος έλξης επιθέσεων. Το ότι η Microsoft είναι κυρίαρχος παίκτης στην αγορά των υπολογιστών αφενός έχει συγκεντρώσει κατά καιρούς αντιπάθειες, αφετέρου εξασφαλίζει στον δημιουργό ενός ιού το ευρύτερο πιθανό σύνολο στόχων. Σ᾽ αυτά έρχονται να προστεθεί μια χαλαρότητα στην πολιτική προστασίας του λειτουργικού και η ομοιομορφία [Noy10] που κάνει πιο εύκολο τον σχεδιασμό του ιού. Η Apple, της οποίας το λειτουργικό σύστημα βασίζεται στο Linux και για ένα διάστημα εθεωρείτο εναλλακτική εταιρία, είχε συγκεντρώσει λιγότερα πυρά, αλλά αυτό έχει μεταβληθεί την τελευταία δεκαετία. Αντικείμενο επίθεσης όμως μπορεί να είναι οποιοδήποτε, ακόμη και εξειδικευμένο λειτουργικό σύστημα.

Η εξέλιξη των κινητών τηλεφώνων σε έξυπνες συσκευές με δικά τους λειτουργικά συστήματα (android, iOS και δεκάδες άλλα) έφερε τους ιούς και στις κινητές συσκευές. Η μεγαλύτερη ίσως πρόκληση για τους επιτιθέμενους είναι ο αριθμός αυτών των συσκευών, καθώς και το γεγονός ότι χρησιμοποιούνται όλο και περισσότερο για οικονομικές συναλλαγές. Επί πλέον είναι εκτεθειμένες σε μια ειδική κατηγορία επιθέσεων που γίνονται από μικρή απόσταση, είτε μέσω bluetooth είτε μέσω του εγγύς πεδίου (near field).

Δημοφιλείς σήμερα συσκευές για την πραγματοποίηση επιθέσεων κίνησης είναι αυτές που έχουν χαμηλής πολυπλοκότητας και ασφάλειας λειτουργικό, αλλά χειρίζονται σημαντικούς όγκους δεδομένων, όπως π.χ. network streamers που χρησιμοποιούνται για την αναπαραγωγή περιεχομένου (μουσικής και video).

Εργαλεία

Μια κατηγορία βλαβερού λογισμικού είναι αυτή που προετοιμάζει μια βλαπτική πράξη με την έννοια ότι δίνει μια σειρά γενικού σκοπού εργαλείων και επιλογών στον επιτιθέμενο. Στη συνέχεια αυτός μπορεί να ενεργοποιήσει και να συνδυάσει τα εκάστοτε πρόσφορα εργαλεία για την επιθυμητή επίθεση. Τέτοιες περιπτώσεις είναι οι εξής:

Rootkit είναι λογισμικό που επιτρέπει στον επιτιθέμενο να αποκτήσει απομακρυσμένη πρόσβαση σε ένα σύστημα και στη συνέχεια να πραγματοποιήσει τις ενέργειες που επιθυμεί.

Backdoor είναι λογισμικό που επιτρέπει την πρόσβαση με μη κανονικό τρόπο, δηλαδή παρακάμπτοντας την αυθεντικοποίηση.

Botnet είναι ένα δίκτυο μηχανών που έχουν παραβιαστεί και εφοδιαστεί με λογισμικό συνήθως γενικής χρήσης, έτσι ώστε να χρησιμοποιηθούν κατά περίπτωση για διαφόρων ειδών επιθέσεις προς εξωτερικά θύματα.

Τα παραπάνω είναι εργαλεία που ο επιτιθέμενος εγκαθιστά στη μηχανή του θύματος ή σε μια μηχανή που δεν ανήκει στο τελικό θύμα, αλλά σε κάποιον ανύποπτο ενδιάμεσο, του οποίου η μηχανή απλώς χρησιμοποιείται. Αμφότεροι πάντως δεν έχουν γνώση της εγκατάστασης.

Υπάρχουν όμως και τα εργαλεία που διευκολύνουν την δημιουργία κακόβουλου λογισμικού οποιασδήποτε κατηγορίας. Συλλογές τέτοιων εργαλείων αποτελούν *εργαλειοθήκες* (crimeware kit, infection kit, exploit kit, DIY attack kit, malware toolkit είναι μερικές από τις ονομασίες που χρησιμοποιούνται). Η ανάπτυξή τους κατέστησε δυνατή την διεξαγωγή επιθέσεων από μη ειδικούς της πληροφορικής, ακόμη και από μικρά παιδιά. Στην άλλη άκρη του φάσματος όμως είναι εξελιγμένες εργαλειοθήκες, των οποίων επωφελούνται είτε εγκληματικές ομάδες είτε μυστικές υπηρεσίες [Jac+13].[3]

Τυποποίηση της ταξινόμησης

Η μεγάλη πρακτική σημασία που έχει η υιοθεσία μιας οντολογίας για το κακόβουλο λογισμικό καταλήγει σε μια λογική τυποποίησης.

Ενδεχομένως οι πιο αρμόδιοι στην πράξη να απαντήσουν στα ερωτήματα της ταξινόμησης και της ονοματολογίας είναι οι δημιουργοί εργαλείων antivirus. Η Kaspersky χρησιμοποιεί το εξής σύστημα ονοματολογίας:[4]

```
[Prefix:]Behaviour.Platform.Name[.Variant]
```

όπου `Prefix` είναι το σύστημα που ανίχνευσε το αντικείμενο, `Behaviour` είναι τι κάνει το αντικείμενο, `Platform` είναι το περιβάλλον (software ή hardware) όπου εκτελείται ο κώδικας, `Name` χαρακτηρίζει την οικογένεια αντικειμένων, `Variant` είναι όνομα ή αριθμός που εξειδικεύει το αντικείμενο μέσα στην οικογένεια. Π.χ. το αντικείμενο `Trojan.Win32.StartPage.xw` είναι ένας δούρειος ίππος που τρέχει σε Windows και μεταβάλλει την αρχική σελίδα του φυλλομετρητή (browser). Παρόμοια είναι η μέθοδος ονομασίας που χρησιμοποιεί η Microsoft,[5] που είναι

```
Type:Platform/Family.Variant!Suffixes
```

π.χ. το `HackTool:Win64 /AutoKMS!bit` επιτρέπει την εκτέλεση προγραμμάτων της Microsoft χωρίς νόμιμη άδεια, αλλά μπορεί να φέρει και άλλο κακόβουλο μέρος. Το αντικείμενο `Trojan :Win32/Wacatac.B!ml` επιτρέπει σε ένα hacker να πραγματοποιήσει διάφορες ενέργειες.

Οι ονοματολογίες αυτές υιοθετούν σε γενικές γραμμές την σύμβαση για την ονοματοδοσία ιών του CARO[6] (Computer Antivirus Research Organization) του 1991 με το εξής format:

```
FamilyName.GroupName.MajorVariant.MinorVariant[[:Modifier]
```

[3] Βλ. και `https://en.wikipedia.org/wiki/NSA_ANT_catalog`.

[4] `https://encyclopedia.kaspersky.com/knowledge/rules-for-naming/`

[5] `https://docs.microsoft.com/en-us/windows/security/threat-protection/intelligence/malware-naming`

[6] `http://www.caro.org/articles/naming.html`

όπου

- `FamilyName` είναι όπως προηγουμένως μια οικογένεια ιών, αλλά στην απόδοση ονόματος πρέπει να τηρείται μια σειρά κανόνων, όπως να αποφεύγονται εταιρικά ονόματα, να μη χρησιμοποιούνται νέα ονόματα ήταν ήδη υπάρχει κατάλληλο όνομα, να μη χρησιμοποιούνται προσβλητικά ονόματα κ.λπ.
- `GroupName` είναι μια «υπο-οικογένεια».
- `MajorVariant`, `MinorVariant`, `Modifier` εξειδικεύουν περαιτέρω το αντικείμενο.

Σχετική είναι και η *Common Malware Enumeration (CME) Initiative* της *Computer Emergency Readiness Team* των ΗΠΑ (US-CERT). Αντιστοίχως για τις τρωτότητες υπάρχει το *Common Weakness Enumeration* (CWE). Η CME έχει καταλήξει στην πρόταση μιας γλώσσας για την περιγραφή του malware. Η γλώσσα αυτή είναι η *Malware Attribute Enumeration and Characterization* (MAEC)[7] και λαμβάνει υπόψη τα ίδια χαρακτηριστικά που αναφέρθηκαν πιο πάνω για την ονοματολογία, δηλαδή οικογένεια, συμπεριφορά κ.λπ.

5.3 Αυτοαναπαραγόμενο λογισμικό

Στην ενότητα αυτήν εκτίθενται συνοπτικά ζητήματα αυτοαναπαραγωγής, μεταμορφισμού και ανίχνευσης λογισμικού. Τα θέματα αυτά σχετίζονται στενά με ιούς και σκουλήκια (worms), αλλά αποτελούν γενικότερα ζητήματα για την περιοχή του λογισμικού. Βασίζονται στο κλασσικό μοντέλο του Turing για τον υπολογιστή και στις θεωρίες του von Neumann για αυτοαναπαραγόμενες μηχανές. Τα πιο πολλά αποτελέσματα που θα δούμε βασίζονται στην διδακτορική διατριβή του Fred(erick) B. Cohen στο University of Southern California (1986). Την επίβλεψη της διατριβής είχε ο Leonard Adleman ("A" στον RSA), ο οποίος επίσης ασχολήθηκε με το θέμα. Οι Cohen και Adleman πρότειναν τον όρο *ιός υπολογιστή* (computer virus) [Coh87]. Ο ορισμός του Cohen έχει ως εξής [Fil06]:

Ορισμός 2. Ιός είναι μια ακολουθία συμβόλων που είναι ικανή, αν διερμηνευθεί σε κατάλληλο περιβάλλον (σε μια μηχανή), να τροποποιήσει άλλες ακολουθίες συμβόλων σ' αυτό το περιβάλλον συμπεριλαμβάνοντας ένα, πιθανώς εξελιγμένο, αντίγραφο του εαυτού του.

Ικό σύνολο

Στη συνέχεια ο Cohen όρισε το *ικό σύνολο* (viral set) [Coh89], το οποίο περιέχει όλες τις διαφορετικές αλλά ισοδύναμες παραλλαγές ενός ιού που προκύπτουν μέσω κάποιου υπολογισμού πάνω σε μια δεδομένη μηχανή. Στην πράξη η γέννηση παραλλαγών εμφανίζεται ως *πολυμορφισμός*: Ο δημιουργός ενός ιού φροντίζει να μεταβάλλεται ο κώδικάς του ώστε να μην είναι εύκολα ανιχνεύσιμος με μεθόδους *υπογραφής*, οι οποίες βασίζονται στην ανίχνευση συγκεκριμένων χαρακτηριστικών ακολουθιών κώδικα. Η μεταβολή του κώδικα μπορεί να βασίζεται στη χρήση ισοδύναμων εντολών ή ακολουθιών εντολών, δηλαδή εντολών που έχουν διαφορετική μορφή, αλλά το ίδιο τελικό αποτέλεσμα.

[7] `http://maecproject.github.io/about-maec/`

Για τον ορισμό του παραπάνω *ιικού συνόλου*, αλλά και την απόδειξη μιας σειράς άλλων θεωρημάτων, ο Cohen έχει χρησιμοποιήσει το υπολογιστικό μοντέλο της μηχανής Turing. Θεωρώντας ότι ο αναγνώστης είναι ήδη εξοικειωμένος με τις μηχανές Turing,[8] υπενθυμίζεται ότι μια μηχανή Turing αποτελείται από μια ταινία, από την οποία μπορεί να διαβάζει σύμβολα, ένα σε κάθε βήμα, καθώς και να γράφει σύμβολα, μέσω μιας κεφαλής, η οποία κινείται πάνω στην ταινία. Κάθε φορά που διαβάζει ένα σύμβολο η μηχανή λαμβάνει υπόψη της την τρέχουσα *κατάστασή* της και περνάει στην επόμενη κατάσταση λαμβάνοντας υπόψη ένα πίνακα, που έχει ως εισόδους τις καταστάσεις και τα δυνατά σύμβολα και ως έξοδο αφενός μια κατάσταση για κάθε τέτοιο ζεύγος κατάστασης και συμβόλου, αφετέρου την επόμενη κίνησή της (μια θέση δεξιά ή μια θέση αριστερά ή παραμονή στην ίδια θέση), εκτός αν έχει φτάσει σε μια από τις *τελικές* καταστάσεις, οπότε η διαδικασία τελειώνει. Στα επόμενα συμβολίζουμε με S τη συμβολοσειρά που εμφανίζεται πάνω στην ταινία και με M την *μηχανή*, δηλαδή όλα τα υπόλοιπα σταθερά της στοιχεία (κατά βάση τον πίνακα μετάβασης).

Ο ιός στο μοντέλο αυτό εμφανίζεται ως μια σειρά συμβόλων πάνω στην ταινία. Διαβάζοντας αυτή τη σειρά η μηχανή δημιουργεί πάνω στην ταινία την επόμενη μορφή του σε μια σειρά διαδοχικών θέσεων, σε ικανή απόσταση από την προηγούμενη σειρά. Κατά συνέπεια ένας ιός S πάνω σε μια μηχανή M δημιουργεί τον ιό S'. Δείτε όμως ότι η ίδια ακολουθία S εν γένει δεν θα είναι ικανή να αναπαραχθεί σε μια άλλη μηχανή M'. (Είναι το ίδιο σαν να λέμε ότι ένας ιός γραμμένος σε συγκεκριμένη γλώσσα προσβάλλει υπολογιστές με συγκεκριμένο λειτουργικό σύστημα.) Συνεπώς έχει νόημα ο ορισμός του ιικού συνόλου μόνο σε σχέση με μια συγκεκριμένη μηχανή.

Ας θεωρήσουμε τη σχέση $v \overset{M}{\Rightarrow} v'$, όπου ο ιός v' παράγεται από τον ιό v μέσω της μηχανής M, και περαιτέρω ας θεωρήσουμε μεταβατικό κάλυμμα ως προς αυτήν την σχέση αρχίζοντας από το v (δηλαδή το σύνολο όλων των v'' που παράγονται από το v σε οσαδήποτε βήματα εφαρμογής της σχέσης). Τότε, οποιοσδήποτε ιός v'' ανήκει σε αυτό το σύνολο λέγεται (εξ ορισμού) ότι αποτελεί *ιική εξέλιξη* (viral evolution) του ιού v.

Αποδεικνύεται ότι η ένωση δύο ιικών συνόλων αποτελεί ιικό σύνολο. Μπορεί περαιτέρω να ορισθεί το μέγιστο τέτοιο σύνολο καθώς και το ελάχιστο (κάθε γνήσιο υποσύνολο του οποίου δεν είναι ιικό σύνολο) [Fil06]. Το μικρότερο ιικό σύνολο μπορεί να είναι μονομελές και αποτελεί την περίπτωση όπου ένας ιός δεν είναι πολυμορφικός. Μάλιστο ο Cohen έδειξε ότι για οποιονδήποτε φυσικό αριθμό μπορεί να βρεθεί μια μηχανή κι ένας ιός με ελάχιστο ιικό σύνολο που θα έχει ως πληθικό αριθμό αυτόν τον φυσικό αριθμό. Επίσης έδειξε την ύπαρξη μηχανών για τις οποίες οποιαδήποτε ακολουθία εισόδου δεν αποτελεί ιό (δεν μπορεί να γεννήσει ιικό σύνολο).

Ανιχνευσιμότητα

Ένα βασικό πρόβλημα σε σχέση με τους ιούς και άλλο κακόβουλο λογισμικό είναι κατά πόσο και με ποιο τρόπο μπορούν να ανιχνευθούν. Ο Cohen έδωσε μια απόδειξη ότι το πρόβλημα είναι *μη αποφασίσιμο* (undecidable) μέσω του προβλήματος *τερματισμού* (halting problem), δηλαδή έδειξε ότι ισχύει το εξής:

Θεώρημα 3. Το πρόβλημα της ανίχνευσης ενός ιού είναι μη αποφασίσιμο. □

Η απόδειξη περιληπτικά έχει ως εξής: Θεωρούμε μια μηχανή M' και μια ακολουθία v' στην ταινία. Κατασκευάζουμε τη μηχανή M και την ακολουθία v έτσι ώστε η M να

[8] Διαφορετικά ο αναγνώστης μπορεί να συμβουλευτεί ένα οποιοδήποτε βιβλίο θεωρητικής επιστήμης υπολογιστών ή ακόμη και τη wikipedia.

αντιγράφει την v' μέσα από την v και να εξομοιώνει την λειτουργία της M' με είσοδο την v'. Αν η M' τερματίζει πάνω στην v', το ζευγάρι (M, v) είναι φτιαγμένο έτσι ώστε να κατασκευάζει ένα αντίγραφο της v. Επομένως το v αναπαράγεται αν και μόνον αν η ακολουθία v' οδηγεί σε τερματισμό την M'. Ωστόσο το πρόβλημα τερματισμού είναι μη αποφασίσιμο, άρα τέτοιο είναι και το πρόβλημα αυτοαναπαραγωγής.

Το θεώρημα αυτό πρακτικά λέει ότι ένα σύστημα anti-virus δεν μπορεί να εγγυηθεί την ανίχνευση όλων των ιών. Αν ανιχνεύσει έναν ιό σ' ένα σύστημα έχει καλώς, αν όμως δεν έχει ανιχνεύσει ιό δεν υπάρχει εγγύηση ότι δεν υπάρχει (ακριβώς όπως συμβαίνει με το πρόβλημα τερματισμού, αν η μηχανή τερματίσει εκτελώντας ένα πρόγραμμα ξέρουμε ότι τερματίζει, αν δεν έχει τερματίσει ως τώρα δεν ξέρουμε αν θα τερματίσει στο μέλλον).

Εξελιξιμότητα

Ένα δεύτερο ερώτημα που μπορεί να απαντηθεί με παρόμοιο τρόπο, αν και φαίνεται εκ πρώτης όψεως πιο εύκολο, είναι κατά πόσο για ένα ζευγάρι ιών v, v' μπορεί να βρεθεί εξέλιξη που οδηγεί από τον ένα στον άλλο ή, με άλλα λόγια, κατά πόσο ανήκουν στο ίδιο ιικό σύνολο. Το ερώτημα έχει και πρακτική αξία, δεδομένου ότι σε περίπτωση καταφατικής απάντησης θα μπορούσαν να ανιχνευθούν ιοί που δεν έχουν μεν ξαναβρεθεί, προέρχονται όμως από την εξέλιξη ήδη γνωστών ιών. Με μια μικρή τροποποίηση της απόδειξης του προηγούμενο θεωρήματος αποδεικνύεται ατυχώς το εξής:

Θεώρημα 4. Το πρόβλημα αν ο ιός v' αποτελεί εξέλιξη ενός ιού v είναι μη αποφασίσιμο.□

Υπολογισιμότητα

Το επόμενο θεώρημα εξηγεί πού μπορεί να καταλήξει η εξέλιξη ενός ιού.

Θεώρημα 5. Οποιαδήποτε ακολουθία μπορεί να υπολογισθεί από μια καθολική μηχανή Turing[9] μπορεί επίσης να προέλθει από την εξέλιξη ενός ιού. □

Το σύνολο των ακολουθιών που μπορούν να προκύψουν από μια καθολική μηχανή Turing έχει πληθικό αριθμό $\aleph_0$ (δηλ. είναι ένα αριθμήσιμο απειροσύνολο), άρα τόσοι είναι και οι ιοί που μπορούν να υπάρξουν. Συνέπεια αυτού είναι πως όταν χρησιμοποιούνται για την ανίχνευση ιών τεχνικές αρίθμησης, όπως είναι η σάρωση, αυτές είναι ανεπαρκείς.

Θέματα πολυπλοκότητας

Ο Cohen ήδη είχε δείξει ότι το πρόβλημα της ανίχνευσης ενός ιού ή της καταγωγής του από άλλον ιό είναι μη αποφασίσιμο. Με το πρόβλημα της πολυπλοκότητας ασχολήθηκε περαιτέρω ο L. Adleman [Adl88]. Νεότερα αποτελέσματα έδωσαν μεγαλύτερη ακρίβεια στην τάξη πολυπλοκότητας που αφορά συγκεκριμένες κατηγορίες ιών. Ο Δ. Σπινέλλης έδειξε το 2003 ότι το πρόβλημα της ανίχνευσης ιών που χρησιμοποιούν κρυπτογραφικές, πολυμορφικές (χρήση ισοδύναμων εντολών) και μεταμορφικές (προσθαφαίρεση, μετάθεση και υποκατάσταση εντολών) τεχνικές είναι NP-complete [Spi03]. Περαιτέρω χαρακτηρισμοί πολυπλοκότητας έχουν δοθεί στις εργασίες των Zuo και Zhou [ZZ04] και Catalin-Valeriu Lita [Lit16].

[9] *Καθολική μηχανή Turing* είναι η μηχανή Turing που μπορεί να προσομοιώσει μια αυθαίρετη μηχανή με αυθαίρετη είσοδο. Αυτό το επιτυγχάνει διαβάζοντας και την περιγραφή της μηχανής που θα που θα προσομοιώσει και την είσοδο.

Δομή του κακόβουλου λογισμικού

Ο E. Filiol εξηγεί ότι για να υπάρξει εισβολή κακόβουλου λογισμικού χρειάζονται οι παρακάτω συνιστώσες ενός υπολογιστικού συστήματος [Fil06]: (α) Μνήμη, όπου το προσβεβλημένο πρόγραμμα θα παραμείνει ανενεργό για μια περίοδο. (β) Προσωρινή μνήμη (RAM) όπου θα εκτελεσθεί, (γ) μια μονάδα επεξεργασίας που θα εκτελέσει τον κώδικα και (δ) ένα λειτουργικό σύστημα.

Περαιτέρω ένας ιός ή ένα σκουλήκι αποτελείται από τις εξής λειτουργικές συνιστώσες:

Ανίχνευση δηλαδή μια λειτουργία με την οποία θα βρει προγράμματα, αρχεία και μηχανές όπου θα διαδοθεί. Η ανίχνευση είναι αποφασιστικής σημασίας για το εύρος της προσβολής και την αποτελεσματικότητα του malware. Η λειτουργία αυτή δεν αρκεί να βρει τον κατάλληλο στόχο, πρέπει να εξασφαλίσει και ότι δεν θα γίνει υπερδιάδοση, δηλαδή ότι δεν θα μολυνθεί πολλές φορές ο ίδιος στόχος. Τυχόν πολλαπλή μόλυνση μπορεί να οδηγήσει σε υπερβολική αύξηση ενός αρχείου-στόχου, το οποίο μπορεί έτσι να αποκαλυφθεί. Μπορεί να αποφευχθεί με μεθόδους *υπογραφής*, π.χ. προσθέτοντας στο προσβεβλημένο αρχείο ένα χαρακτηριστικό string, του οποίου η ύπαρξη ελέγχεται κάθε φορά πριν την επόμενη απόπειρα διάδοσης.

Αντιγραφή που θα δημιουργήσει ένα ακόμη αντίγραφο στο στόχο.

Αντίμετρα ανίχνευσης που θα παρεμποδίσουν την λειτουργία συστημάτων antivirus. Τέτοια μέτρα μπορεί να είναι συμπίεση (για να εμποδίσουν την ανίχνευση λόγω της μεταβολής του μήκους του προσβαλλόμενου αρχείου), κρυπτογράφηση (για να παρεμποδισθούν μέθοδοι υπογραφής), πολυμορφισμός κ.α.

Σκανδάλη δηλαδή μηχανισμός που θα θέσει σε ενέργεια την βλαπτική λειτουργία. Ο μηχανισμός αυτός μπορεί να είναι μια *βόμβα χρόνου* (δηλ. ενεργοποίηση σε συγκεκριμένη στιγμή), ένας μετρητής (π.χ. η εκατοστή φορά που χρησιμοποιηθεί μια εφαρμογή) και γενικά ένας συνδυασμός συνθηκών.

Φορτίο (payload) είναι η ρουτίνα που θα πραγματοποιήσει βλάβες ή άλλες ενέργειες. Από αυτήν εξαρτάται πόσο επιζήμιο είναι το συγκεκριμένο malware.

Κύκλος ζωής

Το κακόβουλο λογισμικό υιοθετεί - εννοείται - τον γενικό κύκλο ζωής του λογισμικού: Αρχική ιδέα, προδιαγραφές, σχεδιασμός, υλοποίηση, εγκατάσταση - διάδοση, αναβάθμιση-συντήρηση, απόσυρση. Ωστόσο ορισμένες φάσεις έχουν διαφορετική έμφαση από το συνηθισμένο λογισμικό. Ένας μεμονωμένος προγραμματιστής που δημιουργεί έναν ιό είναι λιγότερο πιθανό να μπει στον κόπο να καταγράψει προδιαγραφές και μάλλον θα προχωρήσει απ᾽ ευθείας στην υλοποίηση χωρίς καν κάποιο αρχικό σχεδιασμό. Μια οργανωμένη ομάδα ή μυστική υπηρεσία ίσως να κάνει πιο συστηματικό σχεδιασμό που θα περιλάβει και στάδιο προδιαγραφών, ενώ η αρχική ιδέα ίσως να διατυπωθεί σε κάποιου είδους project meeting. Η εγκατάσταση στην περίπτωση ενός ιού έχει πιο ειδική έννοια, ο επιτιθέμενος πιθανώς θα προσβάλει χειροκίνητα έναν υπολογιστή που είναι υπό τον έλεγχό του και θα αφήσει τη συνέχεια στον ίδιο τον ιό.

Ανάμεσα στις φάσεις του κύκλου ζωής του κακόβουλου λογισμικού υπάρχουν και ορισμένες που ίσως αποτελούν έκπληξη για μη εξοικειωμένους με το θέμα αναγνώστες. Ένα τυπικό εμπορικό προϊόν λογισμικού έχει ως αναγκαίο μέρος της φάσης διά-

δοσης την προώθησή του στην κατάλληλη αγορά (marketing), συνήθως με διαφημίσεις. Ωστόσο το ίδιο συμβαίνει και με πολλά προϊόντα ή εργαλεία ή υπηρεσίες στην περιοχή του κακόβουλου λογισμικού, που διαφημίζονται σε μια «μαύρη» αγορά. Για παράδειγμα, όταν κάποιος «στρατολογήσει» και προετοιμάσει τους υπολογιστές ενός botnet κατάλληλου για επιθέσεις κίνησης, είναι πιθανό μετά να προσπαθήσει να πωλήσει υπηρεσίες επιθέσεων σε πιθανούς πελάτες.

Ο εντοπισμός ενεργειών των επιτιθεμένων σε διάφορες φάσεις του κύκλου ζωής του κακόβουλου λογισμικού δεν έχει μόνο θεωρητική χρησιμότητα για την πλευρά της άμυνας, αλλά και πρακτική. Για παράδειγμα ένα botnet που προσφέρεται σε «πελάτες» προς ενοικίαση ή ως υπηρεσία για διεξαγωγή επιθέσεων κίνησης έχει στον κύκλο ζωής του μια φάση διαφήμισης. Ο εντοπισμός τέτοιων διαφημίσεων μπορεί να οδηγήσει σε αποκάλυψη. Ομοίως ο εντοπισμός πληρωμών από τους πελάτες προς τον διαχειριστή του δικτύου. Γενικά η παρεμπόδιση οποιασδήποτε φάσης μπορεί να είναι ευεργετική για την άμυνα [RMG13].

Επίσης, ο κύκλος ζωής μπορεί να διερευνηθεί σε μικρο-φάσεις, δηλαδή να εντοπισθούν αναμενόμενες ακολουθίες ενεργειών εκ μέρους του malware που θα καθοδηγήσουν αντίστοιχα την άμυνα [PM14; KS18].

Κύκλος ζωής ιών και σκουληκιών

Μολονότι ο κύκλος ζωής κακόβουλου λογισμικού που έχει δυνατότητα διάδοσης με τεχνικά μέσα εμπίπτει αναγκαστικά στη γενική περίπτωση, δηλαδή έχει λίγο ως πολύ όλες τις φάσεις που αναφέρθηκαν πιο πάνω, η έμφαση όταν πρόκειται για ιούς και σκουλήκια δίνεται σε όσα συμβαίνουν από την υλοποίηση και πέρα, δηλαδή αφότου αρχίσει η «διανομή», και εξειδικεύεται στον τρόπο που εκδηλώνονται οι μικροφάσεις στην περίοδο της κύριας δράσης:

Προσβολή (infection) είναι η φάση όπου γίνεται η διάδοση του malware σε κατάλληλους στόχους, προγράμματα ή μηχανές. Ιοί και σκουλήκια μετά από μια αρχική εγκατάσταση ψάχνουν κατάλληλους στόχους με διάφορες μεθόδους και εφόσον τους εντοπίσουν προβαίνουν σε τοποθέτηση αντιγράφων τους στους στόχους. Η αρχική εγκατάσταση, πριν αρχίσει η αυτόματη διάδοση, μπορεί να γίνει με οποιοδήποτε μέσο και μέθοδο, περιλαμβανομένης και της κοινωνικής μηχανικής. Ωστόσο έχουν καταγραφεί και ανορθόδοξες περιπτώσεις, όπου η κύρια διάδοση έχει γίνει με «φυσικό» τρόπο από λάθος. Σχετικά παραδείγματα είναι η μετάδοση ιού από φορμαρισμένη δισκέτα, όπου το λογισμικό που έκανε το format ήταν προσβεβλημένο από τον ιό *1099*, ενώ ο μακροϊός *Concept* διαδόθηκε από CD-ROM διανομής[10] εταιρικού προϊόντος [Fil06].

Επώαση (incubation) είναι η φάση στην οποία ο ιός ή το σκουλήκι παραμένει τον περισσότερο χρόνο, δηλαδή ενεδρεύει και κρύβεται μην κάνοντας τίποτε[11] και περιμένοντας την κατάλληλη στιγμή για περαιτέρω ενέργειες. Στη φάση αυτή το κακόβουλο λογισμικό κινδυνεύει να αποκαλυφθεί είτε από το anti-virus είτε από σφάλματα που μπορούν να δημιουργηθούν στην εκτέλεση του προγράμματοσφορέα (αν πρόκειται για ιό). Επομένως ο δημιουργός του λογισμικού παρά την

[10] Η Microsoft διένειμε τον Αύγουστο του 1995 ένα CD με το λογισμικό *Microsoft Windows 95 Software Compatibility Test*. Το CD έφερε τον ιό *Concept*. Ο ιός προσέβαλε το template `normal.dot`, οπότε στη συνέχεια όποιο αρχείο είχε δημιουργηθεί με `save as` κληρονομούσε τον ιό. Ωστόσο ο ιός είχε μηδενικό φορτίο.

[11] Κατά συνέπεια ο όρος *επώαση* (ωρίμανση των αυγών μέσα στα οποία μεγαλώνει ένα έμβρυο) ή (egg) *incubation* δεν είναι ο καταλληλότερος, δεδομένου ότι το κακόβουλο λογισμικό δεν ωριμάζει ή βελτιώνεται σ' αυτό το χρονικό διάστημα, απλώς βρίσκεται σε αναμονή.

ακινησία έχει να αντιμετωπίσει αυξημένες προκλήσεις σε σχέση με την περίοδο επώασης.

Ασθένεια (disease) είναι η φάση όπου ενεργοποιείται το *φορτίο*, δηλαδή το μέρος του κώδικα που υλοποιεί τις βλάβες (ή όποιες άλλες ενέργειες επιθυμεί ο δημιουργός του κώδικα).

5.4 Μη αυτοαναπαραγόμενο λογισμικό

Δούρειος ίππος

Ο δούρειος[12] ίππος (Trojan horse) είναι κακόβουλο λογισμικό που προφανώς παίρνει το όνομά του από το ξύλινο άλογο που χρησιμοποιήθηκε για την άλωση της Τροίας. Ο Filiol δίνει τον εξής ορισμό [Fil06]:

Ορισμός 3. Ο *δούρειος ίππος* είναι ένα απλό πρόγραμμα αποτελούμενο από δύο τμήματα (modules), το τμήμα του πελάτη (client) και το τμήμα του εξυπηρετητή (server). Το δεύτερο αφού εγκατασταθεί στον υπολογιστή του θύματος επιτρέπει στον επιτιθέμενο να αποκτήσει πρόσβαση σε μέρος ή στο σύνολο των πόρων του θύματος (software και hardware). Ο επιτιθέμενος μπορεί να χρησιμοποιήσει τους εν λόγω πόρους μέσω δικτύου χρησιμοποιώντας το τμήμα πελάτη.

Το τμήμα του server εγκαθίσταται αρχικά με εν μέρει χρήση κοινωνικής μηχανικής, δηλαδή περιλαμβάνεται σε ένα ελκυστικό πρόγραμμα που συχνά δίνεται δωρεάν. Το θύμα κατεβάζει το πρόγραμμα χωρίς να γνωρίζει το βλαβερό περιεχόμενο. Με την πρώτη εκτέλεσή του γίνεται η εγκατάσταση του server.

Σε έναν ορισμό πιο λακωνικό από τον παραπάνω:[13]

Ορισμός 4. *Δούρειος ίππος* είναι κακόβουλο λογισμικό που παραπλανεί τους χρήστες για τον πραγματικό του σκοπό.

Ο όρος *δούρειος ίππος* αναφέρεται στην εγκατάσταση κακόβουλου λογισμικού μέσα από ένα επιθυμητό πρόγραμμα «δόλωμα». Στη συνέχεια όμως το εγκατεστημένο κακόβουλο μέρος δεν αποκλείεται να έχει δυνατότητες περαιτέρω διάδοσης, αλλά δεν θα λέγεται πλέον δούρειος ίππος.[14] Ο Adleman [Adl88] (και στη συνέχεια ο Filiol) ονομάζει το μη αυτο-αναπαραγόμενο, πλην όμως βλαπτικό, λογισμικό ως *epeian* (από τον Επειό, κατασκευαστή του Δούρειου Ίππου), κατηγορία στην οποία κατατάσσει προφανώς τους δούρειους ίππους.

Λογικές βόμβες

Ορισμός 5. *Λογική βόμβα* είναι μη αυτο-αναπαραγόμενο κακόβουλο λογισμικό που εγκαθιστά τον εαυτό του σε ένα σύστημα και περιμένει να συμβεί κάποιο γεγονός ή να ικανοποιηθεί κάποια συνθήκη. Μόλις αυτό συμβεί, εκτελεί τις προκαθορισμένες βλαπτικές ενέργειες.

[12] Δούρειος = ξύλινος, αττικός τύπος της λέξης *δουράτεος*, αυτός που φτιάχτηκε από σανίδες ή ξύλινα δοκάρια, μαδέρια (Liddell-Scott, λεξικό της αρχαίας ελληνικής γλώσσας).

[13] `https://en.wikipedia.org/wiki/Trojan_horse_(computing)`

[14] Παρ' όλο που οι Landwehr, Bull κ.α. [Lan+94] προτίμησαν να χωρίσουν τους δούρειους ίππους σε αυτο-αναπαραγόμενους και μη (replicating / non-replicating) και μάλιστα χαρακτηρίζουν διάφορες περιπτώσεις ως *Replicating Trojan horse*, η πλειοψηφία στην κοινότητα της ασφάλειας δέχεται πως ο δούρειος ίππος είναι μη αναπαραγόμενο και διαδιδόμενο λογισμικό. Αν επομένως στη συνέχεια ο δούρειος ίππος εγκαταστήσει αναπαραγόμενο λογισμικό, αυτό θα πρέπει να λέγεται πλέον ιός ή σκουλήκι.

Ο ορισμός αυτός έχει υιοθετηθεί από τον Filiol, ο οποίος προτιμάει να διακρίνει τις λογικές βόμβες από ιούς που χρησιμοποιούν στο φορτίο τους παρόμοια κριτήρια ενεργοποίησης. Χαρακτηριστική περίπτωση είναι αυτή του δυσαρεστημένου υπαλλήλου, ο οποίος φοβάται μήπως απολυθεί και βάζει μια λογική βόμβα, όπως στα επόμενα δύο παραδείγματα.

Το 2004 ένας διαχειριστής του συστήματος της *Medco Health Solutions* στην Montville, New Jersey, φοβούμενος πως θα απολυθεί έβαλε μια λογική βόμβα με την οποία τροποποιούσε τον κώδικα που έτρεχε στους servers της εταιρίας [Vij07]. Η βόμβα ήταν προγραμματισμένη για τη μέρα των γενεθλίων του το 2004, αλλά εξ αιτίας ενός δικού του σφάλματος δεν ενεργοποιήθηκε. Στη συνέχεια την επαναπρογραμμάτισε για ακριβώς ένα χρόνο μετά, παρ' όλο που δεν απολύθηκε. Ωστόσο εξ αιτίας ενός σφάλματος του συστήματος η βόμβα αποκαλύφθηκε και απενεργοποιήθηκε. Η μέγιστη ποινή για τέτοιο αδίκημα είναι $250000 ή δεκαετής φυλάκιση ή και τα δύο. Ο κατηγορούμενος δέχτηκε την ενοχή του. Η τελική ποινή ήταν 30 μήνες φυλακής και αποζημίωση προς την εταιρία ύψους $81200.

Το 2014-16 ένας προγραμματιστής που είχε συμβόλαιο για τη δημιουργία λογισμικού για λογαριασμό της τοπικής Siemens στην Pennsylvania φύτεψε λογικές βόμβες στα προγράμματα που έφτιαξε. Τα προγράμματα ήταν ρυθμισμένα να εμφανίσουν δυσλειτουργίες σε καθορισμένες ημερομηνίες, ώστε να ζητηθούν οι υπηρεσίες του για επιδιόρθωση. Ο κατηγορούμενος παραδέχτηκε την ενοχή του [US 19] και καταδικάστηκε σε έξι μήνες φυλάκισης, δυο χρόνια υπό επιτήρηση και πρόστιμο $7500. Η μέγιστη ποινή ήταν η ίδια όπως παραπάνω.

5.5 Χαρακτηρισμός βάσει σκοπού και ενεργειών

Είδαμε ότι το πάσης φύσεως κακόβουλο λογισμικό μπορεί να χαρακτηρισθεί, να ταξινομηθεί κ.λπ. βάσει των ιδιοτήτων διάδοσης και αναπαραγωγής που το χαρακτηρίζουν, αλλά και πάλι πολλές περιπτώσεις κακόβουλου λογισμικού χρησιμοποιούν ποικίλες μεθόδους διάδοσης. Ταυτόχρονα οι ζημιές που θα προκληθούν επαφίενται στη φαντασία και στην ικανότητα ενός κακόβουλου προγραμματιστή, γεγονός που κάνει μια ταξινόμηση ως προς τις ζημιές δύσκολη. Παρ' όλα αυτά μερικές χαρακτηριστικές περιπτώσεις αξίζουν αναφοράς. Και πάλι υπάρχουν κακόβουλα προγράμματα σχετικά γενικού σκοπού, ικανά να προκαλέσουν ένα ευρύ φάσμα ζημιών.

Bots & botnets

Το botnet είναι ένα σύνολο υπολογιστών προσβεβλημένων από κακόβουλο λογισμικό προκειμένου να χρησιμοποιηθούν για βλαπτικές ενέργειες, οι οποίες υλοποιούνται συντονισμένα σύμφωνα με εντολές από ένα κέντρο ελέγχου. Πολλά botnets αποτελούν ένα γενικό εργαλείο και οι ενέργειές τους εξειδικεύονται από τις εντολές που παίρνουν κάθε φορά.

Στο λεξιλόγιο του *Ευρωπαϊκού Οργανισμού για την Κυβερνοασφάλεια*[15] δίνεται ο εξής ορισμός:

Ορισμός 6. Το *bot* είναι κακόβουλο λογισμικό που δέχεται εντολές από ένα διαχειριστή (master). *Botnet* είναι ένα σύνολο υπολογιστών που έχουν μολυνθεί από bots.

Γενικά ο όρος *bot* έχει χρησιμοποιηθεί κατά διαφόρους τρόπους. Ένα *Internet bot* ή *web robot* ή απλώς *bot* είναι μια εφαρμογή που διαθέτει κάποιο αυτοματισμό, δηλαδή

[15] `https://www.enisa.europa.eu/topics/csirts-in-europe/glossary`

εκτελεί μια σειρά από εργασίες και πιθανώς μπορεί να ακολουθήσει διαφορετικές πορείες εξέλιξης. Τα chatbots (bots συζήτησης) είναι προγράμματα που έχουν την ικανότητα να κάνουν μια στοιχειώδη συζήτηση με ανθρώπους σε φυσική γλώσσα. Το πρώτο τέτοιο πρόγραμμα ήταν η Eliza[16] του Joseph Weizenbaum (1966) [Wei93]. Ορισμένα bots επικοινωνούσαν με χρήστες χρησιμοποιώντας ανταλλαγής αμέσων μηνυμάτων (instant messengers) και κανάλια IRC (Internet Relay Chat) και αργότερα μέσω κοινωνικών δικτύων (Facebook, Twitter, Instagram κ.α.).

Ένα κακόβουλο bot εκτελεί μια σειρά ενεργειών, περισσότερο ή λιγότερο αυτοματοποιημένων, αφού όμως δεχτεί κατάλληλες εντολές από ένα κέντρο εντολών (command and control center - C&C). Η αρχική σχέση των bots με τα κανάλια IRC αξιοποιήθηκε και από τους δημιουργούς κακόβουλων bots, οι οποίοι έδιναν εντολές μέσα από τέτοια κανάλια. Οι εντολές όμως μπορούν να μεταφέρονται με οποιονδήποτε τρόπο δικτυακής επικοινωνίας.

Ο δημιουργός ενός botnet προσπαθεί να εγκαταστήσει κρυφά τα bots σε μηχανές που δεν είναι στην ιδιοκτησία του. Η εγκατάσταση μπορεί να γίνει με οποιαδήποτε μέθοδο θεωρήσει πρόσφορη, περιλαμβανομένων ιών, σκουληκιών και δουρείων ίππων. Το μέγεθος ενος botnet μπορεί να φτάσει τα εκατομμύρια προσβεβλημένων μηχανών. Το Rustock botnet (2006-2011) είχε από 150000 ως 2400000 μηχανές [Kir10]. Οι εκτιμήσεις για το Mirai IoT Botnet (2016) βρίσκονταν μεταξύ 800000 και 2500000. Η προσβολή μεγάλων αριθμών μηχανών είναι δυνατή ακόμη και σε περιορισμένο διάστημα, π.χ. το Satori botnet μάζεψε 280000 μηχανές σε 12 ώρες [Lal18].

Ο Ορισμός 6 επομένως είναι καλό να ερμηνεύεται στο κατάλληλο context ακόμη και εσωτερικά στην περιοχή της ασφάλειας, δεδομένου ότι διαχείριση κακόβουλου λογισμικού με χρήση εντολών από ένα διαχειριστή γίνεται και σε άλλες περιπτώσεις, όπως π.χ. σε ένα rootkit. Στην περίπτωση του botnet η έμφαση βρίσκεται στο ότι η διαχείριση γίνεται μαζικά. Δεδομένου ότι ένα botnet μπορεί να αποτελείται από μεγάλο πληθυσμό υπολογιστών (γεγονός που δεν αποτυπώνεται στον παραπάνω ορισμό), κάθε εντολή μπορεί να αφήνει σημαντικό αποτύπωμα κίνησης στο δίκτυο. Αν η διαχείριση πρόκειται να γίνει πιο διακριτικά, χρειάζονται τα κατάλληλα μέτρα, όπως ετεροχρονισμός των εντολών ή χρήση επιπέδων ιεραρχίας [Oll09]. Επομένως οι προκλήσεις τόσο για την επίθεση, όσο και για την άμυνα, είναι διαφορετικές στην περίπτωση ενός botnet από ένα rootkit.

Ο διαχειριστής ενός botnet αντί να επικοινωνεί απ᾽ ευθείας με τα bots είναι πιθανό να εγκαταστήσει ενδιάμεσους κόμβους επικοινωνίας σε προσβεβλημένους υπολογιστές, δηλαδή να χρησιμοποιήσει ένα τουλάχιστον στρώμα ιεραρχίας ανάμεσα σ᾽ αυτόν και στα τελικά bots. Η χρήση ιεραρχικής τοπολογίας στο δίκτυο επικοινωνίας αραιώνει την κίνηση που φεύγει από τον κεντρικό διαχειριστή (ή και καταλήγει σ᾽ αυτόν αν γίνεται συλλογή πληροφοριών), γεγονός που αφενός τον απαλλάσσει από φόρτο, αφετέρου κάνει την κίνηση αυτή λιγότερο ανιχνεύσιμη από συστήματα άμυνας.

Επικοινωνία σε botnet

Το πρόβλημα που καλείται να επιλύσει ο διαχειριστής ενός botnet είναι πώς θα δώσει εντολές στα bots χωρίς η επικοινωνία να ανιχνευθεί. Οι εντολές σε ορισμένες περιπτώσεις μπορεί να είναι για άμεσες συγχρονισμένες ενέργειες (π.χ. να εκτελέσουν μια επίθεση κίνησης), οπότε οι εντολές πρέπει να φύγουν ταυτόχρονα από το κέντρο διαχείρισης (command and control center), να φτάσουν γρήγορα στα bots και να εκτε-

[16] https://en.wikipedia.org/wiki/ELIZA

λεστούν άμεσα. Σε άλλες περιπτώσεις αρκεί να φτάσουν πριν από ένα χρονικό σημείο και να περιλαμβάνουν οδηγίες για την ώρα εκτέλεσης. Η αποτελεσματικότητα όμως αυτής της λύσης εξαρτάται από τα ρολόγια των μηχανών που φιλοξενούν τα bots. Μηχανές «χαμηλής νοημοσύνης» (τηλεοράσεις, streamers, συστήματα αισθητήρων κ.λπ.) είναι πιθανό να συνδέονται με το internet περιστασιακά και τα ρολόγια τους να έχουν απορρυθμιστεί.

Πιθανοί τρόποι προσβολής και επικοινωνίας είναι οι εξής:

telnet Πολλές «έξυπνες» συσκευές επικοινωνούν με ένα telnet server. Ο επιτιθέμενες σκανάρει τις εισόδους ενός τέτοιου server και όταν βρει μια ανοιχτή θύρα μπορεί να δει με ποιες συσκευές επικοινωνεί και κατά πόσο χρειάζεται αυθεντικοποίηση. Μπορεί να δοκιμάσει διάφορα default login name / passwords (admin/admin, root/root κ.α.) και να μολύνει τον server, ώστε στη συνέχεια να ελέγξει τις συσκευές που επικοινωνούν μαζί του.

IRC Το *Internet Relay Chat* protocol σχεδιάσθηκε για να εξυπηρετήσει τηλεσυνεδριάσεις που γίνονται με χρήση γραπτών μηνυμάτων (βλ. και RFC 1459). Χρησιμοποιεί το μοντέλο client-server πάνω από δίκτυα TCP/IP. Μια ομάδα από clients επικοινωνεί χρησιμοποιώντας το ίδιο κανάλι. Το κέντρο ελέγχου του botnet μπορεί να δώσει εντολές μέσα από το κανάλι και μπορεί να αλλάζει κανάλια μαζί με τα bots προκειμένου να αποφύγει την ανίχνευση. Ένα σύστημα άμυνας μπορεί να κλείσει τα σχετικά κανάλια ή να απαγορεύσει κάποιες λέξεις που αποτελούν εντολές. Ένα πιο προχωρημένο σύστημα καταστολής των botnets μπορεί να μιμηθεί το σύστημα διαχείρισης και ελέγχου.

Δίκτυα ισοτίμων (peer to peer, P2P) Σε ένα τέτοιο δίκτυο η προσβολή είναι σχετικά εύκολη επειδή το περιεχόμενο ενός αρχείου δεν ελέγχεται κεντρικά. Εφόσον ο επιτιθέμενος γίνει μέλος του δικτύου ή μπορέσει να προσβάλει ένα μέλος του δικτύου, μπορεί να μεταδώσει μολυσμένα αρχεία σε άλλα μέλη [WAZ10; YC17].

http Το botnet χρησιμοποιεί το πρωτόκολλο http για την επικοινωνία μεταξύ C&C server και bots. Ο C&C server λειτουργεί όπως ένας κανονικός Web server και τα bots όπως κανονικοί Web clients. Γνωστά τέτοια botnets είναι το *Spyeye* και το *Zeus* [Hsu+17].

Adware & malvertising

Στο Oxford English Dictionary η λέξη *adware* ερμηνεύεται ως εξής: *Λογισμικό που δείχνει ή κατεβάζει αυτόματα διαφημιστικό υλικό όταν ο χρήστης είναι online (software that automatically displays or downloads advertising material such as banners or pop-ups when a user is online)*.

Συχνά ως *adware* χαρακτηρίζεται γενικότερα οποιαδήποτε εφαρμογή ή υπηρεσία προσφέρεται «δωρεάν», δηλαδή χωρίς συνδρομή ενώ βασίζεται σε διαφημίσεις για να δημιουργήσει έσοδα, χρεώνοντας τους διαφημιζόμενους (σε αντιπαραβολή με άλλες περιπτώσεις, όπως shareware, freeware κ.α.). Το ίδιο το Facebook αποτελεί μια τέτοια περίπτωση, χωρίς μάλιστα να προσφέρει εναλλακτική επιλογή, όπως να δίνεται στους χρήστες η δυνατότητα να επιλέξουν ανάμεσα σε μια συνδρομή και στη «δωρεάν» χρήση που επιφέρει διαφημίσεις. Στο «δωρεάν» Spotify η επιθυμητή μουσική διακόπτεται για να ακούσει ο χρήστης διαφημίσεις, αλλά μπορεί να τις αποφύγει (σε συνδυασμό με καλύτερη ποιότητα υπηρεσίας) αν αποδεχτεί μια μηνιαία συνδρομή. Σε άλλες περιπτώσεις προσφέρεται η επιλογή ανάμεσα σε περισσότερο και

λιγότερο «στοχευμένες διαφημίσεις», οι οποίες αντιστοιχούν σε λιγότερη παρακολούθηση. Κατά πόσο οι πιο στοχευμένες διαφημίσεις αποτελούν κέρδος για τον πελάτη είναι ένα ζήτημα που μπαίνει κατά καιρούς σε συζήτηση. Ωστόσο ακόμη και όταν προσφέρονται στον χρήστη επιλογές, αυτές σπάνια είναι ξεκάθαρες, π.χ. δεν παρουσιάζεται στον χρήστη μια προδιαγραφή χρόνου, δηλαδή ότι οι διαφημίσεις δεν θα υπερβαίνουν ένα συγκεκριμένο ποσοστό του χρόνου της κανονικής υπηρεσίας.

Ο αριθμός των διαφημίσεων και ο τρόπος με τον οποίο παρουσιάζονται ώστε να αποσπάσουν την προσοχή ποικίλλουν. Σε ορισμένες περιπτώσεις αποτελούν για τον χρήστη ένα λογικό και αποδεκτό συμβιβασμό για την παρεχόμενη υπηρεσία, στο άλλο άκρο όμως μπορεί να υπάρξει ένας καταιγισμός διαφημίσεων όπου ο χρήστης ποτέ δεν φτάνει στην ίδια την υπηρεσία που χρησιμοποιείται μόνο ως δόλωμα. Οι σύγχρονοι browsers διαθέτουν επεκτάσεις που αναχαιτίζουν την εμφάνιση διαφημίσεων (pop up window blockers), ενώ υπάρχει μια ευρύτερη αγορά λογισμικού που παρεμποδίζει τις διαφημίσεις.[17]

Στο ερώτημα κατά πόσο η υπερβολική προβολή διαφημίσεων μπορεί να θεωρηθεί κακόβουλη ενέργεια οι γνώμες διίστανται [Gao+19], πρόκειται δηλαδή για ζήτημα που πέφτει σε μια γκρίζα περιοχή, επειδή το αποτέλεσμα χαρακτηρίζεται περισσότερο ως ενόχληση και λιγότερο ως (σημαντική) ζημιά. Υπάρχει όμως και η άλλη πλευρά με τα δικά της επιχειρήματα: Η πλευρά των διαφημιζομένων και των διαφημιστών διαμαρτύρεται κατά της υπερβολικής καταπίεσης των διαφημίσεων και κατά της χρήσης αμυντικού λογισμικού με το επιχείρημα ότι κατά την παροχή μιας «δωρεάν» υπηρεσίας είναι λογικό να παρουσιάζονται κάποιες διαφημίσεις, διαφορετικά παραβλάπτονται τα δικαιώματα των παρόχων της υπηρεσίας και των διαφημιζομένων.

Το adware μπορεί να εγκαθίσταται με την άδεια του χρήστη ή χωρίς αυτήν. Στη δεύτερη περίπτωση βρίσκεται πιο κοντά στο να χαρακτηρισθεί malware και μπορεί να ανιχνεύεται και να αφαιρείται από συστήματα antivirus.

Μια από τις συνήθεις πρόσθετες λειτουργίες του adware είναι η παρακολούθηση της συμπεριφοράς του πιθανού πελάτη και καταναλωτή. Το βάθος αυτής της παρακολούθησης και η διάπραξη ενεργειών που προσβάλλουν την ιδιωτικότητα κατά περίπτωση διαφέρουν, αλλά από ένα σημείο και μετά το σχετικό λογισμικό είθισται να χαρακτηρίζεται μάλλον ως *spyware* παρά ως adware [Ayc10; Urb+18].

Spyware

Ο όρος είναι γνωστός από το 1994. Ο John Aycock [Ayc10] δίνει τον εξής κατάλογο για ενέργειες που αντιστοιχούν σε spyware: Καταγραφή πληκτρολόγησης, κινήσεων ποντικιού, καταγραφή οθόνης, καταγραφή της δραστηριότητας του χρήστη μέσω κάμερας και μικροφώνου, κλοπή κλειδιών αδειοδότησης εγκατεστημένου λογισμικού, κλοπή αρχείων, παρακολούθηση του σερφαρίσματος, αλλαγή ρυθμίσεων για διευκόλυνση υποκλοπής, κρυφή εγκατάσταση λογισμικού, μέτρα αποφυγής απεγκατάστασης του παράνομου λογισμικού. Οι παραπάνω καταγραφές αφενός γίνονται χωρίς τη συναίνεση του χρήστη, αφετέρου μπορούν να οδηγήσουν στη συλλογή πληθώρας πληροφοριών που προσδιορίζουν με σαφήνεια τις απόψεις του για διάφορα θέματα και σαφώς παραβιάζουν την ιδιωτικότητά του [Urb+18].

Σε μερικές περιπτώσεις μπορεί να θεωρηθεί ότι το spyware εγκαθίσταται από την πλευρά της άμυνας, όταν π.χ. το λογισμικό ενός ηλεκτρονικού αναγνώστη (e-reader) εντοπίσει ένα βιβλίο που είναι προϊόν παράνομης αντιγραφής και ειδοποιήσει τον νόμιμο προμηθευτή. Η εγκατάσταση ενός τέτοιου προγράμματος χωρίς τη γνώση του

[17] `https://en.wikipedia.org/wiki/List_of_pop-up_blocking_software`

κατόχου του ηλεκτρονικού αναγνώστη μπορεί να είναι ή να μην είναι νόμιμη, ανάλογα με το νομικό πλαίσιο.

Τα δεδομένα που συνδέονται από spyware κατά περίπτωση είτε χρησιμοποιούνται αμέσως είτε καταλήγουν σε «αποθετήρια» για να πωληθούν σε ειδικές αγορές [HEF09].

Μερικές αντιπροσωπευτικές κατηγορίες spyware είναι οι εξής:

HTTP cookies

Είναι μικρά αρχεία κειμένου που αποθηκεύονται στην πλευρά του χρήστη όταν αυτός επισκέπτεται μια ιστοσελίδα. Ορισμένα διατηρούνται μόνο όσο κρατάει η συγκεκριμένη επικοινωνία (*session cookies*) και διαγράφονται όταν κλείσει ο browser. Πληροφορία που είναι χρήσιμη και στην επόμενη επικοινωνία, όπως παράμετροι της συσκευής, ρυθμίσεις που έχει κάνει ο χρήστης, passwords κ.α. αποθηκεύονται σε ένα *επίμονο* (persistent) cookie, το οποίο φέρει μια ημερομηνία λήξης. Επιτρέπουν την αναγνώριση βασικών ιδιοτήτων και προτιμήσεων του χρήστη όταν επιστρέφει στην ίδια ιστοσελίδα. Ωστόσο μπορούν να καταγράφουν περαιτέρω πληροφορίες, όπως π.χ. πού έχει κάνει κλικ ο χρήστης, πόσες φορές, τι ώρα κ.λπ. προκειμένου να βγουν συμπεράσματα για τη συμπεριφορά του (βλ. π.χ. το *Google analytics cookie*[18], το οποίο μεταξύ άλλων μπορεί να χρησιμοποιηθεί για την προσαρμογή των διαφημίσεων στις προτιμήσεις του χρήστη). Γι' αυτό λέγονται και cookies παρακολούθησης (tracking cookies).[19]

Όταν ένας πελάτης έρχεται σε επαφή με ένα site θεωρείται λογικό η εταιρία που βρίσκεται πίσω από το site να συλλέγει ορισμένες απαραίτητες πληροφορίες για τον χρήστη, όπως εξηγήθηκε παραπάνω, και να τις αποθηκεύει σε ένα cookie. Το cookie αυτό φέρει την ένδειξη του site του οποίου η διεύθυνση φαίνεται στον browser. Αυτό είναι ένα *first-party cookie*.

Ας υποτεθεί τώρα ότι ένας χρήστης βλέπει ένα site όπου υπάρχουν διαφημίσεις. Όταν κάνει κλικ πάνω σε μια διαφήμιση την οποία παρέχει η διαφημιστική εταιρία Δ παίρνει ένα cookie με την ένδειξη της Δ, αλλά και την μοναδική ταυτότητα του browser (ένα τυχαίο αριθμό μοναδικό για τον συγκεκριμένο υπολογιστή και τον browser) [Ame04]. Το cookie αυτό φέρει την ένδειξη της προέλευσης του περιεχομένου και όχι του τρέχοντος site και λέγεται *third-party cookie*. Στη συνέχεια ο ίδιος χρήστης σερφάροντας με τον ίδιο browser φτάνει σε ένα άλλο site και πέφτοντας πάνω μια άλλη διαφήμιση που δίνει η Δ παίρνει ένα δεύτερο cookie με την Δ. Σε κάποια από τις επισκέψεις σε διαφημίσεις της Δ θα επιστρέψουν στην Δ όλα τα δικά της cookies. Τώρα η Δ μπορεί να συσχετίσει όλα τα cookies και να καταλάβει τι είδους διαφημίσεις έβλεπε ο χρήστης που χρησιμοποιούσε τον συγκεκριμένο browser. Ένα site επιτρέποντας να τοποθετούνται third party cookies για λογαριασμό «φιλικών» εταιριών) διευκολύνει τη συλλογή πληροφοριών από τις εταιρίες αυτές. Σήμερα τα third party cookies αποκλείονται από τους browsers Firefox, Safari κ.α. Σύμφωνα με την GDPR (General Data Protection Regulation 2016/679) τα cookies μπορεί να οδηγήσουν στον προσδιορισμό ενός προσώπου, άρα είναι προσωπικά δεδομένα, άρα πρέπει να τοποθετούνται στον υπολογιστή ενός χρήστη μόνον αν έχει προηγουμένως ληφθεί η άδειά του.

[18] `https://developers.google.com/analytics/devguides/collection/analyticsjs/cookie-usage`

[19] Σε ένα άλλο παράδειγμα, στην παρακάτω ιστοσελίδα της *enisa* (European Union Agencey for Cybersecurity) είναι γραμμένος υποδειγματικά ο τρόπος με τον οποίο η ίδια η *enisa* χρησιμοποιεί cookies στις δικές της ιστοσελίδες και τι συμβαίνει ότι κάποιος παρακολουθεί την *enisa* σε κοινωνικά δίκτυα. `https://www.enisa.europa.eu/about-enisa/cookies`

Καταγραφείς πληκτρολόγησης - Keyloggers

Οι *καταγραφείς πληκτρολόγησης (keyloggers)* είναι κακόβουλο λογισμικό που καταγράφει τα πλήκτρα που πατάει ένας χρήστης σε πληκτρολόγιο υπολογιστή, τις κινήσεις που κάνει στο ποντίκι, καθώς και άλλες ενέργειες του χρήστη στη διεπαφή (interface) ανθρώπου-μηχανής. Το προϊόν της καταγραφής μπορεί να είναι χαρακτήρες, μπορεί να είναι όμως και πολυμεσικό περιεχόμενο, π.χ. εικόνες από την περιοχή της οθόνης γύρω από το ποντίκι προκειμένου να καταγραφούν τα κλικ πάνω σε εικονίδια ή σε εικονικά πληκτρολόγια.

Ο σκοπός της καταγραφής είναι η συλλογή πρωτογενούς πληροφορίας πριν ακόμη αυτή μετασχηματισθεί και προστατευθεί με κρυπτογραφικό ή άλλο τρόπο. Για παράδειγμα, όταν ένας χρήστης δίνει ένα password, η διεργασία που το διαβάζει είτε το χρησιμοποιεί[20] και το σβήνει, είτε το κρυπτογραφεί, οπότε ο επιτιθέμενος παρεμποδίζεται να το διαβάσει. Όταν ένας πελάτης τράπεζας επικοινωνεί με μια τραπεζική εφαρμογή online, η πληροφορία που ανταλλάσσεται και προς τις δύο κατευθύνσεις είναι κρυπτογραφημένη. Οποιαδήποτε υποκλοπή είναι πιο εύκολο να γίνει στη διεπαφή ανάμεσα στον πελάτη και τον υπολογιστή (human-machine interface – HMI).

Επί πλέον το HMI αποτελεί ένα ελκυστικό σημείο υποκλοπής, διότι είναι σχετικά εύκολο να προσβληθεί και δύσκολο να προστατευθεί, ενώ η συλλεγόμενη πληροφορία είναι συχνά αναγνώσιμη από τον άνθρωπο και μάλλον συμπυκνωμένη σε ευαίσθητα δεδομένα. Πρέπει όμως να γίνει επεξεργασία του συλλεγόμενου ρεύματος, είτε αυτόματη είτε από άνθρωπο.

Η καταγραφή μέσω λογισμικού μπορεί να γίνει με διάφορους τρόπους. Ενδεικτικά μπορεί να γίνει ως εξής:

- Να γίνει από λογισμικό που έχει πρόσβαση στο λειτουργικό σύστημα ή ενσωματώνεται στον πυρήνα του, ώστε να καταγράφει τις κινήσεις που γίνονται στο πληκτρολόγιο [Eli+16]. Ένας τέτοιος καταγραφέας μπορεί να έχει τη μορφή ενός keyboard driver.
- Να ενσωματωθεί σε μια εφαρμογή, καταγράφοντας την είσοδο της εφαρμογής.
- Να γίνει αποσπώντας τις πληροφορίες από μια φόρμα HTML όταν την έχει συμπληρώσει και την υποβάλλει ένας χρήστης.
- Να τροποποιήσει ιστοσελίδες είτε αλλάζοντάς τους το περιεχόμενο είτε εισάγοντας νέες αλληλεπιδράσεις (transactions), εκτελώντας έτσι μια επίθεση τύπου man-in-the middle (που εν προκειμένω λέγεται *Man in the Browser* - MitB - attack). Για παράδειγμα, αν ένας πελάτης μεταφέρει ένα ποσό μεταξύ λογαριασμών, το παρείσακτο λογισμικό του δείχνει τη μεταφορά του σωστού ποσού, ωστόσο προς την τράπεζα μπορεί να έχει δώσει εντολή μεταφοράς άλλου ποσού και προς άλλο λογαριασμό. Μια τέτοια επίθεση δεν αντιμετωπίζεται ούτε με αυθεντικοποίηση τριών παραγοντων (three factor authentication), π.χ. με το γνωστό SMS στο κινητό του πελάτη.

Κάποιες από τις παραπάνω διεργασίες υποκλοπής μπορούν να υλοποιηθούν και με hardware, π.χ. εισάγοντας ειδικό hardware σε περιφερειακά όπως πληκτρολόγιο ή ποντίκι, ή υποκλέπτοντας τα σήματα που περνούν μέσα από το καλώδιο με το οποίο συνδέονται τα περιφερειακά. Ακόμη μπορεί να γίνει ακουστική ή οπτική καταγραφή

[20] Π.χ. υπολογίζει το hash code του password που δίνει ο χρήστης, συγκρίνει τον κωδικό με υπάρχοντες κωδικούς και επιτρέπει ή όχι την πρόσβαση. Στη συνέχεια το δοσμένο password διαγράφεται για να μην αποτελέσει αντικείμενο υποκλοπής.

του πληκτρολογίου. Η ακουστική καταγραφή βασίζεται στο ότι το πάτημα κάθε πλήκτρου παράγει (έστω και ελάχιστα) διαφορετικό ήχο, οπότε η ακολουθία ήχων καταγράφεται και αναλύεται [AA04]. Αυτό σημαίνει ότι η καταγραφή είναι δυνατή και εφόσον ένας υπολογιστής μολυνθεί στο λογμισμικό που ελέγχει το μικρόφωνο. Οπτική καταγραφή μπορεί να γίνει π.χ. βάζοντας μια μικροκάμερα που βλέπει το πληκτρολόγιο ενός ATM. Μπορεί επίσης να γίνει καταγραφή των ασθενών ηλεκτρομαγνητικών κυμάτων που εκπέμπονται από ένα πληκτρολόγιο κατά τη διάρκεια της χρήσης του [VP09].

Spam & Phishing

Σύμφωνα με το το New Oxford Dictionary of English (από την έκδοση του 1998 και μετά) η λέξη *spam* έχει την έννοια των *ασχέτων ή μη αποδεκτών μηνυμάτων που στέλνονται στο Ίντερνετ σε μεγάλο αριθμό ομάδων ειδήσεων ή χρηστών*.

Το απλό, διαφημιστικό κυρίως, spam μπορεί να είναι μια ενόχληση για τον μέσο χρήστη email, αλλά για μια εταιρία αποτελεί πηγή οικονομικής ζημιάς. Μια εταιρία παροχής υπηρεσιών νέφους για τον έλεγχο του spam αρχίζει τη διαφήμισή της ως εξής: *Ο μέσος εργαζόμενος χρειάζεται 4 sec για να αναγνωρίσει και να σβήσει ένα spam email. Για έναν οργανισμό με 1000 εργαζόμενους, καθένας από τους οποίους παίρνει 5 μηνύματα spam την ημέρα η απώλεια είναι ίση με 219 ανθρωπομέρες κατ' έτος.*[21] Η ζημιά αυτή αρχίζει πριν ακόμη υπολογισθεί το κόστος από κακόβουλα μηνύματα.

Το *SPAM* ήταν μια αεροστεγώς κλεισμένη κονσέρβα με κρέας, κατά κύριο λόγο χοιρινό, προϊόν της αμερικανικής εταιρίας Hormel. Παρόμοια προϊόντα αναπτύχθηκαν και από άλλες εταιρίες, εντός και εκτός ΗΠΑ, π.χ. την Ολλανδική Zwanenberg (γνωστή από το ZWAN). Το κρέας μπορούσε να διατηρηθεί εκτός ψυγείου στην κλειστή κονσέρβα για χρόνια. Η τεχνική της εν λόγω κρέατος διατήρησης επινοήθηκε τον καιρό της παγκόσμιας οικονομικής ύφεσης (1929-1939), διαδόθηκε κατά τον 2ο παγκόσμιο πόλεμο στην Ευρώπη από τα αμερικανικά στρατεύματα και από αποστολές τροφίμων και έμεινε δημοφιλές μεταπολεμικά. Το 1970 οι Monty Python στο τηλεοπτικό show *Monty Python Flying Circus* παρουσίασαν ένα κωμικό σκετς με ένα ζευγάρι που προσπαθεί να παραγγείλει σε ένα καφέ, αλλά όλα τα διαθέσιμα πιάτα περιέχουν spam. Τον διάλογο συνόδευε ένα τραγουδάκι, όπου επαναλαμβάνεται πολλές φορές η λέξη spam. Αυτό το επεισόδιο με τον καταιγισμό του spam θεωρείται ως ετυμολογική καταβολή του όρου *spam*.[22]

Το κύριο μέσο αποστολής μηνυμάτων spam παραμένει το email, παρ' όλο που σήμερα η τεχνική έχει επεκταθεί και σε άλλες πλατφόρμες αποστολής μηνυμάτων, όπως SMS, Twitter, Viber, Facebook messenger κ.α. Η έννοια του spam προϋποθέτει την αποστολή σε πολλαπλούς αποδέκτες. Στο email αυτό είναι εύκολο τοποθετώντας πολλές διευθύνσεις παραλήπτη στο ίδιο μήνυμα ή μια ομαδική διεύθυνση.

Ο σκοπός της αποστολής μηνυμάτων spam παραμένει κυρίως διαφημιστικός. Σε ένα μικρότερο ποσοστό περιλαμβάνονται μηνύματα με σκοπό την παραπλάνηση του παραλήπτη ώστε να διευκολύνει την εγκατάσταση κακόβουλου λογισμικού στον υπολογιστή του ή να παρασυρθεί σε ιστοσελίδες με ύποπτο περιεχόμενο. Τα μηνύματα αυτά χαρακτηρίζονται ως *phishing*. Ο όρος αυτός προέρχεται από το fishing (ψάρεμα).

Το phishing ως απάτη βασίζεται ελάχιστα στην τεχνολογία. Κατά κύριο λόγο βασίζεται στην κοινωνική μηχανική [SW11] και, όπως γενικότερα το spam, στο πολύ μικρό κόστος αποστολής μηνυμάτων email. Οποιοδήποτε αφήγημα μπορεί να παρα-

[21] `https://www.spamtitan.com/hosted-spam-filter/`.

[22] `https://en.wikipedia.org/wiki/Spam_(Monty_Python)`.

σύρει εύπιστα θύματα σε ενέργειες που θα αποβούν προς όφελος του επιτιθέμενου ενσωματώνεται σε ένα μεγάλο αριθμό μηνυμάτων. Ένα από τα πρώτα σοβαρά επεισόδια phishing ήταν αυτό της AOL (America on Line) το 1995. Οι επιτιθέμενοι επικοινωνούσαν με τους πελάτης της AOL υποδυόμενοι υπαλλήλους και ζητούσαν κωδικούς και αριθμούς τραπεζικών λογαριασμών υποστηρίζοντας ότι υπάρχει κάποιο πρόβλημα με τον λογαριασμό τους.

Μια δημοφιλής κατηγορία είναι μηνύματα για *προκαταβολή τελών* (advance-fee scam) και περιέχουν συχνά ένα γελοίο σενάριο. Το λεγόμενο *γράμμα από τη Νιγηρία* ή *απάτη του Νιγηριανού πρίγκιπα* (Nigerian prince scam) είναι ένα μήνυμα που φτάνει ξαφνικά στον παραλήπτη και περιέχει ένα αφήγημα ότι θα λάβει ένα μεγάλο ποσό εφόσον διευκολύνει το διαχειριστή ή τον κάτοχο του ποσού στέλνοντάς του ένα «μικρό» ποσό για διαδικαστικά έξοδα [Smi09; Leo19; CR15]. Ή είναι ένα γράμμα από ένα πρίγκιπα που έχει προβλήματα με το καθεστώς της χώρας του και προσπαθεί να βγάλει τα χρήματά του στο εξωτερικό. Θα μπορούσε να είναι ένα γράμμα για κληρονομιά από θείο από την Αμερική, δηλαδή ένα γράμμα που στέλνει ένας συμβολαιογράφος που διαχειρίζεται την διαθήκη και την κληρονομιά από τον αποθανόντα θείου. Ο αποστολέας ισχυρίζεται ότι ο παραλήπτης είναι μεταξύ των κληρονόμων και ότι χρειάζεται ένα ποσό για να ξεκινήσει τις σχετικές διαδικασίες. Σε άλλο παράδειγμα, το μήνυμα λέει ότι ο παραλήπτης δικαιούται να εισπράξει μια επιταγή. Το 2008 μια αμερικανίδα καταδικάσθηκε σε 2 έτη φυλάκισης για τέτοιο αδίκημα, το οποίο περιελάμβανε και χρήση πλαστών επιταγών [Goh08].

Το phishing προσφέρει μεν μια αρχική επαφή με τα πιθανά θύματα, όμως στα τελευταία στάδια της απάτης μπορεί να χρειάζεται σε δεύτερη φάση εκ μέρους του επιτιθέμενου περαιτέρω προσπάθεια για να υποκύψει το θύμα και να προκύψει κέρδος. Οι δυνατότητες του επιτιθέμενου για χειρισμούς δεύτερης φάσης, όπου πρέπει να γίνει προσωποποιημένη επικοινωνία με το κάθε συγκεκριμένο πιθανό θύμα, είναι περιορισμένες. Μια από τις εντυπωσιακές ιδιότητες αυτού του τύπου απάτης είναι ότι φαίνεται πολύ χοντροκομμένη για να έχει πιθανότητα επιτυχίας. Μια πιο προσεκτική ανάλυση δείχνει ακριβώς ότι η στρατηγική της αποστολής σχετικά απίστευτων μηνυμάτων συμβάλλει στην επιλογή των πιο εύπιστων θυμάτων. Με τον τρόπο αυτόν ο επιτιθέμενος μειώνει τις ψευδώς θετικές περιπτώσεις (false positives), δηλαδή τις περιπτώσεις θυμάτων που θα δείξουν αρχικό ενδιαφέρον, αλλά τελικά δεν θα εξαπατηθούν. Την απώλεια σε αριθμό ανταποκρινομένων στο αρχικό μήνυμα μπορεί ο επιτιθέμενος να την αντισταθμίσει αυξάνοντας το πλήθος των αποστελλόμενων μηνυμάτων phishing [Her12].

Την αντίθετη προσέγγιση ακολουθούν οι επιτιθέμενοι με τη μέθοδο *spear phishing* (spearfishing είναι το ψάρεμα με καμάκι ή ψαροντούφεκο - speargun). Εδώ η επίθεση είναι στοχευμένη, δηλαδή το μήνυμα φτάνει σε λίγους παραλήπτες και το μήνυμα είναι πολύ προσεκτικά γραμμένο και εξατομικευμένο προκειμένου να πείσει για την αυθεντικότητά του. Στο μήνυμα δηλώνεται π.χ. ότι προέρχεται από την τράπεζα που εξυπηρετεί τον τραπεζικό λογαριασμό του ή την πιστωτική του κάρτα, παραθέτοντας λεπτομέρειες που ίσως έχουν αλιευθεί προηγουμένως με κάποιο τρόπο, π.χ. την ημερομηνία γέννησης ή τον αριθμό της κάρτας. Το θύμα κατευθύνεται στη συνέχεια σε μια ιστοσελίδα απομίμηση της πραγματικής για να δώσει τους κωδικούς του.

Μια άλλη υποκατηγορία του phishing είναι το *whaling* (φαλαινοθηρία), όπου εμπλέκονται «μεγάλα ψάρια», δηλαδή ο αποστολέας παριστάνει το υψηλόβαθμο στέλεχος μιας εταιρίας (όπως στο CEO fraud[23]) ή οργανισμού και το μήνυμα προορίζεται

[23] CEO fraud είναι μια τεχνική απάτης που βασίζεται σε κοινωνική μηχανική. Ο απατεώνας ζητάει μια εξυπηρέτηση από τον υπάλληλο μιας εταιρίας παριστάνοντας τον διευθύνοντα σύμβουλο.

συχνά για κάποιο επίσης υψηλόβαθμο στέλεχος, ή και όχι. Το 2016 η μεγάλη εταιρία παιχνιδιών *Mattel* (που παράγει την *Barbie*, τα *Hot Wheels* κ.α.) παρά λίγο να στείλει τρία εκατομμύρια δολλάρια σε απατεώνες με που φαίνονταν να έχουν την έδρα τους στην Κίνα [Hil16]. Οι απατεώνες έστειλαν στην διευθύντρια του λογιστηρίου ένα μήνυμα δήθεν προερχόμενο από τον νέο διευθύνοντα σύμβουλο για μεταφορά χρημάτων στον λογαριασμό ενός προμηθευτή στην *Bank of Wenzhou*. Κατά σύμπτωση, την οποία οι απατεώνες δεν υπολόγισαν, η εντολή Παρασκευή και πρωτομαγιά, δηλαδή σε μέρα τραπεζικής αργίας, οπότε η ανάληψη του ποσού δεν μπορούσε να γίνει ως την επόμενη εργάσιμη. Στο διάστημα των τριών ημερών η εταιρία συνεργάσθηκε με τις κινεζικές αρχές και ανέστρεψε τη μεταφορά. Μια παρόμοια απάτη υπέστη το 2015 η *Ubiquiti Networks*, της οποίας ο κλάδος στο Hong Kong έστειλε περί τα 47 εκατ. δολλάρια σε ψεύτικους προμηθευτές.

Τόσο το spear-phishing όσο και το whaling είναι τεχνικές που απαιτούν υψηλό βαθμό εξατομίκευσης του περιεχομένου του μηνύματος προς τον παραλήπτη. Κατά συνέπεια δεν βασίζονται στη χρήση μαζικών email, αλλά στην στόχευση και στην μεγάλη αναμενόμενη αμοιβή.

Στατιστικά στοιχεία

Σύμφωνα με στατιστικά στοιχεία του Φεβρ. 2021 καθημερινά διακινούνται πάνω από 22 δισεκατομμύρια μηνύματα [Bud21], εξ αυτών το 85% είναι spam. Τα μηνύματα αυτά επιφέρουν κόστος, το οποίο εκτιμάται στα 20 δισεκατομμύρια δολλάρια. Μια εκτίμηση του FBI μόνο για τις ΗΠΑ ήταν πάνω από 12 δισεκατομμύρια δολλάρια για το 2018 [Kar+19]. Οι τρείς πολυπληθέστερες κατηγορίες μηνυμάτων spam είναι (α) διαφήμιση (36%), (β) περιεχόμενο για «ενήλικες» (32%), (γ) οικονομικά θέματα (26%). Μόνο το 2.5% είναι μηνύματα απάτης. Η πρώτη σε όγκο χώρα πηγή spam είναι οι ΗΠΑ, όπου εδρεύουν οι 7 στους 10 μεγαλύτερους spammers και ακολουθάει η Κίνα. Αμφότερες στέλνουν περισσότερα από 8 δισεκατομμύρια μηνύματα το μήνα. Όμως ο υπ' αριθμόν ένα spammer είναι η *Canadian Pharmacy* από την Ουκρανία.[24] Ανερχόμενη «δύναμη» είναι η Βραζιλία [Kar+19].

Νομικό πλαίσιο για το spam

Στις ΗΠΑ ισχύει από το 2004 η *CAN SPAM Act* του 2003 της FTC (Federal Trade Commission), η οποία περιέχει κανόνες για την αποστολή διαφημιστικού email. Σύμφωνα με τους κανόνες (α) δεν επιτρέπεται η αποστολή παραπλανητικών επικεφαλίδων στο email, (β) δεν επιτρέπεται η χρήση θέματος που δεν αντιστοιχεί στο περιεχόμενο, (γ) το μήνυμα πρέπει να χαρακτηρίζεται ως διαφημιστικό, (δ) οι παραλήπτες πρέπει να γνωρίζουν την τοποθεσία του αποστολέα, (ε) πρέπει να έχουν δικαίωμα να αρνούνται περαιτέρω λήψη μηνυμάτων (opt out) και οι δηλώσεις τους να λαμβάνονται υπόψη, (στ) όταν ανατίθεται διαφήμιση σε τρίτους, τόσο η εταιρία που αναθέτει όσο και ο ανάδοχος έχουν από κοινού τις ευθύνες για την τήρηση του νόμου. Για τους παραβάτες προβλέπονται πρόστιμα.

Στην Ευρωπαϊκή Ένωση υπάρχουν οι νόμοι κάθε χωριστού κράτους. Επίσης ο *Κανονισμός (ΕΕ) 2019/1150* περιέχει προβλέψεις για πάσης φύσεως online υπηρεσίες, στις οποίες πρέπει να αποφεύγεται η χρήση malware, περιλαμβανομένου του spam. Επίσης η ανακοίνωση COM/2006/0688 της 15/11/2006 *σχετικά με την καταπολέμηση*

[24] Για μια online λίστα με τους top spammers βλ. `https://www.spamhaus.org/statistics/spammers/`

των ανεπίκλητων ηλεκτρονικών μηνυμάτων, του κατασκοπευτικού και του κακόβουλου λογισμικού και η Οδηγία 2009/136/ΕC που συμπληρώνει την 2002/58/ΕC (περί επεξεργασίας προσωπικών δεδομένων).

Για το spam χρησιμοποιείται ο όρος *ανεπιθύμητα μηνύματα* στον Κανονισμό (ΕΕ) 2019/1150, ενώ στην COM/2006/0688 η περίφραση *ανεπίκλητα ηλεκτρονικά μηνύματα εμπορικού χαρακτήρα* ή απλώς *ανεπίκλητα ηλεκτρονικά μηνύματα* ή περαιτέρω στην ίδια ανακοίνωση *αυτόκλητα μηνύματα*.

Μέτρα καταπολέμησης του spam

Όποιος χρησιμοποιεί μια υπηρεσία email όπως gmail, hotmail, yahoo, iCloud κ.λπ. θα έχει παρατηρήσει τον όγκο μηνυμάτων που μαζεύονται σε φακέλους με ονόματα όπως "spam" και "junk". Τα μηνύματα που πέφτουν εκεί είναι πολλαπλάσια αυτών που φτάνουν στο "inbox" και το ξεδιάλεγμα των χρήσιμων μηνυμάτων θα προκαλούσε κάμποση απώλεια χρόνου στο μέσο χρήστη. Ωστόσο η πολιτική αυτών των υπηρεσιών email είναι να διατηρούν το περιεχόμενο αυτών των φακέλων για κάποιο διάστημα, αντί να το στέλνουν σε άμεση διαγραφή, ακριβώς διότι κανένα φίλτρο και κανένα κριτήριο διαχωρισμού δεν θα μπορούσε να αποφασίσει με βεβαιότητα αν ορισμένα από τα μηνύματα που φαίνονται να είναι spam είναι όντως τέτοια. Κάθε χρήστης έχει κατά καιρούς πάρει ένα χρήσιμο μήνυμα που χάθηκε επειδή έπεσε στο spam folder και πρέπει να το ανασύρει από κει.

Ο έλεγχος μηνυμάτων κατά πόσο εμπίπτουν σε spam γίνεται βεβαίως κατά κύριο λόγο για τα εισερχόμενα μηνύματα, γίνεται όμως και για τα εξερχόμενα, προκειμένου να αποφευχθεί η περίπτωση να στέλνει κανείς spam χωρίς να το γνωρίζει, επειδή ο υπολογιστής του είναι μολυσμένος.

Η αγορά προσφέρει μια σειρά από λύσεις είτε με τη μορφή προϊόντων είτε υπηρεσιών που ελέγχουν εισερχόμενα και εξερχόμενα μηνύματα για spam, phishing, αλλοιωμένες διευθύνσεις (spoofing) και γενικά για επιθέσεις μέσω email. Διάφορα σημεία σε ένα δίκτυα είναι πρόσφορα για το φιλτράρισμα των μηνυμάτων. Οι ISPs (Internet Service Providers) μπορούν να κάνουν έλεγχο σε firewalls πριν από τους εξυπηρετητές και προωθητές αλληλογραφίας (email servers & relays). Οι εξυπηρετητές αλληλογραφίας είναι το κατ' εξοχήν σημείο όπου μπορούν να γίνουν έλεγχοι μηνυμάτων. Επίσης φίλτρα μπορούν να μπουν στην πλευρά του χρήστη στον email client. Αν ένας ιδιώτης χρήστης email χρησιμοποιεί την υπηρεσία email ενός μεγάλου παρόχου email είναι πιθανό να μη χρειαστεί άλλη προστασία, αλλά αν θέλει παραπάνω προστασία μπορεί να εγκαταστήσει ένα πρόσθετο spam filter. Μια εταιρία που χρησιμοποιεί τον δικό της email server μπορεί να τον εξοπλίσει με το κατάλληλο λογισμικό. Εναλλακτικά μπορεί να αγοράσει μια ολοκληρωμένη υπηρεσία email (email hosting service) που θα ενσωματώνει τους κατάλληλους ελέγχους.

Το φιλτράρισμα μπορεί να γίνει με διάφορα κριτήρια, ήτοι αναζήτηση συγκεκριμένων λέξεων (opportunity, million, Viagra, Xanax, sex, click, winner) και φράσεων (call now, act now, free trial, meet singles), συνδέσμων που παραπέμπουν σε ιστοσελίδες με κακή φήμη, διευθύνσεις αποστολέα με κακή φήμη, πολλαπλά μηνύματα κ.α. Αντίθετα, ενδείξεις συμμόρφωσης με το νόμο, όπως σύνδεσμοι για unsubscribe, μειώνουν την πιθανότητα χαρακτηρισμού ενός μηνύματος ως spam.

Τα παραπάνω κριτήρια θα μπορούσαν να υλοποιηθούν με τη μιας σειράς ελέγχων, όπου κάθε φορά ελέγχεται η συμμόρφωση με ένα κανόνα (όπως σε ένα firewall). Όταν επιβεβαιωθεί η συμμόρφωση το μήνυμα προωθείται στον επόμενο έλεγχο, διαφορετικά πέφτει στο spam folder. Ωστόσο μια τέτοια πρακτική για τους περισσότερους κανόνες θα ήταν υπερβολικά επιθετική και θα οδηγούσε σε πολλά false positives, δη-

λαδή μηνύματα που αδίκως θα κατατάσσονταν ως spam. Μια πιο συντηρητική προσέγγιση είναι η βαθμολόγηση του μηνύματος ως προς τη συμμόρφωσή του με κάθε κανόνα και η δημιουργία ενός σταθμισμένου αθροίσματος (δηλ. με διαφορετικό βάρος για κάθε κανόνα). Μηνύματα που περνούν ένα κατώφλι απορρίπτονται.

Ένα άλλο μέτρο για την παρεμπόδιση του spam είναι η παρεμπόδιση της αλλοίωσης της διεύθυνσης του αποστολέα, την ταυτότητα του οποίου ελέγχει και εγγυάται κάποια αρχή, συνήθως ο πάροχος του email. Ο έλεγχος βασίζεται σε μεθόδους όπως το DKIM (DomainKeys Identified Mail), όπου ο έλεγχος γίνεται από τον πάροχο email, και το SPF (Sender Policy Framework), όπου ο έλεγχος γίνεται από τον ιδιοκτήτη ενός Internet domain.

Ένα άλλο μέτρο είναι η τήρηση μαύρης ή λευκής λίστας. Η μαύρη λίστα περιέχει τις διευθύνσεις από email servers ή πεδία (domains) που έχουν εντοπισθεί για αποστολή ή προώθηση spam. Αντίστοιχα στη λευκή λίστα (white-list) καταγράφονται εγγυημένοι servers και domains. Τέτοιες υπηρεσίες τήρησης καταλόγων παρέχουν οι εταιρίες και οργανισμοί *Spamhaus*, *Spamcop*, *Invalument*, *Barracuda* κ.α. Μια σειρά από άλλες τεχνικές έχουν προταθεί, βασισμένες σε ποικίλες ιδέες, περιλαμβανομένων νευρωνικών δικτύων και μηχανικής μάθησης [Kar+19; Dad+19].

Rootkit

Το *rootkit* είναι λογισμικό που επιτρέπει τον έλεγχο μιας μηχανής από μακριά. Μπορεί να χρησιμοποιηθεί για στοχευμένες επιθέσεις και συχνά οι υπολογιστές που είναι μολυσμένοι με αυτόν τον τρόπο αποτελούν μέλη ενός botnet. Η λέξη αποτελείται από δύο συνθετικά (root-kit), των οποίων η σύνθεση δηλώνει συλλογή εργαλείων με αυξημένα δικαιώματα πρόσβασης.

Ένα καλογραμμένο rootkit δεν θα δημιουργήσει προβλήματα στον υπολογιστή όπου είναι εγκατεστημένο, προκειμένου να μη γίνει αντιληπτό. Ορισμένα rootkits φτάνουν στο σημείο να εγκαταστήσουν λογισμικό antivirus στον προσβεβλημένο υπολογιστή προκειμένου να εμποδίσουν άλλο κακόβουλο λογισμικό να πάρει τον έλεγχο του υπολογιστή [And08]. Ο κατασκευαστής του rootkit μπορεί να παρέχει υπηρεσίες μετά την πώληση (after sales service): Αν προκύψει αμυντικό λογισμικό που αφαιρεί το rootkit, θα το αναβαθμίσει με κατάλληλα αντίμετρα.

Το rootkit μπορεί να εγκατασταθεί από έναν επιτιθέμενο είτε «χειροκίνητα» σε επιλεγμένες μηχανές, είτε αυτόματα σε περίπτωση που ο στόχος είναι μια μαζική εγκατάσταση (όπως σε ένα botnet). Επίσης στη συνέχεια μπορεί να χρησιμοποιηθεί είτε μεμονωμένα από ένα χειριστή, είτε μαζικά στέλνοντας εντολές από ένα κέντρο ελέγχου.

Στο rootkit μπορεί να περιλαμβάνονται εργαλεία που θα χρησιμοποιηθούν σε κακόβουλες ενέργειες, αλλά και εργαλεία που θα αποκρύψουν την ύπαρξη του rootkit, καθώς και τα ίχνη των κακόβουλων ενεργειών (π.χ. θα απενεργοποιήσουν την δημιουργία log files, όπου θα φαινόταν αυτές οι ενέργειες, ή θα τα σβήσουν εκ των υστέρων). Ένα διάσημο rootkit με τέτοιες δυνατότητες ήταν αυτό που χρησιμοποιήθηκε στις ελληνικές υποκλοπές του 2005 [PS07; Pap18].

Η ανίχνευση των rootkits γίνεται με τις ίδιες τεχνικές που ανιχνεύεται το πάσης φύσεως κακόβουλο λογισμικό. Σε ορισμένες περιπτώσεις ένα rootkit γίνεται αντιληπτό εξ αιτίας της ασυμβατότητάς του με το κανονικό λογισμικό μιας μηχανής και μάλιστα με τις αναβαθμίσεις του.

Backdoor

Το *backdoor* ή *πίσω πόρτα* ή *κερκόπορτα* άρχισε ως διευκόλυνση για προγραμματιστές και άλλους τεχνικούς που έκαναν συντήρηση σε υπολογιστές. Στη δεκαετία του ’90 μια δημοφιλής τότε μάρκα υπολογιστών mini[25] είχε ένα γενικό password που το αποτελούσε μια σειρά διαδοχικών γραμμάτων του αλφαβήτου από το τέλος προς την αρχή. Το προϊόν *Interbase* (μια βάση δεδομένων) της εταιρίας *Borland* διέθετε σε μια σειρά εκδόσεών του το εξής backdoor: Username *politically*, password *correct*.[26]

Γενικά το backdoor αναφέρεται σε ένα μη κανονικό τρόπο πρόσβασης που έχει δημιουργηθεί από τον κατασκευαστή μιας συσκευής, είτε για τεχνικούς και συντηρητές, είτε για την αστυνομία, το στρατό, μυστικές υπηρεσίες κ.α. Ο όρος χρησιμοποιείται γενικότερα για συνιστώσες και συστήματα software και hardware.

Σε ορισμένες περιπτώσεις κρατικές οντότητες έχουν αποπειραθεί να θεσμοθετήσουν και να χρησιμοποιήσουν επισήμως μια κερκόπορτα (βλ. π.χ. το *Clipper* chip στις ΗΠΑ). Πολλές συζητήσεις έχουν γίνει κατά καιρούς για το αν υπάρχει κερκόπορτα σε υλοποιήσεις κρυπτογραφικών αλγορίθμων, σε λειτουργικά συστήματα, σε επεξεργαστές, σε συσκευές κινητής τηλεφωνίας κ.λπ., προς χρήση κρατικών υπηρεσιών της χώρας όπου παράγεται ένα προϊόν ή και ως όρος που θέτει μια τρίτη χώρα σε μια κατασκευάστρια εταιρία για να επιτρέψει τις εμπορικές της δραστηριότητες. Συζητήσεις για θέματα ισορροπίας μεταξύ καταπολέμησης του εγκλήματος και προστασίας της ιδιωτικότητας συχνά συνδέονται με το αν υπάρχουν κερκόπορτες ή αν είναι θεμιτό να υπάρχουν. Ακόμη, φήμες για την ύπαρξη κερκόπορτας σε δημοφιλή προϊόντα (π.χ. συσκευές κινητής τηλεφωνίας) διοχετεύονται ως μέρος ενός εμπορικού επικοινωνιακού πολέμου. Κατά κανόνα οι τέτοιου είδους φήμες και τα δημοσιεύματα δεν βρίσκουν οριστική διάψευση ή επαλήθευση.

Μια άλλη συζήτηση είναι κατά πόσο ένα backdoor είναι άλλου είδους τρωτότητα, πιθανώς μη σκόπιμη. Ένα αρχικό σφάλμα μπορεί να αξιοποιηθεί ως backdoor ή να δικαιολογηθεί ως backdoor, ενώ μια σκόπιμη τρωτότητα μπορεί να χαρακτηρισθεί από τους δημιουργούς ως τυχαία μετά την αποκάλυψή της.

Ένα backdoor μπορεί να έχει εγκατασταθεί σε μια μηχανή με οποιοδήποτε γνωστό τρόπο διάδοσης κακόβουλου λογισμικού. Επίσης μπορεί να συνδυάζεται με άλλες δυνατότητες. Για παράδειγμα η περιγραφή ενός backdoor έχει ως εξής στις ιστοσελίδες[27] της Microsoft: *Backdoor:Win32/Hackdef.R is a backdoor Trojan ... This Trojan is a user-mode rootkit.*

Ransomware

Το *ransomware* είναι κακόβουλο λογισμικό με στόχο την απόσπαση λύτρων (ransom) από το θύμα. Ο επιτιθέμενος κάνει το αντίστοιχο μιας απαγωγής, δηλαδή αφαιρεί από τη θύμα την δυνατότητα πρόσβασης σε κρίσιμα αρχεία του. Τα αρχεία αυτά αντί ο επιτιθέμενος να τα σβήσει και να τα μεταφέρει στον κατοχή του, υλοποιεί την σχετικά απλούστερη λύση να τα κρυπτογραφήσει με ένα κλειδί που μόνο ο ίδιος γνωρίζει. Στη συνέχεια διαπραγματεύεται με το θύμα την πώληση του κλειδιού [KKJ16].

Ωστόσο στην περίπτωση αυτού του τύπου απάτης ο επιτιθέμενος-εκβιαστής έχει να λύσει άλλο ένα πρόβλημα: Πώς θα πληρωθεί από το θύμα χωρίς ο ίδιος να εντοπι-

[25] Minicomputer ήταν ένας υπολογιστής μεσαίου σχετικά κόστους, φυσικού μεγέθους όσο ένα μεγάλο ψυγείο, συνήθως με το δικό του ψυχόμενο χώρο, ενώ εξυπηρετούσε αριθμό τερματικών διανεμημένων σε κοντινά γραφεία στο ίδιο κτήριο. Η χρήση των mini άρχισε το 1960 και κράτησε ως τη δεκαετία του 1990.

[26] Η τρωτότητα περιγράφεται ως εξής: *This back door allows any local user or remote user able to access port 3050/tcp [gds_db] to manipulate any database object on the system ...* [Div01]

[27] `https://www.microsoft.com/en-us/wdsi/threats`

σθεί μέσω των ιχνών της πληρωμής. Η κλασσική οδός πληρωμής είναι μέσω κρυπτονομίσματος. Επίσης, αν το τίμημα πρόκειται να γίνει αντικείμενο διαπραγμάτευσης, ο εκβιαστής χρειάζεται ένα κανάλι επικοινωνίας με το θύμα.

Ταυτόχρονα το θύμα έχει να αντιμετωπίσει τα συνήθη προβλήματα και τις προκλήσεις μιας κλασσικής απαγωγής, δηλαδή πώς θα εμπιστευθεί τον εκβιαστή ότι όντως θα παραδώσει το κλειδί και πώς δεν θα πάθει τα ίδια σε λίγο από τον ίδιο επιτιθέμενο ή άλλον, δεδομένου ότι η επίθεση έγινε μέσω κάποιας τρωτότητας. Πριν αποφασίσει αν πληρώσει, το θύμα είναι πιθανό να αναζητήσει κάποια λύση αποκρυπτογράφησης, που μάλλον είναι καταδικασμένη σε αποτυχία επειδή ένας πεπειραμένος εκβιαστής θα χρησιμοποιήσει δοκιμασμένο αλγόριθμο σε συνδυασμό με ικανού μήκους κλειδί.

Στατιστικά η πλειοψηφία των προσβολών από ransomware οφείλεται σε phishing και γενικά σε μεθόδους με ισχυρή συνιστώσα κοινωνικής μηχανικής. Το κόστος από ransomware για το 2021 εκτιμάται σε μερικές δεκάδες δισεκατομμυρίων €. Ωστόσο η συγκέντρωση στατιστικών στοιχείων είναι δύσκολη, επειδή πολλά από τα θύματα δεν επιθυμούν τη βλάβη φήμης που μπορεί να προκύψει από τη δημοσιοποίηση τέτοιων περιστατικών. Κατά προτίμηση προσβάλλονται όσοι έχουν κρίσιμα αρχεία, είναι ικανοί να πληρώσουν και δεν διαθέτουν επαρκή άμυνα, ως επί το πλείστον μικρο-μεσαίες επιχειρήσεις [Owa20]. Η πλειοψηφία των επιχειρήσεων έχει υποστεί κάποια επίθεση. Το 1/3 περίπου των επιθέσεων αφορά σε θύματα στις ΗΠΑ [Hum+20], σύμφωνα με τα ανακοινωμένα στοιχεία, αλλά πρέπει ίσως να ληφθεί υπόψη ότι η διαφάνεια των στοιχείων δεν είναι ίδια σε όλες τις χώρες. Σε μια ανακοίνωση[28] που εξέδωσε το FBI το 2015 αναφέρεται ότι από τον Απρίλιο του 2014 ως τον Ιούνιο του 2015 έλαβε γνώση 992 επιθέσεων που έγιναν μέσω του *Cryptowall* με συνολική ζημιά 18 εκατομμυρίων δολλαρίων. Το συνηθισμένο εύρος λύτρων σύμφωνα με την ίδια ανακοίνωση ήταν από $200 ως $10000 και το προτιμώμενο μέσο πληρωμής ήταν το Bitcoin. Τον Μάρτιο του 2021 ζητήθηκαν από την Acer $50.000.000.

Προχωρημένη Επίμονη Απειλή

Η δημιουργία κακόβουλου λογισμικού ήταν αρχικά μια περιθωριακή δραστηριότητα την οποία μπορούσαν να ασκήσουν μόνο όσοι είχαν προχωρημένες γνώσεις προγραμματισμού. Σε δεύτερη φάση δημιουργήθηκε λογισμικό που μπορούσε να βοηθήσει στην κατασκευή malware ακόμη και από μικρά παιδιά. Τα τέτοιου είδους εργαλεία εξελίχθηκαν και μαζί τους μια ολόκληρη αγορά εργαλείων και υπηρεσιών. Στο παιχνίδι της χρήσης του κακόβουλου λογισμικού μπήκαν ως πελάτες, αλλά και ως δημιουργοί, απατεώνες, οργανωμένο έγκλημα, ακτιβιστές, τρομοκράτες, αστυνομία, υπηρεσίες κατασκοπείας και αντικατασκοπείας, ειδικές στρατιωτικές υπηρεσίες κ.α. Ορισμένες από αυτές τις ομάδες διαθέτουν πλούσιους πόρους και έχουν τη δυνατότητα να κάνουν επιθέσεις μεγάλης κλίμακας με ποικίλα μέσα και προχωρημένη τεχνολογία. Σε ορισμένες περιπτώσεις συνεχίζουν την επίθεση για όσο χρόνο χρειαστεί, χρησιμοποιούν εναλλακτικές μέθόδους και γενικά κάνουν ό,τι είναι απαραίτητο για να επιτευχθεί ο σκοπός τους. Τέτοιες προσπάθειες μεγάλης κλίμακας χαρακτηρίζονται με τον όρο *Προχωρημένη Επίμονη Απειλή* (Advanced Persistent Threat - APT).

Μια από τις πρώτες επιθέσεις που χαρακτηρίζονται πλέον ως APT είναι αυτή του σκουληκιού *Stuxnet* (βλ. και πιο κάτω). Οι λόγοι κάνουν APT τον Stuxnet είναι ότι ο στόχος ήταν κρατικές βιομηχανικές εγκαταστάσεις του Ιράν, ότι για τον προγραμματισμό του κώδικα χρειάζονται εξειδικευμένες γνώσεις προσβολής βιομηχανικών συστημάτων, ότι χρησιμοποιήθηκαν τρωτότητες μηδενικής μέρας (που πιθανώς αγορά-

[28] `https://www.ic3.gov/Media/PDF/Y2015/PSA150623.pdf`

σθηκαν στη μαύρη αγορά σε πολύ ψηλή τιμή) και ότι ήταν ένα εργαλείο στην γεωστρατηγική αναμέτρηση ΗΠΑ-Ιράν στην περιοχή της πυρηνικής τεχνολογίας. Ο Stuxnet έγινε γνωστός περί το 2010, αλλά θεωρείται ότι η ανάπτυξή του άρχισε γύρω στο 2005.

Μια σειρά δεκάδων «ομάδων» που υλοποιούν επιθέσεις τύπου APT έχουν εντοπισθεί (ορισμένες γνωστές ως APT1, APT2 κ.λπ.). Ένας κατάλογος υπάρχει στο λήμμα *Advanced persistent threat* στη Wikipedia. Διακεκριμένη περίπτωση APT αποτελεί η «υπόθεση SolarWinds» που περιγράφεται πιο κάτω στην ενότητα 5.6.

5.6 Διακεκριμένες περιπτώσεις κακόβουλου λογισμικού

Πιο κάτω είναι ένας πολύ μικρός κατάλογος από χαρακτηριστικές περιπτώσεις κακόβουλου λογισμικού που προκάλεσαν τη μόλυνση μεγάλων πληθυσμών μηχανών και πολλές ζημιές. Ο Filiol έχει θέσει το ερώτημα πώς μπορείς να παρουσιάσεις έναν ιό χωρίς να δώσεις τον κώδικά του. Ταυτόχρονα στην κοινότητα των ειδικών ασφάλειας έχουν γίνει πολλές συζητήσεις σε ποιο βαθμό είναι σωστό να δίνονται λεπτομέρειες για τις τρωτότητες και για το κακόβουλο λογισμικό που τις εκμεταλλεύεται. Η πρακτικά επικρατούσα άποψη, που εφαρμόζεται π.χ. στις ανακοινώσεις τρωτοτήτων και στις ενημερώσεις λογισμικού είναι να δίνεται μια πολύ περιορισμένη περιγραφή. Παρ' όλα αυτά κατά καιρούς εμφανίζονται στο Διαδίκτυο δημοσιεύσεις με αναλύσεις ιών και σκουληκιών που συνοδεύονται από μέρος ή το σύνολο του κώδικα. Σε κάποιες περιπτώσεις ειδικοί ασφάλειας, οι οποίοι μετά την ανάλυση τρωτοτήτων και του αντίστοιχου κακόβουλου λογισμικού έκαναν τέτοιες αποκαλύψεις, θεωρήθηκαν - εκ του αποτελέσματος - υπεύθυνοι για περαιτέρω αξιοποίηση αυτής της γνώσης από την πλευρά της επίθεσης. Πιο κάτω δίνονται περιληπτικές περιγραφές της λειτουργίας του κακόβουλου λογισμικού και κατά περίπτωση παραπομπές στον κώδικα ή σε αναλύσεις του.

Επίσης δίνονται ορισμένα στοιχεία για το ποιοι είναι ή θεωρούνται δημιουργοί του κακόβουλου λογισμικού και πώς αντιμετωπίζονται νομικά. Ορισμένοι διάσημοι της πλευράς της επίθεσης έχουν αξιοποιηθεί στη συνέχεια από την πλευρά της άμυνας, ως ειδικοί ασφάλειας και ιδίως ως ειδικοί στην ανάλυση τρωτότητας συστημάτων. Ορισμένοι άλλοι έχουν καταλήξει στη φυλακή, ενώ όσοι έχουν μείνει ασύλληπτοι συνεχίζουν κατά κανόνα τις δραστηριότητές τους. Κατά πόσον έχουν διασυνδέσεις με εθνικές υπηρεσίες κατασκοπείας και κυβερνοεπιθέσεων είναι ένα σκοτεινό θέμα.

Creeper

Γράφτηκε σε assembly του PDP-10 το 1971 από τον Bob Thomas όταν αυτός ήταν στην εταιρία Bolt, Beranek and Newman Inc. (BBN). Είχε τη δυνατότητα να μεταφέρεται ανάμεσα σε μηχανές που έτρεχαν το λειτουργικό σύστημα TENEX [Bob+72], το οποίο είχε εισαχθεί το 1960 από την BBN για τους PDP-10. Ο Creeper το μόνο που έκανε ήταν να τυπώνει τη φράση `I'M THE CREEPER : CATCH ME IF YOU CAN`. Στη συνέχεια ο Ray Tomlinson[29] έγραψε μια δεύτερη έκδοση που αντί μόνο να κινείται από τη μια μηχανή στην άλλη δημιουργούσε ένα αντίγραφο στην επόμενη μηχανή, επομένως πολλαπλασιαζόταν. Κατόπιν έγραψε τον Reaper που κυκλοφορούσε στο Arpanet και έσβηνε τα αντίγραφα του Creeper. Ο Tomlinson είχε πει ότι ο σκοπός του Creeper ήταν να δημιουργηθεί ένα πρόγραμμα που θα είχε την ικανότητα να αναζητήσει τη μηχανή όπου θα ήταν πιο πρόσφορη η εκτέλεσή του, π.χ. εκεί που υπάρχει διαθέσιμη μνήμη ή διαθέσιμη ικανότητα επεξεργασίας.

[29] Raymond Samuel Tomlinson, 1941–2016, επινόησε το email.

Morris worm

Το σκουλήκι το δημιούργησε ο Robert Tappan Morris Jr. (γενν. το 1965, ο πατέρας του, επίσης Robert Morris, εργαζόταν στην NSA ως κρυπτογράφος) ως μεταπτυχιακός φοιτητής στο παν. Cornell με σκοπό να μετρήσει το μέγεθος του ARPANET. Το σκουλήκι σε κάθε νέα μηχανή που θα μόλυνε θα προκαλούσε την αύξηση ενός μετρητή κατά ένα. Στη συνέχεια (2 Νοέμ. 1988) το έθεσε σε κυκλοφορία από το MIT για να μην αποκαλυφθεί. Παρ' όλα αυτά βρέθηκε κατηγορούμενος με ένα νέο νόμο που λεγόταν *Computer Fraud and Abuse Act*.

Ο Morris έγραψε τον κώδικα[30] ώστε να γίνεται διάδοση σε ένα ποσοστό των στόχων, χωρίς όμως να βάλει έλεγχο αν έχει γίνει ήδη η διάδοση. Το αποτέλεσμα ήταν να αυτοαναπαράγεται το σκουλήκι στον ίδιο υπολογιστή χωρίς περιορισμό, μέχρι του σημείου να φτάνει στον κορεσμό τις δυνατότητες επεξεργασίας του. Δύο χιλιάδες μηχανές με Berkeley Unix μολύνθηκαν σε 15 ώρες και ήταν άχρηστες εξ αιτίας του φόρτου που είχε δημιουργηθεί σε κάθε μια. Το δίκτυο χωρίστηκε σε partitions ώστε να καθαριστούν οι υπολογιστές χωριστά σε κάθε partition. [31]

ILOVEYOU

Το σκουλήκι *ILOVEYOU* ή *Love Bug* ή *Love Letter for you* εμφανίσθηκε από τον Μάιο του 2000 ως συνημμένο αρχείο `LOVE-LETTER-FOR-YOU.txt.vbs` σε μηνύματα email με θέμα `ILOVEYOU`. Η κατάληξη `.vbs` υποδηλώνει ένα Visual Basic script, ωστόσο έμενε κρυμμένη δίνοντας την εντύπωση στον παραλήπτη του μηνύματος ότι το συνημμένο ήταν ένα απλό αρχείο κειμένου (με κατάληξη `.txt`). Ο παραλήπτης άνοιγε το συνημμένο και προκαλούσε έτσι την εκτέλεση του script από την Visual Basic. Η διάδοση βασιζόταν στην απροσεξία του παραλήπτη, άρα στην κοινωνική μηχανική.

Το script περιελάμβανε εντολές με τις οποίες αντικαθιστούσε αρχεία με καταλήξεις `doc, jpg, jpeg, vbs, vbe, js, jse, css, wsh, sct, hta` και `mp2` με αντίγραφα του εαυτού του, στα οποία προσέθετε την κατάληξη `.vbs`. Επίσης έκανε απόκρυψη διαφόρων αρχείων ήχου. Τέλος φρόντιζε για τη διάδοσή του στέλνοντας ένα αντίγραφο του εαυτού του σε όλες τις διευθύνσεις του *Windows Address Book*.

Το σκουλήκι κατάφερε να μεταδοθεί σε πάνω από δέκα εκατομμύρια υπολογιστών με Windows 9x, NT 4.0 και 2000. Ξεκίνησε από τη Μανίλα (Φιλιππίνες) στις 4/5/2020 και τις πρώτες δέκα μέρες είχαν μολυνθεί πάνω από 50 εκατομμύρια μηχανών. Οι ζημιές εκτιμήθηκαν σε μερικά δισεκατομμύρια δολλάρια παγκοσμίως και σε ακόμη μεγαλύτερο κόστος για να διαγραφεί το σκουλήκι και να αποκατασταθούν τα διαγραμμένα αρχεία.

Το γεγονός ότι ο κώδικας ήταν Visual Basic script έδωσε τη δυνατότητα σε πολλούς να κάνουν τις δικές τους παραλλαγές και να τον ξανακυκλοφορήσουν. Οι τροποποιήσεις αφορούσαν στις καταλήξεις των προσβαλλομένων αρχείων, την επικεφαλίδα του μηνύματος κ.α.

Με το ξέσπασμα της εξάπλωσης του ιού συνελήφθησαν μερικά άτομα στη Μανίλα, μεταξύ των οποίων ένας πρώην φοιτητής που είχε αποβληθεί στο τελευταίο έτος των

[30] Ο κώδικας του σκουληκιού του Morris μπορεί να βρεθεί στο `https://0x00sec.org/t/examining-the-morris-worm-source-code-malware-series-0x02/685`.

[31] Ο Morris καταδικάστηκε το 1989 σε πάνω από $13000 πρόστιμο μαζί με 400 ώρες κοινωνικής εργασίας και τρία χρόνια με αναστολή. Το συνολικό κόστος του καθαρισμού εκτιμήθηκε μεταξύ $100.000 και $10.000.000 κατά την έφεση που όμως επαναβεβαίωσε την ποινή. Το 1999 εκλέχτηκε βοηθός καθηγητής στο MIT και το 2006 μόνιμος καθηγητής για *θεμελιώδεις συμβολές στις περιοχές των υπολογιστών, των ασύρματων δικτύων και των λειτουργικών συστημάτων*. Συμβολή στην περιοχή της ασφάλειας δεν αναφέρθηκε.

σπουδών του από ένα κολλέγιο στην Quezon City (Φιλιππίνες), επειδή στην πτυχιακή του εργασία είχε προτείνει ένα δούρειο ίππο για την υποκλοπή κωδικών εισόδου σε υπηρεσίες σύνδεσης στο Internet [Whi20]. Το αρχικό script που είχε δημιουργήσει με σκοπό την υποκλοπή κωδικών το έστελνε σε επιλεγμένα θύματα με τα οποία επικοινωνούσε μέσω chat. Ωστόσο την άνοιξη του 2000 είχε την ιδέα να προσθέσει τη διάδοση σε επαφές Outlook εκμεταλλευόμενος μια τρωτότητα στα Windows 95. Οι κατηγορούμενοι τελικά αφέθηκαν ελεύθεροι επειδή δεν υπήρχαν κατάλληλοι νόμοι για το αδίκημά τους.

Code Red

Το λογισμικό *Internet Information Server* (IIS) της Microsoft επιτρέπει την δημιουργία ενός web server. Ξεκίνησε τον Μάιο του 1995 με την έκδοση 1.0 πάνω από *Windows NT*. Στις 18/6/2001 η Microsoft εξέδωσε μια ανακοίνωση που επεσήμαινε μια τρωτότητα στις εκδόσεις 4.0 και 5.0 του IIS (Microsoft Security Bulletin MS01-033 και στις 19/6 συνοδεύτηκε από την CA-2001-13).[32] Ένα μήνα πιο πριν την ανακοίνωση είχε διανείμει μια επιδιόρθωση (software patch). Στις 13 Ιουλίου, Παρασκευή και 13, κυκλοφόρησε ένα σκουλήκι, το οποίο έγινε αρχικά αντιληπτό από τον Ken Eichman, ειδικό ασφάλειας της *Chemical Abstract Services*, ο οποίος και παρατήρησε ίχνη από 611 επιθέσεις σε logs. Στη συνέχεια οι έρευνες έγιναν από την *eEye Digital Security* η οποία στις 17/6 ανακοίνωσε μια πρώτη ανάλυση για το πώς λειτουργούσε το σκουλήκι. Το σκουλήκι εξαπλώθηκε γρήγορα και μόνο την 19η Ιουλίου προσέβαλε 359.000 μηχανές [Lem01]. Στις μολυσμένες ιστοσελίδες φαινόταν το εξής μήνυμα:

```
HELLO! Welcome to http://www.worm.com! Hacked By Chinese!
```

Το σκουλήκι διέθετε ένα προκαθορισμένο χρονοδιάγραμμα. Τις πρώτες 19 μέρες προσπαθούσε να διαδοθεί, χτυπώντας ανεξαιρέτως μηχανές με Windows. Σε όσες δεν υπήρχε το λογισμικό IIS ή δεν υπήρχε η σωστή έκδοση του IIS η προσβολή σταματούσε χωρίς άλλο αποτέλεσμα. Τις επόμενες 8 μέρες ήταν προγραμματισμένο να πραγματοποιήσει επιθέσεις άρνησης υπηρεσίας (DoS) σε σταθερές διευθύνσεις IP, μία εκ των οποίων ήταν και η τότε διεύθυνση IP 198.137.240.91 της δημόσιας ιστοσελίδας του Λευκού Οίκου (`whitehouse.gov`). Δεδομένου ότι η ανάλυση έγινε πριν την προγραμματισμένη ημερομηνία της επίθεσης, αυτή αποφεύχθηκε αλλάζοντας την προηγούμενη μέρα τη διεύθυνση IP της ιστοσελίδας σε 198.137.240.92 [Ber01b]. Στη συνέχεια το σκουλήκι ήταν προγραμματισμένο να πέσει σε ύπνο, αλλά στις 4/8/2001 προέκυψε η νέα έκδοσή του *Code Red II*. Το νέο σκουλήκι δεν είχε φορτίο που θα αντιστοιχούσε σε συγκεκριμένο πρόγραμμα επιθέσεων, όμως είχε ένα backdoor για τη δημιουργία επιθέσεων.

Σύμφωνα με την άποψη της *eEye Digital Security* το σκουλήκι προερχόταν από το Makati City στις Φιλιππίνες. Ωστόσο η εταιρία κατηγορήθηκε ότι ήταν υπερβολικά λεπτομερής στις ανακοινώσεις της και διευκόλυνε με αυτόν τον τρόπο την πλευρά της επίθεσης.

[32] Η περιγραφή της τρωτότητας στην CA-2001-13 έχει ως εξής: *There is a remotely exploitable buffer overflow in one of the ISAPI extensions installed with most versions of IIS 4.0 and 5.0 (The specific Internet/Indexing Service Application Programming Interface extension is IDQ.DLL). An intruder exploiting this vulnerability may be able to execute arbitrary code in the Local System security context. This essentially can give the attacker complete control of the victim system.* Η υπερχείλιση μπορούσε να γίνει στέλνοντας ένα μακρύ string με επαναλήψεις του γράμματος «N». Βλ. και `https://seclists.org/cert/2001/7`

Nimda

Το σκουλήκι *W32/Nimda* (αντιστροφή της λέξης admin) ήταν μεταξύ των διαδόχων του *Code Red* σε μικρή χρονική απόσταση. Μια αναλυτική περιγραφή του γίνεται στην CERT CA-2001-26 της 18/9/2001. Προσέβαλε μηχανές με λειτουργικά συστήματα *Microsoft Windows* των εκδόσεων 95, 98, ME, NT και 2000. Η καινοτομία του Nimda ήταν ότι χρησιμοποιούσε για την προσβολή ενός υπολογιστή πέντε διαφορετικές μεθόδους μετάδοσης [Hod19]:

1. Διάδοση ανάμεσα σε email clients. Η διάδοση μπορεί να γίνει με εκμετάλλευση μιας τρωτότητας (CVE-2001-0154) του Internet Explorer. Όταν αυτός χρησιμοποιείται για να υποστηρίξει μια υπηρεσία email, ο επιτιθέμενος μπορεί να στείλει ένα μήνυμα με συνημμένο τύπου MIME, το οποίο ο Internet Explorer δεν επεξεργάζεται σωστά και ωθείται να εκτελέσει κακόβουλο κώδικα.

2. Διάδοση ανάμεσα σε clients που υποστηρίζουν τον διαμοιρασμό αρχείων μέσω δικτύου. Ο επιτιθέμενος αρκεί να μοιράσει ένα μολυσμένο αρχείο.

3. Προσβολή από web servers σε clients όταν κάποιος επισκέπτεται σελίδες των πρώτων.

4. Προσβολή από client προς web server με εκμετάλλευση της τρωτότητας *Microsoft IIS 4.0 / 5.0 directory traversal* (VU #111677). Η μολυσμένη μηχανή ψάχνει να βρει έναν IIS server (όπως στον Code Red) και ανεβάζει ένα μολυσμένο αρχείο `readme.eml`.

5. Ψάχνει για backdoors που έχει αφήσει πίσω του ο Code Red.

Μια από τις καινοτομίες του Nimda ήταν ότι μπορούσε να μολύνει ιστοσελίδες και μέσω αυτών να διαδώσει περαιτέρω τη μόλυνση σε όσους τις επισκέπτονταν. Η F-Secure[33] υποστηρίζει ότι η προέλευση του Nimda είναι κινεζική. Τον αρχικό Nimda ακολούθησαν παραλλαγές Nimda.B ως Nimda.E.

SQL Slammer

Το σκουλήκι *SQL Slammer*, γνωστό επίσης ως *Sapphire Worm* και *Helkern*, εκμεταλλεύθηκε μια τρωτότητα σε προϊόντα βάσεων δεδομένων (*SQL Server*, *Desktop Engine*) της Microsoft. Η τρωτότητα είχε εντοπισθεί από την *Next Generation Security (NGS) Software Software* και ανακοινώθηκε δημόσια από την Microsoft (με την MS02-039) τον Ιούλιο του 2002. Η τρωτότητα ανήκε στην κατηγορία της υπερχείλισης καταχωρητή, που συμβαίνει όταν οδηγείται στον καταχωρητή ένα string μεγαλύτερο από το μέγεθός του. Η υπερχείλιση μπορεί να καταλήξει στο να γραφτούν δεδομένα σε άλλη περιοχή, π.χ. σε θέση σχετική με την εκτέλεση ενός προγράμματος. Ο επιτιθέμενος μπορεί να γεμίσει τον καταχωρητή με τον δικό του κώδικα και να προκαλέσει την εκτέλεση του δικού του κώδικα [Lit02]. Πρόκειται δηλαδή για μια τρωτότητα παρόμοια με εκείνη που είχε αξιοποιηθεί στην περίπτωση του Code Red (βλ. παραπάνω). Στην συγκεκριμένη περίπτωση αφορούσε στην υπερχείλιση ενός καταχωρητή στην *SQL Server 2000 Resolution Service*[34]. Η υπερχείλιση οδηγούσε στην εγγραφή δεδομένων στη μνήμη του συστήματος και, αν το string ήταν προσεκτικά επιλεγμένο, αυτό επέτρεπε στον επιτιθέμενο να τρέξει κώδικα με όσα δικαιώματα έχει η υπηρεσία

[33] `https://www.f-secure.com/v-descs/nimda.shtml`
[34] Microsoft Security Bulletin MS02-039, 24 Ιουλίου 2002.

του SQL Server. Η ανακοίνωση περιέγραφε και άλλες τρωτότητες και έδινε μια επιδιόρθωση (patch). Έξι μήνες αργότερα, τον Ιανουάριο του 2003, εμφανίσθηκε ο SQL Slammer.

Τον Αύγουστο του 2002, δηλαδή λίγο μετά την ανακοίνωση της τρωτότητας, ο David Litchfield,[35] διάσημος ειδικός στον εντοπισμό τρωτοτήτων και ιδρυτής της πιο πάνω NGS Software, πήρε μέρος στο συνέδριο *Black Hat* στο Las Vegas και επέδειξε κώδικα που θα μπορούσε να εκμεταλλευτεί την παραπάνω τρωτότητα. Η εταιρία δημοσίευσε στη συνέχεια ένα white paper σχετικά με τα ευρήματά της στον SQL Server. Στο κείμενο περιέλαβε μια περιγραφή του κώδικα που θα μπορούσε να εκμεταλλευτεί την τρωτότητα. Ο SQL Slammer θεωρήθηκε ότι προέκυψε από την αξιοποίηση αυτής της περιγραφής.

Τον Ιανουάριο του 2003 είχε περάσει ένα εξάμηνο αφότου η τρωτότητα είχε αφενός γίνει γνωστή, αφετέρου οι κάτοχοι του ευάλωτου λογισμικού έπρεπε να είχαν εγκαταστήσει τις ενημερώσεις. Ανάμεσα στα συστήματα που αυτό δεν είχε γίνει ήταν και συστήματα της ίδιας της Microsoft. Η ύπαρξη του σκουληκιού έγινε αισθητή στις 25/1/2003 από την υπερβολική κίνηση και την αναστάτωση που δημιούργησαν οι μολυσμένοι servers στο δίκτυο [Moo+03]. Διαδόθηκε στο 90% των «συμβατών» μηχανών μέσα σε δέκα λεπτά με ταχύτητα δυο τάξεις μεγέθους μεγαλύτερη από τον Code Red.

Ο Slammer δεν έφερε βλαβερό φορτίο. Ο κώδικας του Slammer είχε μήκος μόνο 376 bytes [Ray] και ήταν αφιερωμένος μόνο στη διάδοση, που ενσωματώνονται σε ένα πακέτο UDP μήκους 404 bytes (το μήκος του Code Red ήταν 4 Kbytes, ενώ του Nimda 60 Kbytes). Για την αποστολή ενός αιτήματος από ένα client σε ένα SQL server αρκεί η αποστολή ενός πακέτου UDP στη θύρα 1434 του server. Κατά συνέπεια για μια απόπειρα μόλυνσης (σάρωση) αρκεί η αποστολή ενός πακέτου. Δεδομένου ότι χρησιμοποιήθηκε το πρωτόκολλο UDP η αποστολή τέτοιων πακέτων εμποδιζόταν μόνο από την χωρητικότητα της εξερχόμενης σύνδεσης, εφόσον η ταχύτητα της μηχανής ήταν επαρκής. Μια μηχανή με σύνδεση ταχύτητας 100 Mbps είναι επομένως σε θέση να στείλει $10^8/(404 \times 8) = 30940$ πακέτα ανά sec. Οι παρατηρημένες ταχύτητες με τέτοια δεδομένα ήταν γύρω στις 26000 σαρώσεις ανά sec.

Η επίδρασή του Slammer οφειλόταν αποκλειστικά στην υπερφόρτωση του δικτύου και στην εξουδετέρωση εξυπηρετητών βάσεων δεδομένων. Ο μηχανισμός διάδοσης βασιζόταν στην τυχαία σάρωση, δηλαδή επέλεγε τυχαίες διευθύνσεις IP, με αποτέλεσμα στο τέλος να μολύνονται όλες οι συμβατές με το σκουλήκι μηχανές. Οι Moore, Paxson κ.α. [Moo+03] βρήκαν ότι η ρουτίνα παραγωγής τυχαίων αριθμών είχε ορισμένα σφάλματα με αποτέλεσμα να είναι ατελής η σάρωση των διευθύνσεων IP και να μη μολύνεται το σύνολο των συμβατών μηχανών. Περίπου μια ώρα μετά την έναρξη της επίθεσης υιοθετήθηκαν αντίμετρα φιλτραρίσματος των πακέτων UDP με θύρα προορισμού το 1434 σε δρομολογητές και firewalls. Τα μέτρα αυτά εξ αιτίας της ταχύτητας διάδοσης έφτασαν κατόπιν εορτής και είχαν ελάχιστη επίδραση στο πλήθος των μολύνσεων, αν και μετρίασαν την περαιτέρω κίνηση.

Οι κατασκευαστές του Slammer δεν έγιναν γνωστοί. Η Microsoft, της οποίας τα προϊόντα αποτελούσαν τον συνήθη στόχο επιθέσεων, άρχισε να εφαρμόζει μια τακτική επικήρυξης των δημιουργών κακόβουλου λογισμικού με προϋπολογισμό $5.000.000, προσφέροντας από $250.000 για κάθε περίπτωση, όπως για τον *Sasser*, τον *Blaster*, τον *MyDoom*, τον *Sobig*, τον *Conficker* κ.α. Ορισμένες από αυτές τις επικηρύξεις οδήγησαν στη σύλληψη των δραστών.

[35] `https://en.wikipedia.org/wiki/David_Litchfield`

Conficker

Τον Οκτώβρη του 2008 εμφανίσθηκε ο *Conficker*, γνωστός και ως *Downup*, *Downadup* και *Kido*. Εκμεταλλεύθηκε μια τρωτότητα στην υπηρεσία *Windows Server* των Windows 2000, Windows XP, Windows Vista, Windows Server 2003 και Windows Server 2008. Η τρωτότητα καλύφθηκε με την ενημέρωση MS08-067 της 23/10/2008. Ωστόσο από τον Δεκέμβριο του 2008 ως τον Απρίλιο του 2009 κυκλοφόρησαν άλλες 4 παραλλαγές του Conficker. Παρά την ανακοίνωση, το 30% των μηχανών ως τον επόμενο Ιανουάριο έμεινε χωρίς ενημέρωση του λογισμικού. Μεταξύ των μολυσμένων μηχανών ήταν και υπολογιστές του γαλλικού και του αγγλικού πολεμικού ναυτικού και του γερμανικού στρατού με αποτέλεσμα να προκύψουν επιχειρησιακά προβλήματα. Ένα και μισό χρόνο αργότερα και παρά τα μέτρα ήταν μολυσμένοι πάνω από έξι εκατομμύρια υπολογιστές με τις πρώτες δύο εκδόσεις (σύμφωνα με μια ανακοίνωση της Symantec).

Η αρχική έκδοση βασίζεται στην ίδια αρχή υπερχείλισης όπως οι Code Red και Slammer. Μέσω μιας κατάλληλα διαμορφωμένης απομακρυσμένης κλήσης διεργασίας (remote procedure call) προς τη θύρα 139 ή 445 (του υπολογιστή που πρόκειται να προσβληθεί) ένας καταχωρητής οδηγείται σε υπερχείλιση σε χώρο της μνήμης που αντιστοιχεί σε εκτελέσιμο κώδικα. Στον ήδη μολυσμένο υπολογιστή ο ιός τρέχει ένα HTTP server σε θύρα μεταξύ 1024 και 10000, ενώ ο υπό προσβολή υπολογιστής στέλνει ένα αίτημα στον HTTP server να κατεβάσει ένα αρχείο `dll` (dynamically linked library) που περιέχει τον ιό. Στη συνέχεια συνδέει το αρχείο με το `svchost.exe` για να τρέξει την συγκεκριμένη υπηρεσία [Bur08].

Σε τι ακριβώς έχει χρησιμοποιηθεί και αν έχει χρησιμοποιηθεί ο μεγάλος παραμένων πληθυσμός μολυσμένων μηχανών είναι άγνωστο. Η αμοιβή από την επικήρυξη του δημιουργού του Conficker με $250000 από την Microsoft δεν έχει ως τώρα δοθεί. Ορισμένοι βλέπουν σχέση του ρώσου χάκερ *Severa* (Peter Levashov) με τον Konficker [Sat19]. Το 2018 ο εν λόγω χάκερ δικάσθηκε στις ΗΠΑ για τα *Keihos/Waledac* και *Storm* botnets [CDB09]. και ομολόγησε την ενοχή του [Sat18].

Zeus

Πρόκειται για ένα Δούρειο ίππο γνωστό και με τα ονόματα *ZeuS, Zbot*, που υποκλέπτει τραπεζικούς κωδικούς και δημιουργεί ένα botnet. Το 2009 είχε εκτιμηθεί ότι οι μολυσμένες μηχανές μόνο στις ΗΠΑ ήταν πάνω από έξι εκατομμύρια [Ric+13], στρατολογημένες σε botnet αγνώστου πλήθους και είχαν προκαλέσει ζημιές εκατοντάδων εκατομμυρίων δολλαρίων. Ταυτόχρονα, η ανίχνευση της μόλυνσης από τα anti-virus ήταν προβληματική.

Το φορτίο ήταν προστατευμένο με συμμετρική κρυπτογράφηση RC4 (XOR του αρχικού κειμένου με ψευδοτυχαία ακολουθία) εφαρμοσμένη μετά από μετατροπή σε base-64. Το κλειδί ήταν κρυμμένο αρχικά σε συγκεκριμένη θέση της μνήμης, οπότε ήταν εύκολη η ανίχνευσή του, αλλά στη συνέχεια υπήρξε καλύτερη προστασία. Αρκετές από τις εργασίας πάνω στην άμυνα κατά του Zeus επικεντρώνονται στην αποκάλυψη του κλειδιού.

Εμφανίσθηκε στην αγορά και ως «εργαλειοθήκη» (toolkit) με τιμές $800-$4000. Τον Μάρτη του 2011 διέρρευσε ο κώδικας. Στη συνέχεια εμφανίσθηκαν ποικίλες παραλλαγές.

Τα bots στέλνουν ανά 2 min ένα πακέτο με πληροφορίες status και ανά δεκάλεπτο μια «αναφορά», με όσα στοιχεία έχει υποκλέψει το bot [Ric+13]. Η μόλυνση υλοποιεί-

ται με drive-by-downloads από μολυσμένες ιστοσελίδες, καθώς και με μολυσμένα μηνύματα σε email και κοινωνικά μέσα.

Το Zeus είχε θεωρηθεί εξ αρχής προϊόν ρώσων χάκερ [How09]. Για το Zeus, το *GameOverZeus* κ.α. καταζητείται από το FBI ο Evgeny Bogachev (ρώσος, άλλως γνωστό ως *Slavik* και *lucky12345*). Για το *SpyEye*, στο οποίο έχει ενσωματωθεί κώδικας από το Zeus, έχουν συλληφθεί και καταδικαστεί στις ΗΠΑ δύο άτομα.[36]

Shlayer

Ο *Shlayer* είναι ένας δούρειος ίππος με σκοπό την παράνομη διαφήμιση (adware) που προσβάλλει υπολογιστές mac [Bar20; Kas19; Kas20]. Για το 2019 θεωρήθηκε ως η No 1 απειλή για υπολογιστές με λειτουργικό macOS (σχεδόν το 1/3 των μολύνσεων σύμφωνα με την *Kaspersky* που τον ανίχνευσε στον 1/10 των υπολογιστών που προστατεύει) και οι μολύνσεις γίνονται ακόμη και μέσα από δημοφιλείς πλατφόρμες όπως *youTube* (όπου οι σύνδεσμοι προς το malware κρύβονται στην περιγραφή του video) και *Wikipedia* (όπου οι σύνδεσμοι κρύβονται στις παραπομπές). Ο Shlayer εγκαθιστά προγράμματα που φέρνουν διαφημίσεις αλλοιώνοντας αιτήματα των χρηστών όταν χρησιμοποιούν browsers και τροποποιώντας τις απαντήσεις προς τα αιτήματα για να εμφανίζονται ακόμη περισσότερες διαφημίσεις.

Όταν το θύμα πατήσει το link εμφανίζεται ένα αρχείο εγκατάστασης (`.dmg`), το οποίο όμως στην πραγματικότητα είναι ένα Python script. Αυτό δημιουργεί μια ταυτότητα χρήστη και διαχειριστή του συστήματος και συλλέγει πληροφορίες για την έκδοση του λειτουργικού, βάσει των οποίων κάνει ένα αίτημα `get` με το οποίο κατεβάζει το αρχείο zip που περιέχει τον Shlayer. Μετά την αποσυμπίεση προκύπτει ένα εκτελέσιμο το οποίο ενεργοποιείται με `chmod` για να έχει τα κατάλληλα δικαιώματα [Sea20].

Stuxnet

Το σκουλήκι Stuxnet [Lan11; Kus13; DHH15] είναι εξειδικευμένο λογισμικό που προσβάλλει κατ' αποκλειστικότητα συστήματα SCADA (Supervisory Control And Data Acquisition, προϊόν της Siemens) που τρέχουν πάνω από Windows. Το σύστημα SCADA χρησιμοποιείται για τη διαχείριση βιομηχανικών συστημάτων και διεργασιών. Ο Stuxnet εγκαθιστά ένα rootkit, μέσω του οποίου μπορούν να υποκλαπούν αρχεία και να αποκρυβούν οι υποκλοπές. Προσβάλλοντας τα PLC (programmable logic controllers) των φυγοκεντρικών διαχωριστών για τον εμπλουτισμό του ουρανίου στο Ιράν, οδήγησε τα μηχανήματα σε ανώμαλη και υπερβολική περιστροφή που καταλήγει σε φθορά.

Η πλειοψηφία των συστημάτων που προσβλήθηκαν παγκόσμια ανήκει στο Ιράν. Ο Stuxnet διαδίδεται μέσω Windows, αλλά ενεργοποιείται μόνον όταν είναι παρόν το SCADA, προσβάλλει περιορισμένο αριθμό μηχανών και διαγράφεται σε συγκεκριμένη ημερομηνία. Δεδομένου ότι η αρχική στόχευση ήταν συγκεκριμένη, η μόλυνση

[36] Το 2013 οι Hamza Bendelladj (αλγερινός, γνωστός και ως *Bx1*) και Aleksandr Panin (ρώσος, γνωστός και ως *Gribodemon*) προσήχθησαν σε δίκη στην Atlanta των ΗΠΑ και το 2016 καταδικάστηκαν με 15 και 9 χρόνια φυλάκισης αντίστοιχα. Η κατηγορία ήταν ότι δημιούργησαν τον δούρειο ίππο *SpyEye*, ο οποίος μόλυνε πάνω από 50 εκατομμύρια μηχανές και προκάλεσε ζημιές πάνω από ένα δισεκατομμύριο δολλάρια. Ανάμεσα σε όσα καταλογίζονταν στον Panin ήταν ότι αγόρασε τον Νοέμβρη του 2009 τον κώδικα του Zeus από τον Evgeny Bogachev και ότι ενσωμάτωσε πολλά μέρη του κώδικα στο SpyEye [Geo16]. Για τον Evgeny Bogachev υπάρχει ανοιχτή επικήρυξη του FBI για 3 εκατ. δολλάρια, όπου περιλαμβάνονται διάφορες κατηγορίες, μεταξύ αυτών και σχετικές με το Zeus. Η αμοιβή αυτή παρέμεινε η πιο ψηλή επικήρυξη του FBI ως το 2019, οπότε επικηρύχθηκε με 5 εκατομμύρια ο Maksim Yakubets.

περαιτέρω πληθυσμών ακόμη και από ένα τόσο εξειδικευμένο κακόβουλο προϊόν δείχνει πόσο δύσκολο είναι να οριοθετηθεί και να ελεγχθεί η εξάπλωση ενός ιού ή σκουληκιού.

CryptoLocker

Ο *CryptoLocker* είναι κακόβουλο λογισμικό της περιόδου 2013-14 που αποσκοπούσε στο να αποσπάσει λύτρα (ransomware). Διαδιδόταν αρχικά (από τις 5 Σεπτ. του 2013) μέσω μηνυμάτων spam email που έφεραν ένα συμπιεσμένο (zip) συνημμένο, μέσα στο οποίο κρυβόταν ένα εκτελέσιμο αρχείο [Lia+16]. Ένα μήνα αργότερα άρχισε η διάδοση μέσω του *Gameover ZeuS* botnet που ήταν απόγονος του *Zeus* (βλ. και πιο πάνω). Κρυπτογραφούσε αρχεία με τον RSA, κρατώντας το ιδιωτικό κλειδί στους servers ελέγχου του botnet. Εμφάνιζε ένα μήνυμα με απαίτηση για πληρωμή λύτρων μέσω *Bitcoin* ή *MoneyPak* εντός 72 ωρών, με την απειλή ότι θα διαγραφόταν το κλειδί μετά την παρέλευση της προθεσμίας. Αν το θύμα δεν πλήρωνε, εμφανιζόταν στο θύμα μια δήθεν ανεξάρτητη υπηρεσία αποκρυπτογράφησης με ακόμη μεγαλύτερο κόστος. Ποσοστό γύρω στο 1,3 % των θυμάτων πλήρωσε τα λύτρα και υπολογίζεται ότι η συμμορία απέσπασε γύρω στα 3 εκατομμύρια δολλάρια.

Μετά από συνδυασμένη δράση μεγάλων εταιριών λογισμικού και ασφάλειας, πανεπιστημίων, καθώς και αστυνομικών αρχών πολλών χωρών, γνωστή ως *επιχείρηση Tovar*, σταμάτησε η λειτουργία του Gameover ZeuS για δύο εβδομάδες [WB18] και κατηγορήθηκαν για την δημιουργία του Cryptolocker και του Gameover ZeuS συγκεκριμένα άτομα. Επίσης δημιουργήθηκε ένα διαδικτυακό εργαλείο μέσω του οποίου τα θύματα μπορούσαν να ανακτήσουν το ιδιωτικό κλειδί. Ωστόσο μετά από λίγο ανιχνεύθηκαν νέες προσβολές από τον Cryptolocker με την αρχική μέθοδο του spam.

Υπόθεση SolarWinds

Στο τέλος του 2020 διαπιστώθηκε μια εξαιρετικά μεγάλης κλίμακας παραβίαση δεδομένων σε κυβερνητικές υπηρεσίες των ΗΠΑ, αλλά και σε πολλές άλλες χώρες, σε διεθνείς οργανισμούς και σε εταιρίες. Η σχετική κυβερνοεπίθεση ανακαλύφθηκε τον Δεκέμβρη του 2020, αλλά φαίνεται πως είχε αρχίσει πολλούς μήνες νωρίτερα.

Τον Δεκέμβριο του 2020 εμφανίσθηκαν δημοσιεύματα [Bin20] σύμφωνα με τα οποία σημαντικοί οργανισμοί στις ΗΠΑ είχαν υποστεί σοβαρή παραβίαση δεδομένων, ενώ τις επόμενες μέρες αναφέρθηκαν ανάλογα περιστατικά για εκατοντάδες οργανισμών και μεγάλων εταιριών παγκοσμίως. Στους οργανισμούς περιλαμβάνονται το Ευρωπαϊκό Κοινοβούλιο, το NATO, η βρεττανική κυβέρνηση κ.α. Η επίθεση είχε αρχίσει το λιγότερο από τον Μάρτιο του 2020, οπότε οι επιτιθέμενοι είχαν εκμεταλλευθεί τρωτότητες λογισμικού και λογαριασμούς από τουλάχιστον τρεις εταιρίες, την *Microsoft*, την *SolarWinds* και την *VMware*. Η SolarWinds είχε στείλει στους πελάτες της μολυσμένες ενημερώσεις λογισμικού[37] τον Μάρτιο του 2020. Ο τότε υπουργός εξωτερικών των ΗΠΑ Μ. Pompeo απέδωσε την επίθεση σε ρωσικές υπηρεσίες [BBC20], ενώ παρόμοιες ανακοινώσεις εξέδωσαν οι CISA, FBI, NSA κ.α. Η Ρωσική κυβέρνηση αρνήθηκε την ανάμιξή της. Οι επιθέσεις αυτές θεωρήθηκαν πράξεις κατασκοπείας και όχι δολιοφθοράς, επειδή υπήρξε μόνο πρόσβαση στα δεδομένα, αλλά όχι αλλοίωσή τους.

[37] `https://www.solarwinds.com/sa-overview/securityadvisory`

Παραβίαση δεδομένων του MS Exchange Server

Τον Μάρτιο του 2021 η Microsoft έβγαλε μια σειρά από ενημερώσεις του λογισμικού του *Microsoft Exchange Server* [Sea21] που περιλαμβάνουν ως και την έκδοση του 2013 για να διορθώσει μια σειρά από τρωτότητες, των οποίων η εκμετάλλευση από την πλευρά της επίθεσης είχε ξεκινήσει ήδη στις αρχές Ιανουαρίου. Στις επιθέσεις αυτές χρησιμοποιήθηκαν 4 τρωτότητες μηδενικής μέρας.

Με την ανακοίνωση των επιδιορθώσεων το πρόβλημα έγινε ευρέως γνωστό, οπότε οι επιθέσεις πολλασιάστηκαν με στόχο όσα μηχανήματα δεν είχαν περάσει τις ενημερώσεις. Η Microsoft απέδωσε τις αρχικές επιθέσεις σε κινεζικό group (που η ίδια έχει ονομάσει *Hafnium*),[38] στη συνέχεια όμως οι πάντες έσπευσαν να επωφεληθούν, περιλαμβανομένων τουλάχιστον δέκα ομάδων APT. Μέχρι τις αρχές Μαρτίου είχαν προσβληθεί 250000 servers, ενώ σε ήδη προσβεβλημένους servers άρχισε η εγκατάσταση ransomware. Μεταξύ άλλων προσβλήθηκε η κατασκευάστρια υπολογιστών *Acer* και της ζητήθηκαν λύτρα 50 εκατομμυρίων δολλαρίων [Abr21] αν τα λύτρα πληρωθούν ως τις 28/3, ενώ στη συνέχεια το ποσό διπλασιάζεται.

[38] https://www.microsoft.com/security/blog/2021/03/02/hafnium-targeting-exchange-servers/

ΚΕΦΑΛΑΙΟ 6

Επιθέσεις άρνησης υπηρεσίας

> I'm often asked, "Should I study forensics or cryptography or network security or protocols or embedded devices or SCADA systems?" Study what you want. Follow whatever interests you, because what you're really learning is how to think like a security expert.
>
> Bruce Schneier, [Sev16].

6.1 Τι είναι οι επιθέσεις άρνησης υπηρεσίας

Οι επιθέσεις άρνησης υπηρεσίας (Denial of Service attacks, DoS attacks) αποσκοπούν στο να γίνει πολύ δύσκολη ως αδύνατη η παροχή μιας υπηρεσίας. Το πιο συνηθισμένο μέσο επίθεσης είναι η αποστολή υπερβολικής κίνησης στους δικτυακούς ή άλλους πόρους που χρειάζεται η υπηρεσία. Αυτού του είδους η επίθεση είναι δημοφιλής εξ αιτίας κυρίως της ευκολίας της. Ο επιτιθέμενος δεν απαιτείται να κάνει πολύπλοκες ενέργειες διείσδυσης, ούτε χρειάζεται να γνωρίζει σε βάθος το σύστημα-θύμα.

Η υποβάθμιση της ποιότητας μιας υπηρεσίας είναι ένα γενικότερο ζήτημα, που δεν αφορά μόνο την περιοχή της ασφάλειας, αλλά κυρίως την περιοχή της *ποιότητας υπηρεσιών* (service quality). Εν γένει μια υπηρεσία προδιαγράφεται, σχεδιάζεται και υλοποιείται έτσι ώστε να παρέχεται μέσα σε ένα εκ των προτέρων γνωστό ή εκτιμώμενο περιβάλλον με συγκεκριμένη ποιότητα. Η τελευταία κατά κανόνα περιγράφεται από μια σειρά δεικτών που πρέπει να παραμένουν μέσα σε δεδομένα όρια, εφόσον τουλάχιστον η υπηρεσία συνεχίζει να λειτουργεί και δεν έχει καταρρεύσει. Η απώλεια ποιότητας, δηλαδή η υπέρβαση των ορίων, κατά την παροχή της υπηρεσίας μπορεί να οφείλεται σε αστοχίες σχεδιασμού και υλοποίησης, διαθεσιμότητας πόρων κατά τη λειτουργία της και μεταβολή των συνθηκών μέσα στις οποίες λειτουργεί. Μπορεί επομένως να οφείλεται σε μείωση της διαθεσιμότητας υπολογιστικών, αποθηκευτικών και δικτυακών πόρων ή και σε αυξημένη ζήτηση, που ξεφεύγει από τα όρια και τις

ανοχές που έχει προβλέψει ο σχεδιαστής της υπηρεσίας. Όλα αυτά όμως εν γένει δεν είναι απαραίτητο να οφείλονται σε κακόβουλες ενέργειες και σε μια καλοσχεδιασμένη υπηρεσία θα οδηγήσουν σε περιορισμένη ή σταδιακή απώλεια ποιότητας.

Το ζήτημα της πρόθεσης να προξενηθεί βλάβη είναι κεντρικό στις επιθέσεις άρνησης υπηρεσίας, αλλά ένας προσεκτικός ερευνητής θα μπορούσε να θέσει το θέμα ότι δεν είναι πάντοτε ξεκάθαρο κατά πόσο μια ενέργεια μπορεί να χαρακτηρισθεί εμπρόθετη ή βλαπτική. Για παράδειγμα, αν ο γείτονας έχει ένα κλειδωμένο WiFi και καταφέρω να του σπάσω το συνθηματικό και να το χρησιμοποιήσω κάνοντας περιστασιακή και χαμηλού όγκου χρήση, η ζημιά που προκαλώ είναι σχεδόν μηδενική και πιθανότατα ανεπαίσθητη, αλλά αναμφίβολα έχω παραβιάσει εκ προθέσεως το δικαίωμά του να διαφυλάξει το ασύρματο δίκτυό του. Αν πάλι το δίκτυο είναι αφύλακτο και κατεβάσω ταινίες, η ζημιά μπορεί να είναι μεγαλύτερη, αλλά είναι αμφίβολο αν μπορεί να αποδειχθεί κακόβουλη πρόθεση εκ μέρους μου.

Όλες αυτές οι λεπτές διακρίσεις έχουν ελάχιστη πρακτική αξία για τις επιθέσεις άρνησης υπηρεσίας, δεδομένου ότι εκεί η πρόθεση του επιτιθέμενου να βλάψει το θύμα είναι σαφής και τα μέσα που χρησιμοποιούνται είναι κάθε άλλο παρά διακριτικά, αλλά αντίθετα είναι επιλεγμένα έτσι ώστε να μεγιστοποιούν τη ζημιά.

Οι επιθέσεις αυτές μπορούν να ενταχθούν στη γενικότερη στρατηγική ενός επιτιθέμενου στα πλαίσια μεμονωμένων ενεργειών ή μιας γενικότερης στρατηγικής κυβερνοπολέμου (cyber warfare [AW11]). Όμως ως επί το πλείστον είναι «καταστροφικού» χαρακτήρα και με την έννοια αυτή είναι παρόμοιες με τις κλασσικές στρατιωτικές επιθέσεις στις υποδομές του αντιπάλου, σε αντίθεση με πιο «διακριτικές» επιθέσεις που αποσκοπούν στην απόσπαση πληροφοριών χωρίς το θύμα να τις αντιλαμβάνεται. Με την έννοια αυτή είναι ο διάδοχος του «ηλεκτρονικού πολέμου» (electronic warfare [And08]) προηγουμένων δεκαετιών, που είχε ως βασικό μέσο την παρεμβολή ικανού θορύβου στις επικοινωνίες και τα ραντάρ.

Μια γενική διχοτόμηση των επιθέσεων άρνησης υπηρεσίας είναι ανάμεσα σε (α) επιθέσεις στη *λογική* και (β) επιθέσεις σε *πόρους* [Moo+06]. Η διαίρεση αυτή κατά κάποιο τρόπο είναι στην ίδια γραμμή σκέψης με τη διαίρεση των ιδιοτήτων ενός συστήματος σε λειτουργικές (functional) και μη λειτουργικές (non functional). Οι πρώτες είναι αυτές που σχετίζονται με την αλγοριθμική λογική της λειτουργίας του συστήματος και αναλύονται με εργαλεία όπως τα πεπερασμένα αυτόματα και οι τυπικές γλώσσες. Αν το σύστημα πέσει κατά τύχη (μέσα από μια σειρά απρόβλεπτων περιστατικών) ή οδηγηθεί (μέσα από μια ακολουθία εντολών) στη λάθος κατάσταση, αυτό είναι αρκετό για να δημιουργηθεί σοβαρό πρόβλημα ή και να «κρασάρει» εντελώς (να πέσει σε deadlock). Οι μη λειτουργικές ιδιότητες είναι αυτές που έχουν να κάνουν με τις επιδόσεις του συστήματος, πόσο γρήγορο είναι, σε πόσο φορτίο αντέχει κ.ο.κ. Με άλλα λόγια οι επιθέσεις στη λογική είναι επιθέσεις στις λειτουργικές ιδιότητες του συστήματος ενώ οι επιθέσεις πόρων είναι επιθέσεις στις επιδόσεις του, δηλαδή στις μη λειτουργικές ιδιότητες. Φυσικά οι πρώτες, αν πρόκειται να σταματήσουμε ένα σύστημα, είναι πιο αποτελεσματικές και πιο γρήγορες, αλλά προϋποθέτουν λεπτομερέστερη γνώση του συστήματος και καλύτερη πρόσβαση σ᾽ αυτό από τον επιτιθέμενο, δηλαδή προϋποθέσεις που είναι πιο δύσκολο να ικανοποιούνται. Στην περίπτωση αυτή μια επίθεση στους πόρους είναι πιο απλή και μπορεί αργά ή γρήγορα να φέρει σχεδόν το ίδιο αποτέλεσμα.

Αν θέλουμε να είμαστε αυστηροί στην κατάταξή μας, δεν θα πρέπει να βιαστούμε να βγάλουμε το συμπέρασμα ότι όλες οι επιθέσεις που βασίζονται σε εντολές που στέλνει ο επιτιθέμενος στο θύμα είναι επιθέσεις λογικής, διότι πολλές απ᾽ αυτές έχουν σκοπό την αυξημένη κατανάλωση πόρων. Επομένως μπορεί να είναι και πάλι επιθέσεις στις επιδόσεις του θύματος μέσω της κατανάλωσης επιλεγμένων πόρων. Για πα-

ράδειγμα, μια ερώτηση σε μια βάση δεδομένων που φέρνει βαριά αναζήτηση στη βάση και επεξεργασία δεδομένων είναι προφανώς επίθεση στις επιδόσεις.

Όποιος έχει σκοπό να επιτεθεί σε πόρους προφανώς βάζει σε ψηλότερη προτεραιότητα τους πιο κρίσιμους απ' αυτούς, δηλαδή αυτούς που εφόσον καταστούν μη διαθέσιμοι θα δημιουργήσουν σοβαρό πρόβλημα στην παροχή μιας υπηρεσίας. Αλλά και πάλι η πρόσβαση στους κρίσιμους πόρους μπορεί να μην είναι εφικτή. Εφόσον πρόκειται για επιθέσεις από απόσταση, ο επιτιθέμενος αναγκαστικά κάνει χρήση της δικτυακής υποδομής και συχνά η επίθεση εκδηλώνεται χρησιμοποιώντας ως μέσο την τελευταία, εφόσον το δίκτυο αποτελεί πόρο ουσιώδη για την ομαλή παροχή της υπηρεσίας. Μια δημοφιλής κατηγορία επιθέσεων δεν είναι κάτι άλλο από το πλημμύρισμα του δικτύου με άχρηστη κίνηση σε σημεία πιο βολικά για τη δημιουργία συμφόρησης.

Με τις επιθέσεις κίνησης ο επιτιθέμενος δεν χρειάζεται να έχει ιδιαίτερες γνώσεις για τα τρωτά σημεία του συστήματος, στο οποίο θα επιτεθεί. Δεν χρειάζεται καν να έχει λεπτομερή πληροφόρηση για τα κρίσιμα σημεία του δικτύου. Αν στείλει αρκετή κίνηση, αυτή μόνη της θα συσσωρευθεί στα πιο στενά σημεία. Πόση; Κι αυτό μπορεί να ρυθμιστεί χωρίς ιδιαίτερη ακρίβεια κάνοντας δοκιμές. Αν η κίνηση που έστειλε ως τώρα δεν ήταν αρκετή, η σωστή κατεύθυνση είναι να την αυξήσει και θα αποτύχει μόνο αν οι δυνατότητές του για παραγωγή κίνησης δεν είναι αρκετές. Γι' αυτό το λόγο η έμφαση της πλευράς της επίθεσης, όσο μάλιστα η χωρητικότητα των δικτύων αυξάνεται, είναι στην κατεύθυνση της επινόησης μεθόδων που θα επιτρέψουν να δημιουργηθεί και να διευθυνθεί προς το θύμα όλο και μεγαλύτερη κίνηση.

Η δημιουργία έντονης κίνησης βασίζεται κατά κανόνα σε λίγες βασικές ιδέες που μπορούν να χρησιμοποιηθούν σε συνδυασμούς: (α) Στην εξάντληση των δυνατοτήτων μιας μηχανής για ταχεία παραγωγή κίνησης, (β) στη δημιουργία πολλαπλασιαστικών φαινομένων μεταξύ εισερχόμενης και εξερχόμενης κίνησης σε ένα σύστημα που λειτουργεί ως μεγεθυντικός φακός και (γ) στην ικανότητα του επιτιθέμενου να συστρατεύσει για το σκοπό του ένα πληθυσμό από μηχανές που βρίσκονται υπό τον έλεγχό του.

Για παράδειγμα εάν μια εντολή μήκους M B (bytes) προς ένα υπολογιστή έχει ως αποτέλεσμα την αποστολή ενός αρχείου μήκους N B, γίνεται πολλαπλασιασμός της κίνησης κατά παράγοντα N/M. Αν αυτό μπορεί να γίνει από πολλούς υπολογιστές ταυτόχρονα και οι απαντήσεις να διευθυνθούν όλες στο θύμα, το τελευταίο θα κληθεί να παραλάβει μεγάλη κίνηση σε μικρό χρονικό διάστημα. Η κίνηση αυτή μπορεί να προσβάλει τη δυνατότητα επεξεργασίας και μνήμης του θύματος ή να γεμίσει το δίκτυο τοπικής πρόσβασής του και να εμποδίσει έτσι την επικοινωνία του με τον έξω κόσμο. Συνδυασμός από τέτοιες τεχνικές μπορεί να δημιουργήσει ισχυρό ρεύμα κίνησης της τάξης του Tbps (10^{12} bits/sec) [arb15].

6.2 Ταξινόμηση των επιθέσεων

Μια επίθεση άρνησης υπηρεσίας βέβαια έχει ως στόχο τους πόρους ενός συστήματος, με σκοπό να επιτευχθεί η δραστική υποβάθμιση των παρεχόμενων από το σύστημα υπηρεσιών, αλλά γενικότερα οι επιθέσεις τέτοιου είδους μπορούν να ταξινομηθούν κατά διάφορους τρόπους, ανάλογα με την οπτική γωνία αυτού που τις εξετάζει. Μια δημοφιλής ταξινόμηση έχει να κάνει με το στόχο της επίθεσης, κυρίως αν ο στόχος είναι το δίκτυο ή πόροι αυτής καθαυτής της υπηρεσίας στις μηχανές που την υποστηρίζουν. Ποικίλες ταξινομήσεις των επιθέσεων άρνησης υπηρεσίας [MR04] περιλαμβάνουν τον βαθμό αυτοματισμού, την στοχευόμενη τρωτότητα, τη χρήση ή όχι αλλοιωμένων διευθύνσεων, τον ρυθμό μετάδοσης (σταθερό, αυξανόμενο, μετα-

βαλλόμενο), τη δυνατότητα χαρακτηρισμού από τις επικεφαλίδες των πακέτων, τη μεταβλητότητα των μηχανών που συμμετέχουν στην επίθεση, τον τύπο του θύματος, τον τύπο των ζημιών κ.λπ.

Ταξινόμηση ως προς το στόχο

Όλα τα είδη των επιθέσεων είναι ανοιχτά σε καινοτόμες ιδέες του κάθε επιτιθέμενου, σε τρωτότητες που έχουν ίσως βρει άλλοι αλλά έχουν έρθει σε γνώση του κ.ο.κ., άρα ποτέ δεν μπορούν να αποκλεισθούν νέα και άγνωστα ως τώρα είδη επιθέσεων. Μπορούμε όμως να κάνουμε μια κατάταξη με ως τώρα γνωστά δεδομένα και στη βάση του στόχου κάθε επίθεσης. Ούτως ή άλλως οι πόροι τους οποίους απασχολεί μια υπηρεσία είναι εξυπηρετητές, δίκτυα και το λογισμικό της ίδιας της υπηρεσίας και της πλατφόρμας που χρησιμοποιεί. Εφόσον η υπηρεσία σχετίζεται με κάποια πλευρά της πραγματικότητας έχει κι εκεί κάποιο αποτύπωμα, δηλαδή μπορεί να παίρνει δεδομένα από αισθητήρες, κάμερες, έξυπνα αντικείμενα, καθώς και να τους στέλνει πίσω εντολές και πληροφορίες. Μια επίθεση μπορεί να έχει ως στόχο οποιαδήποτε απ᾽ αυτές τις συνιστώσες και κατά προτίμηση τις πιο κρίσιμες.

Επιθέσεις στη χωρητικότητα του δικτύου

Οι περισσότεροι εξυπηρετητές, πάνω στους οποίους τρέχει μια υπηρεσία (ή μέρος της), συνδέονται με το δίκτυο ενός παρόχου με ζεύξεις σχεδιασμένες έτσι, ώστε να επαρκούν για τη συνήθη ζήτηση της υπηρεσίας, φυσικά και με κάποιες ανοχές και πρόβλεψη για το μέλλον. Ωστόσο το μέρος του δικτύου που παρέχει αυτήν την τοπική πρόσβαση του εξυπηρετητή στο ευρύτερο δίκτυο του παρόχου είναι περιορισμένης χωρητικότητας σε σύγκριση με τη συνολική μεταφορική ικανότητα του δικτύου του παρόχου. Όλη η κίνηση από και προς τον εξυπηρετητή είναι καταδικασμένη να περάσει μέσα από αυτό το μέρος του δικτύου, που αποτελεί και σημείο στένωσης (μποτιλιαρίσματος), οπότε η μέγιστη δυνατή κίνηση που θα περάσει στη μια ή την άλλη κατεύθυνση προσδιορίζεται από τη χωρητικότητα της ελάχιστης τομής[1].

Αν και στένωση, δηλ. ελάχιστη τομή, υπάρχει προς αμφότερες τις κατευθύνσεις (όχι υποχρεωτικά η ίδια), πιο «ενδιαφέρουσα» είναι η κατεύθυνση που οδηγεί από τον εξωτερικό κόσμο προς τον εξυπηρετητή, επειδή ο εξωτερικός κόσμος έχει τη δυνατότητα αποστολής κίνησης κατά τεκμήριο μεγαλύτερης από την ελάχιστη τομή. Οι ζεύξεις της ελάχιστης τομής έχουν ως σημείο εισόδου δρομολογητές, όπου θα συσσωρευθούν πακέτα στην περίπτωση που η συνολική ροή που διοχετεύεται στην εν λόγω τομή είναι μεγαλύτερη από τη χωρητικότητά της. Μάλιστα μια λιγότερο έξυπνη δρομολόγηση μπορεί να οδηγήσει σε συμφόρηση με κίνηση μικρότερη από τη χωρητικότητα της ελάχιστης τομής. Εν πάση περιπτώσει οι δρομολογητές που θα υποφέρουν από συμφόρηση αρχικά μπορεί να προσπαθήσουν να αποθηκεύσουν και να καθυστερήσουν πακέτα, αργότερα όμως είναι πιθανό να αναγκασθούν να κάνουν επιλεκτική καταστροφή τους. Σε τέτοια κατάσταση συμφόρησης μπορεί να οδηγηθεί ένα δίκτυο ακόμη και σε σχεδόν κανονικές συνθήκες, αρκεί να υπάρξει υπερβολική ζήτηση μιας υπηρεσίας που περιστασιακά γίνεται πολύ δημοφιλής ή κάποιοι υποχρεώνονται να

[1] Τομή ανάμεσα από τον κόμβο A στον κόμβο B ενός γράφου είναι μια συλλογή ακμών που αν αφαιρεθούν από το γράφο δεν υπάρχει μονοπάτι από τον A στον B. Αν οι ακμές χαρακτηρίζονται από «χωρητικότητα», δηλαδή ένα μέτρο της μέγιστης κίνησης που μπορεί να εξυπηρετήσει κάθε μια, κάθε τομή έχει «χωρητικότητα τομής» το άθροισμα των χωρητικοτήτων των ακμών της. Ελάχιστη τομή είναι η τομή με την ελάχιστη χωρητικότητα. Το θεώρημα μέγιστης ροής - ελάχιστης τομής (max flow - min cut) λέει πως η μέγιστη κίνηση που μπορεί να διοχετευθεί από τον A στον B είναι ίση με την χωρητικότητα της ελάχιστης τομής.

την χρησιμοποιήσουν σε ελάχιστο χρόνο μέχρι μια στενή προθεσμία (για παράδειγμα σχεδόν όλοι οι ψηφοφόροι επιθυμούν να μάθουν πού ψηφίζουν πρακτικά τη μέρα των εκλογών μέσα από την ιστοσελίδα του υπουργείου εσωτερικών κι όλοι οι υποψήφιοι φοιτητές επισκέπτονται το υπουργείο παιδείας τη μέρα των αποτελεσμάτων των εξετάσεων εισαγωγής στα ΑΕΙ). Στην περίπτωση αυτή κάποιοι από τους χρήστες της υπηρεσίας θα αποτύχουν στη χρήση της γιατί π.χ. δεν θα φτάσουν ποτέ στην αντίστοιχη ιστοσελίδα ή η υπηρεσία θα «κρεμάσει» ξαφνικά στη μέση της χρήσης της.

Παρόμοια υπερβολική κίνηση μπορεί να διοχετευθεί σε μια υπηρεσία από κακόβουλους χρήστες. Οι κανονικοί χρήστες εκείνη την ώρα θα υποφέρουν όλο και περισσότερο εφόσον αυξάνεται η κίνηση που στέλνουν οι κακόβουλοι χρήστες. Κάποια στιγμή οι διαχειριστές της υπηρεσίας, εφόσον αντιληφθούν ότι υπάρχει πρόβλημα, προσπαθώντας να το λύσουν είναι πιθανό να σταματήσουν την παροχή της όλωσδιόλου.

Το εν λόγω είδος επίθεσης θα μπορούσε να θεωρηθεί κάπως χοντροκομμένο με την έννοια ότι βασίζεται σε μια απλή ιδέα, αυτήν του υπερβολικού φορτίου πάνω στο δίκτυο που χρησιμοποιεί η υπηρεσία για την επικοινωνία της με τους χρήστες της. Η ρύθμιση του όγκου της επίθεσης είναι επίσης σχετικά απλή, αν δηλαδή το θύμα δεν φαίνεται να υποκύπτει το μόνο που έχουμε να κάνουμε είναι να αυξήσουμε την κίνηση. Αν είμαστε ως επιτιθέμενοι τυχεροί και το σύστημα έχει κι άλλες αδυναμίες, το αποτέλεσμα μπορεί να επιτευχθεί με χαμηλότερη κίνηση επειδή πριν από τις ζεύξεις έχουν προλάβει να εξαντληθούν κάποιοι άλλοι πόροι, αλλά ίσως να μη χρειαστεί ποτέ να μάθουμε ποιοι είναι. Αν όμως αυτούς τους τελευταίους πόρους μπορούσαμε να τους ανιχνεύσουμε, επιλέξουμε και στοχεύσουμε, ίσως τότε η επίθεση να είχε ακόμη πιο χαμηλές απαιτήσεις κίνησης.

Ίσως όμως δεν πρέπει να αδικήσουμε εντελώς τους «καλλιτέχνες» εκείνους που οργανώνουν μια επίθεση βασισμένη στην κίνηση, επειδή η επίτευξη ακραίων τιμών της κίνησης μπορεί να είναι κι αυτή μια ιδιαίτερη πρόκληση. Θα δούμε αργότερα, ειδικά εξετάζοντας τις κατανεμημένες επιθέσεις άρνησης υπηρεσίας, ποιες είναι οι τεχνικές προκλήσεις στη δημιουργία πολύ υψηλής κίνησης.

Στοχευμένη επίθεση

Η επίθεση με παραπλανητικό πακέτο SYN (SYN spoofing attack) είναι μια στοχευμένη επίθεση στο πνεύμα των παρατηρήσεων της προηγούμενης παραγράφου. Ο επιτιθέμενος δημιουργεί σ' ένα δρομολογητή την εντύπωση ότι ανοίγουν συνεχώς νέες συνδέσεις, οι οποίες καταγράφονται σε ειδικό πίνακα. Σκοπός του είναι να φτάσει τον πίνακα στα όριά του (quota). Την επίθεση αυτή θα περιγράψουμε παρακάτω. Ωστόσο η λογική είναι παρόμοια σε μια σειρά τέτοιων στοχευμένων επιθέσεων, που αποσκοπούν στην εξάντληση συγκεκριμένων πόρων ενός συστήματος, όπως είναι καταχωρητές και πίνακες με ανοιχτές συνδέσεις.

Μια άλλου τύπου επίθεση βασίζεται στην εκμετάλλευση σφαλμάτων που υπάρχουν στο δικτυακό λογισμικό. Στην περίπτωση αυτή η λήψη ενός πακέτου («δηλητηριώδους» πακέτου, *poison packet*) που ενεργοποιεί τον εσφαλμένο κώδικα μπορεί να οδηγήσει σε αδιέξοδο το σύστημα, το οποίο χρειάζεται πλέον επανεκκίνηση για να επανέλθει σε ομαλή λειτουργία.

Επίθεση σε πόρους των εφαρμογών

Η επίθεση αυτή βασίζεται συνήθως στο να γίνει προς την εφαρμογή ένα αίτημα που δεν είναι εύκολο να ικανοποιηθεί χωρίς ισχυρή κατανάλωση πόρων. Για παρά-

δειγμα, σε μια βάση δεδομένων μπορεί να σταλεί μια πολύπλοκη αναζήτηση. Σε μια υπηρεσία γεωγραφικών πληροφοριών μπορούμε να ζητήσουμε να βρει κατάλληλη δρομολόγηση για πολλούς και απομακρυσμένους προορισμούς (π.χ. μια ευμεγέθη περίσταση προβλήματος περιοδεύοντος πωλητή). Επίσης μπορεί να γίνει επίθεση του τύπου που αναφέρθηκε στην προηγούμενη παράγραφο, δηλαδή βασισμένη σε σφάλματα του λογισμικού, ώστε το σύστημα να οδηγηθεί σε αδιέξοδο ή άσκοπες επαναλήψεις.

Η επίθεση σε πόρους μιας εφαρμογής μπορεί να είναι αρκετά εκλεπτυσμένη, ώστε να αυξάνονται οι πιθανότητες επιτυχίας της. Στην επίθεση που ονομάζεται *Cyberslam* [Kan05] δίνεται έμφαση στο να μη μπορεί να διακρίνει το θύμα τον επιτιθέμενο από τον κανονικό χρήστη. Ένα δίκτυο ανδρεικέλων (botnet) έχει προηγουμένως δημιουργηθεί προσβάλλοντας μια συλλογή από μηχανές και μετατρέποντας τις τελευταίες σε zombies. Όταν τους δοθεί η εντολή να αφυπνισθούν, αυτές στέλνουν στο θύμα από ένα πακέτο που περιέχει ένα μικρό αίτημα, το οποίο όμως για να ικανοποιήσει ο εξυπηρετητής πρέπει να διαθέσει πόρους (sockets, καταχωρητές TCP κ.λπ.), να ψάξει μια βάση δεδομένων και πιθανώς να φέρει σε συμφόρηση κάποιο άλλο σημείο του συστήματος. Πώς θα μπορούσε κανείς να αποφύγει μια τέτοια επίθεση; Μια μέθοδος είναι να ζητήσει από τον αιτούντα να λύσει αινίγματα που αποδεικνύουν ότι είναι άνθρωπος και όχι μηχανή, π.χ. να αναγνωρίσει από μια σειρά εικόνων με φαγητά ποια περιέχουν πατάτες. Στη συνέχεια ο επιτιθέμενος μπορεί να στοχεύσει αυτόν τον ίδιο το μηχανισμό αναγνώρισης.

Ταξινόμηση ως προς τη δικτυακή γνώση-θέση του επιτιθέμενου

Η ταξινόμηση αυτή είναι κυρίως σχετική με επιθέσεις που βασίζονται στην αλλοίωση δικτυακών διευθύνσεων και περιλαμβάνει τους εξής τύπους επιθέσεων:

Τυφλές επιθέσεις οι οποίες προέρχονται από επιτιθέμενο που δεν είναι στο ίδιο τοπικό δίκτυο με το στόχο ή τη χρησιμοποιούμενη πηγή της επίθεσης ή εν πάση περιπτώσει δεν είναι στη διαδρομή των δεδομένων μεταξύ της πηγής και του στόχου [RFC 6959].

Μη τυφλές επιθέσεις που βασίζονται σε μηχανισμούς λαθρακρόασης (eavesdropping) πάνω σε συνδέσεις ή σε άλλου είδους πειρατεία συνδέσεων. Ο επιτιθέμενος έχει πρόσβαση σε πληροφορίες ενός ρεύματος και τις χρησιμοποιεί για να εξαπολύσει την επίθεσή του. Η *επίθεση του ενδιάμεσου* (man in the middle) είναι μια ενδεικτική τέτοια επίθεση.

Με το θέμα αυτό θα ασχοληθούμε πιο κάτω, αλλά για την ώρα θα μελετήσουμε την πιο κλασσική περίπτωση άρνησης υπηρεσίας, δηλαδή την επίθεση με πλημμύρα.

6.3 Η αλλοίωση της διεύθυνσης

Η αλλοίωση της διεύθυνσης προέλευσης ενός πακέτου είναι ένα γενικότερο ζήτημα, αλλά στην περίπτωση των επιθέσεων κίνησης γίνεται μια πολύτιμη προσθήκη στη συλλογή εργαλείων του επιτιθεμένου με πλημμύρα κίνησης. Πριν λίγες δεκαετίες η αλλοίωση διεύθυνσης ήταν το βασικό εργαλείο με το οποίο ο επιτιθέμενος αφενός παραπλανούσε το θύμα, αφετέρου παρέσυρε σε συνέργεια άλλες μηχανές.

Η σημασία και η έκταση της χρήσης αλλοιωμένων διευθύνσεων έχει μειωθεί με την άνοδο της χρήσης τεράστιων δικτύων ανδρεικέλων (botnets). Επί πλέον η επέκταση του φιλτραρίσματος πακέτων κοντά στην πηγή σε ποσοστό 80% (2015, βλ. [Kwo+15])

έχει επίσης συντελέσει στον περιορισμό του προβλήματος. Ωστόσο το φιλτράρισμα που δεν γίνεται στο υπόλοιπο 20% του δικτύου σε συνδυασμό με νέες τρωτότητες και αντίστοιχες τεχνικές που εκμεταλλεύονται την αλλοίωση διεύθυνσης διαιωνίζουν το πρόβλημα.

Πώς υλοποιείται

Οι συσκευές και τα πρωτόκολλα που λειτουργούν σ᾽ ένα δίκτυο παράγουν πακέτα, των οποίων η επικεφαλίδα και το περιεχόμενο εξαρτώνται από τις λειτουργίες που επιτελούν, την κατάσταση στην οποία βρίσκονται κ.λπ. Με την έννοια αυτή δημιουργείται ίσως η εντύπωση ότι ο μόνος τρόπος να δημιουργήσει κάποιος πακέτα με αλλοιωμένη διεύθυνση ή άλλα στοιχεία είναι να «εξαπατήσει» μια συσκευή. Ωστόσο η πραγματικότητα είναι διαφορετική, δεδομένου ότι υπάρχουν δικτυακοί μηχανισμοί που επιτρέπουν πρακτικά τη δημιουργία οποιουδήποτε είδους πακέτων για λόγους πραγματοποίησης δοκιμών. Ωστόσο κανείς δε μπορεί να εγγυηθεί ότι οι μηχανισμοί αυτοί θα χρησιμοποιηθούν μόνο σε μια τέτοια περίσταση.

Το Internet Protocol (IP) [TW13; Koz05; Gor09] ανήκει στο στρώμα 3 και χρησιμοποιεί αυτόνομα πακέτα (datagrams), των οποίων η επικεφαλίδα περιέχει τη *διεύθυνση προέλευσης* (source address, 4 bytes στο IPv4) και τη *διεύθυνση προορισμού* (destination address, 4 bytes στο IPv4). Κάθε πακέτο δρομολογείται ανεξάρτητα. Ωστόσο ακριβώς πιο πάνω υπάρχει το στρώμα 4, το οποίο φροντίζει για την οργάνωση των πακέτων που ανταλλάσσονται ανάμεσα σε δυο άκρα αναλόγως των αναγκών. Τα βασικά πρωτόκολλα αυτού του στρώματος είναι το *Transmission Control Protocol (TCP)* και το *User Datagram Protocol (UDP)*. Η επικεφαλίδα του TCP περιέχει μεταξύ άλλων και το *πεδίο ακολουθίας* (`Sequence Number`), καθώς και το *πεδίο επιβεβαίωσης* (`Acknowledgement Number`). Τα δύο αυτά πεδία πρέπει επίσης να ληφθούν υπόψη όταν σε επίθεση με αλλοίωση διεύθυνσης εμπλέκεται το TCP [Tan03].

Πώς θα γίνει όμως η αλλοίωση της διεύθυνσης; Η δημοφιλής μέθοδος είναι μέσω του raw IP socket.

Υποδοχές (Sockets)

Πριν φτάσουμε στο raw IP socket ας θυμηθούμε τι είναι και τι συμβαίνει με τα sockets [Gor09].

Οι περισσότεροι ξενιστές (hosts[2]) διαθέτουν μόνο μια δικτυακή σύνδεση που ελέγχεται από μια διεργασία (process) που υλοποιεί τη στοίβα TCP/IP. Η διεργασία αυτή (Internet Daemon, γνωστή ως `inetd, xinetd, launchd` κ.λπ. ανάλογα με το λειτουργικό σύστημα) αφουγκράζεται όλη τη δραστηριότητα που προκαλείται από το TCP/IP πάνω σε μια διεπαφή. Ωστόσο διαφορετικές εφαρμογές έχουν ανάγκη επικοινωνίας και για να τις ικανοποιήσουν χρησιμοποιούν όλες μαζί την ίδια δικτυακή διεπαφή με χρήση πολυπλεξίας. Η επικοινωνία ανάμεσα σε δυο πλευρές γίνεται μέσω συγκεκριμένων θυρών (ports, που χαρακτηρίζονται από αριθμούς μήκους 16 bits) και περιλαμβάνονται στην επικεφαλίδα των τμημάτων του UDP ή TCP (segments, δηλ. των πακέτων επιπέδου TCP ή UDP). Η υποδοχή (socket) είναι ακριβώς ο συνδυασμός μιας διεύθυνσης IP με ένα αριθμό θύρας TCP/IP.

Με τη χρήση των υποδοχών (sockets) ως ζευγών διευθύνσεων-θυρών ένας πελάτης (client), στον οποίο τρέχουν π.χ. δύο διεργασίες που επικοινωνούν με τον ίδιο

[2] Η βάση τηλεπικοινωνιακών όρων του ΟΤΕ (`http://www.moto-teleterm.gr`) δίνει τη λέξη *ξενιστής* ως μετάφραση της αγγλικής λέξης *host*. Επίσης δίνει την *υποδοχή* ως μετάφραση του όρου *socket*.

εξυπηρετητή (server), ο τελευταίος μπορεί να τις διακρίνει εξ αιτίας της χρήσης διαφορετικών υποδοχών, π.χ. ως `147.102.7.100:14901` και ως `147.102.7.100:14905`. Ομοίως αν ένας εξυπηρετητής διαθέτει περισσότερες από μια διεπαφές δικτυακής πρόσβασης (δηλ. έχει συγχρόνως δύο διευθύνσεις `147.102.7.1` και `147.102.7.2`), μπορεί π.χ. να τις στείλει στην ίδια διεργασία FTP, βλέποντας ότι αμφότερες απευθύνονται στην ίδια θύρα, δηλαδή έχουν αριθμούς υποδοχής `147.102.7.1:22` και `147.102.7.2:22`.

Η υποδοχή αποτελεί την προγραμματιστική διεπαφή των στρωμάτων TCP/IP. Ο προγραμματιστής π.χ. στο Unix διαθέτει εντολές με τις οποίες δημιουργεί, διαγράφει, ανοίγει και κλείνει ένα socket, ακριβώς όπως θα έκανε με ένα αρχείο, χωρίς να τον απασχολούν θέματα επικοινωνίας, δηλαδή χωρίς να χρειάζεται να φροντίσει για τις σχετικές συνδέσεις. Ο προγραμματιστής μπορεί να ανοίξει υποδοχή σε οποιοδήποτε από τα δύο στρώματα, δηλαδή είτε προς το TCP (stream socket) ή UDP (datagram socket)[CD08], είτε απ' ευθείας προς το IP. Η υποδοχή απ' ευθείας προς το IP ονομάζεται «ακατέργαστη υποδοχή» (raw socket).

Η ακατέργαστη υποδοχή IP

Στην περίπτωση των υποδοχών TCP και UDP οι αρμοδιότητες του προγραμματιστή ως προς τα πεδία ενός πακέτου που ελέγχει μέσω της υποδοχής είναι περιορισμένες. Ωστόσο μέσω της ακατέργαστης υποδοχής IP ο έλεγχος επεκτείνεται πρακτικά σε όλα τα πεδία. Αυτό σημαίνει ότι ο προγραμματιστής είναι σε θέση να κατασκευάσει παντός είδους πακέτα ελέγχοντας όλες τις παραμέτρους, περιλαμβανομένων των διευθύνσεων πηγής και προορισμού. Έχει δηλαδή τη δυνατότητα να χρησιμοποιήσει και ψευδείς διευθύνσεις πηγής. Άρα μπορεί κάποιος μέσω της ακατέργαστης υποδοχής να εξαπολύσει επιθέσεις με αλλοιωμένες διευθύνσεις, καθώς επίσης και να κάνει επιθέσεις με χρήση τμημάτων (segments) TCP SYN, να δημιουργήσει ψευδείς συνδέσεις TCP κ.λπ. (βλ. ένα παράδειγμα σύνθεσης πακέτου με το raw socket στο [Bal07]).

Έχουν εκφρασθεί πολλές αμφιβολίες ως προς την αναγκαιότητα της ακατέργαστης υποδοχής και κατά πόσο αποτελεί ένα επικίνδυνο εργαλείο στα χέρια των χάκερ. Ωστόσο η ακατέργαστη υποδοχή είναι απαραίτητη σε ορισμένες περιπτώσεις, π.χ. δεν θα μπορούσαμε να γράψουμε πρωτόκολλο ICMP/IGMP με κανονικά sockets, όπως επίσης δεν θα μπορούσαμε να γράψουμε ένα δικό μας νέο πρωτόκολλο στη θέση των TCP, UDP, ICMP, IGMP (Internet Group Management Protocol[3]). Μια άλλη απάντηση που δόθηκε στο ίδιο ερώτημα είναι ότι η πλευρά της επίθεσης είχε ούτως ή άλλως ήδη στη διάθεσή της αυτές τις δυνατότητες με άλλους τρόπους. Η λύση που επικρατεί αυτή τη στιγμή είναι να δίνεται πρόσβαση στην ακατέργαστη υποδοχή μόνο σε χρήστες με αυξημένα δικαιώματα.

6.4 Botnets

Το *διαδικτυακό ανδρείκελο* (Internet bot, web robot) είναι ένα σχετικά αυτόνομο πρόγραμμα που αναλαμβάνει για λογαριασμό του χρήστη του τη διεκπεραίωση διαφόρων εργασιών. Για παράδειγμα *twitterbot* είναι ένα bot που βάζει μηνύματα στο Twitter, επαναλαμβάνει μηνύματα, παρακολουθεί άλλους χρήστες, επαναπροωθεί μηνύματα κ.ο.κ. Μολονότι μια οποιαδήποτε ομάδα από bots μπορεί να οργανωθεί σε δίκτυο, ο όρος *botnet* χρησιμοποιείται σχεδόν αποκλειστικά με κακή χροιά και υποδηλώ-

[3] Το IGMP είναι ένα πρωτόκολλο που χρησιμοποιείται από ξενιστές και γειτονικούς δρομολογητές για να σχηματισθούν ομάδες πολυεκπομπής (multicast) και αποτελεί μέρος του IP multicast.

νει δίκτυο (network) από (ro)bots που εκτελούν κακόβουλες ενέργειες. Ο δημιουργός και εκμεταλλευτής ενός τέτοιου δικτύου δεν χρησιμοποιεί δικούς του υπολογιστές, αν μη τι άλλο γιατί ένα δίκτυο χιλιάδων υπολογιστών θα απαιτούσε μια πολύ μεγάλη δαπάνη, την οποία αποφεύγει χρησιμοποιώντας λαθραία υπολογιστές που ανήκουν σε άλλους. Συνήθως ένα κακόβουλο δίκτυο ανδρεικέλων αποτελείται από μολυσμένους υπολογιστές, γνωστούς και ως ζόμπι (zombies), επειδή το κακόβουλο λογισμικό γενικά είναι σε απόκρυψη και ύπνωση μέχρι τη στιγμή που θα ενεργοποιηθεί με κατάλληλες εντολές από ένα κεντρικό ελεγκτή. Σε εξελιγμένα δίκτυα οι ελεγκτές είναι οργανωμένοι σε περισσότερα στρώματα ιεραρχίας.

Αρχιτεκτονική και λειτουργία

Τοπολογία

Ένα botnet είναι κατά βάση μια συλλογή από μηχανές, οι οποίες ελέγχονται από ένα άνθρωπο χειριστή, γνωστό ως botmaster. Μολονότι στο μέλλον ίσως θα μπορούσαμε να δούμε botnets που θα έχουν επικεφαλής μια μηχανή και θα είναι σχετικά αυτόνομα, ο ρόλος του botmaster περιλαμβάνει αρμοδιότητες στρατηγικών αποφάσεων, προώθησης στην αγορά κ.λπ. που ακόμη δεν είναι δυνατό να εκχωρηθούν σε μια μηχανή. Από την άλλη πλευρά όμως σε ένα δίκτυο με πλήθος ανδρεικέλων χρειάζεται υποστήριξη από μηχανές για να επιτευχθεί μαζική και γρήγορη επικοινωνία ανάμεσα στον botmaster και στα bots. Επίσης χρειάζεται να παραμένει η επικοινωνία αυτή συγκαλυμμένη, πράγμα που σημαίνει ότι πρέπει να γίνεται μέσα από κανάλια κατά το δυνατόν αθέατα στα θύματα και στα όργανα του νόμου.

Ο δημοφιλής όρος για τα σήματα που ανταλλάσσονται μεταξύ κυρίου και ανδρεικέλων είναι C&C (command and control). Στην περίπτωση αυτή μιλάμε για επικοινωνίες και κανάλια C&C. Συχνά όλες αυτές οι επικοινωνίες καταλήγουν σε ένα κεντρικό εξυπηρετητή, δηλαδή σε ένα C&C server. Το μοντέλο που ως τώρα περιγράψαμε έχει προφανώς μόνο δύο στρώματα, (α) του botmaster με το C&C server και (β) των μηχανών που λειτουργούν ως ανδρείκελα (bots). Αν ο πληθυσμός των τελευταίων είναι μεγάλος μπορεί να προστεθεί ένα ακόμη ενδιάμεσο στρώμα από μηχανές, που θα διευκολύνουν την πιο ομαλή, αλλά και πιο κρυφή, επικοινωνία μεταξύ botmaster και bots.

Προετοιμασία

Η ζωή μιας μηχανής ως μέλους ενός botnet αρχίζει με την προσβολή της από κακόβουλο λογισμικό με οποιαδήποτε πρόσφορη μέθοδο. Ο κώδικας που εγκαθίσταται περιλαμβάνει τη διεύθυνση ενός ή περισσότερων εξυπηρετητών, με τους οποίους πρέπει να επικοινωνήσει η μηχανή. Οι διευθύνσεις αυτές μπορεί να είναι προεγκατεστημένες σε μια λίστα στον κακόβουλο κώδικα, εναλλακτικά όμως μπορεί να προσδιορίζεται ένα σύνολο διευθύνσεων, στις οποίες η νέα μηχανή στέλνει μηνύματα κάνοντας τυχαίες επιλογές ή με μια σειρά μέχρι να βρει τη σωστή. Το πλεονέκτημα της τελευταίας μεθόδου είναι ότι κάνει το botnet πιο δύσκολο στην ανακάλυψη εφόσον εντοπισθεί ο κώδικας. Μια άλλη ιδέα που έχει υλοποιηθεί είναι να αναζητήσει η νέα μηχανή τον κατάλογο σε ένα ουδέτερο εξυπηρετητή σε πραγματικό χρόνο, μόλις δηλαδή χρειαστεί να επικοινωνήσει. Εν πάση περιπτώσει η φάση αυτή όπου σχηματίζεται η ομάδα των bots που θα συναποτελέσει το botnet είναι γνωστή ως φάση *εγγραφής* στο botnet.

Κυρίως λειτουργία

Στην κυρίως λειτουργία τα μέλη ενός botnet συνεργάζονται για την διεξαγωγή επιθέσεων.

Επικοινωνία Βασικό εργαλείο για τη λειτουργία του botnet είναι η εσωτερική επικοινωνία. Τα μέλη του δικτύου πρέπει να δέχονται εντολές από τον botmaster και να αναφέρουν σχετικά με την επιτυχία της εκτέλεσής τους (πρόκειται για το γνωστό στρατιωτικό μοντέλο). Το πρόβλημα επιλογής πρωτοκόλλου και μέσου επικοινωνίας θα έβρισκε ίσως απλή λύση αν δεν υπήρχε η απαίτηση όλα αυτά να γίνονται με μυστικότητα. Οι σχεδιαστές δικτύων τέτοιου τύπου έχουν δοκιμάσει μια σειρά από μέσα επικοινωνίας, εκ των οποίων τα πιο δημοφιλή είναι τα κανάλια IRC, οι υπηρεσίες ιστού (HTTP) και τα πρωτόκολλα ισοτίμων (P2P, peer-to-peer).

Τα μέλη του δικτύου για επιτύχουν τους σκοπούς του είναι επίσης υποχρεωμένα να επικοινωνούν με μηχανές και υπηρεσίες έξω από αυτό. Για παράδειγμα, όταν θέλουν να μάθουν τη διεύθυνση IP μιας μηχανής επικοινωνούν με ένα DNS server. Μπορεί να γίνουν μέλη ενός γενικότερου (και αθώου) δικτύου P2P μόνο και μόνο για να μεταφέρουν πληροφορίες.

Απόκρυψη Σε ένα botnet χρησιμοποιούνται μηχανισμοί απόκρυψης του λογισμικού και των επικοινωνιών του. Για το λογισμικό οι μέθοδοι είναι λίγο ως πολύ οι ίδιες με εκείνες της απόκρυψης οποιουδήποτε κακόβουλου λογισμικού. Για τις επικοινωνίες χρησιμοποιείται εναλλαγή καναλιών, κανάλια IRC, επικοινωνίες P2P, κρυπτογράφηση των πληροφοριών κ.ο.κ.

Επιθέσεις Ένα δίκτυο ανδρεικέλων έχει κατά κανόνα ως σκοπό ύπαρξης την διεξαγωγή επιθέσεων διαφόρων τύπων και εν τέλει τη δημιουργία οικονομικού οφέλους για τον επιτιθέμενο και τους πελάτες του και ζημίας για τα θύματα. Προφανώς επιθέσεις κατάλληλες για διεξαγωγή μέσω botnet είναι εκείνες που απαιτούν όγκο ή/και συνδυασμένες ενέργειες. Ο όγκος μπορεί να είναι απαραίτητος σε μια επίθεση άρνησης υπηρεσίας, αλλά και σε μια στατιστική αναζήτηση θυμάτων, που το καθένα χωριστά έχει μικρή πιθανότητα να υποκύψει, αλλά αν σαρωθεί μεγάλος πληθυσμός το αποτέλεσμα μπορεί με μεγάλη πιθανότητα να είναι ικανοποιητικό (με βάση το νόμο των μεγάλων αριθμών). Ως παράδειγμα της τελευταίας περίπτωσης μπορεί να δει κανείς την αποστολή μηνυμάτων spam με τα οποία ο παραλήπτης καλείται να κάνει login σε μια ιστοσελίδα που ψευδώς παριστάνει την τράπεζά του. Τα botnets έχουν σε μεγάλο βαθμό εμπλακεί σε επιθέσεις που παλιότερα χρησιμοποιούσαν αλλοίωση διεύθυνσης.

Δημοφιλείς χρήσεις είναι επιθέσεις άρνησης υπηρεσίας, επιθέσεις παραπομπής σε παραπλανητικές ιστοσελίδες (στις οποίες τα θύματα φτάνουν επειδή ο επιτιθέμενος έχει προσβάλει ένα μεγάλο αριθμό από proxy servers), μαζική αποστολή κακόβουλων και παραπλανητικών μηνυμάτων, υποκλοπή στοιχείων από τις προσβεβλημένες μηχανές, προσέλκυση θυμάτων σε υπηρεσίες με μεγάλο κόστος (π.χ. πρόκληση τηλεφωνικών κλήσεων σε ειδικές υπηρεσίες) κ.λπ.

Μέθοδοι άμυνας

Ανίχνευση

Οι διαθέσιμες τεχνικές ανίχνευσης της λειτουργίας δικτύων ανδρεικέλων βασίζονται σε δίκτυα προσέλκυσης και παγίδευσης των κακόβουλων συστημάτων (honeynets),

στις παρατηρούμενες ανωμαλίες κίνησης, καθώς και σε γνωστές περιπτώσεις, για τις οποίες διατηρούνται βάσεις δεδομένων.

Η ανίχνευση βασίζεται γενικά στον κύκλο ζωής ενός δικτύου ανδρεικέλων, δηλαδή προσπαθεί να ανιχνεύσει συμπτώματα που εκδηλώνονται σε διάφορες φάσεις της ύπαρξής του, από την ανάπτυξη του λογισμικού, την στρατολόγηση υπολογιστών και την διαφήμιση των υπηρεσιών του botnet προς πιθανούς πελάτες ως την τελική επίθεση σε στόχους.

Ανίχνευση μέσω διείσδυσης Εδώ χρησιμοποιείται η παλιά καλή ιδέα της παραπλάνησης του αντιπάλου και διείσδυσης στο δίκτυό του, γνωστή από τα κατασκοπικά δίκτυα.

Οι Cremonini και Riccardi [CR09] ανέπτυξαν λογισμικό, το οποίο, εφόσον εγκατασταθεί σε ένα υπολογιστή, έχει τη δυνατότητα να παραπλανήσει ένα botnet έτσι ώστε να καταστήσει τον υπολογιστή μέλος του. Στη συνέχεια συλλέγει πληροφορίες για τη συμπεριφορά και την τοπολογία του botnet.

Οι Stone-Gross, Cavallaro κ.α. κατάφεραν να αποκτήσουν τον έλεγχο του Torpig botnet (που είναι προσανατολισμένο σε τραπεζικούς λογαριασμούς και πιστωτικές κάρτες) παρεμβάλλοντας ένα δικό τους C&C server [Sto+09].

Ανίχνευση βάσει γνωστής συμπεριφοράς Το σύστημα *BotHunter* [Gu+07] καταγράφει ύποπτες ενέργειες, όπως σάρωση θυρών της επιτηρούμενης μηχανής, αποτυχημένες συνδέσεις προς και από αυτήν, κατανομή των διευθύνσεων που σαρώνονται από τη μηχανή. Καταγράφει επίσης και αναλύει το φορτίο (payload) των εισερχομένων πακέτων προκειμένου να διαπιστώσει αν διαφέρουν από τα συνήθως αναμενόμενα. Τέλος χρησιμοποιεί το Snort με κανόνες που αντιστοιχούν στην ανίχνευση γνωστού κακόβουλου λογισμικού, παρμένους από σχετικές βάσεις δεδομένων. Οι κανόνες αυτοί αφορούν π.χ. στην επικοινωνία μεταξύ του κέντρου ελέγχου του δικτύου και των μελών του, στη μεταφορά εκτελέσιμου κώδικα και στην ανίχνευση τρωτοτήτων. Τα παραπάνω στοιχεία συσχετίζονται και αξιολογούνται προκειμένου να εκτιμηθεί η πιθανότητα να συμμετέχει το επιτηρούμενο σύστημα σε ένα δίκτυο ανδρεικέλων ή να υφίσταται επίθεση.

Ανίχνευση μόνο από τη μορφή της κίνησης Η επικοινωνία μεταξύ μηχανών και μάλιστα αυτών που εμπλέκονται σε ένα botnet δημιουργεί ρεύματα κίνησης με χαρακτηριστικά που μπορούν να αξιοποιηθούν για τους σκοπούς της ανίχνευσης. Κατά συνέπεια ορισμένα συστήματα ανίχνευσης εξετάζουν την κίνηση μόνο εξωτερικά, δηλαδή χωρίς να αναλύουν το περιεχόμενο των μηνυμάτων και πακέτων που ανταλλάσσονται. Αν μια επίθεση κίνησης γίνεται σχετικά εύκολα φανερή λόγω του όγκου της, άλλου είδους επιθέσεις ή η επικοινωνία συντονισμού των ανδρεικέλων διαφέρουν σε πιο «λεπτά» χαρακτηριστικά τους, π.χ. στο χρονισμό των μηνυμάτων. Κίνηση αυτού του είδους μπορεί να είναι χαμηλού όγκου, μπορεί να προέρχεται από λίγα ανδρείκελα μέσα στην επιτηρούμενη περιοχή και εν τέλει μπορεί να είναι και κρυπτογραφημένη. Όλες αυτές οι ιδιότητες αυξάνουν τη δυσκολία ανίχνευσης.

Μια εισαγωγή στις μεθόδους ανίχνευσης μιας επίθεσης μέσω των ανωμαλιών που δημιουργούνται στην κίνηση δίνεται περαιτέρω στην ενότητα 8.3.

Η οικονομική πλευρά

Το προς διάθεση προϊόν

Γενικά ένα botnet αποσκοπεί στη δημιουργία οικονομικού οφέλους. Όμως το όφελος κατά κανόνα δημιουργείται με ταυτόχρονη βλάβη του θύματος. Αυτό μπορεί να γίνει π.χ. κάνοντας επιθέσεις άρνησης υπηρεσίας, συλλέγοντας προσωπικά δεδομένα και μάλιστα συνθηματικά για πρόσβαση σε τραπεζικούς λογαριασμούς, στέλνοντας διαφημιστικά μηνύματα για λογαριασμό πελατών κ.ο.κ. Το FBI ανακοίνωσε το 2012 ότι εξ αιτίας επιθέσεων με botnets δημιουργήθηκαν οικονομικές ζημίες μεγαλύτερες από 20 εκατομμύρια δολλάρια [RMG13]. Στην πλευρά των «ωφελημένων» (επειδή τέτοια δεδομένα συλλέγονται από γνωστές υποθέσεις που καταλήγουν σε καταδίκες) ένας 21-χρονος απέκτησε πάνω από 100.000 δολλάρια από διαφημιστικές εταιρίες, τις οποίες εξυπηρέτησε εμφυτεύοντας κακόβουλο διαφημιστικό κώδικα σε περισσότερους από 400.000 υπολογιστές.

Σύμφωνα με μια παλιότερη εκτίμηση (2007) το ένα τέταρτο των ξενιστών (hosts) στο Διαδίκτυο είχαν προσβληθεί και αποτελούσαν μέλη ενός δικτύου ανδρεικέλων. Στο τελευταίο τρίμηνο του 2014 έγιναν περί τις 26000 κατανεμημένες επιθέσεις άρνησης υπηρεσίας (Distributed Denial of Service, DDoS) με botnets κι άλλες 23000 στο πρώτο τρίμηνο του 2015 [Kasp15]. Ενώ οι περισσότερες επιθέσεις δεν ξεπερνούν το 24ωρο, υπήρξαν και επιθέσεις που κράτησαν έξι μέρες. Η κύρια μορφή των επιθέσεων ήταν SYN DDoS και HTTP DDoS. Οι επιθέσεις έγιναν σε 76 χώρες, αλλά η Κίνα, οι ΗΠΑ και ο Καναδάς είναι οι πρώτες χώρες τόσο σε επιτιθέμενους όσο και σε θύματα, γεγονός που εξηγείται από το γεγονός ότι οι χώρες αυτές είναι κυρίαρχες στο χώρο της φιλοξενίας υπηρεσιών ιστού παρέχοντας τις χαμηλότερες τιμές. Σύμφωνα με εκτιμήσεις του Kaspersky Lab μία και μόνη επίθεση μπορεί να κοστίσει σε μια εταιρία παραπάνω από 400.000 δολλάρια [Kasp15b] (σε ζημιές μόνο, εξαιρουμένων των επιθέσεων για λύτρα, βλ. ενότητα 5.5).

Με δυο λόγια τα botnets αποτελούν οικονομικό αγαθό διακινούμενο μέσα σε μια παράνομη αγορά και μπορεί να πάρει τη μορφή υπηρεσίας (service) ή προϊόντος (product). Αρχικά ένα botnet προσφέρει υπηρεσίες στον ιδιοκτήτη του, αλλά στη συνέχεια οι υπηρεσίες αυτές μπορούν να προσφερθούν σε πελάτες. Ο κώδικας ή και το ίδιο το botnet μπορεί να πωληθεί ως προϊόν. Για τη χρήση έχει δημιουργηθεί ένα τιμολόγιο υπηρεσιών [Nam09]: Μια επίθεση κίνησης κοστίζει από $50 ως χιλιάδες δολλάρια ημερησίως. Ένα αρχείο με 1000000 διευθύνσεις email λιγότερα από 100 δολλάρια, ενώ η αποστολή spam σε 1000000 διευθύνσεις λιγότερο από 200 δολλάρια. Ένα καλό botnet kit κοστίζει γύρω στα 1000 δολλάρια.

Ανάλογα με τη χρήση ένα botnet οργανώνεται διαφορετικά. Αν πρόκειται να κάνει επιθέσεις πλημμύρας είναι σημαντικό να διαθέτει μεγάλο αριθμό μελών. Αν πρόκειται να υποκλέψει στοιχεία είναι πιο σημαντικό να κρατηθεί στην αφάνεια. Ίσως το πιο γνωστό και επικίνδυνο botnet είναι το *Zeus* [HG15] που εξειδικεύεται στην υποκλοπή τραπεζικών δεδομένων. Η υποκλοπή στον υπολογιστή-θύμα γίνεται κυρίως με προγράμματα που παρακολουθούν την πληκτρολόγηση (keyloggers) και τη συμπλήρωση στοιχείων σε φόρμες (form grabbers).

Επιχειρηματικό μοντέλο

Ένα botnet είναι μια ιδιωτική (συνήθως) επιχείρηση, γύρω από την οποία κινούνται διάφοροι *παίκτες* εκτός από τον botmaster, με σύνθετες σχέσεις ανάμεσά τους και χωριστά συμφέροντα. Κάποιοι από αυτούς τους παίκτες δημιουργούνται από την αποσύνθεση αρμοδιοτήτων που αρχική ανήκαν στον botmaster, αλλά στη συνέχεια με

την αύξηση του μεγέθους του δικτύου και της αντίστοιχης επιχείρησης είναι πιθανό να αποδοθούν σε διαφορετικά άτομα.

Το άτομο (ή η ομάδα ατόμων) που θα σχεδιάσει και θα γράψει τον κώδικα είναι προφανώς ο *εκπονητής λογισμικού* (developer), που δεν είναι απαραίτητο να ταυτίζεται με τον botmaster επειδή ο τελευταίος μπορεί να έχει αναθέσει σε άλλον την υλοποίηση ή να έχει αγοράσει το λογισμικό. Σήμερα ανθεί η αγορά των κακόβουλων έτοιμων πακέτων (malware kits) με τα οποία μπορεί κανείς εύκολα να συνθέσει το δικό του botnet.

Ένας άλλος προφανής παίκτης είναι ο *πελάτης*. Αν και τίποτε δεν εμποδίζει τον botmaster να κάνει μόνος του επιθέσεις με το δικό του δίκτυο και να προσπορίζεται οφέλη, π.χ. να εκβιάζει επιχειρήσεις, θα ανοίξει τους ορίζοντές του αν αποφασίσει να δέχεται αιτήματα από πελάτες και να υλοποιεί επιθέσεις για λογαριασμό τους. Οι τελευταίοι μπορεί να μην έχουν καμμιά επιθυμία να γίνουν οι ίδιοι botmasters για τους ίδιους χοντρικά λόγους που κάποιος αναθέτει σε ένα κακοποιό ένα συμβόλαιο, γιατί δηλαδή είναι μια βρώμικη δουλειά με την οποία δε θέλει να αναμιχθεί και γιατί είναι κάτι που το έχει ανάγκη περιστασιακά ή μόνο μια φορά. Οι πελάτες τέτοιου είδους μπορεί απλώς να θέλουν να βλάψουν τον ανταγωνιστή τους (κλείνοντας π.χ. το ηλεκτρονικό του κατάστημα για μερικές ώρες και βλάπτοντας τη φήμη του) ή να μάθουν τα μυστικά του.

Βεβαίως πάντοτε υπάρχει το *θύμα*, ένα ή περισσότερα. Η επιλογή τους μπορεί να είναι στοχευμένη, στη βάση των επιθυμιών του πελάτη, ή στατιστική, δηλαδή να προσβληθούν χιλιάδες ώστε στατιστικά να προκύψουν οφέλη από ικανοποιητικό αριθμό (π.χ. να σαρωθούν χιλιάδες υπολογιστών για εύκολα συνθηματικά ώστε να βρεθούν κάποια).

Ένας άλλος παίκτης που δεν περνάει απ᾽ το μυαλό του μέσου χρήστη υπολογιστή είναι ο *παθητικός συνεργός*. Στον άχαρο αυτό ρόλο μπορεί να εξωθηθεί οποιοσδήποτε από εμάς είναι κάτοχος ενός υπολογιστή, στον οποίο έχει εγκατασταθεί κακόβουλο λογισμικό ενός botnet. Παρ᾽ όλο που εν γένει δεν αναμένεται να ζητάει ο τοπικός νόμος ευθύνες από τον τυχόντα χρήστη επειδή ήταν αμελής στη χρήση του υπολογιστή του, ο τελευταίος μπορεί να εμπλακεί άθελά του σε δυσάρεστες υποθέσεις και να πρέπει να αποδείξει την αθωότητά του. Σε ένα εταιρικό περιβάλλον η αμέλεια μπορεί να είναι πιο άμεσα τιμωρήσιμη.

Ο κύκλος ζωής ενός botnet δεν είναι πολύ διαφορετικός από εκείνον οποιουδήποτε συστήματος λογισμικού, δηλαδή περιλαμβάνει φάσεις όπως σύλληψη της ιδέας, σχεδιασμό (με ή χωρίς τυπικές μεθόδους), υλοποίηση, προώθηση προϊόντος και διανομή, κανονική λειτουργία (δηλαδή διεξαγωγή επιθέσεων), αναβαθμίσεις, απόσυρση.

Ένα παράδειγμα: Mirai

Το Mirai είναι κακόβουλο λογισμικό που μολύνει μηχανές Linux, κυρίως οικιακούς routers και δικτυωμένες κάμερες, και τις στρατολογεί σε ένα botnet. Πιθανώς το όνομα προέρχεται από την ιαπωνική σειρά manga με τίτλο *Mirai Nikki*, δηλαδή *Μελλοντικό Ημερολόγιο*. Ανιχνεύθηκε πρώτη φορά το 2016. Προσπαθεί να διαδοθεί σε μηχανές που διατηρούν το εργοστασιακό όνομα χρήστη και κωδικό (user name - password).[4]

Η μολυσμένη μηχανή αρχικά προσπαθεί να μολύνει άλλες μηχανές. Ανιχνεύει στέλνοντας πακέτα TCP SYN σε τυχαίες διευθύνσεις. Αν πάρει απάντηση δοκιμάζει τα δια-

[4] Τέτοιοι συνδυασμοί είναι root/xc3511, admin/-, root/catch99, default/default, default/-, guest/12345.

θέσιμα ζευγάρια ονομάτων - κωδικών μέσω telnet. Στη συνέχεια η μολυσμένη μηχανή επικοινωνεί με το κέντρο ελέγχου του botnet.

Το Mirai μπορεί να εκτελέσει επιθέσεις με διάφορες μεθόδους [Win21]. Εκτός από τις πλημμύρες DNS, SYN, ACK, UDP, HTTP (βλ. και πιο κάτω), διαθέτει και τα εξής είδη επιθέσεων:

Επίθεση στη Valve Source Engine δηλαδή επίθεση σε servers της *Valve Corporation* που εξυπηρετούν παιχνίδια online.

Επίθεση στο TCP STOMP Εκμεταλλεύεται το Streaming Text Oriented Messaging Protocol (STOMP), ένα πρωτόκολλο παρόμοιο με το HTTP μεταξύ ενός client και ενός server. Μετά από την αρχική χειραψία οι επιτιθέμενοι κόμβοι στέλνουν μια πλημμύρα ACK μέσα στη σύνοδο που έχει δημιουργηθεί.

GREETH/GREIP - πλημμύρα GRE Ethernet Εκμεταλλεύονται το Generic Routing Encapsulation (GRE), ένα πρωτόκολλο σήραγγας της *Cisco* που μπορεί να ενθηλακώσει διάφορα πρωτόκολλα επιπέδου δικτύου μέσα σε εικονικές ζεύξεις πάνω από ένα δίκτυο IP.

Τον Σεπτ. του 2016 ο κώδικας του Mirai δημοσιεύθηκε από τους δημιουργούς του. Το αποτέλεσμα ήταν η δημιουργία πολλαπλών αντιγράφων, παραλλαγών και απογόνων. Το Mirai botnet θεωρείται και σήμερα το πιο επικίνδυνο για την δημιουργία επιθέσεων DDoS. Οι μηχανές που έχουν προσβληθεί από το Mirai και τις παραλλαγές του (Corona, Retard, Unstable, Ares) πέρασαν το ένα εκατομμύριο το 2018 και τα δύο εκατομμύρια το 2019, ενώ υπολογίζονται στα 2300000 το 2020 [Arb20].

Μεταξύ 2014 και 2016 έγινε μια σειρά επιθέσεων στο πανεπιστήμιο Rutgers σε επιλεγμένες κρίσιμες ημερομηνίες, με αποτέλεσμα τη διακοπή της επικοινωνίας του με τον έξω κόσμο, την διακοπή εγγραφών σε μαθήματα και γενικά την παρεμπόδιση υπηρεσιών κ.λπ. Το πανεπιστήμιο ξόδεψε παραπάνω από ένα εκατομμύριο δολλάρια για την ασφάλειά του και δήλωσε αύξηση διδάκτρων για να καλύψει τα έξοδα. Τον Σεπτ. του 2016 το Mirai χρησιμοποιήθηκε για ισχυρές επιθέσεις κίνησης, μεταξύ των οποίων μια επίθεση στην ιστοσελίδα του δημοσιογράφου Brian Krebs που εργαζόταν στην περιοχή της ασφάλειας (*Krebs on Security*) με ένταση 620 Gbit/s (ο δημοσιογράφος αυτός τελικά απεκάλυψε τον δράστη το 2017) και μια επίθεση ενός Tbit/s στον web host *OVH*. Τον Οκτ. του 2016 έγινε μια σειρά επιθέσεων σε *GitHub, Twitter, Reddit, Netflix, Airbnb* κ.α.

Τον Δεκ. του 2017 ο 21χρονος Paras Jha, από το New Jersey ομολόγησε μια σειρά από ενέργειες, στις οποίες περιλαμβάνονται οι πολυήμερες επιθέσεις στο δίκτυο του πανεπιστημίου Rutgers [US 17]. Επίσης οι Josiah White Jha από την Pennsylvania, ετών 20, και Dalton Norman από τη Louisiana, ετών 21, ομολόγησαν οτι δημιούργησαν και λειτούργησαν το Mirai botnet. Οι τρεις τους δημοσίευσαν τον κώδικα. Αντιμετώπιζαν αρχικά ποινές φυλάκισης μέχρι δέκα έτη, χρηματικό πρόστιμο $250000 και αποζημιώσεις. Τελικά καταδικάσθηκαν σε μόλις 6 μήνες περιορισμού κατ᾽ οίκον (δηλαδή παραμονή με τους γονείς και απαγόρευση εξόδου), 5 χρόνια επίβλεψης, 2500 ώρες κοινωνικής εργασίας και αποζημιώσεις ύψους $127000. Οι ποινές αυτές ήταν εξαιρετικά μικρές εξ αιτίας της συνεργασίας των κατηγορουμένων με το FBI στην εξιχνίαση άλλων υποθέσεων botnet. Εκεί διοχετεύθηκαν οι περισσότερες από τις παραπάνω ώρες κοινωνικής εργασίας.

Ο Paras Jha ήταν φοιτητής στο Computer Science του Rutgers [Kre18] και ομολόγησε ότι έκανε την πρώτη επίθεση για να καθυστερήσει την εγγραφή άλλων συμφοιτητών του σε ένα μάθημα, στο οποίο επεδίωκε να εγγραφεί. Είπε επίσης ότι έκανε

τη δεύτερη επίθεση για να καθυστερήσει την εξέτασή του σε μάθημα μαθηματικών. Επίσης προσπάθησε να πουλήσει στο Rutgers λογισμικό προστασίας από DDoS ως ιδιοκτήτης εταιρίας παροχής τέτοιου λογισμικού. Τελικά ο Jha παράτησε τις σπουδές του.

Περαιτέρω καταδικάσθηκαν άλλοι για χρήση αντιγράφων και παραλλαγών του Mirai. Ένας εξ αυτών ήταν ο Daniel Kaye, ετών 29, υπεύθυνος για την παραλλαγή Satori του Mirai, που μεταξύ άλλων προκάλεσε τη μόλυνση 900000 routers της Deutsche Telekom. Το 2018 καταδικάστηκε ο σχετικά άπειρος χάκερ Kenneth Currin Schuchman, ετών 20, στο Anchorage για παράνομη εισβολή σε υπολογιστές. Όλοι αυτοί αποτελούν προφανώς τους πιο άτυχους και την κορυφή του παγόβουνου.

6.5 Επιθέσεις πλημμύρας

Οι επιθέσεις με πλημμύρα πακέτων (flood attacks) αποσκοπούν προφανώς στο να φτάσουν σε κατάσταση συμφόρησης δικτυακοί πόροι, όπως ζεύξεις, καταχωρητές δρομολογητών, πίνακες συνδέσεων κ.α. Η πιο απλή τέτοια επίθεση είναι αυτή που γίνεται με πακέτα `ping`.

Η υλοποίηση επιθέσεων με πλημμύρα στηρίζεται σε μια σειρά από ιδέες, τεχνικές και τρωτότητες που χρησιμοποιούνται σε διάφορους συνδυασμούς. Μερικοί τρόποι με τους οποίους δημιουργείται ισχυρό ρεύμα κίνησης είναι οι εξής:

Ανάκλαση Ο επιτιθέμενος εκμεταλλεύεται ένα σύνολο από άλλες μηχανές, οι οποίες στη συνέχεια στέλνουν κίνηση στο θύμα. Για παράδειγμα, αν τους στείλει μηνύματα με αλλοιωμένη τη διεύθυνση προέλευσης και ίση με τη διεύθυνση του θύματος, οι άλλες μηχανές θα στείλουν απαντήσεις στη διεύθυνση προέλευσης.

Ενίσχυση Δημιουργείται οποτεδήποτε μια μηχανή διεγείρεται από ένα μήνυμα-αίτημα και απαντάει με μεγαλύτερο όγκο κίνησης από αυτόν που αντιστοιχεί. Για παράδειγμα ενίσχυση δημιουργείται αν σταλεί ένα αίτημα αναζήτησης σε μια βάση δεδομένων και η απάντηση συνίσταται σε μεγάλο όγκο στοιχείων.

Botnets Χρησιμοποιείται ένα σύνολο μηχανών-ανδρεικέλων που κατευθύνονται από τον επιτιθέμενο στην πραγματοποίηση μιας επίθεσης.

Με το δεδομένο ότι οι περισσότερες επιθέσεις χρησιμοποιούν περισσότερους από ένα από τους παραπάνω μηχανισμούς είναι δύσκολο να κατατάξει κανείς τις επιθέσεις π.χ. σε καθαρές επιθέσεις ενίσχυσης ή επιθέσεις ανάκλασης.

Πλημμύρα ping

Η πλημμύρα με ping είναι μια επίθεση που γίνεται στέλνοντας στο θύμα πακέτα `ICMP Echo Request`, κοινώς `ping`.

Τι είναι το ICMP Echo Request

Ας θυμηθούμε τι είναι τα πακέτα ICMP Echo Request. Το Internet Protocol (IP) είναι το βασικό δικτυακό πρωτόκολλο που παρέχει τον μηχανισμό ανταλλαγής datagrams ανάμεσα σε δύο άκρα. Η έμφασή του IP βρίσκεται στην απλότητα και χαρακτηρίζεται από το ότι είναι (1) χωρίς συνδέσεις (connectionless), (2) αναξιόπιστο και (3) δεν διαθέτει μηχανισμό ανάδρασης, δηλαδή πακέτα επιβεβαίωσης (unacknowledged). Κατά μείζονα λόγο δεν διαθέτει μηχανισμό διάγνωσης και διαχείρισης σφαλμάτων. Παρά

το λιτό του σχεδιασμό δουλεύει ως επί το πλείστον χωρίς προβλήματα. Ωστόσο η αποστολή εσφαλμένων πακέτων δεν μπορεί να αποφευχθεί εντελώς, ενώ προβλήματα παρουσιάζονται σε διαδρομές, ζεύξεις, καθώς και στις συσκευές που δημιουργούν και δέχονται τα πακέτα. Οι αρμοδιότητες της αντιμετώπισης των προβλημάτων αυτών έχουν εκχωρηθεί σ᾽ ένα χωριστό πρωτόκολλο, το Internet Control Message Protocol (ICMP), το οποίο σχεδιάσθηκε για να συνοδεύει το IPv4 (RFC 792, RFC 1122, RFC 1812), ενώ για το IPv6 δημιουργήθηκε το αντίστοιχο ICMPv6 (RFC 2463). Τα μηνύματα *ελέγχου* που ανταλλάσσονται στα πλαίσια του ICMP ενθηλακώνονται σε πακέτα IP.

Στα πλαίσια του ICMP ορίζονται τύποι μηνυμάτων για την αναγγελία σφαλμάτων και για τη γενικότερη διαφύλαξη της ομαλής λειτουργίας του IP. Υπάρχουν επομένως δύο βασικές κλάσεις (classes), (α) μηνύματα (αναφοράς) σφάλματος και (β) πληροφοριακά μηνύματα ή μηνύματα ερώτησης (informational, query). Κάθε μήνυμα διαθέτει ένα πεδίο `Type` μήκους 8 bits (κοινό και για τις δύο κλάσεις[5]). Ένα δεύτερο πεδίο ονομαζόμενο `Code`, μήκους επίσης 8 bits, εξειδικεύει ακόμη περισσότερο το σκοπό του μηνύματος. Για παράδειγμα, σε ένα μήνυμα όπου το πεδίο Type υποδηλώνει ότι κάποιο πακέτο δεν μπορεί να φτάσει στον προορισμό του (destination unreachable) το πεδίο `Code` εξειδικεύει το πρόβλημα.

Εκείνα τα πακέτα του ICMP που δημιουργούνται από ένα κόμβο μετά από την ανίχνευση κάποιου προβλήματος (error messages) έχουν ως προορισμό τον αποστολέα που φέρουν τα datagrams τα σχετικά με το πρόβλημα. Με άλλα λόγια το μήνυμα οδηγείται στην αρχή του μονοπατιού. Αυτό σημαίνει ότι αν ο εξυπηρετητής A στέλνει πακέτα στον εξυπηρετητή B μέσω των κόμβων N_1, N_2, όταν ο N_2 διαπιστώσει σφάλμα σε ένα datagram που οφείλεται στον κόμβο N_1, είναι υποχρεωμένος να στείλει ένα πακέτο ελέγχου που θα αναφέρει το πρόβλημα στον A και όχι στον N_1. Ο λόγος είναι ότι στη σύνταξη (format) ενός πακέτου IP προσδιορίζονται μόνο διευθύνσεις πηγής και προορισμού.

Η ανταλλαγή περιττών μηνυμάτων σφάλματος είναι ένα υπαρκτό πρόβλημα στα δίκτυα χωρίς να είναι απαραίτητο να εμπλέκονται κακόβουλοι χρήστες. Η κατάσταση αυτή θα μπορούσε να πάει μακριά χωρίς κάποιες προβλέψεις. Αν π.χ. η συσκευή A στείλει ένα μήνυμα σφάλματος στη συσκευή B και στη συνέχεια η B βρει ένα σφάλμα στο τελευταίο μήνυμα θα στείλει ένα ακόμη μήνυμα προς την A, κ.ο.κ. ώστε να δημιουργηθούν στη συνέχεια «άπειρα» τέτοια μηνύματα. Για να αποφευχθούν τέτοιες καταστάσεις υπάρχει ένα σύνολο κανόνων που εμποδίζουν τη διαδοχική γέννηση μηνυμάτων σφάλματος από προηγούμενα μηνύματα σφάλματος.

Τι είναι το ping

Το όνομα *ping* φέρει μια λειτουργία που υλοποιείται χρησιμοποιώντας, όπως αναφέρθηκε και πιο πάνω, μηνύματα ICMP τύπων `Echo (Request)` και `Echo Reply`. Η συσκευή A στέλνει ένα μήνυμα `Echo` στη συσκευή B, η οποία επιστρέφει στην A ένα μήνυμα `Echo Reply`. Αυτό είναι και το βασικό σενάριο λειτουργίας του ping, αλλά συνήθως χρησιμοποιείται στέλνοντας περισσότερα από ένα μηνύματα `Echo` και `Echo Reply` με σκοπό να γίνει μια καλύτερη διάγνωση από το αν απλώς μπορούν να επικοινωνήσουν οι δύο συσκευές. Οι παραπάνω πληροφορίες που μπορούν να εξαχθούν είναι σχετικές με πιθανή συμφόρηση ή απώλεια πακέτων και με τη μέτρηση των καθυ-

[5] Οι αριθμοί 0-255 στο IPv4 δεν διαχωρίζονται με ένα προφανή τρόπο ανάμεσα στις δύο κλάσεις, έτσι ώστε π.χ. να ανήκουν σε δύο συνεχή διαστήματα. Στο IPv6 αντίθετα οι τιμές 0-127 ανήκουν σε μηνύματα σφάλματος και οι τιμές 128-255 σε πληροφοριακά μηνύματα.

```
C:\abc>sudo ping -i 0.001 -c 5 central.ntua.gr
PING central.ntua.gr (147.102.243.243): 56 data bytes
Request timeout for icmp_seq 0
64 bytes from 147.102.243.243: icmp_seq=0 ttl=63 time=1.375 ms
64 bytes from 147.102.243.243: icmp_seq=1 ttl=63 time=0.861 ms
64 bytes from 147.102.243.243: icmp_seq=2 ttl=63 time=0.880 ms
64 bytes from 147.102.243.243: icmp_seq=3 ttl=63 time=0.733 ms
64 bytes from 147.102.243.243: icmp_seq=4 ttl=63 time=0.991 ms

--- central.ntua.gr ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.733/0.968/1.375/0.219 ms

C:\abc>sudo ping -f -c 100000 central.ntua.gr
PING central.ntua.gr (147.102.243.243): 56 data bytes
..Request timeout for icmp_seq 24718
..Request timeout for icmp_seq 39925
..Request timeout for icmp_seq 65808
.
--- central.ntua.gr ping statistics ---
100000 packets transmitted, 99997 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.316/0.511/4.210/0.075 ms
```

Σχήμα 6.1: Παραδείγματα εκτέλεσης της εντολής `ping`.

στερήσεων που δημιουργούνται στην επικοινωνία. Στο Σχ. 6.1 παρατίθεται ένα παράδειγμα εκτέλεσης της εντολής `ping` από command prompt.

Η χρήση του ping μπορεί να γίνει με πρόσθετες επιλογές (options) που ποικίλλουν ανάλογα με το λειτουργικό και την υλοποίηση. Η χρήση της εντολής `ping -f` σε UNIX δημιουργεί μια σειρά από πακέτα `Echo` με μεγάλη ταχύτητα με σκοπό τη δοκιμή της αντοχής ενός δικτύου. Με την επιλογή `-f` φεύγουν πακέτα με το ρυθμό που φτάνουν οι απαντήσεις, αλλά τουλάχιστον με ρυθμό 100 ανά sec. Η συχνότητα των πακέτων του `ping` μπορεί να ορισθεί απ' ευθείας με την επιλογή `-i` (όχι όμως συγχρόνως με την επιλογή `-f`) με default τιμή το 1 sec και περιορισμούς στην ελάχιστη τιμή για κανονικούς χρήστες (όχι super-user).

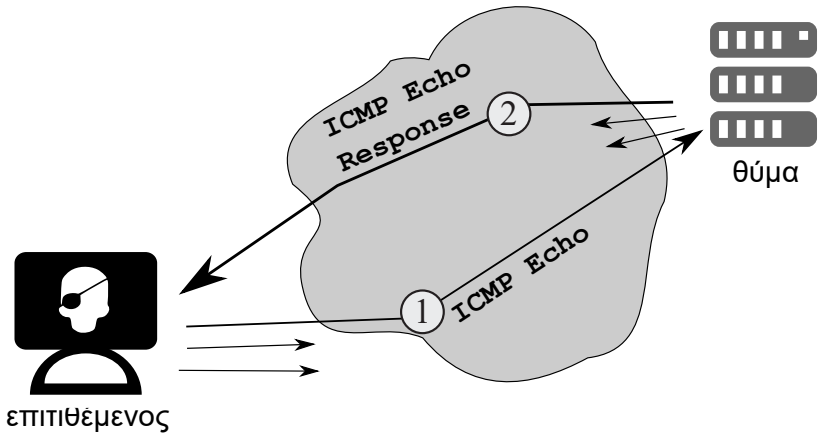
Το ping με κατάλληλες επιλογές (options) μπορεί να γίνει χρήσιμο διαγνωστικό εργαλείο στα χέρια διαχειριστών, αλλά και εισβολέων για την ανίχνευση στόχων (βλ. κεφάλαιο για θέματα ανίχνευσης). Για τους λόγους αυτούς οι διαχειριστές έχουν την ευθύνη να καθορίσουν σε ποια πακέτα ping θα επιτρέψουν τη διέλευση.

Πώς γίνεται η επίθεση με `ping`

Η επίθεση με `ping -f` είναι μια πολύ εύκολη λύση για έναν επιτιθέμενο με χαμηλές αξιώσεις και μπορεί να έχει επιτυχία αν ο εξυπηρετητής-θύμα συνδέεται με το υπόλοιπο δίκτυο μέσα από ένα σχετικά αργό δίκτυο τοπικής πρόσβασης. Στο σημείο στένωσης, που μπορεί να είναι μια μοναδική ζεύξη, αν δημιουργηθεί συμφόρηση κάποια πακέτα θα διαγραφούν, περιλαμβανομένων και πακέτων της «νόμιμης» κίνησης που απευθύνεται στον εξυπηρετητή. Οι κανονικοί χρήστες της υπηρεσίας θα αποκτήσουν την εντύπωση ότι η υπηρεσία έχει προβλήματα τόσο οξύτερα όσο μεγαλύτερες είναι οι απώλειες.

Η επίθεση όμως αυτή έχει μια σειρά από μειονεκτήματα:

- Η διεύθυνση του επιτιθέμενου ενυπάρχει στα πακέτα με τα οποία γίνεται η επίθεση. Το θύμα μπορεί να οργανώσει την άμυνά του, να εντοπίσει τον δράστη και να λάβει αντίμετρα, ενδεχομένως και νομικών περιλαμβανομένων.



Σχήμα 6.2: Επίθεση με `ping`.

- Είναι σαφές στο θύμα ότι όλη η κίνηση προέρχεται από μία και μόνη πηγή, γεγονός που διευκολύνει την ανίχνευση και την άμυνα.

- Ο επιτιθέμενος υφίσταται την «ανακλώμενη» κίνηση που συντίθεται από τις αντίστοιχες αποκρίσεις (πακέτα `Echo Response`). Κατά πόσο αυτό το γεγονός θα βλάψει τον επιτιθέμενο από την άποψη της επικοινωνίας σχετίζεται με το εύρος ζώνης της δικής του πρόσβασης, ενώ αναμένεται επίσης κάποια επίπτωση σε φόρτο επεξεργασίας.

- Η κίνηση που δημιουργείται από μια μόνο μηχανή δεν είναι κατά κανόνα αρκετή για να φέρει το επιθυμητό (από την οπτική γωνία του επιτιθέμενου) αποτέλεσμα, κυρίως επειδή τα σημερινά δίκτυα διαθέτουν ζεύξεις με ικανή χωρητικότητα ακόμη και στο δίκτυο τοπικής πρόσβασης.

Η απάντηση της πλευράς των επιτεθεμένων σ' αυτά τα προβλήματα συνίσταται κυρίως στη χρήση (α) πολλών μηχανών σε μια κατάλληλα ενορχηστρωμένη επίθεση και (β) στη χρήση αλλοιωμένων διευθύνσεων (spoofed IP address). Μια λύση που συνδυάζει τα πλεονεκτήματα των λύσεων (α) και (β) συνίσταται στη χρήση ενός δικτύου ανδρεικέλων (botnet), δεδομένου ότι τότε δεν είναι καν απαραίτητη η χρήση αλλοιωμένων διευθύνσεων.

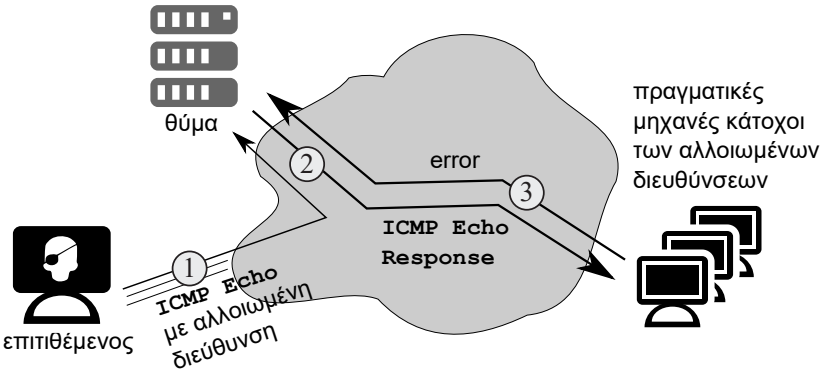
Ping με αλλοιωμένη διεύθυνση

Είδαμε στην προηγούμενη ενότητα ότι εφόσον χρησιμοποιηθεί επίθεση με πλημμύρα ping ο επιτιθέμενος υφίσταται κατά περίπτωση ορισμένες συνέπειες, όπως αυξημένη πιθανότητα αναγνώρισης της επίθεσης και της ταυτότητάς του αφού έχει δημιουργήσει ένα ισχυρό ρεύμα πακέτων και επιστροφή κίνησης με μηνύματα σφάλματος.

Ένας τρόπος να αποφύγει τα παραπάνω είναι μέσω της αλλοίωσης της διεύθυνσης προέλευσης στα μηνύματα που δημιουργεί. Επί πλέον ανοίγει μια νέα δυνατότητα, που συνίσταται στην δημιουργία επιθέσεων εξ ανακλάσεως, δηλαδή βασίζεται στα πακέτα Echo Reply αντί των πακέτων Echo. Αυτό μπορεί να γίνει προφανώς στέλνοντας πακέτα Echo σε ένα πληθυσμό μηχανών, βάζοντας ως διεύθυνση αποστολής τη διεύ-

θυνση του θύματος. Οι μηχανές αυτές θα απαντήσουν με αντίστοιχα Echo Reply προς το θύμα.

Για κάθε πακέτο ICMP echo που φτάνει στον προορισμό-θύμα, γεννιέται ένα πακέτο ICMP echo response, που δρομολογείται προς το υποτιθέμενο σύστημα προέλευσης, που αντιστοιχεί στη διεύθυνση που έχει βάλει ο επιτιθέμενος. Μια δυνατή περίπτωση είναι η αλλοιωμένη διεύθυνση να μην αντιστοιχεί σε υπαρκτό σύστημα (non routable σύμφωνα με την RFC1597), οπότε το θύμα αρχικά προσπαθεί να λύσει το πρόβλημα της αναγνώρισης της προέλευσης του πακέτου. Για παράδειγμα, αν είναι πάνω σε ένα τοπικό δίκτυο μπορεί να εκπέμψει ένα `ARP Request`, δηλαδή ένα μήνυμα στα πλαίσια του Address Resolution Protocol (ARP), περιμένοντας να πάρει απάντηση από τη μηχανή που έχει αυτή τη διεύθυνση. Η απάντηση δεν θα έρθει και το θύμα μπορεί να ξαναπροσπαθήσει, αλλά τελικά θα εγκαταλείψει και δε θα φύγει κανένα πακέτο προς την αλλοιωμένη διεύθυνση, επειδή δεν μπορεί να βρεθεί σε ποια μηχανή ανήκει.



Σχήμα 6.3: Επίθεση με `ping` με αλλοιωμένη διεύθυνση.

Αν όμως η αλλοιωμένη διεύθυνση ανήκει σε υπαρκτό σύστημα, θα φτάσει εκεί ένα πακέτο ICMP Echo Response, το οποίο όμως δεν μπορεί να αντιστοιχηθεί από το εν λόγω υπαρκτό σύστημα σε κάποιο πακέτο ICMP Echo (request) που θα έπρεπε να είχε στείλει πιο πριν. Η κατάσταση αυτή θα προκαλέσει τη γέννηση ενός νέου πακέτου σφάλματος, που θα σταλεί προς το θύμα και θα επιβαρύνει περαιτέρω τη θέση του τελευταίου (βλ. Σχ. 6.3).

Ποιο από τα παραπάνω δύο ενδεχόμενα θα συμβεί εξαρτάται από τις επιλογές του επιτιθέμενου ως προς τη γέννηση αλλοιωμένων διευθύνσεων. Ο επιτιθέμενος θα μπορούσε να τοποθετεί στα πακέτα διευθύνσεις μέσα από κατάλογο, που έχει σχηματισθεί εκ των προτέρων με συγκεκριμένα κριτήρια. Στην περίπτωση που δεν επιθυμεί ανακλώμενη κίνηση μπορεί να επιλέξει μόνο ανύπαρκτες διευθύνσεις. Στην περίπτωση που επιθυμεί να φαίνεται προέλευση της επίθεσης από συγκεκριμένο χώρο μπορεί να επιλέξει διευθύνσεις του χώρου αυτού. Στην περίπτωση που θέλει να ελαχιστοποιήσει την προσπάθεια επιλογής διευθύνσεων μπορεί να αρκεσθεί σε μια γεννήτρια τυχαίων διευθύνσεων, γεγονός που θα έχει πιθανώς μεικτά αποτελέσματα (διευθύνσεις υπαρκτές και διευθύνσεις ανύπαρκτες).

Ωστόσο η χρήση ανύπαρκτων διευθύνσεων μπορεί να χρησιμοποιηθεί από την πλευρά της άμυνας ως ένα ακόμη στοιχείο για την ανίχνευση των επιθέσεων. Αν κάποιος μπορεί να μαζέψει τα «ορφανά» πακέτα που αποστέλλονται σε ανύπαρκτες διευθύνσεις θα μπορέσει να βγάλει συμπεράσματα για την ύπαρξη αντίστοιχων επιθέσεων,

δεδομένου ότι τα «νόμιμα» πακέτα φέρουν υπαρκτές διευθύνσεις. Είναι αλήθεια ότι σφάλματα του λογισμικού γεννούν επίσης λανθασμένη κίνηση και το φαινόμενο αυτό πρέπει επίσης να ληφθεί υπόψη στις εκτιμήσεις για πιθανές επιθέσεις.

Ανάλυση οπισθοσκέδασης

Η ανίχνευση επιθέσεων έχει ως πρώτο σκοπό την άμυνα, είτε σε πραγματικό χρόνο είτε για να βελτιωθεί η άμυνα στο μέλλον. Ωστόσο ακόμη και η καταγραφή περασμένων επιθέσεων δεν είναι εύκολη υπόθεση. Οι πάροχοι υπηρεσιών που έχουν υποστεί επιθέσεις δεν είναι πρόθυμοι να τις αποκαλύψουν, αν μη τι άλλο για να μη βλάψουν τη φήμη της εταιρίας. Όμως τουλάχιστον οι επιθέσεις κίνησης αφήνουν ίχνη στο δίκτυο, οπότε η ανίχνευση σε κάποιο βαθμό μπορεί να γίνει και χωρίς τη συνεργασία των θυμάτων. Ένα παράδειγμα θα δούμε στη συνέχεια.

Η ανίχνευση επιθέσεων μπορεί να γίνει με πολλές μεθόδους, εδώ όμως θα μιλήσουμε περιληπτικά για μια μέθοδο που βασίζεται ακριβώς στην ανίχνευση, συλλογή και εξέταση πακέτων που ανήκουν στην χαμένη κίνηση προς ανύπαρκτες διευθύνσεις. Η κίνηση αυτή είναι γνωστή με τον όρο «κίνηση οπισθοσκέδασης» (back-scattered traffic, όρος που έχει την προέλευσή του στη φυσική σωματιδίων). Δεδομένης της συχνότητας και του όγκου των επιθέσεων κάθε μηχανή δέχεται καθημερινά τέτοια κίνηση, η οποία τις περισσότερες φορές σβήνει πάνω στα firewalls.

Οι Moore, Shannon κ.α. σκέφτηκαν να περισυλλέξουν τα «οπισθοσκεδαζόμενα» πακέτα και μέσω αυτών να εκτιμήσουν την ύπαρξη και τον όγκο πιθανών επιθέσεων [Moo+06]. Η εν λόγω μέθοδος λέγεται *ανάλυση οπισθοσκέδασης* (backscatter analysis).

Ένα βασικό ερώτημα για τη χρήση μιας μεθόδου βασισμένης στην ανίχνευση της οπισθοσκέδασης είναι αν η τελευταία παράγει αποτελέσματα που μπορούν να παρατηρηθούν με ικανοποιητικό τρόπο, αν δηλαδή ο συλλέκτης τέτοιων πακέτων θα έχει τη δυνατότητα να εντοπίσει αρκετά δείγματα ώστε να μπορέσει να διαγνώσει μια επίθεση. Αν θεωρήσουμε ένα δίκτυο με διευθύνσεις IPv4, το πλήθος όλων των δυνατών διευθύνσεων είναι 2^{32}. Όταν γεννηθεί ένα πακέτο με ομοιόμορφα τυχαία αλλοιωμένη διεύθυνση, η πιθανότητα να συμπίπτει η διεύθυνση αυτή με τη διεύθυνση μιας δεδομένης μηχανής, π.χ. της μηχανής που χρησιμοποιείται για την παρατήρηση, είναι ίση με $1/2^{32} = 2^{-32}$. Αν η επίθεση περιλαμβάνει m πακέτα (με τυχαίες πάντοτε διευθύνσεις), η πιθανότητα να ληφθεί τουλάχιστον ένα εξ αυτών είναι $1 - (1 - 2^{-32})^m$. Τέλος, αν η μηχανή παρατήρησης καταφέρει να συλλέγει πακέτα που αντιστοιχούν σε n διαφορετικές διευθύνσεις, η πιθανότητα συλλογής ενός τουλάχιστον πακέτου γίνεται [Moo+06]

$$1 - \left(1 - \frac{n}{2^{32}}\right)^m$$

Ο προσδοκώμενος αριθμός παρατηρούμενων πακέτων είναι $E\{X\} = mn/2^{32}$, και επομένως (διαιρώντας με ένα παράθυρο χρόνου) αν παρατηρούμε πακέτα που ανήκουν σε μια επίθεση και έχουν ρυθμό (σε πακέτα ανά sec) r, για τον πραγματικό ρυθμό r_0 θα ισχύει

$$r = r_0 \frac{n}{2^{32}} \Rightarrow r_0 = r \frac{2^{32}}{n}$$

Ωστόσο το μέγεθος αυτό μάλλον πρέπει να θεωρηθεί ως ισχυρά υποεκτιμημένο, διότι κάποια από τα οπισθοσκεδαζόμενα πακέτα είναι πιθανό να «κοπούν» από μηχανισμούς του δικτύου (π.χ. firewalls) που ελέγχουν την ορθότητα των πακέτων. Η μέθοδος αυτή πρέπει επίσης να λάβει υπόψη ότι «χαμένα» πακέτα μπορούν να γεννηθούν με διάφορους «νόμιμους» τρόπους (από σφάλματα, μηχανισμούς ανίχνευσης κ.α.), οπότε υπάρχει ένας γενικότερος «θόρυβος» μέσα σ᾽ ένα δίκτυο, γνωστός ως

Internet background noise. Παρά τα προβλήματα οι παραπάνω ερευνητές κατάφεραν να παρουσιάσουν αποτελέσματα για χιλιάδες επιθέσεων διαφόρων τύπων.

Έλεγχος εξόδου (egress filtering)

Όπως είδαμε η αλλοίωση της διεύθυνσης προέλευσης ενός πακέτου δίνει μια σειρά από πλεονεκτήματα στην πλευρά της επίθεσης. Ένα εύλογο προληπτικό μέτρο είναι το φιλτράρισμα των πακέτων την ώρα που αυτά εξέρχονται από την πηγή τους (όσο γίνεται πιο κοντά στην πηγή) ώστε να αφαιρούνται πακέτα με διεύθυνση προέλευσης διαφορετική από την ίδια τη διεύθυνση της πηγής. Ο έλεγχος αυτός είναι σχετικά εύκολος και μπορεί να υλοποιηθεί από κάθε εμπορικά διαθέσιμο δρομολογητή [Pie04; Dis08]. Το κλειδί όμως για την επιτυχία αυτής της τεχνικής είναι η υιοθεσία της στην ευρύτερη δυνατή κλίμακα. Ανάλογα φίλτρα μπορούν να δημιουργηθούν για την προστασία από spam email και από ιούς, σκουλήκια κ.λπ. Η γενίκευση της προστασίας αυτού του είδους περνάει από τη συμμόρφωση των ISPs (βλ. και RFC3013 [RFC 3013], όπου υπάρχει η σύσταση να γίνεται έλεγχος από τους δρομολογητές στα όρια του κάθε ISP[6]).

Γενικότερα ο έλεγχος αυτού του τύπου ανήκει σε μια κατηγορία ελέγχων, με τους οποίους γίνεται προσπάθεια να διαπιστωθεί αν ένα datagram φέρει διεύθυνση πηγής συμβατή με την τοπολογία του δικτύου και άλλες διαθέσιμες πληροφορίες. Για παράδειγμα, το πρώτο μέρος της διεύθυνσης IP πηγής είναι συμβατό με το υποδίκτυο απ' όπου έρχεται το datagram; Όταν η πηγή ανήκει σ' ένα Ethernet, μπορεί να γίνει διασταύρωση της διεύθυνσης πηγής μέσω του ARP (Address Resolution Protocol); Τέτοιοι έλεγχοι μπορούν να γίνουν τόσο καλύτερα όσο πιο κοντά βρίσκεται το σημείο ελέγχου στην πηγή ενός πακέτου, ενώ η δυνατότητα αυτή μειώνεται όσο απομακρυνόμαστε από αυτήν[7].

Πλημμύρα TCP SYN

Η πλημμύρα αυτή είναι γνωστή[8] από το 1996. Η επίθεση βασίζεται στο πρωτόκολλο χειραψίας *SYN-SYN-ACK*[9] με το οποίο εγκαθίσταται μια σύνδεση TCP μεταξύ ενός πελάτη (client) και ενός εξυπηρετητή (server) (βλ. Σχ. 6.4) μαζί με τις σχετικές παραμέτρους υποδοχών TCP (TCP sockets). Η πλημμύρα TCP SYN ανάλογα με τις συνθήκες έχει στοιχεία ενίσχυσης και ανάκλασης.

Η κανονική διαδικασία χειραψίας Ο πελάτης ξεκινάει τη χειραψία αποστέλλοντας ένα μήνυμα `SYN` στον εξυπηρετητή, ο οποίος με τη σειρά του απαντάει με ένα `SYN-ACK` (επιβεβαίωση συγχρονισμού). Τέλος ο πελάτης στέλνει μια ακόμη επιβεβαίωση, με την οποία κλείνει η φάση της χειραψίας, οπότε στη συνέχεια οι δύο πλευρές μπορούν να

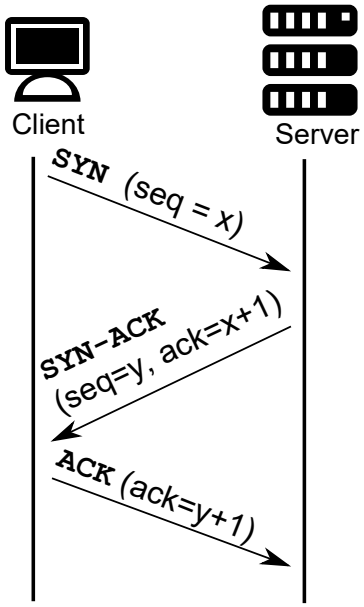
[6] RFC 3013, στην παράγρ. 4.4: "To reduce the exposure of their customers to attacks that rely on forged source addresses ISPs should do the following. At the boundary router with each of their customers they should proactively filter all traffic going to the customer that has a source address of any of the addresses that have been assigned to that customer".

[7] Στην ενότητα 4.1 της RFC6959 [RFC 6959] μπορείτε να δείτε ποιες δυνατότητες ελέγχου έχουμε σε διαφορετικά είδη δικτύων και όταν υπάρχει διαφορετική απόσταση από την πηγή, καθώς επίσης και μια σειρά εργαλείων που υλοποιούν τους ελέγχους.

[8] *Two "underground magazines" have recently published code to conduct denial-of-service attacks by creating TCP "half-open" connections.* CERT CA-1996-21, `https://www.cert.org/historical/advisories/CA-1996-21.cfm?`

[9] Είναι γνωστό με αυτό το όνομα αν και κανονικά θα έπρεπε να λέγεται SYN – SYN-ACK – ACK.

ανταλλάξουν δεδομένα. Η κάθε πλευρά ξεκινάει από τον αύξοντα αριθμό μηνύματος που έχει αναγγείλει.[10]



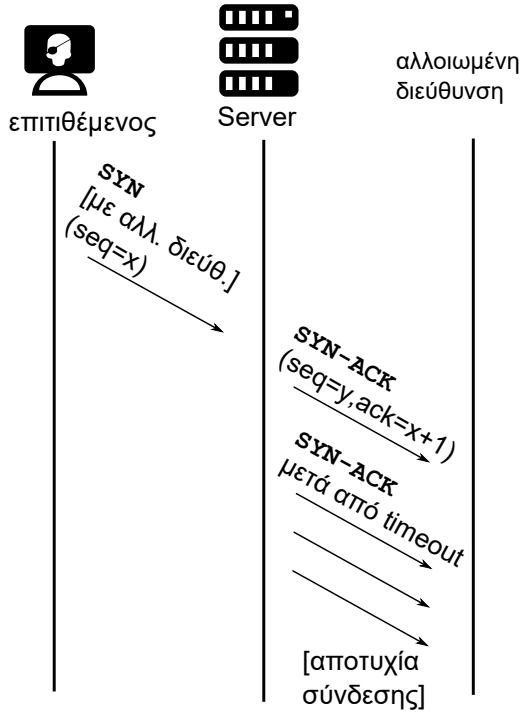
Σχήμα 6.4: Διαδικασία χειραψίας TCP.

Απλή επίθεση SYN Στην απλή εκδοχή μιας επίθεσης με εκμετάλλευση του πρωτοκόλλου χειραψίας ο επιτιθέμενος στέλνει μια σειρά από μηνύματα `SYN` στον εξυπηρετητή - θύμα. Ο δεύτερος σε καθένα από αυτά απαντάει με ένα μήνυμα `SYN-ACK`, σύμφωνα με το πρωτόκολλο χειραψίας, και σημειώνει το αίτημα για νέα σύνδεση σε πίνακα που αναγράφονται οι τρέχουσες συνδέσεις. Οι διαστάσεις αυτού του πίνακα είναι συνήθως τέτοιες ώστε να εξυπηρετούνται οι συνδέσεις κάτω από τις συνηθισμένες συνθήκες. Ωστόσο ο επιτιθέμενος δεν στέλνει το τρίτο μήνυμα της χειραψίας, το `ACK`, γεγονός που αναγκάζει τον εξυπηρετητή να περιμένει για κάποιο διάστημα (που έχει προβλεφθεί για την περίπτωση συμφόρησης), στο τέλος του οποίου θα στείλει πίσω ένα `RST` (reset) και θα τερματίσει τη σύνδεση. Μέχρι τότε όμως συσσωρεύονται «μισάνοιχτες» συνδέσεις, οι οποίες καταναλώνουν θέσεις στον πίνακα συνδέσεων και πιθανώς άλλους πόρους στον εξυπηρετητή.

Επίθεση TCP SYN με αλλοιωμένες διευθύνσεις Ο επιτιθέμενος στέλνει πακέτα `SYN` (ορθότερα *τμήματα* TCP, segments) στο θύμα όπως και πριν, αλλά χρησιμοποιεί πλαστές διευθύνσεις. Οι τελευταίες μπορεί να προέρχονται από ένα προκατασκευασμένο κατάλογο, αλλά μπορεί και να γεννώνται με τυχαίο τρόπο. Ειδικά στη δεύτερη περίπτωση κάποιες από αυτές θα είναι πιθανώς ανύπαρκτες. Ο εξυπηρετητής αποκρίνεται με `SYN-ACK`, το οποίο κατευθύνεται στην εκάστοτε υποτιθέμενη διεύθυνση προέλευσης του `SYN`. Αν η διεύθυνση αυτή είναι υπαρκτή, η μηχανή που την κατέχει θα απαντήσει (αργά ή γρήγορα, ανάλογα με το φόρτο της) με `RST` (reset), το οποίο θα

[10] Για μια πιο αναλυτική περιγραφή βλ. π.χ. [TW13] παρ. 6.5.5 και [Hal05] παρ. 7.3.2.

φτάσει στον εξυπηρετητή κι αυτός θα διαγράψει το αίτημα. Η πιο ευνοϊκή όμως περίπτωση για τον επιτιθέμενο προκύπτει όταν η διεύθυνση είναι ανύπαρκτη (πράγμα που μπορεί να το προσχεδιάσει αν διαθέσει περισσότερη προετοιμασία) ή αν ανήκει σε μηχανή που είναι απασχολημένη και έχει δυσκολίες να απαντήσει (στις δυσκολίες αυτές μπορεί να «βοηθήσει» ο επιτιθέμενος στέλνοντάς της φόρτο εργασίας). Σε κάθε περίπτωση εφόσον ο εξυπηρετητής δεν πάρει απάντηση θα επαναλάβει μερικές ακόμη φορές το μήνυμα `SYN-ACK` και τελικά θα διαγράψει το αίτημα (Σχ. 6.5). Ωστόσο η εγγραφή της ημιτελούς σύνδεσης θα παραμείνει για περισσότερο χρόνο στον σχετικό πίνακα του εξυπηρετητή, γεγονός που μπορεί να καταλήξει σε εξάντληση του εν λόγω πόρου.



Σχήμα 6.5: Επίθεση TCP SYN με αλλοιωμένες διευθύνσεις.

Απαιτούμενος όγκος κίνησης Η εν λόγω επίθεση δεν βασίζεται, όπως άλλες επιθέσεις, στην πλήρη εξάντληση διαθέσιμων πόρων, όπως το εύρος ζώνης ή η μνήμη, αλλά προσπαθεί απλώς να φέρει τη λειτουργία ενός συστήματος στα όρια (quota) που περιγράφονται από διάφορες παραμέτρους του. Συνήθως πρόκειται για το όριο στο πλήθος των ημιτελών (εκκρεμών) συνδέσεων ανά port. Το λειτουργικό σύστημα του εξυπηρετητή μπορεί να εφαρμόζει διαφορετικούς τέτοιους περιορισμούς κατά περίπτωση και ο διαχειριστής μπορεί να μεταβάλλει τα αντίστοιχα όρια. Μπορεί δηλαδή ο περιορισμός να αφορά σε νέες συνδέσεις ανά θύρα (port) ή συνολικά για όλες τις θύρες μαζί. Μπορεί να εφαρμόζει περιορισμούς στο πλήθος όλων των συνδέσεων, ή να διακρίνει τις ημιτελείς από αυτές που είναι επιβεβαιωμένες, αλλά δεν έχουν ακόμη χρησιμοποιηθεί για μεταφορά δεδομένων και αυτές που μεταφέρουν δεδομένα, αυτές στις οποίες η μεταφορά έχει ολοκληρωθεί αλλά κρατούνται μισάνοιχτες για προσεχή

χρήση κ.ο.κ. (βλ. και RFC 4987 [RFC 4987], παρ. 2.2). Η γενική ιδέα πάντως είναι να στείλει ο επιτιθέμενος μια ακολουθία από ριπές τμημάτων SYN (με ή χωρίς αλλοιωμένη διεύθυνση) έτσι ώστε να μην προλάβουν να έρθουν απαντήσεις στα SYN-ACK που θα στείλει γι' αυτές ο εξυπηρετητής. Το μέγεθος της ριπής πρέπει να είναι τέτοιο που να υποχρεώνει το θύμα να φτάσει στα όρια, αλλά ιδανικά όχι πολύ μεγαλύτερο από αυτό, για να περιοριστεί η σπατάλη πόρων του επιτιθέμενου, καθώς και ο κίνδυνος αποκάλυψης της επίθεσης από υψηλή κίνηση. Η συχνότητα με την οποία πρέπει να παράγονται ριπές κυρίως έχει να κάνει με το πόσο γρήγορα το θύμα απαλλάσσεται από τις μη επιβεβαιωμένες ημιτελείς συνδέσεις. Το TCP δεν προβλέπει κάποιο συγκεκριμένο όριο στο χρόνο ολοκλήρωσής τους, αλλά ένα τέτοιο όριο τίθεται συνήθως από το λειτουργικό σύστημα. Η επόμενη ριπή πρέπει να φτάσει έγκαιρα μόλις αρχίσει να ελευθερώνεται ο χώρος από την προηγούμενη ριπή.

Ο J. Lemon [Lem02] έκανε ένα ενδεικτικό υπολογισμό που οδηγεί στο συμπέρασμα ότι ο επιτιθέμενος (κάτω από ορισμένες συνθήκες και παραμετροποιήσεις) δεν είναι απαραίτητο να είναι ιδιαίτερα γρήγορος στην αποστολή τμημάτων `SYN`. Πιο συγκεκριμένα, η προθεσμία διατήρησης μιας σύνδεσης TCP είναι αρκετά μεγάλη ώστε να χωράει ένα αριθμό από επαναμεταδόσεις και ίση με 511 sec (βλ. [FS11], παρ. 13.2.5 για προθεσμίες στο TCP και παρ. 13.7.4 για την ουρά εισερχομένων αιτημάτων σύνδεσης). Στο διάστημα αυτό αν ο σκοπός του ήταν απλώς να γεμίσει ένα πίνακα που χωράει το πολύ 1024 ημιτελείς συνδέσεις ανά υποδοχή (socket) θα αρκούσε να στέλνει 2 αιτήματα σύνδεσης ανά sec. Ωστόσο όταν έρχεται ένα νέο `SYN` για την ίδια σύνδεση, η ημιτελής σύνδεση που αντιστοιχεί στο προηγούμενο ίδιο αίτημα διαγράφεται. Επίσης θεωρείται ληγμένο και διαγράφεται ένα αίτημα όταν περάσει χρόνος περισσότερος από αυτόν που καθορίζεται με την παράμετρο `RTT` (round trip time). Κατά συνέπεια ο επιτιθέμενος πρέπει να είναι σε θέση να κρατάει «ζωντανό» το αίτημα ανανεώνοντάς το πριν λήξει. Αυτό πρέπει να συμβαίνει με όλα τα αιτήματα που θα γεμίσουν τον πίνακα. Αν θεωρήσουμε προθεσμία `RTT` ίση με 100 msec και πίνακα διάστασης 1024, χρειάζονται 10240 ανά sec εκ μέρους του επιτιθέμενου. Με πακέτα μήκους 64 bytes ο τελικός ρυθμός μετάδοσης είναι μόλις 5.24 Mbps.

Μέθοδοι άμυνας Η επίθεση με πλημμύρα SYN είναι αρκετά παλιά και έχει αναλυθεί επαρκώς, κατά συνέπεια υπάρχουν διάφορες ιδέες και προϊόντα που μπορούν να βοηθήσουν την πλευρά της άμυνας ως εξής [RFC 4987]:

Αύξηση των ορίων Δεδομένης της γενικότερης αύξησης των δυνατοτήτων αποθήκευσης και επεξεργασίας πληροφορίας από τους υπολογιστές, ένας προφανής τρόπος να γίνουν πιο δύσκολες οι επιθέσεις είναι να αυξηθούν τα όρια για νέες και ημιτελείς συνδέσεις. Επίσης ο τρόπος αποθήκευσης των σχετικών πληροφοριών θα μπορούσε να βελτιωθεί. Παρά ταύτα υπάρχουν άλλες πιο ελκυστικές λύσεις.

Φίλτρα Η τεχνική ελέγχου της διεύθυνσης πακέτων κοντά στην πηγή τους αναφέρεται στην σελ. 123. Η μέθοδος αυτή θα μπορούσε δραστικά να μειώσει τις επιθέσεις με αλλοιωμένες διευθύνσεις, πάσχει όμως στην έκταση αποδοχής και εφαρμογής της από εταιρίες και παρόχους.

Μείωση της προθεσμίας ημιτελών συνδέσεων Μια άλλη εύκολη και εύλογη λύση για τους διαχειριστές είναι να μειώσουν το χρόνο διατήρησης (`SYN-RECEIVED Timer`) μια εγγραφής (`Transmission Control Block` (TCB)) που αντιστοιχεί σε ημιτελή σύνδεση. Ωστόσο η λύση αυτή δεν είναι ιδιαίτερη αποτελεσματική, εφόσον

ο επιτιθέμενος αρκεί να αυξήσει το ρυθμό αποστολής πακέτων, ενώ μπορεί να δημιουργήσει προβλήματα στα κανονικά αιτήματα σύνδεσης.

Διαγραφή της παλιότερης ημιτελούς σύνδεσης Κατά την άφιξη μηνύματος `SYN` που βρίσκει τον πίνακα ημιτελών συνδέσεων γεμάτο, διαγράφεται η εγγραφή που αντιστοιχεί στο παλαιότερο μη επιβεβαιωμένο αίτημα. Η λύση αυτή θα επιτύχει αν η ολοκλήρωση μιας «νόμιμης» σύνδεσης χρειάζεται εν γένει λιγότερο χρόνο από το χρόνο στον οποίο το αίτημα θα καταντήσει αυτό το ίδιο να γίνει το παλαιότερο. Προφανώς ο επιτιθέμενος αρκεί να αυξήσει το ρυθμό της επίθεσης, ώστε ο «χρόνος παλαίωσης» να μειωθεί όσο χρειάζεται.

SYN cache Ο J. Lemon [Lem02] έχει περιγράψει ένα μηχανισμό σύμφωνα με τον οποίο η πληροφορία για τα εισερχόμενα αιτήματα σύνδεσης αποθηκεύεται με ελαχιστοποιημένο όγκο, ενώ περισσότεροι πόροι διατίθενται μόνον όταν ολοκληρωθεί η σύνδεση. Ο μηχανισμός αυτός λέγεται *SYN cache*. Ο σχετικός πίνακας υλοποιείται και πάλι με άνω όριο εγγραφών, που ισχύει για όλες τις υποδοχές (sockets) μαζί και όχι ανά υποδοχή. Η θέση μιας εγγραφής στον πίνακα προσδιορίζεται μέσω μιας συνάρτησης κατακερματισμού (hash function). Η συνάρτηση αυτή βοηθάει στην πιο εύκολη αναζήτηση εγγραφών και μέσω της χρήσης ενός τυχαίου μυστικού στην κωδικοποίηση αποθαρρύνει επιθέσεις στις ίδιες εισόδους (βλ. ενότητα 5 στην [Lem02]).

SYN cookies Στην περίπτωση αυτή ο εξυπηρετητής που δέχεται ένα αίτημα σύνδεσης δεν διαθέτει καθόλου δικό του χώρο για την αποθήκευση της κατάστασης και κατά κάποιο τρόπο χρησιμοποιεί το ίδιο το δίκτυο ως αποθηκευτικό χώρο [RFC 4987; Lem02; FS11]. Κωδικοποιεί το βασικό μέρος της πληροφορίας κατάστασης (αφήνοντας έξω ορισμένες προαιρετικές ρυθμίσεις) και το στέλνει πίσω στη θέση του αριθμού ακολουθίας (`Sequence Number`) του `SYN-ACK` (ή και του πεδίου `Timestamp Echo` σε νεότερες εκδοχές του μηχανισμού). Αν η διεύθυνση δεν είναι αλλοιωμένη και πρόκειται για πραγματικό αίτημα σύνδεσης, θα φτάσει το τελικό `ACK` με τον ίδιο αριθμό, οπότε η βασική πληροφορία γίνεται και πάλι διαθέσιμη και μπορεί να χρησιμοποιηθεί για να ολοκληρωθεί η σύνδεση. Η κωδικοποίηση εξαρτάται από την υλοποίηση (δεν είναι τυποποιημένη), οπότε κάποιες από τις υλοποιήσεις δεν είναι συμβατές με ορισμένες επιλογές (options) του TCP.

Επίθεση μέσω PMTUD

Πρόκειται για μια επίθεση που βασίζεται στο μηχανισμό εκτίμησης του μέγιστου πακέτου που μπορεί να περάσει από ένα μονοπάτι. Η επίθεση και μηχανισμοί άμυνας περιγράφονται στις συστάσεις RFC 1191, RFC 1981 και RFC 5927. Η βασική ιδέα είναι να παρασυρθεί το δίκτυο σε υπερβολικό κατατεμαχισμό (fragmentation) των πακέτων, ο οποίος δημιουργεί υποβάθμιση της επίδοσής του. Η επίθεση αυτή μπορεί να θεωρηθεί ως επίθεση ενίσχυσης, δεδομένου ότι το ίδιο ρεύμα πληροφορίας τελικά χρειάζεται ένα αρκετά μεγαλύτερο όγκο συνολικών bits (εξ αιτίας της αύξησης των επικεφαλίδων) για να μεταδοθεί (καθώς και συνολικού χρόνου εξ αιτίας της αποστολής περισσότερων πακέτων). Σε κάποιες ακραίες περιπτώσεις, όπου όπως θα δούμε προκύπτουν απρόβλεπτες ή ακραίες συνέπειες, μπορεί να θεωρηθεί και ως επίθεση λογικής.

Τι είναι το MTU

Γενικά όταν πρόκειται να μεταδοθεί μεγάλος όγκος δεδομένων θα προκύψει μια ακολουθία από πακέτα ή άλλες μονάδες μετάδοσης.[11] Αυτές οι μονάδες (που στην περίπτωση του IP θα είναι datagrams) συμφέρει αυτά να είναι όσο γίνεται μεγαλύτερου μήκους. Το μεγαλύτερο μήκος τείνει να βελτιώνει τις επιδόσεις, επειδή τότε η επικεφαλίδα καταλαμβάνει μικρότερο ποσοστό του συνόλου. Ωστόσο αυτός δεν είναι γενικός κανόνας. Αν υπάρχει θόρυβος η πιθανότητα να πρέπει να επαναμεταδοθεί ένα πακέτο επειδή έχει μη διορθώσιμα σφάλματα αυξάνεται με το μήκος του πακέτου (τμήματος ή όποιας άλλης μονάδας μετάδοσης), οπότε τα πολύ μεγάλα πακέτα δεν βελτιώνουν το throughput. Για μια συγκεκριμένη εφαρμογή το ιδανικό μέγεθος μονάδας εξαρτάται από τις απαιτήσεις της εφαρμογής που δημιουργεί τη σχετική κίνηση, π.χ. κατά πόσο μεταφέρει δεδομένα που προέρχονται από δειγματοληψία ή έχει συγκεκριμένο μέγιστο χρόνο απόκρισης.

Ωστόσο οι πληροφορίες καλούνται να περάσουν μέσα από συγκεκριμένα δίκτυα, που συχνά είναι διαφορετικών τύπων και λειτουργούν με μια ποικιλία πρωτοκόλλων. Τα πρωτόκολλα που ασχολούνται με την οργανωμένη μεταφορά δεδομένων ανάμεσα σε δυο άκρα, αλλά και από τους ενδιάμεσους κόμβους, λειτουργούν με διαφορετικές μονάδες, που έχουν επίσης διαφορετικά μήκη. Συνήθως το μήκος της μονάδας που χρησιμοποιεί ένα πρωτόκολλο έχει μια ευελιξία, αλλά μέσα σε συγκεκριμένα άνω και κάτω όρια. Η *μέγιστη μονάδα μετάδοσης* (maximum transmission unit, MTU) ενός τηλεπικοινωνιακού πρωτοκόλλου είναι το μεγαλύτερου μήκους «πακέτο» που μεταδίδεται βάσει του πρωτοκόλλου και των περιορισμών που επιβάλλονται. Το μέγεθος της μέγιστης μονάδας μετάδοσης όπως επιβάλλεται από τα πρωτόκολλα ποικίλλει από μερικές δεκάδες bytes σε κάμποσες χιλιάδες.[12]

Αυτό σημαίνει ότι στα άκρα ενός μονοπατιού που συνδέει δυο πλευρές που επικοινωνούν, μπορεί να έχουν ορισθεί από αυτές τις πλευρές μονάδες κατάλληλου μήκους, αλλά αυτές να μη μπορούν να περάσουν από κόμβους που είναι πιο μέσα στο μονοπάτι. Η λύση που απομένει είναι ο τεμαχισμός (ή «θραύση» ή «κατακερματισμός», fragmentation) των αρχικών μονάδων σε μικρότερες, που λέγονται *θραύσματα* (fragments). Ο τεμαχισμός (και η αντίστοιχη επανασύσταση των αρχικών μονάδων) μπορεί να γίνει σε διάφορα σημεία, αρκεί τα θραύσματα να είναι αρκετά μικρά για οποιοδήποτε σημείο διέλευσης. Οι δύο ακραίες λύσεις είναι (α) να γίνουν εξ αρχής ανάμεσα στα δύο άκρα πακέτα αρκετά μικρά ώστε να περνούν από παντού και (β) να γίνει χωριστός τεμαχισμός-ανασύσταση σε κάθε κομμάτι του δικτύου που απαιτεί αλλαγή μεγέθους της μονάδας (βλ. [TW13], παρ. 5.5.5 και [Hal05], παρ. 6.3).

Στο IP αποφεύγεται ο τεμαχισμός στους ενδιάμεσους κόμβους, οπότε είναι απαραίτητο να υπάρξει τρόπος καθορισμού του κατάλληλου μήκους μονάδας, δηλαδή του ελάχιστου από τα μέγιστα μήκη που ανέχεται το κάθε κομμάτι. Αυτό το κάνει ο αλγόριθμος ανακάλυψης του MTU που περιγράφεται στην επόμενη παράγραφο.

[11] Η επίσημη ονομασία της μεταδιδόμενης μονάδας, μιας δηλαδή οργανωμένης ακολουθίας από bits, ποικίλλει ανάλογα με το πρωτόκολλο. Οι συγγραφείς βιβλίων δικτύων υπολογιστών και επικοινωνιών άλλοτε είναι ακριβείς και χρησιμοποιούν τον κατάλληλο όρο ανάλογα με το επίπεδο, όπως αυτός ο όρος ορίζεται στα αντίστοιχα πρότυπα, κι άλλοτε χρησιμοποιούν αδιακρίτως λέξεις όπως «πακέτο» ή «μονάδα» (unit).

[12] Ενας συγκεντρωτικός πίνακας που δίνει τη μέγιστη μονάδα σύμφωνα με διάφορα πρωτόκολλα υπάρχει στη Wikipedia στην εξής ιστοσελίδα: `https://en.wikipedia.org/wiki/Maximum_transmission_unit`.

Τι είναι το Path MTU Discovery (PMTUD)

Πρόκειται για μια αλγοριθμική τεχνική, βάσει της οποίας διαπιστώνεται με μια σειρά δοκιμών ποιο είναι το μέγιστου μήκους πακέτο που μπορεί να περάσει κατά μήκος ενός μονοπατιού [RFC 1191]. Οι δοκιμές συνίστανται στο να προσπαθεί το ένα άκρο να περάσει στο άλλο μια σειρά από πακέτα φθίνοντος μήκους. Τα πακέτα αυτά είναι κατασκευασμένα έτσι ώστε να απαγορεύεται ο τεμαχισμός τους (μέσω της επιλογής `DF`, Don't Fragment). Όσο υπάρχει κάποιο κομμάτι του μονοπατιού με μικρότερο επιτρεπόμενο μήκος, ο κόμβος στην αρχή του θα απορρίψει το τρέχον πακέτο και θα στείλει πίσω ένα μήνυμα ελέγχου (ICMP) ότι το πακέτο έχει διαγραφεί λόγω περιορισμού μήκους (`Destination Unreachable -Datagram Too Big`). Το πρώτο πακέτο που θα κατορθώσει να φτάσει στην άλλη πλευρά καθορίζει την τιμή του MTU.

Πώς γίνεται η επίθεση στο PMTUD

Ο επιτιθέμενος μπορεί να στείλει ένα κατασκευασμένο πακέτο, με το οποίο παραπλανεί τον αλγόριθμο εκτίμησης του μέγιστου μήκους πακέτου και τον κάνει να φτάσει στο συμπέρασμα ότι το εν λόγω μήκος είναι πολύ μικρό. Συγκεκριμένα μπορεί να στείλει στην πλευρά που είναι στην αρχή του μονοπατιού ένα πακέτο ICMP με τις επιλογές `DF` (Don't Fragment) και `Destination Unreachable`, που υποτίθεται ότι είναι απάντηση σε προηγούμενο πακέτο συγκεκριμένου μήκους που χρησιμοποίησε ο αλγόριθμος (RFC 5927 [RFC 5927]). Συνέπεια θα είναι η μείωση του επιτρεπόμενου μεγέθους πακέτου.

Το γεγονός αυτό έχει δύο συνέπειες: (α) Το ποσοστό που καταλαμβάνει η επικεφαλίδα σε κάθε πακέτο θα αυξηθεί, οπότε η ταχύτητα μετάδοσης της καθαρής πληροφορίας θα μειωθεί, και (β) είναι πιθανό το σύστημα που παράγει τα πακέτα να αυξήσει το ρυθμό μετάδοσης πακέτων προκειμένου να κρατήσει σταθερό το ρυθμό μετάδοσης της πληροφορίας. Σε οποιαδήποτε περίπτωση το σύστημα αυτό επιβαρύνεται. Υπάρχει όμως μια ακόμη χειρότερη εκδοχή. Αν ο επιτιθέμενος καταφέρει να κατεβάσει το όριο πακέτου σε τιμή ίση ή κατώτερη από το μήκος της ίδιας της επικεφαλίδας, το αποτέλεσμα μπορεί να είναι ο τεμαχισμός πακέτων ακόμη και με πληροφορία μήκους ενός byte ή η διακοπή της επικοινωνίας.

Αντίμετρα στην επίθεση στο PMTUD

Στην RFC 5927 περιγράφεται μια διαφοροποίηση της συμπεριφοράς του TCP που βοηθάει στην κατεύθυνση της εξουδετέρωσης της παραπάνω επίθεσης. Όταν κατά την εξέλιξη του αλγορίθμου PMTUD φτάνει ένα πακέτο με την ένδειξη ότι το μήκος πακέτου είναι πολύ μεγάλο η ένδειξη αυτή σημειώνεται, αλλά δεν λαμβάνονται άμεσα μέτρα όσο η σύνδεση φαίνεται να λειτουργεί κανονικά. Μέτρα μείωσης του μήκους λαμβάνονται μόνο όταν διαπιστωθεί υποβάθμιση. Ωστόσο η λειτουργία ενός τέτοιου μηχανισμού θα προκαλούσε καθυστέρηση στη λειτουργία μιας νέας σύνδεσης. Για το λόγο αυτόν ο παραπάνω μηχανισμός έχει διαιρεθεί σε δύο στάδια, (α) στην αρχική ανακάλυψη του μήκους και (β) στην ανανέωση της σχετικής πληροφορίας. Η φάση (α) αφορά σε περιπτώσεις που το TCP προσπαθεί να στείλει πακέτα (τμήματα) μεγαλύτερα από τα προηγούμενα στην ίδια σύνδεση, ενώ η περίπτωση (β) αφορά σε πακέτα ίσα ή μικρότερα από τα προηγούμενα.

Επίθεση μέσω `chargen`

Πρόκειται για μια επίθεση ενίσχυσης που βασίζεται σε ένα πρωτόκολλο που έχει δημιουργηθεί με σκοπό τις δοκιμές. Το *Character Generator Protocol* (CHARGEN) είναι μια υπηρεσία που επιτρέπει σε ένα ξενιστή (host) να συνδεθεί με ένα εξυπηρετητή (server) είτε μέσω TCP είτε μέσω UDP στη θύρα 19 και τότε ο δεύτερος στέλνει μια σειρά χαρακτήρων στον πρώτο (RFC 864). Στην περίπτωση του TCP αυτό διαρκεί όσο παραμένει ανοιχτή η σύνδεση. Στην περίπτωση του TCP κάθε φορά που ο εξυπηρετητής παραλαμβάνει ένα datagram, στέλνει πίσω μια σειρά χαρακτήρων με τυχαίο πλήθος μεταξύ 0 και 512.

Η μέση ενίσχυση που επιτυγχάνεται μέσω UDP chargen είναι γύρω στις 350 φορές. Εν γένει η υπηρεσία παραμένει ανενεργή εκ των προτέρων (by default) και ο διαχειριστής μπορεί να ρυθμίσει το firewall ώστε να μην επιτρέπει κίνηση προς τη θύρα 19. Μπορεί να συνδυασθεί με την υπηρεσία echo για τη δημιουργία ισχυρής κίνησης. Το 2015 η φημολογούμενη τιμή για τη δημιουργία κίνησης 20 Mbps με chargen botnets ήταν στα 5 δολλάρια για μισή ώρα [Sno15; Aka13].

Επίθεση στο SNMP

Τι είναι το SNMP

Το SNMP είναι ένα πρωτόκολλο διαχείρισης εξοπλισμού που είναι δικτυωμένος μέσω Internet, καθώς και διαχείρισης των πρωτοκόλλων Internet (βλ. [Hal05] κεφ. 8.7). Λειτουργεί στο επίπεδο εφαρμογών. Χειρίζεται και δικτυακό εξοπλισμό (π.χ. δρομολογητές, μεταγωγείς, γέφυρες) και συνδεόμενες συσκευές (εκτυπωτές, τηλέφωνα, κάμερες, αισθητήρες), καθώς και τους σχετικούς εξυπηρετητές. Επεμβαίνει είτε όταν δημιουργούνται σφάλματα (π.χ. διακόπτεται απροσδόκητα μια υπηρεσία) είτε όταν βγαίνουν εκτός ορίων οι επιδόσεις δικτύων και υπηρεσιών, ενώ ταυτόχρονα διαχειρίζεται τις παραμέτρους των δικτύων και συστημάτων, έτσι ώστε να λειτουργούν ομαλά.

Ο πράκτορας (agent) του SNMP, ο οποίος είναι εγκατεστημένος σε κάποια συσκευή, επικοινωνεί μέσω UDP με το λογισμικό διαχείρισης του SNMP (SNMP Management Software) με τα μηνύματα (υπό μορφή PDUs, Protocol Data Units) `GetRequest`, `SetRequest`, `GetNextRequest`, `Response`, `Trap`, `GetBulkRequest`, `InformRequest` και είτε απαντάει στο σύστημα διαχείρισης όταν ερωτηθεί (όταν παραλάβει request), είτε το ειδοποιεί με δική του πρωτοβουλία όταν υπάρξουν γεγονότα που απαιτούν επέμβαση της διαχείρισης (*trap events, περιστατικά παγίδευσης*). Το SNMP για να μεταδίδει τα μηνύματά του χρησιμοποιεί τα ίδια πρωτόκολλα TCP/IP που χρησιμοποιούν οι εφαρμογές. Κάθε συνολικό μήνυμα του SNMP αποτελείται από την παραπάνω PDU και μια επικεφαλίδα που περιγράφεται στην RFC 1157.

Οι πληροφορίες που συγκεντρώνει ένας πράκτορας προέρχονται από ένα σύνολο αντικειμένων τα οποία αυτός διαχειρίζεται. Τα αντικείμενα αυτά περιέχονται στη Βάση Πληροφοριών Διαχείρισης (Management Information Base, MIB).

Η ασφάλεια είναι ένα από τα ασθενή σημεία του SNMP. Στις εκδόσεις 1 και 2 ο έλεγχος ταυτότητας (authentication) γίνεται μέσω συνθηματικού (password) το οποίο ανταλλάσσεται ανάμεσα στις δύο πλευρές ακρυπτογράφητο (αυτό είναι το λεγόμενο *community string*). Η έκδοση 3 (SNMPv3, RFC 2271, RFC 2275) κατά βάση έχει βελτιώσει την ασφάλεια. Διατηρεί το πρωτόκολλο της προηγούμενης έκδοσης (αν και έχει αλλάξει δραστικά τους όρους, έτσι ώστε εκ πρώτης όψεως να φαίνεται διαφορετικό) προσθέτοντας ασφάλεια με κρυπτογραφικές μεθόδους (και πρόσθετες δυνατότητες απομακρυσμένου καθορισμού της διάταξης, remote configuration). Μια από τις βασικές αλλαγές στην ορολογία που έφερε το SNMPv3 είναι η μετονομασία των δύο

πλευρών, της πλευράς των πρακτόρων και της πλευράς της διαχείρισης, σε *οντότητες SNMP* (SNMP entities) γενικώς. Κάθε οντότητα του SNMP χαρακτηρίζεται από μια ταυτότητα (identifier) που λέγεται `SNMPEngineID`. Οι παράμετροι που προσδιορίζουν το εκάστοτε χρησιμοποιούμενο μοντέλο ασφάλειας συμπυκνώνονται σε μια οκτάδα.

Η επαλήθευση ταυτότητας (authentication) των μηνυμάτων του SNMPv3 γίνεται μέσω του Message Digest 5 (MD5) ή του Secure Hash Algorithm 1 (SHA1) [MS05]. Ο MD5 δημιουργεί μια σύνοψη (digest) μήκους 128 bits, ενώ ο SHA1 σύνοψη μήκους 160 bits και χρησιμοποιούνται σε συνδυασμό με τον αλγόριθμο HMAC (Hashing for Message Authentication.) Για τον υπολογισμό της σύνοψης μετά τα δεδομένα προστίθεται ένα μυστικό κλειδί (με μήκος 8 bytes ή μεγαλύτερο), γνωστό μόνο στις δύο πλευρές. Τα δεδομένα προστατεύονται με συμμετρική κρυπτογράφηση μέσω του αλγορίθμου CBC-DES, οπότε και πάλι οι δύο πλευρές πρέπει να έχουν ανταλλάξει ένα μυστικό κλειδί.

Προβλήματα ασφάλειας έχουν όλες οι εκδόσεις του SNMP. Σύμφωνα με τις εκδ. 1 και 2 τα δεδομένα είναι απροστάτευτα και σε μορφή αναγνώσιμη από τον άνθρωπο, συνεπώς μπορεί κανείς να αναχαιτίσει τα μηνύματα και να τα διαβάσει ή να αλλάξει το περιεχόμενό τους. Επίσης εξ αιτίας της χρήσης του UDP κινδυνεύουν από αλλοίωση διεύθυνσης. Τέλος, και η 3η έκδοση είναι τρωτή από επιθέσεις αποκάλυψης κλειδιών, μέσω λεξικού ή και εξαντλητικής αναζήτησης (brute force) [Aka13].

Επίθεση ανάκλασης (DrDoS) μέσω SNMP

Στο προπαρασκευαστικό στάδιο ο επιτιθέμενος χρειάζεται ένα κατάλογο από ξενιστές (hosts) του SNMP, καθώς και συνθηματικά (community strings). Την ανίχνευση για τον σχηματισμό του καταλόγου μπορεί να κάνει με σάρωση θυρών (port scanning). Η επίθεση γίνεται ως εξής:

1. Ο επιτιθέμενος στέλνει μηνύματα `GetBulkRequest` με αλλοιωμένη διεύθυνση (τη διεύθυνση του θύματος) σε επιλεγμένους ξενιστές που θα παίξουν το ρόλο του σημείου ανάκλασης.

2. Οι απαντήσεις επιστρέφουν προς το θύμα. Η ενίσχυση είναι περίπου $\times 3$.

3. Το θύμα γίνεται μη προσβάσιμο εξ αιτίας της αυξημένης κίνησης.

Με την εντολή `GetBulkRequest` (SNMPv2) ζητείται η μαζική αποστολή της διαθέσιμης πληροφορίας. Ο επιτιθέμενος μπορεί να διαλέξει πράκτορες με μεγάλο κατά το δυνατό μέγεθος MIB.

Επίθεση μέσω πρωτοκόλλου NTP

Υποδομή και πρωτόκολλο NTP

Το NTP (Network Time Protocol) έχει ως στόχο το συγχρονισμό των ρολογιών υπολογιστών που βρίσκονται σ' ένα δίκτυο (RFC 958, βλ. και `ntp.org`). Χρησιμοποιεί για το σκοπό αυτό ένα κατανεμημένο σύνολο 3960 εξυπηρετητών, από τους οποίους οι 2630 είναι στην Ευρώπη (αριθμός που ισχύει τον Οκτ. 2015). Ο στόχος του NTP είναι να δίνει σφάλμα μερικών δεκάδων msec το πολύ. Χρησιμοποιεί τις ενδείξεις ρολογιών μεγάλης ακρίβειας (ατομικών ρολογιών καισίου και ρουβιδίου, ρολογιών του GPS κ.λπ.) ως είσοδο σε ένα «στρώμα» (stratum) από υπολογιστές, που διαθέτουν με αυτό τον τρόπο «από πρώτο χέρι» ενδείξεις χρόνου με σφάλμα λίγων μsec. Με αυτούς

```
myComputer:~ abc$ ntpq -pn
 remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*17.253.54.125   .GPSs.      1 u  102   64  377   49.885    5.628   0.836
myComputer:~ abc$ ntpdate -q 17.253.54.125
server 17.253.54.125, stratum 1, offset 0.003953, delay 0.07561
15Oct11:59:24 ntpdate[4770]:adjust time server 17.253.54.125 offset 0.003953sec
```

Σχήμα 6.6: «Τι ώρα είναι; Κοντεύει μεσημέρι.»

συνδέεται το δεύτερο στρώμα από υπολογιστές, προφανώς με μεγαλύτερο σφάλμα, ύστερα το τρίτο κ.ο.κ. μέχρι 15 στρώματα. Ένας υπολογιστής που θέλει να έχει πρόσβαση στην ένδειξη χρόνου επικοινωνεί με τρεις ή περισσότερους εξυπηρετητές και παίρνει τις ενδείξεις τους, λαμβάνει όμως υπόψη και την καθυστέρηση των πακέτων που προέρχονται από αυτούς (βλ. RFC 5905 [RFC 5905]). Στο Σχ. 6.6 φαίνεται ένα παράδειγμα επικοινωνίας από command line με κάποιον εξυπηρετητή του στρώματος 1.

Επίθεση μέσω NTP

Η επίθεση μέσω NTP είναι μια κλασσική επίθεση ανάκλασης και ενίσχυσης. Ο επιτιθέμενος στέλνει μια σειρά από αιτήματα χρόνου («τι ώρα είναι;»), κατά προτίμηση χρησιμοποιώντας ένα δίκτυο ανδρεικέλων, αν αυτό είναι διαθέσιμο, σε κάθε περίπτωση όμως με αλλοιωμένη διεύθυνση, τη διεύθυνση του θύματος [Aka13]. Τα αιτήματα μπορούν να απευθύνονται σε διαφορετικούς εξυπηρετητές του NTP, αλλά φέρουν πάντοτε τη διεύθυνση προέλευσης του θύματος. Οι εξυπηρετητές προφανώς απαντούν στο θύμα, το οποίο καλείται να παραλάβει την αυξημένη κίνηση που δημιουργείται από τις απαντήσεις.

ΚΕΦΑΛΑΙΟ 7

Τείχη προστασίας

Το *τείχος προστασίας* (firewall) είναι το πιο κοινό σύστημα προστασίας από εισβολές και χρησιμοποιείται σε παντός είδους συστήματα, υπολογιστές, routers, υποδίκτυα, συσκευές IoT κ.λπ. Είναι ένας «φράχτης» με μια και μοναδική είσοδο, όπου ελέγχονται τα εισερχόμενα και εξερχόμενα πακέτα και είναι δυνατό να υλοποιηθεί για διαφορετικά στρώματα των δικτυακών πρωτοκόλλων. Είναι ένα εργαλείο με σχετικά περιορισμένες δυνατότητες, αλλά προτιμάται εξ αιτίας της απλότητάς του.

Η βασική ιδέα είναι γνωστή από την αρχαιότητα: Σε μια περιτειχισμένη πόλη βάζουμε λίγες πύλες εισόδου-εξόδου και ελέγχουμε όποιον περνάει, όταν η πύλη είναι ανοιχτή, ενώ έχουμε τη δυνατότητα να διαφοροποιούμε την πολιτική εισόδου-εξόδου σε διαφορετικές χρονικές περιόδους ή και να την κλείνουμε όλωσδιόλου. Όπως και τότε, η μέθοδος αυτή έχει γνωστές αδυναμίες, εσωτερικούς εχθρούς, λανθασμένες πολιτικές ελέγχου, εξαπάτηση των ελεγχόντων, παραβιάσεις σε άλλα σημεία αδυναμίας του τείχους, προσπάθειες κατεδάφισης του τείχους από πολιορκητές κ.α. Επίσης τα τείχη χρειάζονται προσεκτικό σχεδιασμό και υλοποίηση, κατάλληλη επιλογή της θέσης τους, ενώ μπορούν εντάσσονται σε ένα συνολικό σύστημα προστασίας και *άμυνας σε βάθος* (δηλαδή άμυνας με πολλαπλά μέσα και σε διάφορα σημεία). Όλα αυτά τα ζητήματα έχουν το ψηφιακό τους αντίστοιχο στα σημερινά firewalls. Ο έλεγχος μπορεί να γίνεται και προς τις δύο κατευθύνσεις στο σημείο ελέγχου.

Οδηγίες για τη χρήση τειχών προστασίας δίνονται μέσω της σύστασης NIST 800-41 [SH09].

7.1 Τύποι ελέγχου

Στο σημείο ελέγχου ένα πακέτο συγκεκριμένου στρώματος συνήθως ελέγχεται ως προς τις επικεφαλίδες του στρώματος, μολονότι στο φορτίο του υπάρχουν δεδομένα και ανώτερων στρωμάτων που θα μπορούσαν να αξιοποιηθούν. Επίσης στον έλεγχο μπορεί να αξιοποιηθεί η πληροφορία η σχετική με το σημείο εισόδου, π.χ. από ποια εισερχόμενη ζεύξη ενός κόμβου έχει έρθει το πακέτο.

Οι συνηθισμένοι έλεγχοι λαμβάνουν υπόψη τις διευθύνσεις πηγής και προορισμού, το πρωτόκολλο δικτύου ή μεταφοράς (TCP, UDP, ICMP), θύρες πηγής και προορισμού σε επίπεδο συνόδου, διεπαφή μέσω της οποίας φτάνει το πακέτο, κατεύθυνση (προς τα μέσα ή προς τα έξω) κ.λπ. Επίσης οι έλεγχοι οφείλουν να λαμβάνουν υπόψη

ορισμένα άλλα ζητήματα, όπως ο κατακερματισμός των πακέτων (packet fragmentation). Ο κατακερματισμός σε ορισμένες περιπτώσεις είναι ένδειξη ύποπτης συμπεριφοράς, ωστόσο η ενσωμάτωση σχετικού ελέγχου χωρίς τη χρήση άλλων κριτηρίων μπορεί να οδηγήσει στην καταστροφή πακέτων που δεν ανήκουν σε επίθεση.

Παρ' όλο που η έμφαση είναι γενικά στον έλεγχο της κίνησης που προσπαθεί να εισέλθει στο δίκτυο ενός οργανισμού ή μιας επιχείρησης (ingress filtering), συνήθως είναι σκόπιμος και ο έλεγχος της εξερχόμενης κίνησης (egress filtering). Μια τέτοια πολιτική μπορεί να εμποδίσει επιτιθέμενους που έχουν ήδη μολύνει υπολογιστές εσωτερικά στον οργανισμό και έχουν εγκαταστήσει malware που θα κάνει μια επίθεση κίνησης σε αυθαίρετους στόχους. Η πολιτική αυτή θα μπορούσε π.χ. να εγκαταστήσει ελέγχους της διεύθυνσης πηγής των εξερχομένων πακέτων, ώστε το malware να μη μπορεί να χρησιμοποιήσει αλλοιωμένες διευθύνσεις πηγής.

Σε κάθε πακέτο εφαρμόζεται με τη σειρά ένα σύνολο κανόνων. Κάθε τέτοιος κανόνας εκφράζεται ως λογική πρόταση πάνω σε διάφορα πεδία του πακέτου, π.χ. η διεύθυνση αποστολής να είναι μια συγκεκριμένη και η διεύθυνση προορισμού ομοίως. Ανάλογα με την τιμή αλήθειας (αληθή ή ψευδή) που θα δώσει η εφαρμογή της πρότασης, το πακέτο γίνεται ή δεν γίνεται δεκτό (περνάει ή δεν περνάει).

Παρ' όλο που η αποδοχή ή απόρριψη ενός πακέτου θα μπορούσε να στηριχθεί σε ένα και μόνο πολύπλοκο κανόνα, συνήθως υλοποιείται μέσα από μια σειρά κανόνων, όπου όλοι είναι θετικοί ή όλοι είναι αρνητικοί. Θετικός είναι ένας κανόνας όταν το πακέτο γίνεται δεκτό παράγοντας την αληθή τιμή (true), διαφορετικά απορρίπτεται, δηλαδή η ικανοποίηση του κανόνα αποτελεί λόγο αποδοχής. Αρνητικός είναι ένας κανόνας όταν το πακέτο γίνεται δεκτό παράγοντας την ψευδή τιμή (false), διαφορετικά απορρίπτεται, δηλαδή η ικανοποίηση του κανόνα αποτελεί λόγο απόρριψης.

Θετικό τείχος

Ένα *θετικό* τείχος επιτρέπει την είσοδο μόνο σε όσα πακέτα ανταποκρίνονται σε συγκεκριμένες προδιαγραφές. Δηλαδή χρησιμοποιείται μια πολιτική του τύπου «μόνο οι καλοί περνούν» (ή «όλοι θεωρούνται κακοί μέχρις αποδείξεως του εναντίου»), όπου κάθε κανόνας προσπαθεί να βρει ένα λόγο να γίνει αποδεκτό το πακέτο κι ένας κανόνας αρκεί. Όταν η ετυμηγορία σύμφωνα με τον κανόνα είναι να γίνει δεκτό το πακέτο, τότε γίνεται αμέσως δεκτό, διαφορετικά προωθείται στον επόμενο κανόνα προς περαιτέρω εξέταση. Στο τέλος υπάρχει ένας κανόνας που απορρίπτει όλα τα πακέτα που φτάνουν ως εκεί. Μια τέτοια πολιτική καταλήγει σε επιλογή πακέτων προς αποδοχή και αντιστοιχεί σε μια εν γένει «σφιχτή» πολιτική εισόδου, που είναι πιο πιθανό να υλοποιεί μια εταιρία και να δέχεται μόνο τη «νόμιμη» κίνηση. Μια εταιρία θα μπορούσε να εφαρμόζει και κανόνες μη σχετικούς με την ασφάλεια, αλλά σχετικούς τους σκοπούς της διερευνώντας περαιτέρω το περιεχόμενο των πακέτων.

Αρνητικό τείχος

Ένα *αρνητικό* τείχος επιτρέπει την είσοδο σε όλη την κίνηση εκτός εκείνης που θεωρείται ύποπτη. Δηλαδή χρησιμοποιείται μια πολιτική «μαύρης λίστας», δηλαδή της λογικής «μόνο οι κακοί δεν περνούν» (ή «όλοι θεωρούνται καλοί μέχρις αποδείξεως του εναντίου»), όπου κάθε κανόνας προσπαθεί να προσδιορίσει κατά πόσο ένα πακέτο είναι ύποπτο και είναι αρκετά ισχυρός για να κοπεί το πακέτο. Όταν η ετυμηγορία σύμφωνα με τον κανόνα είναι να μη γίνει δεκτό το πακέτο, το πακέτο καταστρέφεται, διαφορετικά προωθείται στον επόμενο κανόνα. Στο τέλος υπάρχει ένας κανόνας που επιτρέπει την είσοδο σε όλα τα πακέτα που φτάνουν ως εκεί. Μια τέτοια πολιτική

καταλήγει σε επιλογή πακέτων προς απόρριψη, ενώ αφήνει όλα τα άλλα να περνούν, οπότε αντιστοιχεί σε μια εν γένει «χαλαρή» πολιτική εισόδου. Μια τέτοια πολιτική μπορεί να υλοποιείται π.χ. από ένα πανεπιστήμιο, ώστε να κόβει μόνο την επικίνδυνη κίνηση. Στατιστικά επιτρέπει σε περισσότερη κίνηση να περνάει σε σύγκριση με ένα τείχος του τύπου της προηγούμενης παραγράφου.

Επιθεώρηση με γνώση της κατάστασης

Στην λεγόμενη *επιθεώρηση με γνώση της κατάστασης* (stateful inspection) ένα firewall χρησιμοποιεί περισσότερα δεδομένα από όσα απλώς εμφανίζονται σε ένα πακέτο. Μια συνηθισμένη τέτοια περίπτωση είναι όταν ένα τείχος που κάνει έλεγχο σε επίπεδο δικτύου συντηρεί ένα πίνακα ανοιχτών συνδέσεων (που αποτελούν πληροφορία του υπερκείμενου επιπέδου σύνδεσης), επομένως επιτρέπει πακέτα που αντιστοιχούν σε όσες συνδέσεις είναι εκείνη τη στιγμή ανοιχτές. Ένας πιο προχωρημένος έλεγχος περιλαμβάνει τον αύξοντα αριθμό TCP.

Περαιτέρω επιθεώρηση πακέτου σε βάθος

Ως γνωστόν κάθε στρώμα που βρίσκεται πιο κάτω στην ιεραρχία αντιστοιχεί σε μια επικεφαλίδα που είναι πιο έξω. Με άλλα λόγια ένα φίλτρο επιπέδου δικτύου επιθεωρεί τις αντίστοιχες επικεφαλίδες, ενώ οι επικεφαλίδες του επιπέδου σύνδεσης αποτελούν «φορτίο» για το πακέτο. Είδαμε βέβαια προηγουμένως ότι σε ένα stateful packet inspection αξιοποιούνται και πληροφορίες ανώτερου επιπέδου. Μια ακόμη πιο προχωρημένη πολιτική στην ίδια κατεύθυνση μπορεί να αξιοποιεί δεδομένα από το επίπεδο εφαρμογών, δηλαδή να εισέρχεται σε μεγαλύτερο βάθος στο φορτίο. Ένα τέτοιο φίλτρο θα μπορούσε να ελέγξει π.χ. αν το συνημμένο σε ένα μήνυμα email τηρεί ορισμένες προδιαγραφές. Με την λογική αυτήν μπορούν να ενσωματωθούν σε ένα firewall έλεγχοι που αντιστοιχούν σε επιλεγμένες εφαρμογές. Σε μια προχωρημένη έκδοση ενός τέτοιου ελέγχου μπορεί να παρακολουθείται στενά η κατάσταση της εφαρμογής (μέσω ενός μοντέλου καταστάσεων) και να ελέγχονται τα πακέτα αν εμπίπτουν στην αναμενόμενη κατάσταση της εφαρμογής.

Πύλες αντιπροσώπων εφαρμογών

Όταν είναι επιθυμητός ο έλεγχος σε επίπεδο εφαρμογών μπορεί να χρησιμοποιηθεί μια *πύλη αντιπροσώπων εφαρμογών* (application proxy gateway) ή *πύλη επιπέδου εφαρμογών* (application level gateway - ALG). Σ' αυτήν την περίπτωση η επικοινωνία μεταξύ client και server γίνεται μέσω δύο χωριστών συνδέσεων, μιας σύνδεσης μεταξύ client και server-proxy και μια σύνδεσης μεταξύ server και server-proxy. Όταν ο client θελήσει να συνδεθεί με τον server, εμφανίζεται αντ' αυτού ο αντιπρόσωπος του server στο firewall και δίνει στον client την εντύπωση ότι έχει συνδεθεί με τον server. Η διαπραγμάτευση για την υπηρεσία γίνεται μεταξύ του client και του server proxy. Στη συνέχεια ο proxy επικοινωνεί με τον server, προς τον οποίο εμφανίζεται ως client και προωθεί προς τη μια και την άλλη κατεύθυνση τα σχετικά πακέτα.

ΚΕΦΑΛΑΙΟ 8

Ανίχνευση εισβολής

Ἄνδρα μοι ἔννεπε, Μοῦσα,
πολύτροπον

Οδύσσεια, ραψ. α', 1.

8.1 Γενικά περί ανίχνευσης

Ο όρος *Intrusion Detection System* (IDS) χαρακτηρίζει αυτοματοποιημένα συστήματα που ανιχνεύουν εισβολές, με σκοπό την περαιτέρω αντιμετώπισή τους. Δυστυχώς κάποιες εισβολές μπορούν να είναι πρωτότυπες με μόνο περιορισμό την ανθρώπινη εφευρετικότητα και να εκμεταλλεύονται προηγουμένως άγνωστες τρωτότητες. Ωστόσο υπάρχει χώρος για όλην την κλίμακα δεξιοτήτων των επιτιθεμένων. Οι πιο άπειροι μπορούν να πάρουν έναν ήδη δημοσιευμένο κώδικα, να κάνουν λίγες ή καθόλου μεταβολές, και να κάνουν τη δική τους επίθεση.

Τα συστήματα ανίχνευσης κατατάσσονται σε δύο ευρείες κατηγορίες ως προς τη μέθοδο που χρησιμοποιούν:

Ανίχνευση υπογραφής (signature detection): Βασίζεται σε μια καταγραμμένη και αναλυμένη συμπεριφορά επιθέσεων που έχει αποθηκευτεί σε μια βάση δεδομένων (όπου ως «υπογραφή» νοούνται συγκεκριμένα χαρακτηριστικά της επίθεσης).

Ανίχνευση ανωμαλίας (anomaly detection): Η συνήθης «ομαλή» κατάσταση του συστήματος καταγράφεται σε μια σειρά παραμέτρων που ελέγχονται περιοδικά ως προς το αν βρίσκονται μέσα στα αναμενόμενα διαστήματα τιμών. Αποκλίσεις εκτός των διαστημάτων θεωρούνται ύποπτες.

Η ανίχνευση ενός πακέτου με φορτίο που περιέχει συγκεκριμένο κώδικα κακόβουλου λογισμικού ή περιέχει προδιαγεγραμμένη δρομολόγηση πέφτει στην πρώτη κατηγορία. Αν ο επιτιθέμενος χρησιμοποιήσει διαφορετικό κώδικα με το ίδιο αποτέλεσμα, η επίθεση αυτή δεν θα ανιχνευθεί. Γενικά αυτή η μέθοδος δεν μπορεί να ανιχνεύσει νέες επιθέσεις ή και παραλλαγές παλαιών.

Σε συστήματα που η ανίχνευση συγκεκριμένων τιμών παραμέτρων ή strings είναι εύκολο να δημιουργηθούν λανθασμένα θετικές ανιχνεύσεις (false positives). Π.χ. αν σε υπολογιστή με λειτουργικό linux ανιχνευθεί ένα string ιού που προσβάλλει μόνο υπολογιστές με Windows, ο συναγερμός είναι περιττός. Η λύση ενός τέτοιου προβλήματος βρίσκεται στη χρήση πιο πολύπλοκων κανόνων, με τους οποίους η τελική απόφαση εξαρτάται από περισσότερες προϋποθέσεις [SBK16].

Η μέθοδος ανίχνευσης ανωμαλιών βασίζεται στην ασυνήθιστη συμπεριφορά της επίδοσης ενός συστήματος. Π.χ. υπερβολική κίνηση προς ή από μια συγκεκριμένη διεύθυνση, υπερβολική κατανάλωση υπολογιστικής ισχύος ή μνήμης, διαφορετικός χρονισμός της κίνησης, όλα αυτά μπορεί να είναι σημάδια μιας επίθεσης, αλλά πολλά εξ αυτών μπορεί να προκληθούν από τυχαίους παράγοντες ή και συστηματικά σφάλματα. Η ερμηνεία ενός φαινομένου ως μη κανονικού βασίζεται στη γνώση της κανονικότητας, δηλαδή στη συλλογή δεδομένων γύρω από τις κανονικές συνθήκες, αλλά η κανονικότητα δεν είναι οπωσδήποτε σταθερή. Η ανίχνευση ανωμαλιών παράγει πολλούς λανθασμένα θετικούς συναγερμούς, ενώ ένας έξυπνος επιτιθέμενος μπορεί να ρυθμίσει την επίθεσή του ώστε να περάσει απαρατήρητος. Παρ᾽ όλα αυτά μπορεί να ανιχνεύσει και νέους τύπους επιθέσεων από τα αποτελέσματα που προκαλούν, αν και η ανάλυσή τους θα χρειαστεί περαιτέρω προσπάθεια. Ένα καλό σύστημα ανίχνευσης χρησιμοποιεί συνδυασμό μεθόδων.

Πέραν των αρχείων καταγραφής γεγονότων που συμβαίνουν εσωτερικά σε μια μηχανή (από το λειτουργικό σύστημα ή και από πρόσθετο λογισμικό), διάφορα πρότυπα προβλέπουν ευρύτερες καταγραφές γεγονότων. Πρότυπα ασφάλειας για επιχειρήσεις και οργανισμούς, όπως το ISO 27001, προβλέπουν την συστηματική καταγραφή γεγονότων σχετικών με την ασφάλεια, ώστε στη χειρότερη περίπτωση να μπορεί να γίνει διερεύνηση ενός περιστατικού έστω και εκ των υστέρων, να μπορούν να αναζητηθούν υπεύθυνοι κ.λπ. Επίσης γενικότερα πρότυπα με έμφαση στις καθαυτό λειτουργίες μιας επιχείρησης (π.χ. το ISO 9001) προδιαγράφουν καταγραφές, που μπορεί τελικά να αποδειχτούν χρήσιμες για την διευρεύνηση ενός περιστατικού ασφάλειας. Επιχειρήσεις που θέλουν να πιστοποιηθούν για την τήρηση των σχετικών προτύπων υποχρεώνονται να δημιουργήσουν το κατάλληλο σύστημα καταγραφών και να εκπαιδεύσουν το προσωπικό τους να το τηρεί.

Ένα παράδειγμα (που προέρχεται από πραγματικά περιστατικά) μπορεί να δώσει μια ιδέα για τη σημασία της καταγραφής: Σε εταιρία τηλεπικοινωνιών διαπιστώθηκε ότι είχε εγκατασταθεί παράνομα λογισμικό παρακολουθήσεων. Μετά από έλεγχο και διασταύρωση στοιχείων οι ερευνητές της υπόθεσης κατέληξαν στην υπόνοια ότι είχε γίνει μια πρώτη εγκατάσταση ενός rootkit με φυσική πρόσβαση σε συγκεκριμένο χρόνο εκτός κανονικού ωραρίου και σε μηχανή που ήταν σε συγκεκριμένο χώρο. Στη συνέχεια αναζήτησαν τα βιβλία επισκεπτών των εγκαταστάσεων από το θυρωρείο εισόδου, όμως οι συγκεκριμένες σελίδες έλειπαν.

Κίνητρα και κατηγορίες επιτιθεμένων

Ο επιτιθέμενος μπορεί να έχει πολύ διαφορετικά κίνητρα [Wea+03] που διαμορφώνουν τις προθέσεις του και τελικά τον τρόπο επίθεσης. Καθορίζουν επίσης στατιστικά τις πιθανές του δεξιότητες. Μια απλή ταξινόμηση κινήτρων είναι η εξής:

Περιέργεια και πειραματισμός Εδώ εντάσσονται δημιουργοί κακόβουλου λογισμικού από παιδιά που διασκεδάζουν (και αδιαφορούν για τις ζημιές που μπορεί να προκληθούν) ως ερευνητές που πειραματίζονται (και έχουν μια πιο υπεύθυνη στάση, αλλά μπορεί να χάσουν τον έλεγχο).

Ασκήσεις επίδειξης και δύναμης Περιλαμβάνονται συνήθως ανοργάνωτοι μη συστηματικοί επιτιθέμενοι που θέλουν να δείξουν τι είναι ικανοί να επιτύχουν.

Εμπορικό πλεονέκτημα Εντάσσονται επαγγελματίες ως και μεγάλες εταιρίες που ασκούν έτσι αθέμιτο ανταγωνισμό ή βιομηχανική κατασκοπεία. Εδώ η επίθεση είναι πιθανό να ανατεθεί σε εξωτερικούς επαγγελματίες (με κίνητρο των τελευταίων την αμοιβή).

Τυχαία διαμαρτυρία από κάποιον που θεωρεί εαυτόν αδικημένο, ή έχει απολυθεί από μια εταιρία ή είναι απλώς ανισόρροπος.

Πολιτική, ακτιβιστική, τρομοκρατική δράση από άτομα ή οργανωμένες ομάδες, που όμως δεν ανήκουν σε μυστικές ή στρατιωτικές υπηρεσίες.

Έγκλημα από μεμονωμένα άτομα ή και οργανωμένες εγκληματικές ομάδες με οικονομικό στόχο.

Κυβερνοπόλεμος που ασκείται από ειδικές κρατικές μυστικές υπηρεσίες στα πλαίσια της αντιπαράθεσης μεταξύ κρατών.

Στο σημείο αυτό αξίζει να σχολιασθεί το γεγονός ότι οι παραπάνω κατηγορίες επιτιθεμένων είναι κάθε άλλο παρά κλειστές και σταθερές. Αν παλιότερα ένας μεμονωμένος χάκερ φιλοδοξούσε να γίνει φημισμένος και τελικά να μεταπηδήσει στην πλευρά της άμυνας, σήμερα οι κορυφαίοι κυβερνο-εγκληματίες μπορούν να αξιοποιούνται από μυστικές υπηρεσίες και να διατηρούν το δικαίωμα λαφυραγώγησης για να φαίνονται ανεξάρτητοι. Ταυτόχρονα, στο υψηλό επίπεδο επιθέσεων υπάρχει ένας ακήρυχτος πόλεμος, όπου όλα επιτρέπονται επειδή «κι εμείς τα ίδια θα κάναμε αν μπορούσαμε» κι επειδή οι κυβερνήσεις δείχνουν ανοχή όσο απειλούνται μόνο οι εκτός των συνόρων μιας χώρας, αν δεν παρέχουν και στήριξη. Αν η πυρηνική απειλή με τις ακραίες της συνέπειες έχει ακυρώσει τις μεγάλης κλίμακας πολεμικές συρράξεις, ο κυβερνοπόλεμος είναι ένα πεδίο όπου καθένας μπορεί να εκβιάσει όσους και όποιους μπορεί, επειδή οι συνέπειες είναι σχετικά περιορισμένες. Ωστόσο οι απειλούμενες συνέπειες συνεχώς αναβαθμίζονται μέσω του μαζικού ελέγχου κυβερνο-φυσικών συστημάτων. Ένα μαζικό black-out στο δίκτυο διανομής ρεύματος μπορεί να έχει καταστροφικές συνέπειες και είναι πλέον μέσα στις δυνατότητες όχι μόνο των μυστικών κρατικών υπηρεσιών των υπερδυνάμεων, αλλά και των εγκληματικών οργανώσεων (όπως έδειξε π.χ. η υπόθεση της *Colonial Pipeline*).[1] Η δυσλειτουργία ενός πυρηνικού σταθμού μπορεί να αποδειχθεί μοιραία.

Μια επίθεση μπορεί να είναι ένα πολύπλοκο και συχνά πρωτόγνωρο φαινόμενο που μπορεί να αναλυθεί καλύτερα από τον άνθρωπο και συχνά μόνο από αυτόν, παρά την αυξανόμενη χρήση μεθόδων τεχνητής νοημοσύνης. Ωστόσο ο άνθρωπος είναι μόνο η κορυφή μιας πυραμίδας που ξεκινάει μέσα από τις ίδιες τις μηχανές και χρειάζεται στη βάση οπωσδήποτε τη συλλογή και επεξεργασία δεδομένων, εν πολλοίς με τρόπους αυτοματοποιημένους. Υπάρχουν διάφοροι λόγοι που καθιστούν απαραίτητη τη χρήση αυτοματισμών και μηχανικής αντιμετώπισης σε συστήματα ανίχνευσης επιθέσεων και λήψης μέτρων:

[1] Τον Μάιο του 2021 η εγκληματική οργάνωση DarkSide (εικαζόμενη ρωσικής προέλευσης) έκανε μια επίθεση για λύτρα στην αμερικανική εταιρία διανομής καυσίμων (μέσω δικτύου αγωγών) *Colonial Pipeline*. Η εταιρία φημολογείται ότι τελικά πλήρωσε $ 5000000, ενώ στην ανατολική ακτή των ΗΠΑ σημειώθηκαν ελλείψεις καυσίμων και πανικός.

Η χρονική κλίμακα: Οι καθαυτό βλαβερές ενέργειες μιας επίθεσης (π.χ. η διαγραφή κρίσιμων αρχείων) μπορεί να εκτελούνται σε κλάσμα δευτερολέπτου, δηλαδή κάτω από τον χρόνο αντίληψης και αντίδρασης ενός ανθρώπου χειριστή. Η χρονική κλίμακα των βλαπτικών ενεργειών ή των μεταβολών που επιφέρει μια επίθεση στην πραγματικότητα ποικίλλει σε μεγάλο βαθμό. Ενέργειες όπως η εκμετάλλευση μια τρωτότητας υπερχείλισης ενός καταχωρητή, μαζί με την εγκατάσταση βλαπτικού κώδικα και την εκτέλεση του κώδικα μπορεί να είναι χρονικά πολύ σύντομες. Όμως μια επίθεση κίνησης διαρκεί από μερικά λεπτά ως μερικές μέρες. Η παρακολούθηση ενός χρήστη με keylogging μπορεί να διαρκεί μήνες. Το εγκατεστημένο σε μια προσβεβλημένη μηχανή λογισμικό ενός botnet μπορεί επίσης να παραμένει για καιρό, ακόμη και αν είναι ανενεργό.

Η παρατηρησιμότητα: Τα περισσότερα από τα γεγονότα που αποτελούν ενδείξεις μιας επίθεσης είναι μη παρατηρήσιμα σε ένα σύστημα ανθρώπου-μηχανής που είναι προσανατολισμένο σε συγκεκριμένη λειτουργικότητα. Μια δημοφιλής λύση για την εμπλοκή του ανθρώπου στη διαχείριση ασφάλειας είναι η μέθοδος των συναγερμών. Όταν η συλλογή δεδομένων και η περαιτέρω επεξεργασία τους οδηγεί «ύποπτα» συμπτώματα, παράγεται ένας συναγερμός και αφήνεται στην κρίση ενός χειριστή η περαιτέρω διαχείριση της κατάστασης. Ωστόσο η εν θερμώ διαχείριση ενός συναγερμού έχει νόημα μόνο για γεγονότα που εμπίπτουν στην ανθρώπινη χρονική κλίμακα, διαφορετικά πρόκειται μόνο για διαχείριση post mortem. Μια άλλη δυσκολία ενός τέτοιου συστήματος είναι η αξιολόγηση και προτεραιοποίηση των συναγερμών σε ένα πολύπλοκο σύστημα και η πιθανότητα να αγνοηθούν ή να θεωρηθούν δευτερεύοντα περιστατικά που αργότερα αποδεικνύονται κρίσιμα.

Ο χρόνος αντίδρασης: Η γρήγορη αντίδραση είναι σημαντική σε στιγμιαίες επιθέσεις, αλλά δεν παύει να είναι σημαντική ακόμη και σε επιθέσεις με διάρκεια λεπτών της ώρας ή μεγαλύτερη, δεδομένου ότι μια επίθεση μπορεί να παρεμποδισθεί σε αρχικό της στάδιο, π.χ. πριν να κρυπτογραφηθούν όλα τα αρχεία σε μια επίθεση για λύτρα ή πριν να κορεσθεί από κίνηση μια ζεύξη ή ένας server.

Η πολυπλοκότητα της αντίδρασης: Στην περίπτωση γνωστών επιθέσεων που αντιμετωπίζονται με συνδυασμό μέτρων η ενορχήστρωση της λήψης μέτρων γίνεται καλύτερα από ένα αυτοματοποιημένο σύστημα.

Στον αγώνα μεταξύ επίθεσης και άμυνας αμφότερες οι πλευρές συγκεντρώνουν στοιχεία για την απέναντι πλευρά και προσπαθούν να τα αξιοποιήσουν κατάλληλα. Έχει προταθεί από την πλευρά της άμυνας η μελέτη του κύκλου ζωής είτε του κακόβουλου λογισμικού είτε της επίθεσης, προκειμένου να αξιοποιηθούν οποιεσδήποτε ενέργειες μπορούν να προδώσουν την επίθεση, κατά προτίμηση σε πρώιμο στάδιο και πάντως πριν την εκδήλωση των βλαπτικών ενεργειών. Δυστυχώς ένας κύκλος ζωής επίθεσης δεν αντιστοιχεί σε ένα μοναδικά καθορισμένο σενάριο εξέλιξης, αντίθετα σε ένα πολύπλοκο σύστημα τα σενάρια εξέλιξης ενός συστήματος αποκτούν γενικά εκθετικά αυξανόμενο πλήθος με το μέγεθός του. Παρ' όλα αυτά εμφανίζονται κατά καιρούς τέτοια σενάρια, επειδή κάποια ατελής γνώση είναι καλύτερη από τη μηδενική γνώση.

Ένα ζήτημα που προηγείται της περιγραφής των επιθέσεων είναι ο καθορισμός μιας σχετικής ορολογίας. Ένα γενικό παράδειγμα ορολογίας περί την ασφάλεια είναι η πρόταση του NIST[2], που χρησιμοποιείται στα έγγραφα που παράγονται από αυτόν

[2] `https://csrc.nist.gov/glossary`

τον οργανισμό. Μια άλλη συλλογή όρων τηρείται από τον εκπαιδευτικό οργανισμό NICCS (National Initiative for Cybersecurity Careers and Studies).[3] Μια περιορισμένη πρόταση ορολογίας έχει ξεκινήσει από την ΕυρωπαΪκή *ENISA*[4]. Μια πιο εξειδικευμένη σε περιστατικά ασφάλειας πρακτική ορολογία είναι αυτή που έχει υιοθετηθεί από την *Verizon Communications* και ονομάζεται *VERIS* (Vocabulary for Event Recording and Incident Sharing).[5].

Η ανίχνευση μιας επίθεσης βασίζεται στη συλλογή και ανάλυση δεδομένων. Όσο περισσότερα είναι τα σημεία συλλογής δεδομένων, τόσο πιο πιθανό είναι να ανιχνευθεί μια επίθεση, αλλά πιο εκτεταμένη συλλογή χρειάζεται περισσότερους πόρους επεξεργασίας και αποθήκευσης, καθώς και αντίστοιχη ικανότητα ανάλυσης, δηλαδή κατάλληλους αλγόριθμους. Η συνηθισμένη κατάταξη συστημάτων ανίχνευσης ως προς τα σημεία συλλογής είναι αν πρόκειται για τοπική συλλογή πάνω σε μια μηχανή και γύρω από αυτήν (σημεία επαφής της με το δίκτυο) ή αν εμπλέκεται και το ευρύτερο δικτυακό περιβάλλον.

8.2 Συστήματα ανίχνευσης

Ανίχνευση στον υπολογιστή - HIDS

Η προφανής μέθοδος ανίχνευσης μιας εισβολής σε έναν υπολογιστή είναι να παρακολουθείται η λειτουργία του για ύποπτη συμπεριφορά τόσο εσωτερικά, όσο και στις διεπαφές του, περιλαμβανομένων των δικτυακών διεπαφών (και φυσικά της διεπαφής ανθρώπου μηχανής). Ένα σύστημα που υλοποιεί τέτοιες λειτουργίες είναι γνωστό ως *HIDS* (*Host-based Intrusion Detection System*). Πρόκειται για ένα πράκτορα (agent) που καταγράφει και αναλύει οτιδήποτε μπορεί να οδηγήσει σε σχετική διάγνωση, δηλαδή κλήσεις του συστήματος, αρχεία καταγραφής ενεργειών διαφόρων εφαρμογών (application logs), μεταβολές αρχείων του συστήματος με έμφαση σε αρχεία που είναι πιο κρίσιμα για τη λειτουργία του και την ασφάλειά του, π.χ. το αρχείο που περιέχει ονόματα χρηστών και τα αντίστοιχα συνθηματικά (ή τις τιμές κατακερματισμού τους). Τέτοιες αρμοδιότητες ενσωματώνονται συνήθως σε ένα σύστημα anti-virus.

Ίχνη και ψηφιακή εγκληματολογία

Μια ολοκληρωμένη, άρα επιτυχημένη, επίθεση που έχει εξαντλήσει τον κύκλο ζωής της αφήνει μια σειρά ιχνών σε μια μηχανή. Στα ίχνη περιλαμβάνονται οι κύριες αλλοιώσεις που αποτελούν τον σκοπό της επίθεσης (π.χ. μεταβολή, διαγραφή ή κρυπτογράφηση αρχείων) καθώς και δευτερογενή φαινόμενα, π.χ. εγγραφές σε logs, που κανονικά ο εισβολέας θα προτιμούσε να είχαν αποφευχθεί και είναι πιθανό να προσπαθήσει απαλείψει. Η περιοχή που εξετάζει τα ίχνη του εισβολέα (με σκοπό να του επιβληθούν έννομες συνέπειες) λέγεται *ψηφιακή εγκληματολογία* (*digital forensics*)[Sam12]. Μια αρχή από την κλασσική εγκληματολογία, που έχει εφαρμογή και στην ψηφιακή, είναι ότι ο δράστης πάντοτε κάτι παίρνει από τη σκηνή του εγκλήματος και κάτι αφήνει σ' αυτήν. Στην ψηφιακή εκδοχή ο δράστης διευκολύνεται να πάρει κάτι χωρίς να διαπιστωθεί η έλλειψή του επειδή είναι εύκολη η αναπαραγωγή ψηφιακών αντικειμένων. Ωστόσο θα βρει άλλες δυσκολίες, όπως εξηγούμε στη συνέχεια.

Τα δεδομένα που είναι αποθηκευμένα σε μια μηχανή χαρακτηρίζονται γενικά από πλεονασμό και αλληλεξάρτηση, γεγονός που κάνει παρατηρήσιμη την αλλοίωση απο-

[3] `https://niccs.cisa.gov/about-niccs/cybersecurity-glossary`
[4] `https://www.enisa.europa.eu/topics/csirts-in-europe/glossary`
[5] `http://veriscommunity.net/`

μονωμένων στοιχείων. Ένα παράδειγμα θα εξηγήσει αυτό το φαινόμενο: Έστω ότι ένας εισβολέας μεταβάλλει την κατάληξη ενός αρχείου κειμένου Word από doc σε xls. Σε απόπειρα κάποιου να ανοίξει αυτό το αρχείο με ένα συμβατό επεξεργαστή κειμένου θα παραχθεί ένα μήνυμα σφάλματος. Ο υποψιασμένος χρήστης μπορεί να δοκιμάσει διάφορες πιθανές καταλήξεις και να διαπιστώσει επιτυχημένο άνοιγμα όταν φτάσει στην σωστή κατάληξη. Ωστόσο ένα πιο εξειδικευμένο εργαλείο θα διαπιστώσει αμέσως τον σωστό τύπο αρχείου διαβάζοντας την αντίστοιχη εσωτερική επικεφαλίδα που το χαρακτηρίζει ως Word. Ένα τέτοιο εργαλείο μπορεί να σαρώσει αρχεία και να εντοπίσει ασυμβατότητες μεταξύ εσωτερικής επικεφαλίδας και κατάληξης. Ένα ακόμη πιο εξελιγμένο εργαλείο, π.χ. ένα σύστημα anti-virus, μπορεί να δημιουργήσει κωδικούς κατακερματισμού για ενδιαφέροντα αρχεία, να τους αποθηκεύσει ασφαλώς και στη συνέχεια να τους επανυπολογίζει περιοδικά προκειμένου να διαπιστωθούν αλλοιώσεις.

Μια άλλη ιδιότητα ενός υπολογιστικού συστήματος που είναι πιθανό να χρησιμοποιήσει ένας ερευνητής για να βρει ίχνη είναι ότι η πραγματική διαγραφή δεδομένων δεν συμβαίνει αμέσως στο φυσικό μέσο αποθήκευσης. Π.χ. η διαγραφή ενός log file δεν καταστρέφει αμέσως τις μαγνητικές εγγραφές ενός σκληρού δίσκου που αντιστοιχούν στο διαγραμμένο αρχείο. Αυτό θα γίνει αργότερα, όταν κατά τύχη χρειαστεί αυτό το μέρος του δίσκου για μια νέα εγγραφή.

Με δυο λόγια μια οποιαδήποτε επίθεση ή εισβολή, είτε αφήνει ίχνη σκοπίμως (όπως μια επίθεση για λύτρα αφήνει κρυπτογραφημένα αρχεία) είτε από αδεξιότητα. Συχνά βασίζεται στην εγκατάσταση πρόσθετου λογισμικού κατάλληλης μορφής, που μπορεί να είναι π.χ. σκουλήκι, ιός, rootkit κ.λπ., και το οποίο μπορεί πιθανώς να ανιχνευθεί ως pattern. Ακόμη, κάνει μια σειρά από ενέργειες, π.χ. αποκτάει πρόσβαση διαχειριστή σε κρίσιμες περιοχές της μηχανής. Γενικά μια επίθεση αποτελείται από μια σειρά ενεργειών και καταλήγει σε μια σειρά αλλοιώσεων που, εφόσον καταγραφούν, αναλυθούν και ενσωματωθούν στις δυνατότητες ενός αμυντικού συστήματος, μπορούν την επόμενη φορά που θα γίνει παρόμοια επίθεση να βοηθήσουν στην πρόληψη ή ανίχνευση ή αντιμετώπισή της.

Δημιουργείται με αυτόν τον τρόπο ένας ατέρμων κύκλος προσβολής, ανάλυσης και αναβάθμισης της άμυνας, αναβάθμισης της επίθεσης (για να αποφευχθούν τα νέα μέτρα άμυνας) κ.ο.κ.

Λειτουργία του IDS

Η βασική προσέγγιση ενός IDS μπορεί να βασίζεται στη σάρωση, δηλαδή στη διατήρηση μιας βάσης δεδομένων με χαρακτηριστικές ιδιότητες και αντικείμενα του υπολογιστή, ως επί πλείστον αρχεία. Για κάθε αντικείμενο δημιουργείται και αποθηκεύεται ασφαλώς ένας κωδικός επαλήθευσης και κατά καιρούς γίνεται επανυπολογισμός του κωδικού και σύγκριση με την προηγούμενη τιμή του.

Για παράδειγμα το *OSSEC* (*Open Source HIDS SECurity*) είναι ένα HIDS ανοιχτού κώδικα που αναλύει log files, ελέγχει την ακεραιότητα αρχείων, ανιχνεύει rootkits, συντηρεί μια βάση δεδομένων με γνωστούς ιούς, σκουλήκια, δούρειους ίππους κ.λπ. και είναι διαθέσιμο για όλα σχεδόν τα λειτουργικά συστήματα. Ο διαχειριστής μπορεί να επιλέξει ποιους συναγερμούς θέλει να βλέπει και μπορεί να ορίσει δικούς του κανόνες καθώς και scripts ενεργειών που εκτελούνται όταν γίνεται ανίχνευση ύποπτων συμπτωμάτων.

Επίσης το HIDS στα πλαίσια της παρακολούθησης των δικτυακών διεπαφών του υπολογιστή μπορεί να χρησιμοποιεί μεθόδους ανίχνευσης ανωμαλιών της κίνησης (βλ. ενότητα 8.3. Περαιτέρω μπορεί να παρακολουθεί την ακολουθία εκτέλεσης ενεργειών

που εντάσσονται σε ένα πρωτόκολλο για να διαπιστωθεί κατά πόσο η ακολουθία αυτή αποκλίνει από την κανονική.

Υπάρχει ένα ανοιχτό θέμα ορολογίας και διάκρισης μεταξύ ενός HIDS και ενός anti-virus. Το δεύτερο θεωρείται πως είναι εργαλείο πρόληψης της επίθεσης (HIPS - host-based intrusion prevention system) με έμφαση στη χρήση μεθόδων υπογραφής και ευριστικών μεθόδων. Το πρώτο θεωρείται ότι έχει έμφαση στην παρακολούθηση του συστήματος και τη δημιουργία συναγερμών. Ωστόσο τίποτε δεν εμποδίζει οποιοδήποτε από τα δύο να είναι ενισχυμένο με τις τεχνικές του άλλου. Με άλλα λόγια το πιθανότερο είναι κάποιος να χρησιμοποιεί ένα συνδυασμένο σύστημα.

Ανίχνευση στο δίκτυο - NIDS

Ένα *δικτυακό σύστημα ανίχνευσης εισβολής* (*Network Intrusion Detection System - NIDS*) συγκεντρώνει πληροφορίες από διάφορα σημεία ενός δικτύου και προσπαθεί να ανιχνεύσει επιθέσεις. Σε σύγκριση με ένα HIDS, που μαζεύει πληροφορία μόνο στα όρια μιας μηχανής, ένα NIDS «βλέπει» μια πολύ πιο ολοκληρωμένη εικόνα της κίνησης στο δίκτυο. Ωστόσο οι τύποι των πληροφοριών που μαζεύει ένα NIDS είναι πολύ πιο περιορισμένοι από αυτές που βλέπει ένα HIDS, το οποίο έχει τη δυνατότητα να βλέπει εσωτερικά την εξέλιξη μιας μηχανής πέραν των γεγονότων που συμβαίνουν στις εισόδους και εξόδους της.

Ένα NIDS βασίζεται στην πληροφορία που μεταφέρεται από τη δικτυακή κίνηση. Η κίνηση γενικά ενσωματώνεται σε πακέτα. Το NIDS μπορεί να βασίζεται σε μια επιφανειακή θεώρηση των πακέτων, μετρώντας μόνο στατιστικά μεγέθη σχετικά με τον αριθμό τους και τις χρονικές τους αποστάσεις, ή να προχωράει σε επιθεώρηση των επικεφαλίδων των πακέτων ως επιλεγμένο βάθος (deep packet inspection - DPI), δηλαδή να διαβάζει τις επικεφαλίδες που ανήκουν σε συγκεκριμένα στρώματα πρωτοκόλλων (π.χ. το TCP), ή και σε ακόμη μεγαλύτερο βάθος διαβάζοντας το φορτίο τους, δηλαδή φτάνοντας σε γνώση της εφαρμογής και του περιεχομένου. Είναι αυτονόητο ότι μεγαλύτερο βάθος επιθεώρησης εξασφαλίζει ισχυρότερες δυνατότητες ανάλυσης και διάγνωσης, αλλά καταναλώνει περισσότερους πόρους.

Εκτός από τα τεχνικά προβλήματα και τα θέματα επιδόσεων, η βαθύτερη επιθεώρηση θέτει ζητήματα ιδιωτικότητας και νομιμότητας. Χρησιμοποιείται π.χ. ως εργαλείο λογοκρισίας εντοπίζοντας και διαγράφοντας πακέτα με συγκεκριμένες λέξεις.

Μια σχετικά ελαφράς μορφής επιθεώρηση σε βάθος είναι αυτή που υποδεικνύεται από την RFC 2827 (Network Ingress Filtering), δηλαδή ο ISP να ελέγχει αν τα πακέτα που στέλνει μια μηχανή σε ένα σημείο εισόδου του δικτύου του ISP φέρουν όντως την δικτυακή διεύθυνση πηγής αυτής της μηχανής. Περαιτέρω σημεία ελέγχου περιγράφονται από την RFC 6959 (Source Address Validation Improvement (SAVI) Threat Scope). Όπως δείχνουν αυτές οι δύο συστάσεις, τα συμπεράσματα που μπορούν να εξαχθούν από την εξέταση πακέτων εξαρτώνται από τα σημεία εξέτασης και μπορούν να ανιχνεύσουν συγκεκριμένες επιθέσεις, ή και να τις παρεμποδίσουν εφόσον υπάρχει δυνατότητα εφαρμογής μέτρων.

Οι διαγνώσεις που μπορεί να κάνει ένα NIDS υπάγονται και αυτές στην «κούρσα των εξοπλισμών» μεταξύ επίθεσης και άμυνας και στην εκατέρωθεν προσαρμογή των μεθόδων. Ένα HIDS ενός server που βλέπει μια σειρά πακέτων να φτάνουν σ' αυτόν από την ίδια πηγή με μεγάλη συχνότητα θα ανιχνεύσει πιθανώς μια επίθεση DDoS. Όταν ο επιτιθέμενος αναβαθμίσει την επίθεση χρησιμοποιώντας αλλοιωμένες διευθύνσεις, ίσως ένα NIDS που επιβλέπει το υποδίκτυο όπου ανήκει ο server να καταφέρει να διαγνώσει ότι μια σειρά από πακέτα διαφορετικής προέλευσης ακολουθούν παρ' όλα αυτά την ίδια διαδρομή μέσα στο υποδίκτυο και προχωρούν προς τον ίδιο

server στην ίδια θύρα. Τότε ο επιτιθέμενος μπορεί να αναβαθμίσει περαιτέρω την επίθεση προδιαγράφοντας διαφορετικούς κόμβους από τους οποίους πρέπει να περάσουν ομάδες των πακέτων που εμπλέκονται στην επίθεση, προκειμένου να εμφανισθούν διαφορετικά ρεύματα στο υποδίκτυο. Τότε η άμυνα μπορεί να διαγνώσει ότι τα πακέτα φέρουν ρητές οδηγίες δρομολόγησης και να τα κόψει. Μετά ο επιτιθέμενος μπορεί να χρησιμοποιήσει ένα botnet, ώστε να διαφοροποιήσει περαιτέρω την προέλευση των πακέτων και επιπροσθέτως να γλυτώσει από μέτρα ελέγχου προέλευσης (των τύπων των RFC 2827 και RFC 6959). Και τότε η άμυνα μπορεί να ελέγξει τον χρονισμό αυτών των πακέτων σε συνδυασμό με άλλες παραμέτρους.

8.3 Μέθοδοι ανίχνευσης επιθέσεων κίνησης

Μια λογική προσέγγιση στο πρόβλημα της ανίχνευσης μιας επίθεσης βασίζεται σε προηγούμενη γνώση επιθέσεων με συγκεκριμένα χαρακτηριστικά, που μπορεί να είναι συγκεκριμένες ακολουθίες από bits ή εντολές (signature-based techniques). Μια άλλη προσέγγιση μπορεί να βασίζεται σε γνωστή επιβλαβή συμπεριφορά (misuse-based techniques). Οι μέθοδοι αυτές προϋποθέτουν, όπως γενικότερα η ανίχνευση συγκεκριμένου κακόβουλου λογισμικού (malware), να έχει προηγουμένως διαπιστωθεί και αναλυθεί μια τουλάχιστον επίθεση του συγκεκριμένου τύπου. Εάν η επίθεση περιλαμβάνει στα αποτελέσματά της παρατηρήσιμες βλάβες, κάτι τέτοιο μπορεί να είναι εφικτό. Εάν όμως το αποτέλεσμα ήταν λιγότερο παρατηρήσιμο, όπως στην περίπτωση της υπεξαίρεσης στοιχείων ή και της αλλοίωσης κατά τρόπο που δεν μένουν ίχνη, είναι πιθανό μια τέτοια επίθεση να μην έχει καταγραφεί. Επί πλέον όταν μια επίθεση ενός τύπου προστίθεται στις γνωστές ικανότητες των συστημάτων άμυνας, οι επιτιθέμενοι είναι πιθανό να βελτιώσουν αντίστοιχα την μέθοδο επίθεσης ώστε να μη γίνει αντιληπτή την επόμενη φορά. Στη συνέχεια όμως θα δούμε ορισμένες μεθόδους ανίχνευσης που δεν βασίζονται στην προηγούμενη καλή γνώση της επίθεσης, αλλά μάλλον στην απόκλιση από την κανονική συμπεριφορά.

Μέθοδοι ανίχνευσης ανωμαλιών

Υπάρχει μια κατηγορία μεθόδων ανίχνευσης που στοχεύει στον εντοπισμό ανωμαλιών στην κίνηση. Γενικότερα σε ένα σύνολο δεδομένων μπορεί να υπάρχουν δεδομένα που αποκλίνουν σημαντικά από τα συνηθισμένα, γεγονός που μπορεί να οφείλεται είτε σε ακραία φαινόμενα είτε και σε σφάλματα κατά τη συλλογή των δεδομένων. Τέτοια δεδομένα ονομάζονται *έκτροπα* (outliers) [Agg17]. Ο διαχειριστής ενός συστήματος (ή ο σχεδιαστής ενός συστήματος διαχείρισης) θα κληθεί να αποφασίσει πώς θα διαχειριστεί με τον καταλληλότερο τρόπο τέτοια δεδομένα. Σε ορισμένες περιπτώσεις οι έκτροπες τιμές είναι βέβαιο πως οφείλονται σε σφάλματα της διαδικασίας μέτρησης και απλώς διαγράφονται, όπως π.χ. μια καταγραφή 45^0C βαθμών ανάμεσα σε θερμοκρασίες σώματος που καταγράφει ένα ιατρικό θερμόμετρο. Σε άλλες περιπτώσεις οι έκτροπες τιμές αντιστοιχούν σε σπάνια ακραία γεγονότα, που όμως όταν εμφανισθούν χρειάζονται ιδιαίτερη αντιμετώπιση.

Για την ανίχνευση εκτρόπων δεδομένων έχουν δοκιμαστεί μέθοδοι προερχόμενες από διάφορες περιοχές όπως η επεξεργασία σήματος, η στατιστική κ.α.

Ανάλυση ακραίων τιμών

Η ανάλυση ακραίων τιμών (extreme value analysis - EVA) είναι μια περιοχή της στατιστικής που ασχολείται με την υπερβολική απόκλιση τιμών από την μέση τιμή.

Συνήθως υπολογίζεται ένα μέτρο της απόκλισης για κάθε σημείο των δεδομένων. Η σημασία της απόκλισης εξαρτάται από την κατανομή από την οποία γεννώνται τα δεδομένα (την *υποκείμενη κατανομή*) και ιδιαίτερα από την στατιστική ουρά αυτής της κατανομής.

Ωστόσο είναι διαφορετικό να ψάχνει κανείς για ακραίες τιμές και διαφορετικό να ψάχνει για έκτροπες τιμές. Αυτό μπορεί να γίνει πιο σαφές από το εξής παράδειγμα: Δίνεται το σύνολο δεδομένων $\{1, 2, 3, 2, 49, 98, 100, 99, 99\}$. Η μέση τιμή είναι 50.3, άρα οι ακραίες τιμές, δηλαδή οι πιο απομακρυσμένες τιμές από τη μέση τιμή, είναι όλες εκτός της τιμής 49. Ωστόσο κάποιος που παρατηρεί αυτές τις τιμές θα είχε πιθανώς την άποψη ότι οι κανονικές τιμές είναι όλες εκτός της 49, επειδή το φαινόμενο παίρνει δυο κανονικές ομάδες τιμών, μια ομάδα κοντά στο 1 και μια κοντά στο 100.Ένας τρόπος να περιγραφεί το φαινόμενο θα ήταν μέσω μιας κατανομής, η οποία θα είχε δύο κορυφές, και στη συνέχεια θα υπολογίζαμε ως *μέτρο εκτροπής* την πιθανότητα να εμφανισθεί κάθε μια από τις παραπάνω τιμές. Μια τέτοια κατανομή θα έδινε μικρή πιθανότητα για την τιμή 49, επομένως αυτή η τιμή θα εθεωρείτο έκτροπη.

Η ανάλυση αυτού του τύπου περιλαμβάνεται σε αλγόριθμους εντοπισμού ακραίων τιμών στο βήμα της ποσοτικοποίησης των αποτελεσμάτων [Agg17], εκεί δηλαδή όπου χρειάζεται να γίνει μια εκτίμηση για το πόσο ακραίο είναι το κάθε σημείο ενός συνόλου δεδομένων.

Πιθανοτικά μοντέλα

Δημιουργείται ένα κατάλληλο πιθανοτικό μοντέλο, π.χ. μια κατανομή, του οποίου οι παράμετροι πρέπει να εκτιμηθούν (με κάποια από τις γνωστές μεθόδους), ώστε στη συνέχεια σημεία που δεν ταιριάζουν με το μοντέλο να χαρακτηρισθούν έκτροπα.

Γραμμικά μοντέλα

Στην περίπτωση αυτή το αρχικό σύνολο από σημεία πολλών διαστάσεων προβάλλεται σε υποχώρους λιγότερων διαστάσεων με κριτήριο την συσχέτιση. Για παράδειγμα, αν το αρχικό σύνολο περιλαμβάνει σημεία σε δύο διαστάσεις, δηλαδή είναι της μορφής $\{(x_i, y_i), i = 1, 2, \ldots, N\}$, ψάχνουμε για την πλησιέστερη στα σημεία γραμμή $y = ax + b$. Δηλαδή ψάχνουμε για μια γραμμή τέτοια, ώστε

$$y_i = ax_i + b + \epsilon_i$$

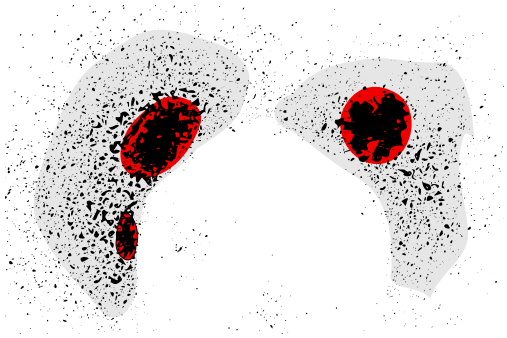
όπου ϵ_i είναι το «σφάλμα» του σημείου i επειδή το $y_i = ax_i + b$ δεν μπορεί να πέσει ακριβώς πάνω στη γραμμή, και υπολογίζουμε τα a, b ώστε να ελαχιστοποιείται το $\sum_{i=1}^{N} \epsilon_i^2$. Στη συνέχεια μπορούμε να βάλουμε ένα κριτήριο εκτροπής που θα βασίζεται στην απόσταση ενός σημείου από την γραμμή. Ο περιορισμός των διαστάσεων μπορεί να γίνει και μέσω της μέθοδου PCA, βλ. σελ. 149.

Φασματικά μοντέλα

Εδώ εφαρμόζονται ιδέες από την περιοχή της επεξεργασίας σήματος στο πεδίο της συχνότητας. Ο σκοπός είναι η ανίχνευση μη κανονικής συμπεριφοράς μιας ακολουθίας γεγονότων στο πεδίο του χρόνου.

Μοντέλα βασισμένα στην εγγύτητα

Σε τέτοιου είδους μοντέλα θεωρείται ότι ένα σημείο των δεδομένων είναι κοντά σε ένα άλλο με κριτήρια απόστασης ή ομοιότητας, εντοπίζονται δηλαδή ομάδες κο-



Σχήμα 8.1: Εντοπισμός εκτρόπων σημείων με τη μέθοδο της πυκνότητας.

ντινών σημείων, ενώ όσα σημεία δεν περιλαμβάνονται στις ομάδες μπορούν να θεωρηθούν έκτροπα.

Ένα κριτήριο πυκνότητας γενικά χαρακτηρίζει τα σημεία του χώρου, π.χ. το κριτήριο πυκνότητας για το σημείο (x, y) μπορεί να είναι το πλήθος $n(x, y)$ των σημείων σε κάθε κύκλο με κέντρο το σημείο (x, y) και ακτίνα r. Μια περιοχή περιλαμβάνει τα σημεία του επιπέδου με την ιδιότητα $n(x, y) > n_0$, για δεδομένο n_0 (ή εναλλακτικά με την ιδιότητα $n_1 > n(x, y) > n_0$ αν θέλουμε να δημιουργήσουμε περιοχές που δεν περιέχει η μια την άλλη). Στο παράδειγμα του Σχ. 8.1 χρησιμοποιείται ένα τέτοιο κριτήριο πυκνότητας για να οριοθετηθούν δύο περιοχές πυκνότητας, η περιοχή υψηλής πυκνότητας (κόκκινη) και η περιοχή χαμηλής πυκνότητας (ανοιχτό γκρι). Όσα σημεία πέφτουν έξω από την γκρι περιοχή (άρα έξω από αμφότερες τις περιοχές) μπορούν να θεωρηθούν έκτροπα.

Ένα κριτήριο εγγύτητας μπορεί να χρησιμοποιηθεί για να χαρακτηρίσει απ' ευθείας τα σημεία του συνόλου δεδομένων. Για παράδειγμα, μπορούμε να δημιουργήσουμε ομάδες σημείων, όπου κάθε σημείο της ομάδας απέχει λιγότερο από μια απόσταση d από ένα άλλο σημείο, χρησιμοποιώντας έναν αλγόριθμο που προσθέτει σε μια ομάδα ένα σημείο εφόσον αυτό απέχει λιγότερο από d από τουλάχιστον ένα σημείο μιας ήδη υφιστάμενης ομάδας. Στο τέλος για δεδομένο μικρό n θα υπάρξουν ομάδες με n το πολύ σημεία, που μπορούμε να θεωρήσουμε έκτροπα. Σε παράδειγμα με (μονοδιάστατα) σημεία

$$\{1, 2, 3, 2, 49, 51, 98, 100, 99, 99\}$$

αν θέσουμε $d = 10$ θα δημιουργηθούν οι ομάδες

$$\{1, 2, 3, 2\}, \quad \{49, 51\}, \quad \{98, 100, 99, 99\}.$$

Αν θέσουμε $n = 2$, έχουμε ως έκτροπα τα σημεία $\{49, 51\}$.

Σε μια κάπως διαφορετική προσέγγιση μπορούμε να υπολογίσουμε για κάθε σημείο x_i την απόσταση d_i του n-οστού εγγύτερου γείτονα. Στην ίδια ως άνω ακολουθία $x_1 = 1, x_2 = 2, x_3 = 3, x_4 = 2, x_5 = 49, x_6 = 51, x_7 = 98, x_8 = 100, x_9 = 99, x_{10} = 99$ για $n = 3$ βρίσκουμε $d_1 = 2, d_2 = 1, d_3 = 2, d_4 = 1, d_5 = 47, d_6 = 48, d_7 = 2, d_8 = 2, d_9 = 1, d_{10} = 1$. Αν ως έκτροπα θεωρήσουμε τα σημεία με $d_i > 10$ προκύπτουν τα σημεία x_5 και x_6.

Μοντέλα προερχόμενα από τη θεωρία πληροφορίας

Κεντρική στη θεωρία πληροφορίας είναι η έννοια της *εντροπίας*. Κατά Shannon η εντροπία είναι μονόμετρο μέγεθος που υπολογίζεται ως συνάρτηση της πιθανοτικής κατανομής ενός συνόλου δεδομένων. Στη συνέχεια η εντροπία αποδίδει το μέσο μήκος ενός βελτιστοποιημένου κώδικα, του οποίου οι λέξεις μπορούν να χρησιμοποιηθούν για την συμπυκνωμένη αποθήκευση ή μετάδοση των δεδομένων. Κατά Kolmogorov ορίζεται ένα μέτρο *πολυπλοκότητας*, που είναι το μήκος του πιο σύντομου (σε αριθμό συμβόλων) προγράμματος υπολογιστή που μπορεί να δημιουργήσει το σύνολο δεδομένων (και αποτελεί γενίκευση της εντροπίας κατά Shannon). Αμφότεροι οι ορισμοί μπορούν να χρησιμοποιηθούν για την ανίχνευση ανωμαλιών. Τον ορισμό Shannon θα τον ξαναδούμε αργότερα στην εφαρμογή του ανωμαλιών κίνησης σε δίκτυα. Στο σημείο αυτό θα δώσουμε ένα παράδειγμα εφαρμογής της πολυπλοκότητας Kolmogorov.

Έστω η ακολουθία $(0101\ldots)$, αποτελούμενη από 1000 bits με σταθερή εναλλαγή 0 και 1 (ή 1002 σύμβολα αν περιλάβουμε τις παρενθέσεις). Η ακολουθία θα μπορούσε να συμπυκνωθεί σε $(01)^{500}$, δηλαδή σε 5-7 σύμβολα, ανάλογα με τον τρόπο γραφής, ενώ αντίστοιχου μικρού μήκους θα ήταν ο σχετικός κώδικας. Αν το bit με αριθμό 500, που είναι 1, αλλοιωθεί σε 0, η νέα ακολουθία θα μπορούσε να γραφτεί ως $(01)^{249}(00)(01)^{250}$ ή ως κάτι που θα τροποποιούσε την παλιά ακολουθία στο συγκεκριμένο bit. Η «συνταγή» για τη νέα ακολουθία, όπως ακριβώς κι αν αποτυπωθεί, είναι αρκετά μακρύτερη από την συνταγή για την ακολουθία χωρίς την αλλοίωση, και το μήκος της εκφράζει την πολυπλοκότητα Kolmogorov. Και πάλι σημεία με εντροπία ή πολυπλοκότητα πέραν κάποιων ορίων μπορούν να θεωρηθούν έκτροπα.

Ανίχνευση ανωμαλιών σε δίκτυα

Στην περίπτωση της κίνησης σε δίκτυα η καταγραφή ακραίων τιμών μπορεί να οφείλεται σε διαφορετικούς λόγους, ανάμεσα στους οποίους είναι αυξημένη προσφερόμενη κίνηση,[6] τυχαίες βλάβες και περιστατικά που οφείλονται σε επιθέσεις. Βασικό μέρος της διαχείρισης ενός δικτύου επικοινωνιών είναι να εντοπίζονται τέτοια περιστατικά και να γίνονται οι κατάλληλοι χειρισμοί για την αντιμετώπισή τους. Το σύστημα διαχείρισης αντλεί δεδομένα από διαφορετικές πηγές και με χρήση διαφορετικών εργαλείων (π.χ. ping και traceroute, φίλτρα πακέτων που είναι εγκατεστημένα σε διάφορα σημεία του δικτύου, συλλογή δεδομένων από δρομολογητές και άλλες συσκευές δικτύου). Ένα σύστημα διαχείρισης μπορεί να περιλαμβάνει ένα αυτόματο σύστημα που αντιμετωπίζει γεγονότα που είτε έχουν γνωστές και ενσωματωμένες στο σύστημα λύσεις, ιδιαίτερα για προβλήματα με πολύ μικρές χρονικές κλίμακες. Όσα όμως προβλήματα δεν κατορθώνει να επιλύσει το αυτόματο σύστημα (καθώς και μια απογραφή προβλημάτων που έχουν επιλυθεί) μεταφέρονται στην ευθύνη ανθρώπων διαχειριστών, μέσω μιας κατάλληλης διεπαφής, που περιλαμβάνει καταγραφές γεγονότων, αναπαράστασή τους με τρόπο κατανοητό από τους διαχειριστές, συναγερμούς, και εργαλεία για την εκτέλεση χειρισμών από τους διαχειριστές.

Η ανίχνευση ανωμαλιών μπορεί να γίνει με διαφορετικές μεθόδους [TJ03]:

Χρήση κανόνων: Ο εντοπισμός ανωμαλιών γίνεται από ένα έμπειρο σύστημα (expert system) που προσπαθεί να εντοπίσει την αιτία μιας ανωμαλίας βάσει κανόνων, που οφείλονται στην υπάρχουσα εμπειρία. Οι κανόνες εφαρμόζονται σε μια

[6] Για παράδειγμα, ένας σεισμός ή άλλη φυσική καταστροφή δημιουργεί κατά κανόνα υπερβολική κίνηση στο σταθερό και κινητό τηλεφωνικό δίκτυο, σε βαθμό που τα δίκτυα αυτά σε ορισμένες περιπτώσεις έχουν φτάσει στην πλήρη κατάρρευση.

σειρά παρατηρούμενων μεγεθών και συμπτωμάτων, όπως π.χ. υπερβολικά φορτία, καθυστερήσεις, πλήθος ανοιχτών συνδέσεων TCP κ.λπ.

Μηχανές πεπερασμένων καταστάσεων που μπαίνουν σε συγκεκριμένες καταστάσεις μετά από μια ακολουθία περιστατικών συναγερμού (alarms) και προσπαθούν να καταλήξουν σε μια ερμηνεία και αντιμετώπιση του προβλήματος.

Αντιστοίχιση με πρότυπα (pattern matching): Δημιουργείται ένα προφίλ κίνησης με χρήση μηχανικής μάθησης, χρησιμοποιώντας μια σειρά από παραμέτρους, όπως φορτίο στις ζεύξεις, πιθανότητα απώλειας πακέτων κ.λπ.. Οι αποδεκτές τιμές των παραμέτρων μπορεί να διαφοροποιούνται με τη μέρα και την ώρα. Μόλις τα μεγέθη ξεφύγουν από τις καθορισμένες τιμές θεωρείται ότι έχει εκδηλωθεί ανωμαλία [FSM93; Pap+00].

Στατιστική ανάλυση: Παρακολουθούνται συγκεκριμένα στατιστικά μεγέθη και στη συνέχεια αναλύονται με διάφορες μεθόδους, όπως εξηγείται πιο κάτω.

Επίσης έχουν εξετασθεί τεχνικές από την περιοχή των μεγάλων δεδομένων [FMF14].

Το βασικό χαρακτηριστικό που εξετάζεται προκειμένου να διαγνωσθεί μια επίθεση κίνησης βεβαίως είναι πρώτα πρώτα ο όγκος της κίνησης, αλλά έχουν αξιοποιηθεί και άλλα χαρακτηριστικά, ώστε να γίνεται δυνατή η ανίχνευση με πιο εκλεπτυσμένο τρόπο. Μια από τις σχετικά γενικές τέτοιες μεθόδους, η *ανάλυση οπισθοσκέδασης*, έχει περιγραφεί στην ενότητα 6.5 και βασίζεται στην ανίχνευση κίνησης ανακλώμενης προς ανύπαρκτες διευθύνσεις δικτύου.

Ένα ρεύμα κίνησης που περνάει από (ή φτάνει σε) μια συνιστώσα ενός δικτύου (π.χ. μια ζεύξη ή ένα δρομολογητή ή μια συγκεκριμένη είσοδο ή έξοδο του δρομολογητή ή ενός server) μπορεί να περιγραφεί στην εξαντλητική της λεπτομέρεια από τους χρόνους άφιξης ή διέλευσης ή αναχώρησης των πακέτων και το πλήρες τους περιεχόμενο (επικεφαλίδες και φορτίο). Ωστόσο ένα σύστημα ανίχνευσης επιθέσεων πρέπει να πάρει μια δυαδική απόφαση (ναι/όχι) αν σε μια δεδομένη στιγμή στο ρεύμα περιλαμβάνονται πακέτα που ανήκουν σε μια επίθεση. Μια τέτοια απόφαση σε πραγματικό χρόνο προϋποθέτει κατανάλωση υπολογιστικών πόρων και η προσπάθεια αυτή δεν μπορεί να βασίζεται στην πλήρη πληροφορία. Κατά συνέπεια όλες οι γενικές μέθοδοι ανίχνευσης βασίζονται στην αφαίρεση, δηλαδή στην επιλογή συγκεκριμένων μόνο παραμέτρων και χαρακτηριστικών που θα τεθούν υπό παρακολούθηση, ενώ άλλα διαθέσιμα χαρακτηριστικά αμελούνται. Ακολουθεί η περαιτέρω επεξεργασία των επιλεγμένων δεδομένων, με σκοπό να λαμβάνεται στο τέλος μια απόφαση που θα βασίζεται σε λίγες παραμέτρους.

Ακόμη κι αν υπήρχαν τα εργαλεία για την επεξεργασία των δεδομένων, παραμένει αμφίβολη η διαθεσιμότητα τέτοιων πληροφοριών που θα συλλέγονται σε κατάλληλα σημεία του δικτύου. Οι πάροχοι δικτύου (ISPs) εν γένει δεν ενδιαφέρονται να συλλέξουν τέτοιες πληροφορίες στην λεπτομέρεια και στην ολότητά τους και καταγράφουν μόνο μέσες τιμές κίνησης, το πολύ να κάνουν καταμέτρηση πακέτων στα άκρα του δικτύου τους.

Ως προς τις παραμέτρους που μπορούν να αξιοποιηθούν σε μια τέτοια ανάλυση, ένας ενδεικτικός κατάλογος έχει ως εξής:

1. Χρόνοι άφιξης πακέτων σε συγκεκριμένα σημεία ενδιαφέροντος ή διέλευσης από κανάλια ή πλήθος πακέτων σε επαναλαμβανόμενα παράθυρα χρόνου. Τέτοιες παράμετροι είναι εξωτερικές του πακέτου, δηλαδή δεν απαιτούν καν να γίνει ανάγνωση του επικεφαλίδων πακέτου.

2. Καταγραφή διευθύνσεων προέλευσης πακέτου ή προορισμού ή αμφοτέρων σε παράθυρο χρόνου. Επίσης καταγραφή θυρών (ports).

3. Καταγραφή περαιτέρω πεδίων της επικεφαλίδας πακέτου (πέραν των πεδίων που έχουν να κάνουν με τη δρομολόγηση) έως και εξερεύνηση του περιεχομένου (φορτίου) των πακέτων (deep packet inspection).

Το πλεονέκτημα μιας γενικής μεθόδου ανίχνευσης είναι ότι δεν είναι στενά εξαρτημένη από μια συγκεκριμένη τεχνική επίθεσης, οπότε μπορεί να δώσει ενδείξεις ακόμη και για μια επίθεση που βασίζεται σε μια εντελώς νέα άγνωστη τεχνική, αρκεί να υπάρχουν ενδείξεις για μεταβολές στην κίνηση. Προκειμένου όμως να διαπιστωθεί ότι ένα ρεύμα κίνησης αποκλίνει από την κανονική συμπεριφορά πρέπει να έχει καταγραφεί προηγουμένως η κανονική συμπεριφορά, ώστε να υπάρχει μέτρο σύγκρισης. Εάν η κανονική συμπεριφορά περιλαμβάνει περιστασιακές ακρότητες που δεν έχουν ληφθεί υπόψη στην καταγραφή της, είναι πιθανό να προκύψουν *ψευδώς θετικές* διαγνώσεις (false positives). Αντίθετα, αν μια επίθεση κίνησης αποκλίνει ελάχιστα από την ακραία κανονική συμπεριφορά, μπορεί να προκύψουν *ψευδώς αρνητικές* διαγνώσεις (false negatives).

Ας σημειωθεί ότι κατά τη διάρκεια μιας επίθεσης ένα σύστημα ανίχνευσης αποσκοπεί σε

ανίχνευση της επίθεσης, δηλαδή διαπίστωση ότι η επίθεση συμβαίνει μέσα σε ένα συγκεκριμένο χρονικό διάστημα,

ταυτοποίηση της επίθεσης, δηλαδή διαπίστωση του συγκεκριμένου τύπου της επίθεσης και

ποσοτικοποίηση της επίθεσης, δηλαδή διαπίστωση του μεγέθους της και της έκτασης των συνεπειών.

Στη συνέχεια δίνουμε μια σειρά από τεχνικές διάγνωσης ανωμαλιών στην κίνηση και πιθανών επιθέσεων.

Περιορισμός των διαστάσεων του σήματος

Η κίνηση, με όποιον τρόπο και αν αποτυπωθεί (αριθμός πακέτων σε παράθυρο χρόνου, χρόνοι άφιξης, ουρές κ.λπ.) είναι ένα «σήμα» θορυβώδες. Εφόσον ληφθούν υπόψη περισσότερες της μιας παράμετροι είναι και πολυδιάστατο, δηλαδή κάθε μέτρηση προερχόμενη π.χ. από ένα πακέτο περιλαμβάνει μια σειρά παραμέτρων, άρα είναι ένα διάνυσμα με πολλές συνιστώσες. Κάτι άλλο που πολλαπλασιάζει τον αριθμό των παραμέτρων είναι το πλήθος των σημείων παρατήρησης, π.χ. να γίνεται καταγραφή σε πολλές ζεύξεις ταυτόχρονα.

Ωστόσο σε ένα τέτοιο πολυπαραμετρικό σήμα κάποιες παράμετροι ή συνδυασμοί παραμέτρων είναι πιο σημαντικές από άλλες για τη διάγνωση μιας επίθεσης. Ένας τρόπος να διαπιστωθούν «αυτόματα» οι σημαντικές παράμετροι είναι να υιοθετηθεί μια μέθοδος περιορισμού του αριθμού των σημαντικών παραμέτρων (διαστάσεων).

Οι Lakhina, Crovella & Diot [LCD04] είχαν προτείνει το 2004 τη διάγνωση ανωμαλιών της κίνησης με μια μέθοδο βασισμένη στην *Ανάλυση Κυρίων Συνιστωσών* (Principal Component Analysis – PCA) [The15; Jol06; Agg17; Jac05], που εφαρμόζουν σε κίνηση με γνωστά ζεύγη διευθύνσεων πηγής-προορισμού.

Ο σκοπός της PCA είναι να περιορίσει προσεγγιστικά τις «διαστάσεις» ενός συστήματος. Κάθε μέθοδος περιορισμού των διαστάσεων βασίζεται στην υπόθεση ότι το

διαθέσιμο σύνολο δεδομένων έχει γεννηθεί από ένα σύστημα με σχετικά μικρό αριθμό μεταβλητών. Ο σκοπός της ανάλυσης είναι να προσδιοριστεί η δομή αυτού του συστήματος. Αποδεικνύεται ότι σε ένα πολυδιάστατο σύνολο δεδομένων, όπου κάθε δείγμα αποτελείται από ένα διάνυσμα συνιστωσών, η κατεύθυνση στην οποία μεγιστοποιείται η διασπορά των δεδομένων προσδιορίζεται από εκείνο το ιδιοδιάνυσμα του πίνακα συμμεταβλητότητας των δειγμάτων που αντιστοιχεί στη μέγιστη ιδιοτιμή. Η δεύτερη κύρια συνιστώσα επιλέγεται ώστε να είναι ορθογώνια προς το προηγούμενο διάνυσμα και να μεγιστοποιεί την προβολή της διασποράς σ' αυτήν την κατεύθυνση κ.ο.κ. (για μια σύντομη εισαγωγή βλ. [Jol90]).

Στα πειράματα που περιέχονται στο [LCD04] χρησιμοποιήθηκε ως διάσταση η κάθε διαφορετική ζεύξη (link), όπου μετρήθηκε η κίνηση με καταγραφή του ζεύγους πηγής-προορισμού σε κάθε πακέτο. Σε πειράματα με πάνω από 40 διαστάσεις (εν προκειμένω ζεύξεις) τα αριθμητικά αποτελέσματα έδειξαν ότι οι πρώτες 4 κύριες συνιστώσες είναι σημαντικές για την απόδοση των διακυμάνσεων κατά την κανονική συμπεριφορά, ενώ οι επόμενες 4 αποκαλύπτουν τη μη κανονική συμπεριφορά και μπορούν να αξιοποιηθούν για τη διάγνωση ανωμαλιών με καθορισμό κατάλληλων κατωφλίων.

Στατιστικές μέθοδοι

Εντροπία Ένα μέτρο ανωμαλίας της κίνησης που έχει χρησιμοποιηθεί είναι η *εντροπία* κατά Shannon [CT12]. Η εντροπία είναι ένα μονόμετρο μέγεθος που υπολογίζεται από μια κατανομή. Αν p_i είναι η πιθανότητα να εμφανισθεί η i-οστή τιμή μιας τυχαίας μεταβλητής που μπορεί να πάρει N διαφορετικές τιμές, η εντροπία της τυχαίας μεταβλητής είναι

$$H = -\sum_{i=1}^{N} p_i \log_2 p_i \tag{8.1}$$

όπου $\log_2$ είναι ο λογάριθμος με βάση το 2. Η εντροπία γενικά είναι μέτρο της αταξίας που εμφανίζεται σε μια κατανομή. Ελάχιστη εντροπία έχει μια κατανομή όταν μηδενίζονται οι πιθανότητες όλων των τιμών της τυχαίας μεταβλητής εκτός μιας, άρα η τυχαία μεταβλητή δεν είναι πια τυχαία (και προκύπτει εύκολα ότι $H = 0$). Μέγιστη είναι η εντροπία όταν όλες οι τιμές είναι ισοπίθανες, δηλ. $p_i = 1/N$ (οπότε $H = \log_2 N$).

Το πρώτο βήμα για την εφαρμογή ενός τέτοιου κριτηρίου είναι η επιλογή μιας παραμέτρου για την οποία θα καταγραφεί η σχετική συχνότητα. Για παράδειγμα, ας υποθέσουμε ότι εξετάζουμε τα πακέτα που διέρχονται από μια ζεύξη και καταγράφουμε τις διευθύνσεις προορισμού τους στη διάρκεια μιας χρονικής περιόδου διάρκειας T. Έστω ότι υπάρχουν N διαφορετικές διευθύνσεις προορισμού (ή ότι παρακολουθούμε τις N στατιστικά πιο σημαντικές). Στη συνέχεια καταγράφουμε το ποσοστό f_i με το οποίο εμφανίζεται η διεύθυνση i στο εν λόγω χρονικό διάστημα. Στο τέλος του διαστήματος θέτουμε $p_i = f_i$ και υπολογίζουμε την εντροπία μέσω του τύπου (8.1). Επαναλαμβάνουμε τις μετρήσεις και τον υπολογισμό σε κάθε διαδοχικό διάστημα μήκους T. Εναλλακτικά μπορούμε να σχηματίσουμε ένα κυλιόμενο παράθυρο με σταθερό αριθμό μετρήσεων προσθέτοντας κάθε φορά τα δεδομένα από το πιο πρόσφατο πακέτο και αφαιρώντας τα δεδομένα από το παλιότερο.

Στα πειράματα της εργασίας [Fei+03] διαπιστώθηκε μια αύξηση της εντροπίας περί το 20% κατά τη διάρκεια μιας επίθεσης DDoS. Δεν είναι απαραίτητο να σημειώνεται πάντοτε αύξηση στη διάρκεια μιας επίθεσης [LWD09], το βασικό είναι ότι μεταβάλλεται η εντροπία σε σύγκριση με την συνήθη κίνηση. Εάν π.χ. ένα botnet στείλει κίνηση προς ένα στόχο με σταθερό ρυθμό, η εντροπία πιθανώς θα πέσει (επειδή αυξάνεται η κανονικότητα και μειώνεται η αταξία).

Κριτήριο χ^2 Ένα άλλο κριτήριο που μπορεί να χρησιμοποιηθεί είναι η στατιστική δοκιμή χ^2 [Fei+03; NPK10]. Ο τρόπος χρήσης και πάλι συνίσταται στην παρακολούθηση μιας παραμέτρου ή ενός συνδυασμού παραμέτρων. Για παράδειγμα, επιλέγεται να παρακολουθείται η παράμετρος που υποδηλώνει τη θύρα υπηρεσίας (service port) κατατάσσοντας τα πακέτα σε κατηγορίες. Στην [Fei+03] το σχετικό πείραμα έχει γίνει διακρίνοντας τα πακέτα σε σχέση με τις τρεις υπηρεσίες HTTP, FTP, DNS (και οτιδήποτε άλλο σε μια τέταρτη κατηγορία). Στη συνέχεια υπολογίζεται για τα πακέτα που είναι μέσα στο παράθυρο η τιμή του χ^2 μέσω του τύπου

$$\chi^2 = \sum_{i=1}^{B} \frac{(N_i - n_i)^2}{n_i}$$

όπου B είναι οι διαφορετικοί τύπου πακέτων (εδώ $B = 4$), N_i είναι το πλήθος πακέτων που πέφτει στην κατηγορία i και n_i είναι το προσδοκώμενο πλήθος σε μια τυπική κατανομή. Και πάλι έχει παρατηρηθεί μια σημαντική αύξηση της τιμής χ^2 κατά τη διάρκεια μιας επίθεσης.

Η παραπάνω τεχνική προϋποθέτει την ανάγνωση των επικεφαλίδων των πακέτων, απομονώνοντας κατάλληλες παραμέτρους. Όμως ορισμένες από τις μεθόδους ανίχνευσης αρκούνται στο να χρησιμοποιήσουν ως βασικό κριτήριο το χρονισμό των μηνυμάτων και ορισμένες αρκούνται μόνο στον τελευταίο. Το εργαλείο ανίχνευσης *BotSniffer* [GZL08] μπορεί να ξεχωρίσει κίνηση επικοινωνίας ανάμεσα σε bots με το κέντρο διαχείρισής τους (C&C) μέσω HTTP ή IRC και χρησιμοποιείται ως plugin πάνω στο Snort. Το BotSniffer εξετάζει πρώτα απ' όλα το χρονισμό της επικοινωνίας. Εφόσον τα bots πάρουν μια εντολή από το κέντρο διαχείρισής τους, θα δημιουργηθεί ένα πλήθος απαντήσεων, πράγμα που δεν θα συνέβαινε στην κανονική χρήση π.χ. ενός καναλιού IRC. Η εν λόγω μορφή (pattern) κίνησης είναι πιθανό να επαναληφθεί ανάμεσα στα ίδια μέλη, πράγμα που θα επιβεβαιώσει περαιτέρω τις «υποψίες», δηλαδή θα αυξήσει την αξιοπιστία της διάγνωσης. Οι μηχανές που εμπλέκονται σε τέτοιου είδους κίνηση εγγράφονται σε ένα κατάλογο υπόπτων μηχανών που τίθενται υπό παρακολούθηση. Η παρακολούθηση συνίσταται κυρίως στο κατά πόσο μια μηχανή εκπέμπει και δέχεται μαζικά εντολές.

Ας σημειωθεί ότι η εν λόγω δοκιμή, όπως γενικά και οι υπόλοιπες μέθοδοι διάγνωσης ανώμαλης κίνησης, μπορεί να δοκιμαστεί και για την ανίχνευση τυχαίων ανωμαλιών, π.χ. οφειλομένων σε βλάβες [MFD19].

ΚΕΦΑΛΑΙΟ 9

Πρωτόκολλα ασφάλειας στο Internet

9.1 Transport Layer Security - TLS

Ο σκοπός του πρωτοκόλλου TLS (Transport Layer Security) είναι να δημιουργήσει ένα ασφαλές κανάλι επικοινωνίας ανάμεσα σε δύο άκρα δεδομένης μιας σύνδεσης στο στρώμα μεταφοράς, δηλαδή να μετατρέψει μια σύνδεση στο επίπεδο μεταφοράς σε ασφαλή τοιαύτη. Το νόημα της ασφαλούς σύνδεσης είναι ότι εξασφαλίζεται ότι οι δύο πλευρές είναι ταυτοποιημένες, ότι οι πληροφορίες που ανταλλάσσονται είναι κρυμμένες από τρίτους και ότι το ρεύμα των πληροφοριών δεν μπορεί να αλλοιωθεί από τρίτους. Δημιουργώντας ένα ασφαλές κανάλι επικοινωνίας μπορούν να εξυπηρετηθούν όλες οι εφαρμογές των ανωτέρων επιπέδων «διαφανώς», δηλαδή χωρίς να γίνεται προσαρμογή του πρωτοκόλλου σε κάθε μία εξ αυτών. Εναλλακτικά όμως μπορεί κανείς να δημιουργήσει εξειδικευμένες εκδοχές του πρωτοκόλλου για ορισμένες εφαρμογές.

Το βασικό σενάριο χρήσης είναι σχετικό με την επικοινωνία client-server και το HTTP. Σύμφωνα με αυτό το συνηθισμένο σενάριο υπάρχει ένας client που γνωρίζει το domain name ενός server, ενώ αντίστροφα ο server δεν γνωρίζει τον client. Ο client θέλει να κατεβάσει μια ιστοσελίδα χωρίς να έχει οπωσδήποτε την υποχρέωση να αυθεντικοποιηθεί προς τον server. Στο απλό σενάριο η ασφάλεια μπορεί να περιορίζεται σε μια αρχική αυθεντικοποίηση του ίδιου του χρήστη με τη μέθοδο login name / password και στη συνέχεια ο server βασίζεται στην υπόθεση ότι τα επόμενα αιτήματα προέρχονται από την ίδια πηγή. Το πρωτόκολλο καλείται να εξυπηρετήσει την ασφαλή ανταλλαγή δεδομένων ανάμεσα στις δύο πλευρές με χρήση κρυπτογραφίας. Για να συμβεί όμως αυτό πρέπει οι δύο πλευρές να συμφωνήσουν στη χρήση μιας κρυπτογραφικής μεθόδου και στις τιμές των σχετικών παραμέτρων. Αυτό γίνεται σε μια φάση συνεννόησης που προηγείται της ανταλλαγής δεδομένων και που ονομάζεται χειραψία (handshake). Αν πρόκειται να χρησιμοποιηθεί ένας συμμετρικός αλγόριθμος όπως AES ή CHACHA, η ανταλλαγή κλειδιού γίνεται πιο πριν με Diffie Hellman ή ελλειπτικό Diffie Hellman.

Σχεδιαστικοί στόχοι και επιλογές

Το πρωτόκολλο TLS είναι αρκετά πολύπλοκο και για την καλύτερη κατανόησή του είναι σκόπιμη μια εισαγωγή στους στόχους και στη λογική, με την οποία έχει σχεδιασθεί. Ο σχεδιαστής ενός τέτοιου πρωτοκόλλου καλείται να λύσει μια σειρά προβλημάτων, που κυρίως είναι (α) με ποια μέθοδο θα γίνει η κρυπτογράφηση ενός μηνύματος, (β) πώς θα διασφαλισθεί η ακεραιότητά του, (γ) πώς θα γίνει η ταυτοποίηση των δύο πλευρών, αλλά για να συμβούν αυτά πρέπει να υπάρξει ανταλλαγή κλειδιών και άλλων παραμέτρων και πρέπει να καθοριστεί η σωστή σειρά των απαραίτητων επεξεργασιών και μεταμορφώσεων του μηνύματος (κρυπτογράφηση-αποκρυπτογράφηση, συμπίεση-αποσυμπίεση). Επίσης πρέπει να αποφασίσει πόσο συχνά θα αλλάζει κλειδιά και πώς θα χειριστεί μηνύματα αυθαίρετα μεγάλου μήκους, καθώς και αν θα κρυπτογραφεί με τον ίδιο τρόπο (αλγόριθμο και κλειδί) όλα μαζί τα ταυτόχρονα ρεύματα πληροφορίας ανάμεσα σε δύο πλευρές για λόγους οικονομίας στη διαπραγμάτευση.

Όλα τα πρωτόκολλα ασφάλειας δημιουργούνται από ένα σχεδιαστή που έχει στο μυαλό του συγκεκριμένα σενάρια επιθέσεων και προσπαθεί να βρει κατάλληλη άμυνα. Όταν δοθεί μια αμυντική λύση που φαίνεται κατάλληλη, αυτή αργά ή γρήγορα ακυρώνεται από νέες πιο εφευρετικές επιθέσεις. Παράλληλα η έρευνα γύρω από τις υφιστάμενες λύσεις αποκαλύπτει νέες τρωτότητες ή κατατάσσει τις χρησιμοποιούμενες λύσεις σε λιγότερο και περισσότερο ασφαλείς. Αυτό το «παιχνίδι» ανάμεσα στην επίθεση και στην άμυνα οδηγεί σε μεταβολές και αναθεωρήσεις ενός πρωτοκόλλου ασφάλειας ή και στην κατάργησή του και αντικατάσταση από κάτι άλλο.

Ανάμεσα στις πιο ριψοκίνδυνες και μεταβαλλόμενες σχεδιαστικές επιλογές για το TLS είναι το σύνολο των εγκεκριμένων αλγορίθμων κρυπτογράφησης, δεδομένου ότι ορισμένοι αλγόριθμοι θεωρούνται τρωτοί από ένα χρονικό σημείο και μετά ή τουλάχιστον χρειάζονται μια νέα παραμετροποίηση, π.χ. ένα πιο μακρύ κλειδί. Ορισμένες από τις σχεδιαστικές επιλογές που καταργούνται στην επόμενη έκδοση του πρωτοκόλλου έχει συμβεί να επανέρχονται στη μεθεπόμενη, γεγονός που κάνει τον σχεδιασμό τέτοιο ώστε να επιτρέπει τέτοιες ανατροπές κρατώντας όλες τις επιλογές ανοιχτές. Μια τέτοια επιλογή που αλλάζει προς τη μιά ή την άλλη κατεύθυνση έχει αποδειχτεί ιστορικά ότι είναι το αν θα γίνει συμπίεση ενός μηνύματος πριν από την κρυπτογράφησή του. Τέλος, ο σχεδιασμός οφείλει να προσπαθεί να διατηρήσει κατά το δυνατό τη συμβατότητα με προηγούμενες εκδόσεις του πρωτοκόλλου.

Τα παραπάνω είναι ζητήματα που απασχολούν τον σχεδιασμό ως προς την φάση της ανταλλαγής δεδομένων ανάμεσα σε δύο μηχανές. Ωστόσο σε πρωτόκολλα που διαθέτουν πλήθος επιλογών οι δύο μηχανές πρέπει πριν να μπουν σε αυτήν την φάση να καθορίσουν τις συγκεκριμένες επιλογές που θα ακολουθήσουν. Μια τέτοια διαπραγμάτευση πρέπει να γίνει σε μια προηγούμενη φάση, όπως για να λειτουργήσει ένα νοητό κύκλωμα (virtual circuit) προηγείται μια φάση συνεννόησης των δύο πλευρών, μια φάση «εγκατάστασης» του νοητού κυκλώματος.

Η ορθή λειτουργία ενός πρωτοκόλλου εξυπηρετεί μια σειρά από *λειτουργικούς* (functional) στόχους που θα μπορούσαν να αποτυπωθούν, ιδιαίτερα στις αρχικές φάσεις ενός συστηματικού σχεδιασμού, σε μια σειρά από *λειτουργικές απαιτήσεις* (functional requirements) και να οδηγήσουν σε αντίστοιχες προδιαγραφές. Η έννοια των λειτουργικών στόχων με απλά λόγια είναι να επιτυγχάνει το πρωτόκολλο τον βασικό του σκοπό, π.χ. εδώ να μεταφέρει ένα μήνυμα ασφαλώς από τη μια πλευρά στην άλλη και να μην «κρασάρει» (να μην πέφτει σε deadlock). Κατά κανόνα όμως ένα πρωτόκολλο οφείλει να καλύπτει και ορισμένες *μη λειτουργικές απαιτήσεις* (non-functional requirements), εκ των οποίων οι πιο κλασσικές είναι απαιτήσεις επίδοσης. Οι τελευταίες έχουν να κάνουν με τιμές παραμέτρων ταχύτητας, κατανάλωσης πόρων, βαθμού

αξιοποίησης πόρων κ.α. Μια από τις κύριες τέτοιες παραμέτρους είναι η διαπερατότητα (throughput) ή βαθμός αξιοποίησης του μέσου επικοινωνίας, που διαφορετικά εκφράζεται ως τελικός ρυθμός μετάδοσης τον οποίο παρέχει το δίκτυο που ενώνει τις δύο πλευρές.

Σε ένα απλό σενάριο όπου δύο μηχανές ενώνονται απ' ευθείας με μια ζεύξη ρυθμού μετάδοσης v, ποιος είναι ο τελικός ρυθμός μετάδοσης $v' = \eta v$, όπου το η είναι η διαπερατότητα; Ισχύει οπωσδήποτε $\eta < 1$ επειδή το εν λόγω πρωτόκολλο θα προσθέσει παραπάνω bits, θα χρειαστεί χρόνο επεξεργασίας των δεδομένων κ.λπ. Σε πόσο χαμηλό η όμως θα οδηγηθούμε είναι θέμα μιας μελέτης επιδόσεων. Για παράδειγμα, η συμπίεση του μηνύματος θα οδηγήσει στη μετάδοση λιγότερων bits, γεγονός που ωθεί το throughput προς τα πάνω, ωστόσο πρέπει να συνυπολογισθεί ο χρόνος επεξεργασίας, ενώ πρέπει να ληφθούν υπόψη οι δυνατότητες μιας μηχανής να υλοποιήσει μια τέτοια επιλογή και οι επιπτώσεις σε άλλες λειτουργίες της. Η χαμηλότερη αποδεκτή τιμή του η θα καθοριστεί πάντοτε σε συνδυασμό με άλλες ιδιότητες, αλλά πάντοτε ο στόχος είναι η βελτίωση ενός τέτοιου δείκτη επίδοσης.

Τα παραπάνω επηρεάζουν τον σχεδιασμό ενός τέτοιου πρωτοκόλλου, αξιολογούνται σε διάφορα χρονικά σημεία και οδηγούν σε νέες επιλογές στην επόμενη κάθε φορά έκδοση. Το τελικό αποτέλεσμα είναι μια πολύπλοκη σύνθεση όλων των διαθέσιμων δεδομένων.

Μικρή ιστορική ανασκόπηση

Το πρωτόκολλο με το όνομα *Secure Sockets Layer* (SSL) προτάθηκε από την *Netscape Communications* της οποίας βασικό προϊόν ήταν ο browser *Netscape Communicator*. Ο βασικός του στόχος ήταν η παροχή ασφάλειας ώστε να μπορεί κανείς να κάνει αγορές μέσω Internet δίνοντας την πιστωτική του κάρτα. Η πρώτη του δημόσια έκδοση SSL 2.0 βγήκε τον Φεβρουάριο του 1995 (και ας σημειωθεί ότι η RFC 1945 για το HTTP/1.0 - με συγγραφέα τον Tim Berners Lee κ.α. - εκδόθηκε μόλις τον Μάιο του 1996). Παρ' όλο που ο αρχικός σκοπός της δημιουργίας του SSL ήταν να εξυπηρετήσει το HTTP, στη συνέχεια χρησιμοποιήθηκε για email, VPNs, μεταφορά φωνής και video και περαιτέρω για ανταλλαγή δεδομένων στο IoT.

Το 1996 κυκλοφόρησε η έκδοση 3.0 και είναι η έκδοση πάνω στην οποία βασίζεται το TLS. Το TLS 1.0 (RFC 2246 του 1999) προέκυψε ως αναβάθμιση του SSL 3.0. Τον Μάρτιο του 2011 αναθεωρήθηκαν όλες οι προηγούμενες εκδόσεις του TLS με την RFC 6176. Η πιο πρόσφατη σήμερα (2021) έκδοση του TLS είναι η έκδοση 1.3 του Αυγ. 2018 (RFC 8446). Από τον Οκτ. του 2018 οι εκδόσεις 1.0 και 1.1 θεωρούνται παρωχημένες (deprecated) από μια ομάδα μεγάλων εταιριών. Στη συνέχεια περιγράφεται το TLS 1.3, με ορισμένες αναφορές σε προηγούμενες εκδόσεις όταν αυτό διευκολύνει την κατανόηση. Με το TLS 1.3 έγινε μια σειρά απλοποιήσεων με στόχο καλύτερη ασφάλεια και μεγαλύτερη ταχύτητα. Οι απλοποιήσεις περιλαμβάνουν την κατάργηση ορισμένων αλγορίθμων κρυπτογράφησης και ορισμένων συναρτήσεων κατακερματισμού, ενώ αφαιρέθηκε η δυνατότητα συμπίεσης του μηνύματος και έγινε μια συμπύκνωση των ανταλλαγών που χρειάζονται κατά τη χειραψία.

Γενική θεώρηση του TLS

Η πιο κάτω περιγραφή βασίζεται στην έκδοση 1.3 του πρωτοκόλου *Transport Layer Security* (TLS) ορίζεται στην RFC 8446[1] (Αύγ. 2018). Προηγούμενες εκδόσεις δεν είναι απαραίτητα συμβατές. Το πρωτόκολλο TLS αποτελείται από δύο κύριες συνιστώσες:

[1] `https://tools.ietf.org/html/rfc8446`

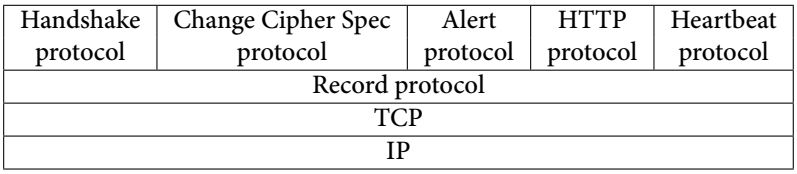
Handshake protocol (πρωτόκολλο χειραψίας) μέσω του οποίου γίνεται η προετοιμασία μιας ασφαλούς συνόδου (session) μεταξύ ενός client και ενός server, δηλαδή η διαπίστωση των ταυτοτήτων των δύο πλευρών (authentication), η συμφωνία στο είδος της κρυπτογράφησης που θα χρησιμοποιήσουν, η ανταλλαγή κλειδιών και άλλων παραμέτρων. Οποιεσδήποτε συνδέσεις (connections) βρίσκονται κάτω από την ίδια σύνοδο χρησιμοποιούν τις ίδιες παραμέτρους κρυπτογράφησης.

Record protocol (πρωτόκολλο εγγραφής ή καταγραφής) που διαχειρίζεται την ασφαλή σύνοδο (που μπορεί να περιλαμβάνει μια ή περισσότερες συνδέσεις - connections) στη φάση της ανταλλαγής δεδομένων. Η κίνηση διαιρείται σε τμήματα που ονομάζονται *εγγραφές* (records) και η κάθε εγγραφή προστατεύεται χωριστά.

Change Cipher Spec protocol (πρωτόκολλο μεταβολής κώδικα) μέσω του οποίου μεταβάλλεται (ή προσδιορίζεται αρχικά) ο επιλεγόμενος κώδικας κρυπτογράφησης.

Alert protocol (πρωτόκολλο συναγερμού) που ενεργοποιείται εφόσον συμβαίνουν σφάλματα ή απρόβλεπτες καταστάσεις.

Heartbeat protocol το οποίο στέλνοντας πλαίσια σε χρονικές περιόδους που δεν υπάρχει δραστηριότητα διασφαλίζει ότι συνεχίζεται η σύνδεση.



Σχήμα 9.1: Στοίβα πρωτοκόλλων TLS

Το πρωτόκολλο εγγραφής

Ο σκοπός του πρωτοκόλλου εγγραφής (ή καταγραφής) είναι να μεταφέρει από τη μια πλευρά της επικοινωνίας στην άλλη ένα προστατευμένο μήνυμα. Αν το μήνυμα είναι μεγάλου μήκους προηγείται ο *τεμαχισμός* (fragmentation) του μηνύματος. Κάθε *τεμάχιο* ή *θραύσμα* (fragment) εφόσον μετατραπεί στο αντίστοιχο προστατευμένο μήνυμα αποτελεί μια *εγγραφή* (record) του TLS (ή του SSL πιο πριν). Σύμφωνα με την σύσταση RFC 8446 για το TLS 1.3 το τεμάχιο χαρακτηρίζεται ως `TLSPlaintext`.[2]

Στη συνέχεια το κάθε συγκεκριμένο τεμάχιο `TLSPlaintext` αυθεντικοποιείται και κρυπτογραφείται, οπότε μεταμορφώνεται σε ένα αντίστοιχο τεμάχιο `TLSCiphertext`.

[2] Ως την προηγούμενη έκδοση (TLS 1.2, RFC 5246) το αρχικό τεμάχιο μπορεί στη συνέχεια να συμπιεσθεί, μετατρεπόμενο έτσι από `TLSPlaintext` σε `TLSCompressed`. Αντίστοιχα στο SSL είχαν ορισθεί τα `SSLplaintext` και `SSLCompressed`. Η συμπίεση ήταν προαιρετική και γινόταν πάντοτε πριν το επόμενο βήμα της κρυπτογράφησης (διότι το κρυπτογραφημένο κείμενο είναι ψευδοτυχαίο και δεν μπορεί να συμπιεσθεί). Ωστόσο διαπιστώθηκαν αδυναμίες ασφάλειας στο συνδυασμό συμπίεσης-κρυπτογράφησης και στο TLS 1.3 καταργήθηκε η συμπίεση. Κατά συνέπεια στην RFC 8446 δεν ορίζεται το `TLSCompressed`.

Αυθεντικοποίηση και κρυπτογράφηση

Οι δύο διαδικασίες αυθεντικοποίησης και κρυπτογράφησης θεωρητικά θα μπορούσαν να γίνουν με οποιαδήποτε σειρά ή και συγχρόνως, δηλαδή είτε (α) πρώτα αυθεντικοποίηση του `TLSPlaintext` και μετά κρυπτογράφηση του συνδυασμού μηνύματος και MAC, είτε (β) πρώτα κρυπτογράφηση του μηνύματος και μετά δημιουργία ενός MAC πάνω `TLSCiphertext`, είτε (γ) ταυτόχρονη κρυπτογράφηση και αυθεντικοποίηση. Το TLS ως την έκδοση 1.2 χρησιμοποιούσε την μέθοδο (α) (ενώ το IPsec την (β) και το SSH την (γ)). Όμως με την έκδοση TLS 1.3 χρησιμοποιούνται πλέον κώδικες που έχουν τη δυνατότητα να κάνουν συνδυασμένη κρυπτογράφηση και αυθεντικοποίηση. Οι κώδικες αυτοί είναι γνωστοί ως κώδικες AEAD - *Authenticated Encryption with Associated Data*. Αυτοί προσδιορίζονται στον πίνακα B.4 της σύστασης (βλ. και πιο κάτω κατάλογο) και περαιτέρω στις συστάσεις RFC 5156, 8439 και 6655.

Οι κώδικες AEAD έχουν τη δυνατότητα να αυθεντικοποιούν μεν ολόκληρο ένα εισερχόμενο string, περιλαμβανομένων δηλαδή των επικεφαλίδων κ.λπ., αλλά να κρυπτογραφούν επιλεγμένα πεδία. Εν προκειμένω πεδία που δεν κρυπτογραφούνται είναι αυτά που δίνουν τον τύπο, το μήκος, την έκδοση, τον αριθμό σειράς για ένα string που ακολουθεί.

Ο συνδυασμός συνάρτησης κατακερματισμού και κρυπτογραφικού κώδικα μαζί με παραμέτρους (όπως το μήκος του MAC) ονομάζεται *cipher spec* στις συστάσεις TLS, είτε γίνεται χωριστά η κρυπτογράφηση από την αυθεντικοποίηση είτε μαζί. Σε κάθε περίπτωση χρειάζονται τα αντίστοιχα κλειδιά, οπότε χρειάζεται ο κατάλληλος μηχανισμός ανταλλαγής κλειδιών. Ο συνδυασμός ενός μηχανισμού ανταλλαγής κλειδιών και ενός cipher spec ονομάζεται *cipher suite*.

Για το TLS 1.3 προσδιορίζονται μόνο 5 τέτοιες σουίτες (περιορίζοντας δραστικά τις αρκετές δεκάδες που ήταν ήταν διαθέσιμες ως το TLS 1.2). Εξ αυτών οι 4 χρησιμοποιούν για την κρυπτογράφηση το AES (Advanced Encryption Standard) με μήκος κλειδιού 128 και 256, ενώ υπάρχει και μια σουίτα για κρυπτογράφηση ρεύματος με τον αλγόριθμο CHACHA [Ber+08]. Για την αυθεντικοποίηση χρησιμοποιείται ο SHA-2 με μήκη 256 και 384.

Οι σουίτες του TLS 1.3 είναι οι εξής:

```
TLS_AES_128_GCM_SHA256
TLS_AES_256_GCM_SHA384
TLS_CHACHA20_POLY1305_SHA256
TLS_AES_128_CCM_SHA256
TLS_AES_128_CCM_8_SHA256
```

Ένας δεύτερος λόγος που οι σουίτες είναι λιγότερες στο TLS 1.3 (πλην της κατάργησης αλγορίθμων κρυπτογράφησης που περιλαμβάνονταν στο TLS 1.2) είναι ότι έχει αφαιρεθεί ο RSA από τις μεθόδους ανταλλαγής κλειδιών και έχει διατηρηθεί μόνο η ανταλλαγή *εφήμερου* (ephemeral) κλειδιού Diffie-Hellman (η ανταλλαγή αυτή είναι γνωστή ως EDH ή DHE - Diffie-Hellman Ephemeral).

Ανταλλαγή κλειδιών

Για να προχωρήσουν δύο πλευρές στην κρυπτογραφημένη ανταλλαγή αυθεντικοποιημένων και κρυπτογραφημένων τεμαχίων χρειάζονται κλειδιά για την συνάρτηση κατακερματισμού, κλειδιά συμμετρικής κρυπτογράφησης, και αρχικές τιμές (IVs - initial values) εφόσον χρησιμοποιήσουν χρησιμοποιήσουν τη μέθοδο cipher block chaining (CBC), δηλαδή συνολικά έξι κλειδιά (τρία ανά κατεύθυνση). Ωστόσο στην περίπτωση των κωδίκων AEAD του TLS 1.3 δεν χρειάζονται χωριστά κλειδιά για την

παραγωγή MAC. Επίσης δεν χρησιμοποιείται CBC. Παρ' όλα αυτά υπολογίζονται δυο αρχικές τιμές (IV) επειδή οι αλγόριθμοι AEAD παίρνουν ως είσοδο ένα κλειδί και ένα τυχαίο αριθμό (nonce), εκτός του plaintext και των επιπρόσθετων δεδομένων που θα μείνουν ακρυπτογράφητα. Ο τυχαίος αριθμός σχηματίζεται από τον αριθμό ακολουθίας και τις αρχικές τιμές (`client_write_iv` ή `server_write_iv`). Επομένως και στο TLS 1.3 χρειάζεται μια ανταλλαγή κλειδιών που θα δώσει 4 αριθμούς. Ωστόσο αυτή η ανταλλαγή κλειδιών, όπως περιγράφεται και παρακάτω, γίνεται σε δύο φάσεις, δηλαδή εν τέλει σχηματίζονται 8 κλειδιά. Στο TLS 1.3 η ανταλλαγή κλειδιών βασίζεται στον αλγόριθμο *ephemeral Diffie-Hellman*.

Ως γνωστόν για να λειτουργήσει η ανταλλαγή Diffie-Hellman οι δύο πλευρές A, B πρέπει να έχουν ανταλλάξει ένα πρώτο αριθμό p, μια αρχική ρίζα a του p, ενώ στη συνέχεια κάθε πλευρά ορίζει ένα ιδιωτικό κλειδί (αντιστοίχως) X_A, X_B και εξ αυτού δημιουργεί ένα δημόσιο κλειδί Y_A, Y_B που το στέλνει στην άλλη πλευρά. Εν τέλει η κάθε πλευρά χρησιμοποιεί το p, το a, το δημόσιο κλειδί της άλλης πλευράς και το δικό της ιδιωτικό κλειδί και υπολογίζει έναν αριθμό K που τελικά είναι ο ίδιος και για τις δύο πλευρές. Εάν η πλευρά A χρησιμοποιεί μονίμως το δημόσιο κλειδί Y_A, η πλευρά B έχει τη βεβαιότητα ότι πρόκειται για την ίδια οντότητα (στη βάση μιας αρχικής αυθεντικοποίησης που είχε γίνει στο παρελθόν). Ωστόσο αν οι δύο πλευρές χρησιμοποιούν μονίμως το ίδιο δημόσιο κλειδί θα φτάνουν πάντοτε στο ίδιο κοινό μυστικό κλειδί K, το οποίο μπορεί κάποια στιγμή να γίνει γνωστό σε κάποιον επιτιθέμενο. Στην εφήμερη ανταλλαγή Diffie-Hellman ο σκοπός είναι να φτάνουν οι A, B σε νέο κοινό κλειδί K που θα χρησιμοποιούν σε μια νέα τους επικοινωνία. Προς τούτο η μία τουλάχιστον πλευρά πρέπει να παρουσιάσει ένα «φρέσκο» δημόσιο κλειδί, αλλά πρέπει να λυθεί το πρόβλημα της αυθεντικοποίησής της. Το τελευταίο πρόβλημα μπορεί να λυθεί αν η εν λόγω πλευρά υπογράψει το μήνυμα με το οποίο στέλνει την νέα τιμή του δημόσιου κλειδιού.

Η γνώση των δύο πλευρών ότι μπορούν να χρησιμοποιήσουν μόνο Diffie-Hellman δίνει τη δυνατότητα για μείωση του μήκους της χειραψίας (σε σύγκριση με προηγούμενες εκδόσεις του TLS ή του SSL), δεδομένου ότι ο client μπορεί αμέσως με το μήνυμα Client Hello να στείλει τις απαραίτητες παραμέτρους.

Μετά από την ολοκλήρωση ενός μέρους της χειραψίας οι δύο πλευρές αποκτούν 4 αριθμούς, εκ των οποίων οι δύο είναι τα κλειδιά της κάθε μιας από τις δύο κατευθύνσεις επικοινωνίας και οι άλλοι δύο είναι αρχικές τιμές (IV - initial values) που εισάγονται στον AEAD. Η συνέχεια της χειραψίας γίνεται κρυπτογραφημένη με αυτές τις τιμές και καταλήγει στον υπολογισμό τεσσάρων αντιστοίχων τιμών για την κρυπτογράφηση του μηνύματος (δηλαδή των δεδομένων εφαρμογής - application data) από το πρωτόκολλο εγγραφής.

Το πρωτόκολλο χειραψίας

Ο βασικός στόχος είναι να συμφωνήσουν οι δύο μηχανές στους αλγόριθμους και τις παραμέτρους για την κρυπτογράφηση των δεδομένων που θα ανταλλάξουν αργότερα. Ως συνήθως το αντικείμενο μιας τέτοιας διαπραγμάτευσης είναι να βρεθούν μέθοδοι που υποστηρίζονται και από τις δύο πλευρές.

Η φάση της χειραψίας στο TLS 1.3 περιέχει τρεις ομάδες μηνυμάτων, πρώτη από τον client στον server, δεύτερη από τον server στον client και τρίτη από τον client στον server. Για καθένα από τα τρία αυτά σύνθετα μηνύματα χρησιμοποιείται συχνά ο όρος *πτήση – flight*. Εξ αυτών το πρώτο λέγεται συνολικά Client Hello, το δεύτερο Server Hello και το τρίτο Client Finish. Καθένα ενσωματώνεται σε ένα πακέτο. Ωστόσο από

ένα σημείο και μετά η ανταλλαγή είναι κρυπτογραφημένη. Στο Σχ. 9.2 φαίνεται το πρωτόκολλο χειραψίας (με τη μορφή ενός Message Sequence Chart).

Client Hello

Αρχικά ο client στέλνει ένα μήνυμα `ClientHello`. Το ότι πρόκειται για μήνυμα τέτοιου τύπου αποτυπώνεται σε κατάλληλη επικεφαλίδα με συγκεκριμένο κωδικό. Το μήνυμα περιέχει στη συνέχεια τις εξής πληροφορίες:

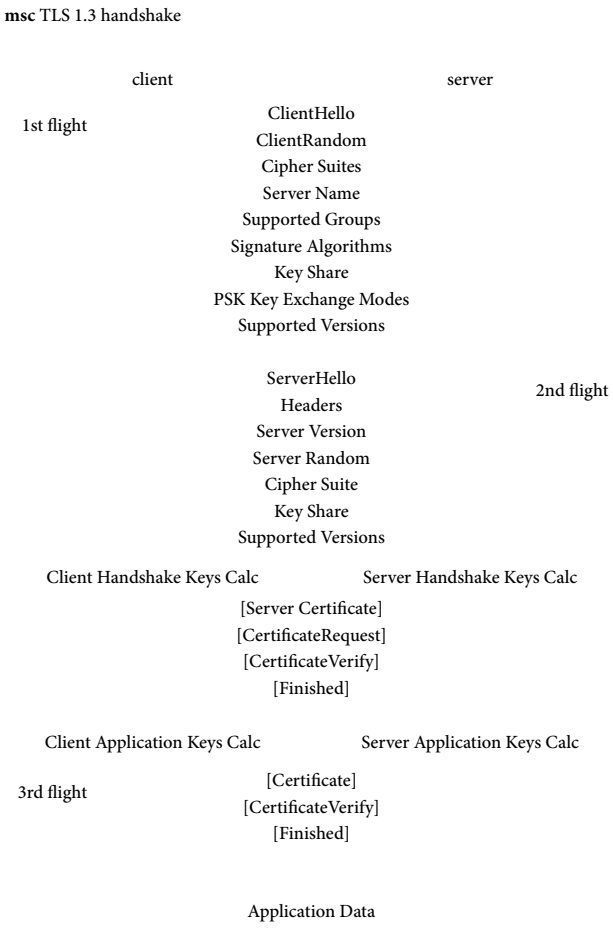
- Ένα τυχαίο αριθμό (nonce) για να αποτρέψει τις επιθέσεις επανάληψης και τις εκδόσεις του TLS που υποστηρίζει ο client.
- Ένα κατάλογο από αλγόριθμους κρυπτογράφησης (Cipher Suites) που υποστηρίζει ο client. Οι αλγόριθμοι αυτοί εφόσον πρόκειται για το TLS 1.3 επιλέγονται εκ των AEAD.
- Τα επόμενα πεδία που ονομάζονται «επεκτάσεις». Η πρώτη εξ αυτών είναι το domain name του server (*Server Name Indication*, γραμμένο σε format UTF8).
- Την επέκταση *Supported Groups* με την οποία ο client δηλώνει την προτίμησή του για αλγόριθμους ECDHE (εκ των secp256r1, secp384r1, secp521r1, x25519, x448) και DHE (ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192), γραμμένους κατά σειρά προτίμησης.
- Μια ομάδα από αλγόριθμους υπογραφής.
- Ένα τουλάχιστον δημόσιο κλειδί για κάποιον από τους αλγορίθμους που είχε στείλει με την επέκταση Supported Groups. Το δημόσιο κλειδί θα επιτρέψει στον server να απαντήσει κρυπτογραφημένα μετά από το `ServerHello`.
- Κατά σειρά προτίμησης τις εκδόσεις του TLS που υποστηρίζει.

Server Hello

Ο server απαντάει με μήνυμα `ServerHello`, με το οποίο οριστικοποιεί την επιλογή αλγορίθμων και παραμέτρων, παρέχει διάφορες πληροφορίες στον client και προαιρετικά ζητάει ένα πιστοποιητικό από τον client. Από ένα σημείο και μετά το μήνυμα είναι κρυπτογραφημένο. Επίσης, ο server επιλέγει μια από τις μεθόδους που είχε παραλάβει από τον client στην επέκταση Supported Groups, επιλέγει ένα κατάλληλο ιδιωτικό κλειδί και υπολογίζει το αντίστοιχο δημόσιο κλειδί. Συνοπτικά ο server παραθέτει τα εξής στο μήνυμά του:

- Ένα τυχαίο αριθμό (nonce).
- Την επιλογή αλγόριθμου κρυπτογράφησης των δεδομένων που θα ανταλλαγούν μετά τη χειραψία. Η επιλογή γίνεται μέσα από τον κατάλογο Cipher Suites που είχε στείλει ο client στον server με το Client Hello. (Σε προηγούμενες εκδόσεις του TLS επιλεγόταν και μια μέθοδος συμπίεσης).

  Ο server είναι υποχρεωμένος να επιλέξει για την κρυπτογράφηση ένα εκ των αλγορίθμων που έχει προτείνει ο client (διαφορετικά η σύνοδος θα οδηγηθεί σε διακοπή).



Σχήμα 9.2: Χειραψία TLS 1.3

- Το προσωρινό κλειδί που υπολόγισε μόλις προηγουμένως και θα χρησιμοποιηθεί για να φτάσουν οι δύο πλευρές σε συμφωνία συμμετρικού κλειδιού.
- Την επιλογή για την έκδοση TLS. Εδώ εφόσον επιλεγεί το TLS 1.3 θα ακολουθήσουν μηνύματα όπως προβλέπονται από το TLS 1.3.

Στο σημείο αυτό ο server χρησιμοποιεί το δημόσιο κλειδί που του είχε στείλει ο client με το Client Hello, το ιδιωτικό κλειδί του server και ένα κωδικό κατακερματισμού (hash) που δημιουργείται εφαρμόζοντας τον SHA256 πάνω στην ως τώρα χειραψία (Client Hello και Server Hello) και εν τέλει κατασκευάζει δύο συμμετρικά κλειδιά - ένα ανά κατεύθυνση - και δύο αρχικές τιμές (IV).

Επίσης ο server δημιουργεί ένα `handshake secret`, δηλαδή ένα string που προέρχεται από τα προηγούμενα κλειδιά και την ως τώρα χειραψία και εμπλέκει την συνάρτηση κατακερματισμού HKDF.

Ομοίως ο client χρησιμοποιεί το δημόσιο κλειδί που του είχε στείλει ο server με το ως τώρα Server Hello, το ιδιωτικό κλειδί του client και ένα κωδικό κατακερματισμού (hash) που δημιουργείται εφαρμόζοντας τον SHA256 πάνω στην ως τώρα χειραψία (Client Hello και Server Hello) και κατασκευάζει από τη δική του πλευρά τα ίδια συμμετρικά κλειδιά και τις αρχικές τιμές. Τα επόμενα που στέλνει ο server είναι κρυπτογραφημένα (στο Σχήμα 9.2 είναι κλεισμένα μέσα σε αγκύλες []):

- Το ψηφιακό πιστοποιητικό του server και προαιρετικά μια αλυσίδα πιστοποιητικών που καταλήγουν σε ένα έμπιστο πιστοποιητικό που υπάρχει στον client.

- Επαλήθευση πιστοποιητικού (Certificate Verify), δηλαδή πληροφορία που αποδεικνύει ότι το δημόσιο κλειδί που έστειλε προηγουμένως ο server έχει όντως δημιουργηθεί από τον κάτοχο του πιστοποιητικού. Δεδομένου όμως ότι στο TLS 1.3 το κλειδί είναι υποχρεωτικά μιας χρήσης (εφήμερο – ephemeral) δεν μπορεί να είναι το ίδιο με το δημόσιο κλειδί του πιστοποιητικού. Προς τούτο ο server υπογράφει ένα hash των μηνυμάτων της χειραψίας με το ιδιωτικό κλειδί του πιστοποιητικού. Ο client επιβεβαιώνει την υπογραφή με το δημόσιο κλειδί του πιστοποιητικού.

- Μαζί με το τελικό `Finished` ο server στέλνει ένα hash πάνω σε όλα τα μέχρι τώρα μηνύματα, περιλαμβανομένου του αμέσως προηγουμένου.

Ο server μπορεί πιο πριν προαιρετικά να στείλει ένα αίτημα πιστοποιητικού (`Certificate Request`) προς τον client.

Client Handshake Finished

Ακολουθεί η τελευταία ομάδα μηνυμάτων του client προς τον server. Ο client γεννάει μια τυχαία ακολουθία μήκους 48 bytes που λέγεται Master Secret και την στέλνει στον server κρυπτογραφημένη με το δημόσιο κλειδί του server. Αν έχει προηγηθεί ένα αίτημα του server προς τον client για πιστοποιητικό, ο τελευταίος πρέπει να στείλει ένα πιστοποιητικό και. Αν έχει ζητηθεί, στέλνει ένα μήνυμα `CertificateVerify`. Εν τέλει στέλνει ένα μήνυμα `Finished` που περιλαμβάνει ένα MAC πάνω σε όλη τη χειραψία και έχει χρησιμοποιήσει ως κλειδί για το MAC το Master Secret.

Προετοιμασία για την ανταλλαγή δεδομένων

Ο server παίρνει το `handshake secret` που είχε υπολογίσει με άλλα κλειδιά και αρχικές τιμές παραπάνω, καθώς και ένα hash πάνω στην ως τώρα χειραψία και κατασκευάζει μέσω και της HKDF δύο μυστικά συμμετρικά κλειδιά (ένα ανά κατεύθυνση) για τη φάση ανταλλαγής δεδομένων και δύο αρχικές τιμές. Τα ίδια κλειδιά κ.λπ. υπολογίζει ο client.

Δομή του μηνύματος Client Hello

Η γενική μορφή ενός μηνύματος `ClientHello` συμμορφώνεται με τη γενική λογική ότι συγκεκριμένες επιλογές περιγράφονται από συγκεκριμένα πεδία. Ορισμένα εξ αυτών έχουν σταθερό μήκος (π.χ. δύο bytes), ενώ άλλα έχουν μεταβλητό μήκος. Τα πεδία μεταβλητού μήκους στην αρχή τους εκτός από μια δήλωση τύπου περιλαμβάνουν και ένα πεδίο μήκους. Η δομή του μηνύματος εξηγείται με ένα παράδειγμα.

```
0040   49 64 16 03 01 02 00 01 00 01 fc 03 03 32 aa 64
0050   da 48 75 2f d0 42 17 c1 a5 d4 51 38 08 1e df 11
0060   d5 42 cf 6f 72 59 e8 fe 15 8e 90 92 5d 20 bf 94
0070   d6 2f 32 45 90 9b a3 4e 58 f7 aa cb 81 28 19 68
0080   0b 4d 1f 7d fd df 69 a4 e6 82 4e 54 4c 07 00 36
0090   fa fa 13 01 13 02 13 03 c0 2c c0 2b cc a9 c0 30
00a0   c0 2f cc a8 c0 24 c0 23 c0 0a c0 09 c0 28 c0 27
00b0   c0 14 c0 13 00 9d 00 9c 00 3d 00 3c 00 35 00 2f
00c0   c0 08 c0 12 00 0a 01 00 01 7d 4a 4a 00 00 00 00
00d0   00 18 00 16 00 00 13 63 6c 69 65 6e 74 73 31 2e
00e0   67 6f 6f 67 6c 65 2e 63 6f 6d 00 17 00 00 ff 01
00f0   00 01 00 00 0a 00 0c 00 0a 4a 4a 00 1d 00 17 00
0100   18 00 19 00 0b 00 02 01 00 00 10 00 0e 00 0c 02
0110   68 32 08 68 74 74 70 2f 31 2e 31 00 05 00 05 01
0120   00 00 00 00 00 0d 00 18 00 16 04 03 08 04 04 01
0130   05 03 02 03 08 05 08 05 05 01 08 06 06 01 02 01
0140   00 12 00 00 00 33 00 2b 00 29 4a 4a 00 01 00 00
0150   1d 00 20 ee db 45 65 a0 f8 10 22 53 81 7f 25 af
0160   d0 f0 f4 bf 67 c9 0b dc c1 4f f0 5c f1 da 3a cb
0170   24 91 22 00 2d 00 02 01 01 00 2b 00 0b 0a 8a 8a
0180   03 04 03 03 03 02 03 01 1a 1a 00 01 00 00 15 00
0190   b6 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Σχήμα 9.3: Hex dump μέρους ενός πακέτου Client Hello.

Παράδειγμα 21. Εδώ δίνεται ένα πραγματικό μήνυμα Client Hello που έχει συλληφθεί με Wireshark. Στο Σχ. 9.3 φαίνεται το αντίστοιχο πακέτο γραμμένο στο δεκαεξαδικό σύστημα, οπότε κάθε δύο ψηφία είναι ένα byte. Το πακέτο φαίνεται από το byte 0x41 (=65 δεκαδικό) και πέρα (η αριστερή στήλη χρησιμεύει στην αρίθμηση κάθε byte, δηλ. στη σειρά 0x40 υπάρχουν τα bytes από 0x41 (=65 δεκαδικό) ως 0x50 (=80)). Το μήνυμα του πρωτοκόλλου TLS αρχίζει από το byte 0x43. Τα πεδία που του μηνύματος έχουν ως εξής:

0x43 είναι ένα byte ίσο με 0x16 (=22 - Handshake) και δηλώνει ότι πρόκειται για μήνυμα χειραψίας.

0x44, 0x45 δύο bytes ίσα με 03 01 με το οποίο δηλώνεται ότι η κατώτερη έκδοση που θα γίνει δεκτή είναι η SSL 3.1 = TLS 1.0 (δηλ. δηλώνονται ως εκδόσεις με αρίθμηση SSL, το TLS 1.1 θα δηλωθεί ως 03 02 κ.ο.κ.).

0x46, 0x47 δύο bytes που δηλώνουν το μήκος του υπόλοιπου μηνύματος και είναι ίσα με 0x0200 ($= 2 \times 16^2 = 512$), άρα το μήκος είναι ίσο με 512 bytes.

0x48 ένα byte με τιμή 01 που δηλώνει ότι πρόκειται για Client Hello.

0x49-0x4b τρία bytes μήκους του υπόλοιπου μηνύματος με την τιμή 0x0001fc ($= 16^2 + 15 \times 16 + 12 = 508$).

0x4c, 0x4d δύο bytes ίσα με 03 03 με τα οποία δηλώνεται ότι η έκδοση του client είναι TLS 1.2, όμως στη συνέχεια η επιλογή του client δηλώνεται πιο κάτω με την επέκταση *Supported Versions*.

0x4e, 0x6d Ακολουθούν 32 bytes που αποτελούν ένα τυχαίο nonce: `0x32aa64da48752fd0-4217c1a5d45138081edf11d542cf6f7259e8fe158e90925d`.

6e ένα byte που δηλώνει το μήκος του επόμενου πεδίου που είναι το Session ID ως ίσο με 0x20 = 32 bytes.

0x6f-0x8e 32 bytes με το Session ID `00xbf94d62f3245909ba34e58f7aacb812819680b4d-1f7dfddf69a4e6824e544c07`.

0x8f, 0x90 δύο bytes με το μήκος του επόμενου πεδίου των υποστηριζομένων από τον client μεθόδων κρυπτογράφησης. Το μήκος δηλώνεται ίσο με 54, άρα 27 σουίτες.

0x91-0xc6 Παρατίθενται στο μήνυμα οι 27 κωδικοί για τις υποστηριζόμενες σουίτες, όπως στον Πίνακα 9.1.

0xc7 Μήκος του πεδίου με τις μεθόδους συμπίεσης ίσο με 1 byte (επειδή δεν θα παρατεθούν μέθοδοι συμπίεσης).

0xc8 ένα byte ίσο με 00 που σημαίνει την απουσία συμπίεσης.

0xc9, 0xca Δύο bytes με το μήκος των επεκτάσεων (extensions) που θα ακολουθήσουν: Το μήκος δηλώνεται ίσο με 0x017d ήτοι 381 (bytes).

0xcb, 0xcc Ο τύπος αυτής της επέκτασης είναι 0x4a4a = 19018 που αφορά στο GREASE[3].

0xcd, 0xce Το μήκος της επέκτασης δηλώνεται μηδενικό.

0xcf, 0xd0 Ο τύπος της επόμενης επέκτασης θα είναι το όνομα του server (00).

0xd1, 0xd2 Το μήκος του επόμενου πεδίου με ονόματα server θα είναι 0x0018, ήτοι 24.

0xd3, 0xd4 δύο bytes ίσα με 0x0016 = 22 που δίνουν το μήκος για τον server που θα δηλωθεί.

0xd5 ένα byte για τον τρόπο περιγραφής του server, ως `host_name`.

0xd6, 0xd7 δύο bytes ίσα με 0x0013 = 19 που δηλώνουν το μήκος του ονόματος του server σε χαρακτήρες.

0xd8 - 0xea 19 bytes με το όνομα του server σε κωδικοποίηση UTF8: `63 6c 69 65 6e 74 73 31 2e 67 6f 6f 67 6c 65 2e 63 6f 6d`. Άρα το όνομα είναι `clients1.google.com`.

0xeb - 0xed Το πεδίο για το master secret (κωδ. 0x0017) δηλώνεται μηδενικό (στο byte 0xed).

0xee - 0xf3 Το πεδίο για επαναδιαπραγμάτευση (κωδ. 0xff01=65281) δηλώνεται κενό.

0xf4 - 0xf5 Τύπος επόμενου πεδίου: Supported groups (κωδ. 0x000a).

0xf6 - 0xf7 Μήκος ίσο με 0x000c = 12 bytes.

0xfa - 0x103 Υποστηρίζονται 4 αλγόριθμοι (με κωδ. των 2 bytes έκαστος): Reserved (GREASE) (0x4a4a), x25519 (0x001d), secp256r1 (0x0017), secp384r1 (0x0018), secp521r1 (0x0019).

0x104 - 0x109 Προσδιορίζεται το format για τις ελλειπτικές καμπύλες.

0x125 - 0x140 Προσδιορίζονται 11 αλγόριθμοι κατακερματισμού για υπογραφή (signature hash algorithms), μετά τα σχετικά πεδία τύπου και μήκους:

```
ecdsa_secp256r1_sha256   (0x0403)
rsa_pss_rsae_sha256      (0x0804)
rsa_pkcs1_sha256         (0x0401)
ecdsa_secp384r1_sha384   (0x0503)
ecdsa_sha1               (0x0203)
rsa_pss_rsae_sha384      (0x0805)
rsa_pss_rsae_sha384      (0x0805)
rsa_pkcs1_sha384         (0x0501)
rsa_pss_rsae_sha512      (0x0806)
rsa_pkcs1_sha512         (0x0601)
rsa_pkcs1_sha1           (0x0201)
```

0x141 - 0x144 Ακολουθεί μια κενή επέκταση για χρονοσφραγίδα (τύπος 0x0012).

0x145 - 0x146 Ο τύπος του επόμενου πεδίου είναι το δημόσιο κλειδί του client (0x0033) για έναν αλγόριθμο που πιθανώς υποστηρίζει ο server. Αυτό σημαίνει ότι ο server μπορεί μετά και το Server Hello να συνεχίσει με κρυπτογραφημένα μηνύματα προς τον client.

0x147 - 0x148 Το μήκος του πεδίου των κλειδιών είναι 0x002b = 43 bytes.

0x149 - 0x14a Το μήκος των κλειδιών είναι 0x0029=41

[3] Βλ. RFC 8701 - Ιαν. 2020, `https://tools.ietf.org/html/rfc8701`.

0x14b - 0x14f Το πρώτο πεδίο κλειδιού αφορά σε GREASE (0x4a4a) με μήκος 1 και τιμή 0 (δεν υπάρχει κλειδί).

0x14f - 0x151 Το δεύτερο κλειδί αφορά στον αλγόριθμο x25519 (κωδ. 0x001d).

0x152 - 0x153 Το εν λόγω κλειδί θα έχει μήκος 0x0020 = 32.

0x154 - 0x173 Το κλειδί είναι `eedb4565a0f8102253817f25afd0f0f4bf67c90bdcc14ff0-5cf1da3acb249122`.

0x174 - 0x175 Θα περιγραφούν οι τρόποι ανταλλαγής προδιαμοιρασμένων κλειδιών (preshared key exchange modes) (τύπος 0x002d).

0x176 - 0x177 Το μήκος του επόμενου πεδίου είναι 2.

0x178 Το μήκος του πεδίου του preshared key exchange mode είναι 1.

0x179 Περιέχει τον κωδικό 0x01 για την ανταλλαγή με EC-DHE.

0x17a - 0x188 Περιέχει τις υποστηριζόμενες εκδόσεις του TLS. Μετά τον τύπο και το μήκος αρχίζοντας από το byte 0x181 δίνονται τα TLS 1.3, 1.2, 1.1, 1.0.

0x189 - 0x18d Reserved GREASE extension.

0x18e - τέλος 182 μηδενικά bytes για padding.

σουίτα	κωδικός
Reserved (GREASE)	0xfafa
TLS_AES_128_GCM_SHA256	0x1301
TLS_AES_256_GCM_SHA384	0x1302
TLS_CHACHA20_POLY1305_SHA256	0x1303
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	0xc02c
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	0xc02b
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	0xcca9
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	0xc030
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	0xc02f
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	0xcca8
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	0xc024
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	0xc023
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	0xc00a
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	0xc009
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	0xc028
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	0xc027
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	0xc014
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	0xc013
TLS_RSA_WITH_AES_256_GCM_SHA384	0x009d
TLS_RSA_WITH_AES_128_GCM_SHA256	0x009c
TLS_RSA_WITH_AES_256_CBC_SHA256	0x003d
TLS_RSA_WITH_AES_128_CBC_SHA256	0x003c
TLS_RSA_WITH_AES_256_CBC_SHA	0x0035
TLS_RSA_WITH_AES_128_CBC_SHA	0x002f
TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	0xc008
TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	0xc012
TLS_RSA_WITH_3DES_EDE_CBC_SHA	0x000a

Πίνακας 9.1: Οι σουίτες που εμφανίζονται στο Παράδειγμα 21.

9.2 HTTPS

Το *HTTPS* (Hypertext Transfer Protocol Secure) ή *HTTP πάνω από TLS* διευκολύνει την ασφαλή ανταλλαγή δεδομένων ενός browser (βλ. και Σχ. 9.1). Προσδιορίζεται μέσω της RFC 2818 (Μάιος 2000).

Το HTTPS διακινεί δεδομένα HTTP πάνω από TLS και είναι κρυπτογραφημένα. Η κρυπτογράφηση προστατεύει και το URL της ιστοσελίδας που ζητάει ο χρήστης, επικεφαλίδες, παραμέτρους και cookies. Ωστόσο οι διευθύνσεις προέλευσης και προορισμού και οι αριθμοί θυρών βρίσκονται στις επικεφαλίδες των πακέτων, έξω από την κρυπτογράφηση. Κατά συνέπεια ο επιτιθέμενος εκτός της ανάλυσης κίνησης (όγκος, διάρκεια) έχει στη διάθεσή του πηγή, προορισμό, θύρα και μπορεί πιθανώς να συναγάγει το όνομα πεδίου (domain name) χωρίς το λοιπό URL (π.χ. `www.ntua.gr`, αλλά όχι το `https://www.ntua.gr/el/schools`).

Η RFC 2818 είναι πολύ περιορισμένου μήκους ακριβώς διότι βασίζεται στο TLS. Εξηγεί ότι γίνεται χρήση του HTTP πάνω από TLS ακριβώς όπως θα γινόταν πάνω από TCP. Για το HTTPS χρησιμοποιείται (στον server) η θύρα 443.

9.3 SSH - Secure Shell

Το SSH είναι ένα σχετικά απλό πρωτόκολλο που παρέχει δυνατότητες απομακρυσμένης πρόσβασης σε υπολογιστή (remote login), η οποία υποστηρίζεται χωρίς ασφάλεια από το telnet κ.α. παρόμοια πρωτόκολλα. Παρέχει επίσης μεταφορά αρχείων και email. Η έκδοση 2 του SSH (SSH2) περιγράφεται στα RFC 4250 ως 4256.

Ο απομακρυσμένος προορισμός μπορεί να δοθεί είτε με τη μορφή ενός ονόματος χρήστη πάνω σ᾽ ένα host, δηλαδή ως `[user@]hostname`, είτε με ένα URI[4] της μορφής `ssh://[user@]hostname[:port]`. Το SSH αποτελείται από τρία πρωτόκολλα, ως

SSH User Authentication Protocol	SSH Connection Protocol
SSH Transport Layer Protocol	
TCP	
IP	

Σχήμα 9.4: Στοίβα πρωτοκόλλων SSH

εξής:

Πρωτόκολλο SSH Στρώματος Μεταφοράς - SSH Transport Layer Protocol το οποίο βρίσκεται πάνω από το TCP και υποστηρίζει αυθεντικοποίηση του server, εμπιστευτικότητα και ακεραιότητα των δεδομένων και προαιρετικά συμπίεση.

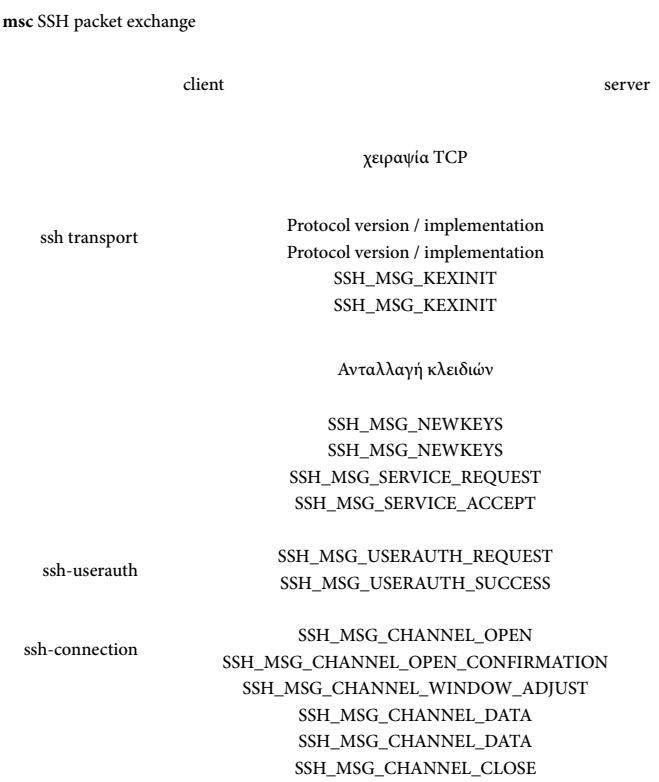
Πρωτόκολλο SSH Αυθεντικοποίησης Χρήστη - SSH User Authentication Protocol το οποίο βρίσκεται πάνω από το SSH Transport Layer Protocol και υλοποιεί την αυθεντικοποίηση του χρήστη προς τον server.

Πρωτόκολλο Σύνδεσης - SSH Connection Protocol το οποίο επίσης βρίσκεται πάνω από το SSH Transport Layer Protocol και υλοποιεί την πολυπλεξία λογικών καναλιών επικοινωνίας πάνω από μια σύνδεση SSH.

[4] Το Uniform Resource Identifier (URI) είναι μια μοναδική ακολουθία χαρακτήρων που προσδιορίζει έναν λογικό ή φυσικό πόρο που χρησιμοποιείται από τεχνολογίες ιστού.

Το Πρωτόκολλο SSH Στρώματος Μεταφοράς ασχολείται με την αυθεντικοποίηση του server προς τον client, προκειμένου ο τελευταίος να γνωρίζει ότι μιλάει με τον σωστό server. Για να το κάνει αυτό ο server χρησιμοποιεί το ιδιωτικό του κλειδί, αλλά ο client πρέπει να γνωρίζει το αντίστοιχο δημόσιο κλειδί. Πώς θα συμβεί αυτό; Χρησιμοποιούνται δύο τρόποι: (α) Ο client μπορεί να διατηρεί μια βάση δεδομένων με τα δημόσια κλειδιά των server με τους οποίους επικοινωνεί και (β) απευθύνεται σε μια αρχή πιστοποίησης που πιστοποιεί όλους τους servers, ενώ ο client αρκεί να ξέρει μόνο το κλειδί της αρχής.

Μια ανταλλαγή πακέτων SSH στην υλοποίηση OPENSSH [Ven07] φαίνεται στο Σχ. 9.5.



Σχήμα 9.5: Ανταλλαγή μηνυμάτων SSH

Μετά από μια ανταλλαγή μηνυμάτων για έκδοση του πρωτοκόλλου που στηρίζουν οι δύο μηχανές (βλ. παρ. 4.2 στην RFC 4253) ακολουθούν μηνύματα για την ανταλλαγή κλειδιών. Αυτή μπορεί να αρχίσει με ένα μήνυμα `SSH_MSG_KEX_INIT` το οποίο μπορεί να στείλει οποιαδήποτε από τις δυο πλευρές, μαζί με ένα κατάλογο μεθόδων που υποστηρίζει (RFC 4253). Κάθε πλευρά έχει ένα προτιμώμενο αλγόριθμο κι εδώ γίνεται η υπόθεση ότι το πιο πιθανό είναι να έχουν την ίδια προτίμηση και οι δύο πλευρές. Ο πρώτος αλγόριθμος στη σειρά είναι αυτός που προτιμάει η πλευρά που στέλνει το μήνυμα και ταυτόχρονα μαντεύει πως θα έχει την ίδια προτίμηση η άλλη πλευρά. Αν και οι δύο πλευρές έχουν εκδηλώσει την ίδια προτίμηση υιοθετούν αυτόν τον αλ-

γόριθμο, διαφορετικά διαλέγουν κάποιον από τους επόμενους στη σειρά που είναι κοινός στις προτιμήσεις τους. Αν δεν υπάρχει κοινός αλγόριθμος η διαπραγμάτευση προφανώς αποτυγχάνει. Οι αλγόριθμοι αυτοί αφορούν (α) στην ανταλλαγή κλειδιών, (β) στην κρυπτογράφηση των δεδομένων (με δυνατότητα διαφορετικής επιλογής ανά κατεύθυνση), (γ) στην παραγωγή MAC (Message Authentication Code) ανά κατεύθυνση, (δ) στην επιλογή ενδεχόμενης συμπίεσης ανά κατεύθυνση.

Η συμπίεση είναι προαιρετική. Αν προτιμηθεί, η RFC 4253 ορίζει μόνο τον Lempel Ziv '77 (LZ77 ή `zlib`, περιγράφεται στις RFC1950 και RFC1951).

Η κρυπτογράφηση εφαρμόζεται όχι μόνο πάνω στο φορτίο, αλλά και στα πεδία του μήκους πακέτου, μήκους παραγεμίσματος (padding) και στο ίδιο το παραγέμισμα. Ο αλγόριθμος μπορεί να είναι διαφορετικός ανά κατεύθυνση, αλλά συνιστάται (στην RFC 4253) να υιοθετείται ο ίδιος αλγόριθμος. Ως προς τις διαθέσιμες επιλογές, περιγράφονται 16 αλγόριθμοι, εκ των οποίων ο `3des-cbc` είναι ο μόνος υποχρεωτικός. Αυτός είναι ο τριπλός DES με τριπλό κλειδί μήκους 3×8 bytes. Ο πλήρης κατάλογος δίνεται στην παρ. 6.3 της σύστασης και περιλαμβάνει αλγορίθμους Blowfish, AES, serpent κ.α.

Η ακεραιότητα υποστηρίζεται με τη χρήση ενός MAC, ο οποίος υπολογίζεται πάνω στο (συμπιεσμένο ή ασυμπίεστο) μη κρυπτογραφημένο μήνυμα, του οποίου όμως προηγούνται δύο strings, το κλειδί (που έχει προσδιορισθεί για τη χρήση του αλγόριθμου MAC) και ο αύξων αριθμός (sequence number). Μπορούν να χρησιμοποιηθούν 4 αλγόριθμοι (βλ. παρ. 6.4 στην σύσταση), εκ των οποίων υποχρεωτικός είναι ο HMAC-SHA1 με μήκος κλειδιού και κωδικού 20 bytes, ενώ είναι συνιστώμενη και η περίπτωση όπου χρησιμοποιούνται μόνο τα πρώτα 12 bytes για τον κωδικό. Η συλλογή συμπληρώνεται με δύο αλγόριθμους HMAC-MD5.

Η ανταλλαγή κλειδιών (προκειμένου να καθοριστούν τα κλειδιά κρυπτογράφησης και αυθεντικοποίησης και να γίνει η αυθεντικοποίηση του server) γίνεται με δύο υποχρεωτικά περιλαμβανόμενες μεθόδους, που αμφότερες βασίζονται στην ανταλλαγή Diffie-Hellman σε συνδυασμό με SHA1.

Η ανταλλαγή κλειδιών γίνεται με τα μηνύματα `SSH_MSG_KEXINIT`. Ο client στέλνει με ένα μήνυμα `SSH_MSG_KEXDH_INIT` ένα μεγάλο ακέραιο e. Ο αριθμός αυτός είναι το δημόσιο κλειδί του που έχει υπολογίσει μέσω της σχέσης Diffie - Hellmann, αφού προηγουμένως έχει επιλέξει ένα ιδιωτικό κλειδί. Ο server απαντάει με ένα μήνυμα `SSH_MSG_KEXDH_REPLY` και στέλνει το δημόσιο κλειδί f που έχει υπολογίσει με τον ίδιο τρόπο επιλέγοντας προηγουμένως ένα ιδιωτικό κλειδί. Στο ίδιο μήνυμα περιλαμβάνονται το δημόσιο κλειδί του server και πιστοποιητικά. Επίσης περιλαμβάνεται η υπογραφή ενός hash H που έχει υπολογισθεί πάνω σε ένα μακρύ string που αποτελείται από τα ονόματα του client και του server, το (μόνιμο) δημόσιο κλειδί του server, τα φορτία από τα προηγούμενα δύο μηνύματα που έχουν ανταλλάξει, τα δύο δημόσια κλειδιά e, f και το κοινό κλειδί K στο οποίο έχει καταλήξει ο server με τη μέθοδο Diffie-Hellmann. Η ίδια η τιμή του H διατηρείται μυστική.

Η υπογραφή του H δημιουργείται σύμφωνα με τον αλγόριθμο κατακερματισμού που έχουν επιλέξει οι δύο πλευρές, αλλά μπορεί να γίνει και ενδιάμεση προσαύξηση (padding), δηλαδή να γίνει πρώτα κατακερματισμός, κατόπιν προσαύξηση και κατόπιν δεύτερος κατακερματισμός με τον SHA-1.

Μετά από κάθε Gbyte δεδομένων ή μια ώρα λειτουργίας της σύνδεσης είναι καλό να γίνεται νέα ανταλλαγή κλειδιών.

Μετά την ανταλλαγή κλειδιών αρχίζει η κρυπτογραφημένη επικοινωνία, για την οποία χρειάζονται τα εξής "κλειδιά" που δημιουργούνται από το K:

- Αρχική τιμή `IV` από client προς server: HASH($K \| H \|$"A"$\|$session_id), όπου "A"

είναι ο χαρακτήρας A σε μορφή ASCII (ASCII 65) και HASH είναι η επιλεγμένη συνάρτηση κατακερματισμού.

- Αρχική τιμή `IV` από server προς client: HASH($K \| H \|$"B"$\|$session_id).
- Κλειδί κρυπτογράφησης από client προς server: HASH($K \| H \|$"C"$\|$session_id).
- Κλειδί κρυπτογράφησης από server προς client: HASH($K \| H \|$"D"$\|$session_id).
- Κλειδί ακεραιότητας από client προς server: HASH($K \| H \|$"E"$\|$session_id).
- Κλειδί ακεραιότητας από server προς client: HASH($K \| H \|$"F"$\|$session_id).

Στην αρχή της κρυπτογραφημένης επικοινωνίας ο client ζητάει κάποια υπηρεσία με ένα μήνυμα `SSH_MSG_SERVICE_REQUEST`. Οι δυνατές υπηρεσίες κατά την RFC 4253 είναι είτε αυθεντικοποίηση του χρήστη είτε σύνδεση (`ssh-connection`) με ένα μήνυμα `SSH_MSG_CHANNEL_OPEN`. Στο μήνυμα αυτό περιλαμβάνονται παράμετροι που αφορούν στο μέγεθος πακέτου, τον τύπο καναλιού (που προσδιορίζει ποια υπηρεσία θα χρησιμοποιηθεί στο SSH) κ.α. Εφόσον το κανάλι ανοίξει αρχίζει η ανταλλαγή δεδομένων με μηνύματα `SSH_MSG_ CHANNEL_DATA` μέχρι να εμφανισθεί από οποιαδήποτε πλευρά ένα μήνυμα `SSH_MSG_ CHANNEL_CLOSE`.

ΚΕΦΑΛΑΙΟ 10

Ασφάλεια λειτουργικού συστήματος

10.1 Εισαγωγή

Ένα από τα πιο παλιά μαθήματα της επιστήμης των υπολογιστών που διατηρείται ως σήμερα είναι τα *λειτουργικά συστήματα* (operating systems). Στο κλασσικό βιβλίο της δεκαετίας του 1970 του Per Brinch Hansen /citehansen1973operating η μελέτη των λειτουργικών συστημάτων περιλαμβάνει ακολουθιακές (sequential) και παράλληλες διεργασίες (concurrent processes), αλγόριθμους χρονοπρογραμματισμού (scheduling algorithms) και διαχείριση επεξεργαστών και μνήμης. Περιλαμβάνει επίσης υπό τον τίτλο της *προστασίας πόρων* (resource protection) στοιχειώδη ζητήματα ασφάλειας, όπως την ταυτοποίηση χρηστών και την απονομή δικαιωμάτων σ' αυτούς.

Λίγο παραπάνω από μια δεκαετία αργότερα, το 1987, ο A. Tanenbaum στην πρώτη έκδοση του βιβλίου του για λειτουργικά συστήματα[Tan87] έχει προσθέσει δύο κεφάλαια για ασφάλεια και προστασία (Κεφ. 5 & 6). Ο ψυχρός πόλεμος δεν έχει τελειώσει και ο Tanenbaum θέλοντας να οριοθετήσει το φάσμα της επικινδυνότητας των πιθανών επιτιθέμενων στο ένα άκρο βάζει φοιτητές και στο άλλο την KGB (αν και στο κεφάλαιο για το κακόβουλο λογισμικό έχουμε δει ότι πολύ μεγάλες ζημιές έχουν προκληθεί από ιούς φοιτητών και μαθητών). Επίσης κάνει μια αναφορά στην ιδιωτικότητα και στα όρια που πρέπει να τίθενται στο κράτος και στις εταιρίες όταν διαχειρίζονται τα δεδομένα πολιτών και πελατών. Στη συνέχεια παραθέτει μια σειρά από τρωτότητες μαζί με τις αντίστοιχες επιθέσεις.

Σε ένα άλλο δημοφιλές βιβλίο λειτουργικών συστημάτων, των Peterson & Silberschatz με πρώτη έκδοση το 1985 [PS85], που έχει φτάσει στη 10η έκδοση ως [SGG18] υπάρχουν επίσης δυο κεφάλαια για ασφάλεια και προστασία του συστήματος, τόσο του λογισμικού και των δεδομένων που φιλοξενούνται στον υπολογιστή, όσο και του hardware. Ο βασικός μηχανισμός προστασίας αφορά στο να εξασφαλίζεται η πρόσβαση στο λειτουργικό σύστημα και τους λοιπούς πόρους του συστήματος εγκεκριμένων χρηστών και μόνο.

Οι πιθανές παραβιάσεις ανήκουν στους γνωστούς τύπους παραβιάσεων που έχουν ήδη εκτεθεί σε άλλα κεφάλαια του παρόντος βιβλίου, όπως παραβίαση της εμπιστευτικότητας των δεδομένων, παραβίαση της ακεραιότητας δεδομένων και λογισμικού,

παρεμπόδιση της διαθεσιμότητάς τους, χρήση πόρων από μη εξουσιοδοτημένους χρήστες κ.λπ. Ο πιο συνηθισμένος τρόπος επίθεσης είναι ο εισβολέας να παριστάνει ότι είναι κάποιος άλλος (με περισσότερα δικαιώματα), π.χ. ότι είναι διαχειριστής του συστήματος, αλλά για να φτάσει εκεί μπορεί να χρησιμοποιήσει οποιαδήποτε από τις γνωστές τεχνικές μεθόδους ή και με κοινωνική μηχανική. Αν ο επιτιθέμενος έχει λιγότερο φιλόδοξους στόχους μπορεί φυσικά να αρκεστεί σε μια επίθεση άρνησης υπηρεσίας.

Η πλευρά της άμυνας μπορεί να πάρει μια σειρά μέτρων *άμυνας σε βάθος* (βλ. και σελ. 133), όπως:

- Μέτρα φυσικής προστασίας, για παράδειγμα περιορίζοντας την φυσική πρόσβαση μόνο σε εξουσιοδοτημένα άτομα στους χώρους όπου βρίσκονται οι μηχανές.

- Μέτρα σχετικά με τη δικτυακή πρόσβαση.

- Μέτρα προστασίας του λειτουργικού συστήματος από μη εξουσιοδοτημένη πρόσβαση, από αλλοιώσεις κ.λπ. Στην περίπτωση μηχανών με ειδικούς ρόλους, π.χ. που φιλοξενούν συστήματα ανίχνευσης και αναχαίτισης επιθέσεων, το λειτουργικό σύστημα μπορεί να είναι πιο «σκληρό» περιορίζοντας το σύνολο των εντολών και χρησιμοποιώντας πιο προσεκτικό σχεδιασμό.

- Με έλεγχο των εφαρμογών που μπορούν να εγκατασταθούν στο σύστημα, θέτοντας περιορισμούς, κάνοντας έλεγχο της συμπεριφοράς τους πριν την κανονική εγκατάσταση (sandboxing) κ.λπ.

10.2 Γενικές αρχές

Η πλευρά της άμυνας μπορεί επίσης να υιοθετεί ορισμένες αρχές που συμβάλλουν στη μείωση των κινδύνων εισβολής, όπως:

Απονομή των ελάχιστων δυνατών δικαιωμάτων σε χρήστες και προγράμματα. Η αρχή αυτή έχει την έννοια ότι δίνονται τα ελάχιστα δυνατά δικαιώματα που χρειάζονται για την επίτευξη των σκοπών του χρήστη ή του προγράμματος.

Διαμερισματοποίηση (compartmentalization) δηλαδή διαχωρισμός του συστήματος σε τμήματα ώστε να δίνονται δικαιώματα μόνο πάνω σε κάθε τμήμα χωριστά.

Ελαχιστοποίηση του συστήματος με εγκατάσταση μόνο των συνιστωσών του συστήματος και περαιτέρω των εφαρμογών που χρειάζονται για να εξυπηρετούνται οι σκοποί του χρήστη.

Η αρχή της διαμερισματοποίησης περιορίζει το πεδίο των απωλειών σε ένα υποσύνολο του συστήματος σε περίπτωση επίθεσης. Χρησιμοποιείται με διάφορους τρόπους. Μια περίπτωση χρήσης αυτής της αρχής είναι ο καθορισμός *αποστρατικοποιημένων ζωνών* (demilitarized zones), δηλαδή περιοχών ειδικού σκοπού ανάμεσα σε τείχη προστασίας. Μια άλλη περίπτωση είναι η χρήση εικονικών μηχανών που αποδίδονται σε διαφορετικούς χρήστες στο ίδιο σύστημα.

Δικαιώματα - μοντέλο Bell-LaPadula

Ένα συνηθισμένο παράδειγμα χρήσης της αρχής απονομής των ελάχιστων δυνατών δικαιωμάτων είναι ο κάτοχος ενός προσωπικού υπολογιστή να μπαίνει ως απλός χρήστης όταν κάνει συνήθη χρήση. Όταν πρέπει να κάνει εγκατάσταση λογισμικού μπορεί κατ' εξαίρεση να μπαίνει (με διαφορετικά συνθηματικά) ως διαχειριστής (administrator). Με τον τρόπο αυτόν ο ίδιος χρήστης εξασφαλίζεται ακόμη και από τα δικά του λάθη, π.χ. να διαγράψει ένα κρίσιμο αρχείο του συστήματος. Εξασφαλίζεται εν μέρει επίσης από εισβολείς, π.χ. όταν ένα κακόβουλο πρόγραμμα μιμηθεί την διαδικασία login, ο εξαπατημένος χρήστης είναι πιθανό να δώσει μόνο τα συνθηματικά απλού χρήστη.

Προκειμένου να διευκολυνθεί η απονομή των κατάλληλων δικαιωμάτων μπορεί να χρησιμοποιηθεί το *μοντέλο Bell-LaPadula*. Το μοντέλο αυτό εισήγαγαν στο υπουργείο άμυνας των ΗΠΑ οι David Elliott Bell και Leonard J. LaPadula [LaP96] το 1973. Περιγράφει ένα σύνολο κανόνων ελέγχου πρόσβασης σύμφωνα με διαβαθμίσεις ασφάλειας που έχουν απονεμηθεί σε αντικείμενα και περιοχές. Οι διαβαθμίσεις κυμαίνονται από τις πιο ευαίσθητες (π.χ. *άκρως απόρρητο*) έως τις ελάχιστα ευαίσθητες (π.χ. *δημόσιο*). Ένας χρήστης του συστήματος μπορεί να έχει δικαιώματα πάνω σε αντικείμενα μιας κατηγορίας και κάτω, π.χ. σε *απόρρητα* και κάτω, αλλά όχι σε αντικείμενα ανώτερης κατηγορίας όπως *άκρως απόρρητα*.

Οι κατηγορίες του μοντέλου Bell-LaPadula συχνά παριστάνονται με ένα σχήμα ομόκεντρων κύκλων, όπου κάθε πιο στενή (ή ανώτερη) κατηγορία είναι μέσα στην προηγούμενη. Με άλλα λόγια σχηματίζουν μια κατακόρυφη ιεραρχία, όπου η πιο στενή περιοχή ανήκει στο ανώτερο επίπεδο, με την έννοια ότι όποιος έχει πρόσβαση σε ένα επίπεδο έχει πρόσβαση και σε όλα τα κατώτερα.

Αντί της ιεραρχίας μπορεί να υιοθετηθεί ένα πιο γενικό μοντέλο, όπου κάθε πόρος του υπολογιστή, που μπορεί να θεωρηθεί ως αντικείμενο (object), αντιστοιχίζεται σε ένα χρήστη ή γενικότερα μια οντότητα (άνθρωπο ή πρόγραμμα) με συγκεκριμένα δικαιώματα και δυνατότητες. Η περιγραφή των δικαιωμάτων κάθε οντότητας πάνω σε αντικείμενα και πόρους μπορεί να γίνει με διάφορους τρόπους, π.χ. με ένα πίνακα που περιγράφει ένα συγκεκριμένο δικαίωμα για κάθε ζεύγος οντότητας-αντικειμένου. Μπορούν επίσης να γίνουν ομαδοποιήσεις ώστε να αποφευχθούν πολύ μεγάλοι πίνακες, π.χ. να ομαδοποιηθούν τα αντικείμενα σε πεδία με κοινή απονομή δικαιωμάτων. Επίσης οι χρήστες μπορούν να τυποποιηθούν σε *ρόλους* (π.χ. διαχειριστή ή απλού χρήστη) και σε ένα χρήστη να αποδοθούν ένας ή περισσότεροι ρόλοι.

Σκλήρυνση και ελαχιστοποίηση

Ως προς την ελαχιστοποίηση του συστήματος, αυτή μπορεί να πάρει πολλές μορφές. Μπορούν να αφαιρεθούν από το λειτουργικό οι εντολές που είναι περιττές ή επικίνδυνες. Μια τέτοια αφαίρεση αφενός περιορίζει τις δυνατότητες ενός επιτιθέμενου, αφετέρου διευκολύνει τον αρχικό του σχεδιασμό και έλεγχο. Μπορούν επίσης να αποφεύγονται εφαρμογές ή τμήματά τους που δεν πρόκειται να χρησιμοποιηθούν. Για παράδειγμα, στις *ελληνικές υποκλοπές* μετά τους Ολυμπιακούς Αγώνες του 2004 βασικό ρόλο έπαιξε η εγκατάσταση για τυχόν μελλοντική χρήση ενός συστήματος νόμιμων παρακολουθήσεων, το οποίο όμως ο συγκεκριμένος πάροχος κινητής δεν είχε τότε αγοράσει. Το σύστημα αυτό ήταν απενεργοποιημένο μετά την εγκατάσταση, ωστόσο οι εισβολείς κατάφεραν να το ενεργοποιήσουν και να πραγματοποιήσουν παράνομες παρακολουθήσεις.

10.3 Κρυπτογράφηση δίσκου υπολογιστή

Οργάνωση του δίσκου

Η κρυπτογράφηση του «σκληρού δίσκου» (μαγνητικού ή SSD) είναι ένα από τα ζητήματα που συνήθως υπάγονται στην ασφάλεια του λειτουργικού συστήματος του υπολογιστή. Στο βιβλίο λειτουργικών συστημάτων του W. Stallings [Sta18] αναφέρεται μόνο ακροθιγώς στο τέλος του κεφαλαίου της ασφάλειας. Σήμερα η κρυπτογράφηση συνηθίζεται να καλύπτει όλον τον δίσκο, δηλαδή και το λειτουργικό σύστημα και τα δεδομένα.

Ο σκληρός δίσκος αρχικά εμφανίσθηκε στην δεκαετία του 1950 ως ένα ηλεκτρομηχανικό σύστημα που επιτρέπει την εγγραφή δεδομένων (bits) πάνω σε μια μαγνητική επιφάνεια. Η τεχνολογία της μαγνητικής εγγραφής ηχητικών σημάτων ήταν ήδη γνωστή από το 1930, όταν αναπτύχθηκαν συσκευές γνωστές ως μαγνητόφωνα, από την πρώτη τέτοια συσκευή που ονομαζόταν *Magnetophone K1* που ανέπτυξε η γερμανική εταιρία AEG με ταινίες που παρήγαγε η επίσης γερμανική BASF. Το 1951 ο αμερικανικός υπολογιστής *UNIVAC I* εμφανίσθηκε με μονάδα μαγνητικής ταινίας που μπορούσε να αποθηκεύει 128 χαρακτήρες ανά ίντσα και είχε ταχύτητα εγγραφής και ανάγνωσης 12800 χαρακτήρων ανά sec. Η ταινία ανταποκρινόταν στην ανάγκη για αποθήκευση δεδομένων πιο μόνιμη από αυτήν που παρείχαν τα ηλεκτρονικά κυκλώματα, δηλαδή για δεδομένα που θα μείνουν γραμμένα κι αφού σβήσει ο υπολογιστής. Η ταινία είχε όμως ένα βασικό μειονέκτημα, ότι έπρεπε να αναγνωσθεί γραμμικά, δηλαδή για να ανακτηθεί ένα αρχείο μπορούσε στη χειρότερη περίπτωση η κεφαλή να πρέπει να διατρέξει όλη την ταινία.

Οι ταινίες διατηρήθηκαν αργότερα ως μέσο μόνιμης αποθήκευσης αντιγράφων ασφαλείας (backup), αλλά για την είσοδο έξοδο δεδομένων κατά τη λειτουργία ενός υπολογιστή αναπτύχθηκαν οι σκληροί δίσκοι. Ο σκληρός δίσκος ουσιαστικά είναι μια εξέλιξη αφενός της μαγνητικής ταινίας ως προς το μέσο εγγραφής και ανάγνωσης, αφετέρου του δίσκου βινυλίου ως προς τον τρόπο ανάγνωσης (και εγγραφής). Η επινόηση του σκληρού δίσκου βασίζεται στην ιδέα ότι μολονότι στον δίσκο μουσικής η κεφαλή κινείται με συγκεκριμένο γραμμικό τρόπο, αυτή θα μπορούσε να καθοδηγηθεί πολύ γρήγορα σε οποιοδήποτε σημείο του δίσκου σε αντίθεση με την κεφαλή του μαγνητοφώνου. Με την έννοια αυτή ο σκληρός δίσκος έδωσε την δυνατότητα για εύκολη πρόσβαση σε πληροφορίες γραμμένες οπουδήποτε πάνω στο δίσκο (random access). Η επόμενη εξέλιξη στο τέλος της δεκαετίας του '80 ήταν οι *δίσκοι στερεάς κατάστασης* (solid state drives - SSD), δηλαδή συσκευές βασισμένες σε μνήμες flash που έχουν τη δυνατότητα μόνιμης αποθήκευσης της πληροφορίας.

Ένας μαγνητικός δίσκος διαιρείται συνήθως σε μονάδες που λέγονται *τομείς* (sectors). Σε ένα μαγνητικό δίσκο ο τομέας είναι τοξοειδής περιοχή που περιέχει ένα (κατά κανόνα) σταθερό πλήθος από bits. Τυπικά μεγέθη είναι 512 και 4096 bytes. Το μέγεθος αυξάνεται όσο οι δίσκοι αποκτούν μεγαλύτερες πυκνότητες. Σε ένα «δίσκο» στερεάς κατάστασης (solid state drive) το αντίστοιχο ενός τομέα είναι μια *σελίδα* (page), π.χ. 4 kbytes. Περαιτέρω η μονάδα εγγραφής και ανάγνωστης στο δίσκο είναι το *block*. Σε ένα SSD ένα block μπορεί να είναι 128 σελίδες, ήτοι 512 kbytes, σε ένα HDD το μέγεθος του block μπορεί να είναι 512 bytes ή 4 kbytes.

Ένα αρχείο αποτελείται από ένα πλήθος blocks, επομένως και το μέγεθός του είναι πολλαπλάσιο του ενός block (και το ελάχιστου μεγέθους αρχείο (file) είναι ένα block). Τα blocks ενός αρχείου μπορεί να είναι ή να μην είναι συνεχόμενα πάνω στο δίσκο, οπότε χρειάζεται μια δομή για να περιγράφει με ποιο τρόπο συνδέονται τα blocks που συναποτελούν ένα αρχείο. Μια τέτοια δομή είναι ένας «πίνακας κατανομής του αρ-

χείου» (File Allocation Table, FAT) (βλ. και ενότ. 12.7 στο [Sta18]). Γενικά ένας σκληρός δίσκος οργανώνεται από ένα κατάλληλο σύστημα διαχείρισης (FAT, NTFS, HFX κ.α.).

Απαιτήσεις για την κρυπτογράφηση σκληρού δίσκου

Η χρήσιμη πληροφορία από αυτήν την εισαγωγή είναι ότι η μονάδα επεξεργασίας κάθε φορά ανασύρει από τη μόνιμη μνήμη μια ομάδα δεδομένων σταθερού μεγέθους και μάλιστα με αυθαίρετη σειρά. Οι μέθοδοι κρυπτογράφησης ολόκληρου του σκληρού δίσκου βασίζονται στο να κρυπτογραφείται κάθε block χωριστά, ώστε κάθε φορά που πρέπει να κληθεί το επόμενο block, αυτό μπορεί να κληθεί και να αποκρυπτογραφηθεί αυτόνομα, ενώ αντίστοιχη αντίστροφη διαδικασία υλοποιείται όταν πρόκειται να εγγραφεί ένα block. Συγχρόνως όμως αυτό σημαίνει ότι η πληροφορία, κρυπτογραφημένη ή ακρυπτογράφητη, πρέπει να έχει το ίδιο μήκος, ώστε να μην προκύπτουν φαινόμενα υπερχείλισης από ένα τομέα ή ένα block όταν κρυπτογραφείται ο δίσκος.

Η δεύτερη απαίτηση είναι ότι τα δεδομένα του λειτουργικού συστήματος πρέπει επίσης να κρυπτογραφούνται, επομένως δεν υπάρχει διαθέσιμος κώδικας για να εκκινήσει το σύστημα. Για να γίνει η αποκρυπτογράφηση του λειτουργικού και αρχικοποίηση του συστήματος αρχείων πρέπει να υπάρχει ένας ελάχιστος κώδικας που θα εδρεύει αλλού [CGM13].

Επίσης κατά την εκκίνηση του συστήματος πρέπει να προκύπτει το κύριο κλειδί (master key) και μάλιστα με ενδεχομένως περισσότερους από ένα τρόπους, δηλαδή δίνοντας ο χρήστης του συστήματος ένα password ή ένα PIN ή ένα κωδικό άλλου τύπου (π.χ. το *recovery password* που δίνεται στην αρχική κρυπτογράφηση ως εναλλακτικός κωδικός αν χαθεί το password).

Στο *FileVault 2* που χρησιμοποιείται στους υπολογιστές της *Apple* ως μέθοδος κρυπτογράφησης έχει επιλεγεί το AES (Advanced Encryption Standard, βλ. ενότ. 2.7) σε *πειραγμένη* (tweaked) εκδοχή. Κάθε block κρυπτογραφείται με ένα «πειραγμένο» (tweaked) κώδικα χωριστά, έτσι ώστε ακόμη και αν το απλό κείμενο ήταν το ίδιο, το αποτέλεσμα θα ήταν διαφορετικό. Αυτό σημαίνει ότι για κάθε block χρειάζονται δύο κλειδιά (μήκους 128 ή 256), όπου το πρώτο είναι το κανονικό κλειδί και το δεύτερο είναι το tweak key, καθώς και μια τιμή που προκύπτει από τη θέση του block στο δίσκο. Η διαδικασία της κρυπτογράφησης περιγράφεται περαιτέρω στο [CGM13].

Οι *πειραγμένοι block κώδικες* (tweaked block codes) προτάθηκαν από τους M. Liskov, R. L. Rivest και D. Wagner το 2002 [LRW02]. Το *FileVault 2* βασίζεται για την κατασκευή του πειραγμένου κώδικα στην κατασκευή XEX (Xor-Encrypt-Xor) [Rog04], η οποία τροποποιεί το AES. Το αποτέλεσμα αυτής της τροποποίησης είναι γνωστό ως XTS (που σημαίνει *XEX-based Tweaked code-book mode with ciphertext Stealing*) [Mar10] και έχει υιοθετηθεί τον Ιαν. 2010 από το NIST ως *NIST Special Publication 800-38E* [Dwo+10]. Η *κλοπή κρυπτογραφημένου κειμένου* (ciphertext stealing) αναφέρεται στον χειρισμό block του δίσκου με μέγεθος που δεν είναι πολλαπλάσιο του μεγέθους του AES block. Αυτό γίνεται χωρίς παραγέμισμα (padding), παίρνοντας («κλέβοντας») ένα μέρος του κρυπτογραφημένου κειμένου του προτελευταίου block και τοποθετώντας το στο τελευταίο block απλού κειμένου. Το συμπληρωμένο με αυτόν τον τρόπο τελικό block κρυπτογραφείται στη συνέχεια ως συνήθως (βλ. και Άσκ. 20.12 στο [SB18]).

10.4 Εικονικές μηχανές

Ένας υπολογιστής μπορεί να εξομοιώσει τη λειτουργία ενός άλλου υπολογιστή εξ αιτίας της *ισοδυναμίας Turing* [Tur37; Mac18]. Αν είναι εξοπλισμένος με κατάλληλες διεπαφές και περιφερειακά μπορεί να εξομοιώσει γενικότερα ένα άλλο ηλεκτρομηχανικό σύστημα, ως παράδειγμα θα μπορούσε να θεωρηθεί ένα ηλεκτρικό αυτοκίνητο. Το ηλεκτρικό αυτοκίνητο ουσιαστικά είναι 4 διεπαφές με κατ' αρχήν ανεξάρτητους τροχούς και μια διεπαφή με τον οδηγό. Οι τροχοί, το τιμόνι, ο λεβιές των ταχυτήτων κ.λπ. αποτελούν περιφερειακά του υπολογιστή. Ο ρόλος του υπολογιστή είναι να περνάει κατάλληλα τις εντολές του οδηγού, όπως αυτές προκύπτουν από την διεπαφή ανθρώπου μηχανής με το τιμόνι κ.λπ., στους τροχούς, ώστε αυτοί να έχουν τις σωστές θέσεις και ταχύτητες περιστροφής. Μολονότι λοιπόν δεν μπορεί ο υπολογιστής του ηλεκτρικού αυτοκινήτου να υποκαταστήσει ολόκληρο το αυτοκίνητο, μπορεί ωστόσο να υποκαταστήσει ένα μεγάλο υποσύνολο των συστημάτων ενός κλασσικού αυτοκινήτου. Για παράδειγμα ρυθμίζοντας τις σχετικές ταχύτητες και θέσεις των τροχών υποκαθιστά το διαφορικό.

Μια άλλη δημοφιλής περίπτωση εξομοίωσης συστημάτων είναι τα δίκτυα που υλοποιούνται με λογισμικό, γνωστά ως *Software Defined Networks* (SDN). Και σ' αυτήν την περίπτωση δεν μπορούν να εξομοιωθούν συνιστώσες όπως κεραίες και αισθητήρες (sensors), όμως όλες οι λειτουργίες επεξεργασίας των σημάτων μπορούν να εξομοιωθούν και να υποκατασταθούν με λειτουργίες που συμβαίνουν μέσα σε υπολογιστές. Γενικά ο υπολογιστής μπορεί να εξομοιώσει επεξεργασίες που θα γίνονταν μέσα σε εξειδικευμένα κυκλώματα, είτε ψηφιακά είτε αναλογικά.

Ακόμη πιο εύκολη, επειδή δεν χρειάζονται εξειδικευμένα περιφερειακά, είναι η περίπτωση της εξομοίωσης υπολογιστών από υπολογιστές. Ένας υπολογιστής ή ένα σύστημα υπολογιστών μπορεί να εξομοιώσει μια ή περισσότερες υπολογιστικές μηχανές, που πιθανώς έχουν διαφορετικά λειτουργικά συστήματα. Αυτή είναι η περίπτωση στην οποία συνήθως αναφερόμαστε χρησιμοποιώντας όρους όπως *εικονικοποίηση* (virtualization) ή *εικονικές μηχανές* (virtual machines).

Μια φυσική μηχανή μπορεί να φιλοξενεί περισσότερες από μια εικονικές μηχανές. Η διαχείρισή τους γίνεται από λογισμικό που λέγεται *διαχειριστής εικονικών μηχανών* (*virtual machine manager* - VMM) ή *υπερεπόπτης* (*hypervisor*)[1]. Η φυσική μηχανή εκτελεί τις λειτουργίες κάθε μιας εικονικής μηχανής ακολουθιακά, δηλαδή με διαμερισμό χρόνου (time sharing). Ωστόσο προκειμένου οι εικονικές μηχανές να φαίνονται ότι λειτουργούν παράλληλα, η φυσική μηχανή δεν είναι σε θέση να διαθέτει μακρές χρονικές περιόδους σε μια εικονική μηχανή, γιατί κάτι τέτοιο θα αύξανε το χρόνο αντίδρασης μιας εικονικής μηχανής.

Βασικό πρόβλημα στην εικονικοποίηση μιας μηχανής, δηλαδή την αντικατάστασή της με λογισμικό που την εξομοιώνει, είναι η ταχύτητα. Η πραγματική μηχανή όπου γίνεται η εξομοίωση μιας άλλης μηχανής χρειάζεται κατά κανόνα να είναι δύο τάξεις μεγέθους ταχύτερη από την τελευταία (βλ. Κεφ. 1, [Puj20]). Η αντίστοιχη υπολογιστική δύναμη μπορεί να πρέπει να αναζητηθεί σε κέντρα δεδομένων (data centers) που διαθέτουν φάρμες υπολογιστών. Από την άποψη αυτήν η εικονοποίηση ρέπει προς την σπατάλη ενέργειας. Ευτυχώς από άλλες απόψεις μπορεί να γίνουν κάποιες οικονομίες.

Η εικονικοποίηση μπορεί να γίνεται με όρους οικονομιών κλίμακος εφόσον ανατεθεί σε κέντρα δεδομένων (που πιθανώς διαθέτουν μια φάρμα υπολογιστών). Για να συμβεί αυτό χρειάζεται ένα δίκτυο επικοινωνιών, συνήθως το Internet, αλλά η συ-

[1] Ο όρος αυτός είναι σχετικά νέος και στην ελληνική γλώσσα απαντάται με διάφορους τρόπους, όπως «ελεγκτής εικονικών μηχανών», «υπερεπόπτης», «επόπτης» ή αφήνεται αμετάφραστη η αγγλική λέξη.

νακόλουθη μεταφορά δεδομένων μπορεί να υπόκειται σε περιορισμούς χρόνου και όγκου. Ένα ηλεκτρικό αυτοκίνητο σήμερα εξομοιώνει μια σειρά λειτουργιών στον τοπικό του υπολογιστή (δηλ. εσωτερικό στο αυτοκίνητο) για να αποφεύγει κυρίως τις καθυστερήσεις από και προς ένα υπολογιστικό σύστημα που εναλλακτικά θα ήταν στο νέφος. Ένα σύστημα αυτόματης οδήγησης βασισμένο σε επεξεργασία δεδομένων από αισθητήρες του περιβάλλοντος του αυτοκινήτου δημιουργεί όγκο δεδομένων, που πρέπει να μεταφέρονται σε πραγματικό χρόνο σε μια ισχυρή μηχανή επεξεργασίας αυτών των δεδομένων. Τα συστήματα κινητών επικοινωνιών πέμπτης γενιάς (5G) φαίνονται να καλύπτουν μια τέτοια τηλεπικοινωνιακή απαίτηση από πλευράς ταχύτητας ανταλλαγής δεδομένων [Che+19].

Περαιτέρω οικονομία μπορεί να επιτευχθεί μέσα από την διακοπτόμενη χρήση (και ύπαρξη) των εικονικών μηχανών. Όταν η εικονική μηχανή δεν χρειάζεται μπορεί να διαγράφεται. Η διαγραφή μιας εικονικής μηχανής θέτει ένα πρόβλημα συνέχειας: Όταν σβήνουμε ένα PC αυτό δεν εξαφανίζεται, τα προγράμματα και τα δεδομένα του βρίσκονται ακόμη εκεί, και είναι έτοιμα να ξαναχρησιμοποιηθούν μόλις το ανάψουμε. Η λύση στο πρόβλημα της συνέχειας των εικονικών μηχανών βρίσκεται στη λήψη ενός *στιγμιοτύπου* (snapshot). Στο στιγμιότυπο αποτυπώνεται πλήρως η κατάσταση της μηχανής, οπότε αυτή μπορεί να ξαναδημιουργηθεί όποτε χρειαστεί και όπου χρειαστεί, πιθανώς σε άλλο περιβάλλον, ή και σε περισσότερα του ενός αντίγραφα.

Η εισαγωγή των εικονικών μηχανών δίνει νέες δυνατότητες για τη δημιουργία και την δοκιμή λογισμικού. Νέος κώδικας μπορεί να δοκιμαστεί εύκολα πάνω από διαφορετικά συστήματα, ενδεχομένως με διαφορετικό λειτουργικό σύστημα. Η εκτέλεση του κώδικα μπορεί να παγώσει, να ληφθούν στιγμιότυπα, να γίνει επιστροφή σε παλαιότερα εξ αυτών εφόσον διαπιστωθούν προβλήματα, με δυο λόγια οι εικονικές μηχανές δίνουν μεγαλύτερη ευελιξία στο debugging και γενικότερα στην ανάπτυξη λογισμικού, περιλαμβανομένου του λογισμικού των ίδιων των λειτουργικών συστημάτων.

Επίσης δίνουν τη δυνατότητα προσαρμογής ενός συστήματος σε μεταβαλλόμενες συνθήκες. Μια εικονική μηχανή που τρέχει το λογισμικό ενός τηλεπικοινωνιακού κόμβου μπορεί να μεταφερθεί από μια φυσική μηχανή σε μια άλλη χωρίς διακοπή της λειτουργίας της. Αυτό μπορεί να κριθεί απαραίτητο για λόγους κατανομής φορτίου στις φυσικές μηχανές, αλλά γενικά ανοίγει μια σειρά νέων δυνατοτήτων χρήσης των εικονικών μηχανών [Cla+05].

Η εισαγωγή της «τεχνολογίας» των εικονικών μηχανών έχει αφενός λύσει θέματα ασφάλειας, αφετέρου έχει δημιουργήσει νέα. Στην θετική πλευρά, η δοκιμή ενός νέου προγράμματος που έχει κατασκευάσει κάποιος άλλος και πρέπει να διαπιστωθεί κατά πόσο περιέχει κακόβουλο κώδικα μπορεί να γίνει πάνω σε μια εικονική μηχανή, η οποία εξασφαλίζει απομόνωση από το υπόλοιπο υπολογιστικό περιβάλλον. Η τεχνική εκτέλεσης ενός προγράμματος σε περιορισμένο περιβάλλον είναι γνωστή ως sandboxing. Ωστόσο η συγκατοίκηση μιας εικονικής μηχανής με άλλες, μερικές από τις οποίες μπορεί να ελέγχονται από επιτιθέμενους, κάνει πιο σημαντική την απομόνωση μεταξύ τους. Ακόμη και όταν μπορεί να εξασφαλιστεί η απομόνωση σε επίπεδο δεδομένων, ο επιτιθέμενος μπορεί να ξεκινήσει μια επίθεση άρνησης υπηρεσίας φροντίζοντας οι δικές του μηχανές να καταναλώνουν όσο γίνεται περισσότερους πόρους πάνω στη φυσική μηχανή.

Ωστόσο ένα σύστημα που φιλοξενεί εικονικές μηχανές διαθέτει πρόσθετη λειτουργικότητα, μέρος της οποίας είναι ο hypervisor, η παραγωγή και διανομή στιγμιοτύπων κ.λπ., καθώς και πρόσθετες διεπαφές που μπορούν να δώσουν νέες τρωτότητες. Ο hypervisor λειτουργεί με δικαιώματα ψηλότερα από εκείνα των λειτουργικών συστημάτων των επί μέρους εικονικών μηχανών, γεγονός που δίνει νέες ευκαιρίες στην

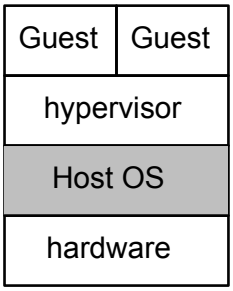
πλευρά της επίθεσης μη αντιμετωπίσιμες από τα κλασσικά συστήματα antivirus. Τρωτότητες μπορούν να ανακαλυφθούν στον hypervisor με τις συνήθεις μεθόδους, όπως υπερχείλιση καταχωρητών, αλλά οι τρωτότητες του hypervisor είναι πολύ πιο επικίνδυνες από τις τρωτότητες μιας συγκεκριμένης μηχανής, διότι μπορεί να επηρεάζουν περισσότερες της μιας μηχανές.

Ορισμένα προβλήματα ασφάλειας σχετίζονται με θέματα δικτύωσης. Οι εικονικές μηχανές, εφόσον μετέχουν σε εικονικά ή πραγματικά δίκτυα, προστατεύονται εν γένει από τείχη (firewalls), από συστήματα ανίχνευσης και πρόληψης εισβολών, από πρωτόκολλα ασφαλούς επικοινωνίας κ.λπ. Όταν όμως γίνεται μετανάστευση μιας εικονικής μηχανής, πρέπει να μεταφερθεί όχι μόνο η κατάστασή της και εκείνη του hypervisor, αλλά και η κατάστασή της σε σχέση με το δίκτυο. Ένα απλό παράδειγμα αποτελεί η μετανάστευση μιας μηχανής που βρίσκεται πίσω από ένα τείχος προστασίας F_1 και στη συνέχεια καταλήγει πίσω από ένα τείχος F_2. Αν οι κανόνες των F_1, F_2 δεν είναι πανομοιότυποι, στις νέες συνθήκες η μηχανή μπορεί να δέχεται περισσότερα ή λιγότερα πακέτα [MD16].

Οι επιθέσεις μπορεί να ευνοηθούν από το γεγονός ότι σε περιβάλλοντα εικονικών μηχανών ισχύουν ειδικές συνθήκες και περιστάσεις που δεν απαντώνται σε κλασσικές μηχανές.

Ταξινόμηση επιθέσεων σε συστήματα εικονικών μηχανών

Συστήματα που φιλοξενούν εικονικές μηχανές μπορούν να γίνουν αντικείμενο επίθεσης με τις γνωστές ευρύτερα μεν μεθόδους, αλλά με νέες δυνατότητες που δημιουργεί η προσθήκη του hypervisor και κατά περίπτωση περισσότερων στρωμάτων. Επίσης μπορεί να δημιουργηθεί νέο κακόβουλο λογισμικό με χρήση της εικονικοποίησης, δηλαδή μπορούν να γίνουν εργαλεία επίθεσης [SBL20].



Σχήμα 10.1: Στρώματα ενός συστήματος εικονικών μηχανών.

Στο Σχ. 10.1 φαίνονται τα στρώματα μιας μηχανής φιλοξενεί εικονικές μηχανές. Οι εικονικές μηχανές σημειώνονται ως Guests, πιο κάτω βρίσκεται ο υπερεπόπτης (hypervisor ή Virtual Machine Monitor - VMM), πιο κάτω το λειτουργικό σύστημα της μηχανής που φιλοξενεί τα παραπάνω και κάτω κάτω το σχετικό hardware. Μηχανές με αυτήν την διαστρωμάτωση είναι *Τύπου ΙΙ*. Το στρώμα του λειτουργικού συστήματος (Host OS) μπορεί να μην υπάρχει σε ορισμένα συστήματα, που ονομάζονται *Τύπου Ι*. Λογισμικό που βρίσκεται σε πιο ψηλά στρώματα προφανώς έχει εν γένει λιγότερα δικαιώματα, όπως ακριβώς σε έναν υπολογιστή το λογισμικό του λειτουργικού συστήματος έχει περισσότερα δικαιώματα από τις εφαρμογές που φιλοξενούνται στον υπολογιστή.

Κατά συνέπεια ένας εισβολέας συνήθως αρχίζει την επίθεσή του προσπαθώντας να αποκτήσει προχωρημένα δικαιώματα στις εικονικές μηχανές, στη συνέχεια στον υπερεπόπτη και στο λειτουργικό της φιλοξενούσας μηχανής (αν είναι Τύπου ΙΙ). Ανάλογα με το πόσο μακριά θα φτάσει μπορεί να κάνει διαφορετικές επιθέσεις. Η απόκτηση δικαιωμάτων σε μια εικονική μηχανή μπορεί γίνει είτε νομίμως, εφόσον ο εισβολέας είναι νόμιμος χρήστης μιας τέτοιας μηχανής, είτε παρανόμως, εφόσον έχει προηγηθεί εισβολή με οποιαδήποτε μέθοδο. Μια χονδρική κατάταξη τέτοιων επιθέσεων έχει ως εξής [SBL20]:

- Στο ρόλο του χρήστη μιας εικονικής μηχανής ο επιτιθέμενος μπορεί να οργανώσει επιθέσεις σε συστεγαζόμενες μηχανές (επίθεση *guest to guest*), π.χ. μέσω υπερβολικής κατανάλωσης πόρων, υπερχείλισης καταχωρητών κ.λπ., ή να προσπαθήσει πρώτα να αποκτήσει ψηλότερα δικαιώματα μέσα στην ίδια εικονική μηχανή (επίθεση *guest to self*, με τις συνήθεις μεθόδους που εφαρμόζονται και σε φυσικές μηχανές), ώστε να έχει τη δυνατότητα περισσότερων χειρισμών.

- Έχοντας υπό τον έλεγχό του μια εικονική μηχανή ο επιτιθέμενος μπορεί επίσης να αποπειραθεί επίθεση ενάντια στον υπερεπόπτη (επίθεση *guest to hypervisor*). Μια τέτοια επίθεση αν είναι επιτυχημένη θα εξασφαλίσει έλεγχο πάνω σε όλες τις μηχανές που επιβλέπει ο υπερεπόπτης.

- Μπορεί επίσης απ᾽ ευθείας από μια εικονική μηχανή να αποκτήσει δικαίωμα εκτέλεσης κώδικα στο λειτουργικό της φιλοξενούσας μηχανής (επίθεση *guest to host OS*).

- Η προηγούμενη επίθεση μπορεί να γίνει σε δυο βήματα, όπου το δεύτερο γίνεται από το λειτουργικό της φιλοξενούσας μηχανής στον υπερεπόπτη (επίθεση *hypervisor to host OS*).

Ως παράδειγμα τρωτότητας μέσω της οποίας ο επιτιθέμενος μπορεί να ξεκινήσει από μια εικονική μηχανή και να εκτελέσει κώδικα στον hypervisor αναφέρεται στην [SBL20] η εξής περίπτωση: Το *Parallels Desktop* είναι ένας εξομοιωτής Windows πάνω σε Mac OS. Ο hypervisor εξομοιώνει ορισμένες λειτουργίες αναχαιτίζοντας ορισμένες εντολές στις φιλοξενούμενες εικονικές μηχανές. Ορισμένες όμως εντολές assembly θέτουν εσφαλμένα τον hypervisor σε κατάσταση συναγερμού, γεγονός που δίνει την ευκαιρία στον επιτιθέμενο να παρακωλύσει ορισμένες υπηρεσίες, δηλαδή να κάνει μια επίθεση άρνησης υπηρεσίας. Με ορισμένες εντολές το πρόγραμμα του hypervisor μπορεί να οδηγηθεί να διακοπή, γεγονός που θα διακόψει και όλες της εικονικές μηχανές. Στο ίδιο άρθρο [SBL20] παρατίθεται μια σειρά παραδειγμάτων για όλες τις παραπάνω περιπτώσεις.

Προβλήματα ασφάλειας κατά την εν λειτουργία μετανάστευση

Όταν η κατανομή των εικονικών μηχανών σε ένα σύνολο φυσικών μηχανών δεν ανταποκρίνεται σε κριτήρια ισορροπημένης κατανομής η λύση είναι να γίνει αναδιανομή των πρώτων στις δεύτερες ενεργοποιώντας κάποια διαδικασία μετανάστευσης (migration) των εικονικών μηχανών. Η μετανάστευση μπορεί να γίνει σχετικά εύκολα δημιουργώντας ένα στιγμιότυπο της υπό μετακόμιση μηχανής και αντιγράφοντάς το στη νέα φυσική μηχανή. Ωστόσο αυτή η μέθοδος απαιτεί ένα χρονικό διάστημα διακοπής της λειτουργίας της εικονικής μηχανής. Το ανεκτό μήκος του χρονικού διαστήματος της διακοπής εξαρτάται από την εφαρμογή. Αν ένας ερευνητής έχει αναθέσει σε μια μηχανή να επιλύσει ένα μαθηματικό πρόβλημα, η διακοπή πιθανότατα δεν έχει

καμμιά σημασία. Αν όμως η μηχανή επιβλέπει ένα κρίσιμο σύστημα πραγματικού χρόνου ο χρόνος διακοπής πρέπει να ληφθεί υπόψη. Μια τέτοια περίπτωση μπορεί να είναι ένα Software Defined Network, όπου καλό είναι να διατηρηθεί η αδιάκοπη συνέχεια των λειτουργιών ενός κόμβου του δικτύου μολονότι η εικονική μηχανή που τις έχει αναλάβει βρίσκεται υπό μεταφορά. Ανάγκες τέτοιου είδους δημιούργησαν μεθόδους μετανάστευσης εν λειτουργία (live migration).

Ένα παρόμοιο πρόβλημα ήταν ήδη γνωστό στην κινητή τηλεφωνία κατά την διαδικασία μεταπομπής (handover), όπου μια κλήση που εξυπηρετείται συνδέοντας ένα κινητό με ένα σταθμό βάσης πρέπει να μεταφερθεί σε άλλο σταθμό βάσης χωρίς να διαταραχθεί η συνέχεια της κλήσης. Αυτό μπορεί να γίνει δημιουργώντας προς στιγμή δύο ρεύματα δεδομένων προς τους δύο σταθμούς βάσης και μόλις επιτευχθεί ο συγχρονισμός τους διακόπτεται η επικοινωνία με τον παλιό σταθμό βάσης. Η μετανάστευση μηχανών εν λειτουργία μπορεί να γίνει με παρόμοια λογική, δηλαδή δημιουργώντας ένα αντίγραφο της μηχανής στη νέα θέση χωρίς ακόμη να έχει διαγραφεί η μηχανή στην παλιά θέση. Η παλιά μηχανή συνεχίζει να λειτουργεί μέχρις ότου η νέα μηχανή καταστεί έτοιμη και ενημερωθεί με όλα τα δεδομένα που της είναι απαραίτητα για να συγχρονιστεί με την παλιά μηχανή.

Κάθε φορά όμως που δημιουργείται μια νέα περίπλοκη διαδικασία προκύπτουν νέες τρωτότητες. Επί πλέον, το υποσύστημα του hypervisor που ασχολείται με τη μετανάστευση συχνά τείνει να είναι λιγότερο προσεκτικά σχεδιασμένο και υλοποιημένο, με τη λογική ότι δεν είναι εκτεθειμένο στο εξωτερικό περιβάλλον. Επιθέσεις στη μετανάστευση έχουν περιγραφεί και εφαρμοσθεί ήδη από το 2008 από τους Oberheide και Cooke [OCJ08]. Οι επιθέσεις αυτές είναι του τύπου του ενδιάμεσου (man-in-the-middle) κατά την διαδικασία μετανάστευσης μιας μηχανής από την αρχική στην τελική θέση. Ο επιτιθέμενος έχει στη διάθεσή του διάφορες στρατηγικές για να εκμεταλλευθεί τη διαδικασία μετανάστευσης. Μπορεί να προκαλέσει τη μεταφορά της μηχανής σε λάθος στόχο, δηλαδή σε σύστημα υπό τον έλεγχό του αντί του συστήματος όπου κανονικά θα γινόταν η μεταφορά. Μπορεί επίσης να προκαλέσει μια επίθεσης άρνησης υπηρεσίας στέλνοντας ένα πλήθος μηχανών σε ένα σύστημα θύμα με σκοπό την υπερφόρτωσή του. Μπορεί είτε να αποσπάσει δεδομένα από την υπό μεταφορά μηχανή είτε και να αλλοιώσει το περιεχόμενο της μνήμης της. Γενικότερα η πλευρά της επίθεσης μέσω της εικονικοποίησης αποκτάει νέα εργαλεία επίθεσης.

Δημιουργία τρωτοτήτων μέσω εικονικότητας

Οι King και Chen [KC06] έχουν εξηγήσει με ποιο τρόπο μπορεί να δημιουργηθεί ένα rootkit. Αυτό γίνεται μέσω της εγκατάστασης ενός hypervisor κάτω από ένα υπάρχον λειτουργικό σύστημα και ανεβάζοντας το αρχικό λειτουργικό σύστημα σε μια εικονική μηχανή. Το εν λόγω rootkit επειδή είναι στον hypervisor δεν μπορεί να γίνει αντιληπτό από τα συστήματα άμυνας του θύματος, ενώ μπορεί να γνωρίζει και να ελέγχει την κατάσταση του τελευταίου. Υλοποιήσεις τέτοιων απειλών όπως το *Blue Pill* και το *Vitriol* [Zov06; Rut04] έδειξαν την ανάγκη να μπορεί να διαπιστωθεί κατά πόσο κάτω από μια μηχανή έχει εγκατασταθεί ένας hypervisor, προτάθηκαν διάφορες τεχνικές για την ανίχνευση [SBL20], αλλά και περαιτέρω μέτρα που μπορεί να λάβει ο επιτιθέμενος για να την αποφύγει. Για παράδειγμα, οι διαφορές στο χρονισμό που δημιουργεί η προσθήκη του hypervisor μπορούν να χρησιμοποιηθούν ως μέσο ανίχνευσης. Η μέθοδος *Blue Chicken* εκ μέρους του επιτιθέμενου συνίσταται στην προσωρινή απεγκατάσταση του hypervisor rootkit όσο γίνονται έλεγχοι από την πλευρά της άμυνας και την επανεγκατάστασή του όταν τελειώσουν.

ΚΕΦΑΛΑΙΟ 11

Νέφος

11.1 Εισαγωγή

Η τεχνολογία συχνά κάνει κύκλους επανερχόμενη σε προηγούμενες ιδέες και σχεδιασμούς. Η επάνοδος σε συγκεντρωτικές λύσεις μέσω του *υπολογιστικού νέφους* (computational cloud, που στη συνέχεια θα αναφέρεται απλώς ως *νέφος*, cloud) είναι μια τέτοια περίπτωση. Αντικείμενο αυτού του κεφαλαίου είναι η εξοικείωση με θέματα ασφάλειας του νέφους. Προφανώς το βασικό εργαλείο για την εξασφάλιση της μυστικότητας των δεδομένων που αποθηκεύονται στο νέφος ή που διακινούνται από και προς αυτό είναι η κρυπτογραφία, ωστόσο οι απαραίτητες λειτουργίες θέτουν νέες απαιτήσεις για το πώς θα χρησιμοποιηθεί η κρυπτογραφία. Ειδικές απαιτήσεις βάζει για παράδειγμα και στην περίπτωση της κρυπτογράφησης του σκληρού δίσκου ενός υπολογιστή, βλ. ενότητα 10.3

.

11.2 Υπηρεσίες παρεχόμενες από το νέφος

Οι υπολογιστές μετά τον πόλεμο και ως τη δεκαετία του 1950 ήταν εργαλεία επίλυσης πολύπλοκων προβλημάτων, για τα οποία ήταν γνωστή η μέθοδος επίλυσης, αλλά το πλήθος των υπολογισμών ήταν μεγάλο. Τέτοια προβλήματα, αν και μικρότερου μεγέθους, παλιότερα τα έλυναν οργανώνοντας ομάδες ανθρώπων και διαιρώντας σ' αυτούς το φορτίο. Για παράδειγμα, αυτός ήταν ο τρόπος με τον οποίο γίνονταν οι υπολογισμοί των λογαρίθμων που στη συνέχεια εκδίδονταν σε τυπωμένους καταλόγους ή ο υπολογισμός της τροχιάς του κομήτη του Halley για το 1758 [Gri13]. Στη δεκαετία του 1960 οι υπολογιστές μπήκαν στην υπηρεσία της επιχείρησης και του γραφείου ως εργαλεία διαχείρισης της πληροφορίας. Η χρήση του υπολογιστή μπορούσε να γίνει μέσα από τερματικούς σταθμούς, που δεν είχαν δυνατότητα επεξεργασίας, αλλά έδειχναν σε μια οθόνη τα αποτελέσματα που αντιστοιχούσαν σε αιτήματα που έθετε ο χρήστης μέσω ενός πληκτρολογίου. Η αποθήκευση δεδομένων και η επεξεργασία τους γινόταν μόνο μέσα στην κεντρική μονάδα και η δεκαετία του '70 είδε μια σειρά μηχανών που ήταν όλο και πιο εξελιγμένες και ταυτόχρονα οικονομικές, ώστε να μπορούν να αξιοποιηθούν από μια μεσαία επιχείρηση.

Η δεκαετία του 1980 είδε την αυγή των προσωπικών μηχανών, όπως το *IBM PC* και ο *MacIntosh*, αλλά αμέσως έγινε αντιληπτό ότι η συνεργασία τέτοιων μηχανών ώστε να αποτελέσουν ένα συνολικό σύστημα απαιτούσε την δικτύωσή τους. Παρ' όλο που η δημιουργία μεγάλων μηχανών, όπως οι υπερυπολογιστές, συνεχίστηκε, οι επιχειρήσεις βασίστηκαν σε πληθυσμούς δικτυωμένων μικρών αυτόνομων μηχανών που μπορούσαν να επικοινωνούν με servers και να ανταλλάσσουν πληροφορίες με βάσεις δεδομένων. Δηλαδή ενώ η αποθήκευση δεδομένων μπορούσε να είναι συγκεντρωμένη ή κατανεμημένη, η επεξεργασία παρέμεινε εν πολλοίς κατανεμημένη και η αντίστοιχη υλική υποδομή ομοίως. Κάθε εταιρία ή οργανισμός είχε τη δική της συλλογή μηχανών και πιθανώς ένα τμήμα τεχνολογίας της πληροφορίας (IT) για να τις συντηρεί σε software και hardware, ενώ έπρεπε να φροντίζει και για τα σχετικά θέματα ασφάλειας. Η κατάσταση αυτή παραμένει σε γενικές γραμμές ως σήμερα, αλλά τώρα υπάρχει η εναλλακτική δυνατότητα του υπολογιστικού νέφους.

Με το υπολογιστικό νέφος επανέρχεται ο συγκεντρωτισμός στην αποθήκευση και την επεξεργασία. Ο πάροχος των υπηρεσιών νέφους διαθέτει τις δικές του μηχανές, ενδεχομένως σε μεγάλους πληθυσμούς και μεγέθη, και τις διαθέτει στους πελάτες του σύμφωνα με τις επιθυμίες τους.

Ένας ορισμός του νέφους βρίσκεται στην σύσταση NIST SP 800-145 *The NIST Definition of Cloud Computing* του Σεπτ. 2011 [MG+11]: *Ο υπολογισμός νέφους είναι ένα μοντέλο που επιτρέπει την πανταχού παρούσα, βολική, on demand πρόσβαση στο δίκτυο σε μια κοινόχρηστη δεξαμενή διαμορφώσιμων υπολογιστικών πόρων (π.χ. δίκτυα, διακομιστές, χώρο αποθήκευσης, εφαρμογές και υπηρεσίες) που μπορούν να παρασχεθούν γρήγορα και να διατεθούν με ελάχιστη προσπάθεια διαχείρισης ή αλληλεπίδραση με τον πάροχο υπηρεσιών. Αυτό το μοντέλο νέφους αποτελείται από πέντε βασικά χαρακτηριστικά, τρία μοντέλα υπηρεσιών και τέσσερα μοντέλα ανάπτυξης.*

Τα πέντε χαρακτηριστικά στα οποία αναφέρεται η NIST SP 800-145 είναι τα εξής:

On demand αυτοεξυπηρέτηση Ο πελάτης μπορεί μονομερώς και χωρίς ανθρώπινη μεσολάβηση να ζητήσει τη διάθεση υπολογιστικών πόρων, όπως χρόνου επεξεργασίας και χρήση αποθηκευτικού χώρου μέσω δικτύου.

Ευρεία δικτυακή πρόσβαση Η πρόσβαση στους παραπάνω πόρους μπορεί να γίνει μέσα από ποικιλία πλατφορμών και τερματικών στην πλευρά του πελάτη.

Δεξαμενές πόρων Ο πάροχος διαθέτει μια δεξαμενή υπολογιστικών πόρων (επεξεργασίας, μνήμης, επικοινωνίας) μέσω της οποίας μπορεί να εξυπηρετήσει πολλούς και πολλών ειδών πελάτες, κατανέμοντας τους πόρους αυτούς με δυναμικό τρόπο ανάμεσά τους. Ο πελάτης δεν χρειάζεται να γνωρίζει πού ακριβώς βρίσκονται αυτοί οι πόροι, ειμή μόνον σε γενικές γραμμές, όπως σε ποια χώρα, περιοχή ή κέντρο επεξεργασίας δεδομένων.

Ταχεία ελαστικότητα Οι δυνατότητες και οι πόροι παρέχονται στον πελάτη ανάλογα με τις ανάγκες του και αυξομειώνονται κατάλληλα για να προσαρμοσθούν σ' αυτές. Από την πλευρά του ο πελάτης έχει την αίσθηση ότι πρόκειται για απεριόριστους πόρους, τους οποίους μπορεί να ζητήσει ανά πάσα στιγμή.

Καταμετρημένη εξυπηρέτηση Ο πάροχος έχει ένα τρόπο μέτρησης των πόρων και υπηρεσιών που παρέχει (και ανάλογης τιμολόγησης). Η μέτρηση αυτή πρέπει να είναι φανερή στον πελάτη.

Το νέφος επομένως υπόσχεται οικονομίες κλίμακος που επιτυγχάνουν ελαχιστοποιήσεις του κόστους, οι οποίες βέβαια επιτρέπουν στον πάροχο να αποκομίζει κέρδη,

αλλά εν μέρει μεταφέρονται στον πελάτη, ο οποίος αποκομίζει οφέλη σε σύγκριση με την περίπτωση της ανάπτυξης δικών του υποδομών και υπηρεσιών. Οι απαιτούμενες για τη χρήση του νέφους υποδομές του πελάτη περιορίζονται σε μηχανές ενός ελάχιστου επιπέδου δυνατοτήτων, προσωπικούς υπολογιστές, tablets, ακόμη και κινητά.

Τα μοντέλα υπηρεσιών στα οποία αναφέρεται ο παραπάνω ορισμός είναι σύμφωνα με την NIST SP 800-145 τα εξής:

Λογισμικό ως υπηρεσία (*Software as a Service - SaaS*) Εδώ ο πελάτης έχει τη δυνατότητα να τρέξει εφαρμογές του παρόχου πάνω στις υποδομές του νέφους χρησιμοποιώντας από την πλευρά του (δηλ. την πλευρά του πελάτη) μια σχετικά απλή διεπαφή, π.χ. ένα φυλλομετρητή (browser) ή ένα απλό πρόγραμμα. Εν γένει ο πελάτης δεν έχει δυνατότητα να ελέγξει τους σχετικούς πόρους, εκτός ίσως κάποιων ρυθμίσεων.

Πλατφόρμα ως υπηρεσία (*Platform as a Service - Paas*) Ο πελάτης βλέπει μια πλατφόρμα, πάνω στην οποία μπορεί να αναπτύξει ή να στήσει τις δικές του εφαρμογές. Τους πόρους που είναι απαραίτητοι εν γένει τους διαχειρίζεται περαιτέρω η πλατφόρμα, δηλαδή είναι στον έλεγχο του παρόχου νέφους.

Υποδομή ως υπηρεσία (*Infrastructure as a Service - IaaS*) Εδώ ο πελάτης βλέπει μια συλλογή πόρων, δηλαδή μνήμης, επεξεργασίας, δικτύου κ.λπ., όπου μπορεί να εγκαταστήσει το δικό του λειτουργικό σύστημα και περαιτέρω τις εφαρμογές του.

Τέλος η παραπάνω σύσταση αναφέρει συγκεκριμένα μοντέλα εγκατάστασης προσπαθώντας να καλύψει το πιθανό φάσμα χρηστών:

Ιδιωτικό νέφος για την αποκλειστική χρήση εσωτερικά σε έναν οργανισμό ή εταιρεία.

Κοινοτικό νέφος Παρέχεται για χρήση από μια συγκεκριμένης κοινότητας χρηστών.

Δημόσιο νέφος για το γενικό κοινό.

Υβριδικό νέφος είναι μείγμα δύο ή περισσότερων από τις παραπάνω περιπτώσεις.

Σε όλες τις παραπάνω περιπτώσεις ο ιδιοκτήτης και ο διαχειριστής του νέφους μπορεί να είναι ο ίδιος ο οργανισμός, εταιρία, κοινότητα κ.λπ. ή το νέφος να παρέχεται από κάποια άλλη οντότητα, και οι υποδομές μπορούν να βρίσκονται εντός ή εκτός των εγκαταστάσεων της πρώτης.

Στην NIST SP 500-292 [Liu+11] περιγράφονται οι παίκτες (στο κείμενο *actors*) γύρω από το νέφος και οι αρμοδιότητές τους:

Καταναλωτής νέφους (*Cloud Consumer*) πρόσωπο ή οργανισμός που διατηρεί μια επιχειρηματική σχέση με παρόχους νέφους και χρησιμοποιεί τις υπηρεσίες τους.

Πάροχος νέφους (*Cloud Provider*) πρόσωπο, οργανισμός ή οντότητα που έχει την ευθύνη διάθεσης υπηρεσιών νέφους στους ενδιαφερόμενους.

Ελεγκτής νέφους (*Cloud Auditor*) μπορεί να διενεργεί ανεξάρτητες αξιολογήσεις των υπηρεσιών νέφους.

Μεσίτης νέφους (*Cloud Broker*), μπορεί να εμφανίζεται ως ενδιάμεσος και να διαπραγματεύεται τη σχέση μεταξύ καταναλωτή και παρόχου, καθώς και να διαχειρίζεται τη χρήση και την παροχή των υπηρεσιών.

Φορέας νέφους (*Cloud Carrier*), παρέχει συνδεσιμότητα και μεταφορά υπηρεσιών νέφους από τον πάροχο στον καταναλωτή.

Με το νέφος εφαρμογές που κλασσικά απαιτούν δυνατότητες επεξεργασίας και αποθήκευσης παρέχονται ως έτοιμες υπηρεσίες, αν ο πελάτης το επιθυμεί, δηλαδή αν αυτές έχουν τις δυνατότητες που επιθυμεί. Αν όχι, ο πελάτης μπορεί να επιλέξει την πλατφόρμα που επιθυμεί και να στήσει πάνω της τις δικές του εφαρμογές και υπηρεσίες. Αν οι απαιτήσεις του πελάτη είναι ακόμη πιο προχωρημένες, μπορεί να χρησιμοποιήσει το νέφος ως υποκατάστατο μιας υπολογιστικής υποδομής, δηλαδή να ορίσει στο νέφος τις δικές του εικονικές μηχανές με κατάλληλες δυνατότητες.

Όλα αυτά μπορεί να τα χρησιμοποιεί έναντι ενός ενοικίου για όσο χρόνο του είναι απαραίτητα και με το ενοίκιο να διαμορφώνεται ανάλογα με την ποιότητα και την ποσότητα των πόρων. Η παρεχόμενη στον πελάτη ευελιξία αφορά και τον απαιτούμενο χρόνο για την απόκτηση μιας νέας υπηρεσίας, ειδικά αν επιλέξει τις έτοιμες υπηρεσίες του νέφους. Επίσης η ευελιξία αφορά στα απαραίτητα μεγέθη πόρων, π.χ. μνήμης ή ταχύτητας υπολογισμού, που μπορούν να αυξάνονται ή να μειώνονται ανάλογα με τις ανάγκες, πράγμα που δεν θα ήταν δυνατόν τόσο εύκολα και γρήγορα αν ο πελάτης είχε τις δικές του μηχανές. Ταυτόχρονα ο τελικός χρήστης απαλλάσσεται από ορισμένες παράπλευρες υποχρεώσεις, όπως το να κρατάει σε τακτικά διαστήματα αντίγραφα των δεδομένων του και να φροντίζει για την ασφάλειά τους.

Ωστόσο όλες αυτές οι διευκολύνσεις, παρέχονται με ορισμένες θυσίες και υποχωρήσεις από την πλευρά του πελάτη, πέρα από το κόστος που θα επωμιστεί ο τελευταίος. Ο πελάτης δεν έχει άμεσο έλεγχο των δεδομένων του και το μόνο που μπορεί να κάνει είναι να εμπιστευθεί τον πάροχο του νέφους και των υπηρεσιών του. Μπορούν να υπάρξουν παραβιάσεις που γίνονται εν γνώσει ή με την ανοχή του παρόχου και παραβιάσεις που γίνονται από την αδυναμία του. Ζούμε σε ένα κόσμο όπου η ανασφάλεια εκδηλώνεται καθημερινά με μαζικές κυβερνοεπιθέσεις εις βάρος εταιριών και οργανισμών οποιουδήποτε μεγέθους και οσοδήποτε καλής φήμης, περιλαμβανομένων στρατιωτικών οργανισμών, μυστικών υπηρεσιών ή άλλων οργανισμών εξειδικευμένων στην ασφάλεια. Ταυτόχρονα δεν είναι απόλυτα εξακριβωμένη η σχέση των παρόχων νέφους με μυστικές ή άλλες κρατικές υπηρεσίες και σχεδόν πάντοτε κυκλοφορούν φήμες για παρακολουθήσεις, backdoors κ.λπ.

Η «μαγική» λύση σε τέτοιες περιπτώσεις (όπως και στην επικοινωνία δύο πλευρών) προβάλλει να είναι η κρυπτογράφηση ως τα άκρα (end-to-end). Για παράδειγμα στην περίπτωση υπηρεσιών αποθήκευσης αρχείων (υπηρεσιών τύπου Storage as a Service, όπως *Box*, *Dropbox*, *Onedrive* κ.λπ.) ο χρήστης θα μπορούσε να στείλει τα δεδομένα του ήδη κρυπτογραφημένα. Ομοίως θα μπορούσε να τα κατεβάζει κρυπτογραφημένα και να κάνει την αποκρυπτογράφηση στον δικό του υπολογιστή. Μια σειρά προϊόντων που έχουν ήδη εμφανισθεί κάνουν κρυπτογράφηση στη μηχανή του πελάτη (client side encryption) είτε βάζοντας ένα κρυπτογραφημένο φάκελλο στη θέση του φακέλλου που ανεβαίνει στο νέφος, είτε κρυπτογραφώντας τα αρχεία ένα ένα χωριστά [LRS14]. Τέτοια προϊόντα είναι τα *Cloudfogger*, *Safemonk*, *BoxCryptor*, *Viivo*, *TrueCrypt* κ.α. Η συγκεκριμένη λύση είναι γενική και δεν εξαρτάται από την περαιτέρω προστασία που παρέχει ή δεν παρέχει ο πάροχος υπηρεσιών αποθήκευσης, αλλά παρεμποδίζει μια σειρά δυνατοτήτων και υπηρεσιών που θα μπορούσε να προσφέρει ο πάροχος, δεδομένου ότι ο τελευταίος δεν μπορεί να κάνει γενικά υπολογισμούς πάνω στα κρυπτογραφημένα δεδομένα.

Εν γένει οι πάροχοι υπηρεσιών που σχετίζονται με την διαχείριση και την αποθήκευση δεδομένων αποφεύγουν την κρυπτογράφηση στις εγκαταστάσεις του παρόχου ακόμη και όταν η υπηρεσία είναι πιο κοντά στην απλή αποθήκευση και δεν περιλαμ-

βάνει επεξεργασία. Το *Dropbox* (στον βαθμό που είναι γνωστό [Wan+12]) χωρίζει τα αρχεία σε κομμάτια (chunks) των 4 MB και υπολογίζει για το καθένα ένα κωδικό κατακερματισμού κατά AES-256. Οι κωδικοί υπολογίζονται και στην πλευρά του πελάτη και στην πλευρά του παρόχου. Όταν ο πελάτης δημιουργεί ένα διαφορετικό αρχείο, συγκρίνονται οι κωδικοί στις δύο πλευρές και ανεβαίνουν στον πάροχο μόνο τα κομμάτια με διαφορετικό κωδικό. Η μεταφορά προς τη μία ή την άλλη κατεύθυνση γίνεται με κρυπτογράφηση κατά AES-256, γεγονός που παρέχει προστασία έναντι τρίτων, ωστόσο τα δεδομένα παραμένουν ακρυπτογράφητα μέσα στις δύο πλευρές.

11.3 Ζητήματα ασφάλειας στο νέφος

Η ενότητα 10.3 δείχνει ποια ζητήματα θέτει και πώς επιλύεται η κρυπτογραφημένη τμηματική αποθήκευση και ανάκληση πληροφορίας που είναι βασικό ζήτημα και στο νέφος. Ωστόσο το νέφος θέτει και άλλα ζητήματα επειδή

- οι εφαρμογές που χρησιμοποιεί ένας οργανισμός μετατοπίζονται από τις δικές του υποδομές στις υποδομές άλλων οργανισμών, όπου συγκατοικούν πιθανώς με τις εφαρμογές ανταγωνιστών του πρώτου οργανισμού,

- οι υπηρεσίες του παρέχονται σε πληθυσμούς χρηστών, μέσα στους οποίους μπορεί να είναι εισβολείς,

- οι υπηρεσίες παρέχονται ομαδικά και κάθε χρήστης πρέπει να διασφαλίζεται από διαρροές δεδομένων προς άλλους και από άλλες παρενέργειες,

- η πρόσβαση γίνεται μέσα από ένα δίκτυο,

- η πρόσβαση γίνεται μέσα από διάφορες διεπαφές χρήστη - υπηρεσίας.

- ο χρήστης έχει μειωμένο έλεγχο σε όσα γίνονται μέσα στις εγκαταστάσεις του παρόχου του νέφους και δεν γνωρίζει σε ποιο βαθμό ο πάροχος έχει πρόσβαση στα δεδομένα του.

Στην NIST SP 500-292 αναφέρεται ότι τα συστήματα που βασίζονται στο νέφος πρέπει να λαμβάνουν υπόψη απαιτήσεις ασφάλειας όπως έλεγχος ταυτότητας, εξουσιοδότηση, διαθεσιμότητα, εμπιστευτικότητα, διαχείριση ταυτότητας, ακεραιότητα, έλεγχος, παρακολούθηση της ασφάλειας, απόκριση σε περιστατικά και διαχείριση της πολιτικής ασφάλειας. Στην πιο συνηθισμένη περίπτωση που η επίθεση στο νέφος γίνεται από εξωτερικό επιτιθέμενο, αυτός θα εμφανισθεί σε μια από τις διεπαφές που το νέφος παρουσιάζει προς τον χρήστη (καταναλωτή). Η διεπαφή αυτή διαφέρει ανάλογα με το μοντέλο παροχής υπηρεσίας, δηλ. SaaS, PaaS, IaaS. Για παράδειγμα στην περίπτωση SaaS ο χρήστης μπορεί έρθει σε επαφή με το νέφος μέσω του Διαδικτύου χρησιμοποιώντας ένα φυλλομετρητή (browser), οπότε στην περίπτωση αυτή πρέπει να δοθεί έμφαση στην ασφάλεια του φυλλομετρητή. Επίσης η NIST SP 500-292 λέει δυο λόγια για τους κινδύνους ιδιωτικότητας (privacy) στο νέφος. Η βασική σύσταση για θέματα ασφάλειας και ιδιωτικότητας στο νέφος είναι η NIST SP 800-144 [JG11].

Ζητήματα ασφάλειας και ιδιωτικότητας των δεδομένων

Στη σύσταση επισημαίνονται μεταξύ άλλων τα εξής που σχετίζονται με τη φύλαξη, την ιδιωτικότητα και τη διαφάνεια (όπου αυτή χρειάζεται) των δεδομένων:

- Με τη χρήση υπηρεσιών νέφους ένας οργανισμός ξεφεύγει από τις συνήθεις διαδικασίες προμήθειας υπολογιστικών πόρων με αποτέλεσμα να μη λαμβάνονται υπόψη κατά περίπτωση πολιτικές και διαδικασίες σχετικές με ασφάλειας και ιδιωτικότητα.

- Η συμμόρφωση με (συχνά μεταβαλλόμενους) νόμους και κανονισμούς μπορεί να τεθεί εν αμφιβόλω. Για παράδειγμα, όλες οι εταιρίες και οργανισμοί στην Ευρωπαϊκή Ένωση έχουν υποχρέωση τήρησης της GDPR και οφείλουν να αναφέρουν περιστατικά παραβίασης ή απώλειας δεδομένων. Δεν είναι πάντοτε σαφές το πώς αυτές οι υποχρεώσεις υλοποιούνται στην περίπτωση σχετικών περιστατικών όταν τα δεδομένα είναι στο νέφος.

- Υπάρχουν σε ορισμένες περιπτώσεις κανονισμοί σχετικοί με την φυσική ασφάλεια των δεδομένων, όπως π.χ. για την αντοχή της εγκατάστασης σε φυσικές καταστροφές και εισβολές. Η τήρηση τέτοιων προδιαγραφών είναι δύσκολο να διαπιστωθεί όταν δεν είναι καν γνωστό πού βρίσκονται τα δεδομένα.

- Ορισμένες υποχρεώσεις των οργανισμών σχετίζονται με τη διατήρηση των μεταδεδομένων, τα οποία μπορεί να αλλοιωθούν στα πλαίσια εργασιών αρχειοθέτησης από πλευράς του παρόχου νέφους.

- Ο κύκλος αυτών που μπορούν να απειλήσουν ένα σύστημα εκ των έσω αυξάνεται, διότι στους συνήθεις υπόπτους (π.χ. δυσαρεστημένους υπαλλήλους) προστίθενται και εργαζόμενοι που ανήκουν στον πάροχο του νέφους, καθώς και άλλοι χρήστες του νέφους με βλαπτικές προθέσεις [KVG13].

- Η ιδιοκτησία των δεδομένων πρέπει να είναι σαφής. Πάροχοι υπηρεσιών νέφους όπως το *Facebook* ή το *youTube* μπορεί να διεκδικούν την ιδιοκτησία δεδομένων που ανεβάζουν οι χρήστες. Τέτοιες διεκδικήσεις υποθάλπονται και από νεφελώδεις όρους στο συμβόλαιο μεταξύ παρόχου και χρήστη. Επίσης, οι πάροχοι νέφους τείνουν να συλλέγουν μεταδεδομένα σχετικά με τη συμπεριφορά των χρηστών τους και συχνά τα μεταπωλούν, με άγνωστες συνέπειες για την ιδιωτικότητα και την ασφάλεια των χρηστών.

- Η σύνθεση υπηρεσιών που χρησιμοποιούν άλλες υπηρεσίες και στρώματα νέφους διαφόρων παρόχων καταλήγει σε περιπτώσεις που είναι δύσκολο να ελεγχθούν.

- Δεδομένου ότι τα δεδομένα ενός οργανισμού βρίσκονται έξω από τη φύλαξη και τον έλεγχό του, η εκτίμηση των κινδύνων στους οποίους αυτά υπάγονται είναι δύσκολη.

Ζητήματα σχετικά με την αρχιτεκτονική του νέφους

Η αρχιτεκτονική του λογισμικού και του υλισμικού (hardware) διαφέρουν κατά περίπτωση ανάλογα με τις επιλογές του παρόχου υπηρεσιών νέφους και οι επιλογές αυτές έχουν συνέπειες στην ασφάλεια και την ιδιωτικότητα. Όταν επιτρέπεται ο καθορισμός εικονικών μηχανών (virtual machines) κάτω από το λειτουργικό σύστημα μιας εικονικής μηχανής προστίθεται ένα στρώμα που επιτρέπει τη δημιουργία και τη λειτουργία τέτοιων μηχανών. Το στρώμα αυτό είναι γνωστό ως *ελεγκτής εικονικών μηχανών* ή *υπερεπόπτης – hypervisor*. Οι εικονικές μηχανές τρέχουν πάνω στη *μηχανή-ξενιστή*

(*host machine*) και λέγονται *φιλοξενούμενες μηχανές* (*guest machines*), οι οποίες περαιτέρω μπορεί να έχουν διαφορετικά λειτουργικά συστήματα. Η ποικιλία αυτή κάνει το σύστημα πιο πολύπλοκο και μπορεί να δημιουργήσει τρωτότητες. Η προσθήκη του υπερεπόπτη αυξάνει τις προσφερόμενες διεπαφές και μαζί τους τις δυνατότητες παρέμβασης των επιτιθεμένων, ενώ μπορεί και ο ίδιος ο υπερεπόπτης να υποστεί επιθέσεις.

Το νέφος είναι σαν μια πολυκατοικία με πολλούς ενοίκους. Όταν στο ίδιο σύστημα φιλοξενούνται πολλές εικονικές μηχανές υπάρχει η πιθανότητα κακόβουλος κώδικας να διαχυθεί από μια μηχανή σε άλλες ή προς τον υπερεπόπτη.

Η δημιουργία εικονικών μηχανών συχνά συνδυάζεται με τη δημιουργία εικονικών μεταγωγέων και δικτύων, προκειμένου να δημιουργηθεί ένα ολοκληρωμένο εικονικό περιβάλλον. Ωστόσο το εικονικό δίκτυο μπορεί να μην είναι ορατό στα συστήματα προστασίας από δικτυακές επιθέσεις (NIDS - Network Intrusion Detection Systems), παρ΄ όλο που κάποια συστήματα υπερεπόπτη μπορεί να διαθέτουν ορισμένες αντίστοιχες ικανότητες.

Ένας ακόμη παράγοντας ανασφάλειας είναι το γεγονός ότι στα πραγματικά δίκτυα ο ρόλος του διαχειριστή μιας μηχανής είναι κατά κανόνα χωριστός από το ρόλο του διαχειριστή του δικτύου, υφίσταται δηλαδή διαχωρισμός καθηκόντων μέσω της ανάθεσής τους σε διαφορετικά πρόσωπα. Στα εικονικά περιβάλλοντα τέτοιος διαχωρισμός είναι πιο πιθανό να μην υφίσταται.

Σε ένα σύστημα που φιλοξενεί εικονικές μηχανές τηρούνται αποθετήρια εικόνων (images) εικονικών μηχανών. Η εικόνα περιλαμβάνει το λειτουργικό μαζί με τις εγκατεστημένες στη μηχανή εφαρμογές και την κατάσταση τη στιγμή που έχει ληφθεί η εικόνα. Οι επιτιθέμενοι μπορούν είτε να εξετάζουν εικόνες προκειμένου να βρουν αδυναμίες είτε να αλλοιώνουν και να διανέμουν εικόνες.

Μια άλλη πηγή αδυναμιών είναι η πλευρά του client. Υπηρεσίες νέφους μπορούν π.χ. να χρησιμοποιούνται μέσω ενός φυλλομετρητή (browser), του οποίου τις αδυναμίες μπορεί να εκμεταλλευτεί ο επιτιθέμενος.

11.4 Προστασία δεδομένων

Η συγκέντρωση δεδομένων στις εγκαταστάσεις των παρόχων νέφους αποτελεί μια πρόκληση για όποιον θέλει με παράνομο τρόπο να αντλήσει δεδομένα ή να τα αλλοιώσει ή απλώς να παρεμποδίσει τη χρήση τους με επιθέσεις άρνησης υπηρεσίας. Τα δεδομένα πρέπει να είναι διασφαλισμένα σε διάφορες καταστάσεις, (α) όταν είναι σε ακινησία (αποθηκευμένα), (β) όταν είναι σε μεταφορά και (γ) όταν χρησιμοποιούνται. Στις περιπτώσεις (α) και (β) η κρυπτογράφηση μπορεί να είναι μια λύση.

Πώς θα προστατευθούν τα δεδομένα ενώ χρησιμοποιούνται είναι ένα ανοιχτό ζήτημα. Θα θέλαμε ιδανικά να μπορούμε να χειριζόμαστε τα δεδομένα χωρίς να τα αποκρυπτογραφούμε. Για παράδειγμα, αν a και b είναι δύο αριθμοί που είναι αποθηκευμένοι σε κρυπτογραφημένη μορφή $e(a), e(b)$ και πρέπει να προστεθούν η συνήθης προσέγγιση είναι να αποκρυπτογραφηθούν πρώτα, να γίνει η πρόσθεση $c = a + b$, και να αποθηκευτεί το $e(c)$. Ιδανικά θα θέλαμε να έχουμε ένα τρόπο υπολογισμού του $e(c)$ χωρίς ενδιάμεση αποκρυπτογράφηση. Μια λύση που έχει προταθεί είναι η *ομομορφική κρυπτογράφηση* (homomorphic encryption). Η επιθυμητή ιδιότητα εδώ θα ήταν να μπορεί να υπολογιστεί το $e(a + b)$ απ' ευθείας από τα $e(a), e(b)$ χωρίς να γίνει αποκρυπτογράφηση. Θα ήταν επίσης επιθυμητές κατά περίπτωση άλλες πράξεις, όπως πολλαπλασιασμός, ή η δυνατότητα για συγκρίσεις τύπου $a < b$ απ' ευθείας συγκρίνοντας τα $e(a), e(b)$. Η ιδέα να αναπτυχθεί κρυπτογράφηση με τέτοιες ιδιότητες

είχε προταθεί από τον Ronald Rivest [RAD+78]. Ο Rivest έθεσε το πρόβλημα χωρίς να πιστεύει ιδιαίτερα ότι έχει λύση. Στη συνέχεια ο Craig Gentry [Gen09; Van+10] πρότεινε μια προσεγγιστική λύση, της οποίας το σφάλμα μπορεί να εξαλειφθεί με θυσία υπολογισμών. Άλλες λύσεις προτάθηκαν στη συνέχεια.

Θέμα ασφάλειας αποτελεί και ο καθαρισμός δεδομένων (sanitization). Ο καθαρισμός συνίσταται στην διαγραφή δεδομένων, επιλεκτική ή ολική με ασφαλείς μεθόδους διαγραφής. Ειδικά όταν ένας χρήστης αποχωρεί από το νέφος πρέπει να έχει τη βεβαιότητα ότι τα δεδομένα του όντως θα διαγραφούν. Στην περίπτωση που το φυσικό μέσο αποσύρεται από τον πάροχο νέφους και δεν θα ξαναχρησιμοποιηθεί, η διαγραφή μπορεί να φτάσει μέχρι και του σημείου της καταστροφής του φυσικού μέσου με τρόπο που να μη μπορεί να ανακτηθεί η πληροφορία (ισχυρή απομαγνήτιση, θραύση, καύση κ.λπ.).

Οι υπηρεσίες νέφους προσφέρονται συγχρόνως σε μεγάλο αριθμό πελατών - χρηστών - συνδρομητών και η διαθεσιμότητά τους έχει ιδιαίτερη βαρύτητα ή και κρισιμότητα κατά περίπτωση. Οποιαδήποτε διακοπή είναι από ενοχλητική ως καταστροφική. Πιθανές αιτίες μπορεί να είναι επιθέσεις, φυσικές καταστροφές, διακοπές ηλεκτροδότησης κ.λπ. Διακοπές λόγω συντήρησης είναι προγραμματισμένες, οι πελάτες μπορούν να προειδοποιηθούν και μέτρα διάθεσης εναλλακτικών πόρων μπορούν να ελαχιστοποιήσουν τις επιπτώσεις.

ΚΕΦΑΛΑΙΟ 12

Blockchain

12.1 Εισαγωγή

Στο κεφάλαιο αυτό θα εξετάσουμε τις βασικές αρχές την τεχνολογία του blockchain και θα δούμε επιλεγμένες εφαρμογές της, με κύριο παράδειγμα το BitCoin. Στην περίπτωση του blockchain η ίδια η χρήση της λέξης «τεχνολογία» είναι υπερβολική. Θα μπορούσε να υποστηριχθεί ότι το blockchain δεν είναι παρά μια σχετικά απλή σειρά ασκήσεων πάνω στις συναρτήσεις κατακερματισμού. Εν πάση περιπτώσει όμως οι ασκήσεις αυτές έχουν αποδειχθεί ιδιαίτερα δημοφιλείς τα τελευταία χρόνια.

Μια άλλη αξιοσημείωτη πλευρά της «τεχνολογίας» αυτής είναι η φιλοδοξία των σχεδιαστών της να προωθήσουν μια κοινωνία λιγότερο κεντρικά ελεγχόμενη, φιλοδοξία με την οποία είδαμε ότι σχετίζεται και η μη συμμετρική κρυπτογραφία. Η χρήση βέβαια τέτοιων εργαλείων εύκολα μπορεί να εκτραπεί, μιας και έχει διαπιστωθεί ότι οι λαθρέμποροι ναρκωτικών και όπλων είναι ενθουσιώδεις χρήστες των κρυπτονομισμάτων, επειδή οι σχετικές συναλλαγές είναι δύσκολο να ιχνηλατηθούν.

Κατάστιχα

Γενικότερα το blockchain μπορεί να εγγυηθεί την αξιοπιστία ενός κατανεμημένου «κατάστιχου», δηλαδή μιας κατανεμημένης συλλογής από εγγραφές που μπορούν να αφορούν σε οτιδήποτε είναι χρήσιμο να καταγραφεί.

Ένα κλασσικό κατάστιχο περιλαμβάνει συνήθως μια σειρά από συναλλαγές, π.χ. το κατάστιχο ενός ενεχυροδανειστή μπορεί να καταγράφει δάνεια που πήραν πελάτες, αντικείμενα που άφησαν ως ενέχυρο, μερικές ή ολικές πληρωμές που έκαναν για να πληρώσουν τόκους ή να επιστρέψουν τα δανεικά κ.ο.κ. Τα κατάστιχα υπάρχουν από την εποχή της αυγής της γραφής και φαίνεται ότι οι πρώτες χρήσεις της γραφής αφορούσαν ακριβώς σε κατάστιχα. Για παράδειγμα, πινακίδες γραμμένες σε Γραμμική-Β, προερχόμενες από μηκυναϊκά κέντρα, απαριθμούν είδος και αριθμό αντικειμένων (όπλων, εργαλείων, τροφίμων, ζώων κ.λπ.) που ήταν αποθηκευμένα στις κρατικές αποθήκες [VC15]. Αντίστοιχες πινακίδες γραμμένες με σφηνοειδή γραφή προέρχονται από την αρχαία Βαβυλώνα.

Οι πινακίδες αυτές ήταν εν γένει μοναδικές, ήταν μόνιμες εξ αιτίας του υλικού τους[1] και κρατούνταν φυλαγμένες σε κάποια κεντρικά κρατικά αρχεία, οπότε η παραχάραξή τους ήταν δύσκολη, αν και προφανώς όχι αδύνατη. Όταν η καταγραφή άρχισε να γίνεται σε λιγότερο μόνιμα μέσα, όπως πάπυρο ή χαρτί, η καλή φύλαξη (π.χ. σε μια θήκη ασφαλισμένη με βουλοκέρι και περαιτέρω σε προστατευμένο χώρο) και μια υπογραφή ή μια σφραγίδα ήταν τα συνήθη εχέγγυα ενάντια στην αλλοίωση ή κλοπή. Οι ίδιες μέθοδοι χοντρικά μας έχουν φτάσει ως τη σημερινή εποχή, με τις συνήθεις απαιτήσεις να είναι η ύπαρξη ενός και μοναδικού πρωτοτύπου, το οποίο φέρει κάποιες ενδείξεις αυθεντικότητας, π.χ. ένα βιβλίο εσόδων-εξόδων που έχει σφραγισθεί από την εφορία σε συγκεκριμένη σελίδα, η οποία δεν μπορεί να αντικατασταθεί. Στη σπάνια περίπτωση που χρειάζεται αντίγραφο από το πρωτότυπο κατάστιχο πρέπει να ακολουθηθεί ειδική διαδικασία που θα εγγυάται την αυθεντικότητα του αντιγράφου, σε κάθε περίπτωση όμως είναι σαφές ότι στο αντίγραφο δεν μπορούν να γίνουν οι ίδιες ενέργειες όπως στο πρωτότυπο, π.χ. δεν μπορούν να προστεθούν νέες εγγραφές. Τα λεγόμενα «διπλά βιβλία» (δηλαδή το να κρατάει κανείς ένα βιβλίο για τα πραγματικά γεγονότα και ένα για να το δείχνει στις κρατικές υπηρεσίες) παραμένουν μια γνωστή μέθοδος απάτης και είναι παράνομα.

Το *κατανεμημένο κατάστιχο* (distributed ledger) [Wal15], δηλαδή ένα κατάστιχο όπου πολλοί μπορούν να τοποθετούν εγγραφές και να τις διαβάζουν, είναι μια καινοτομία της εποχής του υπολογιστή. Ιδανικά σ᾽ ένα τέτοιο κατάστιχο εγγραφές μπορούν να εισάγονται ή να μεταβάλλονται βάσει κανόνων μόνο από εξουσιοδοτημένα άτομα (ή οντότητες) και να διαβάζονται επίσης εφόσον υπάρχει σχετικό δικαίωμα. Ιδανικά, επίσης, οι νέες εγγραφές θα ήταν ορατές αμέσως σε όλους, αλλά κάτι τέτοιο είναι τεχνικά αδύνατο και δημιουργεί προβλήματα προς επίλυση, περιλαμβανομένων και προβλημάτων ασφαλείας. Το *κατανεμημένο κατάστιχο* ως όρος διακρίνεται συνήθως από μια κατανεμημένη βάση δεδομένων ως προς το ότι το κατανεμημένο κατάστιχο είναι διεσπαρμένο στους κόμβους ενός δικτύου ομοτίμων (peer to peer), καθένας εκ των οποίων ενημερώνει το δικό του αντίγραφο. Για να επιτευχθεί αυτό με αξιόπιστο τρόπο οφείλει να διαθέτει μεθόδους (αλγόριθμους) που εξασφαλίζουν ότι τα αντίγραφα είναι σωστά χωρίς την ύπαρξη ενός κεντρικού διαχειριστή. Για την επίτευξη συναίνεσης μεταξύ των ομοτίμων χρησιμοποιούνται διάφορες μέθοδοι που μπορεί να βασίζονται σε ψηφοφορία, εξόρυξη και κίνητρα τιμιότητας.

Κρυπτονομίσματα

Η εφαρμογή πάντως που γνωρίζει το ευρύ κοινό, έστω κι αν δεν έχει άμεση εμπειρία και την γνωρίζει μόνο από τα μέσα ενημέρωσης, είναι τα κρυπτονομίσματα και κυρίως το Bitcoin. Μολονότι στον πραγματικό κόσμο ένα νόμισμα και ένα κατάστιχο φαίνονται να είναι δυο εντελώς διαφορετικά όντα, θα δούμε ότι το χρήμα μπορεί να το δει κανείς ως ειδική περίπτωση ενός κατάστιχου. Σήμερα είναι πιο εύκολο να το καταλάβουμε αυτό αν το σκεφτούμε ως πλαστικό χρήμα, δηλαδή ως μια σειρά από πληρωμές. Ένας εργαζόμενος εισπράττει κάθε μήνα το μισθό του μέσω μιας τράπεζας, που αφαιρεί το αντίστοιχο ποσό από μια εταιρία ή έναν οργανισμό και το μεταφέρει στον τραπεζικό λογαριασμό του. Στη συνέχεια πάει σε ένα σουπερμάρκετ, αγοράζει τρόφιμα και πληρώνει με την κάρτα του, δηλαδή δίνει μια εντολή στην τράπεζα που φιλοξενεί το λογαριασμό του να μεταφέρει ένα ποσό στον λογαριασμό του καταστήματος. Όλες αυτές οι μεταφορές δεν είναι παρά μια σειρά από εγγραφές, που περι-

[1] Η μονιμότητα π.χ. των βαβυλωνιακών πινακίδων ήταν διαφορετική, πιθανώς ανάλογα με τη σημασία τους. Οι πιο συνηθισμένες εξ αυτών αποτελούσαν απλώς ξεραμένο πηλό και το υλικό μπορούσε να ανακυκλωθεί για να ξαναχρησιμοποιηθεί, ενώ κάποιες άλλες ψήνονταν [Bla+00].

γράφουν μεταφορές ποσών ανάμεσα σε λογαριασμούς. Η τράπεζα κρατάει αυτές τις εγγραφές και φροντίζει να μην μπορούν να αλλοιωθούν.

Ωστόσο το πραγματικό (χάρτινο[2] ή μεταλλικό) χρήμα έχει μια ιδιότητα που το διαφοροποιεί από τις τραπεζικές μεταφορές και ενδιαφέρει ιδιαίτερα όσους επιθυμούν να κάνουν πιο «διακριτικές» συναλλαγές. Σε κάθε περίπτωση τα νομίσματα χρησιμοποιούνται εδώ και χιλιάδες χρόνια και οι συναλλαγές με χρήση χρήματος αποτελούν συνήθη πρακτική σε όλον τον πλανήτη. Για παράνομες όμως συναλλαγές («μαύρο χρήμα») ή συναλλαγές που γενικότερα χρειάζονται μυστικότητα αποτελούσαν ως χθες τη μοναδική λύση, μέχρι τη μέρα που εμφανίσθηκε η εναλλακτική λύση των «κρυπτονομισμάτων». Τα κρυπτονομίσματα συνίστανται σε εγγραφές πληρωμών, που έχουν όμως τη δυνατότητα να ικανοποιούν δυο εκ πρώτης όψεως αντιφατικές απαιτήσεις, αφενός να είναι αξιόπιστες, δηλαδή να μη μπορεί κανείς να τις αμφισβητήσει, αφετέρου να διατηρούν ιδιότητες μυστικότητας.

Μια πληρωμή έχει τα εξής λίγα βασικά χαρακτηριστικά: Αυτόν που πληρώνει (ή τον λογαριασμό του), αυτόν που πληρώνεται (ή τον λογαριασμό του), το ποσό και τη χρονική στιγμή. Σε μια πληρωμή με χαρτονομίσματα μια παραπάνω ιδιότητα που διατηρεί τη σημασία της σε ορισμένες περιπτώσεις είναι οι σειριακοί αριθμοί των συγκεκριμένων χαρτονομισμάτων. Θα δούμε ότι τα κρυπτονομίσματα αποκρύπτουν κατά κανόνα τα δύο πρώτα, δηλαδή τις δύο ταυτότητες.[3]

Πριν προχωρήσουμε σε μια σύντομη ανασκόπηση προσπαθειών για την δημιουργία ηλεκτρονικού χρήματος είναι σκόπιμη μια μικρή παρένθεση ορολογίας. Ο όρος *ηλεκτρονικό χρήμα* ή *ψηφιακό χρήμα* χρησιμοποιείται εν γένει σε σχέση με συναλλαγές που πραγματοποιούνται με τη χρήση υπολογιστή. Με την έννοια αυτή περιλαμβάνει και τις μεταφορές ποσών μέσω Διαδικτύου, τις πληρωμές με πιστωτικές κάρτες, αλλά και τα κρυπτονομίσματα. Δηλαδή περιλαμβάνει το ψηφιακό μέσο συναλλαγής είτε αυτό έχει αυξημένη ανωνυμία, όπως στο BitCoin, είτε μειωμένη, όπως η πληρωμή για μια αγορά προϊόντων σε ένα κατάστημα, που είναι γνωστή όχι μόνο στο κατάστημα, στον πελάτη και στην τράπεζα (που μεσολαβεί), αλλά η πλήρης ή μερική πληροφορία για την συναλλαγή διαχέεται σε κρατικές υπηρεσίες, όπως π.χ. στην εφορία. Στην περίπτωση συναλλαγής όμως με μετρητά (νομίσματα ή χαρτονομίσματα) υπάρχει ανωνυμία (εκτός αν προσημειωθούν τα χαρτονομίσματα, δηλαδή σημειωθούν οι σειριακοί τους αριθμοί), γι᾽ αυτό ο όρος *ψηφιακά μετρητά* (digital cash) ή *ηλεκτρονικά μετρητά* (electronic cash) συνήθως υποδηλώνει το μέσο μιας ηλεκτρονικής συναλλαγής που διατηρεί την ανωνυμία των συναλλασσομένων.

Οι προσπάθειες για τη δημιουργία ηλεκτρονικού χρήματος που θα διατηρεί την ιδιότητα της ανωνυμίας των εμπλεκομένων στις συναλλαγές βασίζεται εν γένει στην κρυπτογραφία και έχει ξεκινήσει από τη δεκαετία του 1980. Το 1983 ο David Chaum δημοσίευσε μια μέθοδο για ανώνυμες συναλλαγές [Cha83] βασισμένη σε υπογραφή δημόσιου κλειδιού μιας τράπεζας. Προκειμένου να εφαρμοσθεί η μέθοδος ιδρύθηκε η εταιρία *DigiCash* και το προϊόν ονομάστηκε *ecash*. Ωστόσο στις ΗΠΑ η επιτυχία του συστήματος ήταν πολύ περιορισμένη, υιοθετήθηκε μόνο από μια τράπεζα και η ζωή του έληξε το 1998, αν και στη συνέχεια βρήκε κάποια εφαρμογή σε ευρωπαϊκές τράπεζες. Στο μεταξύ το 1998 η NSA δημοσίευσε ένα άρθρο για το πώς μπορεί να

[2] Τα χαρτονομίσματα δεν είναι απαραιτήτως χάρτινα. Ανάμεσα στα υλικά που έχουν κατά καιρούς χρησιμοποιηθεί είναι πλαστικό, ύφασμα και δέρμα.

[3] Το πραγματικό χρήμα όταν βρεθεί εκ των υστέρων σε ένα πορτοφόλι ή ένα χρηματοκιβώτιο διατηρεί μόνο δύο χαρακτηριστικά από τα παραπάνω πέντε, τον παραλήπτη της πληρωμής και τους σειριακούς αριθμούς, ενώ για το ποσό το μόνο που μπορεί να διαπιστωθεί είναι ένα άνω φράγμα, που προσδιορίζεται από το συνολικό ποσό στο χρηματοκιβώτιο, ομοίως δε και για τη χρονική στιγμή ότι είναι στο παρελθόν σε σχέση με τη στιγμή της παρατήρησης.

δημιουργηθεί ένα κρυπτονόμισμα [LSS96] και την ίδια χρονιά ο Nick Szabo (επιστήμονας υπολογιστών και νομικός συγχρόνως) περιέγραψε ένα σύστημα, [Pec12] όπου ο χρήστης έπρεπε να δώσει *απόδειξη έργου* (λύνοντας με τον υπολογιστή του κρυπτογραφικά αινίγματα), μια έννοια που θα δούμε αργότερα στο BitCoin.

Το πρώτο αποκεντρωμένο κρυπτονόμισμα είναι το BitCoin που δημιουργήθηκε το 2009 από τον Satoshi Nakamoto [Nak08], για τον οποίο δεν είναι γνωστό αν είναι υπαρκτό πρόσωπο. Περισσότερα από 2000 κρυπτονομίσματα έχουν αναπτυχθεί ως σήμερα (2020), αλλά το BitCoin παραμένει το πρώτο με συνολική αξία (market capitalization) περί τα 118 δισεκατομμύρια δολλάρια, ενώ δεύτερο είναι το Ethereum με περίπου 16 δισεκατομμύρια δολλάρια.

12.2 Μέθοδοι και εργαλεία για κατανεμημένο κατάστιχο

Τα δεδομένα που περιέχονται σε ένα κατάστιχο μπορούν προφανώς να οργανωθούν με διάφορους τρόπους, στην περίπτωση όμως ενός κατανεμημένου κατάστιχου είναι δημοφιλής μια δομή που λέγεται *δέντρο Merkle*. Θα φτάσουμε σταδιακά στην περιγραφή αυτής της δομής, αρχίζοντας από τις συνδεδεμένες λίστες (linked lists).

Linked lists

Έστω διάνυσμα $(a_1, a_2, \ldots, a_n)$, όπου κάθε a_i είναι μια ακολουθία από bits. Ας υποθέσουμε ότι θέλουμε να το αποθηκεύσουμε σε μια μνήμη με γραμμική δομή, δηλαδή μια μνήμη που έχει διαδοχικές θέσεις, κάθε μια από τις οποίες είναι ένα καταχωρητής με συγκεκριμένη χωρητικότητα b (αριθμό από bits που μπορεί να δεχτεί). Αν τα παραπάνω strings a_i έχουν μήκος που δεν ξεπερνάει τα b bits η πιο απλή λύση είναι να τοποθετηθούν σε διαδοχικές θέσεις της μνήμης, π.χ. στις θέσεις από 1 ως n. Αν κάποια από αυτές τις θέσεις είναι κατειλημμένη, μπορούμε να χρησιμοποιήσουμε τις θέσεις $K+1$ ως $K+n$, με την προϋπόθεση ότι υπάρχει μια διαδοχική σειρά από άδειες θέσεις που αρχίζουν στη θέση $K+1$, αλλά πρέπει να καταγράψουμε κάπου την τιμή του K.[4] Μια τέτοια δομή γενικά λέγεται *array list*. Στην περίπτωση που το διάνυσμα πρέπει να αναθεωρηθεί προσθέτοντας ένα νέο στοιχείο d μετά το a_i, δηλαδή το νέο διάνυσμα είναι $(a_1, \ldots, a_i, d, a_{i+1}, \ldots, a_n)$, πρέπει να σπρώξουμε κατά μία θέση τα επόμενα στοιχεία a_{i+1} ως a_n προκειμένου να δημιουργηθεί μια θέση για το νέο στοιχείο d, αρχίζοντας προφανώς από το a_n. Αντίστοιχα, αν πρόκειται να σβήσουμε ένα στοιχείο, όλα τα επόμενα πρέπει να τα σπρώξουμε στις προηγούμενες θέσεις.

Μια εναλλακτική λύση είναι να χρησιμοποιηθεί μια άλλη δομή που λέγεται *συνδεδεμένη λίστα* (linked list). Στην περίπτωση αυτήν τα στοιχεία a_i μπορούν να αποθηκευτούν σε οποιαδήποτε θέση της μνήμης, αλλά καθένα συνοδεύεται από ένα δείκτη δ_i, του οποίου η τιμή είναι ίση με τον αριθμό της θέσης που φιλοξενεί το a_{i+1}. Κατά συνέπεια στη μνήμη αποθηκεύεται ένα διάνυσμα της μορφής $(a_1, \delta_1, a_2, \delta_2, \ldots, a_n, \delta_n)$ βάζοντας κάθε ζεύγος (a_i, δ_i) σε μια αυθαίρετη θέση k της μνήμης, αρκεί να τεθεί $\delta_{i-1} = k$.

Τα βασικά πλεονεκτήματα της συνδεδεμένης λίστας είναι (α) ότι μπορεί κανείς να χρησιμοποιήσει οποιεσδήποτε διαθέσιμες θέσεις, όχι υποχρεωτικά διαδοχικές και (β) ότι για την παρεμβολή ενός νέου στοιχείου οπουδήποτε στη λίστα αρκεί αυτό να προστεθεί σε κάποια διαθέσιμη θέση διορθώνοντας τον δείκτη του προηγούμενου

[4] Συνήθως αντί των K και n κρατάμε για τη λίστα δύο δείκτες, έναν που δείχνει το πρώτο στοιχείο κι έναν που δείχνει το τελευταίο.

στοιχείου και ορίζοντας σωστά τον δείκτη προς το επόμενο στοιχείο. Όμως η αποθήκευση τώρα απαιτεί εν γένει θέσεις μεγαλύτερου μήκους, π.χ. αν πρόκειται να αποθηκεύσουμε το πολύ 1024 strings των 16 bits, για την απλή λίστα χρειάζεται χώρος 16384 bits, ενώ για την συνδεδεμένη λίστα θα χρειαστούν $(16 + 10) \times 1024 = 26624$ bits. Επίσης, για την ανάγνωση του i-οστού στοιχείου χρειάζεται να αναγνωσθούν διαδοχικά οι δείκτες που συνοδεύουν όλα τα προηγούμενα $i - 1$ στοιχεία.

Ας υποτεθεί τώρα ότι είναι επιθυμητή κάποιου είδους προστασία των δεδομένων της λίστας. Η προφανής λύση είναι η κρυπτογράφηση όλης της μνήμης, οπότε χειρισμοί πάνω στη λίστα και σε όποια άλλα δεδομένα είναι πάνω στη μνήμη θα μπορούν να γίνουν μόνον από όποιον έχει το κλειδί. Αυτή βέβαια η λύση δεν είναι ιδιαίτερα αποδοτική. Γενικότερα το ζήτημα της κρυπτογράφησης μιας βάσης δεδομένων είναι πρόβλημα, δεδομένου ότι η κρυπτογράφηση δημιουργεί δυσκολίες στις αναζητήσεις [Shm+10; SK15].

Ένα ερώτημα που πρέπει να απαντηθεί πριν ληφθούν μέτρα προστασίας μιας λίστας ή ενός κατάστιχου είναι κατά πόσο είναι απαραίτητη η απόκρυψη των πληροφοριών ή αρκεί η προστασία της ακεραιότητάς τους ενώ το περιεχόμενο θα είναι φανερό. Στην περίπτωση των κρυπτονομισμάτων επικρατεί η δεύτερη άποψη, οπότε στη συνέχεια θα δώσουμε έμφαση στη διατήρηση της ακεραιότητας. Η τελευταία διασφαλίζεται με σχετικά απλό τρόπο μέσω του *δέντρου Merkle*, το οποίο είναι μια δομή που ενσωματώνει στους δείκτες της τις τιμές μιας συνάρτησης κατατερματισμού.

Κρυπτογραφικές συναρτήσεις κατακερματισμού

Μια συνάρτηση κατακερματισμού H από ένα κείμενο x (ένα string οποιουδήποτε είδους και μήκους) παράγει ένα *κωδικό κατακερματισμού* $H(x)$, δηλαδή έναν αριθμό σταθερού (και εν γένει μικρού) μήκους. Τις συναρτήσεις κατακερματισμού εξετάζουμε σε άλλο κεφάλαιο. Συνοψίζουμε ωστόσο ότι μια *κρυπτογραφική συνάρτηση κατακερματισμού* (cryptographic hash function) έχει τις εξής επιθυμητές ιδιότητες [SB18; Nar+16]:

1. Είναι εύκολος ο υπολογισμός της τιμής της για κάθε μήνυμα.
2. Είναι ανέφικτο να υπολογισθεί το μήνυμα από την τιμή κατακερματισμού (pre-image resistance).
3. Δεδομένου ενός μηνύματος m_1, είναι ανέφικτο να βρεθεί ένα άλλο μήνυμα m_2 με την ίδια τιμή κατακερματισμού, δηλ. τέτοιο ώστε $H(m_2) = H(m_1)$ (second pre-image resistance).
4. Είναι ανέφικτο να βρεθούν δύο οποιαδήποτε μηνύματα m_1, m_2 με την ίδια τιμή κατακερματισμού, δηλαδή τέτοια ώστε $H(m_2) = H(m_1)$ (αντίσταση σε σύγκρουση - collision resistance).

Πρόσθετες ιδιότητες κατάλληλες για blockchain

Στην περίπτωση, ωστόσο, των συναρτήσεων κατακερματισμού που χρησιμοποιούνται στο blockchain απαιτούνται δυο πρόσθετες ιδιότητες. Η μία εξ αυτών συνδέεται με την δεύτερη από τις παραπάνω ιδιότητες όταν τα δυνατά αρχικά κείμενα x είναι σχετικά λίγα. Για παράδειγμα αν το x παίρνει μόνο τις τιμές 0, 1, ο υπολογισμός του x όταν είναι γνωστή η τιμή του $H(x)$ είναι τετριμμένος, δοκιμάζοντας τα (0) και (1). Για να αντιμετωπισθούν τέτοιες περιπτώσεις χρησιμοποιείται μια μέθοδος παρόμοια με εκείνη του «αλατιού» στα passwords, δηλαδή σχηματίζεται το string $r\|x$ (η

ένωση των r και x σε ένα string), όπου r είναι ένας τυχαίος αριθμός (nonce) με αρκετά εκτεταμένη κατανομή τιμών, και στη συνέχεια χρησιμοποιείται το $H(r\|x)$. Η *ελάχιστη-εντροπία* (min-entropy) είναι μια έννοια της θεωρίας πληροφορίας που ορίζεται στην [Ren05] και αποδίδει πιο αυστηρά την επιθυμητή ιδιότητα του r.

Συγκεκριμένα, μια συνάρτηση κατακερματισμού καλείται *αποκρύπτουσα* (hiding) όταν ισχύουν τα εξής: Όταν δοθεί μια τιμή r με κατανομή πιθανότητας υψηλής ελάχιστης εντροπίας (min-entropy), είναι ανέφικτο από την τιμή της $H(r\|x)$ να υπολογισθεί η τιμή του x.

Η ελάχιστη εντροπία (min-entropy) είναι μια έννοια της θεωρίας πληροφορίας που ορίζεται στην [Ren05]. Πρακτικά σημαίνει ότι αν ληφθεί ένα δείγμα αριθμού της δεδομένης κατανομής, δεν είναι πιθανό να επιλεγεί κάποια συγκεκριμένη τιμή. Θα δούμε στη συνέχεια πώς αυτό υλοποιείται στην περίπτωση του BitCoin.

Η επόμενη ιδιότητα λέγεται *εύνοια σε αινίγματα* (puzzle friendliness): Μια συνάρτηση κατακερματισμού H λέγεται *ευνοϊκή σε αινίγματα* (puzzle friendly) αν για κάθε δυνατή τιμή y μήκους n bits, αν το r επιλεγεί από μια κατανομή με υψηλή ελάχιστη-εντροπία, είναι ανέφικτο να βρεθεί ένα x τέτοιο ώστε $H(r\|x) = y$ σε χρόνο σημαντικά μικρότερο από 2^n.

Σχήμα Merkle–Damgård

Στην περίπτωση που είναι διαθέσιμη μια συνάρτηση κατακερματισμού που δέχεται ως είσοδο κείμενο σταθερού μήκους, μπορεί να εφαρμοσθεί σε κείμενο αυθαιρέτου μήκους σπάζοντας το αρχικό κείμενο σε κομμάτια κατάλληλου μήκους και εφαρμόζοντας τη συνάρτηση πάνω σε κάθε κομμάτι χωριστά. Η πρακτική αυτή περιγράφεται πιο συγκεκριμένα από το em σχήμα Merkle–Damgård, το οποίο λειτουργεί ως εξής: Εφόσον το αποδεκτό μήκος εισόδου είναι m bits και το μήκος του κωδικού είναι n ($m > n$), αρχικό μήνυμα μήκους M σπάζει σε κομμάτια $x_1, x_2, \ldots x_N$ μήκους $m - n$, όπου ο αριθμός των κομματιών είναι $N = \lceil M/(m-n) \rceil$ (και το τελευταίο πρέπει πιθανώς να συμπληρωθεί για να έχει μήκος m ακριβώς). Αν IV είναι μια αρχική τιμή (initial value) μήκους n, υπολογίζονται τα εξής:

$$\begin{aligned} h_1 &= H(IV\|x_1) \\ h_2 &= H(h_1\|x_2) \\ &\cdots \\ h_N &= H(h_{N-1}\|x_N) \end{aligned}$$

Η τιμή h_N (που λόγω του παραπάνω υπολογισμού είναι συνάρτηση όλου του κειμένου x) χρησιμοποιείται ως τιμή κατακερματισμού του x. Για παράδειγμα, στον SHA-256 ισχύει $m = 768$ και $n = 256$, οπότε το αρχικό κείμενο κομματιάζεται σε κείμενα των 512 bits.

Δείκτες κατακερματισμού

Στη συνέχεια θα δούμε με ποιο τρόπο μπορούμε να εξασφαλίσουμε την ακεραιότητα των εγγραφών μια συνδεδεμένης λίστας αν μαζί με τους δείκτες αποθηκεύουμε και διαδοχικές τιμές κωδικών κατακερματισμού. Το αποτέλεσμα της ένωσης (σε ένα string) ενός δείκτη με ένα κωδικό κατακερματισμού λέγεται *δείκτης κατακερματισμού*.

Συνδεδεμένη λίστα με δείκτες κατακερματισμού

Ας υποθέσουμε ότι χρησιμοποιούμε μια μνήμη, στις θέσεις της οποίας σωρεύουμε εγγραφές $a_1, a_2, \ldots$ και την υλοποιούμε ως συνδεδεμένη λίστα με φορά δεικτών από την νεότερη εγγραφή προς την παλαιότερη: Η εκάστοτε τελευταία εγγραφή a_i συνοδεύεται από ένα δείκτη δ_i, του οποίου η τιμή είναι ίση με τη θέση της προηγούμενης εγγραφής x_{i-1}. Υιοθετώντας τον συμβολισμό $\theta : x$ που δηλώνει ότι η θέση θ της μνήμης περιέχει το string x, μπορούμε να συνοψίσουμε τα παραπάνω ως εξής:

$$\begin{aligned} \theta_1 &: x_1 \\ \theta_2 &: x_2 \| \theta_1 \\ &\ldots \\ \theta_i &: x_i \| \theta_{i-1} \end{aligned}$$

Προκειμένου να διασφαλίσουμε την ακεραιότητα των εγγραφών, κάνουμε μια προσθήκη στο προηγούμενο σχήμα, δηλαδή υπολογίζουμε κάθε φορά και συμπεριλαμβάνουμε όχι μόνο την προηγούμενη θέση, αλλά και τον κωδικό κατακερματισμού της προηγούμενης εγγραφής, εφαρμοσμένο πάνω σε όλο το string της προηγούμενης εγγραφής (δηλ. τον συνδυασμό νέου κειμένου - προηγούμενης θέσης - προηγούμενου κωδικού):

$$\begin{aligned} \theta_1 &: x_1 \\ \theta_2 &: x_2 \| \theta_1 \| H(x_1) \equiv y_2 \\ \theta_3 &: x_3 \| \theta_2 \| H(y_2) \equiv y_3 \\ &\ldots \\ \theta_i &: x_i \| \theta_{i-1} \| H(y_{i-1}) \equiv y_i \end{aligned}$$

Επίσης υπολογίζουμε τον κωδικό $H(y_i)$, του οποίου την τιμή h_i αποθηκεύουμε με ασφαλή τρόπο.

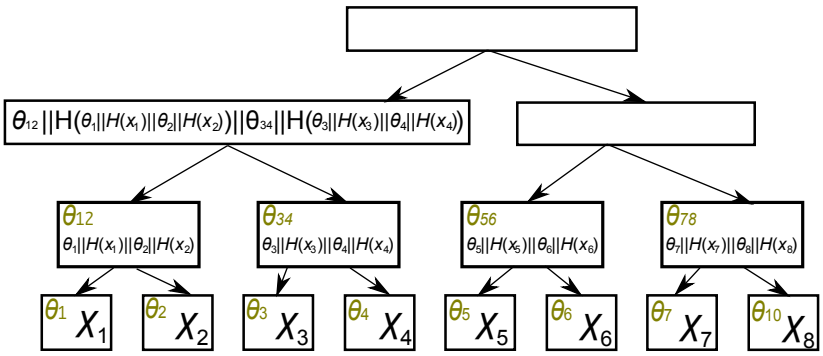
Δεδομένου ότι το παραπάνω σχήμα περιλαμβάνει τους δείκτες θέσης, μπορεί κανείς να φτάσει από την τελευταία εγγραφή σε οποιαδήποτε προηγούμενη, ακριβώς όπως στην απλή συνδεδεμένη λίστα. Αν υπάρξει υποψία αλλοίωσης της εγγραφής x_i, μπορούμε να υπολογίσουμε το $H(x_i \| \theta_{i-1} \| H(y_{i-1}))$ και να ελέγξουμε αν το αποτέλεσμα είναι ίσο με h_i. Αν ισχύει η ισότητα, είναι εγγυημένη η ακεραιότητα όχι μόνο του x_i, αλλά και του δείκτη θέσης θ_{i-1} και του κωδικού $H(y_{i-1}) = h_{i-1}$. Στη συνέχεια μπορούμε να ελέγξουμε την αμέσως προηγούμενη εγγραφή κ.ο.κ. μέχρι και την πρώτη.

Όταν εμφανισθεί ένα νέο κείμενο x_{i+1}, το ενσωματώνουμε στην επόμενη εγγραφή (σε κάποια κενή θέση θ_{i+1}) σε ένα ενωμένο string μαζί με την τιμή του θ_i και το h_i. Κατόπιν υπολογίζουμε το h_{i+1}, το οποίο αντικαθιστά το h_i στην ασφαλή φύλαξη.

Δέντρο Merkle

Το δέντρο Merkle είναι η γενίκευση της προηγούμενης δομής δεδομένων από γραμμική λίστα σε δέντρο. Για να το εξηγήσουμε αυτό θα αρχίσουμε με ένα παράδειγμα οκτών μόνο κειμένων $x_1, \ldots, x_8$, τα οποία θα υποθέσουμε ότι βρίσκονται αντίστοιχα στις θέσεις $\theta_1, \ldots, \theta_8$.

Αρχικά υπολογίζουμε τον κωδικό $h_1 = H(x_1)$ και φτιάχνουμε τον δείκτη κατακερματισμού του x_1 που είναι το string $\theta_1 \| h_1$. Στη συνέχεια φτιάχνουμε τον δείκτη



Σχήμα 12.1: Δέντρο Merkle για οκτώ εγγραφές.

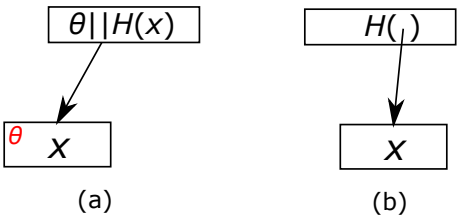
κατακερματισμού του x_2 που είναι το string $\theta_2\|h_2$. Τους δύο αυτούς δείκτες κατακερματισμού αποθηκεύουμε σε μια νέα εγγραφή, η οποία επομένως περιέχει το string $\theta_1\|h_1\|\theta_2\|h_2$. Ομοίως κάνουμε για το ζεύγος των x_3, x_4 φτιάχνοντας το ζεύγος δεικτών κατακερματισμού $\theta_3\|h_3\|\theta_4\|h_4$, καθώς και για τα άλλα δύο ζεύγη κειμένων, όπως φαίνεται στο Σχήμα 12.1.

Στη συνέχεια τους δύο δείκτες κατακερματισμού τους χειριζόμαστε ως κείμενα $y_{12} = \theta_1\|h_1\|\theta_2\|h_2$ και $y_{34} = \theta_3\|h_3\|\theta_4\|h_4$, δηλαδή κάνουμε για τα y_{12} και y_{34} ό,τι ακριβώς κάναμε για τα x_1, x_2, μολονότι δεν είναι κανονικές εγγραφές π.χ. συναλλαγών. Βεβαίως τα y_{12} και y_{34} είναι συνήθως μικρότερου μήκους από τα τα x_1, x_2, διότι περιέχουν μόνο κωδικούς κατακερματισμού και δείκτες θέσης στη μνήμη. Με δυο λόγια στη «δεύτερη γενιά» αποθηκεύουμε μόνο ζευγάρια από δείκτες κατακερματισμού, συνολικά 4 ζευγάρια. Στη συνέχεια φτιάχνουμε μια τρίτη γενιά με δείκτες που δείχνουν προς ζευγάρια δεικτών της δεύτερης γενιάς (δύο συνολικά) και τελικά ένα δείκτη τέταρτης γενιάς που δείχνει προς τους δύο δείκτες της τρίτης γενιάς. Στην ρίζα του δέντρου απομένει ένα ζεύγος δεικτών κατακερματισμού, για την αξιοπιστία όμως όλου του δέντρου αρκεί να διαφυλαχθεί μόνο ο κωδικός κατακερματισμού του string της ρίζας.

Σε ένα δέντρο Merkle, όπως στο Σχήμα 12.1, το περιεχόμενο όλων των θέσεων μνήμης για μια αυθαίρετη επιλογή θέσεων μπορεί να υπολογισθεί μοναδικά. Για τον λόγο αυτον συχνά οι απεικονίσεις τέτοιων δομών παραλείπουν τις λεπτομέρειες και αφήνουν μόνο τη δομή του δέντρου και τις αρχικές εγγραφές. Γενικά η λογική της χρήσης των δεικτών κατακερματισμού σε ένα δέντρο Merkle επεξηγείται στο Σχ. fig:MerkleTree2. Όταν ένας δείκτης «δείχνει» προς ένα block που περιέχει μια εγγραφή x και βρίσκεται στη θέση θ, ο δείκτης προς μια εγγραφή είναι της μορφής $\theta\|H(x)$, δηλαδή είναι ένα string που περιέχει τη θέση που φιλοξενεί την έγγραφή και τον κωδικό κατακερματισμού του περιεχομένου της εγγραφής. Στην περίπτωση που φεύγουν περισσότερα από ένα βέλη από ένα κόμβο, ο κόμβος δηλαδή περιέχει περισσότερους του ενός δείκτες, οι δείκτες αυτοί ενώνονται σε ένα συνεχόμενο string.

Ένα δυαδικό δέντρο μπορεί να φτιαχτεί ακόμη και αν ο αριθμός των αρχικών εγγραφών δεν είναι δύναμη του δύο, βάζοντας ορισμένα φύλλα χωρίς αδέλφια. Ωστόσο είναι επίσης εύκολο να γίνει μια γενίκευση του παραπάνω σχήματος για μη δυαδικά δέντρα.

Η οργάνωση των block σε δέντρα έχει, έναντι των γραμμικών σχημάτων, το πλεονέκτημα της ταχύτερης απόδειξης της ύπαρξης και της ακεραιότητας ενός block σε ένα blockchain, αν μια τέτοια απόδειξη χρειαστεί. Για την απόδειξη αρκεί ο έλεγχος



Σχήμα 12.2: Η έννοια του δείκτη κατακερματισμού σε δέντρο Merkle. Στο (a) φαίνεται πώς σχηματίζεται ένας δείκτης κατακερματισμού ως string αποτελούμενο από ένα δείκτη και ένα κωδικό κατακερματισμού. Στο (b) φαίνεται μια απλοποιημένη συμβολική παράσταση του (a) που θα χρησιμοποιηθεί στη συνέχεια.

στο μονοπάτι από το συγκεκριμένο block ως τη ρίζα, δηλαδή σε γενικές γραμμές αρκεί να ελεγχθούν $\log n$ κόμβοι αντί για n.

Ας υποτεθεί ότι τα φύλλα του δέντρου είναι διατεταγμένα με κάποιο κανόνα, π.χ. λεξικογραφικό. Στην περίπτωση αυτή είναι επίσης δυνατή η απόδειξη ότι ένα block δεν είναι μέλος ενός blockchain. Αυτή μπορεί να γίνει δείχνοντας ότι υπάρχουν (όπως στην προηγ. παράγραφο) δύο διαδοχικά block, ανάμεσα στα οποία θα έπεφτε σύμφωνα με τη διάταξη, αν υπήρχε, το υπό εξέταση block.

Συμπέρασμα

Είδαμε ότι ένα σύνολο κειμένων, που οργανώνονται σε "blocks" μπορούν να «δεθούν» μεταξύ τους μέσω δεικτών κατακερματισμού και να αποτελέσουν μια αλυσίδα ή ένα δέντρο, έτσι ώστε να είναι δύσκολο να γίνει αλλοίωσή τους σε οποιοδήποτε σημείο. Κάθε τέτοιο block μπορεί να περιέχει π.χ. εγγραφές συναλλαγών και μάλιστα χωρίς περιορισμό στο πλήθος των εγγραφών και τελικά στον όγκο του block. Επίσης είδαμε ότι με την οργάνωση σε δέντρα Merkle είναι δυνατοί έλεγχοι με λογαριθμικής τάξης αριθμό πράξεων ως προς το πλήθος των blocks.

Στο σημείο αυτό θα μπορούσε πιθανώς κάποιος να θέσει το ερώτημα για ποιο λόγο δεν βάζουμε όλες τις εγγραφές σε ένα και μοναδικό block, του οποίου τον κωδικό κατακερματισμού στη συνέχεια θα αποθηκεύσουμε ασφαλώς. Η πρώτη απάντηση που μας έρχεται στο μυαλό ίσως είναι ότι επιθυμούμε να σχεδιάσουμε ένα σύστημα, όπου οι εγγραφές θα συσσωρεύονται με το χρόνο, οπότε θα έπρεπε να επαναϋπολογίζεται ο κωδικός κάθε φορά που προστίθεται μια εγγραφή ή έστω ένας προκαθορισμένος αριθμός εγγραφών. Αν όμως θυμηθούμε το σχήμα Merkle–Damgård θα αντιληφθούμε ότι η προσθήκη εγγραφών δεν είναι σπουδαίο πρόβλημα, διότι μέσω αυτού του σχήματος είναι δυνατή κάθε φορά η προσθήκη ενός κομματιούύ και ο κωδικός προσδιορίζεται βάσει του προηγ. κωδικού. Βέβαια το σχήμα αυτό ουσιαστικά δεν είναι πολύ διαφορετικό από τη δημιουργία μιας αλυσίδας από κομμάτια με διαδοχικούς κωδικούς όπως στο blockchain, με τη διαφορά ότι το τελευταίο βασίζεται σε συνδεδεμένη λίστα και ασφαλίζει μέσω του κωδικού και την θέση κάθε block.

Κατόπιν αυτών η κύρια απάντηση στο ερώτημα της προηγούμενης παραγράφου (γιατί να μη άζουμε όλες τις εγγραφές σε ένα και μοναδικό block) είναι ότι βασικός στόχος του blockchain είναι να εξυπηρετήσει τη διασφάλιση της αξιοπιστίας σε ένα κατανεμημένο περιβάλλον, όπου ένα νέο block μπορεί να σχηματισθεί και να προστεθεί στο συνολικό blockchain από διαφορετικούς παίκτες, με τη χρήση βεβαίως κανόνων αποδοχής,

Κανείς δεν αποκλείει τη χρήση του blockchain σε μη κατανεμημένο περιβάλλον, προκειμένου να διασφαλίζεται με τη χρήση μιας συνάρτησης κατακερματισμού ένας πληθυσμός εγγραφών που αυξάνεται, όπως π.χ. σε μια τράπεζα. Στην περίπτωση μάλιστα αυτή επιλύεται αμέσως το πρόβλημα της ύπαρξης μιας οντότητας που θα αποθηκεύει ασφαλώς τον κωδικό κατακερματισμού που σχηματίζεται στη ρίζα. Ωστόσο στην περίπτωση του μοναδικού ή κεντρικού διαχειριστή το βασικό πλεονέκτημα που παρέχουν τα δέντρα Merkle έναντι άλλων δομών είναι η οικονομία στους υπολογισμούς.

Είδαμε επομένως ότι το blockchain μπορεί να αποτελέσει το βασικό εργαλείο για τη δημιουργία ενός κατάστιχου, κατανεμημένου ή μη. Στη συνέχεια θα εξετάσουμε δύο βασικά ζητήματα που είναι εγείρονται στην περίπτωση ενός κρυπτονομίσματος, όπου υποθέτουμε ότι δεν υπάρχει κεντρικός ρυθμιστής: (α) Πώς θα διασφαλισθεί η ανωνυμία των συναλλασσομένων και (β) πώς θα διασφαλισθεί η αξιοπιστία του συνολικού συστήματος εφόσον εμπλέκονται πολλοί ισότιμοι παίκτες.

12.3 Ανωνυμία και ψηφιακές υπογραφές

Ας αναφέρουμε για άλλη μια φορά ότι (α) η ανωνυμία είναι βασική επιθυμητή ιδιότητα στα ψηφιακά μετρητά και ότι (β) υπάρχουν διάφοροι σχεδιασμοί που μπορούν να διασφαλίσουν την ανωνυμία (όπως π.χ. στην περίπτωση του *ecash* που αναφέρθηκε πιο πάνω. Η διαφορά που εισήγαγε το BitCoin είναι κυρίως η έλλειψη κεντρικής ρύθμισης, οπότε και η ανωνυμία πρέπει να εξασφαλισθεί σε ένα κατανεμημένο σύστημα ισοτίμων παικτών.

Γενικά περί ανωνυμίας στις συναλλαγές

Η απαίτηση για ανωνυμία σε μια συναλλαγή, όπου το χρήμα δεν έχει υλική μορφή τέτοια που να αποθηκεύεται σε ένα πορτοφόλι, αλλά απλώς καταγράφεται κάπου ως ποσόν, εμπεριέχει μια αντίφαση: Κάπου δίπλα στο ποσό πρέπει σε ένα σχετικά ασφαλές κατάστιχο να σημειώνεται το όνομα του κατόχου του, αλλά ταυτόχρονα η πληροφορία αυτή πρέπει να μένει γνωστή σε περιορισμένο κύκλο. Μάλιστα η ίδια η γέννηση των χαρτονομισμάτων οφείλεται σε τέτοιες πρακτικές. Στο μεσαίωνα ο ταξιδιώτης έπρεπε ακόμη να μεταφέρει τα χρήματά του σε ένα πορτοφόλι κινδυνεύοντας να πέσει θύμα ληστείας. Γύρω στο 1150 οι Ναΐτες Ιππότες προσέφεραν στους προσκυνητές των Αγίων Τόπων την εξής νέα τότε υπηρεσία: Ο ταξιδιώτης μπορούσε να καταθέσει ένα ποσό στο κοντινότερο γραφείο τους πριν την αναχώρηση, να παραλάβει ένα σημείωμα, να κάνει το επικίνδυνο ταξίδι και στον προορισμό με την επίδειξη αυτού του σημειώματος να παραλάβει το ίδιο ποσό από το γραφείο των Ναϊτών στους Αγίους Τόπους [BN09]. Ένα τέτοιο σημείωμα μπορεί στη συνέχεια να. μεταβιβασθεί σε τρίτον μέχρι την τελική εξαργύρωση, πρακτική που σήμερα ακολουθείται με τις *συναλλαγματικές*. Στο επόμενο βήμα το σημείωμα μπορεί να μην αναγράφει το όνομα του αρχικού καταθέτη, αρκεί να είναι γραμμένο έτσι που να μπορεί να γίνει κάποιος έλεγχος αξιοπιστίας, π.χ. να είναι σφραγισμένο με τη βούλα του τραπεζίτη που το εκδίδει (ένα τέτοιο σημείωμα δεν είναι παρά το σημερινό χαρτονόμισμα). Ένα από τα μειονεκτήματα αυτής της πρακτικής ανωνυμίας είναι ότι η απώλεια του σημειώματος γίνεται πιο οδυνηρή.

Στο πρόβλημα της καταγραφής της ιδιοκτησίας ενός ποσού με ταυτόχρονη διατήρηση της ανωνυμίας του ιδιοκτήτη έχουν δοθεί κατά καιρούς διάφορες λύσεις, τόσο πριν την εποχή της πληροφορικής, όσο και μετά. Για παράδειγμα, στην κλασσική λύση

του «αριθμημένου» τραπεζικού λογαριασμού (συνήθως σε μια χώρα «φορολογικό παράδεισο») η ταυτότητα του κατόχου του λογαριασμού αντικαθίσταται από αριθμό (με πολλά ψηφία) που τον γνωρίζουν μόνον ο πελάτης και επιλεγμένοι τραπεζικοί υπάλληλοι. Ο πελάτης μπορεί να κάνει αναλήψεις εφόσον θυμάται τον αριθμό. Ωστόσο η ταυτότητα του πελάτη εν γένει καταγράφεται σε απόρρητο αρχείο και η τράπεζα μπορεί είτε να πιεσθεί από τις αρχές να την αποκαλύψει είτε να πέσει θύμα διαρροής.

Ωστόσο στις παραδοσιακές λύσεις με περισσότερη ή λιγότερη ανωνυμία οι δύο εμπλεκόμενοι παίκτες είναι η «τράπεζα» και ο «πελάτης». Βασική προϋπόθεση είναι ότι η τράπεζα είναι αξιόπιστη, πράγμα που μπορεί να ισχύει ή να μην ισχύει ή να υπόκειται σε διακυμάνσεις της φερεγγυότητας και της ικανότητας διαφύλαξης της ανωνυμίας. Στις συνήθεις σημερινές τράπεζες ο κοινός πελάτης τελεί υπό καθεστώς μερικής ανωνυμίας, παρ' όλο που σχεδόν όλοι οι πελάτες θα επιθυμούσαν περισσότερη ανωνυμία, καθένας για τους λόγους του. Δεν είναι επομένως περίεργο που κάποιοι ειδήμονες της ασφάλειας προσπάθησαν να αναπτύξουν συστήματα που δεν εξαρτώνται από ένα κεντρικό διαχειριστή και από την ικανότητά του να παρέχει εχέγγυα αξιοπιστίας και ανωνυμίας.

Ωστόσο το ζήτημα της ανωνυμίας στις συναλλαγές είναι γενικότερο από την ανωνυμία των τραπεζικών λογαριασμών. Ας υποθέσουμε ότι κάποιος κάνει μια σειρά από συναλλαγές με χαρτονομίσματα που προέρχονται από το χρηματοκιβώτιό του ή καταλήγουν σ' αυτό και ότι για τις συναλλαγές δεν εκδίδονται κάποια παραστατικά. Οι συναλλαγές αυτές είναι αρκετά δύσκολο να ιχνηλατηθούν. Στη συνέχεια θα συγκρίνουμε την ανωνυμία μιας ψηφιακής λύσης με αυτήν που επιτυγχάνεται με το κλασσικό χαρτονόμισμα.

Στην περίπτωση όμως που οι συναλλαγές γίνονται μέσω ενός τραπεζικού λογαριασμού ή μιας κάρτας αρκεί να συσχετισθούν οι κινήσεις για να προσδιορισθεί ο κάτοχος του λογαριασμού.[5] Θα δούμε στη συνέχεια ότι το BitCoin παρέχει τη δυνατότητα πραγματοποίησης κινήσεων με «ψευδώνυμο» που μπορεί να μεταβάλλεται όσο συχνά επιθυμεί ο συναλλασσόμενος, ακόμη και σε κάθε νέα κίνηση. Τα κρυπτονομίσματα προσπαθούν να επιλύσουν το πρόβλημα της ανωνυμίας χωρίς συγχρόνως να θέτουν σε κίνδυνο την αξιοπιστία μιας συναλλαγής.

Η λύση μέσω κρυπτογραφίας δημόσιου κλειδιού

Η λύση είναι σχετικά εύκολη αν χρησιμοποιηθεί κρυπτογραφία δημόσιου κλειδιού. Η βασική ιδέα σε ένα κρυπτονόμισμα είναι η τήρηση ενός καταστίχου με εγγραφές, όπου κάθε εγγραφή περιγράφει μια συναλλαγή, π.χ. ένα string ως εξής: *Ο Bob πλήρωσε στην Alice 10 μονάδες*. Για να είναι αυτό το string μια εγγραφή στα πλαίσια ενός κρυπτονομίσματος πρέπει να υπάρχουν διαβεβαιώσεις για δύο ζητήματα: (α) ότι η εγγραφή είναι αξιόπιστη (δηλ. ότι έχει γίνει η πληρωμή) και (β) ότι έχει γίνει από τον Bob. Δεδομένου ότι ο Bob επιθυμεί να μη φαίνεται το πραγματικό του όνομα, πρέπει να μπορεί να αποδείξει ότι αυτός είναι που έκανε την πληρωμή και όχι κάποιος άλλος.

[5] Τα προβλήματα στην ανωνυμία εξ αιτίας του πιθανού συσχετισμού συναλλαγών που γίνονται με χρήση μοναδικού αριθμού είναι γνωστά από δεκαετίες. Σε διδακτικό βιβλίο βάσεων δεδομένων της δεκαετίας του '80 αναφέρεται το εξής παράδειγμα: Ο κύριος X χρησιμοποιεί μια πιστωτική κάρτα. Για κάθε αγορά καταγράφεται σε ποια εταιρία καταλήγει η πληρωμή και όλες οι πληρωμές εμφανίζονται στο μηνιαίο κατάλογο πληρωμών του πελάτη. Ο X μετά το ωράριο της εργασίας του περνάει από ένα κατάστημα και αγοράζει ένα γυναικείο εσώρρουχο, όχι στο νούμερο της συζύγου του (όπως προκύπτει από άλλες αγορές). Στη συνέχεια πληρώνει ένα εισιτήριο λεωφορείου, γραμμής που δεν περνάει από το σπίτι του, ενώ αργότερα πληρώνει άλλο ένα εισιτήριο της ίδιας γραμμής. Άρα απατάει τη σύζυγό του.

Μια λύση είναι (1) να υπογράψει με την ψηφιακή του υπογραφή την εγγραφή και (2) να χρησιμοποιήσει στη θέση του ονόματός του το δημόσιο κλειδί που έχει χρησιμοποιήσει στην ψηφιακή υπογραφή. Δηλαδή ο Bob μπορεί να κάνει τα εξής βήματα:

1. Δημιουργεί ένα ζευγάρι δημόσιου και ιδιωτικού κλειδιού $pk_{\text{Bob}}, sk_{\text{Bob}}$ (public key, secret key) για τον μη συμμετρικό αλγόριθμο κρυπτογράφησης E (με αποκρυπτογράφηση D).

2. Παίρνει το αρχικό string, παραλείπει το όνομά του και αντικαθιστά το όνομα της Alice (το οποίο μπορεί και να μην το γνωρίζει ούτως ή άλλως, όπως θα δούμε) με το δημόσιο κλειδί της, δηλαδή το νέο string είναι $x =$ *πλήρωσε 10 στον* pk_{Alice} και το κρυπτογραφεί με το ιδιωτικό του κλειδί, δηλαδή υπολογίζει το $y = D_{sk_{\text{Bob}}}(x)$.

3. Στέλνει στο «σύστημα» την εγγραφή $z = (pk_{\text{Bob}}, y)$.

Το «σύστημα», δηλαδή οποιοσδήποτε «τρίτος» ή ακόμη και η Alice, εφόσον θέλει να διαβάσει την εγγραφή, μπορεί να πάρει το δημόσιο κλειδί του Bob pk_{Bob} και να το εφαρμόσει πάνω στο y για να το αποκρυπτογραφήσει και να πάρει το $x = D_{pk_{\text{Bob}}}(y)$. Ο «τρίτος» από το πρώτο μέρος της εγγραφής z ξέρει ότι η πληρωμή έγινε από αυτόν που έχει το δημόσιο κλειδί pk_{Bob}, ενώ μετά την αποκρυπτογράφηση του δεύτερου μέρους ξέρει ότι το συγκεκριμένο ποσό πήγε σε όποιον έχει το δημόσιο κλειδί pk_{Alice}.

Με την παραπάνω μέθοδο ο Bob έχει επιτύχει ένα διπλό στόχο, δηλαδή αφενός έχει αποδείξει ότι αυτός (που έχει το αντίστοιχο ιδιωτικό κλειδί) είναι που έχει πληρώσει το ποσό, αφετέρου δεν έχει χρησιμοποιήσει το πραγματικό του όνομα. Περαιτέρω η Alice ανά πάσα στιγμή μπορεί να αποδείξει πως αυτή ήταν η παραλήπτρια επειδή διαθέτει στην «κλειδοθήκη» της το σωστό ιδιωτικό κλειδί sk_{Alice} που αντιστοιχεί στο pk_{Alice}. Στη συνέχεια μπορεί να κάνει μια δική της πληρωμή προς άλλον με το ίδιο ή με άλλο δημόσιο κλειδί, εφόσον μπορεί να αποδείξει ότι υπόλοιπό της καλύπτει το ποσό που θα χρησιμοποιήσει.

Στο BitCoin γίνεται ακριβώς χρήση του δημόσιου κλειδιού στην θέση της ταυτότητας ενός συναλλασσόμενου, χωρίς περιορισμό στον αριθμό διαφορετικών κλειδιών που μπορεί να χρησιμοποιήσει το ίδιο άτομο. Στην ορολογία του BitCoin το δημόσιο κλειδί που χρησιμοποιείται με αυτόν τον τρόπο λέγεται *address*.

12.4 Διάφορα απλά μοντέλα ψηφιακού χρήματος

Το βασικό συγκεντρωτικό μοντέλο σήμερα είναι αυτό που εφαρμόζουν οι τράπεζες. Οι συναλλαγές και άλλες πράξεις καταχωρούνται ως εγγραφές σε μια βάση δεδομένων. Οι εγγραφές ελέγχονται για την τήρηση των κανόνων από μηχανές και ανθρώπους που ανήκουν στην τράπεζα και έχουν κάποια εχέγγυα αξιοπιστίας. Τα θέματα ανωνυμίας τα σχολιάσαμε ήδη πιο πάνω και γενικά στο τραπεζικό σύστημα η ανωνυμία του πελάτη είναι πολύ περιορισμένη. Το μοντέλο αυτό είναι όχι μόνο συγκεντρωτικό στις λειτουργίες του, αλλά και βασίζεται στην ύπαρξη ενός και μοναδικού φορέα εμπιστοσύνης που είναι η τράπεζα. Η εμπιστοσύνη είναι αμφίπλευρη: Ο πελάτης συναλλάσσεται με μια τράπεζα που θεωρεί αξιόπιστη και μια τράπεζα «ανοίγει λογαριασμούς» μόνο σε πελάτες των οποίων τα στοιχεία έχει ελέγξει σε κάποιο βαθμό.

Περαιτέρω ορισμένες τράπεζες (συνήθως οι κεντρικές κάθε κράτους) εκδίδουν χρήμα (μεταλλικό και χάρτινο), ενώ τράπεζες και άλλοι οργανισμοί εκδίδουν μια σειρά

από «προϊόντα» (π.χ. ομόλογα, μετοχές κ.λπ.), πολλά από τα οποία είναι μεταβιβάσιμα και επέχουν περισσότερο ή λιγότερο τη θέση χρήματος. Με δυο λόγια οι τράπεζες και άλλοι οργανισμοί διαχειρίζονται αξίες με διάφορους τρόπους και οι ενέργειές τους δεν εξαντλούνται στην αποθήκευση και μεταφορά χρήματος, ψηφιακού ή φυσικού.

Τα κρυπτονομίσματα αποσκοπούν στην δημιουργία ενός συστήματος που σε κάποιο βαθμό θα μπορεί να υποκαταστήσει το τραπεζικό και το οποίο θα χαρακτηρίζεται από δύο βασικές ιδιότητες, (α) την διανομή της ευθύνης για την αξιοπιστία σε όλους τους εμπλεκόμενους χωρίς να υπάρχει ένας κεντρικός ρυθμιστικός οργανισμός και (β) την ανωνυμία των εμπλεκομένων σε συναλλαγές.

Το μοντέλο που χρησιμοποιείται από τα κρυπτονομίσματα είναι αυτό του *κατανεμημένου κατάστιχου*, όπου ο όρος υποδηλώνει όχι μόνο ότι οι εγγραφές είναι διαθέσιμες σε πολλά σημεία ενός δικτύου, αλλά και ότι οι διαχειριστές των κόμβων είναι ισότιμοι (peers). Το μοντέλο πρέπει να μπορεί να δίνει απαντήσεις σε μια σειρά από ερωτήματα:

1. Πού και πώς καταγράφονται οι εγγραφές και από ποιους είναι ορατές;
2. Πώς σημειώνονται σε μια εγγραφή οι δύο πλευρές μιας συναλλαγής έτσι ώστε να διασφαλίζονται ταυτοχρόνως η αξιοπιστία και η ανωνυμία.
3. Δεδομένου ότι υπεύθυνοι για την αξιοπιστία των συναλλαγών είναι οι ισότιμοι διαχειριστές, ποιος είναι ο κατάλληλος μηχανισμός σωστής τήρησης του κατάστιχου και ελέγχου της αξιοπιστίας;

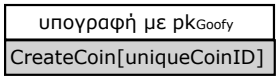
Η αξιοπιστία μιας συναλλαγής αφορά στο να μη μπορούν να αλλοιωθούν τα στοιχεία της (εμπλεκόμενοι, ποσά, είδος συναλλαγής) ή να διαγραφεί η συναλλαγή. Επίσης πρέπει να μη μπορούν να εισχωρήσουν στο σύστημα ψευδείς συναλλαγές.

Στο βιβλίο του [Nar+16] ο A. Narayanan πριν την πλήρη έκθεση της λειτουργίας του BitCoin προτείνει δύο ατελή κρυπτονομίσματα ως αρχικές ασκήσεις. Το πρώτο εξ αυτών και πιο πρωτόγονο είναι το *GoofyCoin*.

GoofyCoin

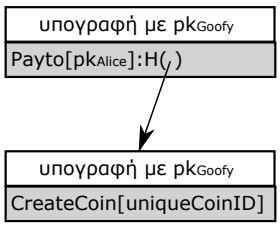
Αυτό το υποτυπώδες παράδειγμα νομίσματος αποσκοπεί στο να εξηγήσει πώς καταγράφονται οι συναλλαγές με τη χρήση δεικτών κατακερματισμού και εξηγεί ένα βασικό πρόβλημα που πρέπει να επιλυθεί, αυτό της απάτης με διπλή δαπάνη του ίδιου ποσού.

Ο Goofy έχει την ικανότητα να δημιουργεί ένα νέο νόμισμα με κάποια ταυτότητα, έστω `uniqueCoinID`, μέσω της εντολής `CreateCoin`, δηλαδή κατασκευάζει ένα string της μορφής `CreateCoin[uniqueCoinID]`. Στη συνέχεια ο Goofy υπογράφει με ένα δημόσιο κλειδί του pk_{Goofy} το παραπάνω string (δηλαδή αν το string είναι x το κρυπτογραφεί με τον D και βρίσκει το $D_{pk_{\text{Goofy}}}(x)$). Ας υποτεθεί ότι για το τελικό string που έχει δημιουργηθεί με αυτόν τον τρόπο χρησιμοποιούμε το εξής εικονίδιο:



Στη συνέχεια ο Goofy μπορεί να πληρώσει την Alice με μια εντολή `Payto` βάζοντας ως παραλήπτη το δημόσιο κλειδί της Alice και προσθέτοντας στη θέση του ποσού μια αναφορά μέσω δείκτη κατακερματισμού στην αρχική εγγραφή γέννησης του ποσού.

Τελικά θα υπογράψει το σχετικό string με το δικό του δημόσιο κλειδί. Με το συμβολισμό του Σχ. 12.2 αυτό μπορεί να παρασταθεί ως εξής:



Στη συνέχεια η Alice μπορεί να δημιουργήσει μια νέα εγγραφή πληρωμής προς ένα τρίτο άτομο, π.χ. τον Bob, με την ίδια μέθοδο, δηλαδή δημιουργώντας ένα string που θα έχει την εντολή πληρωμής στο δημόσιο κλειδί του Bob και θα περιλαμβάνει ένα δείκτη κατακερματισμού προς την π[ληρωμή με την οποία απέκτησε το νόμισμα από τον Goofy. Το string αυτό στο τέλος πρέπει να υπογραφεί από την Alice.

Στην έκθεση αυτού του πρώτου κρυπτονομίσματος έχουμε παραλείψει ορισμένα ζητήματα:

- Με ποιο δικαίωμα ο Scrooge έχει δημιουργήσει ένα νόμισμα και γιατί οι άλλοι θα το δεχτούν ως μέσο συναλλαγής;
- Πώς θα γίνει μια πληρωμή με π.χ. μισό νόμισμα, δηλαδή πώς η Alice όταν αποκτήσει ένα νόμισμα θα μπορέσει να δώσει το μισό στον Bob και το άλλο μισό στην Trudy;
- Σε ποιο κατάστιχο σημειώνονται αυτές οι εγγραφές και από ποιους είναι ορατές;

Το ζήτημα ποιος βλέπει και ποιος ελέγχει μια εγγραφή είναι σημαντικό γιατί κάτω από ορισμένες συνθήκες ένας συναλλασσόμενος μπορεί να αποπειραθεί να ξοδέψει το ίδιο ποσό δυο φορές πληρώνοντας με το ίδιο νόμισμα δύο διαφορετικούς παραλήπτες. Η εύκολη αντιγραφή ενός ψηφιακού αντικειμένου είναι γνωστό ζήτημα στον ψηφιακό κόσμο και εδώ εκδηλώνεται στην περίπτωση του ψηφιακού νομίσματος. Έστω ότι η Alice κάνει μια πληρωμή προς τον Bob. δείχνοντας την είσπραξη του ποσού από τον Goofy και κατόπιν κάνει μια δεύτερη πληρωμή του ίδιου ποσού προς την Trudy, δηλαδή δείχνει και πάλι την ίδια είσπραξη από τον Goofy. Εν πάση περιπτώσει αυτό το παράδειγμα δείχνει ότι η ακεραιότητα του δέντρου Merkle που δημιουργείται με αυτόν τον τρόπο δεν είναι αρκετή για να εμποδίσει διπλές πληρωμές.

Η λύση προφανώς περνάει μέσα από τη δημιουργία ενός μηχανισμού ελέγχου των συναλλαγών. Στο επόμενο παράδειγμα ο έλεγχος αυτός γίνεται κεντρικά.

ScroogeCoin

Το ScroogeCoin μοιάζει με το GoofyCoin, αλλά προσθέτει έλεγχο των συναλλαγών. Στη θέση του Scrooge είναι ο Goofy, που έχει δημιουργήσει το αρχικό νόμισμα. Ο Scrooge αποθηκεύει κάθε επόμενη συναλλαγή σε ένα blockchain (με μια συναλλαγή ανά block) μόνο αφού την ελέγξει (για προβλήματα, στα οποία περιλαμβάνεται και η περίπτωση διπλής δαπάνης) και υπογράφει ο ίδιος ψηφιακά το κάθε block.

Αρχικά νέα νομίσματα μπορούν να δημιουργηθούν μόνο από τον Scrooge κατά βούληση. Περαιτέρω το ScroogeCoin επιλύει το ζήτημα της υποδιαίρεσης ενός ποσού με μια νέα εντολή, μέσω της οποίας ήδη υπάρχοντα νομίσματα καταναλώνονται και στη θέση τους δημουργούνται δύο ή περισσότερα νομίσματα ίσης συνολικής αξίας με

τα αρχικά. Τα νέα νομίσματα περαιτέρω μεταβιβάζονται σε αντίστοιχους παραλήπτες. Εάν η Alice θέλει να κάνει μια πληρωμή στον Bob με μέρος της αξίας ενός νομίσματος που κατέχει, μπορεί να το σπάσει σε δύο νομίσματα και να πληρώσει με το ένα τον Bob και να πληρώσει με το άλλο τον εαυτό της.

Το ScroogeCoin επιλύει το πρόβλημα της διπλής δαπάνης (και πιθανώς άλλα προβλήματα), αλλά μόνο μέσω συγκεντρωτικής λύσης, η οποία δεν είναι επιθυμητή στα κρυπτονομίσματα.

12.5 BitCoin

Είδαμε προηγουμένως ότι ένα ψηφιακό νόμισμα θα μπορούσε να υλοποιείται μέσα από την καταγραφή μιας σειράς συναλλαγών, των οποίων η εγκυρότητα πρέπει να εξασφαλίζεται από ένα μηχανισμό ελέγχου. Οι σχεδιαστές των κρυπτονομισμάτων προσπαθούν να αποφύγουν την ύπαρξη ενός κεντρικού ελεγκτή, όπως μιας τράπεζας ή άλλου υποκατάστατου και να βρουν ένα τρόπο με τον οποίο την εγγύηση της νομιμότητας των συναλλαγών (δηλαδή της συμμόρφωσης με ένα αποδεκτό σύνολο κανόνων) θα παρέχει η ίδια η κοινότητα των χρηστών του κρυπτονομίσματος. Τέτοια αποκεντρωμένα συστήματα είναι γνωστά ως συστήματα *ισοτίμων* (peer-to-peer). Τα συστήματα αυτά έγιναν απότομα δημοφιλή μετά την επιτυχία του Napster, μιας εφαρμογής διανομής μουσικών κομματιών (1999). Πολλά από τα συστήματα διανομής αρχείων μέσα από δίκτυα ισοτίμων αντιμετώπισαν νομικά ζητήματα, κυρίως σχετιζόμενα με τα δικαιώματα πνευματικής ιδιοκτησίας των ανταλλασσομένων αρχείων. Γενικά ο σχεδιασμός και η χρήση τέτοιων αποκεντρωμένων συστημάτων αντανακλά συχνά και μια κοινωνική στάση αντίδρασης ενάντια στον συγκεντρωτισμό και στην καθιερωμένη οικονομία της αγοράς. Στην περίπτωση των κρυπτονομισμάτων ο στόχος είναι η παράκαμψη του τραπεζικού συστήματος.

Πρωτόκολλα συναίνεσης

Ωστόσο η συνεργασία οντοτήτων ή «πρακτόρων» που ενεργούν μέσα από ένα δικτυακό περιβάλλον προϋποθέτει την επίλυση του προβλήματος της συναίνεσης (consensus). Το πρόβλημα της συναίνεσης είναι ένα γνωστό πρόβλημα [OM04; FAT05; CV17; SSS17] στην περιοχή της επιστήμης των υπολογιστών. Συνίσταται στο να κατορθώσουν οι πράκτορες να συμφωνήσουν σε ορισμένα δεδομένα που τους χρειάζονται στη διάρκεια ενός υπολογισμού. Το πρόβλημα μπορεί να διαφοροποιείται ανάλογα με τον τύπο των αστοχιών (failures), στις οποίες υπόκεινται οι πράκτορες. Στην περίπτωση της *βυζαντινής* αστοχίας ένας πράκτορας μπορεί να μην έχει απλώς κάνει λάθος, αλλά να προσπαθεί να παραπλανήσει τους υπόλοιπους.[6]

Η συναίνεση δεν είναι το ίδιο σημαντική σε όλα τα δίκτυα ισοτίμων. Για παράδειγμα στο αρχικό Napster τα μέλη του δικτύου μπορούσαν να ενδιαφέρονται για την ποιότητα ψηφιοποίησης ενός μουσικού κομματιού που προσέφερε ένα συγκεκριμένο μέλος, καθώς και για τις δυνατές ταχύτητες επικοινωνίας, οπότε αυτές οι δύο ιδιότητες ήταν αρκετές για να χαρακτηρίσουν την καλή φήμη ενός μέλους του δικτύου.

[6] Εδώ έχουμε άλλη μια περίπτωση, όπου ο σχεδιαστής ενός συστήματος πρέπει να προβλέψει πώς θα αντιμετωπίσει έναν αντίπαλο που μπορεί να είναι κακόβουλος, γεγονός εξ άλλου που ισχύει για όλην την περιοχή της ασφάλειας. Ο Norbert Wiener στην περιοχή της *Κυβερνητικής* που ο ίδιος είχε ορίσει διέκρινε ανάμεσα στον *Αυγουστινιανό διάβολο* και τον *Μανιχαϊστικό διάβολο* [Gal94]. Ο πρώτος είναι αθώος αλλά περίπλοκος, είναι μια μεταφορά για την πολυπλοκότητα ενός συστήματος που καλείται να αντιμετωπίσει ένας επιστήμονας, π.χ. ο φυσικός επιστήμονας πρέπει να ανακαλύψει τα μυστικά της φύσης. Η φύση όμως δεν χρησιμοποιεί μέσα παραπλάνησης του επιστήμονα. Ο δεύτερος είναι ακριβώς ο αντίπαλος που θα χρησιμοποιήσει τέτοιες μεθόδους.

Μεμονωμένες περίεργες συμπεριφορές, όπως το να ανεβάσει κάποιος ένα αρχείο με θόρυβο ενώ το τιτλοφορεί ως τραγούδι των Beatles, δεν αποτελούσαν σοβαρό πρόβλημα.

Το πρόβλημα των Βυζαντινών στρατηγών

Τα προβλήματα συναίνεσης δεν είναι οπωσδήποτε επιλύσιμα. Στο πρόβλημα των *βυζαντινών στρατηγών* (Byzantine generals problem) [LSP82] που πρέπει να συμφωνήσουν στη δράση που θα αναλάβουν, αποδεικνύεται ότι αρκεί να είναι αναξιόπιστο[7] το ένα τρίτο για να μη μπορούν να φτάσουν σε συμφωνία.

Το αντικείμενο πάνω στο οποίο πρέπει να συμφωνήσουν οι εμπλεκόμενοι σε ένα κρυπτονόμισμα είναι ότι ένα σύνολο εγγραφών (συναλλαγών) είναι συνολικά αξιόπιστο. Οι πληρωμές θα μπορούσαν να καταχωρούνται και να ελέγχονται μία μία σε ένα κατάστιχο, πρακτικά όμως ομαδοποιούνται σε blocks πολλών εγγραφών. Κάθε block όταν περάσει τον έλεγχο επιτυχώς ενώνεται με το σύνολο όλων των προηγούμενων έγκυρων blocks σε ένα δέντρο Merkle.

Βυζαντινή ανοχή σε σφάλματα

Η περιοχή της ασφάλειας γενικά αφορά στην άμυνα απέναντι σε όχι τυχαία σφάλματα, αλλά σε δυσλειτουργίες που προκαλούνται εμπροθέτως από κακόβουλους δρώντες (actors). Η *βυζαντινή* ανοχή σε σφάλματα παρουσιάζεται σε ένα σύστημα εφόσον αυτό μπορεί να συνεχίσει τη λειτουργία του παρ' όλο που μερικές συνιστώσες του παθαίνουν βλάβες ή δρουν κακόβουλα.

Η συμπεριφορά ενός τέτοιου συστήματος εκφράζεται με το πρόβλημα των *Βυζαντινών στρατηγών* [PSL80; LSP82] που έχει ως εξής: Ένα σύνολο από τμήματα του βυζαντινού στρατού που καθένα διοικείται από διαφορετικό στρατηγό έχουν κατασκηνώσει έξω από μια πόλη, την οποία πολιορκούν. Οι στρατηγοί επικοινωνούν με αγγελιαφόρους και πρέπει να συμφωνήσουν σε ένα κοινό σχέδιο δράσης. Ωστόσο μερικοί από τους στρατηγούς μπορεί να είναι προδότες και να προσπαθούν να εμποδίσουν τους άλλους να φτάσουν σε συμφωνία. Το πρόβλημα είναι να βρεθεί ένας αλγόριθμος που (α) θα επιτρέψει στους έντιμους στρατηγούς να υιοθετήσουν ένα κοινό σχέδιο και (β) λίγοι προδότες στρατηγοί δεν μπορούν να παρασύρουν τους υπόλοιπους σε ένα κακό σχέδιο.

Κάθε στρατηγός μπορεί να στείλει ένα μήνυμα σε άλλον στρατηγό σχετικά με τις προθέσεις του ή μεταφέροντας τις προθέσεις άλλων. Επίσης διαθέτει ένα τρόπο να αποφασίζει τι θα κάνει με βάση τα μηνύματα που παίρνει από τους άλλους, π.χ. αν το ερώτημα είναι κατά πόσο θα επιτεθούν ή όχι και πάρει μη ομοιόμορφες απαντήσεις, μπορεί να ταχθεί με την πλειοψηφία. Ο μη έντιμος στρατηγός μπορεί να δημιουργεί ή να μεταφέρει μηνύματα με σκοπό τη δημιουργία σύγχυσης. Για παράδειγμα, αν υπάρχουν 9 στρατηγοί που θέλουν να αποφασίσουν αν θα επιτεθούν ή όχι και ένας μη έντιμος στρατηγός δει ότι οι 4 είναι υπέρ της επίθεσης και οι άλλοι 4 υπέρ της μη επίθεσης, μπορεί να στείλει στους πρώτους 4 μηνύματα ότι είναι υπέρ της επίθεσης και στους άλλους 4 ότι είναι υπέρ της μη επίθεσης.

[7] Την επιλογή των «βυζαντινών» για τους αναξιόπιστους στρατηγούς - που μπορεί να αλλάξουν πλευρά στη διάρκεια μιας αναμέτρησης ή λίγο πριν - την έχει εξηγήσει ο Lamport, δηλ. ήταν κάπως τυχαία και αποσκοπούσε στο να μη βρεθεί προσβεβλημένο συγκεκριμένο υπαρκτό έθνος. Ο αναγνώστης που θέλει να πάρει μια ιδέα για την πρακτική της αλλαγής πλευράς και την έκταση που μπορούσε να πάρει σε κρίσιμες αναμετρήσεις στο μεσαίωνα θα ήταν καλό να διαβάσει μια περιγραφή της μάχης της Sekigahara στην Ιαπωνία το 1600, είτε από τη Wikipedia είτε από το ομότιτλο βιβλίο [Bry13] του Anthony Bryant.

Σε μια πιο συγκεκριμένη διατύπωση ας υποτεθεί ότι ο στρατηγός i ενός συνόλου n στρατηγών παρατηρεί τον εχθρό και καταλήγει στην τιμή μιας παραμέτρου $v(i)$, την οποία κοινοποιεί στους άλλους στρατηγούς. Κάθε στρατηγός διαθέτει μια μέθοδο να αποφασίζει ένα σχέδιο δράσης όταν γνωρίζει το διάνυσμα $v = (v(1), v(2), \ldots, v(n))$. Η παραπάνω συνθήκη (α) μπορεί να ικανοποιηθεί αν η μέθοδος απόφασης είναι ίδια για όλους τους στρατηγούς. Για παράδειγμα, έστω ότι κάθε $v(i)$ είναι δυαδική παράμετρος (π.χ. επίθεση ή όχι). Έστω επίσης ότι η απόφαση λαμβάνεται με πλειοψηφία. Αν οι προδότες είναι πολύ λίγοι, αλλά η πλειοψηφία των εντίμων είναι ισχυρή προς τη μια ή την άλλη άποψη το αποτέλεσμα δεν θα αλλάξει. Αν η πλειοψηφία των εντίμων είναι οριακή το αποτέλεσμα μπορεί να αλλοιωθεί, τότε όμως το σχέδιο ίσως να μην είναι κακό, επειδή και τα δύο σχέδια έχουν συγκεντρώσει ίδιο περίπου αριθμό προτιμήσεων.

Μια ισοδύναμη διατύπωση του προβλήματος είναι εκείνη όπου υπάρχει ένας αρχιστράτηγος (που μπορεί να είναι ή να μην είναι έντιμος) και στέλνει μηνύματα στους άλλους, ίδια προς όλους αν είναι έντιμος ή και διαφορετικά αν δεν είναι. Οι άλλοι στρατηγοί μπορούν να προωθήσουν το μήνυμα που έχουν πάρει ως έχει, αν είναι έντιμοι, ή και αλλοιωμένο αν δεν είναι. Το επιθυμητό αποτέλεσμα είναι (i) οι έντιμοι στρατηγοί να φτάσουν σε κοινή δράση και (ii) αν ο αρχιστράτηγος είναι έντιμος, αυτή η δράση να είναι εκείνη που έχει διατάξει ο αρχιστράτηγος.

Στην σημαντική εργασία [LSP82] των Lamport, Shostak και Pease του 1982 αποδεικνύονται τα εξής αποτελέσματα:

- Αν υπάρχουν m το πλήθος προδότες, δεν υπάρχει λύση αν το πλήθος των στρατηγών είναι κατώτερο από $3m + 1$. Με άλλα λόγια οι προδότες πρέπει οπωσδήποτε να είναι λιγότεροι από το 1/3 του συνόλου για να υπάρχει λύση.

- Η ανεύρεση μιας προσεγγιστικής λύσης (π.χ. να συμφωνήσουν οι στρατηγοί σε μια δράση με ανοχή στο χρόνο της δράσης) είναι εξ ίσου δύσκολη, δηλαδή δεν επιτρέπει τη χαλάρωση της απαίτησης για το 1/3.

- Στην περίπτωση που τα μηνύματα που ανταλλάσσονται είναι *προφορικά* (δηλαδή ένα μήνυμα μπορεί εύκολα να αλλοιωθεί από κάποιον που το παίρνει και το προωθεί στον επόμενο), αλλά ικανοποιούνται ορισμένες άλλες βασικές προϋποθέσεις ορθής επικοινωνίας ανάμεσα σε κάθε ζεύγος στρατηγών που επικοινωνούν, και εφόσον τηρείται η παραπάνω συνθήκη του 1/3, υπάρχει αλγόριθμος ανταλλαγής μηνυμάτων που εξασφαλίζει λύση για τους έντιμους στρατηγούς. Ο αλγόριθμος αυτός βασίζεται στην ανταλλαγή μεγάλου αριθμού μηνυμάτων από όλους προς όλους ώστε να μπορεί να χρησιμοποιηθεί ένα κριτήριο πλειοψηφίας επί των μηνυμάτων που παίρνει τελικά κάθε στρατηγός.

- Στην περίπτωση που όλα τα μηνύματα υπογράφονται από τον αποστολέα τους και δεν είναι δυνατή η αλλοίωση ενός μηνύματος από ενδιάμεσο, το πρόβλημα είναι επιλύσιμο για πλήθος $m + 2$ (η περίπτωση $m + 1$ στερείται νοήματος).

- Σε όλες τις περιπτώσεις έχει υποτεθεί πλήρης γράφος επικοινωνίας, δηλαδή ότι οποιοσδήποτε στρατηγός μπορεί να στείλει ένα μήνυμα απ' ευθείας σε οποιονδήποτε άλλον. Η λύση μπορεί να επεκταθεί σε ειδικές κατηγορίες γράφων. [8]

[8] Ενα σύνολο κόμβων $(i_1, \ldots, i_p)$, που είναι γείτονες του κόμβου i λέγεται *κανονικό σύνολο γειτόνων* του i αν για κάθε κόμβο $k \neq i$ υπάρχουν μονοπάτια γ_{jk} από τον γείτονα i_j ως τον k τέτοια δύο διαφορετικά μονοπάτια γ_{jk} να μην έχουν κοινό κόμβο άλλον από τον k. Ένας γράφος λέγεται p-κανονικός αν κάθε κόμβος του έχει ένα κανονικό σύνολο p γειτόνων.

Το 1999 οι Miguel Castro και Barbara Liskov ανέπτυξαν τον αλγόριθμο *Practical Byzantine Fault Tolerance*, ο οποίος είναι πρακτικά εφαρμόσιμος σε ένα σύνολο μηχανών που επεξεργάζονται πολυάριθμα αιτήματα ανά μονάδα χρόνου και παρ΄ όλο που μπορούν να υποπέσουν σε βυζαντινά σφάλματα δίνουν απαντήσεις με αξιοπιστία στους πελάτες τους (clients) [CL02]. Στο μοντέλο των Castro και Liskov υπάρχουν n servers (replicas στο άρθρο) που εξυπηρετούν ένα πληθυσμό από clients. Οι τελευταίοι υποβάλλουν αιτήματα στους πρώτους, οι οποίοι υπολογίζουν τις απαντήσεις. Οι απαντήσεις που δίνονται είναι αξιόπιστες εφόσον οι προβληματικοί servers είναι το πολύ $\lfloor (n-1)/3 \rfloor$ το πλήθος. Οι αξιόπιστες μηχανές κατορθώνουν να παρέχουν μια υπηρεσία υπολογισμού σαν να εκτελούσε τους υπολογισμούς ένας και μοναδικός υπολογιστής. Αυτό σημαίνει μεταξύ άλλων ότι όλοι οι servers καταλήγουν στην ίδια σειρά εκτέλεσης των υπολογισμών (linearizability). Ο αλγόριθμος βασίζεται σε κρυπτογραφική προστασία των μηνυμάτων, ώστε να μην αλλοιώνονται κατά την επικοινωνία. Διατάσσει τα αιτήματα προκειμένουν να διασφαλίσει τη σειρά των υπολογισμών, βασίζεται σε πλειοψηφικές αποφάσεις και απαρτίες, και εφαρμόζει την πολλαπλή διάχυση των μηνυμάτων σε διατεταγμένα βήματα.

Στην περίπτωση των κρυπτονομισμάτων η έμφαση δίνεται σε σκόπιμα «σφάλματα» και οι κόμβοι που κάνουν τους υπολογισμούς δεν είναι τόσο καλά συντονισμένοι στην εκτέλεση ενός κατανεμημένου αλγορίθμου, όσο π.χ. στο παραπάνω μοντέλο των Castro-Liskov. Η αντιμετώπιση των προβληματικών κόμβων βασίζεται στην παροχή οικονομικών κινήτρων τιμιότητας σε κάθε κόμβο. Οι προβληματικοί κόμβοι γίνονται πιο επικίνδυνοι όταν φτάσουν στο 51% και γενικά ο κίνδυνος αυξάνεται στατιστικά όσο γίνονται περισσότεροι. Στο αρχικό άρθρο του Satoshi Nakamoto για το BitCoin υπάρχει μια ανάλυση που δείχνει ότι εφόσον κανείς στο δίκτυο δεν διαθέτει πάνω από τη μισή υπολογιστική δύναμη του συνόλου η πιθανότητα να περάσει μια διπλή δαπάνη μειώνεται εκθετικά με το χρόνο. Ωστόσο η εν λόγω ανάλυση βασίζεται στην υπόθεση ότι ο χρόνος διάδοσης ενός block στο δίκτυο είναι σημαντικά μικρότερος από το δεκάλεπτο που μεσολαβεί μέχρι να δημιουργηθεί το επόμενο block.

Στα κρυπτονομίσματα εφαρμόζεται η συσσώρευση πληροφορίας σε ένα κατάστιχο, του οποίου το μέγεθος συνεχώς αυξάνεται και όπου μόνο η πρόσφατη πληροφορία μπορεί να τεθεί εν αμφιβόλω. Την νέα πληροφορία την προτείνει ένας και μοναδικός κόμβος και ακολουθεί έλεγχος από τους άλλους κόμβους και απόφαση που παίρνεται με ένα είδος ψηφοφορίας. Ο έλεγχος εξαρτάται, μεταξύ άλλων, από τις παρελθούσες εγγραφές. Ένας ανέντιμος κόμβος μπορεί να υπερψηφίσει αλλοιωμένες ή ανύπαρκτες συναλλαγές και να καταψηφίσει τίμιες συναλλαγές. Αν το κάνει όμως έχει μεγάλη πιθανότητα να αποκαλυφθεί και να υποστεί οικονομική ζημία. Με δυο λόγια οι εγγυήσεις που παρέχονται από τα κρυπτονομίσματα για την αντιμετώπιση βυζαντινών σφαλμάτων είναι περισσότερο στατιστικής φύσης, ότι δηλαδή με συντριπτικά μεγάλη πιθανότητα τα σφάλματα θα αντιμετωπισθούν (ενώ π.χ. στους αλγορίθμους των [LSP82; CL02] η επιτυχία είναι εγγυημένη εφόσον οι ανέντιμοι κόμβοι δεν περνούν το 1/3).

Συναίνεση στο BitCoin

Το πρωτόκολλο συναίνεσης στο BitCoin βασίζεται στα εξής: Κάθε τόσο επιλέγεται τυχαία ένας κόμβος του δικτύου του BitCoin που παίρνει το δικαίωμα να προτείνει το επόμενο block. Το block αυτό στη συνέχεια θα περάσει από τον έλεγχο άλλων κόμβων προκειμένου να ενωθεί με τα έγκυρα blocks.

Ο αλγόριθμος λειτουργεί ως εξής: (α) Κάθε νέα συναλλαγή εκπέμπεται προς όλους τους άλλους κόμβους. (β) Κάθε κόμβος μαζεύει τις νέες συναλλαγές και σχηματίζει

ένα block. (γ) Σε κάθε νέο γύρο ένας κόμβος παίρνει το δικαίωμα να υποβάλει το δικό του block ως υποψήφιο για ένωση με το έγκυρο blockchain. (δ) Οι άλλοι κόμβοι ελέγχουν όλες τις συναλλαγές για παραβιάσεις. (ε) Εφόσον τους φανεί ότι αυτό το block είναι ΟΚ, «χτίζουν» το επόμενό τους block βασισμένοι πάνω σε αυτό (δηλαδή βάζουν ένα κωδικό κατακερματισμού που δείχνει αυτό το block). Το τελευταίο αυτό γεγονός μετράει ως ψήφος υπέρ του προηγούμενου block.

Η παραμετροποίηση του BitCoin είναι τέτοια ώστε ο χρόνος διάδοσης ενός block μέσα στο δίκτυο να είναι αρκετά μικρότερος από το χρόνο μέχρι να δημιουργηθεί το επόμενο block (που κατά μέσο όρο είναι ίσος με ένα δεκάλεπτο). Παρ' όλα αυτά και δεδομένης της πεπερασμένης ταχύτητας διάδοσης ενός block μέσα στο δίκτυο μπορεί σε ένα κόμβο να φτάσουν δύο υποψήφια blocks που έχουν δημιουργηθεί είτε από δύο κόμβους απομακρυσμένους μεταξύ τους είτε και επειδή ο ένας κόμβος είναι κακόβουλος. Ο κανόνας λέει ότι επιλέγεται το block που είναι χτισμένο πάνω στην μακρύτερη αλυσίδα (blockchain), προκειμένου εν τέλει να μείνουν ορφανές οι ύποπτες διακλαδώσεις. Ο κανόνας αυτός εγγυάται ότι στατιστικά θα επικρατήσει η σωστή αλυσίδα, εκτός αν κάποιος έχει πάνω από το 50% της υπολογιστικής δύναμης.

Ας υποτεθεί τώρα ότι η Alice έχει γράψει ένα βιβλίο πουλάει ψηφιακά αντίγραφά του. Ο Bob την πληρώνει σε BitCoin και περιμένει να του στείλει η Alice ένα link να κατεβάσει το βιβλίο. Πόσο γρήγορα πρέπει να ανταποκριθεί η Alice που παρακολουθεί τις πληρωμές; Αν περιμένει λίγο, θα δει την πληρωμή του Bob να μπαίνει μέσα στην επόμενη γενιά από blocks συναλλαγών και ένα από αυτά θα μπει στο blockchain. Αν περιμένει λίγο παραπάνω, θα μπει κι άλλο block, το οποίο θα δείχνει (με ένα δείκτη κατακερματισμού) προς το προηγούμενο κ.ο.κ. Όσο περισσότερη ώρα περιμένει, τόσο πιο βαθιά στο blockchain θα βρεθεί η πληρωμή του Bob προς την Alice και τόσο πιο σίγουρη θα είναι.

Τι θα μπορούσε να πάει στραβά; Έστω ότι ο Bob έχει κάνει μια δεύτερη συναλλαγή με το ίδιο νόμισμα προς την Ελένη. Κάποιοι άλλοι κόμβοι μπορεί να δουν τις δύο συναλλαγές και να προκρίνουν την δεύτερη για εισαγωγή στο block που σχηματίζουν. Τα πράγματα μπορεί να εξελιχθούν τόσο να δημιουργηθεί μια άλλη σειρά από blocks, δηλαδή το αρχικό blockchain να σπάσει σε δύο κλάδους, καθένας από τους οποίους θα στηρίζει μια από τις δύο συναλλαγές. Ωστόσο η πιθανότητα να συνεχίζονται συγχρόνως δυο κλάδοι χωρίς να λαμβάνεται η αντίφαση μειώνεται εκθετικά με τον αριθμό των blocks. Στατιστικά στην πράξη αν η Alice περιμένει να δει έξι blocks μετά τη συναλλαγή που την ενδιαφέρει μπορεί να είναι βέβαιη ότι δεν θα υπάρξει διαφορετική εξέλιξη.

Κίνητρα τιμιότητας

Το γεγονός ότι το υποκείμενο μιας συναλλαγής είναι μια ανώνυμη διεύθυνση (ένα δημόσιο κλειδί που μπορεί να μην έχει ξαναχρησιμοποιηθεί στο παρελθόν) αποτελεί μια πρόσθετη δυσχέρεια στο να επιβληθούν κανόνες σχετικοί με την τιμιότητα των συναλλαγών. Πώς επιβάλλεται ένας κανόνας απαγόρευσης διπλής δαπάνης (του ίδιου ποσού) όταν δεν είναι δυνατό να ανιχνευθεί ποιος έκανε τη διπλή δαπάνη; Το πρόβλημα αυτό επιλύεται από το bitcoin όχι με μια άμεση τιμωρία (δηλαδή αφαίρεση του ποσού και περαιτέρω ποινή, όπως πρόστιμο), κάτι που θα γινόταν στο κανονικό τραπεζικό σύστημα, αλλά μέσα από ένα μηχανισμό ελέγχου από την πλειοψηφία των ομοτίμων και κινήτρων που ωθούν τον κάθε ελεγκτή να δίνει προτεραιότητα στις τίμιες συναλλαγές και να απορρίπτει τις παράνομες. Τα κίνητρα τιμιότητας είναι αυτά που θα ωθήσουν ένα κόμβο-ελεγκτή να χτίσει ένα νέο block με σωστό έλεγχο των συναλλαγών που τοποθετεί μέσα σ' αυτό.

Όταν ένας κόμβος παίρνει (μέσα από μια διαδικασία τυχαίας επιλογής που θα περιγράψουμε πιο κάτω) το δικαίωμα να υποβάλει ένα νέο block, έχει επίσης το δικαίωμα να προσθέσει στο block μια ειδική συναλλαγή αμοιβής προς τον εαυτό του. Η εν λόγω ειδική συναλλαγή δημιουργεί εκ του μηδενός ένα νέο ποσό. Το ποσό αυτό αρχικά ήταν 50 bitcoins και υποδιπλασιάζεται κάθε 210000 blocks.[9] Η είσπραξη αυτής της αμοιβής γίνεται προφανώς μέσω του συστήματος του BitCoin, επομένως εξαρτάται από την τύχη του δημιουργούμενου νέου block. Αν το block αυτό στη συνέχεια αποδειχθεί προβληματικό και απορριφθεί, θα ακυρωθεί και η αμοιβή. Με τον τρόπο αυτόν ο κόμβος που το δημιουργεί έχει κίνητρο να εισαγάγει τίμιες συναλλαγές στο block. Κάθε νέο block που μπαίνει στο blockchain ισχυροποιείται ως προς την αποδοχή του όσο στη συνέχεια σχηματίζονται πάνω του άλλα νέα blocks.

Σύμφωνα με τους κανόνες του BitCoin ο μηχανισμός της αμοιβής παύει να λειτουργεί όταν το συνολικό ποσό φτάσει τα 21 εκατομμύρια bitcoins, γεγονός που αναμένεται να συμβεί περί το 2140. Δεδομένου ότι αυτός είναι ο μόνος μηχανισμός δημιουργίας νέου χρήματος, το συνολικό ποσό που θα υπάρχει δεν μπορεί να υπερβεί τα 21 εκατομμύρια. Πέραν της παραπάνω αμοιβής υπάρχει το *τέλος συναλλαγής* (transaction fee): Κάθε δημιουργός νέας συναλλαγής μπορεί προαιρετικά να κάνει το εξερχόμενο ποσό κατά τι μικρότερο από το εισερχόμενο. Στη συνέχεια τη διαφορά μπορεί (αν θέλει) να εισπράξει ο δημιουργός του νέου block, στο οποίο καταλήγει η συναλλαγή.

Το τέλος συναλλαγής λειτουργεί ως κίνητρο για τον δημιουργό ενός νέου block να συμπεριλάβει μια συγκεκριμένη συναλλαγή στο νέο block ή να μην την συμπεριλάβει, δεδομένων ορισμένων περιορισμών μεγέθους του block. Ο δημιουργός της συναλλαγής, εφόσον δεν βιάζεται, μπορεί να βάλει πολύ χαμηλό ή μηδενικό τέλος, αλλά εφόσον βιάζεται πρέπει να εκτιμήσει με ποιο τέλος θα μπορέσει να την περάσει στο επόμενο block. Πρόκειται δηλαδή για ένα παιχνίδι ανάμεσα στον δημιουργό κάθε συναλλαγής και στον δημιουργό του επόμενου block. Το παιχνίδι αυτό μπορεί να παίζει για λογαριασμό ενός χρήστη του BitCoin το «πορτοφόλι» του (wallet), δηλαδή η εφαρμογή μέσω της οποίας ο χρήστης διαχειρίζεται της συναλλαγές του. Ενδεικτική τιμή για το τέλος ανά συναλλαγή τον Σεπτ. του 2020 είναι λίγα δολλάρια (USD).

Εξόρυξη

Κάθε κόμβος του BitCoin φιλοδοξεί να είναι αυτός που θα υποβάλει το επόμενο block, ώστε να εισπράξει την αμοιβή (που τώρα ανέρχεται σε 6.25 bitcoins). Η εξόρυξη (mining) είναι στο BitCoin και σε άλλα κρυπτονομίσματα ο μηχανισμός, μέσω του οποίου επιλέγεται ποιος θα έχει το προνόμιο. Ο μηχανισμός αυτός εμπίπτει στην κατηγορία της *απόδειξης έργου*, δηλαδή η επιλογή βασίζεται στο να δείξει κάποιος ότι έχει προσπαθήσει αρκετά. Άλλοι μηχανισμοί εμπίπτουν στην κατηγορία της *απόδειξης συμφέροντος* (proof of interest), π.χ. αποδεικνύοντας ότι κάποιος κατέχει μεγάλα ποσά. Θα δούμε πιο κάτω ότι ένας μηχανισμός απόδειξης έργου αποτελεί ενός είδους εγγύηση για την τιμιότητα του αποτελέσματος.

Στο BitCoin κάθε κόμβος αρχικά συλλέγει ένα πλήθος από συναλλαγές, τις οποίες πληροφορείται μέσω του δικτύου. Αυτές τις συναρμολογεί σε ένα string της μορφής $\text{trs} = \text{tr}_1 \| \text{tr}_2 \| \ldots \| \text{tr}_N$, όπου tr_i ($i = 1, 2, \ldots, N$) είναι η i-στή συναλλαγή που θα μπει στο block. Το συνολικό string trs ενώνεται με ένα δείκτη κατακερματισμού h προς το τελευταίο ως τώρα block. Ο κόμβος πρέπει να υπολογίσει ένα nonce (αριθμό)

[9] Ο τρίτος υποδιπλασιαμός έγινε τον Μάιο του 2020 και ο τέταρτος αναμένεται τον Μάιο του 2024. Αυτή τη στιγμή (τέλος 2020) άρα είμαστε σε μια αμοιβή ίση με 6.25 bitcoin.

r τέτοιο ώστε να ισχύει

$$H(r\|h\|\text{trs}) < \epsilon$$

για ένα δεδομένο μικρό θετικό ϵ, του οποίου η τιμή αποτελεί παράμετρο του συστήματος. Δεδομένου ότι η H είναι φιλική προς τα αινίγματα (puzzle friendly) δεν υπάρχει τρόπος απ' ευθείας υπολογισμού του r, παρά μόνο με τυχαίες δοκιμές. Ο «στόχος» ϵ (target) είναι ένας αριθμός μήκους 256 bits και προσδιορίζεται έτσι ώστε με την τρέχουσα υπολογιστική ισχύ όλων μαζί των κόμβων να σημειώνεται μια επιτυχία κάθε περίπου δέκα λεπτά, δηλαδή να δημιουργείται ένα νέο block ανά δεκάλεπτο. Δεδομένης της μεταβολής της υπολογιστικής ισχύος με το χρόνο, ο στόχος επαναπροσδιορίζεται στο BitCoin κάθε 2016 blocks, δηλαδή ανά δύο εβδομάδες.

Επίθεση με πλειοψηφία

Δεδομένου ότι η έγκριση ενός block βασίζεται στη δημιουργία επομένων blocks που χτίζουν πάνω σ' αυτό (δηλαδή συνεχίζουν την αλυσίδα) και ότι το προνόμιο της δημιουργίας νέων blocks δίνεται βάσει απόδειξης έργου, δηλαδή στατιστικά σε όποιον έχει μεγαλύτερη υπολογιστική ισχύ, είναι λογικό να υποθέσει κανείς ότι αν ένας παίκτης κατέχει αρκετές μηχανές, ώστε να αποτελεί πλειοψηφία στο συνολικό υπολογιστικό δυναμικό, θα έχει τη δυνατότητα να περνάει τις συναλλαγές που θέλει ως νόμιμες. Γι' αυτό θα δούμε μια σειρά από επιθέσεις που μπορούν ή δεν μπορούν να γίνουν με μια τέτοια πλειοψηφία.

Μπορεί ο κατέχων το 51% να κάνει μια πληρωμή από την διεύθυνση ενός άλλου προς μια διεύθυνση δική του; Μια τέτοια μη νόμιμη πληρωμή θα ανιχνευθεί από τους άλλους κόμβους (της μειοψηφίας), οι οποίοι δεν θα συνεχίσουν το block του 51%, δηλαδή θα χτίσουν πάνω στο τελευταίο νόμιμο block δημιουργώντας έτσι μια διακλάδωση. Στη συνέχεια κάθε πλευρά μπορεί να συνεχίσει να χτίζει πάνω στον δικό της κλάδο. Ας υποτεθεί ότι το 51% κάνει μια πληρωμή από μια διεύθυνσή του προς μια διεύθυνση της μειοψηφίας [CCF20]. Αν ο ιδιοκτήτης της διεύθυνσης αυτής τρέχει τον δικό του κόμβο, που έχει δεχθεί άρα την τίμια διακλάδωση, θα παρατηρήσει ότι το block είναι εκτός blockchain. Λογικά αργά ή γρήγορα το πρόβλημα των δύο κλάδων θα γίνει αντιληπτό από μεγάλο μέρος των κόμβων και θα ξεκινήσει η λήψη μέτρων.

Γενικά οποιαδήποτε παράβαση, όπως παράλειψη συναλλαγών, μεταβολή της αμοιβής κ.λπ., θα καταλήξει σε διακλαδώσεις και σε ασυνέπειες. Το αποτέλεσμα μιας τέτοιας συστηματικής συμπεριφοράς μπορεί να είναι πως το ευρύ κοινό θα αποσύρει την εμπιστοσύνη του από το κρυπτονόμισμα, πράγμα που αποτελεί τον βασικό κίνδυνο από τέτοιες επιθέσεις [Nar+16].

Δομή του blockchain στο BitCoin

Στο BitCoin κάθε νέα έγκυρη εγγραφή προστίθεται στο σύνολο των έγκυρων εγγραφών. Το σύνολο των εγγραφών περιλαμβάνει όλες τις έγκυρες συναλλαγές από καταβολής του BitCoin, οργανωμένες σε ένα blockchain. Στην σελίδα 192 είδαμε ότι εκτός από την περίπτωση να σχηματίσει κανείς μια γραμμική αλυσίδα από blocks, όπου καθένα περιέχει ένα δείκτη κατακερματισμού προς το προηγούμενο, υπάρχει η δυνατότητα να δημιουργηθεί ένας ακυκλικός γράφος (δέντρο Merkle). Τι από αυτά χρησιμοποιεί το BitCoin;

Στο BitCoin χρησιμοποιούνται και οι δύο δομές συγχρόνως. Χρησιμοποιείται μια γραμμική δομή από block σε block. Όπως είδαμε παραπάνω ένα block περιλαμβάνει πολλές συναλλαγές. Αν δούμε το block ως ένα string, αυτό περιλαμβάνει εκτός από τις νέες συναλλαγές (που σχηματίζονται περίπου σε ένα δεκάλεπτο) και ένα δείκτη

κατακερματισμού προς το προηγούμενο block.Εσωτερικά όμως οι εγγραφές μέσα σε ένα block οργανώνονται πάνω σε ένα δέντρο Merkle.

Η διπλή αυτή δομή έχει σημασία για το πώς επιλύονται διάφορα προβλήματα ελέγχου τήρησης των κανόνων. Για παράδειγμα, εάν πρέπει να ελεγχθεί για την περίπτωση της διπλής δαπάνης μια συναλλαγή με την οποία η Alice πληρώνει τον Bob ένα ποσό, που έχει λάβει με μια προηγούμενο συναλλαγή, αρκεί να ελεγχθούν τα blocks ανάμεσα στις δύο συναλλαγές.

Περιγραφή των συναλλαγών

Με ποιο τρόπο περιγράφονται οι συναλλαγές σε ένα block; Περιγράφονται με κάποιο τυποποιημένο κείμενο κατανοητό από οποιονδήποτε (όχι κρυπτογραφημένες) μέσα στο block. Γράφονται με τρόπο που διευκολύνει τους ελέγχους συμμόρφωσης με τους κανόνες. Πριν δούμε μια τέτοια περιγραφή θα δώσουμε μια συμβολική περιγραφή για να καταλάβουμε τι καλείται να περιλαμβάνει η περιγραφή.

Ας υποθέσουμε ότι οι εγγραφές είναι αποκλειστικά συναλλαγές της μορφής $X \xrightarrow{Z} Y$, δηλαδή από τη διεύθυνση X μεταφέρεται το ποσό Z στη διεύθυνση Y. Για να επαληθεύσουμε ότι μια συγκεκριμένη πληρωμή $x_0 \xrightarrow{z_0} y_0$ είναι δυνατή πρέπει να ελέγξουμε ότι η διεύθυνση x_0 έχει στη διάθεσή της το ποσό z_0, δηλαδή ότι το άθροισμα των εισερχομένων ποσών μείον το άθροισμα των εξερχομένων ποσών στην x_0 είναι μεγαλύτερο από z_0, όπως αυτά τα ποσά έχουν καταγραφεί σε όλες τις συναλλαγές που εμπλέκουν την x_0 στο κατάστιχο.

Το BitCoin προκειμένου να διευκολύνει τους ελέγχους χρησιμοποιεί συναλλαγές της μορφής

$$D : (D_1, D_2, \ldots, D_N), X \xrightarrow{Z_1} Y_1, X \xrightarrow{Z_2} Y_2, \ldots, X \xrightarrow{Z_M} Y_M$$

όπου D είναι η ταυτότητα της τρέχουσας συναλλαγής, $D_1, D_2, \ldots, D_N$ είναι παλιότερες συναλλαγές στις οποίες αναφέρεται η νέα, $X \xrightarrow{Z_i} Y_i$ είναι μια σειρά από νέες πληρωμές από την διεύθυνση X προς τις διευθύνσεις $Y_1, \ldots, Y_M$ με αντίστοιχα ποσά $Z_1, \ldots, Z_M$. Για να είναι δυνατή αυτή η συναλλαγή πρέπει να επαληθευτεί ότι στις προηγούμενες συναλλαγές $D_1, D_2, \ldots, D_N$ η διεύθυνση X έχει συλλέξει ποσό *ακριβώς ίσο* με το $Z_1 + \ldots + Z_M$. Με άλλα λόγια οι συναλλαγές αυτές είναι ισοσκελισμένες, δηλαδή σε κάθε μια καταναλώνεται όλο το ποσό που μια διεύθυνση X έχει εισπράξει από τις αναφερόμενες συναλλαγές $D_1, D_2, \ldots, D_N$. Αν ο κάτοχος της X επιθυμεί να πληρώσει ποσό μικρότερο του διαθέσιμου από τις αναφερόμενες προηγούμενες συναλλαγές, πρέπει να προσθέσει μια νέα συναλλαγή από την X προς X με το υπόλοιπο ποσό. Για τον έλεγχο της εγκυρότητας της παραπάνω συναλλαγής ως προς τη διαθεσιμότητα του ποσού αρκεί κανείς να διαβάσει τις προηγούμενες συναλλαγές χρησιμοποιώντας τους δείκτες $D_1, D_2, \ldots, D_N$ και να προσθέσει τα ποσά που εμφανίζονται να αποτελούν πληρωμές προς την X (βέβαια σ᾽ αυτές μπορεί να υπάρχουν και πληρωμές προς άλλες διευθύνσεις). Στη συνέχεια το άθροισμα αυτό πρέπει να είναι ίσο με το $Z_1 + \ldots + Z_M$.

Οι συναλλαγές στο BitCoin περιγράφονται με scripts στη γλώσσα του BitCoin που μοιάζει με την Forth. Η γλώσσα δεν επιτρέπει βρόχους, οι εντολές εκτελούνται σειριακά. Οι δυνατότητες της γλώσσας είναι προσαρμοσμένες στις ανάγκες του BitCoin, δεν είναι μια γενικής χρήσης γλώσσα (δεν είναι Turing complete).Ένα κατάλογο των διαθέσιμων εντολών μπορείτε να δείτε στο `https://en.bitcoin.it/wiki/Script`. Δείτε ότι ορισμένες από τις εντολές είναι πολύ ισχυρές, π.χ. η εντολή `OP_SHA256`

υπολογίζει απ' ευθείας τον κωδικό κατακερματισμού για ένα κείμενο χρησιμοποιώντας την συνάρτηση κατακερματισμού SHA-256. Περισσότερες πληροφορίες μπορείτε να αναζητήσετε στο βιβλίο του Narayanan [Nar+16].

Ωστόσο μπορείτε να δείτε τις τρέχουσες και τις παρελθούσες συναλλαγές μέσω ενός viewer (π.χ. στο `https://btc.com`), που παρουσιάζει το περιεχόμενο ενός block, δηλαδή τις σχετικές συναλλαγές, με τρόπο φιλικό προς τον άνθρωπο. Επίσης μπορείτε να δείτε ποιος κόμβος εισέπραξε την αμοιβή για κάθε block, πόση ήταν η αμοιβή (από εξόρυξη και τέλη), πόσες άλλες επιτυχίες είχε ο συγκεκριμένος κόμβος κ.ο.κ.

Το πραγματικό σύστημα

Μολονότι η βασική θεωρία για το BitCoin και παρόμοια κρυπτονομίσματα έχει βασικά ήδη δοθεί, η πρακτική εφαρμογή της σε πραγματικά συστήματα βασίζεται σε ένα πλήθος άλλων σχεδιαστικών αποφάσεων και λεπτομερειών. Στη συνέχεια θα δώσουμε μια περίληψη διαφόρων υποσυστημάτων που είναι απαραίτητα για να υλοποιηθεί το κρυπτονόμισμα και οι λειτουργίες του.

Πορτοφόλια

Το *πορτοφόλι* (wallet) είναι λογισμικό για τον κοινό θνητό που δεν επιθυμεί να τρέχει ένα κόμβο, αλλά μόνο να χρησιμοποιεί το BitCoin. Αυτό σημαίνει να ξέρει ποιο ποσό έχει στην κατοχή του, να κάνει πληρωμές, να λαμβάνει πληρωμές, να δημιουργεί νέες δικές του διευθύνσεις, να μεταφέρει χρήματα ανάμεσα στις διευθύνσεις του κ.ο.κ.

Βασικά ένα πορτοφόλι κρατάει και διαχειρίζεται κλειδιά. Θυμηθείτε ότι κάθε διεύθυνση δεν είναι παρά ο κωδικός κατακερματισμού ενός δημόσιου κλειδιού. Η πολυπλοκότητα ενός πορτοφολιού δείχνει πόσο εφιαλτικά περίπλοκο ζήτημα είναι η διαχείριση κλειδιών. Δείχνει επίσης ότι η κρυπτογραφία δημόσιου κλειδιού δεν καταλήγει σε ιδιαίτερα απλές λύσεις.

Το πορτοφόλι «περιέχει» ένα ποσό X εφόσον έχει μια διεύθυνση που έχει εισπράξει διάφορα ποσά και το τρέχον υπόλοιπό της είναι ίσο με X. Δεδομένου ότι ένα άτομο έχει κατά κανόνα πολλές διευθύνσεις, το πορτοφόλι διαχειρίζεται όλες τις διευθύνσεις.

Μπορεί να διαχειρίζεται επίσης την παραγωγή και την αποθήκευση νέων διευθύνσεων, δηλαδή εν τέλει ζευγών από ιδιωτικό και δημόσιο κλειδί. Υπάρχουν δύο βασικοί τρόποι παραγωγής. Ο ένας είναι η ανεξάρτητη παραγωγή ενός νέου ζεύγους κάθε φορά που χρειάζεται μια νέα διεύθυνση. Αυτό έχει ως αποτέλεσμα τη συσσώρευση κλειδιών, που πρέπει να αποθηκευτούν με ασφάλεια. Μια πιο «εύκολη» λύση είναι να αποθηκεύεται ένα βασικό κλειδί (seed) από αυτό να γεννιούνται ακολουθιακά τα επόμενα, όταν χρειάζονται και μπορούν να επανυπολογίζονται σε κάθε χρήση. Προφανώς πέραν του επανυπολογισμού, η δεύτερη λύση εισάγει ένα μοναδικό σημείο πιθανής αστοχίας.

Μια άλλη διάκριση των πορτοφολιών είναι σε πορτοφόλια *θερμής* αποθήκευσης (hot storage), δηλαδή στον υπολογιστή (κινητό, tablet κ.λπ.) που συνδέεται διαδικτυακά και *ψυχρής* αποθήκευσης (cold storage), που είναι οποιοδήποτε μέσο μπορεί να κρατήσει την σχετική πληροφορία, αλλά δεν συνδέεται με το δίκτυο και λογικά είναι πιο ασφαλές. Ο ιδιοκτήτης δυο τέτοιων πορτοφολιών μπορεί να μεταβιβάζει ποσά από το ένα στο άλλο, πράγμα που καταλήγει σε ακόμη περισσότερα προβλήματα διαχείρισης κλειδιών.

Σε όλες τις περιπτώσεις το έσχατο ζήτημα είναι πώς ένα ή περισσότερα κλειδιά θα παραμείνουν ασφαλώς αποθηκευμένα, δεδομένων των περιορισμών και των κιν-

δύνων που διέπουν όλες τις ηλεκτρονικές συσκευές. Μια λύση στο τέλος είναι να εκτυπωθούν τα κλειδιά σ᾽ ένα μόνιμο μέσο, π.χ. χαρτί, κι αυτό να καταλήξει σε ένα χρηματοκιβώτιο ή άλλο ασφαλές μέρος φύλαξης. Θα μπορούσε κανείς να παρακάμψει τον (ενδεχόμενα ανασφαλή) εκτυπωτή σημειώνοντας τα κλειδιά με το χέρι, αλλά μια από τις δυσκολίες είναι το μήκος και η τυχαία τους δομή. Μια λύση που έχει υλοποιηθεί είναι η δημουργία μιας κωδικής φράσης (pass phrase ή seed phrase), την οποία μπορεί κάποιος να θυμάται ή να καταγράψει ευκολότερα και αντιστοιχεί μοναδικά σε κάποιο seed. Ένα παράδειγμα είναι η φράση `witch collapse practice feed shame open despair creek road again ice least` (βλ. και `https://en.bitcoin.it/wiki/Seed_phrase`). Το πρόβλημα που πρέπει να λύσει ο χρήστης αυτής της μεθόδου σχετίζεται βασικά με την ασφάλεια του μετατροπέα από μια συμβολοσειρά σε pass phrase.

Μια άλλη λύση είναι να βρίσκεται το πορτοφόλι στο νέφος (online wallet), δηλαδή να χρησιμοποιήσει κάποιος έτοιμες υπηρεσίες τρίτων. Το πορτοφόλι τότε υλοποιείται ως υπηρεσία ιστού (web service). Μια τέτοια λύση φαίνεται πολύ βολική, αλλά εγείρει ζητήματα ασφάλειας τόσο για τον browser, όσο και την πλευρά που προσφέρει την υπηρεσία.

Ανταλλακτήρια

Ο εύκολος και συνήθης τρόπος να αποκτήσει κανείς κρυπτονομίσματα δεν είναι μέσω της εξόρυξης, αλλά με απ᾽ ευθείας αγορά τους από κατόχους κρυπτονομισμάτων πληρώνοντας με ένα συνηθισμένο νόμισμα. Τέτοιες υπηρεσίες προσφέρονται από ανταλλακτήρια που διαμεσολαβούν ανάμεσα σε κατόχους κρυπτονομισμάτων και σε άλλους πελάτες. Επίσης μπορούν να ανταλλάξουν τα κρυπτονομίσματα με κανονικά νομίσματα.

Η ύπαρξη τέτοιων σημείων είναι αναγκαία ώστε να υπάρχει ανταλλαξιμότητα μεταξύ νομισμάτων και κρυπτονομισμάτων. Τέτοιες ανταλλαγές σχετίζονται πάντοτε και με το θέμα της ισοτιμίας, δηλαδή π.χ. σήμερα σε ποια τιμή σε Ευρώ ανταλλάσσεται ένα BitCoin; Ενδεικτικά, στις 24/9/2020 ένα BitCoin ήταν ίσο με 8866 Ευρώ. Η μεταβολή της ισοτιμίας δίνει την ευκαιρία για διάφορα παιχνίδια στις σχετικές αγορές.

Σε κάθε περίπτωση ένα κρυπτονόμισμα είναι δύσκολο να αποτελέσει ένα απομονωμένο σύστημα, δεδομένου ότι είναι περιορισμένη η αγορά που δέχεται ένα κρυπτονόμισμα, σήμερα τουλάχιστον. Η κατάσταση αυτή μπορεί να μεταβληθεί στο μέλλον. Τον Σεπτ. 2020 η αλυσίδα εστιατορίων *Just Eat*, όπου κανείς μπορεί να παραγγείλει από πριν μέσω Διαδικτύου πριν πάει στο εστιατόριο, ανακοίνωσε ότι δέχεται στη Γαλλία πληρωμές σε μια σειρά από κρυπτονομίσματα για μια σειρά από 15000 εστιατόρια.

Ωστόσο τα ανταλλακτήρια είναι στην ουσία τους τράπεζες, δηλαδή είναι ακριβώς αυτό που προσπάθησαν να αποφύγουν οι δημιουργοί των κρυπτονομισμάτων. Σαν τέτοιες υπόκεινται σε όλους τους σχετικούς περιορισμούς και κινδύνους. Η διαφορά ωστόσο είναι ότι τα ανταλλακτήρια κρυπτονομισμάτων δεν είναι «κανονικές» τράπεζες, δηλαδή δεν υπόκεινται στην τραπεζική νομοθεσία και δεν εγγυάται το κράτος τη λειτουργία τους. Στην πράξη τα ανταλλακτήρια έχουν αποδειχθεί ιδιαίτερα επισφαλή. Κλασσικό παράδειγμα είναι η υπόθεση του Mt Gox. Το τελευταίο ήταν ένα ιαπωνικό ανταλλακτήριο, το οποίο διαχειριζόταν το 70% των παγκόσμιων συναλλαγών σε BitCoin στη διετία 2013-14. Τον Φεβρουάριο του 2014 ανακάλυψε ότι δεν μπορούσε να καλύψει τις υποχρεώσεις του και κήρυξε πτώχευση.[10] Τέτοια περιστατικά δεν αποτε-

[10] `https://en.wikipedia.org/wiki/Mt._Gox`

λούν βεβαίως διαφήμιση για τα κρυπτονομίσματα.

Ενεργειακό αποτύπωμα

Πιο πάνω εξηγήσαμε με ποιο τρόπο προσδιορίζεται ποιος κόμβος θα έχει το δικαίωμα να προτείνει το επόμενο block. Το δικαίωμα αυτό εξασφαλίζεται με εντατικούς υπολογισμούς, οι οποίοι καθώς περνούσαν τα χρόνια γίνονταν όλο και πιο εντατικοί εξ αιτίας του ανταγωνισμού. Γρήγορα οι υπολογισμοί ξέφυγαν από τις δυνατότητες ενός απλού χρήστη υπολογιστή και έγιναν προνόμιο μεγάλων παικτών μόνο. Οι παίκτες αυτοί διαθέτουν σημαντικοί υπολογιστική ισχύ, συνήθως σε φάρμες υπολογιστών.

Σημαντική είναι και η κατανάλωση ρεύματος, γεγονός που σημαίνει έξοδα και ενεργειακό αποτύπωμα. Επίσης δημιουργούνται ζητήματα ψύξης των μηχανημάτων, που σε κάποιες περιπτώσεις επιλύονται με φάρμες σε ψυχρά κλίματα. Βεβαίως τέτοιες φάρμες χρησιμοποιούνται γενικότερα και ιδίως στο νέφος.

Το ενεργειακό αποτύπωμα ίσως είναι μικρότερο από αυτό άλλων τομέων, ωστόσο δεν είναι αμελητέο. Στην κατεύθυνση της μείωσής του είναι ο σχεδιασμός φάρμας που χρησιμοποιεί φιλικούς προς το περιβάλλον τρόπους παραγωγής ενέργειας, καθώς και η περαιτέρω εκμετάλλευση της παραγόμενης θερμότητας.

Η φάρμα του *BitRiver*[11] στεγάζεται σε ένα πρώην σοβιετικό εργοστάσιο στο Bratsk της Ρωσίας. Περιέχει 18000 συσκευές εξόρυξης και καταναλώνει περί τα 100 MWatt ηλεκτρικής ισχύος. Βρίσκεται κοντά σε υδροηλεκτρικό σταθμό παραγωγής ηλεκτρικής ενέργειας και έχει την ευκαιρία εξαιρετικά χαμηλών τιμών στο ρεύμα. Επίσης επωφελείται από το ψυχρό κλίμα της Σιβηρίας για την ψύξη των μηχανημάτων. Το *Minery* είναι μια άλλη ρωσική εταιρία με φάρμες στην ίδια περιοχή που καταναλώνουν συνολικά γύρω στα 150 MWatt. Τέτοιες φάρμες έχουν πελάτες (στους οποίους είτε πουλάνε υπηρεσίες εξόρυξης είτε τους ενοικιάζουν μηχανήματα) από όλον τον κόσμο. Οι ρωσικές φάρμες μόνο καταναλώνουν περί τα 600 MWatt, ενώ η παγκόσμια κατανάλωση βρίσκεται σήμερα στα 7 GWatt (πολύ ψηλότερα από τα 100 MWatt που υπολογίσθηκαν το 2016 από τον Narayanan [Nar+16]).

Δυστυχώς η κατανάλωση ισχύος μεγαλώνει. Το 2017 ήταν της τάξης λίγων εκατοντάδων MWatt. Το 2018 ήταν της τάξης μερικών GWatt. Ένας υπολογισμός του ενεργειακού αποτυπώματος (carbon footprint) για το τέλος του 2018, όταν η κατανάλωση ήταν περί τα 5GWatt, έδωσε 22.9 $MtCO_2$ [SKG19]. Το αποτύπωμα αυτό είναι του μεγέθους μιας χώρας όπως η Ιορδανία και η Sri Lanka ή της πολιτείας του Kansas. Το αποτύπωμα αυτό είναι γύρω στις δύο τάξεις μεγέθους μικρότερο από το αποτύπωμα της πολιτικής αεροπορίας (περί τον 1 $GtCO_2$ ετησίως).

Κατά κάποιο τρόπο μια φάρμα BitCoin είναι ένας μετατροπέας ενέργειας σε χρήμα. Δεδομένου ότι η παραγωγή ρεύματος π.χ. από ανεμογεννήτριες υπόκειται σε ισχυρή διακύμανση της ισχύος, ενώ η αποθήκευση ενέργειας σε μπαταρίες παραμένει μια δύσκολη υπόθεση, η διοχέτευση αυτής της ενέργειας σε εξόρυξη BitCoin θα μπορούσε να θεωρηθεί ως ένα είδος αποθήκευσης. Όταν υπάρχει περίσσεια ενέργειας αυτή θα μπορούσε να διοχετεύεται στην εξόρυξη και στη συνέχεια, όταν υπάρχει έλλειψη, να αγοράζεται ενέργεια με τα χρήματα που έχουν μαζευτεί.

Μια άλλη ιδέα για να ελαττωθεί η άχρηστη παραγωγή ισχύος είναι να μετατραπεί σε κάποιου είδους χρήσιμη παραγωγή, δηλαδή να επιδιώκεται κάποιος άλλος χρήσιμος υπολογισμός. Ωστόσο οι γνωστοί χρήσιμοι υπολογισμοί που θα μπορούσαν να

[11] Βλ. IEEE Spectrum, Φεβρ. 2020, σελ. 13. Βλ. επίσης `https://bitriver.farm`.

θεωρηθούν κατάλληλοι είναι ελάχιστοι. Ένας εξ αυτών αφορά στον εντοπισμό ακολουθιών σχεδόν διαδοχικών πρώτων αριθμών [Kin13].

Η βασική πάντως κατεύθυνση για την ελάττωση της άσκοπης κατανάλωσης ενέργειας είναι η αλλαγή κριτηρίου επιλογής του κόμβου που αποκτάει το προνόμιο κατάθεσης ενός νέου block. Ένα τέτοιο εναλλακτικό κριτήριο είναι η *απόδειξη μεριδίου* (proof of stake, PoS). Στην περίπτωση αυτήν ο κόμβος επιλέγεται στη βάση του ποσού που έχει στην κατοχή του, δηλαδή γίνεται τυχαία επιλογή με πιθανότητες ανάλογες του μεριδίου του καθενός [Wan+19]. Μια παραλλαγή υπολογίζει αυτές τις πιθανότητες λαμβάνοντας υπ᾽ όψη και τον χρόνο κατοχής των ποσών. Το Ethereum, που χρησιμοποιεί την απόδειξη έργου όπως το BitCoin, έχει ανακοινώσει ότι σκοπεύει να πραγματοποιήσει μετάβαση προς το PoS στην έκδοση 2.0 μέσα στο 2020.

Για το νόμισμα *Libra* του Facebook έχει ανακοινωθεί ότι δεν θα ακολουθήσει την απόδειξη έργου, αλλά ενός πρωτοκόλλου συναίνεσης που ονομάζεται *Libra Byzantine Fault Tolerance* (LibraBFT). Το LibraBFT εμφανίζει αντοχή όταν το ποσοστό των δολίων κόμβων δεν ξεπερνάει το 1/3 και όταν το δίκτυο λειτουργεί κάτω από συνθήκες μερικού συγχρονισμού [Bau+19].

12.6 Ethereum

Το *Ethereum* [VJR18; AW18] είναι ένα σύστημα με στόχο τη συγγραφή και ενεργοποίηση συμβολαίων σε ένα κατανεμημένο κατάστιχο. Συνδέεται για την υλοποίηση συναλλαγών με ένα κρυπτονόμισμα που λέγεται *ether*. Τα scripts που χρησιμοποιούνται στο Ethereum γράφονται σε μια γλώσσα, η οποία είναι Turing complete (πράγμα που δεν ισχύει για τα scripts του BitCoin), δηλαδή μπορεί να περιγράψει οποιονδήποτε υπολογισμό μπορεί κάνει ένας γενικού σκοπού υπολογιστής, περιλαμβανομένων και βρόχων. Ένας χαρακτηρισμός που έχει δοθεί στο Ethereum είναι πως είναι ένας *παγκόσμιος υπολογιστής*, δηλαδή μια μηχανή που βρίσκεται κάθε στιγμή σε κάποια κατάσταση γνωστή σε όλους. Είναι μια υπολογιστική υποδομή που βασικό της στόχο έχει να εκτελεί προγράμματα που ονομάζονται *έξυπνα συμβόλαια* (smart contracts) [AW18]. Η βασική ιδέα των σχεδιαστών του Ethereum ήταν να προσφέρουν ένα γενικής χρήσης blockchain, πάνω στο οποίο ο κάθε προγραμματιστής θα μπορούσε να στήσει την εφαρμογή που επιθυμεί, βρίσκοντας όμως έτοιμους τους μηχανισμούς για δίκτυο ισοτίμων, για blockchain και για αλγόριθμους συναίνεσης.

Το Ethereum κληρονομεί πολλές από τις μεθόδους και τα πρωτόκολλα που χρησιμοποιεί το BitCoin, αποσκοπώντας στο να υιοθετεί ήδη δοκιμασμένες τεχνικές και να επαναχρησιμοποιεί ήδη υπάρχον λογισμικό.

Για την επίτευξη συναίνεσης το Ethereum χρησιμοποιεί το πρωτόκολλο GHOST (Greedy Heaviest Observed Subtree)[SZ15]. Το GHOST επιλέγει σε κάθε διακλάδωση το βαρύτερο υπο-δένδρο που έχει ρίζα στη διακλάδωση. Το πρωτόκολλο αυτό επιτρέπει να πραγματοποιούνται περισσότερες συναλλαγές ανά μονάδα χρόνου. Μια από τις κριτικές στο BitCoin είναι ότι πιθανώς δεν θα μπορούσε να αντέξει σε όγκους συναλλαγών αντίστοιχους με ένα κανονικό νόμισμα, δηλαδή ότι δεν θα μπορούσε να εξυπηρετήσει τους χρήστες του αν γινόταν ευρύτερη η αποδοχή του.

Ο χρήστης του Ethereum είναι υποχρεωμένος με κάθε συναλλαγή να πληρώσει ένα επιπρόσθετο ποσό που καλύπτει τα έξοδα για τους απαιτούμενους υπολογισμούς. Κατά τη διάρκεια των υπολογισμών μέρος του ποσού απορροφάται, ενώ το υπόλοιπο επιστρέφεται στον χρήστη που το πλήρωσε.

Το Ethereum είναι ένα εξελισσόμενο σύστημα, του οποίου οι γρήγορες μεταβολές συχνά δεν εγγυώνται την συμβατότητα με παρελθούσες υλοποιήσεις,

ΚΕΦΑΛΑΙΟ 13

Ιδιωτικότητα

13.1 Εισαγωγή

Το δικαίωμα στην ιδιωτικότητα (privacy) σύμφωνα με έναν ορισμό του 1890 θεωρείται ειδική περίπτωση του δικαιώματος στην προσωπική ησυχία και μοναξιά (*right to be let alone*, [WB90]). Σύμφωνα με ένα σχετικά νεότερο ορισμό στο *Random House Unabridged Dictionary* ιδιωτικότητα είναι η κατάσταση του να είναι κάποιος ελεύθερος από εισβολή ή ενόχληση στις προσωπική του ζωή και στις προσωπικές του υποθέσεις.

Έχουν γίνει προσπάθειες να ορισθεί η ιδιωτικότητα ως κάτι απτό μέσω της ζημιάς που προκαλείται όταν υπάρξει απώλεια ιδιωτικότητας. Για παράδειγμα, κάποιος μπορεί να χάσει την εργασία του επειδή συνέβη μια διαρροή από τον ιατρικό του φάκελλο. Η διαπίστωση της ζημιάς είναι απαραίτητη σε πολλές χώρες προκειμένου να κινηθούν νομικές διαδικασίες. Ωστόσο ένας τέτοιος ορισμός γίνεται προβληματικός σε δύο τουλάχιστον περιπτώσεις [Lan16]: (α) Όταν η ζημιά εντοπίζεται στον ψυχικό κόσμο του θύματος. Ένα άτομο μπορεί π.χ. να μην αισθάνεται άνετα με τους γείτονές του αν αυτοί γνωρίζουν το ιατρικό του παρελθόν. (β) Όταν η ζημιά γίνεται χωρίς να το γνωρίζει, π.χ. κάνει μια αίτηση πρόσληψης και ο εργοδότης τον απορρίπτει χωρίς εξηγήσεις επειδή έχει πρόσβαση σε προσωπικά του δεδομένα.

Στην εποχή της πληροφορικής η έννοια της ιδιωτικότητας συνδέεται με το δικαίωμα του ατόμου να ασκεί έλεγχο στη διάδοση πληροφοριών που το αφορούν προς άλλα άτομα και οργανισμούς [Cla99]. Η «προσωπική ησυχία» κάποιου πιθανότατα θα παραβιασθεί αν προηγουμένως παραβιασθεί το δικαίωμα στη διαχείριση της προσωπικής πληροφορίας. Γενικότερα οι συνέπειες μπορεί να είναι βαρύτερες από μια ενόχληση και να επεκτείνονται σε παρεμπόδιση της άσκησης άλλων δικαιωμάτων, όπως το δικαίωμα στην εργασία, στην ελευθερία της έκφρασης, στην ασφάλεια, στη συναισθηματική ισορροπία, σε ένα αποδεκτό επίπεδο ζωής κ.α.

Ωστόσο αν ο έλεγχος της διάδοσης προσωπικών πληροφοριών ορισθεί απλώς ως δικαίωμα, του οποίου κάποιος μπορεί να κάνει χρήση αν το επιθυμεί, το αποτέλεσμα θα είναι ισχνό εφόσον δεν υπάρχουν και τα κατάλληλα εργαλεία άσκησης αυτού του ελέγχου. Ελάχιστοι είναι σε θέση να διαθέσουν χρόνο και χρήμα για την άσκηση ενός τέτοιου ελέγχου. Για το λόγο αυτόν ο οποιοσδήποτε έλεγχος είναι ορθότερο να γίνεται κατά το δυνατόν αυτόματα και στην πηγή.

Δεν υπάρχει ενιαίος ορισμός του τι ακριβώς καλύπτεται από τον «ιδιωτικό χώρο». Τα ζητήματα ιδιωτικότητας για ένα άτομο ή μια κοινωνική ομάδα είναι αυτά που έχει αποφασίσει ότι είναι σημαντικά για οποιονδήποτε λόγο και με οποιοδήποτε κριτήριο. Η απόφαση αυτή δεν είναι αιώνια, μπορεί λίγο αργότερα να μεταβληθεί προς οποιαδήποτε κατεύθυνση, δηλαδή της αυξημένης ή της μειωμένης ιδιωτικότητας και της προστασίας διαφορετικού είδους πληροφοριών.

Ας δούμε μερικά παραδείγματα. Στο τέλος της δεκαετίας του '70 οι βιντεοκάμερες ήταν πολύ ακριβές για ιδιωτική χρήση και υπήρχαν μόνο σε τηλεοπτικά στούντιο. Οι πρώτοι ίσως που χρησιμοποίησαν εκτός τηλεοπτικών σταθμών βιντεοκάμερες ήταν ορισμένα πλοία αναψυχής (κρουαζιερόπλοια) και πλοία που προσέφεραν υπερατλαντικά ταξίδια. Τα πλοία αυτά είχαν ήδη τηλεοράσεις στις καμπίνες και προσπαθούσαν να κάνουν όσο γινόταν πιο ευχάριστο το χρόνο των επιβατών τους. Ανάμεσα στις πρώτες απ᾽ ευθείας «εκπομπές» που δοκίμασαν ήταν να δείχνουν τις εκδηλώσεις, συνήθως χορό με ορχήστρα, που γινόταν που γινόταν στα σαλόνια τους. Οι δοκιμές σ᾽ αυτούς τους χώρους σταμάτησαν πολύ γρήγορα επειδή διαπιστώθηκε ότι κατέληγαν σε καβγάδες ζευγαριών, όταν το ένα μέλος είχε μείνει στην καμπίνα και το άλλο ήταν στο σαλόνι και χόρευε με άλλη κυρία ή κύριο. Φυσικά εκείνη την εποχή δεν υπήρχαν (λόγω κόστους) κάμερες ασφαλείας και δεν είχε προλάβει να διαμορφωθεί καμιά γενικότερη συναντίληψη για θέματα ιδιωτικότητας από τη χρήση καμερών.

Αντίθετα σήμερα, που το κόστος δεν αποτελεί περιοριστικό παράγοντα, οι κάμερες πολλαπλασιάζονται καθημερινά και αντίστοιχα αυξάνεται η ευαισθησία απέναντι στη χρήσης τους, που ποικίλλει ανάλογα με τις συνθήκες και μπορεί να ζυγιστεί απέναντι στα αναμενόμενα οφέλη. Στο μετρό της Αθήνας είχε επιτραπεί από την Αρχή για την Προστασία Δεδομένων Προσωπικού Χαρακτήρα η χρήση στους σταθμούς, όχι όμως μέσα στα βαγόνια. Το 2015 επεκτάθηκε στα τελευταία με νεότερη απόφαση της αρχής με κύρια επιχειρήματα «την πρόληψη πιθανών ανθρώπινων θυμάτων και τη μείωση των ζημιών που θα προκληθούν από την βλάβη των εγκαταστάσεων και τη διακοπή των ζωτικών υπηρεσιών». Ανάμεσα στα λοιπά επιχειρήματα, που ομολογούνται ανοιχτά ή απλώς υπονοούνται σε τέτοιες περιπτώσεις, είναι η προστασία της περιουσίας της εταιρίας από βανδαλισμούς και ίσως ακόμη η προστασία των επιβατών από μικροκακοποιούς. Στη συνέχεια είναι προφανές ότι θα επωφεληθούν υπηρεσίες δίωξης του εγκλήματος και άλλες κρατικές υπηρεσίες. Σε τέτοιες περιπτώσεις το τυπικό ερώτημα είναι αν για να παρασχεθούν οι σχετικές διευκολύνσεις τηρούνται όλες τις φορές οι νόμιμες διαδικασίες (άδεια από εισαγγελική αρχή κ.λπ.). Παράλληλα είναι βέβαιο ότι ένα μέρος της κοινωνίας θεωρεί τέτοια μέσα άμεσης παρακολούθησης ανεπιθύμητα και καταχρηστικά ανεξάρτητα από την οποιαδήποτε θεσμική τους κάλυψη.

Παρ᾽ όλο που οι βιντεοκάμερες σήμερα είναι το κατ᾽ εξοχήν σύμβολο του Μεγάλου Αδελφού (Big Brother), τα πιο ευρέως διαδεδομένα και αποτελεσματικά μέσα παρακολούθησης είναι ένα σύνολο υπηρεσιών και εφαρμογών του διαδικτύου και της κινητής τηλεφωνίας. Ανάμεσά τους εξέχουσα θέση κατέχουν τα κοινωνικά δίκτυα και οι μηχανές αναζήτησης. Τα κοινωνικά δίκτυα είναι ένας εικονικός κοινωνικός χώρος μέσα στον οποίο τα μέλη τους αποκτούν μια αυταπάτη οικειότητας, που σε συνδυασμό με την τάση αυταρέσκειας και αυτοπροβολής τα κάνει να παραχωρούν οικειοθελώς ένα πλήθος πληροφοριών, περισσότερο ή λιγότερο προσωπικών. Η συλλογή πληροφοριών για τον καθένα μας και η μελέτη των προτιμήσεων και συνηθειών μας με σκοπό την στοχευμένη προώθηση προϊόντων και υπηρεσιών αποτελεί σημαντικό μέρος του λειτουργικού και οικονομικού μοντέλου των κοινωνικών δικτύων, καθώς και των μηχανών αναζήτησης [Sch16].

Στις νέες τάσεις που αναπτύσσονται και εξαπλώνονται αυτόν τον καιρό συγκατελέγεται η παρακολούθηση του ανθρώπινου σώματος με σκοπό τη βελτίωση της υγείας.

Μια σειρά από συσκευές, όπως έξυπνα βραχιόλια, ζώνες, ζυγαριές, ρολόγια κ.λπ., καταγράφουν ολοένα και περισσότερους δείκτες ζωτικών μας λειτουργιών. Αυτός είναι μόνο ο πρόλογος για το επερχόμενο *διαδίκτυο των πραγμάτων* (Internet of Things, IoT) και γενικώς τα έξυπνα αντικείμενα. Η συλλογή πληροφοριών από όλους αυτούς τους αισθητήρες που θα βρίσκονται γύρω μας μπορεί να αξιοποιηθεί πλέον ώστε ο κάθε ενδιαφερόμενος να μπορεί, νόμιμα ή παράνομα, να σχηματίζει εικόνα για μας σε πρωτόγνωρη πληρότητα μέσω εξελιγμένων εργαλείων ανάλυσης και να ξέρει περισσότερα από όσα ξέρουμε εμείς οι ίδιοι για τον εαυτό μας. Ένα «ανέκδοτο» που κυκλοφορεί είναι ότι κάποια μέρα ένας σχετικά πιο φιλάνθρωπος υπάλληλος μυστικής κρατικής υπηρεσίας μπορεί να σε πάρει τηλέφωνο και να σου πει ότι πρέπει να δεις ένα καρδιολόγο πριν να είναι πολύ αργά. Αντίστροφα, όταν πολλά από τα αντικείμενα που μας περιβάλλουν γίνουν ελέγξιμα από μακριά, περιλαμβανομένων και αυτών που σχετίζονται με τις ζωτικές μας λειτουργίες, οι δυνατότητες του εγκλήματος, και μάλιστα της ανθρωποκτονίας, θα διευρυνθούν ριζικά.

13.2 Η ιδιωτικότητα άλλοτε και τώρα

Σε προηγούμενους αιώνες η ιδιωτικότητα ήταν έννοια περίπου άγνωστη. Η έννοια του ιδιωτικού χώρου, που βασικά περιοριζόταν στο εσωτερικό του σπιτιού του καθενός, ήταν γνωστή π.χ. στην ελληνική και ρωμαϊκή αρχαιότητα, αλλά πέραν αυτού κανείς δεν μπορούσε να έχει ιδιαίτερες απαιτήσεις. Τα αδιάκριτα μάτια και οι εξωτερικές απειλές έβρισκαν εμπόδιο σε μια αρχιτεκτονική που είχε ανοίγματα μόνο προς το εσωτερικό (ενώ το άλλο άκρο είναι αυτό που συμβαίνει στα σημερινά σπίτια στις Κάτω Χώρες, όπου η διαφάνεια που επιτάσσει η προτεσταντική ηθική αφαιρεί ακόμη και τις κουρτίνες). Γενικά στο μεσαίωνα υπάρχει σε πολλές κοινωνίες μια σύνδεση του ύψους που έχει μια κατοικία (ή του εδάφους στο οποίο πατάει) με τη σχέση κοινωνικά ανώτερου-κατώτερου. Η υψομετρική διαφορά έχει πρακτική αξία επίθεσης και άμυνας, αλλά και παρακολούθησης.

Στο μεσαιωνικό χωριό, όπου τα σπίτια είναι πολύ κοντά, μερικές φορές κυριολεκτικά το ένα πάνω στο άλλο, ό,τι συμβαίνει μέσα σ᾽ ένα σπίτι ακούγεται ή φαίνεται από τα διπλανά, ενώ στη συνέχεια οι ειδήσεις μεταδίδονται με εκπληκτική ταχύτητα. Το πιο καλά φυλαγμένο μυστικό στον οικισμό προηγούμενων αιώνων ίσως ήταν πόσα νομίσματα έχει θαμμένα και σε ποιο σημείο ο εύπορος γείτονας, αλλά αυτό ήταν μόνο μια τεχνική λεπτομέρεια.

Γενικά στην παραδοσιακή κοινωνία των προηγούμενων αιώνων η κοινωνική πληροφόρηση περιορίζεται σε νησίδες. Μέσα σ᾽ αυτές η πληροφόρηση είναι ισχυρή, σχεδόν πλήρης, και η ιδιωτικότητα είναι έννοια άγνωστη, ενώ έξω από αυτές η πληροφόρηση για αυτά που συνέβαιναν μέσα έπεφτε απότομα στο μηδέν. Ο Jared Diamond (στο βιβλίο του "The World until Yesterday", [Dia12]) λέει για τις μικρές παραδοσιακές κοινωνίες ότι (α) όλοι γνωρίζουν όλους, (β) όλοι γνωρίζουν τα πάντα για όλους, (γ) όλοι έχουν γνώμη για το τι κάνουν όλοι οι άλλοι και (δ) αν προκύψουν διαφορές και διαμάχες φροντίζουν να τις λύσουν συναινετικά επειδή θα πρέπει να συνεχίσουν να ζουν μαζί. Το ελληνικό χωριό ως χθες ήταν κάπως έτσι, και πολλά χωριά παραμένουν σ᾽ αυτήν την κατάσταση και σήμερα.

Η διάχυση πληροφορίας προς και από τη μικρή παραδοσιακή κοινωνία ήταν ελάχιστη. Έξω από αυτήν το τυχόν μέλος του παραμένει εν πολλοίς άγνωστο. Συμμετρικά, γνωρίζει εξ ίσου ελάχιστα τι συμβαίνει αλλού. Υπάρχει στον Αστερίξ μια σκηνή όπου φτάνει στο χωριό τους ο Ιούλιος Καίσαρ. Ο Αστερίξ αναρωτιέται τι του θυμίζει ο καβαλάρης που ξεχωρίζει απ᾽ τους Ρωμαίους μέχρι τη στιγμή που καταλαβαίνει ότι μοιάζει

με τη μορφή που είναι χαραγμένη πάνω στα νομίσματα. Σε μια εποχή χωρίς φωτογραφία και χωρίς μαζικά μέσα ενημέρωσης ήταν δύσκολο να ξέρεις ακόμη και πώς μοιάζει ο αρχηγός του κράτους. Κατά κάποιο τρόπο αυτό που εμείς θα λέγαμε ιδιωτικότητα προστατεύεται αυτόματα, αλλά τα μέλη αυτών των κοινωνιών δε χρειάζεται καν να επινοήσουν μια έννοια για ένα πρόβλημα που δεν υπάρχει.

Η ανοχή των διαφορετικών κατοίκων του διπλανού χωριού έγινε δυνατή μέσα στις αυτοκρατορίες (π.χ. ρωμαϊκή, βυζαντινή κ.λπ.), όμως η βασική μεταβολή ήρθε με τις μεγάλες πόλεις, όπου κανείς μαθαίνει να ανέχεται να ζει δίπλα σε αγνώστους ειρηνικά, ενώ ταυτόχρονα μπορεί να παραμένει άγνωστος μεταξύ αγνώστων. Στην πόλη οι κοινωνικές σχέσεις δεν είναι στενές, δεν είναι ιδιαίτερα τοπικές (με την έννοια ότι ο μέσος κάτοικος πόλης γνωρίζει ελάχιστους από τους γείτονές του, ενώ οι συγγενείς του μπορεί να μένουν μακριά), οπότε οι «ειδήσεις» δεν διαδίδονται τόσο αποτελεσματικά και πάντως όχι με ισχυρή γεωγραφική συνάφεια. Στον *Ηλίθιο* του Ντοστογιέφσκι μια διαβόητη καλλονή της Πετρούπολης εμφανίζεται μπροστά στον ήρωα που περιέργως πως την αποκαλεί αμέσως με το όνομά της. Αυτή μένει έκπληκτη και τον ρωτάει πώς την ξέρει εφόσον δεν έχουν ξανασυναντηθεί ποτέ. Αυτός της εξηγεί ότι έχει κατά σύμπτωση δει μια φωτογραφία της στο σπίτι ενός κοινού γνωστού μόλις το ίδιο πρωί. Η γυναίκα αυτή λοιπόν μπορούσε να είναι διάσημη με την έννοια ότι όλοι ήξεραν το όνομά της και κάμποσες φήμες για την προσωπική της ζωή, αλλά μπορούσε να είναι βέβαιη ότι έξω από τον κύκλο της ελάχιστοι θα μπορούσαν να την αναγνωρίσουν εξ όψεως. Οι πόλεις, και μάλιστα πριν την εμφάνιση των μαζικών μέσων ενημέρωσης, αποτελούσαν την ιδανική κρυψώνα.

Η ανωνυμία μέσα στην πόλη φάνηκε ελκυστική σε πολλούς από τους νέους ή παλιούς κατοίκους της, γιατί τους έδινε την ευκαιρία να αναπτύσσουν σχέσεις και δραστηριότητες προσωπικές, πολιτικές, ακτιβιστικές μαζί με μια ποικιλία «άλλων», δραστηριότητες που συχνά ήταν στα όρια της κοινωνικής αποδοχής και του νόμου ή εκτός αυτών, χωρίς ισχυρό κοινωνικό έλεγχο, χωρίς δηλαδή τον κίνδυνο να τους κατακρίνουν γείτονες και γνωστοί. Η έκθεση βέβαια στην πόλη, εφόσον κάποιος γίνει γνωστός για καλό ή κακό, εφόσον εκλεγεί δήμαρχος ή συλληφθεί ως διάσημος κακοποιός, είναι πολύ μεγαλύτερη από τη στιγμή που υπάρχουν εφημερίδες, δελτία ειδήσεων και άλλα μέσα μαζικής πληροφόρησης. Αυτές όμως είναι οι εξαιρέσεις. Η ανωνυμία της μεγάλης πόλης έδωσε σημαντική ώθηση στη διαμόρφωση της σύγχρονης έννοιας και της συναντίληψης της ιδιωτικότητας. Πολλοί άνθρωποι εγκατέλειψαν τα χωριά τους στο τέλος του 20ου αιώνα και μετακόμισαν στις πόλεις όχι μόνο εξ αιτίας των περισσότερων δυνατοτήτων εξεύρεσης εργασίας, αλλά και εξ αιτίας της αυξημένης ιδιωτικότητας. Οι νέες κατακτήσεις δεν άργησαν να θεωρηθούν παγιωμένες και αυτονόητες ή, αν μη τι άλλο, άξιες να διεκδικηθούν κοινωνικά και πολιτικά.

Η διάδοση πληροφοριών προσωπικού χαρακτήρα έχει πάρει με τα κοινωνικά δίκτυα πρωτόγνωρες διαστάσεις. Κατά κανόνα η πληροφορία φτάνει σε επιλεγμένους αποδέκτες, που κι αυτοί είναι ίσως χαμένοι κάπου μέσα στην πόλη, πράγμα που διατηρεί την ανωνυμία του διπλανού. Όμως κι αυτή η βασική υπόθεση κινδυνεύει να διαταραχθεί χάρη σε μια σειρά από νέες υπηρεσίες. Μπορώ να μην περιμένω τη σύμπτωση και να πάω να βρω τους φίλους μου στο μπαρ όπου τώρα βρίσκονται, επειδή είναι μαρκαρισμένοι πάνω σ' ένα χάρτη. Μπορώ να δω πού είναι τραβηγμένες οι φωτογραφίες που έβαλαν φίλοι μου στο *Instagram*. Μπορώ να μάθω πώς λέγονται ορισμένα από τα άγνωστα άτομα που βρίσκονται γύρω μου επειδή είναι κι αυτά μέλη δικτύων με έμφαση στην τρέχουσα θέση, όπως τα *Foursquare, Gowalla, Whrrl, Brightkite, buzzd, GyPSii*. Μπορώ να μάθω πως η κοπέλλα απέναντί μου λέγεται Κατερίνα Χ σαρώνοντας τους υπολογιστές που είναι στο ίδιο WiFi (επειδή έχει δώσει στον υπολογιστή της το default όνομα *Katerina X's laptop*) και μετά να την ψάξω στο Google.

13.3 Το νομικό πλαίσιο

Η ανάλυση του νομικού πλαισίου γύρω από την ιδιωτικότητα ξεφεύγει από τους σκοπούς αυτού του κειμένου, αλλά ορισμένες βασικές γνώσεις είναι χρήσιμες. Μέσα από μια σειρά νόμων και οδηγιών, που μπορείτε να βρείτε στις ιστοσελίδες της Ευρωπαϊκής Επιτροπής και της (ελληνικής) *Αρχής Προστασίας Δεδομένων Προσωπικού Χαρακτήρα* ορίζονται και καθορίζονται τα θέματα ιδιωτικότητας σε γενικό, αλλά και σε πρακτικό επίπεδο. Εκτός από την παραπάνω ανεξάρτητη αρχή, σχετική είναι και η *Αρχή Διασφάλισης του Απορρήτου των Επικοινωνιών (ΑΔΑΕ)*, η οποία έχει σκοπό την προστασία του απορρήτου των επικοινωνιών.

Η γενική ιδέα που διέπει το νομικό πλαίσιο είναι ότι προστατεύεται ο ιδιωτικός βίος μαζί με τις σχετικές πληροφορίες, οι οποίες πρέπει να παραμένουν υπό τον έλεγχο του προσώπου στο οποίο αφορούν. Ωστόσο το δικαίωμα αυτό έρχεται σε αντίθεση με συμφέροντα και θέματα προστασίας του κράτους και του δημοσίου, με δικαιώματα άλλων μελών της κοινωνίας, με ορισμένες πλευρές της παροχής υπηρεσιών επικοινωνίας, καθώς και άλλων υπηρεσιών βασισμένων στην εξειδίκευσή τους στις ανάγκες του πελάτη. Στο βαθμό που όλα τα παραπάνω γίνονται αντιληπτά ως γενικότερες ανάγκες περνούν βαθμιαία στο νομικό πλαίσιο.

Στην περιοχή της πληροφορικής και των επικοινωνιών γενικά προστατεύονται τα προσωπικά δεδομένα από τη συλλογή και επεξεργασία, αλλά μόνον ως εκεί που δεν διακυβεύονται η κρατική και η δημόσια ασφάλεια, δεν παρεμποδίζεται η δίωξη του εγκλήματος και η άσκηση άλλων επίσης βασικών δικαιωμάτων οποιουδήποτε προσώπου. Οι εξαιρέσεις στην προστασία υποτίθεται ότι πρέπει να περιγράφονται σαφώς και οι σχετικές διαδικασίες να τίθενται κάτω από αυστηρό έλεγχο.

Σύμφωνα με το νόμο 2472/1997 του ελληνικού κράτους τα *προσωπικά δεδομένα* είναι εκείνα που οδηγούν κατά τρόπο μονοσήμαντο σε ένα συγκεκριμένο πρόσωπο. Ορίζονται επίσης ως *ευαίσθητα* δεδομένα που έχουν να κάνουν με την φυλετική ή εθνική προέλευση, τα πολιτικά φρονήματα, τις θρησκευτικές ή φιλοσοφικές πεποιθήσεις κ.α. Στο άρθρο 5 ορίζεται ότι η *επεξεργασία δεδομένων προσωπικού χαρακτήρα επιτρέπεται μόνον όταν το υποκείμενο των δεδομένων έχει δώσει τη συγκατάθεσή του.* Συγχρόνως όμως στο ίδιο άρθρο ορίζεται μια σειρά εξαιρέσεων, που έχουν να κάνουν με τις λειτουργίες του κράτους και του δημοσίου, καθώς και με τα δικαιώματα άλλων υποκειμένων που συνάπτουν συμβάσεις με το υποκείμενο των προσωπικών δεδομένων. Στη συνέχεια ορίζονται οι προϋποθέσεις επεξεργασίας τέτοιων δεδομένων και η υποχρέωση όσων κάνουν τέτοια επεξεργασία να ειδοποιούν την αρχή προστασίας.

Στο γενικότερο ευρωπαϊκό πλαίσιο ισχύουν επίσης διάφοροι νόμοι και διατάξεις. Στον *Χάρτη των Θεμελιωδών Δικαιωμάτων της Ευρωπαϊκής Ένωσης* (2010) και στο άρθρο 7 αναφέρεται ότι *κάθε πρόσωπο έχει δικαίωμα στο σεβασμό της ιδιωτικής και οικογενειακής ζωής του, της κατοικίας του και των επικοινωνιών του*. Ανάλογες είναι και οι παλιότερες προβλέψεις του άρθρου 8 της σύμβασης για τα ανθρώπινα δικαιώματα του 1950 (*Human Rights Convention*, 4/11/1950), αλλά συνοδεύονται με μια σειρά εξαιρέσεων που έχουν να κάνουν με την κρατική και δημόσια ασφάλεια, καθώς και με τα έννομα δικαιώματα τρίτων.

Οι οδηγίες 2002/58/ΕΚ και 2009/136/ΕΚ του Ευρωπαϊκού Κοινοβουλίου ασχολούνται με τα δικαιώματα των χρηστών όσον αφορά δίκτυα και υπηρεσίες ηλεκτρονικών επικοινωνιών. Με δυο λόγια περιγράφουν τι συμβαίνει με τα θέματα ιδιωτικότητας και προστασίας προσωπικών δεδομένων κατά τη χρήση τηλεπικοινωνιακών δικτύων και προβλέπουν τις σχετικές υποχρεώσεις των παρόχων τους. Σύμφωνα με την 2002/58/ΕΚ οι χώρες της ΕΕ οφείλουν να εξασφαλίζουν ότι η αποθήκευση και επεξεργασία των πληροφοριών των σχετικών με τον χρήστη των δικτύων επιτρέπονται μόνο

αφού προηγουμένως ο χρήστης έχει πληροφορηθεί την έκταση και το σκοπό της επεξεργασίας και του έχει δοθεί το δικαίωμα άρνησης. Ωστόσο στην παρ. (31) της οδηγίας του 2002 χρησιμοποιείται η εξής λίαν χαλαρή διατύπωση: *Το αν για την επεξεργασία των δεδομένων προσωπικού χαρακτήρα ενόψει της παροχής μιας συγκεκριμένης υπηρεσίας προστιθέμενης αξίας θα πρέπει να απαιτείται η συγκατάθεση του χρήστη ή του συνδρομητή, εξαρτάται από τα δεδομένα προς επεξεργασία και από τον τύπο της υπηρεσίας ...* Με τις οδηγίες αυτές καθορίζονται αρκετές λεπτομέρειες στην επικοινωνία με τηλεφωνία και Internet, π.χ. πότε και κάτω από ποιες συνθήκες επιτρέπεται να αποθηκεύονται cookies.

Παρά τα προβλήματα, το ευρωπαϊκό νομικό πλαίσιο θεωρείται γενικά πιο αυστηρό από το αντίστοιχο βορειοαμερικανικό. Οι αμερικανικές εταιρίες που επιθυμούν να δραστηριοποιούνται στην ΕΕ υποχρεώνονται να σέβονται τους πρόσθετους περιορισμούς της ΕΕ, όπως π.χ. στο θέμα του opt-in για τα cookies. Επίσης, εταιρίες που κάνουν ευρεία συλλογή δεδομένων, όπως οι Google και Facebook, έχουν επανειλημμένα αντιμετωπίσει νομικά προβλήματα. Για παράδειγμα, το βελγικό αρμόδιο δικαστήριο (Belgian Court of First Instance) επέβαλε στο Facebook πρόστιμο 250.000 Ευρώ ημερησίως επειδή συνεχίζει να καταγράφει τη συμπεριφορά και για μη μέλη του [Gel16].

Η γενική εικόνα που προσδιορίζεται από το παραπάνω νομικό πλαίσιο ούτως ή άλλως είναι μακριά από το να είναι ιδανική εφόσον προβλέπονται ποικίλες εξαιρέσεις και μάλιστα με ασαφείς διατυπώσεις. Ταυτόχρονα στην καθημερινή πρακτική οι παραβιάσεις είναι συχνές από ιδιωτικές και από κρατικές οντότητες. Για παράδειγμα, το γεγονός ότι σε πλείστα όσα κράτη υπάρχουν αυστηρές προβλέψεις για τις τηλεφωνικές παρακολουθήσεις δεν εμποδίζει μυστικές και αστυνομικές υπηρεσίες σε αρκετές περιπτώσεις (εκ των οποίων πολλές είναι γνωστές και τεκμηριωμένες στη βιβλιογραφία) να τις παρακάμπτουν κατά περίπτωση (εν μέρει καλυπτόμενες από εξαιρετικές διατάξεις προστασίας της εθνικής ασφάλειας ή του πολίτη). Τέτοιες δραστηριότητες θεωρούνται κατά κανόνα ούτως ή άλλως «νόμιμες» όταν οι παρακολουθούμενοι είναι πολίτες άλλου κράτους. Σε περιόδους πολέμου, ανασφάλειας και άλλων έκτακτων περιστατικών οι νόμοι προστασίας της ιδιωτικότητας αναστέλλονται ή αναθεωρούνται προκειμένου να γίνει ευκολότερη η πρόσβαση των αρχών στα δεδομένα. Στον ιδιωτικό τομέα μια σειρά από εταιρίες συλλέγουν δεδομένα πελατών με αμφίβολες μεθόδους και αποσπώντας τη συναίνεση του αδαούς πελάτη με κοινωνική μηχανική (social engineering). Τα κέρδη από τέτοιες πρακτικές είναι συχνά πολύ μεγαλύτερα από τα πιθανά πρόστιμα. Τέλος, ένα σύνολο ιδιωτών παρέχει εν γένει παράνομες υπηρεσίες παρακολούθησης και υπεξαίρεσης δεδομένων σε όποιον πελάτη τις ζητήσει. Όλα αυτά σίγουρα δεν συνθέτουν μια ιδανική εικόνα, επειδή ακόμη κι αν κανείς δεχτεί την αναγκαιότητα τέτοιων εξαιρέσεων, μένει αρκετός χώρος εκμετάλλευσης από άτομα που είναι στατιστικώς βέβαιο ότι θα υπάρξουν και θα επωφεληθούν εκ του πονηρού από τις εξαιρέσεις.

13.4 Το κόστος από την απώλεια της ιδιωτικότητας

Η απώλεια της ιδιωτικότητας ή γενικότερα η διαρροή «προσωπικών» πληροφοριών σχετίζεται εν γένει με αυξημένους κινδύνους, των οποίων το φάσμα ξεκινάει από την απλή ενόχληση και σε ακραίες περιπτώσεις φτάνει στην απώλεια της περιουσίας και της ζωής. Ως τώρα τις διαρροές ιδιωτικών ή προσωπικών πληροφοριών τις βλέπαμε μέσα σ' ένα πλαίσιο κοινωνικών σχέσεων, σε κοινωνικούς χώρους (ιδιωτικούς, εργασιακούς, δημόσιους, χώρους κοινωνικών ομάδων κ.λπ.), εξετάζοντας τι συμβαίνει κάθε φορά εξ αιτίας αυτών των διαρροών. Αυτό σημαίνει ότι οι πληροφορίες φτά-

νουν σε άτομα που γνωρίζουν ένα συγκεκριμένο υποκείμενο και τα οδηγούν σε περαιτέρω δράσεις επιβλαβείς για το τελευταίο. Για παράδειγμα αν διαρρεύσει μια πληροφορία από τον ιατρικό φάκελλο ενός ατόμου που αιτείται την πρόσληψή του σε μια εταιρία, η πρόσληψη αυτή μπορεί να ακυρωθεί από τον υπεύθυνο για τις προσλήψεις. Αν σε ένα πολιτικό κόμμα φτάσουν πληροφορίες για το ανάρμοστο πολιτικό ή άλλο παρελθόν ενός μέλους του, το περιστατικό αυτό μπορεί να οδηγήσει στη διαγραφή του μέλους.

Ωστόσο η πρόβλεψη των κινδύνων από την απώλεια ιδιωτικότητας ή και η αποτίμηση των ζημιών δεν αποτελούν απλά ζητήματα. Οι ζημίες από τη διαρροή ιδιωτικών πληροφοριών ποικίλλουν εξ αιτίας μιας σειράς παραγόντων όπως [AL09]

- τι είναι νόμιμο και τι είναι παράνομο,
- τι είναι κοινωνικώς αποδεκτό και τι δεν είναι,
- ποιο είναι το καθεστώς των κοινωνικών ελευθεριών στη συγκεκριμένη κοινωνία,
- ποια είναι η κατανομή εξουσίας και πώς διαμορφώνονται οι κοινωνικές, οικονομικές και άλλες σχέσεις,
- ποιος είναι ο χρονισμός της διαρροής και ποια η σύμπτωσή της με άλλα σχετικά γεγονότα,
- ποιο είναι το μέγεθος του ακροατηρίου και πόσο γνωστός είναι κάποιος μέσα σ᾽ αυτό το ακροατήριο,
- πόσο μακριά θα φτάσουν οι πληροφορίες, πόσοι θα τις επεξεργασθούν, με πόσο ισχυρά εργαλεία ανάλυσης και με πόσο ισχυρά μέσα παρέμβασης στη ζωή μας.

Αντίστροφα, οποιαδήποτε επιζήμια διαρροή πληροφορίας μπορεί να χαρακτηρισθεί απώλεια ιδιωτικότητας; Με την τελευταία εν γένει υπονοούνται διαρροές πληροφοριών κάποιας έκτασης, δηλαδή ότι ο «αντίπαλος», όποιος κι αν είναι, γνωρίζει για το «θύμα» κάτι παραπάνω από μια αποσπασματική πληροφορία και ότι οι ζημιές που υφίσταται το τελευταίο έχουν μια κοινωνική απόχρωση ή τέλος πάντων επηρεάζουν καθ᾽ οιονδήποτε τρόπο την καθημερινή του ζωή και τις σχέσεις του με το κοινωνικό περιβάλλον. Αν όμως ένας χάκερ από μια μακρινή περιοχή του πλανήτη σας υποκλέψει ένα μεμονωμένο τραπεζικό κωδικό, επειδή έτυχε να είστε ανάμεσα στα πολλά θύματα μιας ευρείας έκτασης επίθεσης μέσω ενός botnet, αυτό θα το θεωρήσετε απώλεια ιδιωτικότητας; Εδώ ας ληφθεί υπόψη ότι ο θύτης δεν έχει καμμιά άλλη γνώση για το θύμα και αντιστρόφως. Τέτοιες μεμονωμένες διαρροές λίγοι θα τις θεωρούσαν προσβολή της ιδιωτικότητας, παρά το γεγονός ότι μπορεί να είναι εξαιρετικά επιζήμιες.

Η ιδιωτικότητα ως οικονομικό αγαθό

Οι προσωπικές πληροφορίες και τα αντίστοιχα δεδομένα παρά τις υποτιθέμενες προσπάθειες προστασίας τους και περιορισμού της διακίνησής τους, γίνονται αντικείμενο αγοραπωλησιών όπως οποιοδήποτε άλλο οικονομικό αγαθό. Κατά συνέπεια τίθεται αργά ή γρήγορα το ζήτημα της αξίας και της τιμής αυτού του αγαθού. Η αξία της ιδιωτικότητας αποτιμάται στην πράξη σε διαφορετικές περιστάσεις όπως (α) όταν καλείται ένα υποκείμενο να πληρώσει για την προστασία των δικών του δεδομένων,

(β) όταν το ίδιο υποκείμενο καλείται να παραδώσει προσωπικά του δεδομένα με αντάλλαγμα αμοιβές, υπηρεσίες ή άλλα οφέλη, (γ) όταν μια συλλογή προσωπικών δεδομένων γίνεται αντικείμενο αγοραπωλησίας ανάμεσα σε άτομα, εταιρίες και οργανισμούς.

Η αξία των προσωπικών δεδομένων ενός συγκεκριμένου προσώπου για το ίδιο το εν λόγω πρόσωπο θα ήταν λογικό να συναρτάται με την αναμενόμενη ζημιά από την απώλειά τους, θέμα που αναφέρθηκε στην προηγούμενη ενότητα. Ωστόσο ο μέσος άνθρωπος πολύ σπάνια μπαίνει στον κόπο να κάνει κάποιου είδους βαθειά ανάλυση με σκοπό την εκτίμηση της αξίας των προσωπικών του δεδομένων. Οι πρόχειρες αναλύσεις που κάνει εξαρτώνται από το περιβάλλον (context), τις πρόσκαιρες διαθέσεις του και άλλους τυχαίους παράγοντες. Μπορεί ακόμη να διαφέρουν με διαφορετικές αρχικές συνθήκες ιδιωτικότητας, αν δηλαδή ερωτάται ένα άτομο πόσο θα πλήρωνε για να διατηρήσει ένα προνόμιο ιδιωτικότητας που ήδη έχει και θα ήθελε να συνεχίσει να έχει ή πόσο θα πλήρωνε για να το αποκτήσει δεδομένου ότι δεν το έχει.

Πιο συχνά ένα υποκείμενο έρχεται αντιμέτωπο με μια ερώτηση που έχει δίτιμη απάντηση, ναι-όχι. Συχνά χρησιμοποιείται παραπλανητική διατύπωση, π.χ. «τα cookies μας βοηθούν στη λειτουργία αυτής της σελίδας, ναι το κατάλαβα» (παρ' όλο που άλλη είναι η έννοια της κατανόησης και άλλη της αποδοχής). Έρευνες της ανθρώπινης συμπεριφοράς σε τέτοιες περιπτώσεις δείχνουν ότι η απάντηση για την αξία της ιδιωτικότητας δεν εξαρτάται μόνο από το ποιος ερωτάται, αλλά και με ποιον τρόπο διατυπώνεται η ερώτηση [AJL13]. Γενικά οι ερωτώμενοι για προσωπικά τους θέματα είναι πιο πρόθυμοι να τα αποκαλύψουν εφόσον η αποκάλυψη συνδέεται με κάποιο άμεσο όφελος, αλλά το πιο επιτυχημένο κίνητρο είναι αυτό που εκμεταλλεύεται την ανάγκη για αυτο-επίδειξη. Η πλευρά που προσπαθεί να αποσπάσει πληροφορίες έχει αναπτύξει μια σειρά από τεχνικές που παρακάμπτουν τις αντιστάσεις του κάθε ατόμου. Μερικές από αυτές είναι:

- Να γίνεται μια προκατασκευασμένη σειρά από ερωτήσεις στον πελάτη αυξανόμενης αδιακρισίας, μέχρις ότου αυτός να διαμαρτυρηθεί. Η τεχνική αυτή βασίζεται κατά κύριο λόγο στην ανοχή του πελάτη και στη μειωμένη του πληροφόρηση.

- Να δίνονται μικρές αμοιβές. Παράδειγμα είναι οι λεγόμενες κάρτες αφοσίωσης (loyalty cards) που χρησιμοποιούνται σε καταστήματα και μαζεύουν πόντους για λογαριασμό του κατόχου τους. Με τις κάρτες αυτές το κατάστημα μελετάει την αγοραστική συμπεριφορά των κατανωλωτών γενικά και του συγκεκριμένου καταναλωτή ειδικότερα (οπότε μπορεί να του κάνει π.χ. «προσφορές» σχετικές με τα ενδιαφέροντά του). Οι αμοιβές μπορεί να είναι στο χώρο της κοινωνικής καταξίωσης, π.χ. πολλά "like", τα οποία στη συνέχεια μπορεί να μετατραπούν ενδεχομένως σε πιο απτά οφέλη.

- Να γίνονται επανειλημμένες κρούσεις απόσπασης της ίδιας πληροφορίας μέχρις ότου το θύμα υποκύψει. Αυτή είναι μια στρατηγική που εφαρμόζεται από κοινωνικά δίκτυα, εταιρίες παροχής υπηρεσιών, εταιρίες διεξαγωγής δημοσκοπήσεων κ.α.

- Να παρακολουθείται, να καταγράφεται και να αναλύεται η συμπεριφορά του χρήστη μηχανών αναζήτησης, κοινωνικών δικτύων, υπηρεσιών ηλεκτρονικού ταχυδρομείου, παιχνιδιών, ηλεκτρονικών καταστημάτων, διαφημιστικών ιστοτόπων κ.λπ.

Παράλληλα προσπαθεί να δημιουργηθεί και μια αγορά που προβάλλει τη θετική πλευρά της ιδιωτικότητας και προσπαθεί να βρει πελάτες που θα πληρώσουν για να προστατευθούν. Η πρόταση για προστασία μπορεί να είναι άμεση, όπως στις διαφημίσεις των VPN που εγγυώνται ανωνυμία με μια μηνιαία συνδρομή, ή έμμεση, με την αφαίρεση των διαφημίσεων από εφαρμογές. Η τελευταία υπόσχεται κυρίως μείωση μιας ενόχλησης, αλλά δευτερευόντως αφαιρεί κι ένα μέσο παρακολούθησης των ενδιαφερόντων του χρήστη.

13.5 Μέθοδοι προστασίας της ιδιωτικότητας

Όπως δεν υπάρχει μια γενική και ομοιόμορφη αντίληψη για την ιδιωτικότητα, παρομοίως δεν υπάρχει μια γενική λύση για την προστασία της. Υπάρχει αντίθετα μια ποικιλία μέτρων προστασίας για διαφορετικές περιπτώσεις. Παρ' όλα αυτά μερικά μέσα προστασίας έχουν μια γενικότερη εφαρμογή και κάποιοι προσπαθούν ακόμη να ορίσουν ένα σύνολο από στοιχειώδη και αντιπροσωπευτικά προβλήματα ιδιωτικότητας, ώστε να διευκολύνουν την επεξεργασία αντίστοιχων λύσεων.

Ένα γενικό ζήτημα, το οποίο έχει περάσει και στη νομοθεσία, είναι κατά πόσο οι πληροφορίες που συλλέγονται για ένα σκοπό είναι υπερβολικές για την εξυπηρέτηση του συγκεκριμένου σκοπού. Η βιντεοκάμερα θεωρείται σκληρό μέσο παρακολούθησης διότι αποτυπώνει μια πληθώρα πληροφοριών, που κατά κανόνα δεν είναι όλες απαραίτητες. Αν για παράδειγμα ο σκοπός είναι να προλαμβάνονται ατυχήματα σε ένα σταθμό του μετρό από μια εφαρμογή που χτυπάει ένα συναγερμό όταν κάποιο άτομο πλησιάσει πολύ τις γραμμές, η αποτύπωση των προσώπων όσων πλησιάζουν, και ακόμη περισσότερο όσων δεν πλησιάζουν, είναι περιττή πληροφορία.

Ορισμένα συστήματα επιχειρούν να αποκτήσουν και να κρατήσουν μόνο την κρίσιμη πληροφορία.Ένα σύστημα εντοπισμού πυροβολισμών βασίζεται σε δίκτυο αισθητήρων που καταγράφουν τον ήχο των πυροβολισμών και υπολογίζουν τη θέση τους με τριγωνισμό [Zit08; Aue02]. Ένα άλλο είναι το «έξυπνο πάτωμα», το οποίο γνωρίζει τη θέση ενός ατόμου σε εσωτερικούς χώρους με χρήση αισθητήρων πίεσης [OA00]. Μια πιο συνηθισμένη περίπτωση είναι η τακτική ορισμένων καταστημάτων να μη συγκρατούν τα στοιχεία των πιστωτικών καρτών, αντίθετα με άλλα καταστήματα που όχι μόνο τα κρατούν σε μια βάση δεδομένων, αλλά και τα συνδυάζουν με πληροφορίες που παίρνουν από τον πελάτη την ώρα της αγοράς (π.χ. ημερομηνία γέννησης, διεύθυνση κατοικίας, τηλέφωνα κ.λπ.).

Δεν είναι όμως πάντοτε δυνατό να επινοηθεί και να λειτουργήσει ένα σύστημα που θα συλλέγει ακριβώς το είδος και την ποσότητα της πληροφορίας που είναι απολύτως απαραίτητα κι όχι παραπάνω. Πολλές φορές ένα ρεύμα πληροφορίας είναι πιο εύκολο να συλλεγεί ως έχει. Για παράδειγμα, μια σειρά από πακέτα που περνούν από ένα σημείο ενός δικτύου επικοινωνιών είναι πιο απλό να καταγραφούν ως έχουν. Άλλες φορές είναι εντελώς απαραίτητο να καταγράφονται τα πλήρη στοιχεία, όπως π.χ. σε μια βάση ιατρικών δεδομένων ή σ' ένα κτηματολόγιο. Η ιδιωτικότητα του ασθενή ή του πολίτη προφυλάσσονται κυρίως μέσω του ελέγχου πρόσβασης, δηλαδή δίνοντας τη δυνατότητα στα εν λόγω στοιχεία μόνο σε εξουσιοδοτημένα άτομα.

Ωστόσο τα ίδια δεδομένα συχνά πρέπει να χρησιμοποιηθούν για την εξαγωγή στατιστικών συμπερασμάτων, προκειμένου η κοινωνία να ενημερωθεί και να πάρει κατάλληλα μέτρα. Για παράδειγμα, είναι χρήσιμο να ξέρουμε ποιό ποσοστό του πληθυσμού έχει διαγνωσθεί με ηπατίτιδα Β και πώς αυτή κατανέμεται σε περιοχές ή επαγγέλματα. Ένα στατιστικό σύστημα βάσης δεδομένων δίνει απαντήσεις σε τέτοια ερωτήματα χωρίς να δίνει συγκεκριμένα ονόματα ατόμων στους στατιστικούς ερευνητές. Εναλλα-

κτικά, αντί των τελικών απαντήσεων προετοιμάζεται μια τροποποιημένη εκδοχή των ίδιων δεδομένων, όπου όμως δεν είναι δυνατό και πάλι να εντοπισθούν συγκεκριμένα πρόσωπα.

Η τεχνική με την οποία ετοιμάζεται αυτή η «θολωμένη» εκδοχή των στοιχείων ονομάζεται «ανωνυμοποίηση», πρόκειται δηλαδή για επεξεργασία των δεδομένων που αφαιρεί ή αλλοιώνει πληροφορίες που θα οδηγούσαν σε συγκεκριμένα πρόσωπα και θα παραβίαζαν την ιδιωτικότητά τους. Σε μια βάση ιατρικών δεδομένων εκ πρώτης όψεως θα μπορούσε κανείς να σβήσει τα ονόματα των ασθενών. Σε ένα ρεύμα πακέτων μπορεί κανείς να αντικαταστήσει τις πραγματικές διευθύνσεις IP πηγής και προορισμού με ψεύτικες. Θα δούμε όμως στη συνέχεια ότι η διαγραφή ή αντικατάσταση ορισμένων στοιχείων δεν λύνει το πρόβλημα. Σε ένα σύνολο εγγραφών που αφορούν πρόσωπα ακόμη και αν σβηστούν χαρακτηριστικές πληροφορίες όπως το ονοματεπώνυμο και η ημερομηνία γέννησης είναι πολύ πιθανό κάποιος να μπορέσει να αντιστοιχίσει εγγραφές σε συγκεκριμένα πρόσωπα, ενδεχομένως συνδυάζοντας πληροφορίες και από άλλες συλλογές δεδομένων. Γενικά καμμιά δημοσιοποίηση δεδομένων δεν είναι απόλυτα ασφαλής [DD79]. Τα προβλήματα αυτά θα τα εξετάσουμε χωριστά σε άλλη ενότητα πιο κάτω.

Τα θέματα ιδιωτικότητας σε δίκτυα υπολογιστών και επικοινωνιών σε μεγάλο ποσοστό είναι θέματα *απορρήτου της επικοινωνίας* ανάμεσα σε δυο ή περισσότερες πλευρές. Είναι επίσης θέματα που αφορούν στις υπηρεσίες που παρέχονται μέσω δικτύων. Ορισμένα θέματα ή και λύσεις ιδιωτικότητας σχετίζονται με τη χρήση των δικτύων από πολυάριθμους χρήστες. Τα πιο επικίνδυνα ζητήματα δημιουργούνται εξ αιτίας της χρήσης των δικτύων από υπηρεσίες που διαχειρίζονται χρήμα ή κρίσιμες υποδομές.

Οι πρόσφοροι τρόποι για την προστασία της ιδιωτικότητας στα δίκτυα και τις προσφερόμενες μέσω αυτών υπηρεσίες εξαρτώνται κατά κύριο λόγο από τη μορφή της πληροφορίας και τις ιδιότητες του δικτύου. Οι βασικές ιδέες και μέθοδοι στην περιοχή αυτή είναι οι εξής [LM+10]:

- Ασφαλής μεταφορά και αποθήκευση των πληροφοριών με μεθόδους κρυπτογράφησης.
- Παρεμπόδιση της πρόσβασης μη εξουσιοδοτημένων ατόμων σε δίκτυα που μεταφέρουν ή αποθηκεύουν προστατευόμενες πληροφορίες (με ειδικού τύπου δίκτυα και εξυπηρετητές, με μεθόδους ελέγχου πρόσβασης κ.λπ.).
- Απόκρυψη μέσα σε πλήθος χρηστών, μέσα σε πολυπλεγμένη ή ψευδή κίνηση, μέσα σε δίκτυα με δαιδαλώδη τοπολογία και δρομολόγηση, με χρήση κατανεμημένης και τμηματικής αποθήκευσης.
- Απόκρυψη ταυτότητας και χαρακτηριστικών ιδιοτήτων, χρήση κωδικών ονομάτων των οποίων την αντιστοιχία με τους πραγματικούς χρήστες γνωρίζει μόνο μια έμπιστη οντότητα, χρήση ψευδωνύμων για τον χρήστη κατά την πρόσβαση σε υπηρεσίες (που δεν χρειάζονται μνήμη των προτιμήσεων του χρήστη), ανωνυμοποίηση στοιχείων που προορίζονται για στατιστική χρήση.

Ένα άλλο παλιό και δημοφιλές σενάριο έχει να κάνει με τη δυνατότητα εξαγωγής συμπερασμάτων από την αποθήκευση ενός όγκου πληροφοριών. Ο κάτοχος μιας πιστωτικής κάρτας πραγματοποιεί μια σειρά από αγοράς στη διάρκεια μιας χρονικής περιόδου. Αν η κάρτα αυτή χρησιμοποιείται για όλες ή τις περισσότερες από τις αγορές του κατόχου της, μπορεί να εξαχθεί μια σειρά από συμπεράσματα όχι μόνο για τον γενικότερο τρόπο ζωής του, αλλά και για λεπτομέρειες της καθημερινής του συμπεριφοράς. Το μυστικό στην εξαγωγή συμπερασμάτων μπορεί να βρίσκεται στο συνδυασμό

των πληροφοριών που προδίδουν πολλές μαζί εγγραφές, ενώ κάθε μια ξεχωριστά δεν έχει ιδιαίτερη σημασία. Σήμερα υπάρχουν εταιρίες και οργανισμοί που μαζεύουν συστηματικά τέτοιες πληροφορίες, κυρίως μέσω του διαδικτύου, και στη συνέχεια αναλύουν τη συμπεριφορά είτε συγκεκριμένων ατόμων είτε πληθυσμιακών ομάδων.

13.6 Το απόρρητο των επικοινωνιών

Το απόρρητο των επικοινωνιών είναι μια από τις παλαιότερες έννοιες ιδιωτικότητας και δημιουργήθηκε αρχικά στους χώρους της ταχυδρομικής και τηλεφωνικής επικοινωνίας. Αρχαιότερη εξ αυτών προφανώς είναι η πρώτη.

Στην ταχυδρομική υπηρεσία

Ταχυδρομικές υπηρεσίες αναφέρονται αρχαία Αίγυπτο, στη Μεσοποταμία, στην Περσία του 6ου αιώνα π.Χ. (αναφέρονται από τον Ηρόδοτο και τον Ξενοφώντα), στην Κίνα κ.λπ. Ωστόσο δεν είναι γνωστό αν τέτοιες υπηρεσίες μπορούσαν να χρησιμοποιούνται από ιδιώτες και ποιοι κανόνες εφαρμόζονταν για την προστασία του περιεχομένου των επιστολών. Το πιθανότερο είναι ότι αφορούσαν μόνο στην ασφάλεια της κρατικής αλληλογραφίας. Στη Ρώμη η ανταλλαγή επιστολών ήταν οργανωμένη με διαφορετικά μέσα μεταφοράς (άλογα, βόδια, πεζούς) ανάλογα με τη χρήση και σημασία τους (με προτεραιότητα στις κρατικές υποθέσεις).

Η προστασία της ιδιωτικότητας των επιστολών εξασφαλίζεται σήμερα από την *Οικουμενική Διακήρυξη των Δικαιωμάτων του Ανθρώπου*[1] και την αντίστοιχη ευρωπαϊκή. Ωστόσο η πραγματική προστασία, δηλαδή ένας χάρτινος φάκελος, παρέχει μικρή αντίσταση σε πιθανούς εισβολείς και ίχνη λαθροχειρίας στην περίπτωση που ο αντίπαλος είναι σχετικά αδέξιος. Επίσης, ο επίδοξος λαθραναγνώστης συνήθως δεν είναι σε θέση να αναχαιτίσει τη μεταφορά της επιστολής καθ' οδόν (όσο κι αν ληστείες ταχυδρομικής άμαξας έχουν καταγραφεί στην ιστορία, τη λογοτεχνία και τον κινηματογράφο). Μπορεί όμως να παρέμβει στα σημεία εκκίνησης και προορισμού (πολύ εύκολα αν είναι συγκάτοικος με τον παραλήπτη στην ίδια πολυκατοικία).

Η ταχυδρομική υπηρεσία έχει ιδιαίτερη σημασία για την ιδιωτικότητα επειδή είναι από τις πρώτες υπηρεσίες που έχουν ιστορικά δημιουργήσει τη σχετική έννοια στον ανθρώπινο νου. Έχουν επίσης δημιουργήσει επίσημα μέτρα αναστολής της ιδιωτικότητας, δεδομένου ότι οι φάκελοι ανοίγονται και οι επιστολές λογοκρίνονται από κρατικές υπηρεσίες σε περιόδους πολέμου ή αυξημένης καταπίεσης ή εφόσον οι επιστολές αφορούν σε ευαίσθητες κατηγορίες κρατικών υπαλλήλων.

Οι αντιλήψεις περί ιδιωτικότητας στην παροχή ταχυδρομικών υπηρεσιών κληρονομήθηκαν περαιτέρω από τις τηλεπικοινωνιακές υπηρεσίες, από την ηλεκτρονική αλληλογραφία κ.λπ.

Στο τηλεφωνικό δίκτυο

Μέχρι και τη δεκαετία του '70, ή και αργότερα κατά τόπους, στην ελληνική επαρχία υπήρχε ένα «τηλεφωνείο» σε κάθε χωριό, όπου στεγαζόταν μια τηλεφωνική συσκευή ή (εφόσον υπήρχε ανάγκη για περισσότερες) ένας πίνακας μεταγωγής για τη χειροκίνητη σύνδεση με το «κέντρο». Πάνω στον πίνακα συνδεόταν μια τηλεφωνική

[1] «Κανείς δεν επιτρέπεται να υποστεί αυθαίρετες επεμβάσεις στην ιδιωτική του ζωή, την οικογένεια, την κατοικία ή την αλληλογραφία του, ούτε προσβολές της τιμής και της υπόληψης του.» Βλ. `http://www.ohchr.org/EN/UDHR`.

συσκευή, ίσως μερικές ακόμη εφόσον υπήρχε ιατρείο ή τοπικός σταθμός χωροφυλακής. Ο τηλεφωνητής ή η τηλεφωνήτρια που χειριζόταν τις κλήσεις άκουγε φανερά ή κρυφά τις συνδιαλέξεις, ενώ την ίδια δυνατότητα είχε και η χωροφυλακή χωρίς πολλές διαδικασίες. Στις πόλεις αντίστοιχα πολλοί χρησιμοποιούσαν το τηλέφωνο στο κοντινό περίπτερο ή εκείνο σε κάποιο κοντινό σπίτι ή διαμέρισμα. Σε όλες αυτές τις περιπτώσεις ο χρήστης του τηλεφώνου δε μπορούσε να έχει ιδιαίτερες αξιώσεις ιδιωτικότητας εκ των πραγμάτων. Παρ' όλα αυτά καθένας ήθελε και διεκδικούσε να μένει μόνος όταν έπρεπε να συζητήσει ευαίσθητα θέματα.

Η ιδιωτικότητα στην τηλεφωνία, ως δυνατότητα και ως έννοια εμπεδωμένη από μέρος του πληθυσμού, ήρθε με τη μαζική εγκατάσταση τηλεφωνικών συσκευών στα σπίτια και ακόμη περισσότερο με την κινητή τηλεφωνία.

Αν η τηλεφωνία στις ΗΠΑ άρχισε το 1876, παρακολουθήσεις με συνακρόαση καταγράφηκαν ήδη από το 1890, οπότε και κηρύχθηκε η σχετική δραστηριότητα παράνομη. Εδώ και παραπάνω από ένα αιώνα η απλή τεχνικά και δημοφιλής μέθοδος στηρίζεται σε συνακρόαση από οποιοδήποτε βολικό σημείο (στην ίδια την τηλεφωνική συσκευή, στον κοντινό κατανεμητή καλωδίων έξω απ' το σπίτι ή στην είσοδο της πολυκατοικίας, στον υπαίθριο κατανεμητή, στο τοπικό τηλεφωνικό κέντρο). Το σήμα καταλήγει είτε απ' ευθείας στα ακουστικά είτε (συχνότερα) καταγράφεται σε μαγνητόφωνο. Δραστηριότητες τέτοιου είδους φαίνονται στην κινηματογραφική ταινία *Οι Ζωές των Άλλων* (*Das Leben der Anderen*, 2006), που εικονογραφεί μια ιστορία παρακολουθήσεων από το 1984.

Με δυο λόγια το κλασσικό τηλεφωνικό δίκτυο που βασίζεται στην μεταφορά αναλογικού σήματος φωνής είναι εκ κατασκευής ευάλωτο από συνακροάσεις. Το βασικό ενδιαφέρον περί την ιδιωτικότητα παραμένει κυρίως στο απόρρητο της ίδιας της συνομιλίας. Η επίσης κλασσική απάντηση συνίσταται στην κρυπτογράφηση, η οποία μπορεί να γίνει μεταξύ των δύο άκρων, αλλά προφανώς προϋποθέτει ειδικό εξοπλισμό, ανταλλαγή κλειδιών κ.λπ.

Επιπρόσθετη ασφάλεια και ιδιωτικότητα στην επικοινωνία μέσω του δημόσιου τηλεφωνικού δικτύου μπορεί να εξασφαλισθεί μέσω ενός *περιπλέκτη* (*scrambler*). Οι πρώτοι περιπλέκτες εμφανίσθηκαν πριν τον Β' παγκόσμιο πόλεμο και βασίζονταν στη μίξη του επιθυμητού σήματος φωνής στην πλευρά του πομπού με ένα σήμα θορύβου, το οποίο έπρεπε να αφαιρεθεί στην πλευρά του δέκτη. Για τον σκοπό αυτόν κατασκευάζονταν φωνογραφικοί δίσκοι σε ζευγάρια με το ίδιο σήμα θορύβου και διανέμονταν στον πομπό και τον αντίστοιχο δέκτη. Οι νεότεροι περιπλέκτες βασίζονται σε ψηφιοποίηση του σήματος (εφόσον είναι αναλογικό) και κρυπτογράφηση με κλειδιά των οποίων η ανταλλαγή γίνεται με μεθόδους δημόσιου κλειδιού.

Το σημερινό δημόσιο τηλεφωνικό δίκτυο (public switched telephone network (PSTN)) θεωρείται «ασφαλές» εξ αιτίας της δυσκολίας υποκλοπής πληροφοριών σηματοδοσίας, της δυσκολίας πρόσβασης στις φυσικές του εγκαταστάσεις, του γεγονότος ότι τα πρωτόκολλά του δεν ερευνώνται τόσο όσο αυτά του Internet, και της νομικής του προστασίας [GSB05]. Επί πλέον στο τηλεφωνικό δίκτυο η ταυτότητα του καλούντος μεταφέρεται στον καλούμενο από τον ίδιο τον πάροχο, ο οποίος δρα ως έμπιστος ενδιάμεσος.

Μεταδεδομένα Επιθέσεις μπορούν να γίνουν και στα μεταδεδομένα (metadata) των συνομιλιών, δηλαδή μπορεί κάποιος να προσπαθήσει να αποκτήσει τις *καταγραφές λεπτομερειών κλήσης* (*call data record, CDR*). Τα CDRs δεν έχουν τον πλούτο της πληροφορίας που περιέχει μια συνομιλία, είναι όμως αποτελεσματικά στο να διαπιστωθεί πότε και ποιος μίλησε με ποιόν. Αποτελούν ψηφιοποιημένη πληροφορία έτοιμη για

εξόρυξη στοιχείων και περαιτέρω σχηματισμό ερμηνειών και μοντέλων συμπεριφοράς. Οι κρατικές υπηρεσίες μεγάλων δυνάμεων έχουν επανειλημμένα κατηγορηθεί ότι κάνουν μαζική συλλογή και επεξεργασία τέτοιων στοιχείων, συχνά εκτός νομιμότητας. Εξ άλλου μαζική παρακολούθηση σπάνια μπορεί να επιτευχθεί με τήρηση όλων των διαδικασιών, π.χ. με έκδοση εισαγγελικού εντάλματος.

Γενικότερα οποιαδήποτε επίθεση στην ασφάλεια ενός τηλεφωνικού δικτύου μπορεί να έχει εν τέλει ως αποτέλεσμα την απόκτηση πληροφοριών που οδηγούν σε προσβολές ιδιωτικότητας.

Πληροφορίες καταλόγου Στόχους επιθέσεων αποτελούν επίσης πληροφορίες του τύπου που υπάρχουν στον τηλεφωνικό κατάλογο, δηλαδή κατά πόσο ένας αριθμός τηλεφώνου είναι ενεργός, αν ανήκει σε ιδιώτη, σε οργανισμό ή σε εταιρεία, αν χρησιμοποιείται για τηλεφωνική επικοινωνία ή για άλλη υπηρεσία (fax, φωνητικό ταχυδρομείο κ.λπ.). Ορισμένες από τις πληροφορίες αυτές είναι διαθέσιμες σε όλους, ενώ άλλες σε περιορισμένο κοινό, π.χ. στο εσωτερικό μιας εταιρίας. Η διαρροή τους έξω από τον προκαθορισμένο κύκλο διάδοσής τους μπορεί να προκαλέσει ενοχλήσεις που εκλαμβάνονται και ως προσβολές ιδιωτικότητας. Για παράδειγμα αν μια τηλεφωνική εταιρεία σας κάνει συχνές κλήσεις σε αριθμό που έχετε αιτηθεί να μείνει «απόρρητος» (δηλαδή να μην αναγράφεται στον τηλεφωνικό κατάλογο), είναι πιθανό να αισθανθείτε ότι προσβάλλεται η ιδιωτικότητά σας.

Οι πληροφορίες του παραπάνω τύπου θα μπορούσαν να αποκτηθούν με επιθέσεις σε βάσεις δεδομένων, σε υπηρεσίες καταλόγου ή και με κοινωνική μηχανική. Μπορούν όμως να αποκτηθούν και απ' ευθείας μέσα από το τηλεφωνικό δίκτυο. Το war-dialing είναι μια τέτοια τεχνική. Συνίσταται στην σάρωση μιας λίστας αριθμών τηλεφώνου (με κλήση καθενός στη σειρά) προκειμένου να διαπιστωθεί ο κάτοχός τους, η χρήση τους, μηχανές που βρίσκονται πίσω τους κ.ο.κ. Η τεχνική αυτή μπορεί να θεωρηθεί σήμερα ξεπερασμένη εξ αιτίας της μειωμένης χρήσης του τηλεφωνικού δικτύου για σύνδεση υπολογιστών, αλλά σε ορισμένες περιπτώσεις είναι ένας εύκολος τρόπος παράκαμψης των κλασσικών συστημάτων άμυνας (π.χ. των firewalls) ώστε να πραγματοποιηθούν επιθέσεις σε εταιρικά συστήματα [MSK12]. Στην αγορά διατίθενται εξειδικευμένες συσκευές και κατάλληλο λογισμικό, αλλά το ελάχιστο σύνολο απαραίτητων εργαλείων είναι μόνο ένας υπολογιστής που τρέχει DoS κι ένα modem.

Μια άλλη επίθεση συνίσταται στην απόκτηση πρόσβασης σε εταιρικά τηλεφωνικά κέντρα (PBX) μέσω τηλεφωνικού δικτύου με χρήση modem. Η τηλεφωνική σύνδεση παραμένει δημοφιλές μέσο για τη συντήρηση και τη διαχείριση ενός κέντρου. Η ορθή διαδικασία περιλαμβάνει την ενεργοποίηση ενός modem από την πλευρά του κέντρου μόνον κατά τη διάρκεια μιας σαφώς δηλωμένης δραστηριότητας συντήρησης. Ωστόσο σε πολλές περιπτώσεις το modem αυτό παραμένει μονίμως ανοιχτό και μπορεί να ανιχνευθεί π.χ. μέσω μιας σειράς κλήσεων, δηλαδή της τεχνικής που είναι γνωστή ως war-dialing [MSK12].

Στην κινητή τηλεφωνία

Η ψηφιακή κινητή τηλεφωνία με το σύστημα GSM από το 1991 και μετά έβαλε πρόσθετες δυσκολίες στην πλευρά της παρακολούθησης κρυπτογραφώντας το περιεχόμενο της συνδιάλεξης στην ασύρματη ζεύξη. Η αρχική ιδέα ήταν να δημιουργηθεί μια ασφαλής και ουσιαστικά απόρθητη μετάδοση, μέχρι τη στιγμή που μπήκαν στη διαπραγμάτευση οι κρατικές υπηρεσίες, που απαίτησαν τη μείωση της δυσκολίας αποκρυπτογράφησης με σκοπό τις παρακολουθήσεις. Η κινητικότητα του «στόχου» ήταν

μια επιπρόσθετη δυσκολία, που σήμαινε ότι όλος ο εξοπλισμός έπρεπε να μεταφέρεται εύκολα και κατέληξε στο λεγόμενο «βαλιτσάκι».

Η απάντηση της πλευράς των νομίμως ή σχεδόν νομίμως παρακολουθούντων ήρθε με την ανάπτυξη συστημάτων συνακρόασης μέσα από το ίδιο το τηλεφωνικό κέντρο, όπου πλέον δεν υφίσταται και ανάγκη αποκρυπτογράφησης. Η χρήση ενός τέτοιου συστήματος από τις αρχές προϋποθέτει συνήθως την ύπαρξη εισαγγελικής εντολής, η οποία επιτρέπει την ενεργοποίηση της δυνατότητας συνακρόασης μέσα από ένα σύστημα ελέγχου. Η παράνομη εκμετάλλευση του τελευταίου από μυστικές υπηρεσίες είναι εκείνη που δημιούργησε τις λεγόμενες «ολυμπιακές παρακολουθήσεις» του 2004-5 στην Ελλάδα [PS07].

Σε δίκτυα VoIP

Η *μετάδοση φωνής μέσω IP* (Voice over IP) είναι μια τεχνολογία μετάδοσης φωνής μέσω του Διαδικτύου που υποκαθιστά όλο και περισσότερο την κλασσική τηλεφωνία. Οι παρεχόμενες υπηρεσίες αφορούν σε υπηρεσίες που μέχρι πρότινος χρησιμοποιούσαν το τηλεφωνικό δίκτυο (PSTN), δηλαδή τηλεφωνία, fax, SMS και φωνητικά μηνύματα. Οι υλοποιήσεις ποικίλλουν ανάλογα με τον πάροχο της υπηρεσίας. Σε κάθε περίπτωση η φωνή μεταδίδεται ως ψηφιοποιημένο σήμα, το οποίο παράγεται με κατάλληλους κωδικοαποκωδικευτές (codecs) που προσπαθούν να βελτιστοποιήσουν τη μετάδοση δεδομένου του διαθέσιμου εύρους ζώνης και των απαιτήσεων της υπηρεσίας. Η διαχείριση της «κλήσης» εξασφαλίζεται με τη χρήση των πρωτοκόλλων του στρώματος συνόδου. Τα δυο πιο συνηθισμένα πρωτόκολλα είναι το H.323 και το Session Initiation Protocol (SIP).

Ένα σύστημα VoIP προσφέρει στην πλευρά του χρήστη ένα πλήθος από πρωτόκολλα και διεπαφές, που μπορούν να αξιοποιηθούν από επιτιθέμενους. Για τους τελευταίους, και για όσους θέλουν να διαπιστώσουν τις τρωτότητες του συστήματός τους, υπάρχει μια σειρά από εξειδικευμένα εργαλεία. Το SiVuS για παράδειγμα σαρώνει συστήματα SIP, όπως επίσης και το SIPVicious [MSK12]. Ο εισβολέας μπορεί να επιχειρήσει να ανακτήσει το αρχείο των ρυθμίσεων, το οποίο διαβάζει μια τηλεφωνική συσκευή SIP κατά την εκκίνησή της και το κατεβάζει από ένα TFTP (Trivial File Transfer Protocol) server. Το αρχείο αυτό είναι διαθέσιμο στο server χωρίς προστασία, αρκεί να ξέρει κανείς το όνομά του, το οποίο μπορεί να βρεθεί ψάχνοντας ανάμεσα σε πιθανά ονόματα, όπως για παράδειγμα `SIPDefault.cnf`. Τις ίδιες πληροφορίες μπορεί κανείς να αντλήσει απ᾽ ευθείας από την τηλεφωνική συσκευή. Μια προφανής άμυνα είναι να γίνεται έλεγχος IP από τον server πριν δώσει το αρχείο ρυθμίσεων, αλλά ο επιτιθέμενος και πάλι μπορεί να χρησιμοποιήσει αλλοιωμένη διεύθυνση.

Η *επίθεση απαρίθμησης* (enumeration) έχει σκοπό να αποσπάσει από ένα server τα ονόματα έγκυρων χρηστών του συστήματος VoIP, ακριβώς όπως κάποιος θα ήθελε να μάθει αριθμούς τηλεφώνου που όντως χρησιμοποιούνται. Η επίθεση μπορεί να υλοποιηθεί αποστέλλοντας διαδοχικά αιτήματα εγγραφής για πιθανά ονόματα χρηστών (`REGISTER`) σε μια πύλη Asterisk SIP. Ο server κάθε φορά ανταποκρίνεται διαφορετικά στα έγκυρα ονόματα χωρίς αυθεντικοποίηση και διαφορετικά στα ανύπαρκτα ονόματα. Οι διαδικασίες αυτές μπορούν να γίνουν αυτόματα από τα υπάρχοντα εργαλεία σάρωσης και είναι δύσκολο να υπάρξει άμυνα επειδή ο server είναι υποχρεωμένος να δίνει αποκρίσεις που είναι χρήσιμες στην κανονική λειτουργία του συστήματος.

Υποκλοπή μπορεί να γίνει με απ᾽ ευθείας απόπειρα αναχαίτισης της ροής του RTP (Real-time Transport Protocol), δηλαδή του πρωτοκόλλου που είναι υπεύθυνο για τη μεταφορά του ήχου και της εικόνας μέσα από τα δίκτυα VoIP. Ένας τρόπος να πραγματοποιηθεί η υποκλοπή είναι μέσω της μεθόδου man-in-the-middle και εξαπάτησης

μέσω ARP (Address Resolution Protocol). Μια τέτοια επίθεση περιγράφεται λεπτομερώς στο Κεφ. 6 στο [MSK12].

Τέλος, αν ο επιτιθέμενος θέλει απλώς να παρενοχλήσει το θύμα και να του αφαιρέσει τη δυνατότητα να κάνει κλήσεις μπορεί να αρκεστεί σε μια επίθεση κίνησης, π.χ. στέλνοντας μεγάλο αριθμό από πλαστές κλήσεις (`SIP INVITE`). Μάλιστα αν το θύμα είναι τηλεφωνική συσκευή, αυτή θα χτυπάει συνεχώς. Περιττό να ειπωθεί ότι και γι᾽ αυτόν τον τύπο επίθεσης υπάρχουν έτοιμα εργαλεία.

13.7 Η ιδιωτικότητα στο Internet

Το Internet είναι ένα όχημα που χρησιμοποιείται για την παροχή ποικίλων υπηρεσιών. Το κοινό γνώρισμα αυτών των υπηρεσιών είναι μόνο ότι έχουν ανάγκη επικοινωνίας και βασίζονται σ᾽ αυτήν περισσότερο ή λιγότερο. Ωστόσο η ιδιότητα του Internet να αγκαλιάζει όλον τον πλανήτη και να διακινεί τεράστιο όγκο πληροφορίας με πρωτόγνωρες ταχύτητες προσθέτει νέες δυνατότητες και νέα προβλήματα, μεταξύ των οποίων και προβλήματα ιδιωτικότητας, άλλα από τα οποία είναι νέα και άλλα παλιά, αλλά που εμφανίζονται πιο οξυμένα.

Μια εξαντλητική ανάλυση της επίδρασης του Internet σε θέματα ιδιωτικότητας με ευρεία κάλυψη του φάσματος των εφαρμογών που φιλοξενεί θα ήταν εξαιρετικά φιλόδοξος στόχος. Στη συνέχεια παραθέτουμε μερικές χαρακτηριστικές περιπτώσεις που έχουν προσελκύσει αυξημένο ενδιαφέρον σχετικά με την ιδιωτικότητα.

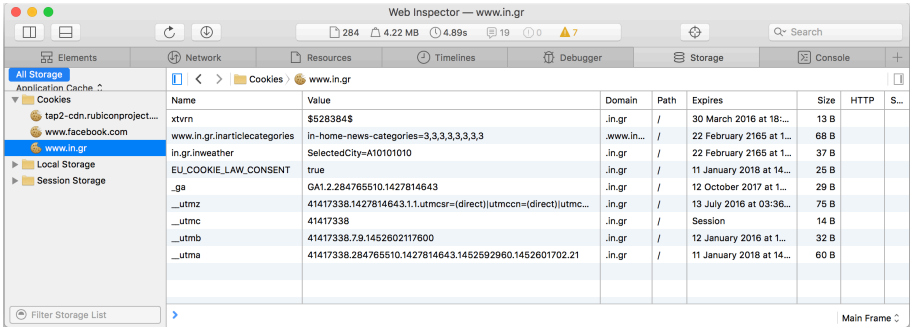
Το ηλεκτρονικό ταχυδρομείο

Το ηλεκτρονικό ταχυδρομείο είναι μια από τις παλιότερες υπηρεσίες που προσφέρονται μέσα από δίκτυα υπολογιστών.

Η βασική του λειτουργία είναι ανταλλαγή κειμένων ανάμεσα σε δυο πλευρές. Με αυτή την έννοια βρίσκεται πολύ κοντά στην ταχυδρομική υπηρεσία ή και στην τηλεφωνική, οπότε το βασικό ζήτημα ιδιωτικότητας δεν είναι παρά το απόρρητο της επικοινωνίας. Οι περισσότεροι από μας θεωρούμε ότι το εν λόγω απόρρητο δεν είναι ιδιαίτερα κρίσιμο για τις συνηθισμένες μας επικοινωνίες με την ίδια λογική που χρησιμοποιούμε μικρής ασφάλειας φακέλους όταν ανταλλάσουμε γράμματα ταχυδρομικά. Ας σημειωθεί ότι στην ταχυδρομική υπηρεσία η ευαισθησία είναι μειωμένη ακόμη και για πιο ευαίσθητες πληροφορίες. Για παράδειγμα, ως πολύ πρόσφατα οι φορολογικές δηλώσεις μπορούσαν να ταχυδρομηθούν. Παρ᾽ όλα αυτά όσοι συμπολίτες μας επιθυμούν να διασφαλίσουν τις επικοινωνίες τους μέσω email μπορούν να χρησιμοποιήσουν κρυπτογραφημένες εκδοχές αυτής της υπηρεσίας και αρκετοί το κάνουν ήδη.

Cookies

Το πρωτόκολλο HTTP δεν διατηρεί πληροφορίες κατάστασης μετά το τέλος μιας σύνδεσης. Είναι ευθύνη των δύο πλευρών που επικοινωνούν (clint-server) να λάβουν τα μέτρα τους, δηλαδή να αποθηκεύουν μέσα από τις αντίστοιχες εφαρμογές τις απαραίτητες πληροφορίες κατάστασης. Συνήθως αυτό είναι κάτι που απασχολεί περισσότερο την πλευρά του εξυπηρετητή, δηλαδή την πλευρά του παρόχου μια υπηρεσίας, τον οποίο φυσικά κανείς δεν μπορεί να εμποδίσει να αποθηκεύσει όποια πληροφορία επιθυμεί (νόμιμα ή παράνομα). Για παράδειγμα, ένα ηλεκτρονικό κατάστημα είναι πιθανό να συγκρατήσει το όνομα, τη διεύθυνση και ίσως την πιστωτική κάρτα ενός πελάτη. Μπορεί ακόμη να συγκρατήσει μια ιστορία των ως τώρα συναλλαγών, με σκοπό



Σχήμα 13.1: Cookies για την ιστοσελίδα `www.in.gr` μέσα από τον Web Inspector του φυλλομετρητή Safari.

να διαπιστώσει τις προτιμήσεις, τις συνήθειες ή και τη φερεγγυότητα του πελάτη και να του στείλει ενδεχομένως στο μέλλον προσφορές.

Ωστόσο οι προγραμματιστές του φυλλομετρητή Netscape είχαν περί τα μέσα της δεκαετίας του '90 την φαεινή ιδέα πως θα μπορούσε να χρησιμοποιείται ο υπολογιστής του πελάτη για την αποθήκευση βασικών πληροφοριών που περιγράφουν την κατάσταση μιας αλληλεπίδρασης [Ber01a; Kri01]. Οι πληροφορίες αυτές αποθηκεύονται στα λεγόμενα *cookies*. Η μέθοδος των cookies όχι μόνο απαλλάσει την πλευρά του παρόχου υπηρεσίας από κόστος αποθήκευσης, αλλά διευκολύνει και την παροχή στοιχείων του πελάτη προς τρίτους, π.χ. εμπορικούς τους εταίρους που θα μπορούσαν να αξιοποιήσουν τα ίδια δεδομένα.

Τα cookies είναι αρχεία με κατάλληλα κωδικοποιημένα στοιχεία που αποθηκεύονται στον υπολογιστή ενός χρήστη όταν αυτός ανοίξει ένα φυλλομετρητή (browser). Περιέχουν επιλεγμένα δεδομένα σχετικά με την αλληλεπίδραση των δύο πλευρών (π.χ. στοιχεία του χρήστη και των προτιμήσεών του) με σκοπό τη χρήση τους είτε στη διάρκειά της είτε σε μελλοντικές επισκέψεις της ίδιας ιστοσελίδας ή σχετικών ιστοσελίδων. Τα cookies δημιουργούνται και διακινούνται μέσω του πρωτοκόλλου HTTP. Ο εξυπηρετητής μέσω του πρωτοκόλλου HTTP και με μια εντολή `Set-Cookie` μπορεί να δημιουργήσει ένα νέο cookie στον υπολογιστή του πελάτη.Την επόμενη φορά που ο ίδιος πελάτης (client) θα επικοινωνήσει με τον ίδιο εξυπηρετητή θα του στείλει το ήδη αποθηκευμένο cookie.

Στο παράδειγμα της Εικόνας 13.1 φαίνονται cookies (μέσα από τον Web Inspector του φυλλομετρητή Safari) που έχουν αποθηκευθεί εξ αιτίας του γεγονότος ότι ο φυλλομετρητής έχει επισκεφθεί την ιστοσελίδα `www.in.gr`. Τα cookies στην προκειμένη περίπτωση αποθηκεύονται όλα μαζί σε ένα αρχείο[2] σε κωδικοποιημένη μορφή, αλλά μέσω ενός viewer όπως ο Web Inspector μπορεί ο χρήστης να τα δει σε συνοπτική μορφή αναγνωρίσιμη από τον άνθρωπο και να τα σβήσει, εάν το επιθυμεί.

Τα cookies μπορούν να ταξινομηθούν στις εξής κατηγορίες [QQ10]:

Μόνιμα Πρόκειται για την παραδοσιακή κατηγορία, που αποσκοπεί στην αποθήκευση βασικών πληροφοριών για το χρήστη. Η ημερομηνία λήξης τους μπορεί να αντιστοιχεί σε μέρες, μήνες ή χρόνια. Ωστόσο συχνά αποθηκεύονται στα cookies ευαίσθητες πληροφορίες για τον χρήστη (περισσότερο ή λιγότερο επικίνδυνες, όπως π.χ. αριθμοί πιστωτικών καρτών) ή πληροφορίες που συνάγονται από τη

[2] Στο λειτουργικό Mac OSX αποθηκεύονται στο αρχείο `~/Library/Cookies/Cookies.binarycookies`.

συμπεριφορά του (δεν έχουν δηλαδή δηλωθεί ευθέως από αυτόν και πολύ περισσότερο δεν έχουν τη συγκατάθεσή του).

Προσωρινά cookies (ή cookies συνόδου - session cookies), των οποίων η ζωή διαρκεί όσο διαρκεί μια σύνοδος ανάμεσα στις δύο πλευρές.

Cookies πρώτης πλευράς (first party cookies) είναι εκείνα που χρησιμοποιεί η πλευρά του εξυπηρετητή για να αποθηκεύσει πληροφορίες σχετικές με τον χρήστη, π.χ. πώς να του δείξει κατάλληλα μια ιστοσελίδα, τι διαφημίσεις να βάλει κ.ο.κ.

Cookies για τρίτους (third party cookies), τα οποία προορίζονται για χρήση από τρίτους, π.χ. από μια διαφημιστική ιστοσελίδα, η οποία θα πληροφορηθεί έτσι τις προτιμήσεις του πελάτη ώστε να δώσει στοχευμένες διαφημίσεις.

Flash cookies που αφορούν σε πληροφορίες σχετικές με ψηφιακό περιεχόμενο (ακίνητη και κινούμενη εικόνα, ήχο κ.λπ.) και αποθηκεύονται για λογαριασμό του Adobe Flash Player.

Cookies και ιδιωτικότητα

Οι πληροφορίες που μπορούν να συγκρατηθούν μέσα σε ένα cookie είναι επί της ουσίας οτιδήποτε μπορεί να ενδιαφέρει την πλευρά του παρόχου μιας υπηρεσίας. Ορισμένες από αυτές είναι χρήσιμες και μπορούν ξεκάθαρα να δικαιολογηθούν από τις ανάγκες της καλής ποιότητας παροχής της υπηρεσίας. Τέτοιες περιπτώσεις είναι cookies που εξυπηρετούν την αυθεντικοποίηση του πελάτη, cookies που συγκρατούν τη γλώσσα προτίμησης του πελάτη, flash cookies που θυμούνται τις προτιμήσεις του πελάτη όταν είδε ένα video, cookies που κάνουν καταμέτρηση αποτυχημένων προσπαθειών εισόδου σε ένα λογαριασμό και βοηθούν στην ασφάλεια του πελάτη.

Μια άλλη κατηγορία όμως αποσκοπεί στη διαδικτυακή προώθηση προϊόντων μέσω της συγκέντρωσης στοιχείων για τις προτιμήσεις του πελάτη (profiling). Ο σκοπός τέτοιων cookies μπορεί να καλύπτει ένα φάσμα από τη γενικότερη ανάλυση της αγοράς (για τη βελτίωση προϊόντων και καλύτερη προώθηση) μέχρι τη στοχευμένη προώθηση στον συγκεκριμένο πελάτη.[3]

Προφανώς ο πάροχος μιας υπηρεσίας που έχει ήδη επιλεγεί από ένα πελάτη βρίσκεται σε προνομιακή θέση ως προς τον πελάτη ως προς το να κάνει εκτιμήσεις για τις προτιμήσεις του σε σύγκριση με άλλους παρόχους ανάλογων υπηρεσιών και μπορεί να τις χρησιμοποιήσει για να διατηρήσει και για να βελτιώσει περαιτέρω τις συναλλαγές του με το συγκεκριμένο πελάτη. Σε ορισμένες όμως περιπτώσεις είναι πιθανό να σκεφτεί ότι θα μπορούσε να αποκομίσει κέρδος μέσω της ανταλλαγής τέτοιων πληροφοριών με άλλους παρόχους, ή και απλώς από την αγοραπωλησία των πληροφοριών. Με τον τρόπο αυτόν μπορεί να αναπτυχθεί μια βιομηχανία επεξεργασίας και ανταλλαγής στοιχείων των πελατών πίσω από την πλάτη των τελευταίων. Η σχετική με τα cookies προσπάθεια προς αυτήν την κατεύθυνση είναι τα λεγόμενα *cookies τρίτων* (third party cookies), τα οποία είναι διαθέσιμα όχι μόνο προς τον πάροχο που τα δημιούργησε, αλλά και προς άλλους ενδιαφερόμενους. Τα cookies αυτού του είδους εί-

[3] Διδακτική για τις ακρότητες στις οποίες μπορεί να φτάσουν αυτές οι μέθοδοι είναι μια ιστορία από την προ του Internet εποχή, την οποία είχε διηγηθεί γιατρός με περίεργα ωράρια, λόγω των εφημεριών. Ο εν λόγω γιατρός είχε εγκατασταθεί σε αμερικανική πόλη. Η τηλεφωνική εταιρία, στην οποία ήταν συνδρομητής, ήταν προφανώς σε θέση να γνωρίζει ποιες ώρες ήταν στο σπίτι εξ αιτίας των κλήσεων που έκανε. Τις ίδιες όμως ώρες άρχισε να παίρνει κλήσεις από διαφημιστές ποικίλων προϊόντων, ακόμη και από άλλες τηλεφωνικές εταιρίες. Προφανώς η διακίνηση «προσωπικών» πληροφοριών είχε λάβει ήδη διαστάσεις.

ναι εκείνα που θεωρούνται πιο προκλητικά και απετέλεσαν βασικό ζήτημα στη διεθνή διαδικασία τυποποίησης.

Σήμερα μια σειρά προϊόντων λογισμικού εντοπίζει και διαγράφει ανεπιθύμητα cookies. Η διάδοση τέτοιων μηχανισμών και η διαγραφή σημαντικού ποσοστού τέτοιων cookies θα αποτελέσει καλό κίνητρο για την εξ αρχής αποφυγή της δημιουργίας τους, δεδομένου ότι η πλευρά που τα δημιουργεί τα χάνει τότε ολωσδιόλου.

Μια άλλη ενδιαφέρουσα άποψη που προσπαθεί να εφαρμόσει έλεγχο στην πηγή, δηλαδή στη δημιουργία των cookies, συνίσταται στη χρήση «πιστοποιημένων cookies», των οποίων την καλή χρήση θα εγγυάται κάποια έμπιστη αρχή πιστοποίησης. Στην ουσία εδώ πρόκειται για ένα είδος αυτορρύθμισης, δηλαδή για χρήση ενός μηχανισμού καλής φήμης, που δίνει μια σφραγίδα ποιότητας σε όσους κάνουν καλή χρήση.

Νομικά ζητήματα

Η χρήση των cookies εγείρει μια σειρά από νομικά ζητήματα. Μολονότι η γενικότερη κάλυψη νομικών θεμάτων που σχετίζονται με την ασφάλεια και την ιδιωτικότητα δεν είναι ανάμεσα στους στόχους αυτού του κειμένου, εδώ θα κάνουμε μια σύντομη εξαίρεση.

Τα cookies δημιουργούν ζητήματα de facto χρήσης ενός υπολογιστή από κάποιον άλλον, που δεν είναι ιδιοκτήτης ή νόμιμος χρήστης του. Αυτό συμβαίνει όταν εγκαθίσταται από ένα εξυπηρετητή ένα cookie στον υπολογιστή της πλευράς του πελάτη, δηλαδή εκεί όπου είναι ανοιγμένος ένας φυλλομετρητής που «βλέπει» πληροφορίες παρεχόμενες από τον εξυπηρετητή.

Η χρήση από άλλη πλευρά κανονικά απαιτεί τη συναίνεση του νόμιμου χρήστη του υπολογιστή. Η συναίνεση δεν μπορεί να αποσπασθεί από τον τελευταίο χωρίς κάποια θετική του ενέργεια. Η συνήθης πρακτική παγκόσμια είναι να εγκαθίστανται τα cookies εφόσον το επιτρέπουν οι ρυθμίσεις του φυλλομετρητή. Δεδομένου ότι οι περισσότεροι φυλλομετρητές στις αρχικές του ρυθμίσεις επιτρέπουν αυτήν την εγκατάσταση, και οι χρήστες σπάνια μπαίνουν στον κόπο να τις αλλάξουν, τα cookies εγκαθίστανται χωρίς προειδοποίηση. Επομένως στην περίπτωση αυτή δεν υπάρχει θετική ενέργεια του χρήστη που θα μπορούσε να ερμηνευθεί ως συναίνεση. Διαφορετική θα ήταν η κατάσταση αν οι αρχικές ρυθμίσεις απαγόρευαν τα cookies, οπότε ο χρήστης θα ήταν υποχρεωμένος να κάνει ενέργειες που θα αλλάξουν τις ρυθμίσεις και θα τα κάνουν δεκτά. Βέβαια μια τέτοια πρακτική θα ήταν στατιστικά οδυνηρή για τις συνήθεις εμπορικές πρακτικές, επειδή η αλλαγή ρυθμίσεων προϋποθέτει κόπο (μια σειρά ενεργειών) και γνώση και θα κατέληγε στην κατά πλειοψηφία απόρριψη των cookies. Κατά συνέπεια οι φυλλομετρητές παραδίδονται στον πελάτη με ρυθμίσεις που επιτρέπουν τα cookies, και χρησιμοποιείται η μάλλον παρακινδυνευμένη υπόθεση ότι η μη αλλαγή τους από τον χρήστη δημιουργεί τις προϋποθέσεις συναίνεσης. Ειδικά στην Ευρωπαϊκή Ένωση η κατάσταση αυτή έχει κατά τι μεταβληθεί τα τελευταία χρόνια, όπως θα δούμε παρακάτω.

Η εγκατάσταση cookies σε ένα υπολογιστή από κάποιον που δεν είναι κάτοχος ή νόμιμος χρήστης του μπορεί να συνδεθεί με μια σειρά ζητημάτων που προβλέπονται από το νόμο. Τέτοια ζητήματα είναι [Opp06]:

- Κατά πόσο η ενέργεια αυτή αποτελεί εισβολή σε ξένο υπολογιστή (με την έννοια της καταπάτησης ξένης περιουσίας και με την έννοια που προβλέπουν νόμοι για την προστασία των υπολογιστών και των πληροφοριών που περιέχονται σ᾽ αυτούς).

- Όταν ο εξυπηρετητής ανήκει σε κρατική υπηρεσία, κατά πόσο αποτελεί παράνομη έρευνα (με την έννοια π.χ. της αυθαίρετης έρευνας της αστυνομίας σε ιδιωτικό χώρο χωρίς σχετικό ένταλμα από εισαγγελέα).
- Στην περίπτωση που κατά τα ανωτέρω στοιχειοθετούνται ποινικά αδικήματα, παραβιάσεις συνταγματικά κατοχυρωμένων δικαιωμάτων κ.λπ., δημιουργούνται δευτερευόντως ευθύνες για τους προμηθευτές του φυλλομετρητή, του οποίου η κατασκευή επιτρέπει την τέλεση των προηγούμενων αδικημάτων;

Φυσικά τέτοια ζητήματα εγείρονται και από άλλες κατηγορίες λογισμικού, όπως είναι οι ιοί και παντός είδους προγράμματα παρακολούθησης. Όμως οι κατηγορίες αυτές είναι εξ υπαρχής παράνομες, ενώ η δημιουργία cookies αποτελεί καθημερινή πρακτική εταιριών και οργανισμών που φιλοδοξεί να παραμένει μέσα στα όρια της νομιμότητας. Περαιτέρω, η χρήση των cookies, ακόμη και όταν αυτά είναι νόμιμα, δημιουργεί μια σειρά επιπρόσθετων κινδύνων, συνήθως χωρίς γνώση του χρήστη. Για παράδειγμα, η αποθήκευση του αριθμού μιας πιστωτικής κάρτας δημιουργεί τον κίνδυνο ανάγνωσής της από κακοποιούς.

Η αξία των πληροφοριών που αποκτώνται μέσω των cookies και οι δυνατότητες που ανοίγονται θέτουν μια σειρά ζητημάτων και μπορεί να σχετίζονται με μια αντίστοιχη σειρά αδικημάτων. Ειδικά οι κρατικές υπηρεσίες είναι σε θέση ισχύος απέναντι στον πολίτη, με την έννοια ότι το κράτος διαθέτει πληθώρα τέτοιων υπηρεσιών και έχει τη δυνατότητα του συνδυασμού των πληροφοριών που προσλαμβάνει από κάθε μια ξεχωριστά. Όταν πάλι εμπλέκονται ιδιωτικοί οργανισμοί μπορεί να εφαρμόζονται νόμοι για την υπεξαίρεση εμπορικών μυστικών και την καταπολέμηση της βιομηχανικής κατασκοπείας. Τα δικαστήρια που ασχολούνται με τέτοιες περιπτώσεις συχνά αποφασίζουν για τη σοβαρότητά τους χρησιμοποιώντας το κριτήριο των ζημιών που έχουν προκληθεί ή που θα μπορούσαν να προκληθούν.

Στην ΕΕ η χρήση των cookies διέπεται από την Οδηγία 2002/58/ΕΚ του Ευρωπαϊκού Κοινοβουλίου και του Συμβουλίου, της 12ης Ιουλίου 2002, σχετικά με την επεξεργασία των δεδομένων προσωπικού χαρακτήρα και την προστασία της ιδιωτικής ζωής στον τομέα των ηλεκτρονικών επικοινωνιών. Σύμφωνα με την οδηγία αυτή «απαιτείται η συγκατάθεση του χρήστη προτού αποθηκευθούν πληροφορίες (cookies) σε υπολογιστές ή συσκευές του ή προτού επιτευχθεί η πρόσβαση σε τέτοιες πληροφορίες. Θα πρέπει να παρέχονται ακριβείς και πλήρεις πληροφορίες στον χρήστη, μεταξύ άλλων, για τους σκοπούς της αποθήκευσης ή της πρόσβασης». Ανάλογες είναι η προβλέψεις της προσαρμοσμένης ελληνικής νομοθεσίας, όπως αυτές περιγράφονται στην παράγραφο 5 του άρθρου 4 του ν. 3471/2006, όπως αυτή τροποποιήθηκε από το ν. 4070/2012. Οι νόμοι και οδηγίες για τη χρήση των cookies μπορούν να αναζητηθούν στις ιστοσελίδες της Αρχής Προστασίας Δεδομένων Προσωπικού Χαρακτήρα.[4]

Από την υποχρέωση εξασφάλισης της συγκατάθεσης του χρήστη εξαιρούνται cookies που είναι τεχνικώς απαραίτητα για την παροχή μιας υπηρεσίας, ως εξής:

- Cookies που σχετίζονται με δεδομένα που εισάγει ο χρήστης, π.χ. κατά τη συμπλήρωση μιας φόρμας, και κρατούν όσο κρατάει μια σύνοδος.
- Cookies σχετικά με την αυθεντικοποίηση του χρήστη, που καταστρέφονται με τη λήξη της συνόδου.
- Cookies που σχετίζονται με την ασφάλεια και χρησιμοποιούνται για να αποφεύγονται παραβιάσεις της αυθεντικοποίησης. Διατηρούνται για περιορισμένο χρόνο.

[4] `http://www.dpa.gr`

- Cookies που είναι απαραίτητα για την αναπαραγωγή ήχου και εικόνας κατά τη διάρκεια μιας συνόδου.
- Cookies συνόδου με σκοπό την ορθή κατανομή του φορτίου στο δίκτυο.
- Cookies για την προσαρμογή της διεπαφής με το χρήστη στις προτιμήσεις του τελευταίου, π.χ. cookies που συγκρατούν τις προτιμήσεις γλώσσας ή γραμματοσειράς και διατηρούνται για μια σύνοδο ή λίγο παραπάνω.
- Cookies τρίτων που εγκαθίστανται μέσω plugins ιστοσελίδων κοινωνικών δικτύων και αφορούν στη διανομή περιεχομένου προς τα μέλη του δικτύου.

Η υποχρέωση να εξασφαλίζεται η συναίνεση του χρήστη πριν την εγκατάσταση cookies, όπως αυτή νοείται μέσα στην ΕΕ, δεν έχει εξαιρεθεί από την κριτική. Ορισμένα βασικά σχόλια εναντίον αυτής της πολιτικής είναι (α) το περιττό κόστος που δημιουργεί στην πλευρά του εξυπηρετητή, (β) η ενόχληση στην πλευρά του χρήστη για κάτι που (γ) ο τελευταίος θα αναγκασθεί ούτως ή άλλως να δεχθεί και (δ) δεν είναι σε θέση να καταλάβει ή δεν πρόκειται να μελετήσει, ενώ τελικά (ε) στατιστικά το ευεργετικό αποτέλεσμα για την ιδιωτικότητα είναι μηδαμινό.

Στις ΗΠΑ η αντίστοιχη κίνηση ενάντια στην ανεξέλεγκτη χρήση των cookies εκπροσωπείται κυρίως από την *Do Not Track Me Online Act* του 2011, δηλαδή ένα νομοσχέδιο σύμφωνα με το οποίο κάθε δημιουργός cookies θα έπρεπε να δημοσιοποιεί πληροφορίες σχετικές με το περιεχόμενό τους και με ποιους άλλους τα μοιράζεται. Μέχρι τώρα δεν έχει μετατραπεί σε νόμο. Προχωρημένοι περιορισμοί στη χρήση των cookies έχουν υιοθετηθεί μόνο τοπικά, π.χ. στην πολιτεία της Καλιφόρνιας.[5]

Κοινωνικά δίκτυα

Τα κοινωνικά δίκτυα βρίσκονται σε μια ιδιαίτερη σχέση με τα θέματα ιδιωτικότητας επειδή αποτελούν ένα εικονικό χώρο, μέσα στον οποίο τα μέλη τους εκούσια παραχωρούν πληροφορίες που εμπίπτουν στην ιδιωτική σφαίρα.

Μερικά από τα ζητήματα ιδιωτικότητας που εγείρονται στα κοινωνικά δίκτυα δεν είναι διαφορετικά από ανάλογα θέματα σε κλασσικά τηλεπικοινωνιακά δίκτυα, όπως το απόρρητο της επικοινωνίας. Ωστόσο εμφανίζονται και νέα ζητήματα που σχετίζονται με την ομαδικότητα της επικοινωνίας, με το γεγονός δηλαδή ότι ένα μήνυμα απευθύνεται σε μια ομάδα προσώπων, που πολλές φορές έχει όρια που δεν είναι σαφή στο δημιουργό του μηνύματος. Επίσης εμφανίζονται θέματα σχετικά με την προσπάθεια των παρόχων των κοινωνικών δικτύων να αποκτούν, να χρησιμοποιούν και να εμπορεύονται πληροφορίες σχετικές με τα μέλη των δικτύων.

Τα σημερινά (2016) κοινωνικά δίκτυα (Facebook, Google+, MySpace, Twitter, Instagram κ.λπ.) χρησιμοποιούν μεθόδους πραγματοποίησης κέρδους βασισμένες στη συλλογή πληροφοριών σχετικών με τα μέλη τους και στη διάθεση των πληροφοριών αυτών σε τρίτους. Αξιοσημείωτο είναι ότι αυτά συμβαίνουν παρά την ύπαρξη μιας σειράς από θεωρητικά πολύ αυστηρούς νόμους προστασίας του πολίτη από την αποθήκευση και επεξεργασία στοιχείων από τρίτους (ιδιαίτερα στην Ευρωπαϊκή Ένωση). Οι περιστασιακές ενέργειες των κατά τόπους αρχών προστασίας του απορρήτου των επικοινωνιών και της ιδιωτικότητας, που υπάρχουν λίγο ως πολύ σε όλες τις χώρες, σπάνια αποθαρρύνουν τέτοιες πρακτικές.

Ορισμένα κοινωνικά δίκτυα, όπως το Facebook, χρησιμοποιούν μια τακτική που περιλαμβάνει επίμονες και επαναλαμβανόμενες ερωτήσεις («σε ποια πόλη ζείτε;») που

[5] Για τις σχετικές απόπειρες βλ. `http://en.wikipedia.org/wiki/Do_Not_Track_legislation`.

υποτίθεται πως εξυπηρετούν το σκοπό της κοινωνικής δικτύωσης, σε συνδυασμό με «αμοιβές» και ψυχολογικούς εκβιασμούς («τόσο % από το προσωπικό σας προφίλ είναι τώρα ολοκληρωμένο»). Ένα άλλο παράδειγμα είναι ερωτήσεις του τύπου «ποιες ταινίες έχετε δει ως τώρα;» ή και «πώς τις βαθμολογείτε;» Ας σημειωθεί ότι ο συνδυασμός ταινιών που έχει δει κάποιος, και μάλιστα όταν αφαιρεθούν οι πιο δημοφιλείς εξ αυτών, είναι συνήθως μοναδικός για κάθε άτομο και αρκετός για να το ταυτοποιήσει. Ένα περίφημο παράδειγμα είναι αυτό προέκυψε από το Netflix Prize, όπου φάνηκε ότι 4 βαθμολογημένες ταινίες είναι αρκετές για να ταυτοποιήσουν οποιονδήποτε σε μια ανωνυμοποιημένη κατά τ᾽ άλλα συλλογή δεδομένων.[6] Αυτό σημαίνει ότι πληροφορίες τέτοιου είδους ακόμη και αν παραδοθούν σε τρίτους ανωνυμοποιημένες (βλ. πιο κάτω ενότητα 13.8), τα πρόσωπα μπορούν να επαναταυτοποιηθούν εύκολα.

Ο Bruce Schneier, διάσημος ειδικός και συγγραφέας στην περιοχή της ασφάλειας, αποδίδει την εκτεταμένη παρακολούθηση, στην οποία επιδίδονται σήμερα τα κοινωνικά δίκτυα, σε δύο λόγους [Sch16]. Ο ένας είναι η εξέλιξη του διαδικτύου ως κατά κύριο λόγο δωρεάν μέσου. Οι χρήστες του ήταν πάντοτε απρόθυμοι να πληρώνουν για υπηρεσίες που παίρνουν μέσω του διαδικτύου, γεγονός που οδήγησε σε ένα οικονομικό μοντέλο γνωστό ήδη από τη χρήση του στην τηλεόραση, δηλαδή στη χρηματοδότηση μέσω διαφημίσεων. Οι ιδιοκτήτες ιστοσελίδων χρεώνουν πιο ακριβά στοχευμένες διαφημίσεις και η στόχευση, δηλαδή ο εντοπισμός των πιο πιθανών αγοραστών ενός προϊόντος, μπορεί να βελτιωθεί μέσω της παρακολούθησης. Στα πλαίσια της βελτίωσης της στόχευσης η ιδιωτικότητα μειώνεται συνεχώς.[7] Ο άλλος λόγος είναι ότι η συμπεριφορά μας απέναντι στα «δωρεάν» (free) αγαθά δεν είναι και τόσο ορθολογική. Η κατανάλωση γίνεται μεγαλύτερη ακόμη και όταν δεν υπάρχει ανάγκη και η «δωρεάν τιμή» μπορεί εύκολα να ξεστρατίσει τη συνηθισμένη ανάλυση που κάνουμε όλοι μας για την τιμή εν σχέσει με το όφελος. Το τίμημα που πληρώνουμε εδώ είναι η ιδιωτικότητα και βάζοντάς την στη ζυγαριά απέναντι σε δωρεάν προϊόντα ή υπηρεσίες η ζυγαριά γέρνει υπερβολικά υπέρ της δεύτερης πλευράς (στα μάτια του συνηθισμένου χρήστη που δεν είναι σε θέση να εκτιμήσει τις πιθανές μακροπρόθεσμες ζημιές του). Στο μεταξύ οι εταιρείες που την εμπορεύονται κάνουν ό,τι μπορούν για να μας εμποδίσουν να καταλάβουμε πόση ιδιωτικότητα χάνουμε κάθε φορά και θα έπρεπε να τη βάλουμε στη ζυγαριά.

Η απώλεια ιδιωτικότητας είναι εγγενής στα κοινωνικά δίκτυα εξ αιτίας του διακηρυγμένου τους σκοπού, που είναι η κοινωνική δικτύωση, δηλαδή θα συνέβαινε ακόμη κι αν αυτά παραιτούνταν εντελώς από τα διαφημιστικά έσοδα κι από τη συλλογή στοιχείων που περιγράφτηκε παραπάνω. Για παράδειγμα σε ένα δίκτυο όπως το Facebook αν οι φίλοι του Χ είναι ορατοί μόνο από τους ίδιους τους φίλους του Χ, τότε ο Χ θα δυσκολευτεί πολύ να αποκτήσει νέους φίλους. Η πιθανότητα αύξησης του κύκλου των φίλων μεγιστοποιείται (για τους άλλους) αν ο κατάλογος φίλων είναι δημόσιος, ενώ μια μέση λύση είναι να είναι τουλάχιστον προσβάσιμος στους φίλους των φίλων. Ας ληφθεί εδώ υπόψη ότι ο μέσος βαθμός διαχωρισμού των μελών του Facebook ήταν τον

[6] Το Netflix Prize είναι ένας ετήσιος διαγωνισμός που έκανε η Netflix από το 2006 ως το 2009 για την δημιουργία του καλύτερου αλγορίθμου σύστασης ταινιών στους πελάτες της. Οι συστάσεις βασίζονται σε ένα συνδυασμό των βαθμολογιών για κάθε ταινία που έχουν δώσει μέχρι στιγμής οι πελάτες. Η Netflix δημοσιοποίησε ως training set μια ανωνυμοποιημένη εκδοχή της συλλογής βαθμολογιών με περίπου 100.000.000 βαθμολογίες, τις οποίες έδωσαν περίπου 500.000 θεατές για 18.000 ταινίες. Το 2007 δημοσιεύθηκε μια επιστημονική εργασία που εξηγεί πώς μπορεί να απο-ανωνυμοποιηθεί η συλλογή χρησιμοποιώντας 4 βαθμολογίες για κάθε πελάτη. Το 2009 κατατέθηκαν αγωγές εναντίον της εταιρείας από πελάτες που δυσαρεστήθηκαν επειδή έγιναν γνωστές οι βαθμολογίες τους.

[7] "... Internet companies can improve their product offerings to their actual customers by reducing user privacy. Facebook has done it systematically over the years, regularly updating its privacy policy ... Google has done much the same", Bruce Schneier [Sch16].

Φεβρουάριο του 2016 ίσος περίπου με 3,5 [Edu+16], πράγμα που σημαίνει ότι ένα μέλος του Facebook μπορεί να φτάσει σε οποιοδήποτε άλλο δια μέσου 3-4 άλλων μελών που είναι μεταξύ τους διαδοχικά συνδεδεμένα.

Στην καθημερινή πράξη τα μέλη των κοινωνικών δικτύων σπάνια τα απασχολεί πόσο μακριά φτάνει κάτι που δημοσιοποιούν και ακόμη σπανιότερα είναι σε θέση να αποκτήσουν ακριβή εικόνα για τη διάδοση αυτή. Ο μέσος χρήστης του κοινωνικού δικτύου το χρησιμοποιεί μέσα από την ησυχία και την άνεση του προσωπικού του χώρου ή της προσωπικής του συσκευής και θεωρεί ότι όσα θα πει ή θα γράψει, καθώς και οι φωτογραφίες ή άλλες πληροφορίες που θα ανεβάσει, φτάνουν σε κάποιο έμπιστο χώρο. Συνήθως θεωρεί ότι είναι κάτι ανάλογο με ένα ένα δωμάτιο με φίλους ή ένα καφέ, ένα εστιατόριο ή ένα μπαράκι. Σε όλους αυτούς τους χώρους όμως έχει τη δυνατότητα να ελέγξει ποιοι είναι γύρω, να χαμηλώσει τη φωνή του αν χρειάζεται και σε κάθε περίπτωση το ακροατήριο είναι ελέγξιμο και περιορισμένο.

Ο David Rosemblum [Ros07] έχει επισημάνει αυτήν την αυταπάτη και έχει σημειώσει ότι μια πιο ακριβής εικόνα για το κοινωνικό δίκτυο θα ήταν εκείνη ενός ατόμου που κρατάει ένα μεγάφωνο και απευθύνεται σε μια τεράστια συγκέντρωση. Επί πλέον παρατηρεί ότι στους συνηθισμένους χώρους συνάθροισης, π.χ. σε ένα καφέ, κανείς δεν κρατάει πρακτικά των όσων λέγονται, κανείς δεν τα αποθηκεύει για πολλά πολλά χρόνια και κανείς δεν έρχεται κατόπιν να τα διερευνήσει και να τα αναλύσει, πράγματα που αποτελούν κοινή πρακτική στα κοινωνικά δίκτυα. Όλα αυτά είναι γεγονότα που παραβλέπονται βραχυπρόθεσμα από τα μέλη των κοινωνικών δικτύων, αλλά είναι πιθανό να έχουν ακόμη και μακροπρόθεσμες συνέπειες. Τέτοιες συνέπειες γίνονται πραγματικότητα π.χ. επειδή οι εργοδότες κάνουν εκτεταμένη έρευνα στα κοινωνικά δίκτυα για το ποιόν των υποψηφίων προς πρόσληψη.

Οι διαμαρτυρίες για όλα τα παραπάνω δεν έχουν λείψει και σε ορισμένες περιπτώσεις έχουν οδηγήσει στη δημιουργία μηχανισμών, με τους οποίους τα μέλη των κοινωνικών δικτύων ασκούν έλεγχο πάνω στη διάδοση των πληροφοριών που δίνουν. Στατιστικά όμως τα μέλη των δικτύων δεν κάνουν ιδιαίτερη χρήση αυτών των μηχανισμών, συχνά δεν γνωρίζουν καν την ύπαρξή τους. Ακόμη και όσοι τους χρησιμοποιούν με μεγάλη επιμέλεια κινδυνεύουν από μεταβολές στο πλαίσιο έκθεσης των πληροφοριών, από την διάδοση των δεδομένων προς τρίτους έναντι ανταλλάγματος, από σφάλματα και από επιθέσεις ασφάλειας και ιδιωτικότητας.

Η έρευνα γύρω από τα κοινωνικά δίκτυα έχει διερευνήσει πιθανές λύσεις. Ορισμένα δίκτυα υπόσχονται προστασία των μελών τους και απουσία διαφημίσεων, αλλά η διάδοσή τους είναι περιορισμένη. Οι Anderson και Stajano [AS13] έχουν προτείνει ένα μοντέλο κοινωνικού δικτύου, του οποίου τα μέλη αποθηκεύουν τις πληροφορίες τους κρυπτογραφημένες στο υπολογιστικό νέφος (έναντι ενός ελάχιστου αντιτίμου), ενώ γύρω από τις πληροφορίες αυτές κινείται ένα σύνολο εφαρμογών, που η λειτουργία τους εξαρτάται αυστηρά από την συναίνεση των μελών.

13.8 Προβλήματα σχετικά με τη συλλογή και ανάλυση δεδομένων

Οι ψηφιακές μνήμες συσσωρεύουν καθημερινά νέους, όλο και μεγαλύτερους, όγκους δεδομένων. Μια από τις πρώτες χρήσεις των υπολογιστών ήταν στην υποστήριξη των λειτουργιών γραφείου και της βιομηχανικής παραγωγή. Η εποχή αυτή δημιούργησε βάσεις δεδομένων, όπου μια εταιρία μπορούσε να αποθηκεύσει τα στοιχεία των υπαλλήλων της, των πελατών της, των προϊόντων της, των αγορών της, των προμηθευτών της, των δεικτών παραγωγής της και γενικά όλων των χρήσιμων στοιχείων. Τα στοιχεία αυτά τις περισσότερες φορές προέρχονταν από πληκτρολόγηση και ήταν σχετικά

περιορισμένα σε όγκο. Βέβαια υπήρχαν σε ορισμένες περιπτώσεις και ολόκληρα εργοστάσια πληκτρολόγησης, όπως η υπηρεσία που περνούσε τα δεδομένων των φορολογικών δηλώσεων των πολιτών αντιγράφοντας (με μεγάλη ταχύτητα και συχνά με λάθη) τις χάρτινες φόρμες που οι τελευταίοι είχαν συμπληρώσει. Σήμερα τις πληροφορίες αυτές πληκτρολογούν οι ίδιοι οι πολίτες και οι λογιστές τους και καταλήγουν πάλι σε βάσεις δεδομένων.

Με την συλλογή και ανάλυση δεδομένων μπορούν να εξαχθούν ενδιαφέροντα συμπεράσματα. Για παράδειγμα, σε ένα δίκτυο συλλογής μετεωρολογικών δεδομένων από ιδιωτικούς σταθμούς στο λεκανοπέδιο Αττικής, πόσες φορές το χρόνο πέφτει η θερμοκρασία κάτω από το μηδέν σε ποσοστό μεγαλύτερο από το 50% των σταθμών; Στο video ενός drone που πετάει πάνω από μια λεωφόρο πόσα αυτοκίνητα έχουν καταγραφεί; , Πολλά από τα συμπεράσματα που προέρχονται από την ανάλυση δεδομένων έχουν σημαντική οικονομική αξία. Π.χ. τι μπορεί να καταλάβει το τμήμα marketing από τις αγορές που καταγράφηκαν από την loyalty card ενός πελάτη super-market και τι από τις αγορές πελατών που προέρχονται από μια συγκεκριμένη περιοχή και ηλικιακή ομάδα; Τι συμπέρασμα βγαίνει για τις προτιμήσεις ενός καταναλωτή από τη συμπεριφορά του στο Facebook, στις αναζητήσεις Google, σε καταστήματα on-line και αλλού;

Σήμερα πληθαίνουν όλο και περισσότερο δεδομένα που προέρχονται από μια πληθώρα πηγών και δεν είναι πάντοτε τόσο καλά οργανωμένα όσο αυτά που φιλοξενούνται σε μια βάση δεδομένων. Αν ένας διαχειριστής δικτύου αντιγράψει την κίνηση που περνάει από μια ζεύξη για συγκεκριμένο χρονικό διάστημα θα μαζέψει εγγραφές που προκαλεί μια σειρά από πακέτα. Παρομοίως ο χειριστής ενός drone που πετάει πάνω από μια περιοχή θα καταγράψει ένα video, που εν τέλει δεν είναι παρά μια ακολουθία από bits. Τα μέλη ενός κοινωνικού δικτύου εισάγουν δεδομένα διαφόρων τύπων (κείμενα, φωτογραφίες, video) που αποθηκεύονται στους servers που εξυπηρετούν το κοινωνικό δίκτυο. Όλο και περισσότερα «έξυπνα» αντικείμενα καταγράφουν καθημερινά νέους όγκους δεδομένων. Με δυο λόγια υπάρχει μια συσσώρευση πληροφοριών που γίνεται με αυξανόμενους ρυθμούς από μια επίσης αυξανόμενη ποικιλία πηγών.

Κάποιες απαντήσεις στα περίπλοκα προβλήματα που δημιουργούνται από τον όγκο και την πολυμορφία των δεδομένων προσπαθούν να δώσουν δύο τεχνολογίες που αναπτύσσονται τα τελευταία χρόνια, η *εξόρυξη δεδομένων* (data mining) και τα *δεδομένα μεγάλου όγκου* (big data).

Τόσο οι πιο παραδοσιακοί, όσο και οι νεότεροι τρόποι συλλογής, αποθήκευσης και ανάλυσης δεδομένων δημιουργούν μια σειρά από προβλήματα (ασφάλειας και) ιδιωτικότητας, μερικά από τα οποία θα εξετάσουμε παρακάτω.

Το πρόβλημα της αποκάλυψης ή επαναταυτοποίησης

Έστω ότι ένας εισβολέας έχει πρόσβαση σε μια συλλογή δεδομένων. Το πρόβλημα της αποκάλυψης (disclosure) ή επαναταυτοποίησης (re-identification) προκύπτει με τις εξής μορφές [TMK13]:

Αποκάλυψη της ταυτότητας Ο εισβολέας προσπαθεί να συνδέσει ένα γνωστό άτομο με κάποια εγγραφή, χρησιμοποιώντας πρόσθετες πληροφορίες που γνωρίζει. Στη συνέχεια εφόσον έχει γίνει η ταυτοποίηση ο εισβολέας μπορεί να αποκτήσει κι όλες τις άλλες πληροφορίες που περιέχει η συλλογή για το άτομο.

Αποκάλυψη χαρακτηριστικών Ο εισβολέας καταφέρνει να μάθει νέα χαρακτηριστικά για ένα άτομο χρησιμοποιώντας δεδομένα της συλλογής.

Συμπερασματική αποκάλυψη Ο εισβολέας μπορεί να συνδέσει δεδομένα με πρόσωπα με κατάλληλη επεξεργασία των δεδομένων.

Ο όρος *προσωπικά δεδομένα* (personal data) ή *δεδομένα προσωπικού χαρακτήρα* χρησιμοποιείται στην Ευρωπαϊκή Ένωση και προσδιορίζει πληροφορίες που ανήκουν σε ένα αναγνωρίσιμο πρόσωπο.[8] Στην Β. Αμερική χρησιμοποιούνται αντίστοιχα οι όροι *ευαίσθητες προσωπικές πληροφορίες* (Sensitive Personal Information, SPI) και *προσωπικά αναγνωρίσιμες πληροφορίες* (personally identifiable information, PII).[9]

Οι προφανείς πληροφορίες που δίνουν εύκολη ταυτοποίηση είναι ονόματα, αριθμοί ταυτότητας, ασφάλισης, άδειες οδήγησης, εκλογικοί αριθμοί και γενικώς αριθμοί σε διάφορα μητρώα, ημερομηνίες γέννησης, τόποι γέννησης, διευθύνσεις κατοικίας, φωτογραφίες, βιομετρικά δεδομένα, login names, διευθύνσεις IP, διευθύνσεις email κ.λπ.

Ο στόχος ορισμών όπως οι παραπάνω είναι να ασκηθεί έλεγχος στη συλλογή και επεξεργασία προσωπικών δεδομένων. Η έννοια του ελέγχου είναι να χρησιμοποιούνται τα προσωπικά δεδομένα με μέτρο και μόνο εφόσον και όταν είναι απαραίτητα για ένα συγκεκριμένο εγκεκριμένο σκοπό, αναγνωρίζοντας ότι γενικά η αγορά, η περίθαλψη, το κράτος κ.λπ. δεν μπορούν να λειτουργήσουν αν παραιτηθούν όλωσδιόλου από τη συλλογή προσωπικών δεδομένων. Ορισμοί όπως οι παραπάνω περιλαμβάνονται μέσα σε οδηγίες που έχουν ως σημείο εκκίνησης την επιθυμία για διακίνηση δεδομένων προκειμένου να διευκολυνθούν διάφορες δραστηριότητες, οι οποίες είναι σημαντικές για την οικονομία της κάθε χώρας και της ΕΕ συνολικά. Η Ευρωπαϊκή Οδηγία 95/46 θέτει αυτά τα ζητήματα ως εξής: *Η οικονομική και κοινωνική ολοκλήρωση που απορρέει από την εγκαθίδρυση και τη λειτουργία της εσωτερικής αγοράς ... συνεπάγονται κατ' ανάγκη αισθητή αύξηση της διασυνοριακής ροής δεδομένων προσωπικού χαρακτήρα ... η ανταλλαγή δεδομένων προσωπικού χαρακτήρα μεταξύ επιχειρήσεων που είναι εγκατεστημένες στα διάφορα κράτη μέλη βρίσκεται σε ανέλιξη.*

Ωστόσο το πραγματικό πρόβλημα της πιθανής κατάχρησης των προσωπικών δεδομένων δεν επιλύεται μόνο με τη χρήση ορισμών γύρω από τα προσωπικά δεδομένα, δηλαδή φτιάχνοντας ουσιαστικά ένα κατάλογο «απαγορευμένων» πληροφοριών, που αν λείψουν θα διατηρηθεί η ανωνυμία [NS10]. Κατά πόσο μια πληροφορία οδηγεί σε αποκάλυψη ενός προσώπου εξαρτάται από το σύνολο των διαθέσιμων πληροφοριών και δεν υπάρχει εύκολος τρόπος να αποφασίσουμε αν μια πληροφορία είναι κρίσιμη. Η υπόθεση του Netflix Prize (βλ. σελ. 233) είναι χαρακτηριστική επειδή δείχνει πώς εκ πρώτης όψεως επαρκώς απρόσωπα δεδομένα μπορούν να οδηγήσουν σε ταυτοποίηση.

Προβλήματα σε βάσεις δεδομένων

Τα προβλήματα της σύνδεσης δεδομένων με συγκεκριμένα πρόσωπα εντοπίσθηκαν αρχικά σε βάσεις δεδομένων, αν και αποτελούν γενικότερα προβλήματα όταν υπάρχει αποθηκευμένος όγκος δεδομένων, είτε τα δεδομένα είναι οργανωμένα σε βάση είτε όχι. Δίνουμε όμως αρχικά ορισμένα παραδείγματα από βάσεις, επειδή τα παραδείγματα αυτά είναι πιο απλά και δείχνουν τις βασικές αρχές.

[8] Σύμφωνα με το άρθρο 2α της Ευρωπαϊκής Οδηγίας 95/46/ΕC είναι *«δεδομένα προσωπικού χαρακτήρα», κάθε πληροφορία που αναφέρεται σε φυσικό πρόσωπο του οποίου η ταυτότητα είναι γνωστή ή μπορεί να εξακριβωθεί «το πρόσωπο στο οποίο αναφέρονται τα δεδομένα» - ως πρόσωπο του οποίου η ταυτότητα μπορεί να εξακριβωθεί λογίζεται το πρόσωπο εκείνο που μπορεί να προσδιοριστεί, άμεσα ή έμμεσα, ιδίως βάσει αριθμού ταυτότητας ή βάσει ενός ή περισσοτέρων συγκεκριμένων στοιχείων που χαρακτηρίζουν την υπόστασή του από φυσική, βιολογική, ψυχολογική, οικονομική, πολιτιστική ή κοινωνική άποψη.*

[9] *Personal information means any information about an identified or identifiable individual* [apec05].

Το πλεονέκτημα μιας βάσης δεδομένων σε σύγκριση με άλλους τρόπους αποθήκευσης (π.χ. σε έναν πίνακα ή την απλή αποθήκευση ενός ρεύματος δεδομένων όπως έρχεται από την πηγή) βρίσκεται στους μηχανισμούς ανάσυρσης των δεδομένων. Οι σχετικοί μηχανισμοί για βάσεις προϋπήρξαν των σημερινών μηχανών αναζήτησης. Για παράδειγμα χρησιμοποιώντας μια βάση είναι σχετικά εύκολο να ψάξει κανείς μέσα σε ένα πλήθος δεδομένων χρησιμοποιώντας μια καλά δομημένη γλώσσα και να πάρει απαντήσεις σε μια σειρά από απλά ή πιο πολύπλοκα ερωτήματα. Αυτά μπορεί να σχετίζονται είτε με συγκεκριμένα πρόσωπα και αντικείμενα («πού ψηφίζω;», «ποια είναι σήμερα τα δρομολόγια του τρένου από Αθήνα προς Θεσσαλονίκη;») είτε να έχουν στατιστικό χαρακτήρα («πόσοι επιβάτες ταξίδεψαν από Αθήνα προς Λάρισα μέσα στο 2015;»).

Μια βάση δεδομένων μπορεί να έχει στοιχεία για ένα μεγάλο πληθυσμό προσώπων, πραγμάτων κ.λπ., αλλά διαφορετικοί χρήστες της βάσης (άνθρωποι ή προγράμματα) έχουν διαφορετικά δικαιώματα πρόσβασης στη βάση και ανάκτησης δεδομένων. Όταν π.χ. πρόκειται να κάνει ένας επιβάτης μιας πτήσης web check-in δίνει τον αριθμό εισιτηρίου του (ή ένα άλλο μοναδικό στοιχείο) και το λογισμικό που υλοποιεί αυτήν την υπηρεσία κάνει την κατάλληλη ερώτηση στη βάση δεδομένων με τις κρατήσεις και τα εισιτήρια. Με δυο λόγια ο επιβάτης δεν έχει απ᾽ ευθείας πρόσβαση στη βάση δεδομένων, οπότε δεν είναι σε θέση να μάθει π.χ. ποιοι άλλοι είναι στην ίδια πτήση.

Το περιεχόμενο μιας βάσης δεδομένων μπορεί να είναι κρυπτογραφημένο για προστασία από παραβιάσεις και αλλοιώσεις. Στους μηχανισμούς προστασίας συχνά περιλαμβάνονται μηχανισμοί καταγραφής των ενεργειών των χρηστών, προκειμένου να δημιουργηθούν αρχεία κατάλληλα για την εξιχνίαση παραβιάσεων.

Στατιστικές βάσεις δεδομένων

Στατιστική βάση δεδομένων λέγεται μια βάση δεδομένων που μπορεί να χρησιμοποιείται για στατιστικές αναλύσεις [CO81]. Η αναζήτηση μπορεί να γίνεται με τους συνήθεις τρόπους (π.χ. SQL) ή να παρέχονται πιο προχωρημένα εργαλεία στατιστικής ανάλυσης. Όταν η βάση περιέχει στοιχεία προσωπικού χαρακτήρα διαθέτει μηχανισμούς με τους οποίους προστατεύεται η ιδιωτικότητα. Μερικοί από τους μηχανισμούς αυτούς είναι οι εξής:

Περιορισμένη πρόσβαση Στη βάση δεδομένων έχουν πρόσβαση μόνο εξουσιοδοτημένα άτομα, π.χ. σε μια βάση ιατρικών δεδομένων ενός νοσοκομείου μπορούν να έχουν πλήρη πρόσβαση μόνο οι γιατροί του νοσοκομείου. Στατιστικοί ερευνητές έχουν περιορισμένη πρόσβαση, όπως πιο κάτω.

Αοριστία Επιτρέπονται μόνον ερωτήσεις μειωμένης ακρίβειας και δίνονται αντίστοιχες απαντήσεις, π.χ. δίνονται ημερομηνίες γέννησης κατά προσέγγιση πενταετίας και τόποι γέννησης μόνο ως ένα ευρύτερο γεωγραφικό διαμέρισμα.

Συλλογικά μεγέθη Επιτρέπονται μόνο στατιστικά ερωτήματα πάνω στο σύνολο του πληθυσμού ή σ᾽ ένα δείγμα αρκετά μεγάλο, και δίνονται μόνο συλλογικές απαντήσεις, π.χ. `COUNT`, `SUM`, `AVG`, `MIN`, `MAX`.

Απάντηση υπό συνθήκη Δίνεται απάντηση σε οποιαδήποτε ερώτηση, αλλά μόνον αν η απάντηση δεν μπορεί να οδηγήσει σε αποκάλυψη με κάποιο κριτήριο. Π.χ. δίνεται απάντηση μόνον αν οι απαντήσεις είναι περισσότερες από ένα ελάχιστο αριθμό.

Τεχνητή νοημοσύνη Χρησιμοποιείται ένα μοντέλο απόφασης που κρίνει για το αν πρέπει να απαντήσει με βάση το ιστορικό των ερωτήσεων που έχουν γίνει από τον ίδιο ερευνητή και γενικότερα προσπαθεί να καταλάβει αν γίνεται κατάχρηση της βάσης.

Καταγραφή και έλεγχος Οι ερωτήσεις του κάθε ερευνητή καταγράφονται με σκοπό να μπορεί να γίνεται εκ των υστέρων έλεγχος για πιθανές καταχρήσεις και να υπάρχουν ίχνη εφόσον συμβούν δυσάρεστα περιστατικά.

Παρά τους παραπάνω και άλλους μηχανισμούς δεν υπάρχει εγγύηση μη παραβίασης μιας βάσης κάτω από οποιοδήποτε σενάριο επίθεσης [AW89]. Κατά κανόνα σε ο εισβολέας χρησιμοποιεί ανεξέλεγκτες πρόσθετες πληροφορίες τις οποίες μπορεί να συνδυάσει με τα αποτελέσματα από τη χρήση της στατιστικής βάσης. Τα επόμενα δύο σενάρια δίνει μια ιδέα για τέτοιου είδους παραβιάσεις.

Σενάριο 1 Σε μια σχολή το ιατρείο διατηρεί μια βάση δεδομένων με τα στοιχεία των φοιτητών που το έχουν επισκεφθεί και τη διάγνωση που έχει γίνει για τον καθένα. Ο γιατρός που απασχολείται στο συγκεκριμένο ιατρείο έχει πλήρη πρόσβαση σε όλα τα στοιχεία, ενώ ένα τμήμα του διοικητικού προσωπικού έχει το δικαίωμα να παίρνει μόνο αριθμητικές ερωτήσεις, π.χ. πόσοι είναι άρρωστοι με γρίπη και ανήκουν πιθανώς σε κάποιο υποσύνολο, αλλά δεν επιτρέπεται να ζητήσει κάποιο πίνακα ασθενών και δεν μπορεί να κάνει αριθμητική ερώτηση για συγκεκριμένο όνομα.

Ένας υπάλληλος είναι συγγενής μιας φοιτήτριας, έχει ακούσει ότι αυτή έχει επισκεφθεί το ιατρείο, και θέλει να μάθει από τι πάσχει. Κάνει μια αναζήτηση για τον αριθμό γυναικών που έχουν επισκεφθεί το ιατρείο με την ερώτηση

```
SELECT COUNT(name) FROM [studentnames]
WHERE sex="F";
```

και παίρνει την απάντηση 5. Στη συνέχεια, δεδομένου ότι γνωρίζει την χρονολογία γέννησής της, περιορίζει την προηγούμενη αναζήτηση σε όσες έχουν γεννηθεί το συγκεκριμένο έτος με την ερώτηση

```
SELECT COUNT(name) FROM [studentnames]
WHERE sex="F" AND  yearofbirth=1999;
```

και παίρνει την απάντηση 1. Τέλος κάνει την ίδια ερώτηση με τον επί πλέον περιορισμό να έχει ο ασθενής γρίπη:

```
SELECT COUNT(name) FROM [studentnames]
WHERE sex="F" AND yearofbirth=1999 AND diagnosis="flu";
```

Δεδομένου ότι και πάλι παίρνει την απάντηση 1, αντιλαμβάνεται ότι η μόνη ασθενής με γρίπη είναι η συγγενής του.

Σενάριο 2 Στο ίδιο ιατρείο όπως παραπάνω το σύστημα που ελέγχει την πρόσβαση στους υπαλλήλους έχει αναβαθμισθεί και αποκρύπτει την απάντηση αν το αποτέλεσμα είναι μικρότερο από 20. Ο αδιάκριτος συγγενής όμως μπορεί τώρα να κάνει μια σειρά από τροποποιημένες ερωτήσεις, που περιέχουν ένα ξένο προς την αρχική ερώτηση πληθυσμό, έτσι ώστε να αυξήσει το αποτέλεσμα και να περάσει τον πήχυ. Πρώτα κάνει την ερώτηση πόσοι άρρενες φοιτητές έχουν επισκεφθεί το ιατρείο, δηλαδή

```
SELECT COUNT(name)FROM [studentnames] WHERE sex="M";
```

και παίρνει την απάντηση 314. Στη συνέχεια κάνει την εξής σύνθετη ερώτηση που περιέχει το προηγούμενο σύνολο και το σύνολο που τον ενδιαφέρει μαζί:

```
SELECT COUNT(tem.name) FROM (
SELECT name FROM [studentnames]
WHERE sex="F" AND yearofbirth=1999
UNION ALL
SELECT name FROM [studentnames] WHERE sex="M"
) AS tem;
```

Στην ερώτηση αυτή παίρνει την απάντηση 315, οπότε γνωρίζει ότι το υπάρχει μόνο μια φοιτήτρια με τις γνωστές ιδιότητες. Τέλος κάνει την ερώτηση

```
SELECT COUNT(tem.name) FROM (
SELECT name FROM [studentnames]
WHERE sex="F" AND yearofbirth=1999 AND diagnosis="flu"
UNION ALL
SELECT name FROM [studentnames] WHERE sex="M"
) AS tem;
```

και ξαναπαίρνει την απάντηση 315, άρα η φοιτήτρια που ψάχνει έχει γρίπη.

Τεχνικές ανωνυμοποίησης

Οι τεχνικές ανωνυμοποίησης αποσκοπούν στο να επιτρέψουν τη χρήση πληροφοριών από μια βάση ή μια συλλογή δεδομένων χωρίς να αποκαλύπτονται πληροφορίες για συγκεκριμένα πρόσωπα.

Για την ανωνυμοποίηση έχουν προταθεί πλείστες όσες μέθοδοι, άλλες περισσότερο κι άλλες λιγότερο συστηματικές. Μερικές από τις πιο απλές συνίστανται στην απλή διαγραφή πεδίων, την αντικατάστασή τους με ψευδή αντίστοιχα και την κρυπτογράφηση και την ομαδοποίηση εγγραφών.

Η χρήση τέτοιων μεθόδων κατά κανόνα έχει ως αποτέλεσμα την απώλεια πληροφορίας, αλλά η απώλεια εξαρτάται από την τεχνική ανωνυμοποίησης και περαιτέρω κατά πόσο θα αλλοιώσει το στατιστικό αποτέλεσμα εξαρτάται από την τελική χρήση των δεδομένων. Η εύρεση ενός σημείου ισορροπίας μεταξύ απωλειών και αποκάλυψης αποτελεί σημαντικό πρόβλημα.

Οι δύο επόμενοι όροι περιγράφουν τεχνικές ανωνυμοποίησης. Από άλλους συγγραφείς θεωρούνται διακριτοί κι από άλλους όχι. Ο όρος *έλεγχος στατιστικής αποκάλυψης* (statistical disclosure control) χρησιμοποιείται για να περιγράψει γενικά τεχνικές αλλοίωσης των δεδομένων. Ο όρος *περιορισμός στατιστικής αποκάλυψης* (statistical disclosure limitation) αναφέρεται σε τεχνικές που προσπαθούν να περιορίσουν τις λεπτομέρειες [MH11; AS15], όπως οι εξής:

- Περιορισμός λεπτομερειών: Π.χ. σε μια απογραφή πληθυσμού δεν δίνονται γεωγραφικοί προσδιορισμοί που θα μπορούσαν να οδηγήσουν σε δείγμα μικρότερο των N ατόμων.

- Κωδικοποίηση-στρογγυλοποίηση των ακραίων τιμών: Ακραίες τιμές ορισμένων χαρακτηριστικών παραμέτρων (π.χ. του εισοδήματος) προς τα πάνω ή προς τα κάτω αντικαθίστανται με περίπου περιγραφές (π.χ. «εισόδημα πάνω από 100000»).

- Εγγραφές με σπάνιες τιμές παραμέτρων δεν εμφανίζονται καθόλου.

- Στρογγυλοποίηση τιμών των παραμέτρων. Η στρογγυλοποίηση μπορεί να γίνεται και τυχαία, δηλαδή με πιθανότητα p η τιμή στρογγυλοποιείται και με πιθανότητα $1 - p$ δεν στρογγυλοποιείται.

- Προσθήκη θορύβου στις τιμές ορισμένων παραμέτρων.

k-ανωνυμία, ℓ-ποικιλία

Δεδομένης της απώλειας πληροφορίας που συνεπάγονται όλες αυτές οι μέθοδοι, τίθεται το πρόβλημα της ελάχιστης δυνατής αλλοίωσης για την επίτευξη ενός επιθυμητού επιπέδου κινδύνου.

Η ιδιότητα της *k-ανωνυμίας* (k-anonymity) ισχύει εφόσον οποιοσδήποτε συνδυασμός των πεδίων των εγγραφών δεν μπορεί να οδηγήσει σε σύνολο με πληθικό αριθμό μικρότερο από k [SS98]. Με δυο λόγια, οποιαδήποτε αναζήτηση δε μπορεί να δώσει κάτω από k εγγραφές, οπότε η πιθανότητα αποκάλυψης δεν μπορεί να είναι μεγαλύτερη από $1/k$. Μια συνηθισμένη πρακτική είναι να τίθεται $k = 3$ [TMK13]. Προφανώς η k-ανωνυμία δεν είναι αρκετή σε όλες τις περιπτώσεις. Αν για παράδειγμα ένας εισβολέας γνωρίζει ότι και οι k εικοσάχρονες από μια αναζήτηση σε μια ιατρική βάση έχουν γρίπη των πτηνών, τότε και η γνωστή του εικοσάχρονη έχει γρίπη των πτηνών.

Προβλήματα όπως στο προηγούμενο παράδειγμα προσπαθεί να αντιμετωπίσει η μέθοδος της ℓ-ποικιλίας [Mac+07]. Όταν ισχύει αυτή η ιδιότητα κάθε ομάδα περιέχει αρκετές εγγραφές ώστε να περιέχονται το λιγότερο ℓ επαρκώς αντιπροσωπευόμενες τιμές της ευαίσθητης μεταβλητής (που θα μπορούσε να οδηγήσει σε αποκάλυψη).

Κοινωνικά δίκτυα και ψυχομετρικές μέθοδοι

Το 2012-13 ο M. Kosinski και άλλοι ερευνητές από το Psychometrics Center του πανεπιστημίου του Cambridge ανακοίνωσαν [Bac+12; KSG13] ότι είχαν καταφέρει να εκτιμήσουν με ικανοποιητική ακρίβεια μια σειρά από προσωπικά χαρακτηριστικά χρησιμοποιώντας δεδομένα του τύπου που συλλέγονται στο Facebook, κυρίως *Likes*. Τα χαρακτηριστικά αυτά είναι για παράδειγμα πολιτικές και θρησκευτικές απόψεις, χρήση ουσιών, ηλικία, φύλο, κατά πόσο προέρχονται από χωρισμένους γονείς, σεξουαλικός προσανατολισμός κ.α., με την κατανόηση ότι τα μέλη του κοινωνικού δικτύου δεν έκαναν ποτέ άμεσες ή έμμεσες δηλώσεις από τις οποίες να μπορούν να συναχθούν τα παραπάνω με προφανή τρόπο. Στη συνέχεια το 2015 οι Youyou, Kosinski και Stillwell υποστήριξαν [YKS15] ότι τέτοιου είδους εκτιμήσεις είναι πιο αξιόπιστες από τις αντίστοιχες που θα μπορούσαν να είχαν γίνει από πρόσωπα του στενού κύκλου ενός προσώπου, δηλαδή συζύγους, γονείς, στενούς φίλους και συναδέλφους. Τον Ιανουάριο του 2017 δημοσιεύθηκε στο *Vice* ένα άρθρο [GK17], όπου οι H. Grassegger και M. Krogerus εξέφραζαν την άποψη ότι τα ερευνητικά αποτελέσματα του Kosinski όχι μόνον ήταν εξαιρετικά επικίνδυνα, αλλά είχαν ήδη αξιοποιηθεί στις πρόσφατες αμερικανικές προεδρικές εκλογές, καθώς και σε άλλες εκλογικές αναμετρήσεις, όπως σε Ουκρανία, Νιγηρία, περιλαμβανομένου του δημοψηφίσματος για το Brexit στο Ηνωμένο Βασίλειο. Στο άρθρο υποστηριζόταν ότι η αγγλική εταιρία Cambridge Analytica είχε χρησιμοποιήσει τη μέθοδο Kosinski (χωρίς την άδειά του τελευταίου) αγοράζοντας δεδομένα από το Facebook και στη συνέχεια ότι τα δεδομένα αυτά είχαν αναλυθεί και τελικά χρησιμοποιηθεί για να πεισθούν οι ψηφοφόροι να ψηφίσουν τον Donald Trump. Στη συνέχεια το ίδιο άρθρο εξηγούσε με ποιο τρόπο αξιοποιήθηκαν τα δεδομένα για επηρεάσουν το εκλογικό αποτέλεσμα. Οι μέθοδοι άσκησης πειθούς ήταν είτε κλασσικές (π.χ. «πόρτα-πόρτα») είτε μιντιακές και ψηφιακές, αλλά προσω-

ποιημένες αξιοποιώντας τη γνώση για το προφίλ του συγκεκριμένου ψηφοφόρου. Π.χ. το «συνεργείο» που θα χτυπούσε την πόρτα κάποιου με υπερβολική φοβία για κλοπές θα έδινε έμφαση στις δηλώσεις του υποψηφίου που στηρίζουν την οπλοκατοχή. Ταυτόχρονα αξιοποιήθηκαν τα κοινωνικά δίκτυα για να παρουσιάζονται τα κατάλληλα μηνύματα στους κατάλληλους αποδέκτες.

Δραστηριότητες, όπως αυτές για τις οποίες κατηγορήθηκε η Cambridge Analytica, μπορούν να είναι περισσότερο ή λιγότερο σύννομες ανάλογα με τους νόμους του κάθε κράτους. Η νομοθεσία των ΗΠΑ, για παράδειγμα, είναι πιο χαλαρή σε θέματα επεξεργασίας δεδομένων από την αντίστοιχη της Ευρωπαϊκής Ένωσης. Ωστόσο ακόμη και με την πιο αυστηρή νομοθεσία είναι δύσκολο να αποδειχθεί ότι κάτι επιλήψιμο συμβαίνει μέσα στο «μαύρο κουτί» μιας εταιρίας, αν αυτή καταφέρνει να φυλάει τα μυστικά της. Ανάμεσα σ' αυτά εν προκειμένω περιλαμβάνονται και οι αλγόριθμοι με τους οποίους γίνεται η επεξεργασία και ανάλυση των δεδομένων. Τα ίδια τα δεδομένα που φέρεται η Cambridge Analytica να έχει αντλήσει από τα κοινωνικά δίκτυα δεν είναι διαφορετικά από εκείνα που αντλεί οποιαδήποτε εταιρία marketing.

Ιδιωτικότητα και IoT

Το διαδίκτυο των πραγμάτων είναι, υποτίθεται, ο κόσμος που έρχεται, και σε κάποιο βαθμό έχει ήδη έρθει με εφαρμογές έξυπνου σπιτιού και έξυπνων εργασιακών χώρων. Στη συνέχεια θα δούμε τι φέρνουν αυτές οι τεχνολογίες και ποιες είναι οι συνέπειές τους για την ιδιωτικότητα.

Διάχυτος υπολογισμός και διαδίκτυο των πραγμάτων

Γύρω στο 1990 ο Mark Weiser μίλησε για τις νέες τεχνολογίες που θα ενσωματωθούν στα καθημερινά μας αντικείμενα και στο χώρο που ζούμε κατά τρόπο «διαφανή», δηλαδή θα μας διευκολύνουν και θα εργάζονται για μας χωρίς να μας απασχολούν ιδιαίτερα [Wei91]. Ο Weiser εργαζόταν τότε στη Xerox PARC (Palo Alto Research Center) και είχε την ατυχία να πεθάνει νέος, το 1999, μόλις 47 ετών. Θεωρείται πατέρας του *διάχυτου υπολογισμού* (ubiquitous computing), περιοχής που στη συνέχεια έγινε γνωστή και ως *pervasive computing* ή *ambient intelligence* (στην Ευρώπη).

Στη συνέχεια ακολούθησε μια σειρά από προσπάθειες να διερευνηθεί ο τρόπος της υλοποίησης του οράματος του Weiser. Ένα παράδειγμα ενσωμάτωσης ευφυΐας σε αντικείμενο καθημερινής χρήσης ήταν το MediaCup project, που προσπάθησε να δημιουργήσει την «έξυπνη κούπα» [GSB02]. Σ' αυτήν ενσωματώθηκε ένα σύνολο αισθητήρων, ανάμεσά τους αισθητήρες θερμοκρασίας, κίνησης, βάρους και επαφής με μια υποκείμενη επιφάνεια (ένας διακόπτης που κλείνει όταν το ποτήρι πατάει στην επιφάνεια). Τα δεδομένα αυτών των αισθητήρων αναλύονται από κατάλληλο λογισμικό που μπορεί να συμπεράνει αν το ποτήρι είναι σε μια από τις εξής τέσσερεις κινητικές καταστάσεις: (α) Το ποτήρι είναι ακίνητο, (β) κάποιος πίνει απ' αυτό, (γ) κάποιος παίζει μ' αυτό, (δ) κάποιος το μεταφέρει. Με τους αισθητήρες θερμοκρασίας και βάρους μπορεί να καταλάβει αν είναι γεμάτο και αν έχει κρυώσει το περιεχόμενο. Τέλος, επικοινωνεί με το περιβάλλον και μπορεί να μεταφέρει τα συμπεράσματα και τις μετρήσεις για περαιτέρω επεξεργασία ή δράσεις (π.χ. να πάει κάποιος να ξαναγεμίσει το ποτήρι με ζεστό καφέ).

Μια άλλη χαρακτηριστική περίπτωση ήταν το *έξυπνο πάτωμα* [OA00], που διαθέτει αισθητήρες της πίεσης, η οποία ασκείται πάνω στα πλακάκια του όταν κάποιος πατάει σ' αυτά. Με κατάλληλο λογισμικό το έξυπνο πάτωμα μπορεί να βρει τη θέση

ενός ή περισσότερων ανθρώπων, την πορεία τους, ακόμη και να καταγράψει το χαρακτηριστικό τους βάδισμα ώστε να διακρίνει τον ένα από τον άλλο.

Ο βασικός στόχος τέτοιων πειραμάτων ήταν να διαπιστωθεί τι είδους υπηρεσίες μπορούσαν να προσφέρουν τα έξυπνα αντικείμενα, τελικά δηλαδή αν άξιζε να επενδύσει κανείς στην προσθήκη ευφυΐας σε αντικείμενα για χάρη των ωφελειών που θα προέκυπταν. Η έξυπνη κούπα θα μπορούσε ίσως να ειδοποιεί για την ποσότητα και τη θερμοκρασία του καφέ, ή να δίνει μια ένδειξη κατά πόσο έχει τελειώσει ένα meeting σε κάποιο χώρο (επειδή πια τα ποτήρια είναι ακίνητα και κρύα). Το έξυπνο πάτωμα θα μπορούσε ίσως να χρησιμοποιηθεί σαν μηχανισμός εντοπισμού των υπαλλήλων μέσα σ᾽ έναν εργασιακό χώρο, ώστε περαιτέρω να ανοίγουν πόρτες, να θερμαίνονται χώροι, να κατευθύνονται μηνύματα σε οθόνες τοίχου κ.ο.κ. Ένας τέτοιος μηχανισμός είναι πιο διακριτικός από ένα σύστημα με κάμερες.

Τα περισσότερο ή λιγότερο έξυπνα αντικείμενα εφόσον επικοινωνούν με το περιβάλλον τους, άρα διαθέτουν και δικτυακές διευθύνσεις, αποτελούν συνιστώσες του *διαδικτύου των πραγμάτων* (Internet of Things, IoT). Τα λιγότερο έξυπνα απ᾽ αυτά τα αντικείμενα είναι σήμερα αυτά που διαθέτουν απλώς ένα RFID και μπορούν να προδώσουν την ταυτότητα και τη θέση τους. Τα έξυπνα αντικείμενα θα μπορούν σε λίγο να προσφέρουν τις υπηρεσίες τους μέσα από κοινωνικά δίκτυα και να αλληλεπιδρούν με τα μέλη των δικτύων (ο σχετικός όρος είναι *Social Web of Things*).

Το συνολικό σύστημα θα περιλαμβάνει στη βάση του ένα τεράστιο πληθυσμό αντικειμένων, αλλά πιο πάνω (από πλευράς δικτυακής αρχιτεκτονικής) ένα δίκτυο πρόσβασης, ένα γενικό δίκτυο διασύνδεσης και ακόμη πιο πάνω ένα σύνολο εφαρμογών και υπηρεσιών που θα ελέγχουν τα αντικείμενα, θα μαζεύουν δεδομένα απ᾽ αυτά, και θα προσφέρουν υπηρεσίες (και θα τρέχουν σε servers και στο υπολογιστικό νέφος) [KVA12].

Πέρα από την κύρια λειτουργία τέτοιων έξυπνων αντικειμένων, μερικοί είδαν σ᾽ αυτά και μια υπόσχεση για βελτίωση της ιδιωτικότητας, επειδή μπορούσαν να αισθανθούν την ύπαρξη ανθρώπων σ᾽ ένα χώρο και να βγάλουν κάποια προσεγγιστικά συμπεράσματα για την κατάσταση και τη δραστηριότητά τους χωρίς να χρειάζονται το αδιάκριτο βλέμμα μιας κάμερας. Θα μπορούσαν δηλαδή να περιορίσουν στην πηγή το παραγόμενο ρεύμα δεδομένων στα απολύτως απαραίτητα.

Προβλήματα με αναρτήματα RFID

Η προσθήκη ενός *αναρτήματος RFID* (RFID tag) σε ένα φυσικό αντικείμενο είναι ο απλούστερος και πιο οικονομικός τρόπος συμμετοχής του στο έξυπνο περιβάλλον, δεδομένου ότι το σύστημα μπορεί να γνωρίζει τουλάχιστον την ύπαρξη και τη θέση του αντικειμένου.

Ένα απλό ανάρτημα RFID εκπέμπει έναν αριθμό που το ταυτοποιεί σε κάθε κοντινό αναγνώστη. Ο αριθμός ταυτότητας (ID) του αναρτήματος δεν περιέχει μόνο τη συνηθισμένη πληροφορία που περιέχεται σε ένα γραμμοκωδικό (barcode) και ταυτοποιεί γενικά ένα προϊόν (π.χ. γάλα της εταιρείας τάδε), αλλά και το σειριακό του αριθμό, γεγονός που το ταυτοποιεί μοναδικά. Κατά συνέπεια μπορεί να συνδεθεί με το πρόσωπο που μεταφέρει το συγκεκριμένο ανάρτημα και την παρουσία του σε διάφορες περιστάσεις, όπου υπάρχουν αναγνώστες ικανοί να διαβάσουν το ανάρτημα.

Πριν εξετάσει κανείς τα προβλήματα ιδιωτικότητας είναι καλό να αντιληφθεί τους λόγους, για τους οποίους είναι επιθυμητή η χρήση αναρτημάτων RFID, διαφορετικά η εύκολη λύση θα ήταν η κατάργησή τους. Τα καταστήματα πώλησης ευνοούν τη χρήση τους για να μπορούν να χρεώνουν προϊόντα στο ταμείο, να διαχειρίζονται τυχόν επιστροφές και επισκευές, να κάνουν απογραφή στα προϊόντα που είναι στα ράφια τους

και να προωθούν σχετικά προϊόντα πριν βγει ο πελάτης από το κατάστημα. Οι αγοραστές και χρήστες προϊόντων θα μπορούσαν να διευκολυνθούν στην αναζήτηση, τη χρήση, τη συλλογή τους (*ποια ρούχα ή ποια CD έχω στο ντουλάπι μου;*) ή την αντικατάστασή τους (π.χ. στο έξυπνο ψυγείο που γνωρίζει τι περιέχει και μπορεί ίσως να παραγγείλει αυτόματα ό,τι λείπει) [JRS03],

Τα προβλήματα ασφάλειας και ιδιωτικότητας τα σχετικά με αναρτήματα RFID (RFID tags) θα μπορούσαν να θεωρηθούν ότι αποτελούν το πρώτο κύμα έρευνας σχετικής με το έξυπνο περιβάλλον. Τα περισσότερα απ' αυτά τα προβλήματα έχουν να κάνουν με ενδεχόμενες ανεπιθύμητες αναγνώσεις ενός αναρτήματος.

Μερικά από τα πιο πρώιμα τέτοια προβλήματα εμφανίσθηκαν με την τοποθέτηση αναρτημάτων σε καταναλωτικά προϊόντα. Για παράδειγμα, η τοποθέτηση πάνω σ' ένα ρούχο είναι βολική για να απαριθμηθεί κατά την αποθήκευση και αποστολή, για να ελεγχθεί η τιμή του από διαχειριστές και πελάτες, για να χρεωθεί στο ταμείο όταν το αγοράζει ένας πελάτης και για να εμποδισθεί η παράνομη έξοδός του αν δεν έχει πληρωθεί. Ωστόσο πολλοί άρχισαν να αναρωτιούνται κατά πόσο φορώντας τα ρούχα με τα RFID θα καταγράφονται όταν περνούν από εισόδους και άλλα σημεία ελέγχου (βλ. π.χ. την περίπτωση της διαμαρτυρίας ενάντια στην Benetton την άνοιξη του 2003 [Sta03]).

Είναι ενδιαφέρον ότι τόσο οι επιθέσεις, όσο και τα μέτρα άμυνας, δεν έχουν να κάνουν μόνο με αλγόριθμους και πρωτόκολλα, αλλά μπορούν να εκμεταλλεύονται διάφορες ιδιότητες των αντικειμένων, π.χ. την απόσταση ανάγνωσης που ποικίλλει κατά περίπτωση και την φυσική κατασκευή των αντικειμένων (δείτε την τελευταία μέθοδο για τα ηλεκτρονικά διαβατήρια).

Κλωβός Faraday Ο πιο απλός τρόπος παρεμπόδισης της ανάγνωσης ενός RFID είναι να διατηρείται μέσα σε ένα μεταλλικό περίβλημα, π.χ. μια σειρά από πιστωτικές κάρτες μπορεί να προστατεύεται σε ένα «μεταλλικό» πορτοφόλι. Όμως για πολλά προϊόντα κάτι τέτοιο είναι άβολο ή αδύνατο.

Παρεμβολή Μια πιο ενεργητική μέθοδος θα ήταν η παρεμβολή στην περιοχή συχνοτήτων του RFID. Η μέθοδος αυτή χρησιμοποιείται σπάνια επειδή δεν έχει επιλεκτικότητα (δεν μπορεί να εμποδίσει την ανάγνωση συγκεκριμένων αναρτημάτων), χρειάζεται κατανάλωση ενέργειας και οδηγεί σε εκπομπή ακτινοβολίας που ίσως είναι βλαβερή για τον άνθρωπο.

RFID μιας χρήσης Ένας επίσης απλός τρόπος άμυνας στην καταγραφή των RFID είναι να γίνονται μιας χρήσης, δηλαδή να αφαιρούνται ή να απενεργοποιούνται μετά την πρώτη χρήση, π.χ. την αγορά στο ταμείο [Nay+15; Sta10]. Η μέθοδος αυτή βέβαια δεν μπορεί να εξυπηρετήσει συστήματα όπου το ίδιο ανάρτημα πρέπει να χρησιμοποιείται περισσότερες φορές, π.χ. δεν μπορεί να χρησιμοποιηθεί για να επιτρέπεται καθημερινά η είσοδος των εργαζομένων σε ένα χώρο. Μια άλλη ιδέα είναι να έχουν το δικαίωμα ανάγνωσης συγκεκριμένοι μόνο αναγνώστες, πράγμα που υλοποιεί η επόμενη λύση.

Χρήση συνάρτησης κατακερματισμού Η πρόσβαση στις πληροφορίες που περιέχει το ανάρτημα προστατεύεται από μια συνάρτηση κατακερματισμού (hash function) ως εξής [Wei+04]: Σε κάθε ανάρτημα T αντιστοιχίζεται ένα κλειδί K_T. Η βασική ιδέα είναι ότι στο ανάρτημα αποθηκεύεται σε μια μνήμη M αντί του K_T το αποτέλεσμα της επεξεργασίας του από μια συνάρτηση κατακερματισμού h, δηλαδή το $h(K_T)$. Επίσης

το ανάρτημα έχει δύο καταστάσεις, μπορεί να είναι κλειδωμένο ή ξεκλείδωτο. Όταν είναι κλειδωμένο η λειτουργικότητά του είναι περιορισμένη.

Πριν την πρώτη του χρήση (ή αφού γίνει διαγραφή) η μνήμη M είναι κενή και το ανάρτημα είναι ξεκλείδωτο. Κάποιος μπορεί να γίνει «ιδιοκτήτης» αυτού του προσαρτήματος επιλέγοντας ένα κλειδί K_T και υπολογίζοντας το $h(K_T)$. Εγγράφει (χρησιμοποιώντας κάποια συσκευή εγγραφής) στην M το $m = h(K_T)$ και τότε το ανάρτημα κλειδώνει. Ταυτόχρονα ο ιδιοκτήτης κρατάει σε ένα πίνακα το K_T μαζί με το αντίστοιχο $h(K_T)$.

Όταν πρόκειται να χρησιμοποιήσει το ανάρτημα, ο αναγνώστης του ιδιοκτήτη διαβάζει το m από το ανάρτημα. Στη συνέχεια ψάχνει τον πίνακά του προκειμένου να δει αν υπάρχει τέτοια κατακερματισμένη τιμή. Αν αυτό ισχύει μπορεί να βρει το αντίστοιχο $h^{-1}(m) = K_T$. Στη συνέχεια στέλνει το K_T στο ανάρτημα. Το ανάρτημα έχει τη δυνατότητα υπολογισμού της h και υπολογίζει το $h(K_T)$, το οποίο συγκρίνει με το αποθηκευμένο m. Αν είναι ίσα, το ανάρτημα ξεκλειδώνει.

Τυχαιοποιημένος έλεγχος Η προηγούμενη μέθοδος έχει το μειονέκτημα ότι το ανάρτημα στέλνει πάντοτε το ίδιο σήμα. Επομένως διάφορες εμφανίσεις του μπορούν να συσχετισθούν και να οδηγήσουν σε αποκάλυψη. Η βελτίωση που θα μπορούσε να γίνει είναι να χρησιμοποιηθεί μαζί ένας τυχαίος αριθμός, δηλαδή το ανάρτημα γεννάει ένα τυχαίο αριθμό r και εκπέμπει το $(r, h(K_T \| r))$. Ο επιτιθέμενος μη γνωρίζοντας το K_T πρέπει να ερευνήσει ένα μεγάλο χώρο. Ο νόμιμος αναγνώστης πρέπει να δοκιμάσει αν κάποιο από τα γνωστά του K_T ταιριάζει υπολογίζοντας σε κάθε δοκιμή το $(r, h(K_T \| r))$. Δεδομένου ότι αυτό απαιτεί κατανάλωση πόρων επεξεργασίας, η μέθοδος αυτή θεωρείται κατάλληλη μόνον όταν υπάρχουν λίγα αναρτήματα με διαφορετικούς κωδικούς (π.χ. δεν είναι κατάλληλη για ένα πολυκατάστημα) [Sta10].

Τροποποιημένος αλγόριθμος απομόνωσης Ένα πρόβλημα που εμφανίζεται σε ένα περιβάλλον με πολλά κοντινά αναρτήματα RFID είναι πώς ο αναγνώστης θα τα διαβάσει με τη σειρά. Παράδειγμα ενός τέτοιου προβλήματος προκύπτει αν ένας πελάτης εμφανισθεί στο ταμείο με ένα καλάθι γεμάτο προϊόντα και ο αναγνώστης πρέπει να τα ανακαλύψει όλα και να τα διαβάσει. Παρόμοιο πρόβλημα εμφανίζεται και σε ασύρματα τοπικά δίκτυα, αλλά εκεί οι διευθύνσεις των σταθμών είναι γνωστές εκ των προτέρων. Τέτοιες διαδικασίες είναι γνωστές ως αλγόριθμοι απομόνωσης ή αλγόριθμοι επίλυσης των συγκρούσεων.

Ας υποθέσουμε ότι ένας αναγνώστης πρέπει να καταλάβει ποια αναρτήματα είναι παρόντα, δεδομένου ότι έχουν διευθύνσεις με μήκος n bits. Ο απλοϊκός εξαντλητικός αλγόριθμος να κάνει όλες τις 2^n δυνατές ερωτήσεις προφανώς δεν είναι αποδοτικός, εκτός αν το n είναι πολύ μικρό. Ένας γνωστός αλγόριθμος (*binary tree walking*) είναι ο εξής: Ο αναγνώστης ζητάει το επόμενο (στην αρχή το πρώτο) bit από τα παρόντα αναρτήματα. Αν υπάρξουν διαφορετικές απαντήσεις γίνεται «σύγκρουση» και πρέπει να διαλέξει ανάμεσα μόνο σ' αυτούς που έχουν το bit ίσο με 0 ή με 1.

Ένας εισβολέας μπορεί να διαβάσει τα μηνύματα του αναγνώστη, αλλά όχι τα μικρής εμβέλειας μηνύματα των αναρτημάτων. Οι Weiss, Sarma κ.α. [Wei+04] πρότειναν την εξής παραλλαγή του παραπάνω αλγόριθμου για βελτίωση της ασφάλειας: Ας υποθέσουμε ότι υπάρχουν δύο αναρτήματα με διευθύνσεις $b_1 b_2$ και $b_1 \bar{b}_2$. Ο αναγνώστης στο πρώτο βήμα, όπου δεν υπάρχει σύγκρουση μαθαίνει το b_1, το οποίο όμως δεν γνωρίζει ο εισβολέας. Στη συνέχεια αντί να ζητήσει το 0 ή το 1 για να απομονώσει ένα από τα δύο αναρτήματα, ζητάει είτε το $b_1 \oplus b_2$ είτε το $b_1 \oplus \bar{b}_2$, χρησιμοποιεί δηλαδή το προηγούμενο γνωστό bit ως μάσκα για το επόμενο.

Η ίδια ιδέα μπορεί να αξιοποιηθεί κατά τη φάση της επικοινωνίας μεταξύ του αναρτήματος και του αναγνώστη. Κάθε φορά που ο αναγνώστης επιθυμεί να στείλει ένα μήνυμα a στο ανάρτημα, το ανάρτημα του στέλνει ένα μήνυμα b μιας χρήσης που χρησιμεύει ως μάσκα για τον αναγνώστη, δηλαδή ο αναγνώστης εκπέμπει προς το ανάρτημα το $a \oplus b$. Το μέτρο αυτό βασίζεται και πάλι στη δυνατότητα του εισβολέα να λαμβάνει μόνο τα μηνύματα του αναγνώστη, αλλά όχι τα μηνύματα του αναρτήματος.

Το ανάρτημα φραγής (the blocker tag) παρεμποδίζει την ανάγνωση επιλεγμένων αναρτημάτων δημιουργώντας στον αναγνώστη δυσκολίες εξερεύνησης του χώρου των σειριακών αριθμών των αναρτημάτων όταν χρησιμοποιείται ένας αλγόριθμος εξερεύνησης όπως αυτός της εξερεύνησης δέντρου (tree walking) [JRS03]. Η πιο απλή περίπτωση είναι να δημιουργηθεί μια προστατευμένη ζώνη αριθμών, π.χ. όλοι οι αριθμοί που αρχίζουν από 1. Κάθε φορά που ένας αναγνώστης αποπειράται να βρει αναρτήματα στην προστατευμένη ζώνη το ανάρτημα φραγής στέλνει ταυτόχρονα 1 και 0, οπότε ο αναγνώστης αναγκάζεται να εξερευνήσει όλο το χώρο της περιοχής (π.χ. όλο το δέντρο των αριθμών που αρχίζουν από 1, δηλαδή όλους τους 2^{k-1} σειριακούς αριθμούς, όταν k είναι το συνολικό μήκος του κάθε αριθμού).

Ωστόσο η χρήση ενός τέτοιου συστήματος θέτει μια σειρά από τεχνικά προβλήματα. Ένα εξ αυτών είναι ότι ο αναγνώστης μπορεί να «χαθεί» μέσα στην προστατευμένη περιοχή και να μην προχωρήσει ποτέ στην ανάγνωση των αναρτημάτων εκτός αυτής και τα οποία οφείλει να μπορεί να δει. Αυτό σημαίνει ότι πρέπει να τροποιηθεί κατάλληλα ο αλγόριθμος εξερεύνησης του δέντρου.

Ένα δεύτερο ζήτημα είναι πόσο πολύπλοκο θα είναι το σύστημα φραγής. Η απλούστερη υλοποίηση μπορεί να γίνει όταν η ζώνη φραγής είναι μια περιοχή που περιλαμβάνεται ολόκληρη κάτω από ένα κόμβο του δυαδικού δέντρου, π.χ. όλα τα αναρτήματα που αρχίζουν από 1 ή από 1001. Τότε όμως η προστασία ενός αναρτήματος (που προφανώς αρχικά δεν είναι μέσα στην περιοχή) θα χρειαστεί μεταβολή του σειριακού αριθμού. Για παράδειγμα αν η προστατευμένη περιοχή είναι αυτή με αριθμούς που αρχίζουν από 1, ενώ όλα τα κανονικά αναρτήματα αρχίζουν από 0, η προστασία παρέχεται για όσο χρειαστεί αλλάζοντας το πρώτο bit (βλ. [JRS03] για μια σειρά από παραδείγματα χρήσης). Τέλος, ένα άλλο μειονέκτημα της μεθόδου είναι ότι μπορεί να χρησιμοποιηθεί για επιθέσεις άρνησης υπηρεσίας, εξωθώντας και πάλι τους αναγνώστες σε «ατέλειωτες» αναζητήσεις.

Πρωτόκολλα περιορισμού της απόστασης (distance bounding protocols) Μια συνηθισμένη προσέγγιση σε μηχανισμούς ελέγχου πρόσβασης, τηλεχειρισμούς κ.λπ., περιλαμβανομένων και περιπτώσεων που υλοποιούνται με τη χρήση RFID, είναι τα *πρωτόκολλα πρόκλησης και απόκρισης* (challenge and response protocols). Συνηθισμένη επίθεση στα πρωτόκολλα αυτής της κατηγορίας είναι η *επίθεση του ενδιάμεσου* (man in the middle attack), κατά την οποία ο επιτιθέμενος παρεμβάλλεται ανάμεσα σε δυο «νόμιμες» πλευρές και απλώς προωθεί τη μηνύματα της μιας στην άλλη, χωρίς να χρειάζεται να τα αποκρυπτογραφεί ή να τα επεξεργάζεται με οποιοδήποτε τρόπο.

Μια ιδέα για να αμυνθεί κανείς σε τέτοιου είδους επιθέσεις στην περίπτωση που οι δύο πλευρές είναι λογικά σε μικρή απόσταση, όπως στην περίπτωση της χρήσης RFID, είναι να γίνει ανίχνευση της απόστασης [BC93]. Απλές μετρήσεις απόστασης (χωρίς δηλαδή πρόσθετους αισθητήρες) είναι αναγκασμένες να βασίζονται στη διάδοση του ηλεκτρομαγνητικού κύματος. Πρόκειται δηλαδή για ένα χρονικό διάστημα της τάξης μεγέθους του 0,1 m / 3×10^8 m/sec, δηλαδή περί το ένα nsec ή και λιγό-

τερο. Επιπρόσθετες καθυστερήσεις λόγω επεξεργασίας μπορούν να αλλάξουν αρκετά το συνολικό χρόνο, οπότε η επιβεβαίωση μπορεί να αποτύχει χωρίς στην πραγματικότητα να υπάρχει πρόβλημα. Για τον λόγο αυτόν τέτοιες επιβεβαιώσεις γίνονται με την ελάχιστη δυνατή επεξεργασία, π.χ. με αποστολή και λήψη ενός μόνο bit, αλλά με αρκετές επαναλήψεις προκειμένου να υπάρξει αξιόπιστο αποτέλεσμα τόσο από πλευράς ασφάλειας όσο και από πλευράς στατιστικής μέτρησης του χρόνου. Ως προς το ζήτημα της ασφάλειας, εάν οι δύο πλευρές είναι προετοιμασμένες να ανταλλάξουν n συγκεκριμένες προκλήσεις και αποκρίσεις του ενός bit, μια σειρά από τυχαίες απαντήσεις εκ μέρους του επιτιθέμενου έχει πιθανότητα επιτυχίας ίση με 2^{-n}, οπότε αρκεί να χρησιμοποιηθεί αρκετά μεγάλο n [Sta10].

Πολλαπλός έλεγχος πρόσβασης (multi-factor access control) Ένα παράδειγμα χρήσης τέτοιας μεθόδου είναι η πρόσβαση στα δεδομένα του ηλεκτρονικού διαβατηρίου. Το διαβατήριο μπορεί να έχει αποθηκευμένα διάφορα βιομετρικά δεδομένα του κατόχου του, όπως δακτυλικά αποτυπώματα, χαρακτηριστικά της ίριδας του ματιού κ.α. Το σενάριο απειλής εδώ είναι ότι ένας εισβολέας θα αποπειραθεί να αναγνώσει τα εν λόγω χαρακτηριστικά χρησιμοποιώντας ένα κοντινό κρυφό αναγνώστη, αλλά δεν έχει πρόσβαση στο οπτικό τυπωμένο περιεχόμενο του διαβατηρίου. Ωστόσο το διαβατήριο περιέχει, μεταξύ άλλων μια τυπωμένη λωρίδα με πληροφορίες κατάλληλες προς ανάγνωση με σύστημα οπτικής αναγνώρισης χαρακτήρων (OCR) και κατά τον έλεγχο τοποθετείται πάνω σ' ένα τέτοιο σύστημα με τη σελίδα που περιέχει αυτές τις πληροφορίες ανοιχτή. Στη συνέχεια το νόμιμο σύστημα ελέγχου στέλνει στο RFID του διαβατηρίου επιλεγμένες πληροφορίες από αυτές που έχει μόλις διαβάσει, και τότε το RFID ξεκλειδώνει και στέλνει τις υπόλοιπες πληροφορίες στον αναγνώστη [Sta10]. Φυσικά εδώ μπορεί κανείς να σκεφτεί διάφορες επιθέσεις που βασίζονται στην προηγούμενη απόκτηση του τυπωμένου περιεχομένου του διαβατηρίου. Ένα εκ πρώτης όψεως απλό αντίμετρο θα ήταν να τοποθετηθεί ένα μεταλλικό φύλλο στα εξώφυλλα, αλλά και πάλι αυτό θα ήταν αποτελεσματικό όσο το διαβατήριο παραμένει σωστά κλειστό.

Γενικά προβλήματα στο IoT

Το διαδίκτυο των πραγμάτων (Internet of Things, IoT) οξύνει τους παλιούς ενώ φέρνει και νέους πονοκέφαλους γύρω από την ιδιωτικότητα για μια σειρά από λόγους [Cho15]:

- Ένα πλήθος από ρεύματα δεδομένων θα παράγονται από ένα πληθυσμό αισθητήρων. Τα ρεύματα αυτά μπορούν να καταγραφούν, να αναλυθούν συνδυαστικά, να υποκλαπούν κ.ο.κ.

- Οι συνηθισμένοι από το παρελθόν τρόποι ελέγχου της καταγραφής και παρακολούθησης δεν είναι διαθέσιμοι. Ο χρήστης δεν θα έχει τη δυνατότητα να επιλέξει ποιες λειτουργίες «τρέχουν» και να σταματήσει εντελώς μια υπηρεσία (π.χ. γεωγραφικού εντοπισμού) ή μια συσκευή (όπως σήμερα το κινητό ή τον υπολογιστή). Οι γνωστοί τρόποι άμυνας πιθανότατα θα είναι εν πολλοίς ανύπαρκτοι (π.χ. λογισμικό προστασίας κατά των ιών).

- Θα υπάρχει ένα πλήθος αισθητήρων που θα περνούν απαρατήρητοι, στην καλύτερη περίπτωση για να μην ενοχλούν. Το σημερινό μοντέλο της προειδοποίησης και αποδοχής (όπως π.χ. στην περίπτωση των cookies στην ΕΕ) θα είναι

πολύ δύσκολο να τεθεί σε εφαρμογή, αν μη τι άλλο γιατί ο χρήστης δεν θα μπορεί να ανταποκριθεί σε τόσες ειδοποιήσεις και να δώσει τις ανάλογες εγκρίσεις. Γενικότερα το μοντέλο αυτό είναι αντίθετο στην κλασσική ιδέα του pervasive computing, όπου ο χρήστης είναι διαφανώς βυθισμένος στο έξυπνο περιβάλλον. Μια ιδέα που έχει προταθεί, αλλά όχι και δοκιμασθεί επαρκώς, είναι η χρήση ρυθμίσεων ειδοποίησης, δηλαδή φίλτρων που θα επιτρέπουν στον χρήστη να επεμβαίνει μόνο στις περιπτώσεις που τον ενδιαφέρουν.

13.9 Συμπεράσματα

Με δυο λόγια η ιστορία της έννοιας της ιδιωτικότητας στα δίκτυα επικοινωνιών, της προστασίας της και της καταστρατήγησής της είναι σχετική με την ανάπτυξη μιας σειράς από τεχνολογίες επικοινωνίας, τεχνολογίες προστασίας και τεχνολογίες παρακολούθησης. Έχει να κάνει επίσης με τη βαθμιαία εμπέδωση μιας έννοιας ιδιωτικότητας στο βαθμό που την καθιστούν δυνατή κυρίως οι δύο πρώτες κατηγορίες τεχνολογιών. Παράλληλα χτίζεται κι ένα σχετικό νομικό πλαίσιο.

Αντίστροφα στην κατεύθυνση της μειωμένης ιδιωτικότητας οδηγούν (α) η συνεχής παρακολούθηση των συνηθειών του πελάτη από τον εμπορικό και βιομηχανικό τομέα, κυρίως μέσα από παρόχους μηχανών αναζήτησης, παρόχους υπηρεσιών αποθήκευσης δεδομένων και υπηρεσιών γενικότερα, κοινωνικά δίκτυα και άλλους μεσάζοντες του Internet, και (β) η πρόοδος στις τεχνολογίες εισβολής και παρακολούθησης και οι ανάγκες και πρακτικές των κρατικών υπηρεσιών εθνικής ασφάλειας, δημόσιας τάξης κ.λπ. Οι δύο αυτοί παίχτες συχνά συνεργάζονται μεταξύ τους και ανταλλάσσουν πληροφορίες [Sch16]. Στους εισβολείς μικρότερων δυνατοτήτων συγκαταλέγονται ιδιωτικοί ερευνητές (detectives), ακτιβιστές, περιστασιακοί ή επαγγελματίες χάκερς, καθώς και διάφοροι κακοποιοί. Ενδιάμεσες δυνατότητες φαίνεται ότι έχουν αποκτήσει επαγγελματίες που κινούνται στο χώρο της βιομηχανικής και επιχειρηματικής κατασκοπείας. Όλες αυτές οι ομάδες δεν είναι πάντοτε καλά διαχωρισμένες.

Βιβλιογραφία

[AA04] Dmitri Asonov and Rakesh Agrawal. "Keyboard acoustic emanations." *IEEE Symposium on Security and Privacy, 2004. Proceedings. 2004*. IEEE. 2004, pp. 3–11.

[Abr21] Lawrence Abrams. *Computer giant Acer hit by $50 million ransomware attack*. Bleeping Computer. 2021. URL: https://www.bleepingcomputer.com/news/security/computer-giant-acer-hit-by-50-million-ransomware-attack/ (visited on 03/19/2021).

[Adl88] Leonard M Adleman. "An abstract theory of computer viruses." *Conference on the Theory and Application of Cryptography*. Springer. 1988, pp. 354–374.

[Agg17] Charu C Aggarwal. *Outlier analysis*. 2nd. Springer, 2017.

[AJL13] Alessandro Acquisti, Leslie K John, and George Loewenstein. "What is privacy worth?": *The Journal of Legal Studies* 42.2 (2013), pp. 249–274.

[Aka13] *An Analysis of DrDOS SNMP/NTP/CHARGEN Reflection Attacks*. Prolexic-Akamai, 2013.

[AL09] Miltiades E. Anagnostou and Maria A. Lambrou. "Privacy now and in the age of ambient intelligence." *International Journal of Electronic Security and Digital Forensics* 2.4 (2009), pp. 355–364.

[Ame04] W. Ames. "Understanding spyware: risk and response." *IT Professional* 6.5 (2004), pp. 25–29.

[And08] Ross Anderson. *Security Engineering*. 2nd. Wiley, 2008.

[apec05] *APEC Privacy Framework*. Asia-Pacific Economic Cooperation, 2005.

[arb15] *Worldwide Infrastructure Security Report*. Vol. X. Arbor Networks, 2015.

[Arb20] *NETSCOUT Threat Intelligence Report*. Arbor Networks, 2021.

[AS13] Jonathan Anderson and Frank Stajano. "Must social networking conflict with privacy?": *IEEE Security & Privacy* 3 (2013), pp. 51–60.

[AS15] John M Abowd and Ian M Schmutte. "Economic Analysis and Statistical Disclosure Limitation." *Brookings Papers on Economic Activity* (2015).

[Aue02] M. Auerbach. *Automatic location of gunshots detected by mobile devices*. US Patent App. 09/755,529. Jan. 2002.

[AW11] Jason Andress and Steve Winterfeld. *Cyber Warfare - Techniques, Tactics and Tools for Security Practitioners*. Elsevier, 2011.

[AW18] Andreas M Antonopoulos and Gavin Wood. *Mastering ethereum: building smart contracts and dapps*. O'reilly Media, 2018.

[AW89] Nabil R Adam and John C Worthmann. "Security-control methods for statistical databases: a comparative study." *ACM Computing Surveys (CSUR)* 21.4 (1989), pp. 515–556.

[Ayc10] John Aycock. *Spyware and Adware*. Vol. 50. Springer Science & Business Media, 2010.

[Bac+12] Yoram Bachrach et al. "Personality and patterns of Facebook usage." *Proceedings of the 4th Annual ACM Web Science Conference*. ACM. 2012, pp. 24–32.

[Bal07] Murat Balaban. *IP Spoofing with BSD raw sockets interface*. 2007.

[Bam15] James Bamford. "A death in Athens." *The Intercept* (Sept. 29, 2015).

[Bar17] Elaine Barker. "Sp 800-67 rev. 2, recommendation for triple data encryption algorithm (tdea) block cipher." *NIST special publication* 800 (2017), p. 67.

[Bar20] Brian Barrett. "Thousands of websites distribute macOS malware: Shlayer Trojan family spreads via partner network of entertainment sites." *Wired* (2020).

[Bau+19] Mathieu Baudet et al. "State machine replication in the Libra Blockchain." *The Libra Assn., Tech. Rep* (2019).

[BBC20] BBC. "US cyber-attack: Russia 'clearly' behind SolarWinds operation, says Pompeo." (2020).

[BC93] Stefan Brands and David Chaum. "Distance-bounding protocols." *Advances in Cryptology—EUROCRYPT'93*. Springer. 1993, pp. 344–359.

[Bec+13] Georg T Becker et al. "Stealthy dopant-level hardware trojans." *International Conference on Cryptographic Hardware and Embedded Systems*. Springer. 2013, pp. 197–214.

[Bel11] Steven M Bellovin. "Frank Miller: Inventor of the one-time pad." *Cryptologia* 35.3 (2011), pp. 203–222.

[Ben+12] Boldizsár Bencsáth et al. "The cousins of stuxnet: Duqu, flame, and gauss." *Future Internet* 4.4 (2012), pp. 971–1003.

[Ber+06] Guido Bertoni et al. "RadioGatún, a belt-and-mill hash function." *IACR Cryptol. ePrint Arch.* 2006 (2006), p. 369.

[Ber+07] Guido Bertoni et al. "Sponge functions." *ECRYPT hash workshop*. Vol. 2007. 9. Citeseer. 2007.

[Ber+08] Daniel J Bernstein et al. "ChaCha, a variant of Salsa20." *Workshop record of SASC*. Vol. 8. 2008, pp. 3–5.

[Ber+09] Guido Bertoni et al. "The Road from Panama to Keccak via RadioGatún." *Symmetric Cryptography*. Ed. by Helena Handschuh et al. Dagstuhl Seminar Proceedings 09031. Dagstuhl, Germany: Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2009.

[Ber01a] Hal Berghel. "Digital Village: Caustic Cookies." *Commun. ACM* 44.5 (May 2001), pp. 19–22.

[Ber01b] Hal Berghel. "The code red worm." *Communications of the ACM* 44.12 (2001), pp. 15–19.

[BH08] Pierre-Marc Bureau and David Harley. "A dose by any other name." *Virus Bulletin Conference, VB*. Vol. 8. 2008, pp. 224–231.

[Bin20] Christopher Bing. "REFILE-EXCLUSIVE-U.S. Treasury breached by hackers backed by foreign government - sources." *Reuters* (2020).

[Bla+00] J.A. Black et al. *A Concise Dictionary of Akkadian*. Santag Arbeiten und Untersuchungen zur Keilschriftkunde. Harrassowitz, 2000.

[BN09] Ross P Buckley and Justen Nixon. "The role of reputation in banking." *Published in the Journal of Banking and Finance Law and Practice* 20 (2009), pp. 37–50.

[Bob+72] Daniel G Bobrow et al. "TENEX, a paged time sharing system for the PDP-10." *Communications of the ACM* 15.3 (1972), pp. 135–143.

[BR93] Mihir Bellare and Phillip Rogaway. "Random oracles are practical: A paradigm for designing efficient protocols." *Proceedings of the 1st ACM conference on Computer and communications security*. 1993, pp. 62–73.

[Bry13] A.J. Bryant. *Sekigahara 1600: The final struggle for power*. Campaign. Bloomsbury Publishing, 2013.

[Bud21] Nikola Budanović. "What's On the Other Side of Your Inbox – 20 SPAM Statistics for 2021." *DataProt* (2021).

[Bur08] Kelly Burton. *The Conficker Worm*. SANS. 2008. URL: `https://www.sans.org/security-resources/malwarefaq/conficker-worm`.

[CCF20] Kaylash C Chaudhary, Vishal Chand, and Ansgar Fehnker. "Double-spending analysis of bitcoin." *PACIS 2020 Proceedings* (2020).

[CD08] Kenneth L. Calvert and Michael J. Donahoo. *TCP/IP Sockets in Java - Practical Guide for Programmers*. 2nd. Elsevier, 2008.

[CDB09] Joan Calvet, Carlton R Davis, and Pierre-Marc Bureau. "Malware authors don't learn, and that's good!": *2009 4th International Conference on Malicious and Unwanted Software (MALWARE)*. IEEE. 2009, pp. 88–97.

[CGH04] Ran Canetti, Oded Goldreich, and Shai Halevi. "The random oracle methodology, revisited." *Journal of the ACM (JACM)* 51.4 (2004), pp. 557–594.

[CGM13] Omar Choudary, Felix Grobert, and Joachim Metz. "Security analysis and decryption of filevault 2." *IFIP International Conference on Digital Forensics*. Springer. 2013, pp. 349–363.

[Cha+12] Shu-jen Chang et al. "Third-round report of the SHA-3 cryptographic hash algorithm competition." *NIST Interagency Report* 7896 (2012), p. 121.

[Cha83] David Chaum. "Blind signatures for untraceable payments." *Advances in cryptology*. Springer. 1983, pp. 199–203.

[Che+19] Djabir Abdeldjalil Chekired et al. "5G-slicing-enabled scalable SDN core network: Toward an ultra-low latency of autonomous driving service." *IEEE Journal on Selected Areas in Communications* 37.8 (2019), pp. 1769–1782.

[Cho15] Richard Chow. "IoT Privacy: Can We Regain Control?": *Proceedings of the 3rd ACM Workshop on Information Hiding and Multimedia Security*. IH&MMSec '15. Portland, Oregon, USA: ACM, 2015, pp. 3–3.

[CJB99] Balakrishnan Chandrasekaran, John R Josephson, and V Richard Benjamins. "What are ontologies, and why do we need them?": *IEEE Intelligent Systems and their applications* 14.1 (1999), pp. 20–26.

[CL02] Miguel Castro and Barbara Liskov. "Practical Byzantine fault tolerance and proactive recovery." *ACM Transactions on Computer Systems (TOCS)* 20.4 (2002), pp. 398–461.

[Cla+05] Christopher Clark et al. "Live migration of virtual machines." *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2*. 2005, pp. 273–286.

[Cla99] Roger Clarke. "Internet privacy concerns confirm the case for intervention." *Communications of the ACM* 42.2 (1999), pp. 60–67.

[CO81] Francis Y. Chin and Gultekin Ozsoyoglu. "Statistical Database Design." *ACM Trans. Database Syst.* 6.1 (Mar. 1981), pp. 113–139.

[Coh87] Fred Cohen. "Computer Viruses." *Comput. Secur.* 6.1 (Feb. 1987), pp. 22–35.

[Coh89] Fred Cohen. "Computational aspects of computer viruses." *Computers and Security* 8.4 (1989), pp. 297–298.

[CR09] Marco Cremonini and Marco Riccardi. "The Dorothy project: an open botnet analysis framework for automatic tracking and activity visualization." *Computer Network Defense (EC2ND), 2009 European Conference on*. IEEE. 2009, pp. 52–54.

[CR15] J. A. Chaudhry and R. G. Rittenhouse. "Phishing: Classification and Countermeasures." *2015 7th International Conference on Multimedia, Computer Graphics and Broadcasting (MulGraB)*. 2015, pp. 28–31.

[CT12] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[CV17] Christian Cachin and Marko Vukolić. "Blockchain consensus protocols in the wild." *arXiv preprint arXiv:1707.01873* (2017).

[Dad+19] Emmanuel Gbenga Dada et al. "Machine learning for email spam filtering: review, approaches and open research problems." *Heliyon* 5.6 (2019), e01802.

[Dan+12] Quynh Dang et al. "NIST Special Publication 800-107 Revision 1 Recommendation for Applications Using Approved Hash Algorithms." (2012).

[DC98] Joan Daemen and Craig Clapp. "Fast hashing and stream encryption with PANAMA." *International Workshop on Fast Software Encryption*. Springer. 1998, pp. 60–74.

[DD79] Dorothy E. Denning and Peter J. Denning. "The Tracker: A Threat to Statistical Database Security." *ACM Trans. Database Syst.* 4.1 (Mar. 1979). Ed. by Mayer D. Schwartz, pp. 76–96.

[DH76] Whitfield Diffie and Martin Hellman. "New directions in cryptography." *IEEE transactions on Information Theory* 22.6 (1976), pp. 644–654.

[DHH15] H. Dreggs, T. Hobson, and X. Hollenbeck. *Cyber Attack Threat Trends: Stuxnet*. Lulu.com, 2015.

[Dia12] Jared Diamond. *The World Until Yesterday: What Can We Learn from Traditional Societies?* 1st ed. Viking Adult, 2012.

[Dis08] Dennis Distler. *Performing Egress Filtering*. Sans Institute, Aug. 2008.

[Div01] CERT Division. "Interbase Server Contains Compiled-in Back Door Account." *Software Engineering Institute - Carnegie Mellon Univ.* (2001).

[Dob98] Hans Dobbertin. "Cryptanalysis of MD4." *Journal of Cryptology* 11.4 (1998), pp. 253–271.

[Dwo] Morris J Dworkin. "FIPS PUB 202 SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions." *NIST PUB Ser* 202 ().

[Dwo+10] Morris J Dworkin et al. "Recommendation for block cipher modes of operation: The XTS-AES mode for confidentiality on storage devices." (2010).

[Dwo15] Morris J Dworkin. *SHA-3 standard: Permutation-based hash and extendable-output functions*. Tech. rep. 2015.

[Edu+16] Sergey Edunov et al. "Three and a half degrees of separation." *Research at Facebook* (Feb. 2016).

[Eli+16] C.C. Elisan et al. *Hacking Exposed Malware & Rootkits: Security Secrets and Solutions, Second Edition*. Hacking Exposed. McGraw-Hill Education, 2016.

[FAT05] Lei Fang, Panos J Antsaklis, and Anastasis Tzimas. "Asynchronous consensus protocols: Preliminary results, simulations and open questions." *Proceedings of the 44th IEEE Conference on Decision and Control*. IEEE. 2005, pp. 2194–2199.

[Fei+03] Laura Feinstein et al. "Statistical approaches to DDoS attack detection and response." *DARPA Information Survivability Conference and Exposition, 2003. Proceedings*. Vol. 1. IEEE. 2003, pp. 303–314.

[Fei73] Horst Feistel. "Cryptography and computer privacy." *Scientific american* 228.5 (1973), pp. 15–23.

[Fil06] Éric Filiol. *Computer Viruses: from theory to applications*. Collection IRIS. Springer Paris, 2006.

[FIP99] PUB FIPS. "46-3." *Data encryption standard (DES)* 25 (1999).

[FMF14] Romain Fontugne, Johan Mazel, and Kensuke Fukuda. "Hashdoop: A MapReduce framework for network anomaly detection." *2014 IEEE conference on computer communications workshops (INFOCOM WKSHPS)*. IEEE. 2014, pp. 494–499.

[FS11] K.R. Fall and W.R. Stevens. *TCP/IP Illustrated*. Addison-Wesley professional computing series v. 1. Addison-Wesley, 2011.

[FSM93] Frank Feather, Dan Siewiorek, and Roy Maxion. "Fault detection in an ethernet network using anomaly signature matching." *ACM SIGCOMM Computer Communication Review* 23.4 (1993), pp. 279–288.

[Gal94] Peter Galison. "The ontology of the enemy: Norbert Wiener and the cybernetic vision." *Critical inquiry* 21.1 (1994), pp. 228–266.

[Gao+19] J. Gao et al. "Should You Consider Adware as Malware in Your Study?": *2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. 2019, pp. 604–608.

[Gel16] Tom Geller. "In Privacy Law, It's the U.S. Vs. the World." *Commun. ACM* 59.2 (Jan. 2016), pp. 21–23.

[Gen09] Craig Gentry. "Fully homomorphic encryption using ideal lattices." *Proceedings of the forty-first annual ACM symposium on Theory of computing*. 2009, pp. 169–178.

[Geo16] Northern District of Georgia. *Two Major International Hackers Who Developed the "SpyEye" Malware get over 24 Years Combined in Federal Prison*. Department of Justice. 2016. URL: https://www.justice.gov/usao-ndga/pr/two-major-international-hackers-who-developed-spyeye-malware-get-over-24-years-combined (visited on 04/20/2016).

[GK17] Hannes Grassegger and Mikael Krogerus. "The data that turned the world upside down." *Vice Magazine, January* 30 (2017).

[Goh08] Nancy Gohring. "Woman Gets Two Years for Aiding Nigerian Internet Check Scam." *PCWorld* (June 25, 2008).

[Gor09] Walter Goralski. *The Illustrated Network: How TCP/IP Works in a Modern Network*. Morgan Kaufmann series in networking. Elsevier Science, 2009.

[Gré+15] André Ricardo Abed Grégio et al. "Toward a taxonomy of malware behaviors." *The Computer Journal* 58.10 (2015), pp. 2758–2777.

[Gri13] David Alan Grier. *When computers were human*. Princeton University Press, 2013.

[GSB02] Hans W. Gellersen, Albercht Schmidt, and Michael Beigl. "Multi-sensor Context-awareness in Mobile Devices and Smart Artifacts." *Mob. Netw. Appl.* 7.5 (Oct. 2002), pp. 341–351.

[GSB05] Vijay K Gurbani, Xian-He Sun, and Alec Brusilovsky. "Inhibitors for ubiquitous deployment of services in the next-generation network." *Communications Magazine, IEEE* 43.9 (2005), pp. 116–121.

[Gu+07] Guofei Gu et al. "BotHunter: Detecting Malware Infection Through IDS-Driven Dialog Correlation." *Proceedings of the 16th USENIX Security Symposium*. 2007, pp. 167–182.

[GZL08] Guofei Gu, Junjie Zhang, and Wenke Lee. "BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic." *Proceedings of the 15th Annual Network and Distributed System Security Symposium*. 2008.

[Hal05] Fred Halsall. *Computer Networking and the Internet*. 5th. Pearson Education, Limited, 2005.

[Hea06] John Heasman. "Rootkit threats." *Network Security* 2006.1 (2006), pp. 18–19.

[HEF09] Thorsten Holz, Markus Engelberth, and Felix Freiling. "Learning more about the underground economy: A case-study of keyloggers and dropzones." *European Symposium on Research in Computer Security*. Springer. 2009, pp. 1–18.

[Her12] Cormac Herley. "Why do nigerian scammers say they are from nigeria?": *WEIS*. 2012.

[HG15] Kyle Hannah and Steven Gianvecchio. "Zeuslite: A Tool for Botnet Analysis in the Classroom." *J. Comput. Sci. Coll.* 30.3 (Jan. 2015), pp. 109–116.

[Hil16] Michael Hill. "Toy Giant Mattel Narrowly Escapes Phishing Scam." *Infosecurity Magazine* (2016).

[Hod19] C.J. Hodson. *Cyber Risk Management: Prioritize Threats, Identify Vulnerabilities and Apply Controls*. Kogan Page, 2019.

[Hoo78] R. Hooke. *Lectures de Potentia Restitutiva, Or of Spring Explaining the Power of Springing Bodies*. [Cutlerian lecture. John Martyn, 1678.

[How09] Rick Howard. *Cyber fraud: tactics, techniques and procedures*. CRC press, 2009.

[Hsu+17] Fu-Hau Hsu et al. "Detecting web-based botnets using bot communication traffic features." *Security and Communication Networks* 2017 (2017).

[Hum+20] Mamoona Humayun et al. "Internet of things and ransomware: evolution, mitigation and prevention." *Egyptian Informatics Journal* (2020).

[Jac+13] Appelbaum Jacob et al. "Documents Reveal Top NSA Hacking Unit." *Der Spiegel* (Dec. 29, 2013).

[Jac05] J.E. Jackson. *A User's Guide to Principal Components*. Wiley Series in Probability and Statistics. Wiley, 2005.

[JG11] Wayne Jansen and Timothy Grance. "Guidelines on security and privacy in public cloud computing." *NIST Special Publication 800-144* (Dec. 2011).

[Jol06] I.T. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer New York, 2006.

[Jol90] Ian T Jolliffe. "Principal component analysis: a beginner's guide—I. Introduction and application." *Weather* 45.10 (1990), pp. 375–382.

[JRS03] Ari Juels, Ronald L Rivest, and Michael Szydlo. "The blocker tag: Selective blocking of RFID tags for consumer privacy." *Proceedings of the 10th ACM conference on Computer and communications security*. ACM. 2003, pp. 103–111.

[Kal92] Burt Kaliski. *RFC1319: The MD2 Message-Digest Algorithm*. 1992.

[Kan05] Srikanth Kandula. "Surviving DDoS Attacks." *USENIX ;login:* 30.5 (Oct. 2005), pp. 35–39.

[Kar+19] Asif Karim et al. "A comprehensive survey for intelligent spam email detection." *IEEE Access* 7 (2019), pp. 168261–168295.

[Kas19] Kaspersky. *Threats to macOS users*. Kaspersky. 2019. URL: `https://securelist.com/threats-to-macos-users/93116/` (visited on 09/11/2019).

[Kas20] Kaspersky. *Thousands of websites distribute macOS malware: Shlayer Trojan family spreads via partner network of entertainment sites*. Kaspersky. 2020. URL: `https://usa.kaspersky.com/about/press-releases/2020_thousands-of-websites-distribute-macos-malware` (visited on 01/23/2020).

[Kasp15] Kaspersky Lab. *Statistics on botnet assisted DDoS attacks in Q1 2015*. May 2015.

[Kasp15b] Kaspersky Lab. *A single DDoS attack can cost a company more than $400,000*. Jan. 2015.

[KC06] Samuel T King and Peter M Chen. "SubVirt: Implementing malware with virtual machines." *2006 IEEE Symposium on Security and Privacy (S&P'06)*. IEEE. 2006, 14–pp.

[Kin13] Sunny King. "Primecoin: Cryptocurrency with prime number proof-of-work." *July 7th* 1.6 (2013).

[Kir10] Jeremy Kirk. "Rustock botnet responsible for 40 percent of spam." *GoodGear-Guide* (2010).

[KKJ16] Deneesha Kansagra, Malaram Kumhar, and Dhaval Jha. "Ransomware: a threat to cyber security." *CS Journals* 7.1 (2016).

[Kon07] Alan G Konheim. *Computer security and cryptography*. John Wiley & Sons, 2007.

[Koz05] Charles Kozierok. *The TCP/IP Guide: A Comprehensive, Illustrated Internet Protocols Reference*. No Starch Press, 2005.

[Kre18] Brian Krebs. *Mirai Co-Author Gets 6 Months Confinement, $8.6M in Fines for Rutgers Attacks*. 2018. URL: `https://krebsonsecurity.com/2018/10/mirai-co-author-gets-6-months-confinement-8-6m-in-fines-for-rutgers-attacks/` (visited on 10/26/2018).

[Kri01] David M. Kristol. "HTTP Cookies: Standards, Privacy, and Politics." *ACM Trans. Internet Technol.* 1.2 (Nov. 2001), pp. 151–198.

[KS18] Ehsan Khoshhalpour and Hamid Reza Shahriari. "BotRevealer: Behavioral Detection of Botnets based on Botnet Life-cycle." *ISeCure* 10.1 (2018).

[KSG13] Michal Kosinski, David Stillwell, and Thore Graepel. "Private traits and attributes are predictable from digital records of human behavior." *Proceedings of the National Academy of Sciences* 110.15 (2013), pp. 5802–5805.

[Kus13] D. Kushner. "The real story of stuxnet." *IEEE Spectrum* 50.3 (2013), pp. 48–53.

[KVA12] Denis Kozlov, Jari Veijalainen, and Yasir Ali. "Security and Privacy Threats in IoT Architectures." *Proceedings of the 7th International Conference on Body Area Networks*. BodyNets '12. Oslo, Norway: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2012, pp. 256–262.

[KVG13] Miltiadis Kandias, Nikos Virvilis, and Dimitris Gritzalis. "The Insider Threat in Cloud Computing." *Critical Information Infrastructure Security*. Ed. by Sandro Bologna et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 93–103.

[Kwo+15] Jonghoon Kwon et al. "An incrementally deployable anti-spoofing mechanism for software-defined networks." *Computer Communications* 64 (2015), pp. 1–20.

[Lal18] Marc Laliberte. "IoT Botnets Are Evolving – How Big Can They Get?": *Secplicity* (2018).

[Lan+94] Carl E Landwehr et al. "A taxonomy of computer program security flaws." *ACM Computing Surveys (CSUR)* 26.3 (1994), pp. 211–254.

[Lan11] R. Langner. "Stuxnet: Dissecting a Cyberwarfare Weapon." *IEEE Security Privacy* 9.3 (2011), pp. 49–51.

[Lan16] Carl Landwehr. "Privacy Research Directions." *Commun. ACM* 59.2 (Jan. 2016), pp. 29–31.

[LaP96] Leonard J LaPadula. "About the Bell-LaPadula model." *Journal of Computer Security* 4 (1996), pp. 233–238.

[LCD04] Anukool Lakhina, Mark Crovella, and Christophe Diot. "Diagnosing network-wide traffic anomalies." *ACM SIGCOMM computer communication review* 34.4 (2004), pp. 219–230.

[Lem01] Robert Lemos. *Code Red for security*. CNET News. 2001. URL: `https://www.cnet.com/news/code-red-for-security/` (visited on 07/27/2001).

[Lem02] *Resisting SYN flood DoS attacks with a SYN cache*. 2002, pp. 89–98.

[Leo19] Megan Leonhardt. "'Nigerian prince' email scams still rake in over $700,000 a year—here's how to protect yourself." *makeit* (2019).

[Lia+16] Kevin Liao et al. "Behind closed doors: measurement and analysis of CryptoLocker ransoms in Bitcoin." *2016 APWG Symposium on Electronic Crime Research (eCrime)*. IEEE. 2016, pp. 1–13.

[Lit02] David Litchfield. "Non-stack Based Exploitation of Buffer Overrun Vulnerabilities on Windows NT/2000/XP." *NGSSoftware Insight Security Research* 5 (2002).

[Lit16] C. Lita. "On Complexity of the Detection Problem for Bounded Length Polymorphic Viruses." *2016 18th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*. 2016, pp. 371–378.

[Liu+11] Fang Liu et al. "NIST cloud computing reference architecture." *NIST Special Publication 500-292* (2011).

[LM+10] Κωνσταντίνος Λαμπρινουδάκης κ.α. *Προστασία της ιδιωτικότητας & Τεχνολογίες Πληροφορικής και Επικοινωνιών*. Παπασωτηρίου, 2010.

[LRS14] S Latha, K Raju, and S Santhi. "Overview of dropbox encryption in cloud computing." *Transactions on Engineering and Sciences* 2.3 (2014), pp. 27–32.

[LRW02] Moses Liskov, Ronald L Rivest, and David Wagner. "Tweakable block ciphers." *Annual International Cryptology Conference*. Springer. 2002, pp. 31–46.

[LSP82] Leslie Lamport, Robert Shostak, and Marshall Pease. "The Byzantine generals problem." *ACM Transactions on Programming Languages and Systems* 4.3 (July 1982), pp. 382–401.

[LSS96] Laurie Law, Susan Sabett, and Jerry Solinas. "How to make a mint: the cryptography of anonymous electronic cash." *Am. UL Rev.* 46 (1996), p. 1131.

[LWD09] A. T. Lawniczak, H. Wu, and B. N. Di Stefano. "Detection of anomalous packet traffic via entropy." *2009 Canadian Conference on Electrical and Computer Engineering*. 2009, pp. 137–141.

[Mac+07] Ashwin Machanavajjhala et al. "L-diversity: Privacy Beyond K - anonymity." *ACM Trans. Knowl. Discov. Data* 1.1 (Mar. 2007).

[Mac18] J. MacCormick. *What Can Be Computed?: A Practical Guide to the Theory of Computation*. Princeton University Press, 2018.

[Mar10] Luther Martin. "XTS: A mode of AES for encrypting hard disks." *IEEE Security & Privacy* 8.3 (2010), pp. 68–69.

[MD16] Santosh Kumar Majhi and Sunil Kumar Dhal. "A study on security vulnerability on cloud platforms." *Procedia Computer Science* 78 (2016), pp. 55–60.

[Mer78] Ralph C Merkle. "Secure communications over insecure channels." *Communications of the ACM* 21.4 (1978), pp. 294–299.

[Mer79] Ralph Charles Merkle. *Secrecy, authentication, and public key systems.* Stanford University, 1979.

[MFD19] A. Milolidakis, R. Fontugne, and X. Dimitropoulos. "Detecting Network Disruptions At Colocation Facilities." *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*. 2019, pp. 2161–2169.

[MG+11] Peter Mell, Tim Grance, et al. "The NIST definition of cloud computing." *NIST Special Publication 800-145* (Sept. 2011).

[MH11] Gregory J. Matthews and Ofer Harel. "Data confidentiality: A review of methods for statistical disclosure limitation and methods for assessing privacy." *Statist. Surv.* 5 (2011), pp. 1–29.

[Moo+03] David Moore et al. "Inside the slammer worm." *IEEE Security & Privacy* 1.4 (2003), pp. 33–39.

[Moo+06] David Moore et al. "Inferring internet denial-of-service activity." *ACM Transactions on Computer Systems (TOCS)* 24.2 (2006), pp. 115–139.

[MR04] Jelena Mirkovic and Peter Reiher. "A taxonomy of DDoS attack and DDoS defense mechanisms." *ACM SIGCOMM Computer Communication Review* 34.2 (2004), pp. 39–53.

[MS05] Douglas Mauro and Kevin Schmidt. *Essential SNMP*. 2nd. O'Reilly, 2005.

[MSK12] Stuart McClure, Joel Scambray, and George Kurtz. *Hacking Exposed 7: Network Security Secrets and Solutions*. Hacking Exposed. McGraw-Hill Education, 2012.

[Mul04] Frédéric Muller. "The MD2 hash function is not one-way." *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2004, pp. 214–229.

[Nak08] Satoshi Nakamoto. *Bitcoin: A peer-to-peer electronic cash system*. Tech. rep. 2008.

[Nam09] Yuri Namestnikov. *The economics of Botnets*. July 2009.

[Nar+16] Arvind Narayanan et al. *Bitcoin and cryptocurrency technologies: a comprehensive introduction*. Princeton University Press, 2016.

[Nay+15] Rajkishore Nayak et al. "RFID in textile and clothing manufacturing: technology and challenges." *Fashion and Textiles* 2.1 (2015), pp. 1–16.

[Nec+01] James Nechvatal et al. "Report on the development of the Advanced Encryption Standard (AES)." *Journal of Research of the National Institute of Standards and Technology* 106.3 (2001), p. 511.

[Noy10] Katherine Noyes. "Why Linux Is More Secure Than Windows." *PCWorld* (Aug. 3, 2010).

[NPK10] M. N, A. Parmar, and M. Kumar. "A flow based anomaly detection system using chi-square technique." *2010 IEEE 2nd International Advance Computing Conference (IACC)*. 2010, pp. 285–289.

[NS10] Arvind Narayanan and Vitaly Shmatikov. "Myths and Fallacies of "Personally Identifiable Information"." *Commun. ACM* 53.6 (June 2010), pp. 24–26.

[OA00] Robert J. Orr and Gregory D. Abowd. "The smart floor: a mechanism for natural user identification and tracking." *CHI'00 extended abstracts on Human factors in computing systems*. ACM. 2000, pp. 275–276.

[OCJ08] Jon Oberheide, Evan Cooke, and Farnam Jahanian. "Empirical exploitation of live virtual machine migration." *Proc. of BlackHat DC convention*. Citeseer. 2008, pp. 1–6.

[Oll09] Gunter Ollmann. "Botnet communication topologies." *Retrieved September* 30 (2009), p. 2009.

[OM04] Reza Olfati-Saber and Richard M Murray. "Consensus problems in networks of agents with switching topology and time-delays." *IEEE Transactions on automatic control* 49.9 (2004), pp. 1520–1533.

[Opp06] Max Stul Oppenheimer. "Internet Cookies: When Is Permission Consent." *Neb. L. Rev.* 85 (2006), p. 383.

[Owa20] Amer Owaida. "Small and medium-sized businesses: Big targets for ransomware attacks." *welivesecurity* (2020).

[Pap+00] Symeon Papavassiliou et al. "Implementing enhanced network maintenance for transaction access services: Tools and applications." *2000 IEEE International Conference on Communications. ICC 2000. Global Convergence Through Communications. Conference Record*. Vol. 1. IEEE. 2000, pp. 211–215.

[Pap18] Vasileios Papageorgiou. "The Greek wiretapping scandal and the false promise of intelligence cooperation in the information era." *KEDISA-Center for International Strategic Analyses* (2018).

[Pec12] M Peck. "Bitcoin: The Cryptoanarchists' Answer to Cash-How Bitcoin brought privacy to electronic transactions." *IEEE Spectrum* (2012).

[Pie04] Jason Pierce. *Egress Filtering for a Better Internet*. Sans Institute, Sept. 2004.

[PM14] Sudhir Kumar Pandey and BM Mehtre. "A lifecycle based approach for malware analysis." *2014 Fourth International Conference on Communication Systems and Network Technologies*. IEEE. 2014, pp. 767–771.

[PS07] Vassilis Prevelakis and Diomidis Spinellis. "The athens affair." *Spectrum, IEEE* 44.7 (2007), pp. 26–33.

[PS85] James L Peterson and Abraham Silberschatz. *Operating system concepts*. Addison-Wesley Longman Publishing Co., Inc., 1985.

[PSL80] Marshall Pease, Robert Shostak, and Leslie Lamport. "Reaching agreement in the presence of faults." *Journal of the ACM (JACM)* 27.2 (1980), pp. 228–234.

[Puj20] Guy Pujolle. *Software Networks: Virtualization, SDN, 5G, and Security*. John Wiley & Sons, 2020.

[QKC19] Attia Qamar, Ahmad Karim, and Victor Chang. "Mobile malware attacks: Review, taxonomy & future directions." *Future Generation Computer Systems* 97 (2019), pp. 887–909.

[QQ10] Anderson A. L. Queiroz and Ruy J. G. B. de Queiroz. "Breach of Internet Privacy Through the Use of Cookies." *Proceedings of the 3rd International Conference on PErvasive Technologies Related to Assistive Environments*. PETRA '10. Samos, Greece: ACM, 2010, 71:1–71:5.

[RAD+78] Ronald L Rivest, Len Adleman, Michael L Dertouzos, et al. "On data banks and privacy homomorphisms." *Foundations of secure computation* 4.11 (1978), pp. 169–180.

[Ray] Edward Ray. *Malware FAQ: MS-SQL Slammer*. SANS. URL: `https://www.sans.org/security-resources/malwarefaq/ms-sql-exploit`.

[RC95] N Rogier and Pascal Chauvaud. "The compression function of MD2 is not collision free." *Selected Areas in Cryptography*. Vol. 95. 1995, pp. 18–19.

[RC97] N Rogier and Pascal Chauvaud. "MD2 is not secure without the checksum byte." *Designs, Codes and Cryptography* 12.3 (1997), pp. 245–251.

[Reg+09] Andrew Regenscheid et al. *Status report on the first round of the SHA-3 cryptographic hash algorithm competition*. US Department of Commerce, National Institute of Standards and Technology, 2009.

[Ren05] R Renner. "Security of quantum key distribution." *PhD Thesis, Swiss Federal Institute of Technology* (2005).

[RFC 1191] J. Mogul and S. Deering. *Path MTU Discovery (RFC 1191)*. Nov. 1990.

[RFC 3013] T. Killalea. *Recommended Internet Service Provider Security Services and Procedures (RFC 3013)*. Nov. 2000.

[RFC 4987] W. Eddy. *TCP SYN Flooding Attacks and Common Mitigations (RFC 4987)*. Aug. 2007.

[RFC 5905] D. Mills et al. *Network Time Protocol Version 4: Protocol and Algorithms Specification (RFC 5905)*. June 2010.

[RFC 5927] F. Gont. *ICMP Attacks against TCP (RFC 5927)*. Aug. 2007.

[RFC 6959] D. McPherson, F. Baker, and J. Halpern. *Source Address Validation Improvement (SAVI) Threat Scope (RFC 6959)*. May 2013.

[Ric+13] Marco Riccardi et al. "Titans' revenge: Detecting Zeus via its own flaws." *Computer Networks* 57.2 (2013), pp. 422–435.

[Rit02] Rudolf J Ritter. *Das Fernmeldematerial der Schweizerischen Armee seit 1875. Folge 10: Codes und Chiffrierverfahren*. 2002.

[Riv+08] Ronald L Rivest et al. "The MD6 hash function." *Invited talk at CRYPTO* (2008).

[RMG13] Rafael A. Rodríguez-Gómez, Gabriel Maciá-Fernández, and Pedro García-Teodoro. "Survey and Taxonomy of Botnet Research through Life-Cycle." *ACM Comput. Surv.* 45.4 (Aug. 2013).

[Rog04] Phillip Rogaway. "Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC." *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2004, pp. 16–31.

[Ros07] David Rosenblum. "What anyone can know: The privacy risks of social networking sites." *IEEE Security & Privacy* 3 (2007), pp. 40–49.

[RSA78] Ronald L Rivest, Adi Shamir, and Leonard Adleman. "A method for obtaining digital signatures and public-key cryptosystems." *Communications of the ACM* 21.2 (1978), pp. 120–126.

[Rut04] Joanna Rutkowska. "Redpill: Detect vmm using (almost) one cpu instruction." *http://invisiblethings. org/papers/redpill. html* (2004).

[Sam12] John Sammons. *The basics of digital forensics: the primer for getting started in digital forensics*. Elsevier, 2012.

[Sat18] Jason Sattler. *Russian National Who Operated Kelihos Botnet Pleads Guilty to Fraud, Conspiracy, Computer Crime and Identity Theft Offenses*. Department of Justice, District of Connecticut. 2018. URL: https://www.justice.gov/usao-ct/pr/russian-national-who-operated-kelihos-botnet-pleads-guilty-fraud-conspiracy-computer (visited on 09/12/2018).

[Sat19] Jason Sattler. *What we've learned from 10 years of the Conficker mystery*. F-Secure. 2019. URL: https://blog.f-secure.com/what-weve-learned-from-10-years-of-the-conficker-mystery/.

[SB18] W Stallings and L Brown. *Computer Security: Principles and Practice . Harlow*. 2018.

[SBK16] Basant Subba, Santosh Biswas, and Sushanta Karmakar. "False alarm reduction in signature-based IDS: game theory approach." *Security and Communication Networks* 9.18 (2016), pp. 4863–4881.

[SBL20] Federico Sierra-Arriaga, Rodrigo Branco, and Ben Lee. "Security issues and challenges for virtualization technologies." *ACM Computing Surveys (CSUR)* 53.2 (2020), pp. 1–37.

[Sch16] Bruce Schneier. *Data and Goliath: The Hidden Battles to Collect Your Data and Control Your World*. W. W. Norton, 2016.

[Sea20] Tara Seals. *Shlayer, No. 1 Threat for Mac, Targets YouTube, Wikipedia*. threatpost. 2020. URL: https://threatpost.com/shlayer-mac-youtube-wikipedia/152146/ (visited on 01/23/2020).

[Sea21] Tara Seals. *Microsoft Exchange Servers Face APT Attack Tsunami*. threatpost. 2021. URL: https://threatpost.com/microsoft-exchange-servers-apt-attack/164695/ (visited on 03/11/2021).

[Sev16] Charles Severance. "Bruce Schneier: The Security Mindset." *IEEE Computer* 49.2 (Feb. 2016), pp. 7–8.

[SGG18] Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne. *Operating system concepts*. Wiley, 2018.

[SH09] Karen Scarfone and Paul Hoffman. *NIST Special Publication 800-41 Revision 1 - Guidelines on Firewalls and Firewall Policy*. National Institute of Standards and Technology (NIST), US Department of Commerce, Sept. 2009.

[Sha49] Claude E Shannon. "Communication theory of secrecy systems." *The Bell system technical journal* 28.4 (1949), pp. 656–715.

[Shm+10] Erez Shmueli et al. "Database encryption: an overview of contemporary challenges and design considerations." *ACM SIGMOD Record* 38.3 (2010), pp. 29–34.

[SK15] Prabhsimran Singh and Kuljit Kaur. "Database security using encryption." *2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE)*. IEEE. 2015, pp. 353–358.

[SKG19] Christian Stoll, Lena Klaassen, and Ulrich Gallersdœrfer. "The Carbon Footprint of Bitcoin." *Joule* 3.7 (2019), pp. 1647–1661.

[SLD07] Marc Stevens, Arjen Lenstra, and Benne De Weger. "Chosen-prefix collisions for MD5 and colliding X. 509 certificates for different identities." *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2007, pp. 1–22.

[Smi09] Andrew Smith. "Nigerian scam e-mails and the charms of capital." *Cultural studies* 23.1 (2009), pp. 27–47.

[Sno15] Chris Snow. "Massive Distributed Reflection Denial of Service (DrDoS) DoSNETs for hire – NTP, Chargen, SNMP, SSDP." *Articles Factory* (Aug. 2015).

[Sot+08] Alexander Sotirov et al. "MD5 considered harmful today, creating a rogue CA certificate." *25th Annual Chaos Communication Congress*. CONF. 2008.

[SP18] Douglas Robert Stinson and Maura Paterson. *Cryptography: theory and practice*. CRC press, 2018.

[Spi03] D. Spinellis. "Reliable identification of bounded-length viruses is NP-complete." *IEEE Transactions on Information Theory* 49.1 (2003), pp. 280–284.

[SS98] Pierangela Samarati and Latanya Sweeney. "Generalizing data to provide anonymity when disclosing information." *PODS*. Vol. 98. 1998, p. 188.

[SSS17] Lakshmi Siva Sankar, M Sindhu, and M Sethumadhavan. "Survey of consensus protocols on blockchain applications." *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*. IEEE. 2017, pp. 1–5.

[Sta03] Vince Stanford. "Pervasive computing goes the last hundred feet with RFID systems." *Pervasive Computing, IEEE* 2.2 (2003), pp. 9–14.

[Sta10] Frank Stajano. "Security Issues in Ubiquitous Computing*." *Handbook of Ambient Intelligence and Smart Environments*. Springer, 2010, pp. 281–314.

[Sta18] W. Stallings. *Operating Systems: Internals and Design Principles*. Pearson, 2018.

[Sta95] Secure Hash Standard. "FIPS Pub 180-1." *National Institute of Standards and Technology* 17 (1995), p. 15.

[Ste+17] Marc Stevens et al. "The First Collision for Full SHA-1." *Advances in Cryptology – CRYPTO 2017*. Ed. by Jonathan Katz and Hovav Shacham. Cham: Springer International Publishing, 2017, pp. 570–596.

[Sto+09] Brett Stone-Gross et al. "Your botnet is my botnet: analysis of a botnet takeover." *Proceedings of the 16th ACM conference on Computer and communications security*. ACM. 2009, pp. 635–647.

[SW11] Frank Stajano and Paul Wilson. "Understanding scam victims: seven principles for systems security." *Communications of the ACM* 54.3 (2011), pp. 70–75.

[SZ15] Yonatan Sompolinsky and Aviv Zohar. "Secure high-rate transaction processing in bitcoin." *International Conference on Financial Cryptography and Data Security*. Springer. 2015, pp. 507–527.

[Tan03] Matthew Tanase. *IP Spoofing: An Introduction*. 2003.

[Tan87] A.S. Tanenbaum. *Operating Systems*. Prentice-Hall software series v. 1. Pearson, 1987.

[The15] S. Theodoridis. *Machine Learning: A Bayesian and Optimization Perspective*. .NET Developers Series. Elsevier Science, 2015.

[TJ03] Marina Thottan and Chuanyi Ji. "Anomaly detection in IP networks." *IEEE Transactions on signal processing* 51.8 (2003), pp. 2191–2204.

[TMK13] Matthias Templ, Bernhard Meindl, and Alexander Kowarik. "Introduction to statistical disclosure control (sdc)." *Project: Relative to the testing of SDC algorithms and provision of practical SDC, data analysis OG* (2013).

[Tuc70] Bryant Tuckerman. *A study of the Vigenere-Vernam single and multiple loop enciphering systems*. IBM Thomas J. Watson Research Center, 1970.

[Tur37] Alan Mathison Turing. "On computable numbers, with an application to the Entscheidungsproblem." *Proceedings of the London mathematical society* 2.1 (1937), pp. 230–265.

[TW13] A.S. Tanenbaum and D.J. Wetherall. *Computer Networks*. Pearson custom library. Pearson Education, Limited, 2013.

[Urb+18] Tobias Urban et al. "Towards understanding privacy implications of adware and potentially unwanted programs." *European Symposium on Research in Computer Security*. Springer. 2018, pp. 449–469.

[US 17] District of New Jersey U.S. Attorney's Office. "Justice Department Announces Charges And Guilty Pleas In Three Computer Crime Cases Involving Significant Cyber Attacks." *Department of Justice* (2017).

[US 19] Western District of Pennsylvania U.S. Attorney's Office. "Siemens Contract Employee Intentionally Damaged Computers by Planting Logic Bombs into Programs He Designed." *Department of Justice* (2019).

[Van+10] Marten Van Dijk et al. "Fully homomorphic encryption over the integers." *Annual international conference on the theory and applications of cryptographic techniques*. Springer. 2010, pp. 24–43.

[VC15] M. Ventris and J. Chadwick. *Documents in Mycenaean Greek*. Cambridge University Press, 2015.

[Ven07] Girish Venkatachalam. "The OpenSSH protocol under the hood." *Linux Journal* 2007.156 (2007), p. 6.

[Vij07] Jaikumar Vijayan. "Unix Admin Pleads Guilty to Planting Logic Bomb." *PC World* (2007).

[VJR18] Dejan Vujičić, Dijana Jagodić, and Siniša Ranđić. "Blockchain technology, bitcoin, and Ethereum: A brief overview." *2018 17th international symposium infoteh-jahorina (infoteh)*. IEEE. 2018, pp. 1–6.

[VP09] Martin Vuagnoux and Sylvain Pasini. "Compromising Electromagnetic Emanations of Wired and Wireless Keyboards." *USENIX security symposium*. 2009, pp. 1–16.

[Wal15] Mark Walport. *Distributed Ledger Technology: beyond block chain*. 2015.

[Wal20] Amy Walker. "UK '95% sure' Russian hackers tried to steal coronavirus vaccine research." *Guardian* (July 2020).

[Wan+12] Haiyang Wang et al. "On the impact of virtualization on dropbox-like cloud file storage/synchronization services." *2012 IEEE 20th International Workshop on Quality of Service*. IEEE. 2012, pp. 1–9.

[Wan+19] Wenbo Wang et al. "A survey on consensus mechanisms and mining strategy management in blockchain networks." *IEEE Access* 7 (2019), pp. 22328–22370.

[WAZ10] Ping Wang, Baber Aslam, and Cliff C Zou. "Peer-to-peer botnets." *Handbook of Information and Communication Security*. Springer, 2010, pp. 335–350.

[WB18] J. Wolff and S. Braman. "The Most Wanted Cybercriminal in the World: GameOver ZeuS, Cryptolocker, and the Rise of Ransomware." *You'll see this message when it is too late: The Legal and Economic Aftermath of Cybersecurity Breaches*. 2018, pp. 59–78.

[WB90] Samuel D Warren and Louis D Brandeis. "The right to privacy." *Harvard law review* (1890), pp. 193–220.

[Wea+03] Nicholas Weaver et al. "A taxonomy of computer worms." *Proceedings of the 2003 ACM workshop on Rapid Malcode*. 2003, pp. 11–18.

[Wei+04] Stephen A Weis et al. "Security and privacy aspects of low-cost radio frequency identification systems." *Security in pervasive computing*. Springer, 2004, pp. 201–212.

[Wei91] Mark Weiser. "The computer for the 21st century." *Scientific american* 265.3 (1991), pp. 94–104.

[Wei93] J. Weizenbaum. *Computer Power and Human Reason: From Judgment to Calculation*. Pelican books. Penguin Books, 1993.

[Whi20] Geoff White. *Love Bug's creator tracked down to repair shop in Manila*. 2020.

[Wie65] N. Wiener. *Cybernetics Or Control and Communication in the Animal and the Machine*. DE-601)251474038: MIT paperback series. M.I.T. Press, 1965.

[Win21] Ron Winward. *IoT Attack Handbook*. Tech. rep. Radware, 2021.

[WYY05] Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. "Finding collisions in the full SHA-1." *Annual international cryptology conference*. Springer. 2005, pp. 17–36.

[YC17] Wujian Ye and Kyungsan Cho. "P2P and P2P botnet traffic classification in two stages." *Soft Computing* 21.5 (2017), pp. 1315–1326.

[YKS15] Wu Youyou, Michal Kosinski, and David Stillwell. "Computer-based personality judgments are more accurate than those made by humans." *Proceedings of the National Academy of Sciences* (2015).

[Zit08] Jonathan Zittrain. "Privacy 2.0." *U. Chi. Legal F.* (2008), p. 65.

[Zov06] Dino A Dai Zovi. "Hardware virtualization rootkits." *Black Hat 2006, August* (2006).

[ZZ04] Z. Zuo and M. Zhou. "Some Further Theoretical Results about Computer Viruses." *The Computer Journal* 47.6 (2004), pp. 627–633.

Ευρετήριο