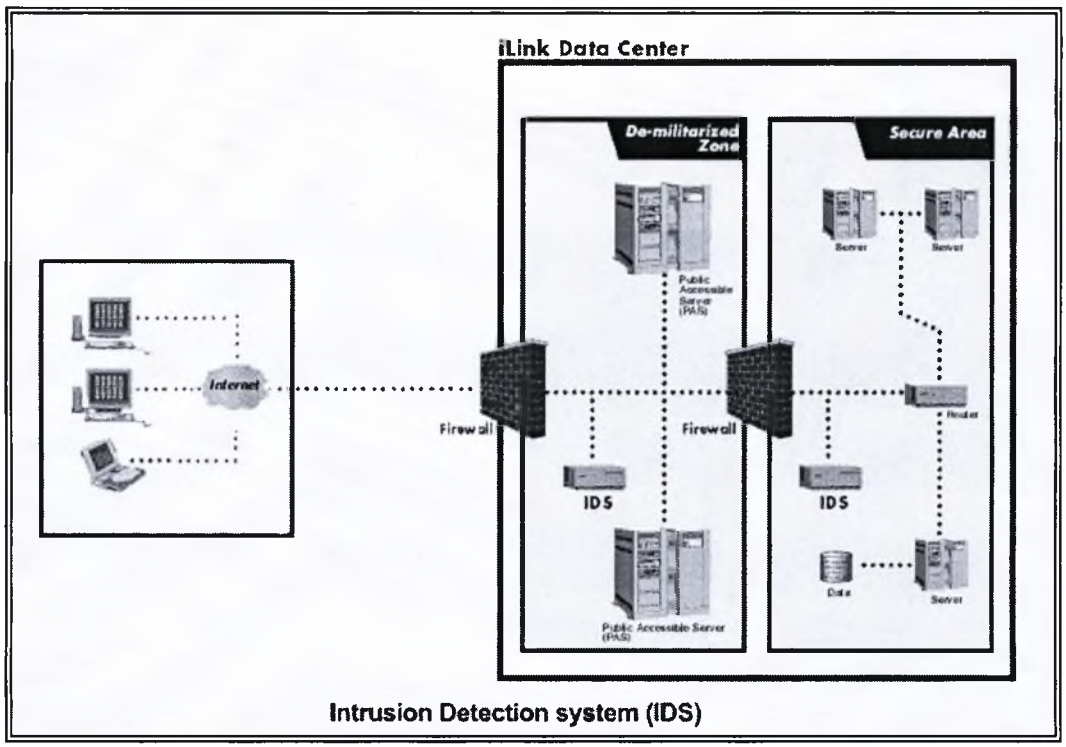
ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

Τμήμα Μηχανικών Η/Υ, Τηλεπικοινωνιών & Δικτύων

Διπλωματική Εργασία

"Συστήματα Ανίχνευσης Εισβολέων: Κοινωνική Μηχανική"



Ειρήνη Νάρη

ΒΟΛΟΣ 2005

ΠΕΡΙΕΧΟΜΕΝΑ

Πρόλογος

Η συγκεκριμένη διπλωματική εργασία εκπονήθηκε από την Ειρήνη Νάρη, φοιτήτρια του τμήματος Μηχανικών Η/Υ Τηλεπικοινωνιών και Δικτύων, υπό την επίβλεψη του καθηγητή κ. Χρήστου Ηλιούδη.

Το θέμα της διπλωματικής εργασίας είναι: «Συστήματα Ανίχνευσης Εισβολέων: Κοινωνική Μηχανική». Στόχος της είναι αρχικά να αναδείξει τα χαρακτηριστικά των Συστημάτων Ανίχνευσης Εισβολέων καθώς και του φαινόμενου που καλείται Κοινωνική Μηχανική και στη συνέχεια να εξετάσει τον τρόπο με τον οποίο τα Συστήματα Ανίχνευσης Εισβολέων μπορούν να αντιμετωπίσουν το φαινόμενο αυτό.

Για το σκοπό αυτό θα πραγματοποιηθεί μία πιλοτική εφαρμογή που θα περιλαμβάνει την προσαρμογή ενός ανοιχτού συστήματος προσδιορισμού εισβολέων τύπου Snort σε διαφορετικούς τύπους επιθέσεων και την εκπόνηση πειραμάτων στο δίκτυο του τμήματος.

Αναλυτικότερα, η πρώτη ενότητα ασχολείται με το θέμα της ασφάλειας των πληροφοριακών συστημάτων και τις επιθέσεις που πραγματοποιούνται σε αυτά και εισάγει τις έννοιες των Συστημάτων Ανίχνευσης Εισβολέων και του φαινομένου της Κοινωνικής Μηχανικής. Στη δεύτερη ενότητα παρουσιάζονται τα Συστήματα Ανίχνευσης Εισβολέων (Intrusion Detection Systems - IDS), τα οποία αποτελούν τη βασική γραμμή άμυνας σε κάθε δίκτυο. Στην τρίτη ενότητα αναλύεται το φαινόμενο της Κοινωνικής Μηχανικής, που αποτελεί μία νέα απειλή για τα δίκτυα υπολογιστών. Στην τέταρτη ενότητα δίνεται η περιγραφή του Snort, το οποίο είναι ένα δικτυακό σύστημα ανίχνευσης εισβολέων, καθώς και τα πειράματα υλοποίησης που έγιναν για την αντιμετώπιση επιθέσεων κοινωνικής μηχανικής, χρησιμοποιώντας το Snort. Στην πέμπτη ενότητα αναφέρονται τα συμπεράσματα που εξήχθησαν από τη συγκεκριμένη διπλωματική εργασία ενώ στη έκτη ενότητα αναφέρονται μελλοντικές επεκτάσεις που μπορούν να πραγματοποιηθούν για την αποτελεσματικότερη αντιμετώπιση των επιθέσεων κοινωνικής μηχανικής, με τη χρήση του εργαλείου Snort. Το παράρτημα περιέχει ένα αρχείο του Snort που χρησιμοποιήθηκε στα πειράματα υλοποίησης. Τέλος, ακολουθεί η βιβλιογραφία που αναφέρει τις πηγές πληροφόρησης που χρησιμοποιήθηκαν για την συγγραφή μέρους αυτής της εργασίας.

Ενότητα

1

ΕΙΣΑΓΩΓΗ

Το διαδίκτυο (Internet) σχεδιάστηκε από ακαδημαϊκές και επιστημονικές κοινότητες προκειμένου να επιτευχθεί η ανταλλαγή πληροφοριών μεταξύ έμπιστων οντοτήτων. Αρχικά, το θέμα της ασφάλειας των ευαίσθητων πληροφοριών δεν απασχόλησε τους σχεδιαστές του. Ο λόγος είναι ότι κανένας δεν μπορούσε να προβλέψει τότε ότι το διαδίκτυο θα επεκταθεί και θα συνδέσει την πλειοψηφία των δημόσιων και ιδιωτικών δικτύων που υπάρχουν σήμερα.

Ωστόσο με τη ραγδαία εξάπλωση του διαδικτύου παγκοσμίως αυξήθηκαν και τα περιστατικά επιθέσεων από τρίτους σε συγκεκριμένους υπολογιστές ή ομάδες υπολογιστών που ανήκαν σε κάποιο υποδίκτυο ενός δικτύου [1]. Τα φαινόμενα όμως αυτά ήταν αναμενόμενο να πραγματοποιηθούν αν ληφθεί υπόψιν το γεγονός ότι ενώ δαπανήθηκαν μεγάλα χρηματικά ποσά για την ανάπτυξη και την εξέλιξη της παραπάνω τεχνολογίας, δαπανήθηκαν πολύ μικρά και σε μερικές περιπτώσεις μηδαμινά ποσά στον τομέα της ασφάλειας των υπολογιστών και των πληροφοριών.

Στη σημερινή εποχή το πρόβλημα της ασφάλειας των πληροφοριών και των δικτύων γενικότερα είναι ιδιαίτερα σημαντικό στα σύγχρονα πληροφοριακά συστήματα. Στα πλαίσια των αρμοδιοτήτων που έχουν οι διαχειριστές συστημάτων περιλαμβάνεται και η προστασία των υπολογιστικών συστημάτων από ποικίλες επιθέσεις. Τα συστήματα πρέπει να παρακολουθούνται με σκοπό την ανίχνευση και καταγραφή όλων των προσπαθειών για επιτυχή αλλά και ανεπιτυχή παραβίαση της ασφάλειας. Οι διαχειριστές δικτύων για να προστατεύσουν τα δίκτυα για τα οποία είναι υπεύθυνοι, χρησιμοποιούν firewalls και proxy servers (ακόμη και μερικά ιδιωτικά σπίτια ακολουθούν αυτήν την τακτική). Μερικά δίκτυα αναγκάζουν την κίνηση για το Internet να περνά μέσα από έναν HTTP proxy server ή mail server ώστε να μην επιτρέπεται η απευθείας πρόσβαση στο δίκτυο [2]. Αυτό δυσκολεύει περισσότερο τον

κακόβουλο χρήστη για την πραγματοποίηση του στόχου του, που αρχικά είναι η πρόσβαση το εσωτερικό υποδίκτυο ενός δικτύου.

Συνήθως η μη εξουσιοδοτημένη πρόσβαση επιτυγχάνεται με την εκμετάλλευση των ευπαθειών των λειτουργικών συστημάτων. Αυτό μπορεί να πραγματοποιηθεί με ποικίλους τρόπους. Για παράδειγμα, όταν ο επιτεθείς επιλέξει ένα στόχο, εκτελεί λογισμικό το οποίο του προσδιορίζει τον τύπο του λειτουργικού που χρησιμοποιεί το απομακρυσμένο σύστημα. Στη συνέχεια αναζητά στους κατάλληλους δικτυακούς τόπους τα ελαττώματα του συγκεκριμένου λειτουργικού συστήματος και scripts τα οποία εκμεταλλεύονται το υπολογιστικό σύστημα του θύματος. Τα εργαλεία ανίχνευσης επιθέσεων βοηθούν τους διαχειριστές συστημάτων να σταματούν επιθέσεις που στοχεύουν στο δίκτυο και επίσης βοηθούν στον εντοπισμό των ατόμων που τις πραγματοποίησαν.

Αξίζει να σημειωθεί ότι ο αριθμός των επιθέσεων που δέχονται τα υπολογιστικά συστήματα συνεχώς αυξάνεται, γεγονός που προκαλεί την ανησυχία στους διαχειριστές και απαιτεί δραστικότερα μέτρα για την έγκυρη ανίχνευση και αντιμετώπιση των επιθέσεων. Σύμφωνα με τα αποτελέσματα της έκθεσης της Symantec, που είναι παγκόσμιος ηγέτης στην ασφάλεια πληροφορικών συστημάτων, παρατηρούνται συνεχώς όλο και περισσότερες απειλές για το Ηλεκτρονικό Εμπόριο και τις Δικτυακές Εφαρμογές [3]. Η έκθεση αυτή, που δημοσιεύτηκε τον Σεπτέμβριο του 2004, αναλύει και καταγράφει τις τάσεις που παρατηρούνται όσον αφορά στις Διαδικτυακές απειλές, τα τρωτά σημεία (vulnerabilities), και την εμφάνιση επιβλαβούς κωδικού (malicious code) κατά την χρονική περίοδο μεταξύ 1ης Ιανουαρίου και 30 Ιουνίου του 2004. Τις περισσότερες απειλές δέχτηκε το ηλεκτρονικό εμπόριο με ποσοστό 16%, ποσοστό που αντιστοιχεί σε μία αύξηση επιθέσεων κατά 400%, σε σχέση με το προηγούμενο εξάμηνο. Επίσης, παρατηρήθηκε αύξηση των επιθέσεων σε δικτυακές εφαρμογές και τεχνολογίες, οι οποίες αποτελούν προσφιλή στόχο για επιθέσεις λόγω της σχετικής ευκολίας με την οποία μπορούν οι εισβολείς να τις εκμεταλλευτούν. Οι δικτυακές εφαρμογές επιτρέπουν στους εισβολείς να αποκτήσουν πρόσβαση σε ένα σύστημα-στόχο απλά μέσω της εισβολής στον υπολογιστή ενός τελικού χρήστη, παρακάμπτοντας τα παραδοσιακά μέτρα ασφαλείας που εγκαθίστανται στα περιμετρικά σημεία ενός εταιρικού δικτύου. Σχεδόν το 82% των τρωτών σημείων που καταγράφονται όσον αφορά στις δικτυακές

εφαρμογές, χαρακτηρίζονται ως εύκολες να «αξιοποιηθούν» από εισβολείς, και συνεπώς αποτελούν μία σημαντική απειλή για την πληροφορική υποδομή και την ακεραιτότητα των επιχειρησιακών πόρων ενός οργανισμού. Αξίζει να αναφερθεί ότι ο χρόνος μεταξύ της καταγραφής τρωτών σημείων και της εκμετάλλευσής τους από εισβολείς είναι ιδιαίτερα μικρός. Σύμφωνα με τα στοιχεία που συνέλεξε η Symantec, το μέσο χρονικό διάστημα ανταπόκρισης ενός εισβολέα σε κάποιο τρωτό σημείο, μετά την καταγραφή του, ήταν μόνο 5,8 ημέρες. Αυτό το μικρό περιθώριο αφήνει στους οργανισμούς λιγότερο από μία εβδομάδα χρόνο για την αποκατάσταση των ευάλωτων συστημάτων.

Επιπλέον, παρατηρήθηκε αύξηση των bot networks. Με τον όρο bots (συντομογραφία της λέξης "robot") εννοούμε «κρυφά» προγράμματα που εγκαθίσταται σε ένα σύστημα-στόχο, επιτρέποντας σε έναν μη εξουσιοδοτημένο χρήστη να ελέγχει από ένα απομακρυσμένο σημείο κάποιον υπολογιστή και να τον χρησιμοποιεί για ποικίλους σκοπούς. Οι εισβολείς συχνά συντονίζουν μεγάλα γκρουπ συστημάτων ή δικτύων που ελέγχονται μέσω bot, αναζητώντας τρωτά συστήματα με στόχο να τα χρησιμοποιήσουν για την ενίσχυση της ταχύτητας και του εύρους των επιθέσεων που προγραμματίζουν. Η Symantec κατέγραψε σημαντική αύξηση στον αριθμό των bots. Κατά τους πρώτους έξι μήνες του 2004, ο μέσος όρος των ελεγχόμενων bots αυξήθηκε από 2000 σε περισσότερα από 30000 την ημέρα – φτάνοντας μέχρι και τα 75000 μέσα σε μία ημέρα. Τα λεγόμενα bot networks δημιουργούν εξαιρετικά προβλήματα στους οργανισμούς γιατί μπορούν να αναβαθμισθούν από μακριά για χρήση σε νέες επιθέσεις, οι οποίες μπορούν πιθανά να επιτρέψουν στους εισβολείς να ξεπεράσουν τις προσπάθειες ενός οργανισμού για τη διασφάλιση και την αποκατάσταση των ευπαθών συστημάτων του.

Η IBM κατά τη δημοσίευση πρόσφατης έρευνάς της τονίζει πως «*Η αύξηση των επιθέσεων έχει επιφέρει την οχύρωση των επιχειρήσεων, που πλέον κατανοούν τι χρειάζεται για να αντιμετωπίσουν τέτοιες απειλές. Τα νέα στοιχεία δημιουργούν νέα δεδομένα στα οποία θα πρέπει οι εταιρείες να προσαρμοστούν...*»[4]

Σύμφωνα με τους Bandy, Money και Worsted η ανίχνευση των επιθέσεων είναι σημαντική διότι είναι αδύνατο να τις αγνοήσει κανείς ελπίζοντας ότι τα δικά του υπολογιστικά συστήματα δεν θα δεχτούν επίθεση, από τη στιγμή που υπάρχει

πληθώρα επιθέσεων και όλα τα υπολογιστικά συστήματα αποτελούν πιθανό στόχο. Επιπλέον υποστηρίζουν ότι εξαιτίας της αλματώδης εξάπλωσης του Διαδικτύου, η παρακολούθηση και διαχείριση των συστημάτων απαιτούν ατελείωτες προσπάθειες για να επιτευχθούν [5]. Τα εργαλεία για την ανίχνευση των επιθέσεων βοηθούν τους διαχειριστές να διατηρήσουν ένα ασφαλές δικτυακό περιβάλλον.

Ένα από τα συστήματα που χρησιμοποιούνται για την ασφάλεια των δικτύων είναι το Σύστημα Ανίχνευσης Εισβολέων (Intrusion Detection System - IDS), το οποίο παρακολουθεί συνεχώς όλες τις λειτουργίες που εκτελούνται στο σύστημα του οποίου έχει αναλάβει την προστασία και γνωρίζοντας τη φυσιολογική συμπεριφορά και λειτουργία του συστήματος μπορεί να εντοπίσει την εισβολή που δέχεται αυτό από κακόβουλους χρήστες. Τα Συστήματα Ανίχνευσης Εισβολέων θα αναλυθούν διεξοδικά σε επόμενη ενότητα.

1.1 Εισβολές Με Χρήση Κοινωνικής Μηχανικής

Από την πληθώρα επιθέσεων σε αυτή τη μελέτη θα ασχοληθούμε με αυτές τις επιθέσεις που ανήκουν στο φαινόμενο που καλείται Κοινωνική Μηχανική. Ο όρος «κοινωνική μηχανική» (social engineering) αναφέρεται σε μέθοδο που χρησιμοποιείται για εξασφάλιση τρόπου διείσδυσης σε συστήματα και δίκτυα ηλεκτρονικών υπολογιστών με δόλιο τρόπο, με στόχο στην παράνομη απόκτηση δεδομένων μέσω της εξαπάτησης: Ο χρήστης-θύμα πείθεται είτε να αποκαλύψει ευαίσθητες πληροφορίες, όπως κωδικούς, είτε να προβεί σε ενέργειες που ανοίγουν την πόρτα στον εισβολέα. Στην περίπτωση αυτή ο κακόβουλος χρήστης δεν προσπαθεί να εκμεταλλευτεί ευπάθειες του συστήματος μα προσπαθεί να εκμεταλλευτεί τρωτά σημεία της ανθρώπινης φύσης, όπως τη ματαιοδοξία, το φόβο και την καλόπιστη εμπιστοσύνη [6].

Χρησιμοποιώντας την επιρροή και την πειθώ οι κοινωνικοί μηχανικοί εξαπατούν τα θύματά τους, είτε πείθοντάς τα ότι η ταυτότητά τους είναι άλλη από την πραγματική είτε οδηγώντας τα σε ανεπίτρεπτες πράξεις. Τελικός σκοπός του κοινωνικού μηχανικού είναι η απόκτηση ευαίσθητων πληροφοριών με ή χωρίς τη χρήση της τεχνολογίας [7].

Η κοινωνική μηχανική διακρίνεται στις δύο παρακάτω μορφές:

- 1η περίπτωση: Ο χρήστης-θύμα πείθεται να αποκαλύψει ευαίσθητες πληροφορίες, όπως κωδικούς, μετά από την επικοινωνία που έχει μαζί του ο hacker
- 2η περίπτωση: Ο χρήστης-θύμα δίχως να το αντιληφθεί προβαίνει σε ενέργειες που ανοίγουν την πόρτα στον εισβολέα

Η κοινωνική μηχανική θεωρείται ο ευκολότερος τρόπος για την απόκτηση passwords. Μία συνηθισμένη τακτική είναι η ακόλουθη. Συνήθως οι εισβολείς επικοινωνούν τηλεφωνικά ή μέσω e-mail με μεγάλες εταιρείες, βρίσκουν «ευαίσθητους» ανθρώπους, που αποτελούν τα θύματα τους. Στη συνέχεια οι εισβολείς, ισχυριζόμενοι ότι ανήκουν στο τμήμα τεχνικής υποστήριξης δικτύων, επικοινωνούν μαζί τους για να τους ενημερώσουν ότι υπάρχει κάποιο πρόβλημα ασφαλείας στο δίκτυο της εταιρείας και για το λόγο αυτό τους ζητούν τον κωδικό τους (password) για να τον αλλάξουν. Η παραπάνω τακτική αποτελεί τον ευκολότερο τρόπο απόκτησης passwords και βασίζεται μόνο στην εμπιστοσύνη που δείχνουν οι χρήστες.

Ίσως το πιο γνωστό πρόσφατο παράδειγμα κοινωνικής μηχανικής είναι ο ιός "Love Bug" ο οποίος επιτέθηκε σε πολλά δίκτυα το Μάιο του 2000. Από τεχνικής άποψης δεν ήταν ιδιαίτερα δύσκολο να κατασκευαστεί. Ο ιός εκμεταλλεύτηκε την εμπιστοσύνη των χρηστών χρησιμοποιώντας τη φράση "I love you" και βασίστηκε στις αδυναμίες του Microsoft Outlook και Outlook Express, που είναι e-mail προγράμματα.

Μέχρι πριν εμφανιστεί ο ιός "Love Bug", οι χρήστες εμπιστεύονταν τα περισσότερα e-mail που λάμβαναν καθημερινά και είχαν ως θέμα "I love you", κυρίως αν προερχόταν από έμπιστο προορισμό. Αυτό συνέβαινε για το λόγο ότι η συγκεκριμένη φράση συνήθως χρησιμοποιείται για να εκφράσει τα συναισθήματα μας. Αρχικά, όταν ο ιός παρουσιάστηκε, οι περισσότεροι χρήστες άνοιγαν το συγκεκριμένο e-mail για έναν από τους παρακάτω τρεις λόγους:

- Θεωρούσαν ότι το e-mail προερχόταν από έμπιστο προορισμό (για παράδειγμα από έναν συνάδερφο ή φιλικό πρόσωπο)
- Πολλοί χρήστες νόμιζαν ότι αποτελούσε μέρος κάποιου αστείου (τα e-mail είναι συνήθης τακτική για κατανεμημένα αστεία)

- Υπάρχουν ορισμένοι χρήστες που άνοιξαν το e-mail, για τον απλό λόγο ότι διαβάζουν όλα τα e-mails που τους στέλνονται.[8]

Ο ιός μόλυνε μόνο τα συστήματα που έτρεχαν Microsoft Windows και δεν επηρέαζε αυτά που χρησιμοποίησαν λειτουργικό σύστημα της Apple ή Linux. Επιπλέον, ο ιός εξαπλώνονταν διαμέσου του MIRC, του δημοφιλούς chat προγράμματος. Ο εν λόγω ιός είχε ως θέμα "I Love You" και ένα συνημμένο αρχείο με το όνομα: Love-Letter-For-You.txt.vbs.[9] Όταν ο χρήστης άνοιγε το συνημμένο αρχείο συνέβαιναν τα παρακάτω:

- Ο ιός αντέγραφε τον ίδιο στέλνοντας e-mail σε όλους όσους αναφέρονταν στη λίστα του βιβλίου διευθύνσεων του θύματος.
- Έψαχνε για τα αρχεία που είχαν τις καταλήξεις .jpeg, .mp3, .mp2, .jpg, .js, .jse, .css, .wsh, .sct, και .hta και έγραφε σε αυτά τον εαυτό του αλλάζοντας τις καταλήξεις σε .vbs ή .vbe. Τα αρχεία αυτά, αν ο χρήστης δεν τα είχε κάνει backups, δεν μπορούσαν να ανακτηθούν ξανά.

Ο ιός "Love Bug", που εκμεταλλεύτηκε την εμπιστοσύνη των χρηστών, εξαπλώθηκε σε 45 εκατομμύρια υπολογιστές και απαίτησε δισεκατομμύρια δολαρίων για την επισκευή των υπολογιστών που είχαν μολυνθεί [10]. Όπως κάποιοι υποστήριζαν τότε «*Η εμπιστοσύνη ποτέ δεν ήταν τόσο ακριβή...*»

Ένα ακόμη παράδειγμα κοινωνικής μηχανικής είναι το ακόλουθο. Ο hacker αναπαριστά, ένα IT πρόσωπο που καλεί τον υπάλληλο του κέντρου υπηρεσιών (facility operations center) λέγοντας του: "Εντοπίσαμε κάποια ύποπτη δραστηριότητα σε εσάς. Μπορούμε να επαληθεύσουμε την IP σας διεύθυνση;" Με αυτήν την πληροφορία, ο hacker μπορεί να αρχίσει την αναγνώριση όλων των HVAC συστατικών. Στη συνέχεια αφού εντοπίσει πιθανούς στόχους, προσπαθεί να βρει τις ευπάθειες των στόχων αυτών με τη βοήθεια κάποιων εργαλείων (tools) κοινωνικής μηχανικής. Τα εργαλεία αυτά θα εντοπίσουν αρκετές ευπάθειες στα συστατικά μα ο hacker θα επικεντρωθεί σε αυτό το οποίο θα τον βοηθήσει γρηγορότερα να υποκλέψει το όνομα χρήστη και τον κωδικό που τον ενδιαφέρουν [11].

Τα παραπάνω είναι μερικές τεχνικές επιθέσεων κοινωνικής μηχανικής, οι οποίες έχουν ως στόχο να εκμεταλλευτούν τις αδυναμίες των χρηστών. Άλλες μορφές

κοινωνικής μηχανικής είναι τα e-mail σκουλήκια (worms) και τα spam μηνύματα. Σύμφωνα με τον Mitnick [7]:

> *«...ένας κοινωνικός μηχανικός μπορεί να στείλει έναν ιό ή Δούρειο Ίππο ως συνημμένο αρχείο σε ένα e-mail, να ανακαλύψει τα πλήκτρα που πάτησε το θύμα-χρήστη χρησιμοποιώντας διαθέσιμα προγράμματα, να αφήσει μία δισκέτα ή ένα CD με επιβλαβή κώδικα (malicious software) στο χώρο εργασίας του θύματος, να χρησιμοποιήσει ψεύτικα pop-up παράθυρα ζητώντας από το χρήστης να κάνει log on ξανά ή να συνδεθεί με το δικτυακό του password, να στείλει δωρεάν λογισμικό για να το εγκαταστήσει το θύμα...»*

Πλέον η κοινωνική μηχανική στα e-mails δεν χρησιμοποιείται μόνο για να ξεγελάσει τους χρήστες αλλά και τα φίλτρα (filters). Για παράδειγμα, παρατηρώντας κάποια spam μηνύματα καταλήγουμε στο συμπέρασμα ότι αρχικά τα μηνύματα spam ήταν εύκολα αντιληπτά από το χρήστη, διαβάζοντας απλώς το θέμα και τη διεύθυνση αποστολέα. Σήμερα τα spam μηνύματα χρησιμοποιούν πιο πειστικά στοιχεία ώστε να τα διαβάσει και να τα ανοίξει ο χρήστης. Αν δηλαδή κάποιος λάβει μήνυμα από τη διεύθυνση "loan@loanoffice.org" με θέμα "Re: Loan Proposal", υπάρχει περίπτωση να το διαβάσει γιατί πιθανόν ενδιαφέρεται να πάρει κάποιο δάνειο [12]. Τότε θεωρούμε ότι η κοινωνική μηχανική έλαβε χώρα. Ακόμη και αν ο χρήστης δεν βρει ενδιαφέρον το μήνυμα και δεν κάνει καμία επιπλέον ενέργεια, ο σκοπός της κοινωνικής μηχανικής έχει επιτευχθεί.

Στόχος των διαχειριστών δικτύων και υπεύθυνων ασφαλείας είναι να αντιμετωπιστεί όσο γίνεται αποτελεσματικότερα το πρόβλημα του προσδιορισμού εισβολέων μπροστά στον αδύναμο κρίκο που είναι ο χρήστης μέσω των επιθέσεων κοινωνικής μηχανικής. Στα κεφάλαια που ακολουθούν θα αναλυθεί πόσο ικανοποιητικά μπορεί αυτό να επιτευχθεί χρησιμοποιώντας τα Συστήματα Ανίχνευσης Εισβολέων.

Ενότητα

2

ΤΕΧΝΟΛΟΓΙΑ ΚΑΙ ΑΡΧΙΤΕΚΤΟΝΙΚΗ IDS

Στη συγκεκριμένη ενότητα αναπτύσσονται θέματα που αφορούν κυρίως την τεχνολογία και την αρχιτεκτονική των Συστημάτων Ανίχνευσης Εισβολέων (Intrusion Detection Systems - IDS). Επιπλέον, αναφέρονται κάποια γενικά στοιχεία για τα IDS, τα επιθυμητά χαρακτηριστικά τους, τα μοντέλα εισβολών, οι τύποι κατηγοριοποίησης τους, ο τρόπος ανίχνευσης επιθέσεων, η απόκριση στις εισβολές, οι αδυναμίες τους καθώς και μερικά ενδεικτικά προϊόντα IDS που χρησιμοποιούνται.

2.1 Εισαγωγή

Η ασφάλεια του Internet είναι ένα θέμα που απασχολεί τόσο τους διαχειριστές δικτύων όσο και τους χρήστες υπολογιστών. Αυτό συμβαίνει εξαιτίας του συνεχώς αυξανόμενου ρυθμού με τον οποίο πραγματοποιούνται οι επιθέσεις στα συστήματα δικτύων καθώς και του αυξημένου επιπέδου πολυπλοκότητας που αυτές παρουσιάζουν. Τα παραπάνω ενισχύει η διαθεσιμότητα αυτοματοποιημένων εργαλείων επιθέσεων που παρέχονται δωρεάν μέσω του Διαδικτύου καθώς και η φύση του ηλεκτρονικού εγκλήματος, η οποία είναι απρόβλεπτη, αφού προηγούμενες απειλές ή επιθέσεις δεν μπορούν πάντα να χρησιμοποιηθούν ως βάση για να προβλέψουν μελλοντικές.

Ωστόσο, σχεδόν όλες οι επιθέσεις που στοχεύουν σε δίκτυα υπολογιστών μπορεί να ανιχνευτούν αν οι διαχειριστές των συστημάτων λάβουν τα απαραίτητα μέτρα για την ασφάλεια και παρακολούθηση των δικτύων στα οποία είναι υπεύθυνοι. Η διαδικασία της πρόληψης και ανίχνευσης γεγονότων παραβίασης της ασφάλειας μέσω της παρακολούθησης των δραστηριοτήτων του χρήστη και του συστήματος καλείται ανίχνευσης εισβολής.

Στη συνέχεια παρατίθενται μερικές ενδείξεις που δηλώνουν μία εισβολή [13]:

- ένας συναγερμός (alarm) ή άλλη ένδειξη που προέρχεται από το σύστημα ανίχνευσης εισβολών
- ύποπτη πρόσβαση στο UNIX root χωρίς να ακολουθείται η κανονική διαδικασία
- αποτυχημένες logon προσπάθειες
- ανεξήγητοι νέοι λογαριασμοί χρηστών
- ασυνήθιστα νέα ονόματα αρχείων
- ασυνήθιστες αλλαγές σε μεγέθη αρχείων ή ημερομηνίες, κυρίως σε εκτελέσιμα αρχεία του συστήματος
- ασυνήθιστη αλλαγή ή διαγραφή δεδομένων
- άρνηση εξυπηρέτησης (denial of service) ή αδυναμία ενός ή περισσοτέρων χρηστών να κάνουν login
- κατάρρευση (crashes) ή τερματισμός του συστήματος
- ασυνήθιστες ώρες χρήσης του συστήματος (νωρίς το πρωί ή αργά τη νύχτα)
- ασυνήθιστες ενέργειες του χρήστη (μεταγλώττιση προγραμμάτων από χρήστη που δεν ξέρει να προγραμματίσει)

Οι εισβολές πραγματοποιούνται με τα εργαλεία επίθεσης (attack tools), τα οποία είναι αυτοματοποιημένα προγράμματα σχεδιασμένα με σκοπό την παραβίαση της πολιτικής ασφάλειας ενός συστήματος. Τα εργαλεία επίθεσης και γενικότερα οι μέθοδοι με τις οποίες μπορεί κάποιος να πραγματοποιήσει μία εισβολή είναι ιδιαίτερα προηγμένα και αυτοματοποιημένα, και τα περισσότερα από αυτά διατίθενται δωρεάν στο Διαδίκτυο ενώ ο επιτιθέμενος δεν απαιτείται να είναι εξειδικευμένος για να τα χρησιμοποιήσει.

Ένα γνωστό εργαλείο επίθεσης είναι το rootkit [14] το οποίο διατίθεται για πολλές εκδόσεις του λειτουργικού συστήματος UNIX. Παρακολουθεί (sniff) την κίνηση του δικτύου και υποκλέπτει τα συνθηματικά που χρησιμοποιούνται, χωρίς να γίνεται αντιληπτό [15]. Επιπλέον περιλαμβάνει τροποποιημένες εκδόσεις εργαλείων συστήματος (system utilities). Το ifconfig, που είναι πρόγραμμα διαμόρφωσης δικτύου και αναφέρει τις ρυθμίσεις των συσκευών δικτύου, ισχυρίζεται ότι η συσκευή δικτύου δε βρίσκεται σε ασυνήθιστη κατάσταση μεταφοράς (promiscuous mode), όπως πρέπει να συμβαίνει για να πραγματοποιηθεί η παρακολούθηση στο δίκτυο. Οι επιτιθέμενοι έχουν τη δυνατότητα να επιστρέψουν και να ανακτήσουν τα

συνθηματικά που υποκλάπηκαν εκμεταλλευόμενοι το πρόγραμμα login που δέχεται ένα ειδικό συνθηματικό το οποίο μπορεί να αυθεντικοποιήσει οποιονδήποτε χρήστη. Τα προγράμματα αυτά είναι τροποποιημένα με συστήματα ανίχνευσης εισβολών ώστε να μην γίνονται αντιληπτά και τόσο τα τροποποιημένα όσο και τα γνήσια παράγουν ακριβώς τα ίδια αθροίσματα ελέγχου (check sum).

Το rootkit περιέχει διάφορα άλλα προγράμματα για την απόκρυψη του επιτιθέμενου, όπως το πρόγραμμα zapper που διαγράφει τις εγγραφές των χρηστών από το αρχείο utmp και ο χρήστης δεν είναι αντιληπτός μετά τη διαδικασία εισόδου στο σύστημα, καθώς και το πρόγραμμα fixer που εγκαθιστά τα προγράμματα αυτά και ρυθμίζει τα δικαιώματά τους, ώστε να ταιριάζουν με αυτά των αντικατεστημένων προγραμμάτων.

Το rootkit, όπως και τα περισσότερα εργαλεία επίθεσης έχουν τη δυνατότητα να εξαλείψουν αρκετά ίχνη που δημιουργήθηκαν κατά τη διάρκεια της εγκατάστασης τους όταν πραγματοποιούσαν την επίθεση. Δεν μπορούν όμως να εξαλείψουν όλα τα ίχνη, γεγονός που εκμεταλλεύονται τα συστήματα ανίχνευσης εισβολέων για να εντοπίσουν τις επιθέσεις.

2.2 Υπογραφές Εισβολών

Υπάρχουν τρία είδη επιθέσεων που θεωρούνται γνωστές «υπογραφές εισβολών»:

- **Αναγνώριση (reconnaissance):** Στην κατηγορία αυτή περιέχονται ping sweeps, DNS zone transfers, e-mail recons, TCP ή UDP port scans, OS fingerprinting και πιθανή αναζήτηση στους δημόσιους web servers για εύρεση cgi «τρυπών».
- **Εκμετάλλευση (exploits):** Οι εισβολείς εκμεταλλεύονται κρυμμένα χαρακτηριστικά, bugs ή ευπάθειες για να κερδίσουν πρόσβαση στο σύστημα.
- **Denial of Service (DoS) επιθέσεις:** Στη περίπτωση αυτή ο εισβολέας προσπαθεί να καταρρεύσει μία υπηρεσία ή το ίδιο το μηχάνημα, να υπερφορτώσει δικτυακές συνδέσεις ή τη CPU ή να γεμίσει με άχρηστα δεδομένα το δίσκο. Ο εισβολέας δεν προσπαθεί να έχει πρόσβαση σε πληροφορίες απλώς αντιδρά σα βάνδαλος ώστε να αποτρέψει τους χρήστες να χρησιμοποιούν το σύστημα.

2.3 Βασικές Αρχές - Ορισμός IDS

Σύμφωνα με τους Denning [16] και Bishop [17] τα υπολογιστικά συστήματα που δέχονται επίθεση δεν πληρούν ένα από τα ακόλουθα χαρακτηριστικά:

1. Το σύνολο των ενεργειών των χρηστών και των διεργασιών ακολουθούν ένα στατιστικά προβλέψιμο πρότυπο. Για παράδειγμα, αναφέρονται περιπτώσεις όπου επιτρέπεται η πρόσβαση στο δίκτυο από διευθύνσεις που παλαιότερα απαγορευόταν ή εκτελούνται συγκεκριμένες εντολές από το χρήστη που πριν δεν εκτελούνταν.
2. Οι ενέργειες των χρηστών και των διεργασιών δεν περιλαμβάνουν ακολουθίες εντολών που να υπονομεύουν την πολιτική ασφάλειας του συστήματος. Θεωρητικά, κάθε τέτοια ακολουθία εντολών πρέπει να μη γίνεται δεκτή. Στην πραγματικότητα, όμως, μπορούν να ανιχνευθούν μόνο γνωστές ακολουθίες υπονόμευσης του συστήματος. Για παράδειγμα, όταν ένας εισβολέας θέλει να αποκτήσει δικαιώματα root σε κάποιο σύστημα τότε χρησιμοποιεί κάποιες τεχνικές οι οποίες μπορεί να περιλαμβάνουν ακολουθίες εντολών, σχεδιασμένες με τρόπο ώστε να παραβιάζεται η πολιτική ασφαλείας του συστήματος.
3. Οι ενέργειες των διεργασιών συμμορφώνονται με ένα σύνολο προδιαγραφών που περιγράφουν επιτρεπτές ενέργειες.

Όταν ένα πρόγραμμα παρακολούθησης δικτύου sniffer προσπελάσει τη συσκευή δικτύου, τη θέτει σε ασυνήθιστη κατάσταση μεταφοράς (promiscuous mode) και το γεγονός αυτό, σε μερικά συστήματα, δημιουργεί μία εγγραφή στο αρχείο καταγραφής (log). Μία τέτοια εγγραφή παρουσιάζει μία γνωστή επίθεση και αποτελεί μία εισβολή.

Για την ανίχνευση επιθέσεων χρησιμοποιούνται τα Συστήματα Ανίχνευσης Εισβολέων, τα οποία παρακολουθούν συνεχώς όλες τις λειτουργίες που εκτελούνται στο σύστημα του οποίου έχουν αναλάβει την προστασία. Διαθέτουν εργαλεία και μηχανισμούς με τα οποία κάνουν συνεχείς και λεπτομερείς αναλύσεις της συμπεριφοράς ολόκληρου του συστήματος. Είναι σε θέση να γνωρίζουν ποια είναι η φυσιολογική συμπεριφορά και λειτουργία του συστήματος και να εντοπίζουν την εισβολή. Αξίζει να σημειωθεί ότι τα IDS μπορούν να παίξουν σπουδαίο ρόλο στον εντοπισμό και αξιολόγηση των DoS επιθέσεων.

Τα Συστήματα Ανίχνευσης Εισβολών πραγματοποιούν τον έλεγχο για τυχόν παραβιάσεις της ασφάλειας ενός συστήματος. Στη γενική περίπτωση, τα IDS θα μπορούσαν να καταγράφουν απλώς την κίνηση του δικτύου για μετέπειτα ανάλυση. Στη συγκεκριμένη περίπτωση, θα αποτελούσαν περισσότερο μηχανές καταγραφής συμβάντων (logging engines), παρά μηχανισμούς εντοπισμού εισβολής.

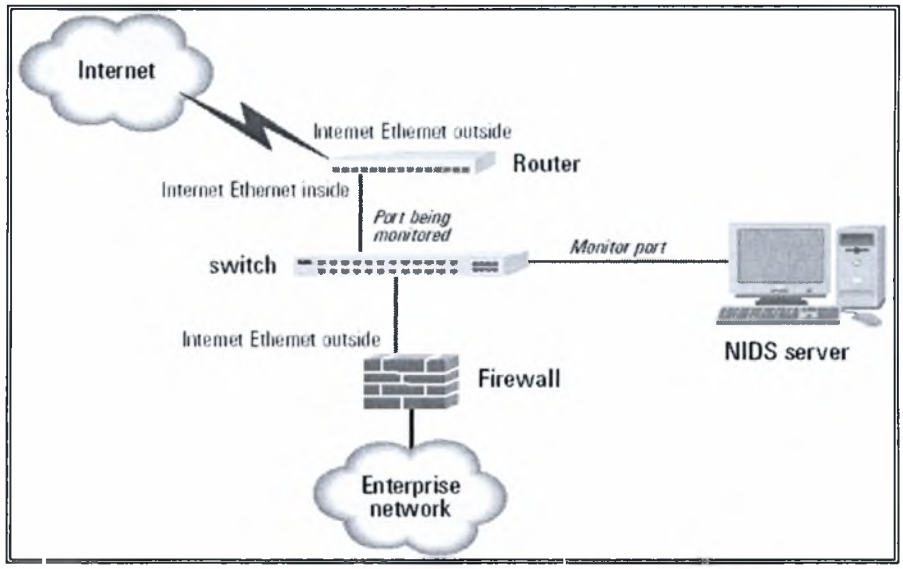
Τα Συστήματα Ανίχνευσης Εισβολέων είναι η απαραίτητη γραμμή άμυνας για κάθε δίκτυο για τους παρακάτω λόγους [1]:

- Εξαιτίας της αδυναμίας παροχής πλήρους ασφάλειας από περιμετρικές συσκευές ασφάλειας όπως των firewalls, τα οποία δεν ανιχνεύουν και δεν αντιδρούν σε περίπτωση επίθεσης.
- Μία κακή ρύθμιση σε μία περιμετρική συσκευή ασφάλειας επιτρέπει κάποια επίθεση.
- Οι εξουσιοδοτημένοι χρήστες, που κάνουν χρήση το σύστημα, είναι πιο επικίνδυνοι, λόγω παραχωρημένων προνομίων.
- Η πρόσβαση σε κατάλληλους πόρους, μπορεί να επιφέρει "προαγωγή" (root).
- Τα IDS γνωρίζουν τη δομή του δικτύου και μπορούν να αντιδράσουν κατάλληλα σε μία επίθεση.
- Είναι ικανά να σταματήσουν την επίθεση προτού συμβούν σημαντικές απώλειες στο σύστημα ή το δίκτυο
- Συγκεντρώνουν πληροφορία για επιθέσεις που έγιναν και προσπαθούν να τις σταματήσουν προτού ξανασυμβούν

Ειδικά σε δικτυακά περιβάλλοντα τα συστήματα ανίχνευσης εισβολέων χαρακτηρίζονται ως μία κάμερα παρακολούθησης που βοηθούν τους διαχειριστές να συλλάβουν κάθε κίνηση που πραγματοποιεί ο επιτιθέμενος [18]. Με τον τρόπο αυτό, ένα σύστημα ανίχνευσης εισβολέων επιτρέπει στους διαχειριστές να αναγνωρίζουν πότε και πώς το δίκτυο, για το οποίο είναι υπεύθυνοι, «παρακολουθείται» από άλλους καθώς και πότε και πώς να παρακολουθήσουν οι ίδιοι τον εισβολέα χωρίς εκείνος να το αντιληφθεί. Χρησιμοποιώντας προσαρμογέα δικτύου (network adapter) σε κατάσταση παρακολούθησης (surveillance mode) ή σε κατάσταση μεταφοράς (promiscuous mode), ένα σύστημα ανίχνευσης εισβολέων μπορεί να παρακολουθεί και να αναλύει σε πραγματικό χρόνο (real time) κάθε ύποπτο frame το οποίο ταξιδεύει από και προς το δίκτυο. Αυτή η δομή επιτρέπει στους διαχειριστές να

παρακολουθούν για υπογραφές εισβολών – συγκεκριμένα πρότυπα (patterns) τα οποία υποδεικνύουν μία μη εξουσιοδοτημένη προσπάθεια για την απόκτηση πρόσβασης στα συστήματα. Οι διαχειριστές μπορούν να κατεβάσουν τις πιο πρόσφατες υπογραφές εισβολών από το Διαδίκτυο με τον ίδιο τρόπο που κατεβάζουν τα αρχεία, που χρησιμοποιούνται για την προστασία του συστήματος από τους ιούς (virus definition files), ώστε να διατηρήσουν αναβαθμισμένο το anti-virus λογισμικό με τις νέες ρυθμίσεις ασφαλείας.

Αξίζει να αναφερθεί ότι ένα θέμα που απασχολεί τους διαχειριστές δικτύων είναι σε ποιο σημείο πρέπει να τοποθετήσουν τα IDS ώστε να είναι πιο αποτελεσματικά. Σε πολλούς οργανισμούς χρησιμοποιείται ένα ζευγάρι από IDS. Το ένα εξ αυτών τοποθετείται πριν από το firewall και το άλλο μετά από αυτό. Αυτός ο τρόπος τοποθέτησης δίνει αναφορά για τις απειλές οι οποίες εμποδίστηκαν εξαιτίας του firewall και για τις επιθέσεις που πέρασαν διαμέσου αυτού στο δίκτυο του οργανισμού. Εξετάζοντας τα alert logs και στα δυο συστήματα IDS, οι διαχειριστές μπορούν να καταλάβουν ποια κίνηση έχει περάσει από φίλτρο και ποια όχι. Πάντως, η τοποθέτηση των συστημάτων IDS ποικίλει ανάλογα με την αρχιτεκτονική του κάθε δικτύου που καλείται να προστατεύσει.



Ενδεικτική τοπολογία ενός IDS

2.4 Στόχοι – Επιθυμητά Χαρακτηριστικά Των IDS

Στην συγκεκριμένη υποενότητα παρουσιάζονται οι στόχοι και τα επιθυμητά χαρακτηριστικά των συστημάτων ανίχνευσης εισβολέων.

2.4.1 Στόχοι Των IDS

Παρακάτω δίνονται οι στόχοι που έχουν τα συστήματα ανίχνευσης εισβολέων [15]:

- ✓ Ανίχνευση μεγάλου εύρους εισβολών: Τα IDS μπορούν να εντοπιστούν γνωστές και άγνωστες επιθέσεις που προέρχονται από το εξωτερικό και το εσωτερικό του δικτύου. Για να πραγματοποιηθεί αυτό απαιτείται ένας μηχανισμός εκμάθησης και προσαρμογής στα νέα είδη εισβολών.
- ✓ Έγκαιρη ανίχνευση εισβολών: Τα IDS πρέπει να ανακαλύπτουν μία εισβολή σε εύλογο χρονικό διάστημα και όχι απαραίτητα σε πραγματικό χρόνο.
- ✓ Παρουσίαση της ανάλυσης με απλή και εύκολα αντιληπτή μορφή: Είναι επιθυμητό τα αποτελέσματα ανίχνευσης μίας εισβολής να προκύπτουν σε εύκολα αντιληπτή μορφή. Συνήθως αυτό είναι δύσκολο να πραγματοποιηθεί και έτσι ο μηχανισμός ανίχνευσης εισβολών παρουσιάζει σύνθετα δεδομένα τα οποία εξετάζει ο υπεύθυνος ασφαλείας του συστήματος και αποφασίζει αν πρέπει να ληφθούν κάποια μέτρα.
- ✓ Να είναι ακριβή: Υπάρχουν δύο είδη ψευδών σημάτων:
  - ➢ Το ψευδές θετικό σήμα υπάρχει όταν ένα σύστημα ανίχνευσης εισβολών αναφέρει μία επίθεση, ενώ στην πραγματικότητα δεν υπάρχει επίθεση σε εξέλιξη. Τα ψευδώς θετικά σήματα μειώνουν την αξιοπιστία του συστήματος και αυξάνουν την εργασία των υπευθύνων.
  - ➢ Τα ψευδώς αρνητικά σήματα υπάρχουν όταν ένα σύστημα ανίχνευσης εισβολών αποτυγχάνει να αναφέρει μία πραγματική επίθεση που βρίσκεται σε εξέλιξη. Στην περίπτωση όμως αυτή δεν πραγματοποιείται ο σκοπός των συστημάτων ανίχνευσης εισβολέων.

  Γενικός σκοπός ενός συστήματος ανίχνευσης εισβολέων είναι να ελαχιστοποιήσει τις λανθασμένες ενδείξεις και από τις δύο παραπάνω κατηγορίες σημάτων.

2.4.2 Επιθυμητά Χαρακτηριστικά Των IDS

Τα επιθυμητά χαρακτηριστικά των συστημάτων ανίχνευσης εισβολέων μπορούμε να τα συνοψίσουμε στα παρακάτω:

- ✓ Θα πρέπει να έχουν την ικανότητα γρήγορης παρακολούθησης μεγάλου όγκου δεδομένων.
- ✓ Η ειδοποίηση για την ανακάλυψη μίας εισβολής από κακόβουλους χρήστες εντός ή εκτός δικτύου θα πρέπει να γίνεται σε εύλογο χρονικό διάστημα.
- ✓ Τα IDS πρέπει να έχουν τη δυνατότητα επέκτασης και τροποποίησης ώστε κάθε φορά να προστατεύουν καλύτερα το σύστημα που έχουν αναλάβει.
- ✓ Θα πρέπει να κάνουν όσο το δυνατό οικονομία στη χρήση πόρων του συστήματος χωρίς δηλαδή να τους εξαντλούν ή να τους χρησιμοποιούν άσκοπα.
- ✓ Τέλος, τα συστήματα ανίχνευσης εισβολέων πρέπει να είναι ανθεκτικά σε επιθέσεις που στοχεύουν στα ίδια.

2.5 Μοντέλα Εισβολών

Τα συστήματα ανίχνευσης εισβολέων, έχοντας υπόψιν τα μοντέλα εισβολών, καθορίζουν αν κάποιες ενέργειες αποτελούν εισβολές. Ένα μοντέλο ταξινομεί μία ακολουθία καταστάσεων ή ενεργειών και τις χαρακτηρίζει «καλές» (αν δεν υπάρχει εισβολή) ή «κακές» (αν υπάρχουν πιθανές εισβολές). Τα μοντέλα εισβολέων είναι τα ακόλουθα [15]:

- ✓ Τα **μοντέλα ανίχνευσης διαταραχών** (anomaly models). Αυτά αποτελούν τον πιο συνηθισμένο τρόπο με τον οποίο προσεγγίζεται η ανίχνευση εισβολών στα δίκτυα, ανακαλύπτοντας στατιστικές ανωμαλίες. Τα παραπάνω μοντέλα εξετάζουν στατιστικά στοιχεία και στη συνέχεια ταξινομούν τις ενέργειες που είναι στατιστικά ασυνήθιστες ως «κακές». Στην περίπτωση αυτή η απροσδόκητη συμπεριφορά αποτελεί τεκμήριο εισβολής. Στα μοντέλα αυτά, αναπτύσσεται αρχικά ένα βασικό πλαίσιο αναμενόμενων γεγονότων ή χαρακτηριστικών των διεργασιών και των χρηστών. Η ιδέα στη συγκεκριμένη προσέγγιση είναι να μετρηθούν και να καταγραφθούν σε μία βάση τα στοιχεία του συστήματος, όπως η χρήση της CPU, η δραστηριότητα του δίσκου (disk activity) και των αρχείων καθώς και τα logins του χρήστη. Στη συνέχεια το σύστημα μπορεί να εντοπίσει πότε υπάρχει απόκλιση από τα στοιχεία που έχουν καταγραφεί. Δηλαδή, χρησιμοποιώντας στατιστικά στοιχεία οι ενέργειες ή οι καταστάσεις που είναι

στατιστικά ασυνήθιστες ταξινομούνται ως «κακές». Η μέθοδος αυτή ορίζει ένα μοτίβο κανονικής συμπεριφοράς και θεωρεί κάθε απόκλιση από αυτή ως εισβολή. Το μοτίβο κανονικής συμπεριφοράς ενημερώνεται συνεχώς καθώς το σύστημα διδάσκεται από τη συμπεριφορά του χρήστη. Το πλεονέκτημα της συγκεκριμένης μεθόδου είναι ότι μπορεί να αναγνωρίσει και άγνωστες μέχρι τότε επιθέσεις. Υπάρχουν τρία στατιστικά μοντέλα:

Μοντέλο τιμών κατωφλίου	Χρησιμοποιεί μία μετρική η οποία σχετίζεται με τιμές κατωφλίου. Τα γεγονότα εμφανίζονται κατ' ελάχιστο m και κατά μέγιστο n, για κάποιο γεγονός και κάποιες τιμές m και n. Αν σε ένα χρονικό διάστημα εμφανίζονται λιγότερα από m γεγονότα ή περισσότερα από n, τότε η συμπεριφορά θεωρείται διαταραγμένη.
Μοντέλο στατιστικών ροπών	Χρησιμοποιεί στατιστικές ροπές. Ο αναλυτής γνωρίζει τις τιμές των ροπών και αν βρίσκονται εκτός του αναμενόμενου διαστήματος, η συμπεριφορά που αντιπροσωπεύουν οι τιμές θεωρείται διαταραγμένη.
Μοντέλο Markov	Εξετάζει ένα σύστημα σε ένα συγκεκριμένο χρονικό διάστημα. Όλα τα γεγονότα που πραγματοποιήθηκαν έχουν θέσει το σύστημα σε μία συγκεκριμένη κατάσταση και κάθε φορά που πραγματοποιείται το επόμενο γεγονός, το σύστημα μεταβαίνει σε μία νέα κατάσταση. Ένα γεγονός κρίνεται διαταραγμένο, όταν προκαλεί μία μετάβαση με μικρή πιθανότητα. Το συγκεκριμένο μοντέλο προτείνει τη χρήση προϊστορίας για τον εντοπισμό των διαταραχών. Οι διαταραχές δεν είναι πλέον βασισμένες σε στατιστικά γεγονότων, αλλά σε ακολουθίες γεγονότων.

- ✓ Τα **μοντέλα κακής συμπεριφοράς** (misuse models) (ή αλλιώς αναγνώριση υπογραφής - signature recognition) συγκρίνουν ενέργειες ή καταστάσεις με ακολουθίες που είναι ήδη γνωστό ότι αποτελούν εισβολές, ή με ακολουθίες που θεωρείται ότι αποτελούν εισβολές και τις ταξινομούν ως «κακές». Η πλειοψηφία των εμπορικών προϊόντων βασίζεται στην εξέταση γνωστών τύπων επιθέσεων. Αυτό σημαίνει πως για κάθε τεχνική που χρησιμοποιούν οι hackers, οι μηχανικοί γράφουν κώδικα που προστατεύει το σύστημα από τη τεχνική αυτή. Δηλαδή, η

μέθοδος αυτή προσπαθεί να ανιχνεύσει την ύπαρξη δοσμένων ακολουθιών που σηματοδοτούν μία εισβολή. Ένα τυπικό παράδειγμα είναι η εξέταση κάθε πακέτου που μεταφέρεται με την ακολουθία "/cgi-bin/phf?", η οποία μπορεί να υποδηλώνει την προσπάθεια κάποιου να αποκτήσει πρόσβαση εξαιτίας της ευπάθειας CGI script σε έναν web server. Μερικά IDS συστήματα αναπτύσσονται από μεγάλες βάσεις δεδομένων που περιέχουν εκατοντάδες (ή χιλιάδες) από τέτοιες ακολουθίες. Απλώς, ελέγχουν τα πακέτα και ειδοποιούν αν ανιχνεύουν κάποιο που περιέχει μία ύποπτη ακολουθία. Η συγκεκριμένη μέθοδος δεν μπορεί να αναγνωρίσει άγνωστες μέχρι τότε επιθέσεις.

- ✓ Τα **μοντέλα που βασίζονται στις προδιαγραφές** (specification-based models) ταξινομούν τις καταστάσεις που παραβιάζουν τις προδιαγραφές ως «κακές». Η ανίχνευση που βασίζεται στις προδιαγραφές αναζητά ενέργειες εκτός των προδιαγραφών των βασικών προγραμμάτων. Κάθε πρόγραμμα διαθέτει ένα σύνολο κανόνων που διευκρινίζει τις επιτρεπτές ενέργειες. Εάν το πρόγραμμα προσπαθεί να προβεί σε οποιαδήποτε άλλη ενέργεια, ο μηχανισμός ανίχνευσης εισβολής αναφέρει μία πιθανή εισβολή.

Ωστόσο, καλύτερα αποτελέσματα επιτυγχάνονται όταν συνδυάζονται δύο ή και τρεις από τους παραπάνω τύπους μοντέλων. Οι συνδυασμοί αυτοί παρέχουν τη δυνατότητα ανίχνευσης μίας πληθώρας επιθέσεων και διατηρούν αναβαθμισμένο το σύστημα που βασίζεται σε αναγνώριση υπογραφής.

2.6 Κατηγοριοποίηση

Τα συστήματα ανίχνευσης εισβολέων χωρίζονται σε δύο βασικές κατηγορίες:

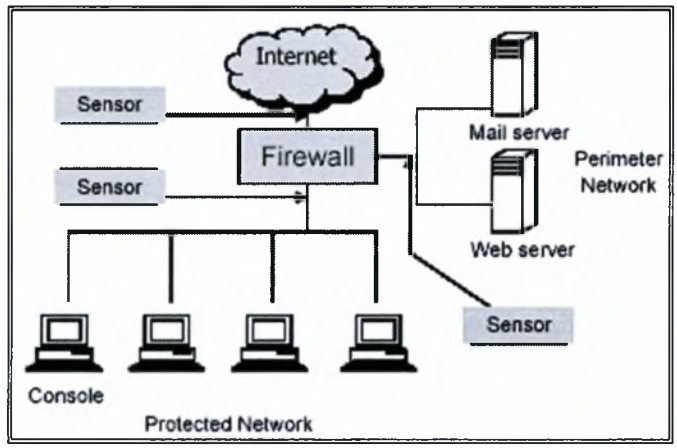
- Network Based Intrusion Detection Systems (NIDS)
- Host Based Intrusion Detection Systems (HIDS)

Τα χαρακτηριστικά των κατηγοριών αναλύονται στις ακόλουθες υποενότητες.

2.6.1 Network Based Intrusion Detection Systems

Τα Network Based Intrusion Detection Systems (NIDS) τοποθετούνται στο δίκτυο και παρακολουθούν τα πακέτα προσπαθώντας να ανακαλύψουν μία εισβολή αντιστοιχίζοντας την με μία βάση δεδομένων από γνωστές εισβολές. Ένα σύνηθες παράδειγμα είναι να αναζητούν έναν μεγάλο αριθμό από αιτήσεις TCP (SYN) σε

πολλά διαφορετικά ports σε ένα μηχάνημα στόχο, ανακαλύπτοντας έτσι κάποιον που προσπαθεί να κάνει TCP port scan. Ένα δικτυακό σύστημα ανίχνευσης εισβολέων παρακολουθεί (sniffs) αδιάκριτα όλη την κίνηση ενός δικτύου. Επιπλέον, αφού την εξετάσει, προσδιορίζει κατά πόσο πέφτει εκτός των αποδεκτών ορίων και ανιχνεύει ανωμαλίες σε ολόκληρο το δίκτυο. Με τις δυνατότητες αυτές μπορεί για παράδειγμα ένα NIDS σύστημα να σταματήσει κατανεμημένες επιθέσεις άρνησης εξυπηρέτησης (DDoS) προτού καταρρεύσουν τα μηχανήματα στα οποία στοχεύουν οι επιθέσεις.



Τοπολογία NIDS

Τα NIDS μπορούν να εκτελέσουν και άλλες εργασίες, όπως για παράδειγμα [19]:

- Να παρακολουθούν το δίκτυο για προηγούμενα port scans. Προτού ο επιτιθέμενος παραβιάσει ένα σύστημα, συνήθως κάνει port scan σε αυτό ώστε να προσδιορίσει τις ευπάθειες και αδυναμίες που μπορεί να έχει. Προσπάθειες port scan από έναν host (από το Διαδίκτυο) μπορεί να είναι προάγγελος ότι ο χρήστης που βρίσκεται πίσω από τον host έχει πρόθεση να βλάψει το δίκτυο.
- Να παρακολουθούν νόμιμες συνδέσεις για γνωστές επιθέσεις. Η προσπάθεια κάποιου να έχει πρόσβαση σε έναν web server host από το web server port (80) μπορεί να φαίνεται ως μία σχετικά ακίνδυνη δραστηριότητα, αλλά πολλαπλές προσπάθειες πρόσβασης είναι στην πραγματικότητα προμελετημένες επιθέσεις. Για παράδειγμα, μία πρόσβαση της μορφής "GET /../../../etc/passwd HTTP/1.0" είναι πιθανώς ένα κακό σημάδι και πρέπει να μπλοκαριστεί.
- Να αναγνωρίζουν προσπάθειες IP spoofing. Το πρωτόκολλο ARP το οποίο χρησιμοποιείται για την μετατροπή των IP διευθύνσεων σε MAC διευθύνσεις είναι συχνά στόχος επίθεσης. Στέλνοντας «πλαστά» ARP πακέτα πάνω από ένα Ethernet, ένας εισβολέας που απέκτησε πρόσβαση σε ένα σύστημα μπορεί να

προσποιηθεί ότι χειρίζεται κάποιο άλλο σύστημα. Το γεγονός αυτό μπορεί να προκαλέσει επιθέσεις άρνησης εξυπηρέτησης (DoS) διαφόρων ειδών, καθώς και system hijacking, όπου ένας σημαντικός server (όπως ο DNS server ή ο authentication server) έχουν υποστεί spoofing. Οι εισβολείς μπορούν να χρησιμοποιήσουν αυτό το spoofing ώστε να ανακατευθύνουν τα πακέτα προς το δικό τους σύστημα και να προετοιμάζουν επιθέσεις τύπου "man in the middle" στο δίκτυο αυτό, που παλαιότερα ήταν ασφαλές. Κρατώντας μία καταγραφή (register) των ARP πακέτων, ένα NIDS μπορεί να αναγνωρίσει τον προορισμό (διεύθυνση του Ethernet) του δικτύου που μεταφέρονται τα πακέτα και να ανακαλύψει τους εισβολείς.

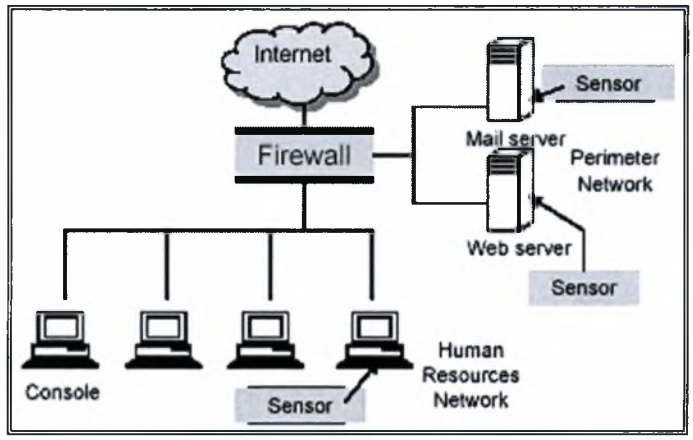
Όταν το NIDS ανιχνεύει ανεπιθύμητη δραστηριότητα αναλαμβάνει δράση. Δηλαδή, παρεμβαίνει όταν το δίκτυο το οποίο προστατεύει δεχτεί κίνηση από το δίκτυο του επιτιθέμενου. Σε αυτές τις περιπτώσεις είτε εκτελεί προκαθορισμένες ενέργειες είτε τροποποιεί τις ρυθμίσεις του firewall που υπάρχει στο δίκτυο ώστε να μπλοκάρει την εισερχόμενη κίνηση από τον υπολογιστή του επιτιθέμενου ή από το δίκτυό του.

2.6.2 Host Based Intrusion Detection Systems

Τα Host Based Intrusion Detection Systems (HIDS) δεν παρακολουθούν την κίνηση στο δίκτυο αλλά παρακολουθούν τι συμβαίνει σε μεμονωμένα μηχανήματα. Ανιχνεύουν ανωμαλίες σε κάποιο συγκεκριμένο μηχάνημα και προσδιορίζουν κατά πόσο η δραστηριότητα του συστήματος είναι αποδεκτή. Πιο συγκεκριμένα, παρακολουθούν τα security event logs και ελέγχουν για αλλαγές στο σύστημα, όπως για παράδειγμα αλλαγές σε κρίσιμα αρχεία συστήματος ή σε καταχωρίσεις συστήματος. Μπορούν να χωριστούν στις ακόλουθες δύο κατηγορίες [20]:

- **System integrity checkers** – Παρακολουθούν αρχεία συστήματος και καταχωρήσεις συστήματος για αλλαγές που έγιναν από τους εισβολείς. Υπάρχει μία πληθώρα από File/System integrity checkers, όπως το "Tripwire" ή το "LANguard File Integrity Checker".
- **Log file monitors** - Παρακολουθούν log αρχεία που δημιουργούνται από συστήματα υπολογιστών. Τα Windows NT/2000 και XP συστήματα δημιουργούν γεγονότα ασφαλείας (security events) όταν πραγματοποιούνται κρίσιμα γεγονότα στο μηχάνημα (για παράδειγμα όταν ένας χρήστης απαιτεί προνόμια που έχει ο

διαχειριστής του συστήματος). Ανακτώντας και αναλύοντας αυτά τα γεγονότα ασφαλείας ένα HIDS μπορεί να ανιχνεύει εισβολείς.



Τοπολογία HIDS

Μετά το firewall και το network monitor, το HIDS αποτελεί την τρίτη γραμμή άμυνας για ένα πακέτο δικτύου προτού φτάσει στον host που προορίζεται. Κατά [21] οι δύο βασικοί τύποι HIDS είναι οι ακόλουθοι:

- *Network monitors*: Αυτά παρακολουθούν τις εισερχόμενες συνδέσεις δικτύου προς τον host και προσπαθούν να προσδιορίσουν αν κάποιες από αυτές τις συνδέσεις αποτελούν απειλή. Να σημειωθεί ότι αυτό είναι διαφορετικό από το NIDS αφού το μόνο που εξετάζει είναι η κίνηση του δικτύου που εισέρχεται στον host πάνω στον οποίο τρέχει το HIDS και δεν εξετάζει όλη την κίνηση του δικτύου.
- *Host monitors*: Αυτά παρακολουθούν αρχεία, αρχεία συστήματος, logs, ή άλλα μέρη του host για ύποπτες δραστηριότητες οι οποίες μπορεί να αντιπροσωπεύουν μία προσπάθεια εισβολής (ή μία επιτυχημένη εισβολή). Αφού ανακαλυφτούν προβλήματα, στη συνέχεια ενημερώνονται οι διαχειριστές συστήματος για αυτά.

Να σημειωθεί ότι συνήθως χρησιμοποιείται συνδυασμός των δύο παραπάνω συστημάτων (NIDS - HIDS) για την επίτευξη καλύτερου αποτελέσματος.

2.6.3 Πλεονεκτήματα – Μειονεκτήματα Των NIDS Και HIDS

Στη συνέχεια δίνονται τα πλεονεκτήματα και τα μειονεκτήματα των Network Based Intrusion Detection Systems [1].

Τα πλεονεκτήματα τους είναι:

- Απαιτείται η τοποθέτηση λιγότερων αισθητήρων και επομένως το κόστος είναι χαμηλό
- Ανιχνεύουν επιθέσεις δικτυακής κίνησης
- Είναι δυσκολότερο για τον επιτιθέμενο να καλύψει τα ίχνη του
- Πραγματοποιείται real-time αντίδραση και ειδοποίηση
- Είναι ανεξάρτητο από το Λειτουργικό Σύστημα

Τα μειονεκτήματα τους είναι:

- ✘ Είναι κρίσιμη η τοποθέτησή τους και όχι πάντα εύκολη
- ✘ Δίκτυα τα οποία έχουν switch τα δυσκολεύουν
- ✘ Πρέπει να ελέγχουν μεγάλο πλήθος δεδομένων
- ✘ Είναι ευάλωτα σε DoS με πολλά «ύποπτα» πακέτα
- ✘ Δεν μπορούν να ελέγξουν κρυπτογραφημένα πακέτα

Τα πλεονεκτήματα και τα μειονεκτήματα των Host Based Intrusion Detection Systems δίνονται παρακάτω [1].

Τα πλεονεκτήματα τους είναι:

- Ελέγχουν αν η επίθεση ήταν επιτυχής
- Μπορούν να ελέγξουν συγκεκριμένες δραστηριότητες (σκληρό δίσκο, logon - logoff)
- Ανιχνεύουν τροποποιήσεις αρχείων και απόπειρες brute-force login.
- Μπορούν να ελέγξουν κρυπτογραφημένα δεδομένα
- Έχουν τη δυνατότητα να αντιδρούν σε real-time
- Χρησιμοποιούν λίγο ή καθόλου hardware (επιβαρύνει τον host)

Τα μειονεκτήματα τους είναι:

- ✘ Αδυνατούν να ελέγξουν host όπου δεν είναι εγκατεστημένα
- ✘ Πραγματοποιούν ανίχνευση από έμμεσες πηγές
- ✘ Παρουσιάζουν αδυναμία κάλυψης μεγάλων δικτύων

2.6.4 Άλλοι Τρόποι Κατηγοριοποίησης

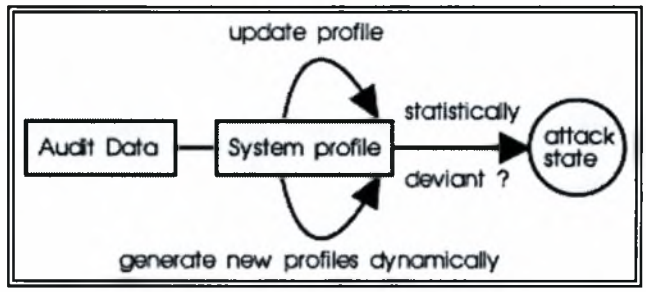
Στην πραγματικότητα υπάρχουν διάφοροι τρόποι κατηγοριοποίησης ενός IDS. Μερικοί από αυτούς παρατίθενται παρακάτω:

- **Misuse detection** vs. **Anomaly detection**: Στην ανίχνευση κακής συμπεριφοράς (misuse detection), το IDS αναλύει την πληροφορία που έχει και τη συγκρίνει με μία βάση δεδομένων που περιέχει ακολουθίες επιθέσεων που είναι ήδη γνωστές, δηλαδή έχουν συμβεί παλαιότερα. Στην ανίχνευση διαταραχών (anomaly detection), ο διαχειριστής του συστήματος καθορίζει μία λίστα με τα όρια της κίνησης του δικτύου, το μέγεθος που θα πρέπει συνήθως να έχουν τα πακέτα και άλλα. Έτσι το IDS συγκρίνει την πληροφορία που έχει συλλέξει με τη λίστα αυτή και ελέγχει για διαταραχές.
- **Network-based** vs. **host-based systems**: Αναλύθηκαν παραπάνω.
- **Passive system** vs. **Reactive system**: Σε ένα παθητικό (passive) σύστημα, το IDS ανιχνεύει μία πιθανή παραβίαση ασφάλειας, αποθηκεύει την πληροφορία και σημαίνει συναγερμό. Σε ένα αντιδραστικό (reactive) σύστημα, το IDS απαντά στην ύποπτη δραστηριότητα αποσυνδέοντας (log of) το χρήστη ή τροποποιεί τις ρυθμίσεις του firewall ώστε να μπλοκάρει την κίνηση από την κακόβουλη IP διεύθυνση.

Στη συνέχεια αναλύονται περαιτέρω η ανίχνευση διαταραχών και η ανίχνευση κακής συμπεριφοράς.

2.6.4.1 Ανίχνευση Διαταραχών

Στην ανίχνευση διαταραχών (anomaly detection) το σύστημα ανίχνευσης εισβολέων καταγράφει τις δραστηριότητες που πραγματοποιούν οι χρήστες στα συστήματα και δημιουργούν στατιστικά στοιχεία για τις δραστηριότητες αυτές βασισμένα στις καταγραφές που έχουν γίνει. Το σύστημα ανίχνευσης εισβολέων θεωρεί ως εισβολές τις δραστηριότητες που διαφέρουν από την κανονική χρήση που είχε ως τότε το σύστημα. Αν το IDS αντιληφτεί τέτοιες δραστηριότητες και γεγονότα σημάνει συναγερμό.



Ενδεικτικό σύστημα ανίχνευσης διαταραχών

Μερικές συνήθεις προσεγγίσεις ανίχνευσης διαταραχών είναι οι ακόλουθες [22]:

- *Statistical approaches*. Αρχικά δημιουργούνται τα προφίλ κανονικής συμπεριφοράς του συστήματος ενώ στην συνέχεια, τα IDS που χρησιμοποιούν τη μέθοδο αυτή, μαθαίνουν τη συμπεριφορά των χρηστών του συστήματος και αυτό μπορούν να το επιτύχουν σε ικανοποιητικότερο βαθμό από τον άνθρωπο - ειδικό. Το αρνητικό σημείο είναι ότι μπορούν να εκπαιδευτούν από τους εισβολείς και τελικώς ενέργειες που προκαλούν οι ίδιοι (επιθέσεις) να θεωρούνται φυσιολογικές.
- *Predictive pattern generation*. Η συγκεκριμένη μέθοδος προσπαθεί να προβλέψει μελλοντικά γεγονότα βασιζόμενη σε γεγονότα που έχουν ήδη πραγματοποιηθεί χρησιμοποιώντας μία βάση δεδομένων που ονομάζεται "rulebase", για παράδειγμα, E1 - E2 → (E3 = 80%, E4 = 15%, E5 = 5%)
- *Neural networks*. Με τη μέθοδο αυτή εκπαιδεύεται το νευρωνικό δίκτυο (neural network) ώστε να προβλέψει την επόμενη ενέργεια ή εντολή του χρήστη αφού όμως το νευρωνικό δίκτυο έχει στη διάθεση του τις n προηγούμενες ενέργειες ή εντολές που πραγματοποίησε ο χρήστης.

Τα συστήματα ανίχνευσης εισβολέων που χρησιμοποιούν τη μέθοδο ανίχνευσης διαταραχών έχουν τα πλεονεκτήματα ότι:

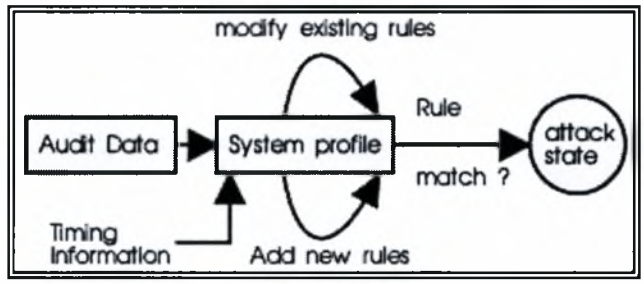
- Μπορούν να καταλάβουν οποιαδήποτε πιθανή επίθεση
- Μπορούν να αντιληφθούν επιθέσεις που δεν έχουν ξαναπραγματοποιηθεί στο παρελθόν
- Δεν απαιτούν συνεχή προστασία από hacking τεχνικές

Τα μειονεκτήματα τους είναι ότι:

- ✘ Προκαλούν πολλούς ψευδείς θετικούς και αρνητικούς συναγερμούς
- ✘ Απαιτούν ειδικούς για να ανακαλύψουν τι προκάλεσε το συναγερμό

2.6.4.2 Ανίχνευση Κακής Συμπεριφοράς

Στην ανίχνευση κακής συμπεριφοράς (misuse detection), το σύστημα ανίχνευσης εισβολέων αναγνωρίζει εισβολές που ακολουθούν συγκεκριμένα πρότυπα εισβολών (intrusion patterns). Τα πρότυπα αυτά περιέχονται στο σύστημα και κάθε ενέργεια που πραγματοποιείται συγκρίνεται με αυτά. Επίσης, καλούνται "expert systems".



Ενδεικτικό σύστημα ανίχνευσης κακής συμπεριφοράς

Μερικές συνήθεις προσεγγίσεις ανίχνευσης κακής συμπεριφοράς είναι οι ακόλουθες:

- *Expert systems*. Τα profiles αναβαθμίζονται σε περιοδικά διαστήματα και απαιτείται ένας ειδικός ασφάλειας που θα ενημερώνει το σύστημα με νέα δείγματα εισβολών.
- *Keystroke monitoring*. Είναι μία απλή τεχνική που παρακολουθεί το πάτημα των πλήκτρων για δείγματα εισβολών.
- *"Network grep"*. Παρακολουθεί ακολουθίες στις δικτυακές συνδέσεις που μπορεί να υποδεικνύουν μία επίθεση σε εξέλιξη.
- *Pattern matching*. Κωδικοποιεί ακολουθίες καταστάσεων που υπάρχουν όταν πραγματοποιείται μία επίθεση, για παράδειγμα: "change ownership of /etc/passwd" → "open/etc/passwd for write" → ***alert***

Τα συστήματα ανίχνευσης εισβολέων που χρησιμοποιούν τη μέθοδο κακής συμπεριφοράς έχουν τα πλεονεκτήματα ότι:

- Είναι εύκολο να υλοποιηθούν, να αναπτυχθούν και να αναβαθμιστούν
- Αντιδρούν γρήγορα
- Έχουν μικρό ποσοστό ψευδή θετικών συναγερμών

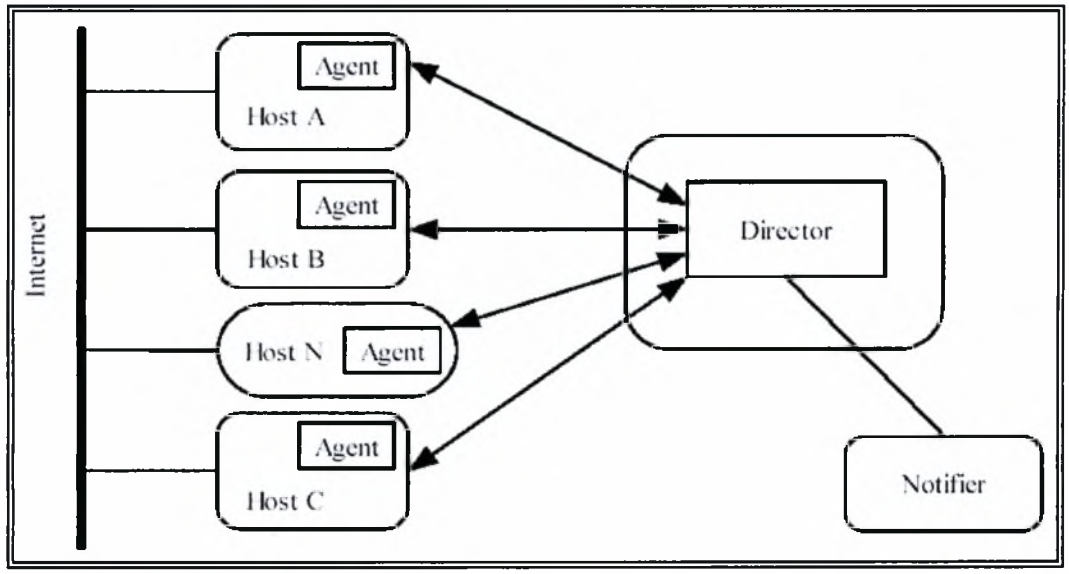
Τα μειονεκτήματα τους είναι ότι:

- ✘ Δεν μπορούν να αντιληφθούν επιθέσεις που δεν πραγματοποιήθηκαν παλαιότερα
- ✘ Απαιτείται συνεχή αναβάθμιση του συστήματος με νέα δείγματα επιθέσεων
- ✘ Θεωρείται σχετικά εύκολο να πέσουν θύματα των εισβολέων

2.7 Αρχιτεκτονική Των IDS

Στη συγκεκριμένη υποενότητα αναλύεται η αρχιτεκτονική των συστημάτων ανίχνευσης εισβολών. Ένα σύστημα ανίχνευσης εισβολέων αποτελείται από τρία μέρη [15]:

- Ο **αντιπρόσωπος (agent)** αντιστοιχεί στον logger. Αποκτά πληροφορίες από ένα στόχο, για παράδειγμα από ένα υπολογιστικό σύστημα.
- Ο **διευθυντής (director)** αντιστοιχεί στον αναλυτή. Αναλύει τα δεδομένα που προέρχονται από τους αντιπροσώπους και προσδιορίζει εάν μία επίθεση είναι σε εξέλιξη ή έχει ήδη συμβεί.
- Ο διευθυντής μεταδίδει την πληροφορία στον **αγγελιοφόρο (notifier)**. Ο αγγελιοφόρος αποφασίζει το χρόνο και τον τρόπο που θα ειδοποιήσει την αναγκαία οντότητα και επικοινωνεί με τους αντιπροσώπους για να ρυθμίσει θέματα εισαγωγής στοιχείων.



Αρχιτεκτονική ενός συστήματος ανίχνευσης εισβολέων

2.7.1 Ο αντιπρόσωπος (agent)

Ο αντιπρόσωπος παίρνει πληροφορίες από μία ή περισσότερες πηγές δεδομένων. Οι πηγές μπορεί να είναι αρχεία καταγραφής, διεργασίες ή ένα δίκτυο υπολογιστών. Ο αντιπρόσωπος έχει τη δυνατότητα αποστολής της πληροφορίας στο διευθυντή, αλλαγή της μορφής της πληροφορίας ώστε να διευκολύνει το διευθυντή άλλα και την απόρριψη πληροφοριών που θεωρεί ότι είναι άχρηστες.

Αν ο διευθυντής θεωρεί ότι χρειάζεται περισσότερη πληροφορία από κάποια πηγή τότε:

- καθοδηγεί τον αντιπρόσωπο να συλλέξει πρόσθετα στοιχεία
- καθοδηγεί τον αντιπρόσωπο να επεξεργαστεί με διαφορετικό τρόπο τα στοιχεία που συλλέγει
- ζητά περισσότερες πληροφορίες από ότι παλαιότερα αν ανιχνεύσει πιθανή επίθεση

Παρακάτω αναφέρονται τα είδη της πληροφορίας που μπορεί να συλλέξει ένας αντιπρόσωπος και τον τρόπο που αυτά μπορούν να συλλεχθούν.

- *Συλλογή Πληροφοριών Βασισμένη στον Υπολογιστή (Host-Based Information Gathering):* Οι αντιπρόσωποι που βασίζονται στον υπολογιστή συνήθως χρησιμοποιούν τα αρχεία καταγραφής του συστήματος και της εφαρμογής για να ενημερωθούν για τα γεγονότα που πραγματοποιήθηκαν και να τα αναλύσουν ώστε να αποφασίσουν την πληροφορία που πρέπει να μεταβιβάσουν στο διευθυντή. Τα αρχεία καταγραφής μπορεί, για παράδειγμα, να είναι αρχεία σχετιζόμενα με την ασφάλεια (όπως τα windows NT logs). Υπάρχουν περιπτώσεις που ο αντιπρόσωπος παράγει μόνος του τις πληροφορίες. Η διαδικασία αυτή πραγματοποιείται από τους ελεγκτές της πολιτικής (policy checkers), οι οποίοι συνήθως καταγράφουν τα αποτελέσματά τους σε ένα αρχείο καταγραφής και ο αντιπρόσωπος αναλύει το αρχείο αυτό.
- *Συλλογή Πληροφοριών Βασισμένη στο Δίκτυο (Network-Based Information Gathering):* Οι αντιπρόσωποι που βασίζονται στα δίκτυα καταγράφουν την κυκλοφορία που υπάρχει σε αυτά. Με τη συλλογή πληροφοριών βασισμένη στο δίκτυο παίρνουμε διαφορετικές πληροφορίες από ότι με τη συλλογή πληροφοριών βασισμένη στον υπολογιστή. Μπορούν να ανιχνευτούν επιθέσεις, όπως επιθέσεις άρνησης εξυπηρέτησης που προκαλούνται από υπερφόρτωση δικτύου. Επιπλέον, η τεχνική αυτή μπορεί να πραγματοποιήσει παρακολούθηση περιεχομένου της κυκλοφορίας που υπάρχει στους υπολογιστές του δικτύου, χρησιμοποιώντας τεχνικές παρακολούθησης δικτύου (network sniffing). Αν το δίκτυο έχει μικρό αριθμό υπολογιστών, οι αντιπρόσωποι πρέπει να ελέγχουν μόνο την κυκλοφορία μεταξύ αυτών των υπολογιστών. Στις περιπτώσεις στις οποίες οι υπολογιστές που ελέγχουν τα σημεία εισόδου πραγματοποιούν εκτενή καταγραφή της κυκλοφορίας

του δικτύου που λαμβάνουν, η συλλογή πληροφοριών βασισμένη στο δίκτυο έχει μειωμένη αποτελεσματικότητα σε σχέση με την προηγούμενη τεχνική.

- *Συνδυασμός Πηγών Πληροφοριών:* Απαιτείται η συγκέντρωση της πληροφορίας, ώστε ο αντιπρόσωπος να στείλει πληροφορίες στο διευθυντή για να μπορέσει ο διευθυντής με τη σειρά του να αναφέρει πιθανές επιθέσεις.

2.7.2 Ο διευθυντής (director)

Ο διευθυντής μπορεί να μειώσει την πληροφορία που δέχεται από το αρχείο καταγραφής, αν θεωρεί την πληροφορία που λαμβάνει περιττή, για να εξαλείψει τις περιττές εγγραφές. Ο διευθυντής με τη βοήθεια μίας μηχανής ανάλυσης προσδιορίζει αν μία επίθεση βρίσκεται σε εξέλιξη. Το πρόγραμμα αυτό εκτελείται συνήθως σε ένα ξεχωριστό σύστημα, μιας και ο ρόλος του διευθυντή είναι κρίσιμος για την αποτελεσματικότητα του συστήματος ανίχνευσης εισβολών.

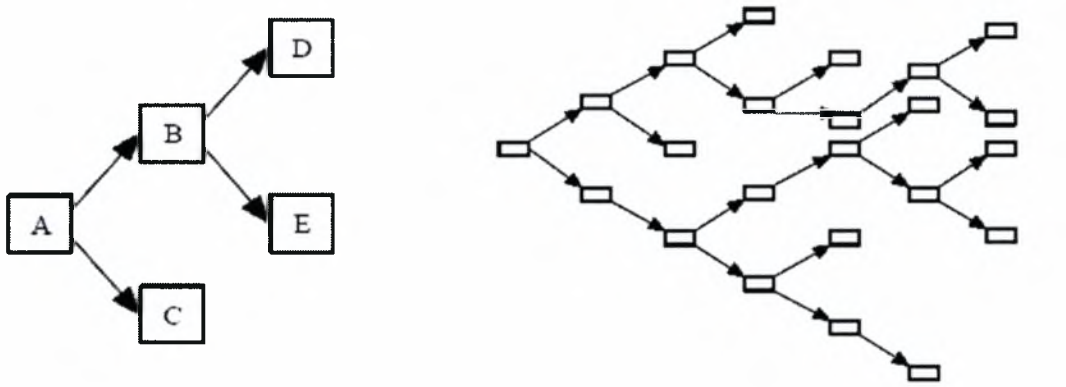
Υπάρχουν κάποιοι διευθυντές οι οποίοι αλλάζουν τους κανόνες που χρησιμοποιούν για τη λήψη αποφάσεων, δηλαδή προσθέτουν ή αφαιρούν κανόνες ή προσαρμόζονται στις αλλαγές των συστημάτων που παρακολουθούν. Οι διευθυντές αυτοί ονομάζονται προσαρμοστικοί (adaptive). Συνήθως, οι διευθυντές χρησιμοποιούν διάφορες τεχνικές ανάλυσης ώστε να ανακαλύψουν περισσότερες επιθέσεις που στοχεύουν το σύστημα.

2.7.3 Ο αγγελιοφόρος (notifier)

Ο αγγελιοφόρος αφού λάβει την πληροφορία από το διευθυντή δρομολογεί τις κατάλληλες ενέργειες. Ο αγγελιοφόρος είτε θα στείλει μία ειδοποίηση στον υπεύθυνο ασφαλείας ότι μία επίθεση βρίσκεται σε εξέλιξη είτε εκτελεί ενέργειες οι οποίες θα προστατεύσουν το σύστημα από τις επιθέσεις που δέχεται. Σε πολλά συστήματα ανίχνευσης εισβολών χρησιμοποιείται γραφικό περιβάλλον ώστε να είναι πιο εύκολο από το χρήστη να αντιληφθεί τις πληροφορίες που του μεταβιβάζονται για μία επίθεση που βρίσκεται σε εξέλιξη. Πάντως, είναι επιθυμητό να δίνεται στον υπεύθυνο ασφαλείας η πληροφορία αν υπάρχει πιθανότητα να είναι ψευδή το σήμα που δηλώνει την επίθεση.

Το σύστημα Graphical Intrusion Detection System – GrIDS [15] [23] παρουσιάζει την πρόοδο των επιθέσεων διαμέσου πολλαπλών συστημάτων χρησιμοποιώντας ένα περιβάλλον με γράφους. Στους γράφους οι υπολογιστές παρουσιάζονται ως κόμβοι οι

οποίοι συνδέονται με τις κορυφές ώστε να παρουσιάζεται η πρόοδος της επίθεσης από ένα σύστημα σε ένα άλλο. Το παρακάτω σχήμα απεικονίζει μία από τις οθόνες χρήστη του GrIDS και δείχνει την πρόοδο μιας επίθεσης ενός προγράμματος σκουληκιού (worm) καθώς εξελίσσεται μέσω ενός δικτύου.



Ο αγγελιοφόρος είτε θα στείλει ένα ηλεκτρονικό μήνυμα στην κατάλληλη οντότητα, είτε θα δημιουργήσει εγγραφές στα κατάλληλα αρχεία καταγραφής.

2.8 Αντιστοίχηση Υπογραφών Με Την Εισερχόμενη Κίνηση

Στη συνέχεια θα δούμε πως ένα δικτυακό Σύστημα Ανίχνευσης Εισβολέων αντιστοιχίζει τις υπογραφές με την εισερχόμενη κίνηση. Η κίνηση αποτελείται από IP δεδομένα που «ρέουν» μέσα σε ένα δίκτυο. Ένα δικτυακό IDS είναι ικανό να «συλλάβει» αυτά τα πακέτα καθώς ρέουν διαμέσου του καλωδίου. Ένα δικτυακό IDS αποτελείται από μία ειδική TCP/IP στοίβα η οποία συναθροίζει τα IP δεδομένα και τις TCP ροές. Έπειτα, εφαρμόζει μερικές από τις ακόλουθες τεχνικές [19]:

- **Protocol stack verification:** Ένας αριθμός εισβολών, όπως "Ping-O-Death" και "TCP Stealth Scanning", κάνουν παραβιάσεις στα IP, TCP, UDP, και ICMP πρωτόκολλα με στόχο να επιτεθούν στο μηχάνημα. Ένα απλό σύστημα επαλήθευσης (verification) μπορεί να εντοπίσει μη - έγκυρα (invalid) πακέτα.
- **Application protocol verification:** Ένας αριθμός εισβολών χρησιμοποιούν μη - έγκυρη συμπεριφορά πρωτοκόλλων, όπως "WinNuke", το οποίο χρησιμοποιεί μη - έγκυρο NetBIOS πρωτόκολλο ή DNS cache poisoning, το οποίο έχει έγκυρη μα ασυνήθιστη υπογραφή. Για να εντοπίσει αποτελεσματικά αυτού του είδους τις επιθέσεις ένα δικτυακό IDS θα πρέπει να ξαναυλοποιήσει μία πληθώρα

πρωτοκόλλων στο επίπεδο εφαρμογής ώστε να ανιχνεύσει ύποπτη ή μη - έγκυρη συμπεριφορά.

- **Creating new loggable events:** Ένα δικτυακό IDS μπορεί να χρησιμοποιηθεί για να προεκτείνει ικανότητες ελέγχου του λογισμικού διαχείρισης δικτύου. Για παράδειγμα, ένα δικτυακό IDS μπορεί να ενημερώσει όλα τα πρωτόκολλα στο επίπεδο εφαρμογής που χρησιμοποιούνται από ένα μηχάνημα. Κατεβάζοντας event log συστήματα (WinNT Event, UNIX syslog, SNMP TRAPS, κ.ά.) μπορεί να συσχετίσει αυτά τα events με άλλα events πάνω στο δίκτυο.

2.9 Απόκριση Στις Εισβολές

Αφού ανιχνευτεί μία εισβολή στο δίκτυο είναι ουσιώδες να ληφθούν ενέργειες οι οποίες θα προστατεύσουν το σύστημα. Ο βασικός στόχος είναι να αντιμετωπιστεί η επίθεση με τέτοιο τρόπο ώστε να ελαχιστοποιούνται οι απώλειες του συστήματος.

Όταν ένα δικτυακό IDS ανιχνεύσει μία επίθεση συμβαίνουν τα παρακάτω [19]:

- **Reconfigure firewall**

  Το IDS σχεδιάζει (configure) το firewall ώστε να φιλτράρει τις IP address του εισβολέα. Βέβαια, αυτό εξακολουθεί να επιτρέπει στον εισβολέα να επιτίθεται από άλλες διευθύνσεις. Σημείο ελέγχου για την υποστήριξη του firewall είναι το "Suspicious Activity Monitoring Protocol (SAMP)". Το σημείο ελέγχου έχει το "OPSEC" standard για τον επανασχεδιασμό του firewalls ώστε να μπλοκάρει την IP διεύθυνση από την οποία προέρχεται η επίθεση.

- **Chime**

  Beep ή παίξιμο ενός .wav αρχείου. Για παράδειγμα, μπορεί ο χρήστης να ακούσει το ηχογραφημένο μήνυμα: "You are under attack".

- **SNMP Trap**

  Το IDS στέλνει μία SNMP παγίδα δεδομένων σε μία κονσόλα διαχείρισης όπως η HP OpenView, η Tivoli, η Cabletron Spectrum κ.ά.

- **NT Event**

  Το IDS στέλνει ένα event στο WinNT event log.

- **syslog**

  Το IDS στέλνει ένα event στο UNIX syslog event σύστημα.

- **Send e-mail**

Το IDS στέλνει e-mail στον διαχειριστή για να τον ενημερώσει για την επίθεση.

- **Page**

  Σελιδοποιεί (page) το σύστημα διαχείρισης.

- **Log the attack**

  Αποθηκεύει τις πληροφορίες της επίθεσης (ώρα επίθεσης, IP διεύθυνση του εισβολέα, IP διεύθυνση/port του θύματος, πληροφορίες πρωτοκόλλου).

- **Save evidence**

  Αποθηκεύει τα ίχνη των πακέτων της επίθεσης σε αρχείο για μετέπειτα ανάλυση.

- **Launch program**

  Ενεργοποιεί ένα ξεχωριστό πρόγραμμα για να χειριστεί το συμβάν.

- **Terminate the TCP session**

  Παραποιεί ένα TCP FIN πακέτο για να αναγκάσει τον τερματισμό της σύνδεσης στο δίκτυο.

Κατά [24] ο χειρισμός των εισβολών περιλαμβάνει έξι φάσεις:

1. Προετοιμασία (preparation) για μία επίθεση.

   Αυτή η φάση εμφανίζεται πριν ανιχνευθούν οποιεσδήποτε επιθέσεις και εγκαθίστανται οι διαδικασίες και οι μηχανισμοί για την ανίχνευση και την απόκριση στις επιθέσεις.

2. Ταυτοποίηση (identification) μιας επίθεσης.

   Η φάση αυτή διαμορφώνει τις υπόλοιπες φάσεις.

3. Περιορισμός (containment) της επίθεσης.

   Η φάση αυτή προσπαθεί να εμποδίσει τον εισβολέα να αποκτήσει πρόσβαση στους πόρους του συστήματος, ώστε να περιορίσει τη ζημιά σε αυτό. Υπάρχουν δύο προσεγγίσεις:

   - *Ο περιορισμός της πρόσβασης* (constrain access) προκειμένου να αποτραπεί περαιτέρω ζημιά στο σύστημα (η ζημιά αναφέρεται σε οποιαδήποτε πράξη που αναγκάζει το σύστημα να παρεκκλίνει από μία ασφαλή ενέργεια)
   - *Η παθητική παρακολούθηση* (passive monitoring) της επίθεσης, που απλώς καταγράφει τις ενέργειες του επιτιθεμένου για μετέπειτα χρήση ενώ δεν παρεμβαίνουν στην επίθεση με κανένα τρόπο. Αυτή η τεχνική έχει περιορισμένη χρησιμότητα γιατί αν και θα αποκαλύψει πληροφορίες σχετικά με την επίθεση, αφήνει το σύστημα ευπαθές και ο επιτιθέμενος μπορεί να επιτεθεί και σε άλλα συστήματα.

4. Εξουδετέρωση (eradication) της επίθεσης.

   Σε αυτή τη φάση η επίθεση διακόπτεται και παρεμποδίζονται περαιτέρω παρόμοιες επιθέσεις. Στη φάση αυτή ο επιτιθέμενος δεν έχει πρόσβαση στο σύστημα και αυτό επιτυγχάνεται είτε με τον τερματισμό της σύνδεσης στο δίκτυο είτε με τον τερματισμό των διαδικασιών που σχετίζονται με την επίθεση. Το σημαντικό είναι ότι εγγυημένα η ίδια επίθεση δεν θα πραγματοποιηθεί ξανά άμεσα. Μία κοινή μέθοδος εφαρμογής της εξουδετέρωσης είναι η τοποθέτηση περιτυλιγμάτων (wrappers) γύρω από τους πιθανούς στόχους τα οποία εφαρμόζουν διάφορες μορφές ελέγχου πρόσβασης και μπορούν να ελέγξουν την πρόσβαση στα συστήματα είτε τοπικά είτε μέσω δικτύου.
5. Αποκατάσταση (recovery) από την επίθεση. Στη φάση αυτή αποκαθίσταται η ασφαλής κατάσταση στο σύστημα, σύμφωνα με την ισχύουσα πολιτική ασφάλειας.
6. Συνεχής παρακολούθηση (follow-up) της επίθεσης.

   Αυτή η φάση περιλαμβάνει τη λήψη μέτρων κατά του εισβολέα, τον προσδιορισμό των προβλημάτων κατά το χειρισμό του γεγονότος και καταγραφή των σχετικών εμπειριών που αποκτήθηκαν. Δύο γνωστές τεχνικές ανίχνευσης είναι:

   - *Η τεχνική αποτυπωμάτων* (thumbprinting) [25] [26]. Εκμεταλλεύεται τις συνδέσεις των υπολογιστών. Ένας επιτιθέμενος μπορεί ξεκινώντας από έναν υπολογιστή, να προσπελάσει πολλούς ενδιάμεσους, μέχρι να επιτεθεί στο στόχο του. Εάν κάποιος παρακολουθήσει τις συνδέσεις σε δύο οποιουσδήποτε υπολογιστές από τους οποίους διέρχονται συνδέσεις, το περιεχόμενο των συνδέσεων θα είναι το ίδιο. Με τη σύγκριση του περιεχομένου των συνδέσεων που περνούν μέσω των υπολογιστών, μπορεί να κατασκευαστεί η αλυσίδα των υπολογιστών που αποτελούν τις συνδέσεις.
   - *Η εξέταση δημιουργίας μιας IP επικεφαλίδας*. Μία εναλλακτική προσέγγιση είναι να εξεταστούν οι επικεφαλίδες αγνοώντας το περιεχόμενο των πακέτων. Τη λειτουργία αυτή επιτελεί η σήμανση επικεφαλίδων IP (IP header marking). Ένας δρομολογητής τοποθετεί πρόσθετες πληροφορίες στην επικεφαλίδα IP κάθε πακέτου για να υποδείξει την πορεία που έχει ακολουθήσει το πακέτο. Αυτές οι πληροφορίες μπορούν να εξεταστούν προκειμένου να εντοπιστεί η διαδρομή επιστροφής του πακέτου μέσω του Internet [27].

2.10 IDS Και Firewall

Τα Συστήματα Ανίχνευσης Εισβολέων δεν είναι αρκετά από μόνα τους να προστατεύσουν το δίκτυο. Απαιτούνται και άλλα συστήματα ασφάλειας, όπως firewall, κρυπτογραφία και αυθεντικοποίηση. Μάλιστα τα τελευταία θεωρούνται τα πρωτεύοντα συστήματα και τα IDS δευτερεύοντα που σχεδιάστηκαν ως backup για τα προηγούμενα.

Αξίζει να σημειωθεί ότι πολλές φορές υπάρχει σύγχυση νομίζοντας ότι το IDS και το firewall χρησιμοποιούνται για τον ίδιο σκοπό. Μία συνηθισμένη αντίληψη είναι ότι τα firewalls αναγνωρίζουν επιθέσεις και τις μπλοκάρουν. Αυτό δεν αληθεύει. Παρόλο που και τα δύο σχετίζονται με τη ασφάλεια του δικτύου, επιτελούν διαφορετικές λειτουργίες. Το firewall οριοθετεί την πρόσβαση μεταξύ δικτύων ώστε να αποτρέψει μία εισβολή και δεν μπορεί να εντοπίσει μία επίθεση που προέρχεται από το εσωτερικό του δικτύου. Τα firewalls είναι απλώς συσκευές που απενεργοποιούν (shut off) όλες τις υπόλοιπες και στη συνέχεια επιτρέπουν μόνο σε μερικές να λειτουργούν [26]. Αν τα συστήματα μπορούσαν από μόνα τους να «κλειδώνουν» ώστε να είναι ασφαλή, τα firewalls δεν θα χρειάζονταν. Ο λόγος, λοιπόν, που απαιτείται η χρήση τους είναι ότι αφήνονται, κατά λάθος ή τυχαία, τρύπες στην ασφάλεια. Όταν εγκαθίσταται ένα firewall, η πρώτη ενέργεια που κάνει είναι να σταματά όλες τις επικοινωνίες. Στη συνέχεια, ο διαχειριστής του firewall προσθέτει κανόνες οι οποίοι επιτρέπουν συγκεκριμένη κίνηση να περάσει από αυτό. Για παράδειγμα, ένα firewall μιας εταιρείας που επιτρέπει την πρόσβαση στο Internet σταματά όλη την UDP και ICMP κίνηση, τις εισερχόμενες TCP συνδέσεις μα επιτρέπει τις εξερχόμενες TCP συνδέσεις. Με τον τρόπο αυτό, σταματά όλες τις εισερχόμενες συνδέσεις που προέρχονται από hackers του Διαδικτύου, μα επιτρέπει τους εσωτερικούς χρήστες να συνδεθούν με το εξωτερικό δίκτυο. Συνεπώς, το firewall δεν έχει τη δυνατότητα να ανιχνεύσει αν κάποιος προσπαθεί να μπει στο εσωτερικό δίκτυο εκμεταλλευόμενος μία τρύπα μα αυτό που κάνει είναι να απαγορεύει πρόσβαση από προκαθορισμένα σημεία.

Αντίθετα ένα σύστημα ανίχνευσης εισβολέων έχει δυναμικότερο μηχανισμό άμυνας. Ένα IDS εκτιμά μία ύποπτη εισβολή (που το firewall δεν μπορεί να εντοπίσει) και σηματοδοτεί συναγερμό. Επίσης, ένα μεγάλο πλεονέκτημα των IDS είναι ότι ελέγχει

για επιθέσεις που προέρχονται από το εσωτερικό του δικτύου. Το 80% των οικονομικών απωλειών εξαιτίας του hacking προέρχεται από το εσωτερικό των δικτύων. Τα firewalls, τα οποία τοποθετούνται στην περίμετρο του δικτύου, δεν αντιλαμβάνονται τα γεγονότα που συμβαίνουν στο εσωτερικό του μα βλέπουν μόνο την κίνηση μεταξύ του εσωτερικού δικτύου και του Internet. Μερικοί λόγοι για τους οποίους πρέπει να προστεθεί IDS στο firewall ενός δικτύου είναι οι ακόλουθοι:

- Γίνεται διπλός έλεγχος σε περιπτώσεις που τα firewalls δεν έχουν προγραμματιστεί σωστά.
- Εντοπίζουν επιθέσεις τις οποίες τα firewalls θεωρούν νόμιμες και επιτρέπουν να περάσουν από αυτά, όπως επιθέσεις εναντίον των web servers.
- Εντοπίζουν επιθέσεις που ενώ τα firewalls έχουν προγραμματιστεί να τις εντοπίσουν απέτυχαν.
- Εντοπίζουν επιθέσεις που προέρχονται από το εσωτερικό του δικτύου.

Τα πρωτεύοντα συστήματα, που αναφέραμε στην πρώτη παράγραφο της συγκεκριμένης υποενότητας, μερικές φορές έχουν bugs ή λάθος ρυθμίσεις που μπορεί να οδηγήσουν σε προβλήματα, όμως οι βασικές τους λειτουργίες θα εκτελεστούν ορθά. Δεν συμβαίνει όμως το ίδιο με τα IDS, στα οποία αν γίνουν λάθος ρυθμίσεις ή περιέχουν bugs δεν εκτελούν τις ορθά τις λειτουργίες τους. Ένα άλλο πρόβλημα είναι ότι παρόλο που η κίνηση είναι φυσιολογική υπάρχει πιθανότητα το IDS να διαγνώσει ψεύτικο συναγερμό επίθεσης και τότε οι hackers έχουν τη δυνατότητα να το αποφύγουν ή να το εξουδετερώσουν. Αυτό βέβαια δεν σημαίνει ότι τα συστήματα ανίχνευσης εισβολέων δεν είναι έγκυρα. Το hacking είναι μία μεγάλη απειλή ενάντια στην ασφάλεια των δικτύων και για το λόγο αυτό οι άνθρωποι εγκαθιστούν τέτοια συστήματα πριν και μετά το firewall. Το σίγουρο είναι ότι καλά σχεδιασμένα IDS μπορούν να βελτιώσουν σημαντικά την ασφάλεια ενός site.

Επιπλέον, τα πολυάριθμα πρωτόκολλα που τα IDS αναλύουν τα αφήνουν απροστάτευτα σε πιθανή κατάρρευση όταν παρουσιαστεί απροσδόκητη κίνηση. Αυτοί που οργανώνουν μία επίθεση συχνά αγοράζουν τα ίδια IDS που χρησιμοποιούνται από το θύμα και πειραματίζονται πάνω σε αυτά για να βρουν ποια πακέτα είναι ικανά να το καταστρέψουν. Κατόπιν, κατά τη διάρκεια της επίθεσης, ο εισβολέας χρησιμοποιεί αυτά τα πακέτα και αφού καταστρέψει το IDS συνεχίζει χωρίς να ανιχνευτεί.

Αξίζει να σημειωθεί, ότι το Korea Computer Emergency Response Team and Coordination Center ανέφερε ότι οι ζημιές εξαιτίας του hacking αυξήθηκαν σημαντικά από το 1998: 1.943 προσπάθειες το 2000, 5.333 προσπάθειες το 2001 και 15.192 προσπάθειες το 2002. Βέβαια κατά αντιστοιχία και η αγορά των Συστημάτων Ανίχνευσης Εισβολέων αυξήθηκε με αλματώδεις ρυθμούς από 183 εκατ. $ το 2000 σε 422 εκατ. $ το 2002 με συνεχή παραγωγή νέων IDS προϊόντων.

2.11 Προβλήματα με τα NIDS

Υπάρχουν δύο βασικά προβλήματα με τα δικτυακά συστήματα ανίχνευσης εισβολέων. Το πρώτο είναι ότι υπάρχει διαθέσιμη ανεπαρκής πληροφορία η οποία διαβάζεται από τα πακέτα που μεταφέρονται μέσω του καλωδίου ώστε να αναδομηθούν σωστά τα γεγονότα που πραγματοποιούνται μεταξύ δύο μηχανημάτων που επικοινωνούν μεταξύ τους. Το δεύτερο είναι τα συστήματα ανίχνευσης εισβολέων είναι από τη φύση τους ευπαθή σε επιθέσεις άρνησης εξυπηρέτησης, όπως αναφέραμε και παραπάνω. Το πρώτο πρόβλημα μειώνει την ακρίβεια (accuracy) του συστήματος ενώ το δεύτερο διακινδυνεύει τη διαθεσιμότητά (availability) του [28].

2.11.1 Ανεπαρκής Πληροφορία

Ένα δικτυακό σύστημα ανίχνευσης εισβολέων «αιχμαλωτίζει» τα πακέτα που μεταφέρονται στο καλώδιο ώστε να καθορίσει τι ακριβώς συμβαίνει στο μηχάνημα που παρακολουθεί. Τα NIDS λειτουργούν προβλέποντας τη συμπεριφορά των δικτυωμένων μηχανημάτων η οποία εξαρτάται από τα πακέτα που ανταλλάσσουν μεταξύ τους.

Το πρόβλημα με αυτή τη τεχνική είναι ότι ένα παθητικό (passive) δικτυακό μέσο παρακολούθησης δεν μπορεί επακριβώς να προβλέψει αν ένα μηχάνημα που είναι συνδεδεμένο στο δίκτυο θα λάβει ένα πακέτο, το οποίο δεν πρόκειται να βλάψει το σύστημα. Για το λόγο αυτό το IDS πρέπει να «αιχμαλωτίσει» και να ελέγξει τα πακέτα.

Ένα NIDS είναι συνήθως εγκατεστημένο σε εντελώς διαφορετικό μηχάνημα από τα συστήματα που παρακολουθεί. Συχνά, το IDS βρίσκεται σε διαφορετικό σημείο στο δίκτυο. Το βασικό πρόβλημα που αντιμετωπίζει ένα NIDS είναι ότι αυτές οι διαφορές προκαλούν ασυνέπειες μεταξύ του συστήματος ανίχνευσης εισβολέων και του

μηχανήματος που παρακολουθεί. Αυτές οι ασυνέπειες είναι το αποτέλεσμα βασικών φυσικών διαφορών, όπως οι διαφορετικές υλοποιήσεις από ένα δίκτυο σε ένα άλλο. Για παράδειγμα, θεωρείστε ένα IDS και ένα τερματικό (end-system) τοποθετημένα σε διαφορετικά μέρη σε ένα δίκτυο. Τα δύο συστήματα θα λάβουν τα πακέτα που τους αντιστοιχούν σε διαφορετικές χρονικές στιγμές. Είναι σημαντική αυτή η διαφορά στο χρόνο επειδή κατά τη διάρκεια αυτής της αργοπορίας κάτι μπορεί να συμβεί στο τερματικό και μην λάβει το πακέτο. Παρόλα αυτά, το IDS είχε στείλει το πακέτο στον προορισμό του (στο τερματικό) πιστεύοντας ότι όλα λειτουργούν φυσιολογικά. Υποθέστε ένα IP πακέτο με κακό UDP checksum. Τα περισσότερα σύγχρονα λειτουργικά συστήματα δεν θα επέτρεπαν ένα τέτοιο πακέτο. Το IDS πρέπει να γνωρίζει αν κάθε σύστημα που παρακολουθεί θα δεχόταν ένα τέτοιο πακέτο. Μερικά λειτουργικά συστήματα θα δεχόταν ένα πακέτο που είναι προφανώς «κακό». Μία ανεπαρκής υλοποίηση, για παράδειγμα, επιτρέπει ένα IP πακέτο να έχει λανθασμένο checksum. Αν το IDS δεν το γνωρίζει αυτό, τότε απορρίπτει πακέτα τα οποία, λόγω υλοποίησης, το end-system δέχεται. Με τον τρόπο αυτό μειώνεται η ακρίβεια του συστήματος.

Ακόμη και αν το IDS γνωρίζει τι λειτουργικό σύστημα τρέχει σε κάθε μηχάνημα του δικτύου, δεν μπορεί να γνωρίζει, κοιτάζοντας ένα πακέτο, αν τελικά το εκάστοτε μηχάνημα θα το δεχθεί. Ένα μηχάνημα του οποίου έχει εξαντληθεί η μνήμη θα απορρίψει όλα τα εισερχόμενα πακέτα. Το IDS δεν μπορεί να καταλάβει ότι το τερματικό αντιμετωπίζει τέτοιο πρόβλημα και υποθέτει ότι το μηχάνημα έλαβε το πακέτο. Η κατάρρευση της CPU και ο κορεσμός του δικτύου σε ένα τερματικό μπορεί να προκαλέσουν τα ίδια προβλήματα.

Συγκεντρωτικά, όλα αυτά τα προβλήματα καταλήγουν σε καταστάσεις που το IDS συχνά δεν μπορεί να καθορίσει τις ασυνέπειες σε ένα πακέτο μόνο εξετάζοντας το. Απαιτείται να γνωρίζει τη δικτυακή συμπεριφορά των τερματικών τα οποία παρακολουθεί, καθώς και συνθήκες κίνησης των δικτύων στα οποία ανήκουν.

2.11.2 Ευπάθεια Στις DoS Επιθέσεις

Ένα IDS σύστημα μπορεί το ίδιο να δεχθεί DoS (Denial of Service) επιθέσεις. Οι DoS επιθέσεις έχουν ως στόχο να πλήξουν την διαθεσιμότητα του υπολογιστικού πόρου. Από την κατασκευή του, ένα IDS είναι ένα εξαιρετικά πολύπλοκο σύστημα,

ισοδύναμο με μία TCP/IP στοίβα που τρέχει υπηρεσίες. Αυτό σημαίνει ότι το IDS παρουσιάζει ευαισθησία σε DoS επιθέσεις όπως SYN floods και smurf attacks. Ειδικότερα, DoS επιθέσεις που πραγματοποιούνται σε IDS είναι οι ακόλουθες και έχουν ως στόχο:

- την εξάντληση των πόρων του συστήματος
- την εξάντληση των πόρων της CPU
- την εξάντληση της μνήμης
- το bandwidth του δικτύου
- την εκμετάλλευση αντιδραστικών IDS
- μικροεπιθέσεις ή λαθεμένα πακέτα

Οι DoS επιθέσεις μπορούν να προκαλέσουν τόσο την ανικανότητα των συστημάτων να παρέχουν τις υπηρεσίες του όσο και να συντελέσουν στην κατάρρευση αυτών. Όταν συζητά κανείς τη σχέση μεταξύ των DoS επιθέσεων και του ασφαλούς συστήματος προκύπτει η ερώτηση αν το σύστημα είναι "fail-open". Ένα "fail-open" σύστημα σταματά να παρέχει προστασία όταν καταρρέει από μία επίθεση τύπου DoS. Αντίθετα, ένα σύστημα "fail-closed" αφήνει το δίκτυο προστατευμένο όταν καταρρεύσει από επίθεση.

Οι όροι "fail-open" και "fail-closed" χρησιμοποιούνται συχνότερα για τα firewalls. Ένα fail-open firewall σταματά να ελέγχει την πρόσβαση στο δίκτυο όταν το ίδιο καταρρεύσει αλλά αφήνει το δίκτυο διαθέσιμο. Ένας επιτιθέμενος που μπορεί να καταρρεύσει ένα fail-open firewall μπορεί να το παρακάμψει εντελώς και να έχει πρόσβαση στο δίκτυο. Τα firewalls πρέπει να είναι τύπου "fail-closed" ώστε να μην επιτρέπουν την πρόσβαση στο δίκτυο αν αυτά καταρρεύσουν, διατηρώντας με τον τρόπο αυτό προστατευμένο.

Τα δικτυακά ID συστήματα είναι fail-open. Αν ένας εισβολέας καταρρεύσει ή εξαντλήσει τους πόρους του IDS μπορεί να επιτεθεί στο υπόλοιπο δίκτυο. Εξαιτίας της φανερής ευπάθειας των NIDS σε επιθέσεις άρνησης εξυπηρέτησης, είναι αδήριτη ανάγκη τα τελευταία να προστατευτούν όσο το δυνατόν καλύτερα ενάντια σε αυτού του είδους των επιθέσεων.

Δυστυχώς όμως, είναι αρκετά δύσκολο να πραγματοποιηθεί αυτό. Το πρόβλημα της εξάντλησης πόρων δεν λύνεται εύκολα και υπάρχουν πολλά διαφορετικά σημεία στα οποία οι πόροι ενός IDS μπορούν να καταναλωθούν. Μπορούν εύκολα να σχεδιαστούν επιθέσεις που καταστρέφουν ένα σύστημα ανίχνευσης εισβολέων άλλα η εύρεση όλων των ευπαθειών του, ώστε να επιτευχθεί η προστασία του, είναι δύσκολη υπόθεση.

2.12 Επιθέσεις

Στη συνέχεια αναλύονται δύο τύποι επιθέσεων ενάντια στα NIDS. Οι συγκεκριμένες επιθέσεις προσπαθούν έξυπνα να εμποδίσουν την ανάλυση πρωτοκόλλων, ώστε να μην πραγματοποιηθεί η αναγνώριση υπογραφής από το σύστημα που έχει ως στόχο την απόκτηση επαρκών πληροφοριών για την εξαγωγή συμπερασμάτων.

Σε όλες τις επιθέσεις υπάρχει ο εισβολέας που παραπλανεί τη χρήση του δικτύου ώστε να δημιουργήσει υπερβολική κίνηση πακέτων. Η πρώτη από αυτές που θα εξετάσουμε ονομάζεται επίθεση προσθήκης (insertion attack) και περιλαμβάνει έναν εισβολέα που γεμίζει το σύστημα με «έξυπνα» μη έγκυρα πακέτα. Η δεύτερη ονομάζεται επίθεση αποφυγής (evasion attack) και περιλαμβάνει την εκμετάλλευση των ασυνεπειών μεταξύ του αναλυτή και του τερματικού με στόχο να περάσουν τα πακέτα απαρατήρητα από τον αναλυτή.

2.12.1 Προσθήκη

Ένα IDS μπορεί να δεχτεί ένα πακέτο που το τερματικό απορρίπτει. Στην περίπτωση αυτή το IDS κάνει το λάθος να πιστεύει ότι το τερματικό δέχτηκε και προώθησε το πακέτο ενώ στην πραγματικότητα δεν το έκανε. Ένας εισβολέας μπορεί να εκμεταλλευτεί αυτή την κατάσταση στέλνοντας πακέτα σε ένα τερματικό, τα οποία αυτό θα τα απορρίψει. Ωστόσο, το IDS νομίζει ότι τα πακέτα είναι έγκυρα. Στην περίπτωση όμως αυτή, ο εισβολέας έχει τη δυνατότητα να προσθέτει επιπλέον δεδομένα τα οποία βλέπει το IDS (κανένα άλλο σύστημα στο δίκτυο δεν ενδιαφέρεται για τα «κακά» πακέτα).

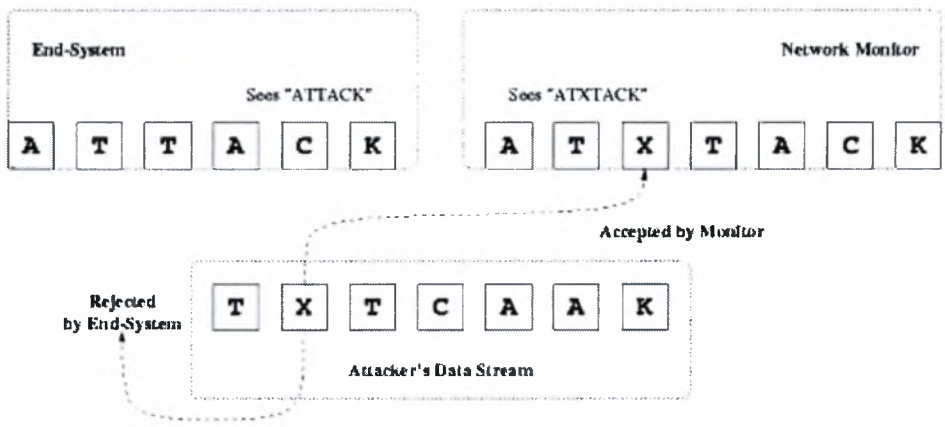
Αυτό ονομάζεται επίθεση προσθήκης (insertion attack) και οι συνθήκες που οδήγησαν σε τέτοιες επιθέσεις είναι οι επικρατέστερες ευπάθειες που παρουσιάζουν τα συστήματα ανίχνευσης εισβολέων στα οποία έγιναν δοκιμές σύμφωνα με [28]. Ο

επιτιθέμενος χρησιμοποιεί τις επιθέσεις προσθήκης ώστε να ματαιώσει την ανάλυση υπογραφής, επιτρέποντας έτσι τις επιθέσεις να περάσουν διαμέσου του IDS.

Για να γίνει κατανοητό γιατί οι επιθέσεις προσθήκης ματαιώνουν την αναγνώριση υπογραφής πρέπει να αναφέρουμε πως η τεχνική αυτή χρησιμοποιείται σε πραγματικά ID συστήματα. Στις περισσότερες περιπτώσεις η αναγνώριση υπογραφής χρησιμοποιεί pattern-matching αλγόριθμους για να ανιχνεύσει μία συγκεκριμένη ακολουθία (string) σε μία ροή δεδομένων. Για παράδειγμα, ένα IDS που προσπαθεί να ανιχνεύσει μία PHF επίθεση θα ψάξει για την ακολουθία "phf" σε μία HTTP "GET" αίτηση, η οποία είναι από μόνη της μία μεγάλη ακολουθία καθώς μπορεί να έχει τη μορφή "GET /cgi-bin/phf?".

Ένα IDS μπορεί εύκολα να ανιχνεύσει την ακολουθία "phf" σε μία HTTP αίτηση χρησιμοποιώντας μία απλή substring αναζήτηση. Παρόλα αυτά, το πρόβλημα είναι δυσκολότερο να λυθεί όταν ο επιτιθέμενος στείλει την ίδια αίτηση σε έναν web server, μα αναγκάσει το IDS να δει μία διαφορετική ακολουθία, όπως "GET /cgi-bin/pleasedontdetectthisforme?". Ο επιτιθέμενος χρησιμοποίησε επίθεση προσθήκης για να προσθέσει τα "leasedontdetectt", "is", και "orme" στην αρχική ακολουθία. Πλέον το IDS δεν μπορεί να ανιχνεύσει την ακολουθία "phf" από τη ροή δεδομένων που παρακολουθεί.

Η εικόνα που ακολουθεί απεικονίζει μία απλή τέτοια περίπτωση. Ο επιτιθέμενος αντιμετωπίζει το IDS με μία ροή πακέτων ενός χαρακτήρα (1-character packets) (τα αρχικά δεδομένα του επιτιθεμένου), στα οποία ένας χαρακτήρας – το γράμμα 'X' – θα γίνει αποδεχτό μόνο από το IDS. Ως αποτέλεσμα, το IDS και τερματικό ανασυγκροτούν δύο διαφορετικές ακολουθίες χαρακτήρων. Γενικά, οι επιθέσεις προσθήκης λαμβάνουν χώρα όταν ένα IDS είναι λιγότερο «αυστηρό» στην προώθηση πακέτου από ότι ένα τερματικό. Για να αντιμετωπιστεί αυτό θα πρέπει να υλοποιήσουμε το IDS έτσι ώστε να είναι όσο το δυνατό περισσότερο αυστηρό στην προώθηση πακέτων που διαβάζονται από το καλώδιο. Αυτό θα ελαχιστοποιούσε τον αριθμό των επιθέσεων προσθήκης. Παρόλα αυτά, εμφανίζεται ένα άλλο πρόβλημα (επιθέσεις αποφυγής) όταν ακολουθείται αυτή η τακτική.



Insertion of the letter 'X'

2.12.2 Αποφυγή

Ένα τερματικό μπορεί να αποδεχτεί ένα πακέτο που το IDS απέρριψε. Ένα IDS που λανθασμένα απέρριψε ένα τέτοιο πακέτο χάνει εντελώς τα περιεχόμενα του. Κάποιος κακόβουλος χρήστης μπορεί να εκμεταλλευτεί αυτή την κατάσταση, περνώντας σε πακέτα σημαντικές πληροφορίες που το IDS είναι πολύ αυστηρό για να προωθήσει. Αυτά τα πακέτα έχουν αποφύγει τον έλεγχο από το IDS.

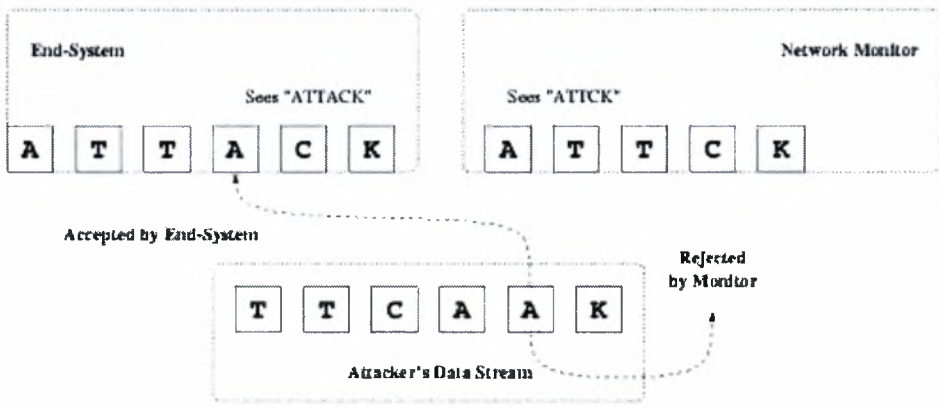
Αυτές οι επιθέσεις ονομάζονται αποφυγής (evasion attacks) και είναι ο ευκολότερος τρόπος με τον οποίο οι κακόβουλοι χρήστες μπορούν να εκμεταλλευτούν την ακρίβεια ενός IDS. Με τον τρόπο αυτό, μπορούν να περάσουν δεδομένα σε πακέτα αποφεύγοντας το IDS και τέτοιες επιθέσεις μπορούν να πραγματοποιηθούν ακόμη και αν χρησιμοποιούνται οι καλύτερες μηχανές ανάλυσης.

Οι επιθέσεις αποφυγής ματαιώνουν το ταίριασμα των προτύπων (pattern matching) με τρόπο παρόμοιο με τις επιθέσεις προσθήκης. Ομοίως, ο επιτιθέμενος προκαλεί το IDS να δει μία διαφορετική ροή δεδομένων από το τερματικό. Στην περίπτωση αυτή, ωστόσο, το τερματικό βλέπει περισσότερα από το IDS και η πληροφορία που χάνει το IDS είναι σημαντική για την ανίχνευση της επίθεσης.

Στην επίθεση προσθήκης που αναφέραμε παραπάνω, ο επιτιθέμενος στέλνει μία HTTP αίτηση, μα μπερδεύει τα περιεχόμενα της, που περνούν από το IDS, με επιπρόσθετα δεδομένα τα οποία δείχνουν την αίτηση έγκυρη και αβλαβή. Σε μία επίθεση αποφυγής, ο επιτιθέμενος στέλνει τμήματα της ίδιας αίτησης σε πακέτα τα οποία το IDS λανθασμένα απορρίπτει. Σε αυτή την περίπτωση το IDS δεν βλέπει όλα

τα κομμάτια από τη ροή δεδομένων γιατί κάποια αφαιρούνται. Για παράδειγμα, η αρχική αίτηση μπορούσε να γίνει "GET/gin/f", που δεν θα είχε κανένα νόημα για τα περισσότερα ID συστήματα.

Στην ακόλουθη εικόνα φαίνεται μία επίθεση αποφυγής.



Evasion of the letter 'A'

2.13 Είδη IDS Που Χρησιμοποιούνται

Διατίθενται τόσο ελευθέρα όσο και εμπορικά IDS συστήματα. Μερικά ελευθέρα προϊόντα είναι το Bro, NID, το Snort. Το ανοιχτό σύστημα προσδιορισμού εισβολέων Snort (http://www.snort.org), το οποίο αποτελεί μία αξιόλογη λύση IDS, έχει γίνει πολύ διαδεδομένο τα τελευταία χρόνια. Είναι ικανό να παρουσιάζει real-time ανάλυση κίνησης και logging πακέτων σε IP δίκτυα. Επίσης, παρουσιάζει ανάλυση πρωτοκόλλων, αναζήτηση περιεχομένου και μπορεί να χρησιμοποιηθεί για να ανιχνεύσει μία πληθώρα επιθέσεων όπως, buffer overflows, stealth port scans, CGI επιθέσεις, SMB probes, OS fingerprinting attempts. Έχει τρεις βασικές χρήσεις: σαν sniffer, σαν packet logger (χρήσιμο για traffic debugging), ή σαν πλήρες σύστημα ανίχνευσης εισβολέων. Το Snort θα αναλυθεί διεξοδικότερα στην τέταρτη ενότητα.

Στη συνέχεια ακολουθεί μία λίστα από εμπορικά συστήματα IDS που χρησιμοποιούνται:

- Real Secure by ISS, που είναι πολύ δημοφιλές. http://www.iss.net/products_services/enterprise_protection/
- CyberCop Monitor by Network Associates, Inc., το οποίο είναι ένας συνδυασμός host και network based IDS και αναλύει την κίνηση από και προς το host.

http://www.nai.com

- VCC/Tripwire™
  http://www.tripwire.com/
- CMDS (Computer Misuse and Detection System) by SAIC
  http://www.intrusion.com/
- INTOUCH NSA (Network Security Agent) by TTI
  http://www.ttinet.com/tti/nsa_www.html
- Symantec Intruder Alert by Symantec
  http://enterprisesecurity.symantec.com/products/products.cfm?ProductID=48
- POLYCENTER Security Intrusion Detector by TTI
  http://www.ttinet.com/products.html
- BlackICE by Network ICE, το οποίο έχει πολλές εκδόσεις
  http://www.digitalriver.com/dr/v2/ec_dynamic.main?SP=1&PN=10&sid=26412
- NetRanger by WheelGroup/Cisco
  http://newsroom.cisco.com/dlls/fspnisapi32b3.html
- eTrust Intrusion Detection από την Computer Associates
  http://www3.ca.com/Solutions/Solution.asp?ID=271
- NetProwler by Axent
  http://www.c2000.com/products/sec_netp.htm
- Centrax by Cybersafe
  http://www.cybersafe.com/solutions/centrax.html.
- NFR by Network Flight Recorder
  http://www.nfr.net/
- Dragon by Security Wizards
  http://www.network-defense.com/

2.14 Το Μέλλον Των IDS

Τα συστήματα ανίχνευσης εισβολέων είναι μία σχετικά νέα περιοχή στον τομέα της ασφάλειας των πληροφοριών και των συστημάτων, αφού μόλις λίγα χρόνια πριν έγινε αντιληπτή η σημαντικότητα της.

Οι τωρινές λύσεις και η τεχνολογία των συστημάτων ανίχνευσης εισβολέων δεν ανταποκρίνονται επακριβώς στις απαιτήσεις ασφάλειας που έχουν τα δίκτυα

παγκοσμίως και είναι αποδεκτό ότι υπάρχουν πολλά πράγματα που πρέπει να υλοποιηθούν ώστε να ανιχνεύονται στο αρχικό στάδιο οι επιθέσεις που στοχεύουν σε κάποιο δίκτυο, προτού αυτό να αντιμετωπίσει απώλειες.

Σύμφωνα με [29], [30] το μέλλον των IDS βασίζεται στη συσχέτιση δεδομένων (correlation data). Τα μελλοντικά IDS θα παράγουν αποτελέσματα εξετάζοντας εισόδους από πολλές διαφορετικές πηγές. Το λογισμικό θα συσχετίζει δεδομένα και θα τα μεταφράζει βασιζόμενο σε τεχνολογία τεχνητής νοημοσύνης (AI technology).

Τα μελλοντικά IDS θα συγχωνέψουν όλα τα ανεξάρτητα δικτυακά συστατικά και εργαλεία που υπάρχουν σήμερα σε ένα πλήρες και συνεργάσιμο σύστημα, το οποίο θα έχει στόχο να διατηρήσει τα δίκτυα «σταθερά» (stable). Θα υπάρχουν πολλά κατανεμημένα στοιχεία που θα έχουν συγκεκριμένα καθήκοντα και το καθένα θα δίνει τα αποτελέσματα του σε ένα υψηλότερο επίπεδο συσχέτισης και ανάλυσης.

Ο τομέας της ασφάλειας απέκτησε πλέον, τόσο από τους χρήστες όσο και από τους μηχανικούς, το ενδιαφέρον που του αντιστοιχεί και συνεπώς το μέλλον των συστημάτων ανίχνευσης εισβολέων διαφαίνεται λαμπρό και ελπιδοφόρο.

2.15 Συμπέρασμα

Καθώς αυξάνεται ο αριθμός των επιθέσεων και απειλείται όλο και περισσότερο η ασφάλεια των υπολογιστικών συστημάτων, τα συστήματα ανίχνευσης εισβολέων αποτελούν απαραίτητο εργαλείο για την προστασία των δικτύων. Τα IDS σε συνεργασία με άλλα εργαλεία ασφάλειας, όπως τα firewalls, επιτρέπουν την πλήρη παρακολούθηση των δραστηριοτήτων που λαμβάνουν χώρα σε ένα δίκτυο.

Όσο δύσκολη και να θεωρείται η προστασία των συστημάτων, χρησιμοποιώντας τα πιο αναβαθμισμένα (up to date) εργαλεία που διατίθενται είναι εφικτή η προστασία ενάντια σε κάθε είδος απειλή που θεωρείται γνωστή. Δυστυχώς, νέες απειλές και τρύπες στην ασφάλεια σε πακέτα λογισμικού ή υπολογιστικά συστήματα ανακαλύπτονται σε καθημερινή βάση.

Είναι σημαντικό να γνωρίζει κανείς ποιοι είναι οι τύποι των απειλών που μπορεί να αντιμετωπίσει στο περιβάλλον που δουλεύει. Θα πρέπει να αναγνωρίζει ποιες είναι οι πιθανές τρύπες ασφαλείας στο σύστημα και να φροντίσει να αποτρέψει τις επιθέσεις

ενάντια σε αυτές. Για παράδειγμα, ένας web server που συνδέεται στο Internet και τοποθετείται πίσω από ένα firewall μπορεί να είναι ασφαλής από πολλές επιθέσεις μα ένα CGI πρόγραμμα του server μπορεί να εκθέσει μία αδυναμία. Ένα πρόγραμμα ανίχνευσης εισβολέων μεταξύ του firewall και του web server μπορεί να απορρίψει όλες τις κινήσεις που θεωρεί ύποπτες.

Ενότητα

3

ΚΟΙΝΩΝΙΚΗ ΜΗΧΑΝΙΚΗ

Όπως αναφέρθηκε τα Συστήματα Ανίχνευσης Εισβολέων καλούνται να αντιμετωπίσουν διάφορους τύπους εισβολών και επιθέσεων. Ένα από τους τύπους επιθέσεων είναι αυτές που περιλαμβάνονται στο φαινόμενο της κοινωνικής μηχανικής. Στη συγκεκριμένη ενότητα αναλύεται ο όρος «κοινωνική μηχανική», αναφέρονται παραδείγματα του φαινομένου, εξηγούνται οι τεχνικές κοινωνικές μηχανικής καθώς και οι ενέργειες που μπορούν να γίνουν για την προστασία από αυτήν. Στη συνέχεια αναλύεται το φαινόμενο κοινωνική μηχανική σε σχέση με τα συστήματα ανίχνευσης εισβολέων, τους τρόπους με τους οποίους τα παρακάμπτει και τέλος αναφέρονται συστήματα IDS τα οποία εντοπίζουν επιθέσεις κοινωνικής μηχανικής.

3.1 Εισαγωγή

Ο όρος «κοινωνική μηχανική» (social engineering) αναφέρεται σε μέθοδο που χρησιμοποιείται για εξασφάλιση τρόπου διείσδυσης σε συστήματα και δίκτυα ηλεκτρονικών υπολογιστών με δόλιο τρόπο με στόχο στην παράνομη απόκτηση δεδομένων μέσω της εξαπάτησης: Ο χρήστης-θύμα πείθεται είτε να αποκαλύψει ευαίσθητες πληροφορίες, όπως κωδικούς, είτε να προβεί σε ενέργειες που ανοίγουν την πόρτα στον εισβολέα.

Σε αντίθεση με τους σχετικά γνωστούς τεχνικούς τρόπους, όπου κάποιος hacker προσπαθεί να σπάσει την ασφάλεια ενός συστήματος χρησιμοποιώντας τρωτά σημεία της τεχνολογίας, αυτός ο τρόπος βασίζεται, κατά κύριο λόγο, στην εξασφάλιση των αναγκαίων στοιχείων πρόσβασης, εκμεταλλευόμενος τρωτά σημεία της ανθρώπινης φύσης, όπως τη ματαιοδοξία, το φόβο, την καλόπιστη εμπιστοσύνη [6]. Συνήθης τακτική είναι η μέσω ψευδών παραστάσεων εξασφάλιση της εμπιστοσύνης κάποιου υπαλλήλου που έχει δικαίωμα πρόσβασης σε ένα σύστημα, με σκοπό την αποκάλυψη

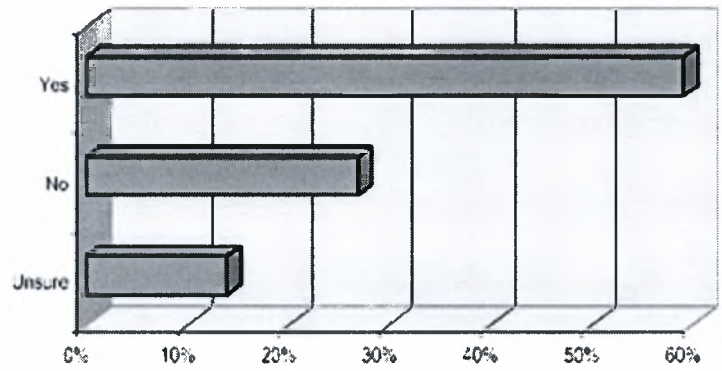
από τον εν λόγω υπάλληλο πληροφοριών που μπορεί να οδηγήσουν στην εύκολη παραβίαση των πυλών ασφάλειας του συστήματος. Ο Kevin Mitnick, ο πιο διάσημος hacker στον κόσμο, ανέφερε ότι: *«...βασική αρχή της Κοινωνικής Μηχανικής είναι η δημιουργία ψυχολογικής σύνδεσης με το άτομο στην άλλη πλευρά της (τηλεφωνικής) γραμμής, με σκοπό τη δημιουργία κλίματος εμπιστοσύνης... κατάσταση την οποία αργότερα να εκμεταλλευτείς...»*.

Οι κοινωνικοί μηχανικοί χρησιμοποιούν την επιρροή και την πειθώ τους για να εξαπατήσουν τα θύματά τους, είτε πείθοντάς τα ότι η ταυτότητά τους είναι άλλη απ' την πραγματική είτε οδηγώντας τα σε ανεπίτρεπτες πράξεις. Τελικός σκοπός του κοινωνικού μηχανικού είναι η απόκτηση ευαίσθητων πληροφοριών με ή χωρίς τη χρήση της τεχνολογίας [7]. Για παράδειγμα, στην πιο απλή της μορφή, η τακτική αυτή μπορεί να είναι ένα τηλεφώνημα από τον κακόβουλο χρήστη σε έναν υπάλληλο, στον οποίον αναφέρει ότι είναι μηχανικός από το τμήμα πληροφορικής και ότι χρειάζεται επειγόντως τους κωδικούς πρόσβασής του για την επιδιόρθωση ενός σοβαρού προβλήματος που εμποδίζει μία βασική λειτουργία, κατάσταση για την οποία ο διευθυντής είναι έξω φρενών.

Η κοινωνική μηχανική εκμεταλλεύεται την ανθρώπινη φύση και την ταυτόχρονη έλλειψη σωστής εκτίμησης της αξίας που έχουν οι πληροφορίες στην εποχή της Κοινωνίας της Πληροφορίας. Έτσι, οδηγεί σε έλλειψη επαρκούς προσοχής στη φύλαξη και ασφάλειά τους. Στο προαναφερόμενο παράδειγμα, ένας υπάλληλος μπορεί εύκολα να έδινε τους κωδικούς πρόσβασής του στον hacker ακόμα και από το τηλέφωνο, ενώ αν κάποιος του ζητούσε το κλειδί ενός χρηματοκιβωτίου με την πρόφαση ότι ήταν μηχανικός ελέγχου, ο υπάλληλος δύσκολα θα το αποχωριζόταν πριν ελέγξει την αυθεντικότητα της περίπτωσης, παρόλο που συγκριτικά η αξία του περιεχομένου του χρηματοκιβωτίου μπορεί να ήταν ελάχιστη.

Ο Mitnick επισημαίνει ότι ο ανθρώπινος παράγοντας παίζει πάντα ρόλο στην προστασία και τον έλεγχο της πληροφορίας, αλλά η γνώση της απειλής και η τήρηση των διαδικασιών μετριάζει τον κίνδυνο. Η προστασία ενός οργανισμού από τους κινδύνους που η κοινωνική μηχανική επιφυλάσσει, ξεκινά από την επιμόρφωση του προσωπικού του σχετικά με την πραγματική αξία της πληροφορίας, την ενημέρωσή του για τις τεχνικές παραπλάνησης που μπορεί να χρησιμοποιηθούν και την

ταυτόχρονη εκπαίδευσή του σε μεθόδους προστασίας για διαφύλαξη του απορρήτου των πληροφοριών. Αξίζει να σημειωθεί ότι, παρόλο που η κοινωνική μηχανική ως όρος δεν είναι ευρέως γνωστός, οι ειδικοί ασφάλειας ανά τον κόσμο συμφωνούν ότι η μέθοδος αυτή είναι και θα παραμείνει ίσως ο μεγαλύτερος κίνδυνος για τα συστήματα ασφάλειας της πληροφορικής. Στο ακόλουθο διάγραμμα φαίνεται, σύμφωνα με τη μελέτη του CSO's SecuritySensor VIII κατά τη διάρκεια του τριών τελευταίων μηνών του 2004, ότι το 60% των οργανισμών που έλαβαν μέρος στην έρευνα ανέφεραν περιστατικά με κατασκοπευτικό λογισμικό (spyware) [38].



Προβλήματα που αντιμετώπισαν οι παραπάνω οργανισμοί	
Διακοπή λειτουργίας του συστήματος	92%
Νόμιμες αποκαλύψεις	11%
Απώλεια ή φθορά εσωτερικών αρχείων	10%
Οικονομικές απώλειες	8%
Αποκαλύψεις αρχείων των εργαζομένων	6%
Αποκαλύψεις εμπιστευτικών αρχείων	6%
Αποκαλύψεις αρχείων των πελατών	6%
Κλοπή πνευματικής ιδιοκτησίας	5%
Απώλεια άξιας της μετοχής	2%

Πάντως, όπως αναφέρει ο Mitnick, η ασφάλεια των συστημάτων είναι υπόθεση ισορροπιών. Πολύ λίγη ασφάλεια κάνει το σύστημα ευάλωτο ενώ υπερβολική ασφάλεια εμποδίζει την ομαλή ροή των εργασιών και σε τελευταία ανάλυση των λειτουργιών τις οποίες είναι σχεδιασμένο το σύστημα να εκτελεί. Η πρόσκληση είναι η επίτευξη της σωστής ισορροπίας μεταξύ ασφάλειας και παραγωγικότητας. Και δεν θα πρέπει να ξεχνάμε ότι η σημαντικότερη από όλες τις απειλές είναι η ανθρώπινη απάτη που αποτελεί τον πιο αδύναμο κρίκο στην αλυσίδα της ασφάλειας.

Η κοινωνική μηχανική διακρίνεται στις δύο παρακάτω μορφές:

- $1^{η}$ περίπτωση: Ο χρήστης-θύμα πείθεται να αποκαλύψει ευαίσθητες πληροφορίες, όπως κωδικούς, μετά από την επικοινωνία που έχει μαζί του ο hacker
- $2^{η}$ περίπτωση: Ο χρήστης-θύμα δίχως να το αντιληφθεί προβαίνει σε ενέργειες που ανοίγουν την πόρτα στον εισβολέα

Στη συγκεκριμένη μελέτη (στη ενότητα της υλοποίησης) θα ασχοληθούμε με τη δεύτερη περίπτωση και τον τρόπο που τα συστήματα ανίχνευσης εισβολέων είναι ικανά να ανιχνεύσουν και να αντιμετωπίσουν τέτοιου είδους επιθέσεις.

3.2 Ορισμός

Σύμφωνα με το ορισμό που περιέχεται στο λεξικό της Merriam Webster, κοινωνική μηχανική είναι «η διαχείριση των ανθρώπων σε συμφωνία με τη θέση τους και τη λειτουργία στην κοινωνία, εφαρμόζοντας κοινωνική επιστήμη». Η κοινωνική μηχανική χρησιμοποιεί τις ανθρώπινες σχέσεις για να πετύχει ένα στόχο. Στον τομέα των πληροφοριακών συστημάτων η κοινωνική μηχανική είναι η μέθοδος που χρησιμοποιείται για εξασφάλιση τρόπου διείσδυσης σε συστήματα και δίκτυα ηλεκτρονικών υπολογιστών με δόλιο τρόπο με στόχο στην παράνομη απόκτηση δεδομένων μέσω της εξαπάτησης.

Πολλοί υποστηρίζουν ότι η κοινωνική μηχανική είναι ένα ταλέντο ή ένα προσόν που δεν το έχουν όλοι. Οι περισσότεροι άνθρωποι έχουν «εκπαιδευτεί» να είναι κοινωνικοί μηχανικοί από πολύ μικρή ηλικία. Όπως τα παιδιά, οι άνθρωποι έχουν μάθει να παίρνουν αυτό που επιθυμούν χρησιμοποιώντας τεχνικές κοινωνικής μηχανικής. Με λίγο σκέψη και μικρή προσπάθεια η κοινωνική μηχανική είναι ένας εύκολος και αποτελεσματικός τρόπος για έναν κακόβουλο χρήστη ώστε να κάνει δύσκολη τη ζωή ενός οργανισμού. Η κοινωνική μηχανική δεν απαιτεί πολλές τεχνικές γνώσεις μα βασίζεται στις κοινωνικές ικανότητες. Ο hacker, αντί να προσπαθεί για ώρες να ανακαλύψει κάποιο κωδικό, μπορεί κατευθείαν να τηλεφωνήσει σε έναν υπάλληλο μιας εταιρείας και προσποιούμενος ότι είναι υπάλληλος του τμήματος τεχνικής υποστήριξης, να του το ζητήσει. Μερικά ενδεικτικά παραδείγματα κοινωνικής μηχανικής φαίνονται στον ακόλουθο πίνακα [31].

Τύπος	Περιγραφή
Friendships	Εκμεταλλεύεται την εμπιστοσύνη μεταξύ φίλων
E-mail	Τα Forum χρησιμοποιούνται για να εκμεταλλεύονται την εμπιστοσύνη των ανθρώπων διαδίδουν συγκεκριμένους τύπους επιθέσεων
Dumper Diving	Τεχνική κατά την οποία οι επιτιθέμενοι χρησιμοποιούν έγγραφα που δεν χρειάζονται οι χρήστες, για την απόκτηση πληροφοριών
Office Snooping	Τεχνική που απαιτεί snooping και ακάλυπτες «πόρτες»
Trust	Η κοινωνική μηχανική εκμεταλλεύεται την εμπιστοσύνη των ανθρώπων
Time	Οι εισβολείς έχουν αυτό το χαρακτηριστικό στο πλευρό τους

Οι τεχνικές κοινωνικής μηχανικής χωρίζονται σε δύο κατηγορίες [34]:

- Η απάτη επιτελείται βασιζόμενη σε υπολογιστή ή τεχνολογία. Παραπλανεί το χρήστη κάνοντας του να πιστεύει ότι αλληλεπιδρά με ένα «πραγματικό» σύστημα υπολογιστή και αφού ο χρήστης το εμπιστευτεί τότε του παρέχει εμπιστευτικές πληροφορίες. Για παράδειγμα, ο χρήστης βλέπει ένα popup παράθυρο που τον ενημερώνει ότι μία εφαρμογή του υπολογιστή παρουσιάζει πρόβλημα και ο χρήστης πρέπει να επαληθεύσει τα στοιχεία του για να συνεχίσει. Μόλις ο χρήστης πληκτρολογεί το όνομα χρήστη και τον κωδικό τότε ο hacker βλέπει τα στοιχεία αυτά και αποκτά πρόσβαση στο δίκτυο και το σύστημα.
- Η απάτη επιτελείται βασιζόμενη σε ανθρώπους. Εκμεταλλεύεται την άγνοια του θύματος και τη φύση του ανθρώπου που βοηθά τους συνανθρώπους τους αν αντιμετωπίζουν ένα πρόβλημα. Για παράδειγμα, όταν ο κακόβουλος χρήστης τηλεφωνεί στην τεχνική υποστήριξη παριστάνοντας το διευθυντή και ζητά τον κωδικό του τον οποίο έχει ξεχάσει.

3.3 Παραδείγματα

Όπως αναφέρθηκε παραπάνω πολλές φορές ο χρήστης-θύμα δίχως να το αντιληφθεί προβαίνει σε ενέργειες που ανοίγουν την πόρτα στον εισβολέα. Ένα παράδειγμα που περιλαμβάνεται σε αυτή τη περίπτωση είναι το ακόλουθο.

Ένας κακόβουλος χρήστης στέλνει ένα e-mail σε έναν πελάτη της PayPal, μία εταιρεία που προσφέρει ένα γρήγορο και άνετο τρόπο για να κάνει πληρωμές μέσω του Internet [7]. Το e-mail, που υποτίθεται ότι προερχόταν από τη PayPal, ζητούσε από τον πελάτη να επισκεφτεί έναν ασφαλή δικτυακό τόπο της εταιρείας και να ενημερώσει την καρτέλα του, κερδίζοντας ταυτοχρόνως πέντε δολάρια. Ο πελάτης,

που εμπιστευόταν την εταιρεία, έκανε κλικ στο σύνδεσμο κι εισήγαγε τις πληροφορίες που του ζητούσαν, το όνομα, τη διεύθυνση, τον αριθμό τηλεφώνου και πληροφορίες σχετικά με την πιστωτική του κάρτα, και περίμενε να του εμφανιστούν τα πέντε δολάρια στον επόμενο λογαριασμό της πιστωτικής του κάρτας. Αντίθετα, αυτό που του εμφανίστηκε ήταν μία σειρά από χρεώσεις για πράγματα που δεν είχε ποτέ αγοράσει. Τι είχε συμβεί; Καθώς ο πελάτης επισκέφτηκε το δικτυακό τόπο, συμπλήρωσε μία ψεύτικη οθόνη εισαγωγής ονόματος χρήστη και κωδικού πρόσβασης που είχε δημιουργήσει ο επιτιθέμενος και που είναι ολόιδια με την πραγματική. Η διαφορά είναι ότι η ψεύτικη αυτή οθόνη δεν εκτελεί τη λειτουργία που αναμένεται μα φανερώνει στον επιτιθέμενο το όνομα χρήστη και τον κωδικό πρόσβασης του πελάτη καθώς και τα υπόλοιπα στοιχεία που εκείνος πληκτρολόγησε.

Κατά το Mitnick, υπάρχει ένα κόλπο που επαναλαμβάνεται συχνά: Στέλνονται e-mails σε χρήστες που τους ζητούν να επισκεφτούν ένα δικτυακό τόπο για διάφορους λόγους, παρέχοντας τους τη διεύθυνση ώστε να μεταβούν κατευθείαν στον τόπο αυτόν. Αν προσέξει κανείς καλύτερα, η διεύθυνση δεν είναι η πραγματική αλλά μοιάζει σε αυτή. Για παράδειγμα η διεύθυνση www.PayPai.com ή www.PayPa1.com μοιάζουν με την πραγματική διεύθυνση www.PayPal.com της εταιρείας PayPal. Οι περισσότεροι χρήστες δεν θα το προσέξουν αλλά και αν το κάνουν απλώς θα νομίσουν ότι κάποιο λάθος υπάρχει στη διεύθυνση. Το κόλπο αυτό είναι ιδιαίτερα δημοφιλές για τους κλέφτες πιστωτικών καρτών

3.4 Τεχνικές Κοινωνικής Μηχανικής

Στη συγκεκριμένη υποενότητα αναφέρονται μερικές τεχνικές τις οποίες χρησιμοποιεί ο κοινωνικός μηχανικός ώστε να αποκτήσει τις πληροφορίες που επιθυμεί και τελικά να επιτύχει το στόχο του.

3.4.1 Οικοδόμηση Εμπιστοσύνης

Οι κοινωνικοί μηχανικοί χρησιμοποιούν τεχνικές για να αυξήσουν την εμπιστοσύνη, τη χρησιμότητα, την απόκτηση πληροφοριών, τη γνώση για τις εσωτερικές διαδικασίες, την τεχνολογία και ότι άλλο επιθυμούν. Για την επίτευξη του στόχου τους συνήθως χρησιμοποιούν πολλές μικρές επιθέσεις. Για παράδειγμα, ένας τηλεφωνικός κατάλογος θα οδηγήσει σε ένα τηλεφώνημα. Η πληροφορία που

αποκτήθηκε από το τηλεφώνημα πιθανώς θα οδηγήσει σε ένα άλλο τηλεφώνημα. Ένας κοινωνικός μηχανικός, συγκεντρώνοντας κάθε ένα κομμάτι πληροφορίας, τελικά θα μπορέσει να πραγματοποιήσει τη μεγάλη του επίθεση στο σύστημα. Μία επιτυχημένη προσπάθεια κοινωνικής μηχανικής μπορεί να έχει σοβαρές επιπτώσεις στην εταιρεία-στόχο [32].

Για να πραγματοποιηθούν τα παραπάνω, δηλαδή για να αποκτηθούν οι πληροφορίες, όπως ένας αριθμός τηλεφώνου ή ένας κωδικός, οι κοινωνικοί μηχανικοί πρέπει να κερδίσουν την εμπιστοσύνη των θυμάτων. Αυτό επιτυγχάνεται δημιουργώντας φιλίες (friendships) [31]. Όταν δημιουργηθεί μία φιλία υπάρχει εμπιστοσύνη μεταξύ αυτών των προσώπων. Συνήθως, την εμπιστοσύνη αυτή εκμεταλλεύονται οι κοινωνικοί μηχανικοί. Πολλοί φίλοι μοιράζονται πληροφορίες μεταξύ τους για διάφορα θέματα, στις οποίες περιέχονται στοιχεία συσχετιζόμενα με τη δουλειά τους. Αν ο εισβολέας θέλει να επιτεθεί σε μία εταιρεία Χ τότε το εναρκτήριο σημείο είναι να κερδίσει την εμπιστοσύνη των υπαλλήλων της. Αν ο εισβολέας έχει ήδη φίλους στην εταιρεία θα ξεκινήσει χρησιμοποιώντας τεχνικές κοινωνικές μηχανικής για την απόκτηση των πληροφοριών που επιθυμεί. Αυτός ο τύπος κοινωνικής μηχανικής, που παρατηρείται συχνά, δεν μπορεί να εντοπιστεί από τα firewalls και τα συστήματα ανίχνευσης εισβολέων.

Πάντως, τις περισσότερες φορές, ο κοινωνικός μηχανικός έχει αναπτυγμένες δεξιότητες χειραγώγησης των ανθρώπων οι οποίοι του επιτρέπουν να αποκτά πρόσβαση στις πληροφορίες τους με τρόπους που κανείς δεν θα το πίστευε ότι είναι εφικτοί. Σε αρκετές περιπτώσεις οι κοινωνικοί μηχανικοί προσεγγίζουν άμεσα τα θύματα τους. Απλώς επικοινωνούν μαζί τους και κατά τη συνομιλία τους ο κοινωνικός μηχανικός δεν ρωτά ύποπτες πληροφορίες, οι οποίες μπορεί να προκαλέσουν διστακτικότητα στο θύμα, μα συζητά μαζί τους φιλικά για διάφορα θέματα. Αυτή αποκαλείται μέθοδος της φιλικής πειθούς. Οι κοινωνικοί μηχανικοί χρειάζεται να έχουν υπομονή για να πετύχουν το σκοπό τους και σε περιπτώσεις που το θύμα δεν πρόκειται να δώσει πληροφορίες, απλώς θα αναζητήσουν ένα πιο ευκολόπιστο θύμα για να τις αποκτήσουν. Αποδεικνύεται ότι όσο μεγαλύτερος είναι ένας οργανισμός τόσο πιο εύκολα κερδίζεται η εμπιστοσύνη. Σε ένα μικρότερο περιβάλλον ο στόχος είναι πιθανόν να γνωρίζει αν ο επιτιθέμενος είναι πραγματικά

αυτός που υποστηρίζει ή όχι. Η οικοδόμηση εμπιστοσύνης είναι απαραίτητη και από μόνη της αλλά και με συνδυασμό με άλλες τεχνικές.

3.4.2 Αντίστροφη Κοινωνική μηχανική

Ένας κοινωνικός μηχανικός μπορεί να χρησιμοποιήσει μία τεχνική που καλείται αντίστροφη κοινωνική μηχανική (reverse social engineering). Υπάρχουν τρία μέρη στην αντίστροφη κοινωνική μηχανική [33]:

- σαμποτάζ (sabotage)
- διαφήμιση (advertising)
- βοήθεια (assisting)

Στην αντίστροφη κοινωνική μηχανική ο επιτιθέμενος στήνει ένα σενάριο κατά το οποίο το θύμα αντιμετωπίζει ένα πρόβλημα και επικοινωνεί με τον εισβολέα ζητώντας βοήθεια. Είναι ένας έξυπνος τρόπος οικοδόμησης εμπιστοσύνης γιατί από τη στιγμή που το θύμα βοηθηθεί θα είναι πρόθυμο να βοηθήσει τον επιτιθέμενο. Ο επιτιθέμενος θα επιλέξει το άτομο στόχο σύμφωνα με το αν έχει πληροφορίες για να τον βοηθήσουν και θα σκεφτεί μία κατάσταση που όλα λειτουργούν κανονικά ώστε να χρησιμοποιήσει την αντίστροφη προσέγγιση.

Ο επιτιθέμενος προσποιούμενος τον IT υπάλληλο, για παράδειγμα, μπορεί να τηλεφωνήσει και να προειδοποιήσει το θύμα ότι μία βλάβη θα επηρεάσει τη συνδέσεις του δικτύου. Μετά το ψεύτικο παράθυρο που δηλώνει τη βλάβη στο σύστημα του θύματος, ο επιτιθέμενος θα επιβεβαιώσει το θύμα ότι τελικά δεν υπάρχουν προβλήματα στο σύστημα εξαιτίας της βλάβης. Με αυτή τη διαδικασία το θύμα αποκτά εμπιστοσύνη στον εισβολέα νομίζοντας ότι τον προστάτευσε. Έτσι, ο εισβολέας μελλοντικά θα έχει τη δυνατότητα να επικοινωνήσει μαζί του και να του ζητήσει τις πληροφορίες που τον ενδιαφέρουν. Αυτή είναι μία καλή ευκαιρία για τον επιτιθέμενο να εγκαταστήσει κακόβουλο λογισμικό στους στόχους-μηχανήματα. Ο κοινωνικός μηχανικός προσποιούμενος τον IT υπάλληλο ή τον πωλητή λογισμικού μπορεί να ζητήσει από το θύμα να επισκεφτεί μία σελίδα ή να ανοίξει ένα επισυναπτόμενο αρχείο το οποίο περιέχει ιό ή άλλο κακόβουλο λογισμικό υποστηρίζοντας ότι το λογισμικό είναι απαραίτητο να εγκατασταθεί για την αναβάθμιση ενός προγράμματος.

3.4.3 Τεχνολογία

Παρόλο που μία επιτυχημένη επίθεση κοινωνικής μηχανικής δεν απαιτεί σημαντικές τεχνικές γνώσεις, χρησιμοποιώντας τεχνολογία σε συνδυασμό με άλλες τεχνικές κοινωνικής μηχανικής μπορεί να αποβεί πολύ αποτελεσματικό. Ένας κοινωνικός μηχανικός μπορεί να χρησιμοποιήσει τις παραπάνω τεχνικές για να μάθει σχετικά με την τεχνική υποδομή της επιχείρησης και στη συνέχεια να εξαπολύσει έναν ιό ο οποίος θα προκαλέσει την κατάρρευση του δικτύου. Οι κοινωνικοί μηχανικοί μπορούν να στείλουν έναν ιό ή έναν Δούρειο Ίππο σαν συνημμένο αρχείο ενός e-mail ή να αφήσουν ένα CD με επιβλαβή κώδικα στο χώρο εργασίας του θύματος ή ακόμη να στείλουν δωρεάν λογισμικό για να το εγκαταστήσει το θύμα.

Δούρειος Ίππος είναι ένα πρόγραμμα που περιέχει βλαπτικό κώδικα σχεδιασμένο να προκαλέσει ζημιά στον υπολογιστή ή στα αρχεία του θύματος ή να υποκλέψει πληροφορίες από τον υπολογιστή ή το δίκτυο. Μερικοί Δούρειοι Ίπποι κρύβονται μέσα στο λειτουργικό σύστημα του υπολογιστή και παρακολουθούν κάθε πλήκτρο που πατιέται και κάθε ενέργεια που επιτελείται ή ακόμη δέχονται να εκτελούν εντολές μέσω δικτυακής σύνδεσης και όλα αυτά χωρίς το θύμα να έχει γνώση της παρουσίας του [7]. Στην κατηγορία αυτή ανήκει και το κατασκοπευτικό λογισμικό (spyware), το οποίο είναι πρόγραμμα που χρησιμοποιείται για να παρακολουθεί κρυφά τις δραστηριότητες του υπολογιστή-στόχου. Το λογισμικό καταγράφει τις δραστηριότητες του χρήστη, όπως τους κωδικούς πρόσβασης και τα πλήκτρα που πατιούνται, το ηλεκτρονικό ταχυδρομείο, τις ιστοσελίδες που έχει επισκεφτεί ακόμα και τις απεικονίσεις της οθόνης του ανά πάσα στιγμή. Το αρνητικό είναι ότι το κατασκοπευτικό λογισμικό δεν ανιχνεύεται από το λογισμικό προστασίας από τους ιούς γιατί αυτού του είδους τα προγράμματα δεν θεωρούνται κακόβουλα παρότι σκοπός τους είναι η παρακολούθηση άλλων χρηστών.

Ένα σύνηθες φαινόμενο που παρατηρούν οι χρήστες όταν κάνουν log on στον ISP τους είναι εμφάνιση ενός e-mail το οποίο υποστηρίζει ότι προέρχεται από την υπηρεσία πελατών του ISP ζητώντας το όνομα, κωδικό και αριθμό πιστωτικής κάρτας του χρήστη. Το ερώτημα που προκύπτει είναι αν ένα IDS ή ένα anti-virus πρόγραμμα μπορεί να το ανιχνεύσει. Αν αυτό συμβαίνει συνέχεια τότε η απάντηση είναι θετική. Στην αντίθεση περίπτωση, πιθανώς, δεν ανιχνεύεται. Δυστυχώς μέσω του e-mail η

κοινωνική μηχανική μπορεί εύκολα να λάβει χώρα αφού ξεκινά μόνο με το άνοιγμα του e-mail από το χρήστη. Τα e-mails παρέχουν σημαντικές ευκαιρίες στους εισβολείς για την απόκτηση των πληροφοριών που τους ενδιαφέρουν. Κάθε σύστημα ανίχνευσης εισβολέων και κάθε anti-virus πρόγραμμα απαιτεί υπογραφές για να αιχμαλωτίσει τα κακόβουλα πακέτα ή mail.

Γενικότερα, υπάρχουν πολλοί τρόποι, χρησιμοποιώντας την τεχνολογία, με τους οποίους οι κοινωνικοί μηχανικοί μπορούν να αποκτήσουν πρόσβαση στο σύστημα-στόχο και να υποκλέψουν πληροφορίες.

3.4.4 Μεταστροφή

Μία από τις πιο ισχυρές τεχνικές του κοινωνικού μηχανικού είναι η λεγόμενη μεταστροφή. Ο κοινωνικός μηχανικός δημιουργεί ένα πρόβλημα, για παράδειγμα διακόπτει οικειοθελώς τη σύνδεση ενός δικτύου παρουσιάζοντας το σαν βλάβη, και μετά κατά μαγικό τρόπο, το επιλύει, εξαπατώντας το θύμα και πείθοντας το να αποκαλύψει τα πιο κρυφά μυστικά της εταιρείας ή ενός συστήματος.

3.4.5 Ρακοσυλλογή

Η ρακοσυλλογή είναι ένας όρος που περιγράφει το ψάξιμο των σκουπιδιών του στόχου με σκοπό την ανεύρεση πολύτιμων πληροφοριών. Αποτελεί μία απλή τεχνική η οποία δεν απαιτεί τεχνικές δυνατότητες. Έχει μηδενικό κόστος μα το κέρδος που μπορεί να αποκομίσει κανείς είναι εκπληκτικό. Το μόνο που απαιτείται είναι η διάθεση του κοινωνικού μηχανικού να ψάξει για τις πληροφορίες που ενδιαφέρεται σε πληροφορίες που θεωρούνται άχρηστες για τους χρήστες. Πολλοί άνθρωποι πετούν σημαντικά χαρτιά χωρίς να σκεφτούν τι πληροφορίες είναι γραμμένες πάνω σε αυτά. Ένα καλό παράδειγμα είναι οι αποδείξεις της πιστωτικής κάρτας. Αφού κάποιος επιβεβαιώσει ότι οι συναλλαγές πραγματοποιήθηκαν σωστά στη συνέχεια τις πετάει. Το πρόβλημα ξεκινά όταν οι αποδείξεις πετιούνται χωρίς να σκίζονται με συνέπεια να βρεθούν από κακόβουλο χρήστη και να χρησιμοποιηθούν ενάντια στους νόμιμους χρήστες τους. Το ίδιο συμβαίνει και με τις δικτυακές πληροφορίες. Πολλές εταιρείες και οργανισμοί έχουν πολιτικές για την αντιμετώπιση των σημαντικών εγγράφων μα δεν λαμβάνουν υπόψιν έγγραφα όπως τηλεφωνικούς καταλόγους της εταιρίας, IP διευθύνσεις, πληροφορίες που αφορούν το server. Ότι έγγραφο δεν είναι απαραίτητο θα πρέπει να καταστρέφεται με τέτοιο τρόπο ώστε η πληροφορία που

περιείχε να μην μπορεί να ανακτηθεί από κακόβουλους χρήστες. Η παραπάνω τεχνική δεν μπορεί να ανιχνευτεί από IDS ή firewall μα σίγουρα θεωρείται επίθεση.

3.4.6 Κερδίζοντας Φυσική Πρόσβαση

Πολλά εταιρικά γραφεία κρατούν εύκολα προσβάσιμες πληροφορίες που μπορούν να βοηθήσουν έναν κοινωνικό μηχανικό να διαπράξει την επίθεσή του. Κυρίως στους μεγάλους οργανισμούς, οι υπάλληλοι δεν γνωρίζονται μεταξύ τους και είναι εύκολο για έναν κοινωνικό μηχανικό να μπει στην εταιρεία παριστάνοντας τον υπάλληλο και να δει πληροφορίες (όπως κωδικούς, οικονομικά στοιχεία, τεχνική υποδομή) σε γραφεία υπαλλήλων.

Οι κοινωνικοί μηχανικοί αποκτούν φυσική πρόσβαση και με έναν άλλο τρόπο. Προσποιούμενοι ότι είναι εργάτες καθαρισμού ή συντήρησης μπαίνουν στο χώρο της εταιρείας που τους ενδιαφέρει και ψάχνουν να βρουν τις πληροφορίες που χρειάζονται. Ένα πλεονέκτημα αυτή της μεθόδου είναι ότι ο κοινωνικός μηχανικός μπορεί να εντοπίσει ένα μηχάνημα στο οποίο ο υπάλληλος δεν αποσυνδέθηκε (log out) και να εγκαταστήσει επιβλαβή κώδικα, να κλέψει πληροφορίες ή να βρει ευαίσθητες πληροφορίες που δεν καταστράφηκαν σωστά. Η τεχνική αυτή καλείται office snooping. Ο εισβολέας μπορεί να κάνει αντίγραφο και να κλέψει ένα σημαντικό αρχείο, ενώ το πρωτότυπο να το αφήσει στη θέση του. Οι παραπάνω ενέργειες δεν γίνονται αντιληπτές από τον ιδιοκτήτη του αρχείου. Ακριβώς, το ίδιο μπορεί να συμβεί με τις απροστάτευτες δικτυακές πληροφορίες. Ο κακόβουλος χρήστης έχει τη δυνατότητα να υποκλέψει απροστάτευτες πληροφορίες χωρίς να γίνει αντιληπτός. Αυτές οι επιθέσεις δεν αφήνουν ίχνη και δεν μπορούν αν αντιμετωπιστούν ούτε από τα firewalls από αλλά ούτε και από τα IDS.

3.5 Άμυνα Ενάντια Στην Κοινωνική Μηχανική

Τι όμως μπορεί να γίνει ώστε να προστατευτεί μία εταιρία ή ένας οργανισμός από την κοινωνική μηχανική; Η κοινωνική μηχανική βασίζεται στους υπαλλήλους μιας εταιρείας ή ενός οργανισμού. Συνεπώς, για να αποφευχθεί μία επίθεση θα πρέπει οι υπάλληλοι να είναι εκπαιδευμένοι και εξοικειωμένοι με συνηθισμένες τεχνικές κοινωνικής μηχανικής. Είναι απαραίτητο οι οργανισμοί να έχουν μία αυστηρή πολιτική ασφαλείας, η οποία θα συμπεριλαμβάνει standards και διαδικασίες που θα

βοηθούν στην εξάλειψη της απειλής κοινωνική μηχανική. Μία καλή γραμμή άμυνας μπορεί να περιέχει [33]:

- Πολιτικές κωδικών
- Ταξινόμηση δεδομένων
- Αποδεκτή πολιτική χρήσης
- Προστασία των δεδομένων
- Προσδιορισμός των ευπαθειών
- Ελέγχους
- Διαδικασία τερματισμού
- Άμεση απάντηση
- Φυσική προστασία
- Ενημέρωση για τις μεθόδους ασφάλειας

3.5.1 Πολιτικές Κωδικών

Για έναν κοινωνικό μηχανικό η απόκτηση πρόσβασης στο σύστημα μπορεί να σημαίνει την επιτυχία ή όχι μίας επίθεσης. Στις εταιρείες και στους οργανισμούς θα πρέπει να υπάρχει μία πολιτική δημιουργίας και παράδοσης των κωδικών (passwords). Μία καλή πολιτική κωδικών πρέπει να περιλαμβάνει πληροφορίες για:

- ✧ Να μη μοιράζονται οι κωδικοί αν κάποιος τους ζητήσει
- ✧ Να μη γράφονται οι κωδικοί σε πρόχειρα χαρτιά και σημειώσεις
- ✧ Να μη χρησιμοποιούνται οι χρήστες default κωδικούς
- ✧ Να υπάρχουν μέθοδοι που αυθεντικοποιούν τους χρήστες που θέλουν να αλλάξουν κωδικό
- ✧ Να υπάρχουν μέθοδοι για την παράδοση των κωδικών
- ✧ Να ακολουθούνται κανόνες για τη δημιουργία κωδικών, για παράδειγμα ελάχιστο μήκος κωδικού, χρήση αλφαριθμητικών και αριθμών
- ✧ Περιοδική αλλαγή κωδικών
- ✧ Αποτυχία login, για παράδειγμα ο λογαριασμός κλειδώνει μετά από τρεις αποτυχημένες προσπάθειες
- ✧ Κωδικούς για διαχειριστές συστήματος

Οι υπάλληλοι πρέπει να γνωρίζουν τη σημαντικότητα ενός «δυνατού» κωδικού. Δεν πρέπει να επιτρέπονται λέξεις που υπάρχουν στο λεξικό ή συνδυασμούς αυτών.

Συχνά, χρησιμοποιούνται εργαλεία που ελέγχουν αν οι κωδικοί είναι κατάλληλοι. Πρέπει να χρησιμοποιούνται κωδικοί που είναι δύσκολο να ανακαλυφτούν σε περιπτώσεις που οι χρήστες χρησιμοποιούν ένα κωδικό για να έχουν πρόσβαση σε μία πληθώρα εφαρμογών στο δίκτυο. Σε αυτές τις περιπτώσεις ενώ βοηθιούνται οι χρήστες και δεν απαιτείται να θυμούνται πολλούς κωδικούς, ωστόσο διευκολύνεται η δουλειά των κοινωνικών μηχανικών καθώς έχουν μόνο ένα κωδικό να ανακαλύψουν. Σύμφωνα με το Mitnick, δεν θα πρέπει οποιαδήποτε επιχείρηση να επιτρέπει την ανταλλαγή κωδικών πρόσβασης. Είναι πολύ ευκολότερο να εφαρμόσει κάποιος έναν αυστηρό κανόνα που θα απαγορεύει στο προσωπικό την αποκάλυψη ή την ανταλλαγή απόρρητων κωδικών πρόσβασης. Είναι και ασφαλέστερο. Ωστόσο, κάθε επιχείρηση πρέπει να εκτιμήσει τη δική της ατμόσφαιρα και της δικές της ανάγκες ασφαλείας όταν αποφασίζει ποια πολιτική θα ακολουθήσει.

Άλλωστε, ένας κωδικός ασφαλείας, όταν χρησιμοποιείται σωστά, προσθέτει ένα ακόμα πολύτιμο επίπεδο προστασίας. Ένας κωδικός ασφαλείας που χρησιμοποιείται εσφαλμένα μπορεί να γίνει χειρότερος από την απουσία κωδικού, και αυτό γιατί δίνει την ψευδαίσθηση της ασφάλειας εκεί που δεν υπάρχει. Οι κωδικοί θα πρέπει να διατηρούνται κρυφοί αλλιώς δεν υπάρχει λόγος ύπαρξης τους. Τα άτομα που τους χρησιμοποιούν θα πρέπει να ακολουθούν πιστά την πολιτική ασφαλείας σχετικά με τους κωδικούς και να μην τους δίνουν σε όποιον τους ζητήσει.

Επιπλέον, απαιτείται μία πολιτική κωδικών που να μην επιτρέπει στους υπαλλήλους να γράφουν πρόχειρα κάπου τους κωδικούς που χρησιμοποιούν. Θα πρέπει οι υπάλληλοι να κατανοούν τους κινδύνους που προκύπτουν αφήνοντας εκτεθειμένους τους κωδικούς τους με αυτόν το τρόπο γιατί είναι υποχρεωμένοι να συμβάλλουν ενεργά στην ασφάλεια του οργανισμού.

Το προσωπικό τεχνικής υποστήριξης θα πρέπει να ακολουθεί μία αυστηρή πολιτική αναγνώρισης και παράδοσης των κωδικών. Η επιβεβαίωση αναγνώρισης σημαίνει ότι θα πρέπει να αυθεντικοποιείται ο χρήστης ώστε να επιβεβαιώνεται ότι πραγματικά είναι αυτός που ισχυρίζεται. Όσον αφορά την παράδοση των κωδικών κάποιοι οργανισμοί επιμένουν ότι ο καλύτερος τρόπος για να γίνει αυτό είναι μέσω e-mail χρησιμοποιώντας το εσωτερικό δίκτυο της εταιρείας. Υπάρχει όμως περίπτωση ένας κακόβουλος χρήστης να διαβάσει το e-mail που περιέχει τον κωδικό. Μία άλλη

τεχνική που συνηθίζεται είναι η ύπαρξη ενός υπαλλήλου που καλείται ID. Σε αυτή την περίπτωση, αν αλλάξει ο κωδικός, ο ID θα τηλεφωνήσει στο χρήστη και αφού επαληθεύσει την ταυτότητα του θα τον ενημερώσει για τον καινούριο κωδικό. Αυτή η μέθοδος δεν παρέχει ασφάλεια, καθώς ο οποιοσδήποτε θα μπορούσε να παριστάνει το χρήστη. Γενικότερα, οι πιο πολλές τακτικές που χρησιμοποιούνται έχουν ποσοστό αποτυχίας 100%. Το συμπέρασμα είναι ότι θα πρέπει πάντα να βρίσκεται ο ασφαλέστερος τρόπος παράδοσης των κωδικών. Για παράδειγμα, το e-mail μπορεί να χρησιμεύσει για να ενημερώσει το χρήστη ότι έγινε αίτηση για αλλαγή του κωδικού. Το e-mail αυτό θα στέλνεται αυτόματα στο χρήστη και θα τον ενημερώνει για την ημερομηνία και ώρα της αλλαγής καθώς και την IP διεύθυνση του μηχανήματος που πραγματοποίησε την αλλαγή. Αν δεν στέλνεται το e-mail αυτόματα στο χρήστη τότε δεν θα μπορεί να γίνεται log-in με νέο κωδικό.

Κάτι άλλο που θα πρέπει να ληφθεί υπόψιν όταν δημιουργείται η πολιτική αναγνώρισης και παράδοσης των κωδικών είναι ο τύπος του λογαριασμού που αναβαθμίζεται. Ένας λογαριασμός διαχειριστή συστήματος προφανώς πρέπει να ακολουθεί μία εντελώς διαφορετική διαδικασία αλλαγής και παράδοσης των κωδικών. Για το λόγο αυτό είναι προφανές ότι απαιτούνται διαφορετικές πολιτικές ανάλογα με τον τύπο λογαριασμού, δηλαδή διαφορετική πολιτική για τους διαχειριστές και διαφορετική για τους χρήστες. Ένας κωδικός που χρησιμοποιεί ο διαχειριστής σε ένα firewall πρέπει να έχει περισσότερους χαρακτήρες και να ακολουθεί κανόνες ονοματολογίας πιο σύνθετους από ότι ένας κωδικός που χρησιμοποιεί ο χρήστης για να έχει πρόσβαση στο δίκτυο.

3.5.2 Ταξινόμηση Δεδομένων

Από τη στιγμή που οι κοινωνικοί μηχανικοί χρησιμοποιούν τη γνώση άλλων για την απόκτηση πληροφοριών είναι απαραίτητο να υπάρχει ένα μοντέλο για την ταξινόμηση των δεδομένων που θα πρέπει να ακολουθούν όλοι οι υπάλληλοι.

Μία πολιτική ταξινόμησης δεδομένων βοηθά στην εφαρμογή των σωστών ελέγχων σε σχέση με την αποκάλυψη πληροφοριών. Χωρίς πολιτική ταξινόμησης δεδομένων, όλες οι εσωτερικές πληροφορίες πρέπει να θεωρούνται εμπιστευτικές, εκτός εάν έχουν χαρακτηριστεί ρητά αλλιώς. Για να προστατευτεί ένα σύστημα από την αποκάλυψη φαινομενικά αθώων πληροφοριών, τα άτομα που έχουν το ρόλο και την

ευθύνη να σχεδιάσουν μία πολιτική ταξινόμησης δεδομένων, πρέπει να εξετάσουν τους τύπους των λεπτομερειών που μπορεί να χρησιμοποιηθούν για την απόκτηση πληροφοριών που φαίνονται αθώες αλλά στην πραγματικότητα δεν είναι. Αν και ποτέ δεν δίνει κάποιος το pin της πιστωτικής του κάρτας, πιθανόν θα έλεγε ποιον server χρησιμοποιεί η εταιρεία για την κατάστρωση του λογισμικού. Αυτού του είδους η πληροφορία μπορεί να χρησιμοποιηθεί από κάποιον που προσποιείται ότι έχει νόμιμη πρόσβαση στο εταιρικό δίκτυο [7]. Μία σαφής πολιτική ταξινόμησης δεδομένων είναι θεμελιώδης για την προστασία των πληροφοριών των συστημάτων γιατί καθορίζει κατηγορίες σύμφωνα με τις οποίες αποκαλύπτονται ή όχι οι ευαίσθητες πληροφορίες. Στην ουσία θέτει κατευθυντήριες γραμμές για την ταξινόμηση πολύτιμων πληροφοριών σε διάφορα επίπεδα.

Στη συνέχεια ακολουθεί ένα παράδειγμα μιας ταξινόμησης δεδομένων [7], [32]:

- Απόρρητη: Στην κατηγορία αυτή ανήκουν τα εξαιρετικά ευαίσθητα έγγραφα και δεδομένα, όπως σχέδια και στρατηγικές ενός οργανισμού, δηλαδή πληροφορίες που αν δημοσιοποιηθούν ή χαθούν θα βλάψουν την εταιρεία. Οι πληροφορίες που χαρακτηρίζονται ως απόρρητες θα πρέπει να προστατεύονται συνεχώς και το επίπεδο ασφαλείας είναι το υψηλότερο δυνατό.
- Άκρως εμπιστευτική: Πληροφορίες οι οποίες αν δημοσιοποιηθούν επηρεάζουν τις λειτουργίες ενός οργανισμού. Οι πληροφορίες αυτές περιλαμβάνουν μεταξύ άλλων πληροφορίες λογαριασμών, ευαίσθητα στοιχεία πελατών και επιχειρηματικά σχέδια. Τέτοιου είδους πληροφορίες δεν πρέπει να αντιγραφούν χωρίς συγκεκριμένη εξουσιοδότηση. Το επίπεδο ασφαλείας είναι αρκετά υψηλό.
- Ιδιοκτησιακή: Οι πληροφορίες έχουν ιδιοκτησιακή χρήση στην εταιρεία και αφορούν διαδικασίες, σχέδια project, σχέδια και διαδικασίες που καθορίζουν το τρόπο που ένας οργανισμός λειτουργεί. Στις πληροφορίες τέτοιας φύσεως έχει πρόσβαση μόνο εξουσιοδοτημένο προσωπικό.
- Εσωτερική: Πληροφορίες οι οποίες δεν πρέπει να αποκαλυφτούν μα ακόμη και αν αυτό συμβεί συνήθως δεν προκαλείται οικονομική απώλεια στην εταιρεία και δεν κινδυνεύει να πληγεί η αξιοπιστία της. Τέτοιες πληροφορίες είναι οι ώρες που ορίζονται οι συναντήσεις, εσωτερικές υπενθυμίσεις, αναφορές projects. Το επίπεδο ασφαλείας είναι ελεγχόμενο μα «φυσιολογικό».

- Δημόσια: Οι πληροφορίες μπορούν να διατεθούν δημόσια στο κοινό, για παράδειγμα ετήσιες αναφορές και δηλώσεις τύπου. Το επίπεδο ασφαλείας είναι το χαμηλότερο.

Γενικότερα, τα ευαίσθητα αρχεία μπορούν να προστατεύονται πίσω από κατάλληλους έλεγχους πρόσβασης ούτως ώστε μόνο άτομα που έχουν το σχετικό δικαίωμα να τα ανοίξουν. Κάποια λειτουργικά συστήματα διαθέτουν εγγενείς μεθόδους ελέγχου που μπορεί να ρυθμιστούν έτσι ώστε να καταγράφουν ενέργειες, όπως κάθε πρόσβαση ή απόπειρα πρόσβασης σε προστατευμένους φακέλους και το πρόσωπο που την επιχείρησε.

Χρησιμοποιώντας μία πολιτική ταξινόμησης δεδομένων εμποδίζονται οι κοινωνικοί μηχανικοί από την απόκτηση πληροφοριών με εύκολο τρόπο. Επιπλέον, οι υπάλληλοι είναι ενημερωμένοι και γνωρίζουν ποιες πληροφορίες μπορούν να αποκαλύψουν στον καθένα χωριστά και ποιες όχι.

3.5.3 Αποδεκτή Πολιτική Χρήσης

Μία αποδεκτή πολιτική χρήσης συμβάλει στο ότι τα ευαίσθητα δεδομένα δεν θα δοθούν σε τρίτους και προστατεύει το σύστημα από πιθανή παραβίαση. Μία πολιτική χρήσης θα πρέπει να έχει πληροφορίες σχετικά με τα ακόλουθα [32]:

- Οι πληροφορίες που αφορούν τα συστήματα και τους πόρους δικτύου να δίνονται μόνο σε εξουσιοδοτημένα άτομα
- Να απαγορεύεται η παροχή απόρρητων και εμπιστευτικών πληροφοριών σε τρίτους
- Η χρήση e-mail επιτρέπεται μονό για νόμιμους σκοπούς
- Επιθέσεις άρνησης εξυπηρέτησης (DoS)
- Παραποίηση / ψεύτικη καταχώρηση στοιχείων
- Προσπάθειες για απόκτηση πρόσβασης σε μη εξουσιοδοτημένους πόρους
- Μη εξουσιοδοτημένο / παράνομο λογισμικό
- Εμπορική χρήση των πληροφοριακών πόρων
- Κατάχρηση της σύνδεσης του Internet
- Χρήση του δικτύου που αντιτίθεται στους νόμους του κράτους

3.5.4 Προστασία Των Δεδομένων

Είναι επικίνδυνο να στέλνει κάποιος ένα αρχείο σε έναν άλλον τον οποίο δεν γνωρίζει, ακόμα κι να φαίνεται ότι είναι συνάδελφος της ίδιας εταιρείας και το αρχείο αποστέλλεται εσωτερικά, σε κάποια διεύθυνση ή σε κάποιο φαξ της εταιρείας [7]. Η τακτική ασφαλείας της εταιρείας πρέπει να είναι ξεκάθαρη σχετικά με τις προφυλάξεις που πρέπει να παίρνει κανείς όταν πρόκειται να παραδώσει πολύτιμα δεδομένα σε κάποιον που δεν το γνωρίζει προσωπικά. Θα πρέπει να ακολουθούνται ορισμένα βήματα επαλήθευσης, με διάφορα επίπεδα πιστοποίησης ανάλογα με το επίπεδο διαβάθμισης της πληροφορίας. Μερικές τεχνικές είναι:

- Βεβαίωση για το δικαίωμα απόκτησης της πληροφορίας (μπορεί να απαιτεί την άδεια του ιδιοκτήτη)
- Καταγραφή των συναλλαγών σε ένα προσωπικό αρχείο
- Κατάρτιση ενός κατάλογου προσώπων που έχουν εκπαιδευτεί ειδικά στις διαδικασίες που απαιτούνται πριν επιτραπεί η αποστολή ευαίσθητων πληροφοριών. Μονό αυτά τα άτομα μπορούν να στέλνουν πληροφορίες σε οποιοδήποτε πρόσωπο δεν είναι γνωστό
- Εάν η αίτηση για πληροφορίες έχει γίνει με e-mail ή φαξ, πρέπει να παρθούν επιπρόσθετα μέτρα ασφαλείας για να επαληθευτεί ότι το αίτημα προέρχεται πράγματι από το πρόσωπο από το οποίο φαίνεται να προέρχεται

3.5.5 Προσδιορισμός Των Ευπαθειών

Οι οργανισμοί πρέπει περιοδικά να κάνουν ελέγχους για να ανακαλύψουν τυχόν ευπάθειες στο σύστημα τους ώστε να προστατευτούν από τυχόν επιθέσεις, συμπεριλαμβανομένου της κοινωνικής μηχανικής, οι οποίες θα εκμεταλλεύονταν τις ευπάθειες του συστήματος για να το βλάψουν. Στους ελέγχους χρησιμοποιούνται εργαλεία και συνηθισμένες τεχνικές hacking που σε άλλες περιπτώσεις θα χρησιμοποιούνταν για την παραβίαση του δικτύου. Επιπλέον, πρέπει να χρησιμοποιούνται μέθοδοι κοινωνικής μηχανικής, ώστε οι υπάλληλοι να είναι γνώστες τέτοιων καταστάσεων. Με τους παραπάνω τρόπους επιτυγχάνεται μεγαλύτερη προστασία σε τυχόν επίθεση που δέχεται το σύστημα.

3.5.6 Έλεγχοι Παρελθόντος

Όπως αναφέρθηκε σε πολλές περιπτώσεις οι κοινωνικοί μηχανικοί χρησιμοποιούν τους υπαλλήλους μιας εταιρείας για να επιτύχουν τους στόχους τους. Άλλες φορές γίνονται οι ίδιοι υπάλληλοι στην εταιρεία και έτσι μαθαίνουν ακριβώς τι πραγματικά επικρατεί στο εσωτερικό της και οργανώνουν την επίθεση τους με περισσότερες πιθανότητες επιτυχίας. Για το λόγο αυτό ένας έλεγχος του ιστορικού είναι απαραίτητος για όλους τους νεοπροσλαμβανομένους, συμβαλλόμενους, συμβούλους, προσωρινούς εργαζομένους ή εκπαιδευμένους πριν τους γίνει πρόταση εργασίας ή συνεργασίας. Αν μετά το πέρας των ελέγχων διαπιστωθεί, για παράδειγμα, ότι ένα άτομο έχει βεβαρημένο παρελθόν στις επιθέσεις κοινωνικής μηχανικής, θα πρέπει να απορριφθεί η αίτηση του προτού αποτελέσει απειλή για την εταιρεία. Τυπικοί έλεγχοι που πρέπει να κάνουν οι εταιρίες τόσο στους υπάλληλους τους όσο και σε αυτούς που πρόκειται να προσλάβουν είναι έλεγχοι για:

- ✧ προηγούμενο εγκληματικό παρελθόν των υπαλλήλων
- ✧ παλιότερα ηλεκτρονικά εγκλήματα (πιθανές επιθέσεις σε συστήματα υπολογιστών και ειδικότερα με χρήσης κοινωνικής μηχανικής)
- ✧ την επαλήθευση των γνώσεων και των σπουδών των ατόμων
- ✧ τις συστάσεις που έχουν από προηγούμενους εργοδότες.

Οι τύποι των ελέγχων ποικίλουν από οργανισμό σε οργανισμό. Η ύπαρξη αυστηρών ελέγχων που σχετίζονται με το παρελθόν των υπαλλήλων συμβάλλει στην ασφάλεια της εταιρείας καθώς ένας κακόβουλος χρήστης δεν θα γίνει μέλος της. Αυτό μειώνει σημαντικά την πιθανότητα να συμβούν επιθέσεις κοινωνικής μηχανικής στην εταιρεία αυτή.

3.5.7 Διαδικασία Τερματισμού

Η διαδικασία τερματισμού αναφέρεται στη διαδικασία που θα πρέπει να ακολουθεί η εταιρεία όταν απολύει κάποιον υπάλληλο της ή παραιτείται με τη θέληση του. Όταν συμβούν τα παραπάνω τότε θα πρέπει να γίνουν κάποιες ενέργειες ώστε ο πρώην υπάλληλος να μην μπορεί να βλάψει την εταιρεία χρησιμοποιώντας την πρόσβαση στα συστήματα της. Οι βασικότερες είναι:

- ✧ Να αφαιρεθεί το όνομα του προσώπου από τον κατάλογο του προσωπικού
- ✧ Να ειδοποιηθεί το προσωπικό σε όλες τις εισόδους της εταιρείας

- ✧ Να γίνει διακοπή των λογαριασμών του πρώην υπαλλήλου, περιλαμβανομένου και των λογαριασμών για πρόσβαση στο δίκτυο της εταιρείας, σε βάσεις δεδομένων αυτής ή στο διαδίκτυο από χώρους εκτός εταιρείας.

Η διαδικασία τερματισμού πρέπει να υφίσταται ώστε να αφαιρεθεί από τον παλαιότερο υπάλληλο η δυνατότητα να βλάψει την εταιρεία έχοντας πρόσβαση στις πληροφορίες που την αφορούν. Όταν ένας υπάλληλος φύγει από την εταιρεία αυτομάτως πρέπει να του αφαιρείται η πρόσβαση στο δίκτυο, η απομακρυσμένη πρόσβαση, η πρόσβαση σε εφαρμογές της εταιρίας και όλες οι σχετικές διευκολύνσεις που παρέχονται στο προσωπικό. Ακόμη και όταν ένας υπάλληλος λείπει για ένα χρονικό διάστημα ο λογαριασμός του πρέπει να κλειδώνει ώστε να μεγιστοποιείται η ασφάλεια της εταιρείας.

Θα πρέπει να δοθεί ιδιάζουσα σημασία στην περίπτωση του υπαλλήλου που είχε δικαιώματα διαχείρισης του συστήματος. Οι κωδικοί διαχείρισης που χρησιμοποιούσε οφείλουν να αλλαχθούν αμέσως από τους υπόλοιπους διαχειριστές ώστε να μη διακυβεύεται η πρόσβαση σε ευαίσθητες πληροφορίες. Έχοντας οι εταιρείες μία διαδικασία τερματισμού είναι μία αποτελεσματική άμυνα για να προστατευτούν από προηγούμενους πιθανούς δυσαρεστημένους υπαλλήλους οι οποίοι θα προσπαθήσουν να χρησιμοποιήσουν την εσωτερική γνώση που έχουν εναντίον της εταιρείας.

3.5.8 Άμεση Απάντηση

Σε περίπτωση επίθεσης θα πρέπει να υπάρχει άμεση αντίδραση ώστε να συλλεχθεί όσο το δυνατόν περισσότερη πληροφορία για την επίθεση. Μία επίθεση που δεν θα γίνει αντιληπτή μπορεί να είναι ο προάγγελος μίας σειράς επιθέσεων. Η αναγνώριση και η αντιμετώπιση μιας επίθεσης είναι διαδικασίες που βοηθούν και σε μελλοντικές επιθέσεις. Δηλαδή, αφενός αντιμετωπίζεται άμεσα η επίθεση και προστατεύεται το σύστημα, αφετέρου με την αναγνώριση της το σύστημα προστατεύεται μελλοντικά σε περίπτωση που δεχτεί την ίδια επίθεση. Κάθε οργανισμός πρέπει να σχεδιάζει τους δικούς του μηχανισμούς άμεσης απάντησης ανάλογα με τις ανάγκες του. Οι μηχανισμού αυτοί πρέπει να σχετίζονται με τη συγκέντρωση πληροφοριών και την ανάλυση αυτών. Σε περιπτώσεις κοινωνικής μηχανικής οι υπάλληλοι πρέπει να επικοινωνούν με το διαχειριστή και να τον ενημερώνουν για τυχόν ύποπτα γεγονότα.

Με τον τρόπο αυτό ο διαχειριστής μπορεί να αντιμετωπίσει τον πιθανό στόχο του επιτιθεμένου και να προστατεύσει την εταιρία από την επίθεση.

3.5.9 Φυσική Προστασία

Η φυσική προστασία αναφέρεται στα μέτρα που πρέπει να λαμβάνει μία εταιρεία ώστε να μην εισχωρήσει στο χώρο της ένας κοινωνικός μηχανικός παριστάνοντας τον υπάλληλο. Αν και συνήθως οι κοινωνικοί μηχανικοί αποφεύγουν να εμφανίζονται αυτοπροσώπως σε κάποιο χώρο εργασίας που έχουν ως στόχο, υπάρχουν κάποιες περιπτώσεις που το κάνουν.

Απαραίτητο είναι να γίνεται, σε καθημερινή βάση, επαλήθευση όλων των υπάλληλων που εργάζονται στην εταιρεία. Επιπλέον, πρέπει να γίνεται αναγνώριση των επισκεπτών εργαζομένων, όπως οδηγών, οι οποίοι θα πρέπει να έχουν μαζί τους μία ισχύουσα άδεια οδήγησης ή κάποιο άλλο στοιχείο ταυτότητας με φωτογραφία για να γίνουν δεκτοί στο χώρο εργασίας. Επίσης, η εταιρεία πρέπει να διατηρεί τα στοιχεία τους στο ημερολόγιο επισκεπτών για κάποιο χρονικό διάστημα σε περίπτωση που προκύψει κάποιο ατύχημα μετά την αναχώρηση τους. Οι επισκέπτες θα πρέπει να συνοδεύονται συνεχώς από κάποιο υπάλληλο, ώστε μην έχει τη δυνατότητα να περιφέρεται στο κτίριο αποκτώντας πιθανότατα πρόσβαση σε ευαίσθητες πληροφορίες.

Γενικότερα, η φυσική προστασία, ελέγχοντας και επαληθεύοντας των στοιχεία όσων εισέρχονται στην εταιρεία, έχει ως στόχο να εξαλείψει την πιθανότητα εισβολής ενός κοινωνικού μηχανικού σε αυτήν.

3.5.10 Ενημέρωση για τις μεθόδους ασφάλειας

Είναι αποδεκτό ότι η μεγαλύτερη προστασία ενάντια μίας επίθεσης είναι η γνώση ότι υπάρχει. Για να προστατευτεί ένας οργανισμός από επιθέσεις κοινωνικής μηχανικής θα πρέπει να ενημερώσει τους υπαλλήλους του για την ύπαρξη τέτοιων επιθέσεων και να τους εκπαιδεύσει ώστε να μπορούν να τις αντιλαμβάνονται και να τις αντιμετωπίζουν με τον πιο κατάλληλο τρόπο. Μόλις οι εργαζόμενοι καταλάβουν καλύτερα τους τρόπους με τους οποίους είναι δυνατόν κάποιος να τους χειραγωγήσει, θα μπορούν ευκολότερα να αναγνωρίσουν τα σημάδια μιας τυχόν επίθεσης [7].

Περιοδικά όλοι οι υπάλληλοι θα πρέπει να ενημερώνονται για τους κανόνες, τις μεθόδους και τις διαδικασίες ασφαλείας που πρέπει να ακολουθούν. Οι υπάλληλοι που προσλαμβάνονται απαιτείται να διαβάζουν τις πολιτικές ασφαλείας καθώς και να υπογράφουν ένα έγγραφο ότι τις διάβασαν, τις κατάλαβαν και θα τις εφαρμόσουν [23]. Ένα πρόγραμμα εκπαίδευσης σχετικά με την ασφάλεια θα πρέπει να περιέχει όλες τις πληροφορίες για τις πολιτικές ασφαλείας καθώς και για την εκπαίδευση σε τεχνικές κοινωνικής μηχανικής. Οι υπάλληλοι θα πρέπει να γνωρίζουν σε ποιον θα απευθυνθούν μόλις εντοπίσουν μία επίθεση τύπου κοινωνικής μηχανικής.

Σύμφωνα με το Mitnick [7], ένα πρακτικό πρόγραμμα εκπαίδευσης ως προς την ασφάλεια, το οποίο λαμβάνει υπόψιν τα διάφορα πρόσωπα της κοινωνικής μηχανικής, θα πρέπει να περιλαμβάνει τα ακόλουθα:

- ✧ Περιγραφή των τρόπων που μπορεί να χρησιμοποιήσει ένας επιτιθέμενος με ικανότητες κοινωνικής μηχανικής για να εξαπατήσει κάποιον
- ✧ Τις μεθόδους που χρησιμοποιούν για να εξαπατήσουν τα θύματα τους
- ✧ Πώς θα αναγνωριστεί μία πιθανή επίθεση κοινωνικού μηχανικού
- ✧ Τον τρόπο χειρισμού ενός ύποπτου ζητήματος
- ✧ Πού να αναφερθούν οι απόπειρες ή επιτυχημένες επιθέσεις κοινωνικής μηχανικής
- ✧ Τη σημασία που έχει η επαλήθευση της ταυτότητας και του βαθμού εξουσιοδότησης οποιουδήποτε ζητάει πληροφορίες ή ζητάει από τον εργαζόμενο να προβεί σε κάποιες ενέργειες
- ✧ Τις διαδικασίες προστασίας των ευαίσθητων πληροφοριών
- ✧ Τις πολιτικές ασφαλείας της εταιρείας και τη σημασία τους για την προστασία των πληροφοριών και των πληροφοριακών συστημάτων της
- ✧ Την υποχρέωση κάθε εργαζομένου να συμμορφώνεται με τους κανόνες και τις συνέπειες της μη τήρησής τους

Το βασικότερο είναι όμως να κατανοήσουν οι υπάλληλοι την ευθύνη που έχουν ώστε να διατηρήσουν τα δεδομένα της επιχείρησης και την υποδομή της ασφαλή. Η εκπαίδευση πάνω σε θέματα ασφαλείας μπορεί να σημαίνει την επιτυχία ή όχι μιας επίθεσης κοινωνικής μηχανικής.

3.6 Κοινωνική Μηχανική Και IDS

Η συγκεκριμένη υποενότητα αναφέρεται αρχικά στις ευπάθειες που παρουσιάζουν τα συστήματα ανίχνευσης εισβολέων, τις οποίες εκμεταλλεύονται οι κοινωνικοί μηχανικοί για την πραγματοποίηση των επιθέσεων τους. Στη συνέχεια αναφέρονται τρόποι με τους οποίους οι κακόβουλοι χρήστες μπορούν να παρακάμψουν ένα IDS. Επίσης, αναλύονται γεγονότα του δικτύου τα οποία υποδηλώνουν πιθανές επιθέσεις κοινωνικής μηχανικής. Τα γεγονότα αυτά πρέπει να ανιχνεύονται από τα συστήματα ανίχνευσης εισβολέων και στη συνέχεια να εκτελούνται ενέργειες που έχουν ως σκοπό την προστασία του συστήματος.

3.6.1 Ευπάθειες Των IDS Που εκμεταλλεύονται Οι Κοινωνικοί Μηχανικοί

Οι παραδοσιακοί μηχανισμοί ασφάλειας, IDS, firewall, λογισμικό προστασίας από τους ιούς, θεωρούνται τα πιο δημοφιλή προϊόντα προστασίας. Δυστυχώς όμως, αποδεικνύονται όλο και λιγότερο αποτελεσματικά απέναντι στη νέα γενιά επιθέσεων περιλαμβανομένου των επιθέσεων κοινωνικής μηχανικής. Οι επιτιθέμενοι ανακαλύπτουν αδυναμίες και τρωτά σημεία στους παραπάνω μηχανισμούς και υλοποιούν μεθόδους με τις οποίες τους παρακάμπτουν.

Στη συνέχεια αναφέρονται μερικά παραδείγματα αδυναμιών των IDS [38]:

- Τα IDS τοποθετούνται σε «στρατηγικά» σημεία του δικτύου, όπως πριν ή/και μετά το firewall, και δεν μπορούν παρακολουθούν ολόκληρο το δίκτυο.
- Τα IDS πραγματοποιούν έλεγχο ανά πακέτο (δεν «κάθονται» "in-line" στο δίκτυο) και δεν μπορούν ενεργά να μπλοκάρουν επιθέσεις σε πραγματικό χρόνο.
- Τα IDS παρακολουθούν την κίνηση και σημάνουν συναγερμούς όταν ανακαλύψουν κακόβουλο λογισμικό. Αυτό επιτρέπει σε γρήγορα διαδιδόμενες επιθέσεις να επιτύχουν.
- Τα IDS μπορούν να ξεγελαστούν χρησιμοποιώντας ακατάλληλες ακολουθίες πακέτων.
- Ανακατευθύνοντας την κίνηση από πολλά Gigabit ports τα IDS υπερχειλίζουν (overrun) και χάνουν τα πακέτα και τις επιθέσεις.
- Τα IDS μπορούν να δημιουργήσουν ψευδή θετικά σήματα. Για να αποφευχθεί αυτό απαιτείται συνεχής παρακολούθηση και ρυθμίσεις. Υπάρχει, συνεπώς, υψηλός φόρτος συντήρησης. Κάποιοι διαχειριστές δεν διαθέτουν τον απαραίτητο

χρόνο για τον συνεχή έλεγχο των IDS και συνεπώς ύποπτη κίνηση περνά απαρατήρητη.

- IDS συστήματα τα οποία συνδυάζονται με firewalls αντιδρούν αργά σε γρήγορα διαδιδόμενα προγράμματα δικτυακών σκουληκιών.

3.6.2 Τρόποι Παράκαμψης Των IDS

Σύμφωνα με τον Cohen [35] μερικοί τρόποι παράκαμψης των IDS είναι οι ακόλουθοι:

- Εισάγοντας εξωτερικούς χαρακτήρες σε μία αναγνωρισμένη επίθεση προκαλείται αποτυχία στην ανίχνευση της επίθεσης.
- Αλλάζοντας το διαχωριστικό χαρακτήρα στο σύστημα ώστε για παράδειγμα το % να είναι ο διαχωριστής. Αυτό μπερδεύει σχεδόν (χωρίς εξαιρέσεις) όλα τα συστήματα ανίχνευσης εισβολέων.
- Αναδιάταξη μιας εντοπισμένης ακολουθίας. Αν η επίθεση έχει την ακολουθία "a;b;c" θα μπορεί επίσης να δουλεύει ως "b;a;c" και δεν θα γίνει αντιληπτή από τα συστήματα ανίχνευσης.
- Χρησιμοποιώντας διαφορετικές εντολές για την εκτέλεση ίδιων λειτουργιών. Για παράδειγμα, η εντολή "echo *" είναι σχεδόν ίδια με την "ls" στο Unix.
- Αλλάζοντας τα ονόματα αναγνωρισμένων επιθέσεων. Για παράδειγμα, αν η αναγνωρισμένη επίθεση χρησιμοποιεί προσωρινά αρχεία με όνομα "xxx" μπορεί να αλλάξει το όνομα των αρχείων σε "yyy" ώστε να μη γίνει αντιληπτή.
- Εξουδετερώνοντας τα ports του αισθητήρα των IDS. Για παράδειγμα, χρησιμοποιώντας έναν echo ιό ενάντια σε ένα UDP port, συνήθως τα ports του αισθητήρα καθίστανται ανίκανα να δεχτούν επιπλέον εισόδους από άλλους αισθητήρες.
- Προκαλώντας κατάρρευση του IDS με ping πακέτα. Στέλνοντας μεγάλα πακέτα IPNG, πολλά συστήματα που τρέχουν IDS μπορούν να καταρρεύσουν. Σαν συνέπεια αυτά τα IDS δεν μπορούν πλέον να ανιχνεύουν ακολουθίες επιθέσεων.
- Ένα IDS μπορεί να εξουδετερωθεί πραγματοποιώντας επίθεση στην πλατφόρμα του. Τα περισσότερα συστήματα ανίχνευσης εισβολέων τρέχουν σε κανονικούς υπολογιστές οι οποίοι μπορούν να δεχτούν επιθέσεις. Όταν η πλατφόρμα καταρρεύσει τότε το IDS καθίσταται ανίκανο.

- Τυπώνοντας τα πάντα προς τα πίσω και χρησιμοποιώντας ένα πρόγραμμα μεταφραστή για να τα αντιστρέψει. Η ίδια διαδικασία ακολουθείται για τις συναλλαγές που επιστρέφονται στο χρήστη.
- Ξεκινώντας μία εξερχόμενη διαδικασία από το θύμα διαμέσου του modem και πραγματοποιώντας επίθεση στη σύνδεση αυτή. Αν το IDS είναι δικτυακό (NIDS) θα χάσει αυτά τα πακέτα.
- Ξεκινώντας μία διαδικασία από ένα ασυνήθιστο IP port. Τα ports αυτά συχνά δεν παρακολουθούνται από τα συστήματα ανίχνευσης εισβολέων.
- Χρησιμοποιώντας για την επίθεση ένα διαφορετικό tunneled πρωτόκολλο διαδικασίας (όπως IP αντί HTML).
- Πραγματοποιώντας επίθεση dial-ins αντί δικτυακή. Τα NIDS συστήματα δεν θα εντοπίσουν αυτή τη δραστηριότητα.
- Κωδικοποιώντας την επίθεση σε διαφορετική γλώσσα από αυτή που είχε αρχικά παρουσιαστεί.
- Δημιουργώντας μεγάλο αριθμό από ψευδή θετικά σήματα ώστε να αυξήσει το επίπεδο θορύβου.
- Τοποθετώντας την επίθεση σε ένα εκτελέσιμο πρόγραμμα, όπως Δούρειο Ίππο, και πείθοντας το θύμα να το κατεβάσει και να το τρέξει.
- Χρησιμοποιώντας για την πραγματοποίηση της επίθεσης ένα σπανίως χρησιμοποιούμενο πρωτόκολλο. Πιθανόν τα IDS να μην γνωρίζουν πώς να μεταφράσουν τα αυτά πακέτα.
- Χρησιμοποιώντας μη τεχνικές επιθέσεις, όπως κοινωνική μηχανική. Ενώ τα IDS ελέγχουν τα bits και bytes των δεδομένων, δεν ανιχνεύουν πολλές τέτοιες επιθέσεις που χρησιμοποιούνται στη σημερινή εποχή από τους hackers.

Για να ξεπεραστούν οι αδυναμίες των IDS μερικοί διαχειριστές μετατρέπουν τα IDS σε Intrusion Prevention Systems (IPS) [38]. Τα IPS συστήματα κάθονται in-line με τη δικτυακή κίνηση και έχουν την δυνατότητα να σταματούν ύποπτα και επικίνδυνα πακέτα και επιθέσεις που σχετίζονται με ανωμαλίες σε πρωτόκολλα, όπως SYN πλημμύρες και ICMP επιθέσεις.

3.6.3 IDS – Ανίχνευση Γεγονότων Που Υποδηλώνουν Επιθέσεις Κ.Μ.

Όπως αναφέρθηκε και παραπάνω, η κοινωνική μηχανική αποτελεί πλέον μία σημαντική απειλή για τα πληροφοριακά συστήματα και τα δίκτυα. Τα συστήματα ανίχνευσης εισβολέων (όπως και οι υπόλοιποι μηχανισμοί ασφαλείας που χρησιμοποιούνται για την προστασία του δικτύου) αποδεικνύονται ανίκανα να εντοπίσουν τέτοιου είδους επιθέσεις. Για να επιτευχθεί αυτό, θα πρέπει να ρυθμιστούν κατάλληλα ώστε να ανιχνεύουν γεγονότα που πραγματοποιούνται στο σύστημα και τα οποία υποδηλώνουν επιθέσεις «ηλεκτρονικής» κοινωνικής μηχανικής. Με τον όρο «ηλεκτρονική» κοινωνική μηχανική αναφερόμαστε στην κοινωνική μηχανική που χρησιμοποιεί την τεχνολογία (για παράδειγμα e-mails, Δούρειους Ίππους, κατασκοπευτικό λογισμικό) για να επιτύχει τους στόχους της.

Συνεπώς, ένα σύστημα ανίχνευσης εισβολέων για να αντιμετωπίσει επιθέσεις ηλεκτρονικής κοινωνικής μηχανικής θα πρέπει να προγραμματιστεί έτσι ώστε να ανιχνεύει εκείνα τα γεγονότα που υποδηλώνουν τέτοιες επιθέσεις. Αφού τα ανιχνεύσει μπορεί είτε να ενημερώσει το διαχειριστή του συστήματος είτε να εκτελέσει το ίδιο ενέργειες για να προστατεύσει το σύστημα για το οποίο είναι υπεύθυνο.

Στη συνέχεια αναλύονται γεγονότα και καταστάσεις του συστήματος που πρέπει να ανιχνεύονται από τα IDS και υποδηλώνουν επιθέσεις ηλεκτρονικής κοινωνικής μηχανικής.

- Τα IDS πρέπει να εντοπίζουν προσπάθειες αποστολής μεγάλου όγκου εξερχόμενων πληροφοριών από τους δικτυακούς υπολογιστές σε εξωτερικούς. Το γεγονός αυτό θεωρείται ύποπτο και μπορεί να σημαίνει ότι ένας κακόβουλος χρήστης, παραβιάζοντας το σύστημα, κατεβάζει δεδομένα και πληροφορίες του συστήματος που αφορούν την υποδομή του, τα αρχεία ή τους κωδικούς των χρηστών, χωρίς να έχει γίνει αντιληπτός από τους μηχανισμούς ασφαλείας.
- Τα IDS πρέπει να βοηθούν στην ανίχνευση προγραμμάτων κατασκοπευτικού λογισμικού (spyware programs). Το κατασκοπευτικό λογισμικό θεωρείται σημαντικό πρόβλημα τόσο για τους διαχειριστές όσο και για τους χρήστες. Αφενός ενοχλεί τους χρήστες με popping up διαφημίσεις αφετέρου μπορεί να υποκλέψει πληροφορίες για τη συμπεριφορά χρηστών ή ακόμη και να στείλει

δεδομένα σε εξωτερικούς servers. Επιπλέον, τα κατασκοπευτικά λογισμικά μπορούν να μειώσουν την απόδοση του συστήματος και απαιτείται αρκετός χρόνος για την κατάργησή τους από αυτό. Συνεπώς, τα IDS πρέπει να εντοπίζουν έγκυρα αυτά τα προγράμματα προτού βλάψουν με τους παραπάνω τρόπους τα συστήματα.

- Τα IDS πρέπει να ανιχνεύουν προγράμματα backdoor. Όταν ένας hacker επιτίθεται σε ένα δίκτυο που δεν έχει άμεση πρόσβαση στο internet, τότε στέλνει στο χρήστη, με e-mail, ένα Δούρειο Ίππο ώστε να εκμεταλλευτεί τον browser του χρήστη. Όταν το πρόγραμμα εγκατασταθεί, το επόμενο βήμα είναι να δημιουργήσει ένα τρόπο επικοινωνίας. Αυτό επιτυγχάνεται εγκαθιστώντας ένα backdoor πρόγραμμα. Το πρόβλημα που προκύπτει χρησιμοποιώντας τέτοια προγράμματα σε δίκτυα που προστατεύονται από firewall είναι ότι παρακολουθούν μία εισερχόμενη σύνδεση σε ένα συγκεκριμένο port, ενώ όλη η εισερχόμενη κίνηση μπλοκάρεται [36].
- Τα IDS πρέπει να εξετάζουν τα εξερχόμενα πακέτα που αποστέλλονται από τους δικτυακούς υπολογιστές σε εξωτερικούς servers και να ελέγχουν αν σε αυτά περιέχονται κωδικοί χρηστών. Σε περίπτωση που μεταφέρονται κωδικοί χρηστών οι οποίοι δεν είναι κρυπτογραφημένοι, θα πρέπει να διακόπτεται η μετάδοση των πακέτων αυτών διότι είναι πιθανόν οι κωδικοί να υποκλέπτονται από κακόβουλους χρήστες.
- Τα IDS πρέπει να εντοπίζουν Δούρειους Ίππους προτού εγκατασταθούν στο σύστημα και εκτελέσουν τη λειτουργία που τους έχει προγραμματίσει ο επιτιθέμενος. Ένας χρήστης μπορεί να κατεβάσει Δούρειο Ίππο, χωρίς να το αντιληφθεί, με τους δύο παρακάτω τρόπους:
  - Με την επίσκεψη του σε ένα δικτυακό τόπο που περιέχει λογισμικό αναβάθμισης για κάποιο πρόγραμμα που χρησιμοποιεί. Καθώς κατεβάζει το λογισμικό αυτό κατεβαίνει και ο Δούρειος Ίππος στον υπολογιστή του.
  - Με το άνοιγμα ενός e-mail που περιέχει Δούρειο Ίππο.

  Τα IDS αφού ανιχνεύσουν τους Δούρειους Ίππους μπορούν να διακόψουν τη μεταφορά αυτή ή να αποτρέψουν την εγκατάστασή τους ώστε να προστατευτεί το σύστημα.

3.7 IDS - Αντιμετώπιση Επιθέσεων Κοινωνικής Μηχανικής

Τα συστήματα ανίχνευσης εισβολέων ανιχνεύουν επιθέσεις που πραγματοποιούνται στο επίπεδο TCP. Οι επιθέσεις κοινωνικής μηχανικής λαμβάνουν χώρα στο επίπεδο εφαρμογής και συνεπώς είναι δύσκολο να εντοπιστούν από τα IDS. Συνεπώς, τα IDS, για να μπορούν να αντιμετωπίσουν τέτοιου είδους επιθέσεις, πρέπει να ρυθμιστούν κατάλληλα ώστε να καλύψουν τις ευπάθειες που παρουσιάζουν. Στη συνέχεια θα αναφερθούν συστήματα ανίχνευσης εισβολέων τα οποία σχεδιάστηκαν για να αντιμετωπίζουν επιθέσεις κοινωνικής μηχανικής.

3.7.1 Web Tap

Το Web Tap σχεδιάστηκε (από τους Borders και Prakash του Πανεπιστημίου του Michigan) για να ανιχνεύει προσπάθειες αποστολής σημαντικού όγκου εξερχόμενων πληροφοριών διαμέσου HTTP καναλιών (tunnels) ξεγελώντας τους Web servers του δικτύου το οποίο προστατεύεται από firewall [36]. Ένας επιπλέον στόχος του Web Tap είναι να βοηθάει στην ανίχνευση κατασκοπευτικών προγραμμάτων (spyware) τα οποία στέλνουν σε servers τα προσωπικά δεδομένα των χρηστών χρησιμοποιώντας HTTP συναλλαγές και αφήνουν ανοιχτές τρύπες ασφαλείας στο δίκτυο. Υπάρχουν φίλτρα που βοηθούν στην ανίχνευση ανωμαλιών της εξερχόμενης HTTP κίνησης χρησιμοποιώντας στοιχεία του δικτύου όπως τη χρήση του bandwidth, τον αριθμό αιτήσεων, το διάστημα καθυστέρησης μεταξύ αιτήσεων και το μέγεθος συναλλαγής.

Το Web Tap είναι ένα δικτυακό σύστημα που βασίζεται στην ανίχνευση διαταραχών και εκμεταλλεύεται τα νόμιμα web αιτήματα ώστε να ανιχνεύει «κρυφές» επικοινωνίες, backdoors και δραστηριότητες κατασκοπευτικού λογισμικού που περνούν από το κανάλι χρησιμοποιώντας HTTP συνδέσεις. Το Web Tap αναλύει την εξερχόμενη HTTP κίνηση από τα προστατευμένα δικτυακά μηχανήματα σε εξωτερικούς υπολογιστές. Η τακτική αυτή είναι προτιμότερη από το να προστατεύει τους web servers από επιθέσεις. Σκοπός είναι να γίνει δυσκολότερο για τους hackers ή τους κακόβουλους χρήστες το τρέξιμο ενός Δούρειου Ίππου και προγραμμάτων HTTP tunnel μέσα σε ένα οργανισμό τα οποία έχουν ως στόχο την αποκάλυψη πληροφοριών του οργανισμού αυτού. Το Web Tap σχεδιάστηκε για να ενισχύσει (ενεργητικά ή παθητικά) τον HTTP proxy server του οργανισμού στην ανίχνευση ανωμαλιών της εξερχόμενης κίνησης.

3.7.2 Dragon IDS

Το Dragon IDS (από την αμερικάνικη εταιρεία Enterasys) είναι ένα σύστημα που παρακολουθεί το δίκτυο και αναφέρει την κακόβουλη δραστηριότητα που ανιχνεύει [37]. Τρέχει σε Unix συστήματα και βασίζεται σε συνδυασμό των τεχνικών αναγνώρισης υπογραφής και ανίχνευσης διαταραχών. Ανιχνεύει πιθανές δικτυακές επιθέσεις (από τις πληροφορίες που έχει στη βιβλιοθήκη του) μα το πιο σημαντικό είναι ότι ανιχνεύει λάθη και τρύπες ασφαλείας του συστήματος που προστατεύει και επιθέσεις άρνησης εξυπηρέτησης.

3.7.3 Fortinet's IDS

Τα συστήματα ανίχνευσης εισβολέων της Fortinet βασίζονται στο συνδυασμό της ανίχνευσης διαταραχών και αναγνώριση υπογραφής και αναλύουν ολόκληρο το πακέτο περιλαμβανομένου της επικεφαλίδας του πακέτου, των πληροφοριών πρωτοκόλλου, των πληροφοριών εφαρμογής, του περιεχόμενο πακέτου και της δραστηριότητας της εφαρμογής (session activity). Χρησιμοποιώντας έξι διαφορετικές διαδικασίες ανίχνευσης διαταραχών σε συνδυασμό με το δυναμικό σύστημα πρόληψης απειλών (Dynamic Threat Prevention System) για την πραγματοποίηση των ελέγχων, συγκεντρώνονται και εν συνεχεία αναλύονται οι πληροφορίες ώστε να ανιχνευτούν και οι πιο εξελιγμένες απειλές γνωστές ως "zero hours" επιθέσεις. Μερικά είδη επιθέσεων που ανιχνεύονται είναι:

- Δικτυακά σκουλήκια και Δούρειοι Ίπποι
- Επιθέσεις άρνησης εξυπηρέτησης
- Spoofing
- Επιθέσεις Brute Force
- Buffer overflow

Η Fortinet επισημαίνει ότι για την καλύτερη προστασία από επιθέσεις κοινωνικής μηχανικής απαιτείται οι χρήση όλων των μηχανισμών ασφαλείας, όπως firewall, antivirus, web-content filtering, spyware και IDS.

3.8 Συμπέρασμα

Η κοινωνική μηχανική αποτελεί μία αποτελεσματική και επικίνδυνη μέθοδο την οποία χρησιμοποιούν οι κακόβουλοι χρήστες να αποκτήσουν στοιχεία και πληροφορίες από ένα σύστημα. Η κοινωνική μηχανική θεωρείται μεγάλη απειλή για τα πληροφοριακά συστήματα για το λόγο απαιτείται να ληφθούν περισσότερα μέτρα για την προστασία από αυτή. Καταφέρνει και διαπερνά τη τεχνολογία άμυνας που σχεδιάστηκε για να ανιχνεύει κακόβουλη δραστηριότητα όπως τα λογισμικά προστασίας από τους ιούς, τα firewalls, και τα συστήματα ανίχνευσης εισβολέων.

Οι μέθοδοι και οι τεχνικές που μπορούν να συντελέσουν στην επιτυχία μίας επίθεσης κοινωνικής μηχανικής είναι πολλές μα στην ουσία απαιτείται ένας μόνο απρόσεχτος χρήστης ο οποίος θα αφήσει το σύστημα ευάλωτο και θα βοηθήσει στην επιτυχία της επίθεσης. Για να μειωθεί ο κίνδυνος των επιθέσεων η πιο αποτελεσματική μέθοδος είναι η συνδυαστική χρήση των μηχανισμών ασφαλείας, οι απαραίτητες ρυθμίσεις στα συστήματα ανίχνευσης εισβολέων, η κατάλληλη εκπαίδευση των χρηστών και η εφαρμογή των πολιτικών ασφαλείας που υπάρχουν σε κάθε σύστημα.

Ενότητα

4

ΥΛΟΠΟΙΗΣΗ

Όπως αναλύθηκε στην προηγούμενη ενότητα, τα συστήματα ανίχνευσης εισβολέων πρέπει να ρυθμιστούν κατάλληλα για να εντοπίσουν επιθέσεις κοινωνικής μηχανικής. Στην συγκεκριμένη ενότητα αναλύεται μία επέκταση του ανοιχτού συστήματος προσδιορισμού εισβολέων Snort έτσι ώστε να είναι σε θέση να αντιλαμβάνεται συγκεκριμένα γεγονότα που υποδηλώνουν επιθέσεις κοινωνικής μηχανικής. Αρχικά περιγράφεται το περιβάλλον που θα γίνουν τα πειράματα ενώ στην συνέχεια ακολουθεί η περιγραφή του εργαλείου Snort. Στην τελευταία υποενότητα αναλύονται τα σενάρια και τα πειράματα υλοποίησης που πραγματοποιήθηκαν χρησιμοποιώντας το Snort.

4.1 Περιβάλλον Υλοποίησης

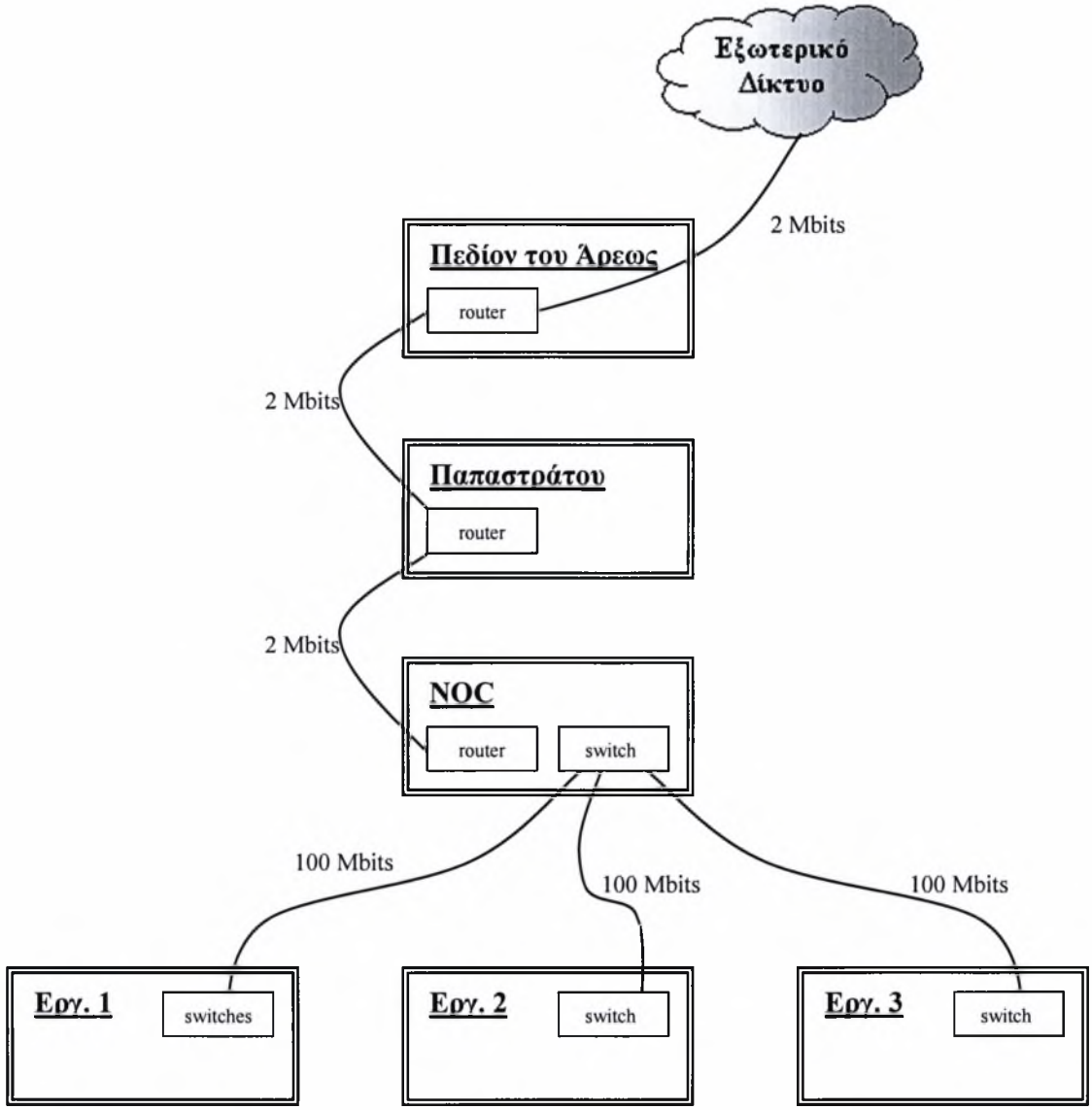
Τα πειράματα υλοποίησης πραγματοποιήθηκαν στα εργαστήρια της Πολυτεχνικής σχολής "Μηχανικών Ηλεκτρονικών Υπολογιστών Τηλεπικοινωνιών και Δικτύων" του Πανεπιστημίου Θεσσαλίας.

Υπάρχουν τρία εργαστήρια των οποίων οι συσκευές παρουσιάζονται λεπτομερέστερα στον ακόλουθο πίνακα.

Συσκευές	*Εργαστήριο 1*	*Εργαστήριο 2*	*Εργαστήριο 3*
Υπολογιστές	40	20	12
Switches	2 x 24 ports	1 x 24 ports	1 x 24 ports

Οι υπολογιστές του κάθε εργαστηρίου συνδέονται με τα αντίστοιχα switches που υπάρχουν σε καθένα από αυτά. Στη συνέχεια τα switches ενώνονται με το switch του Κέντρου Δικτύων (NOC) και η κίνηση δρομολογείται, μέσω ενός router, από το NOC στο κτήριο Παπαστράτου και από εκεί, πάλι με τη βοήθεια ενός router, στο Πεδίον του Άρεως για να καταλήξει τελικά να συνδεθεί με το GUnet (εξωτερικό δίκτυο).

Στο ακόλουθο σχήμα φαίνεται μία ενδεικτική αναπαράσταση των όσων αναφέρθηκαν παραπάνω.



Οι subnets masks που χρησιμοποιεί το δίκτυο είναι οι:

- 195.251.17.192/27
- 195.251.18.128/26
- 194.177.204.64/26

4.2 Περιγραφή Του Snort

Στη συγκεκριμένη υποενότητα περιγράφεται το δικτυακό σύστημα προσδιορισμού εισβολέων Snort του οποίου δημιουργός του είναι ο Martin Roesch. Το Snort παρουσιάζεται για να γίνουν πιο κατανοητά τα πειράματα υλοποίησης που περιγράφονται στην υποενότητα 4.3.

4.2.1 Εισαγωγή

Το Snort είναι ένα open source δικτυακό σύστημα προσδιορισμού εισβολέων το οποίο παρακολουθεί και αναλύει την κίνηση του δικτύου και πραγματοποιεί καταγραφή πακέτων [39]. Παρόλο που έχουν αναπτυχθεί πολλά συστήματα ανίχνευσης εισβολέων, εμπορικά αλλά και open source, το Snort θεωρείται ένα από τα πιο δημοφιλή εργαλεία για την ανίχνευση δικτυακών επιθέσεων. Χρησιμοποιείται από πολλές εταιρείες οι οποίες δεν μπορούν να αντέξουν το κόστος των εμπορικών συστημάτων ανίχνευσης εισβολέων και αποδεικνύεται αποτελεσματικό γιατί μπορεί ο καθένας να αναπτύξει τον κώδικα του Snort ανάλογα με τις ανάγκες του.

Χαρακτηρίζεται ως lightweight NIDS αφού είναι μικρό σε μέγεθος, εύκολο στη χρήση και εφαρμόζεται σε σχετικά μικρά δίκτυα. Το Snort βασίζεται σε κανόνες και χρησιμοποιεί υπογραφές για να αναγνωρίσει τους διαφόρους τύπους επιθέσεων. Μάλιστα, η ομάδα Sourcefire Vulnerability Research Team (VRT) αναζητά συνεχώς νέες ευπάθειες και δημιουργεί νέους κανόνες για την προστασία από αυτές. Με τους κανόνες αυτούς μπορεί ο χρήστης να αναβαθμίζει την έκδοση του Snort που χρησιμοποιεί.

Όπως αναφέρεται και στο επίσημο δικτυακό τόπο του Snort (www.snort.org), το Snort πραγματοποιεί ανάλυση πρωτοκόλλου, αναζήτηση και ταίριασμα περιεχομένου (content searching/matching) και μπορεί να χρησιμοποιηθεί για την ανίχνευση μιας πληθώρας επιθέσεων, όπως υπερχείλιση μνήμης (buffer overflows), stealth port scans, CGI επιθέσεις, SMB probes, και προσπάθειες OS fingerprinting. Οι επιθέσεις ανιχνεύονται σε πραγματικό χρόνο και ενεργοποιούνται οι κατάλληλοι συναγερμοί. Το Snort παρέχει πληροφορίες για τον τρόπο πραγματοποίησης των επιθέσεων, την

προέλευση τους καθώς και άλλα στοιχεία που τις αφορούν. Ένας από τους στόχους του είναι να μεγιστοποιεί τους σωστούς συναγερμούς και να ελαχιστοποιεί τους λανθασμένους. Το Snort είναι συμβατό με πολλές πλατφόρμες, όπως Windows, Linux, Solaris, MacOS X, BSD, και TRU-64.

4.2.2 Καταστάσεις Λειτουργίας

Το Snort χαρακτηρίζεται από τρεις καταστάσεις λειτουργίας (modes) [40]:

- Sniffer mode
- Packet Logger mode
- Network Intrusion Detection (NIDS) mode,

οι οποίες αναλύονται παρακάτω.

4.2.2.1 Sniffer mode

Σε αυτή την κατάσταση λειτουργίας το Snort διαβάζει τα πακέτα που μεταφέρονται στο δίκτυο και τα παρουσιάζει με μία συνεχόμενη ροή στην οθόνη του χρήστη. Επιπλέον, ο χρήστης έχει τη δυνατότητα να καθορίσει το είδος των πακέτων που του εμφανίζονται στην οθόνη ανάλογα με τα χαρακτηριστικά του πακέτου που αυτός επιθυμεί, όπως τον αποστολέα, τον παραλήπτη ή το πρωτόκολλο του πακέτου. Αν ο χρήστης επιθυμεί την εμφάνιση των επικεφαλίδων των TCP/IP πακέτων, τότε θα πρέπει να χρησιμοποιήσει την εντολή:

./snort -v

Αν επιθυμεί την εμφάνιση των δεδομένων και των επικεφαλίδων, θα πρέπει να χρησιμοποιήσει την εντολή:

./snort -vd

Αν επιθυμεί την εμφάνιση των επικεφαλίδων του Link Layer, θα πρέπει να χρησιμοποιήσει την εντολή:

./snort -vde ή ./snort -d -v -e

4.2.2.2 Packet Logger mode

Σε αυτή την κατάσταση λειτουργίας το Snort αποθηκεύει στο σκληρό δίσκο τα πακέτα που μεταφέρονται στο δίκτυο με την εντολή:

./snort -dev -l ./log

Ο χρήστης θα πρέπει να προσδιορίσει ένα logging directory και το Snort αυτόματα θα μεταβεί σε αυτή την κατάσταση λειτουργίας. Ας υποθέσουμε ότι υπάρχει ένας κατάλογος με το όνομα log στον τρέχοντα κατάλογο. Με την εντολή:

./snort -dev -l ./log

το σύστημα μεταβαίνει στην κατάσταση λειτουργίας Packet Logger. Όταν το Snort τρέχει σε αυτό το mode, συλλέγει κάθε πακέτο που βλέπει και το τοποθετεί μέσα σε ένα κατάλογο, ανάλογα με την IP που έχει καθένας από τους hosts του datagram.

Με την εντολή:

./snort -dev -l ./log -h 192.168.1.0/24

το Snort εκτυπώνει τις data link και TCP/IP επικεφαλίδες και καταγράφει τα δεδομένα στον κατάλογο ./log. Η διεύθυνση 192.168.1.0/24 αναφέρει το δίκτυο στο οποίο βρίσκεται ο χρήστης.

Επιπλέον, ο χρήστης έχει τη δυνατότητα να επιλέξει σε ποια μορφή θα αποθηκευτούν τα πακέτα. Μπορεί να είναι σε δυαδική (binary), αν χρησιμοποιηθούν για μετέπειτα ανάλυση, σε ASCII ή XML μορφή. Για να αποθηκευτούν σε δυαδική μορφή χρησιμοποιείται η εντολή:

./snort -l ./log -b

Στην κατάσταση αυτή τα πακέτα αποθηκεύονται σε tcpdump μορφή σε ένα δυαδικό αρχείο στο φάκελο καταχώρισης.

Αν ο χρήστης επιθυμεί την εμφάνιση συγκεκριμένων πακέτων, όπως icmp πακέτα, τότε θα πρέπει να ορίσουμε στην εντολή ένα BPF φίλτρο. Για παράδειγμα η εντολή:

./snort -dvr packet.log icmp

μας επιτρέπει να δούμε μόνο τα icmp πακέτα:

4.2.2.2 Network Intrusion Detection mode

Αυτή είναι η κύρια και η πιο πολύπλοκη κατάσταση λειτουργίας του Snort. Επιτρέπει στο Snort να αναλύει τη δικτυακή κίνηση, να εξετάζει αν πραγματοποιούνται δραστηριότητες που αντιτίθενται στους κανόνες που του έχουν τεθεί, δηλαδή να εξετάζει αν πραγματοποιείται μία επίθεση, και να αντιδρά ανάλογα. Με την εντολή:

./snort -dev -l ./log -h 192.168.1.0/24 -c snort.conf

το Snort λειτουργεί ως NIDS, όπου snort.conf είναι το σύνολο των κανόνων. Το Snort εφαρμόζει τους κανόνες σε κάθε πακέτο που μεταφέρεται στο δίκτυο και σε περίπτωση που οι κανόνες ικανοποιούνται, εκτελούνται οι αντίστοιχες ενέργειες που αναφέρονται σε αυτούς.

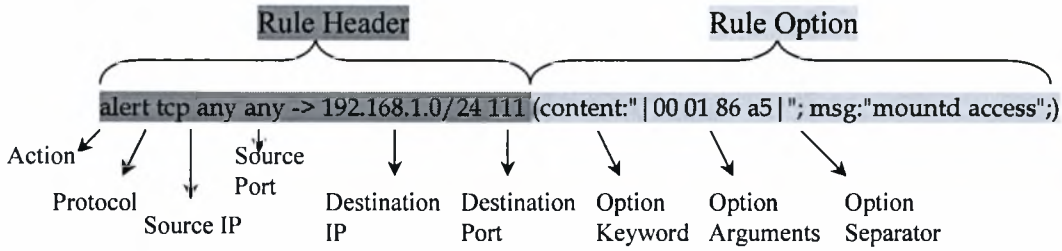
Στη συνέχεια ακολουθούν οι εντολές εξόδου (Output Options) για NIDS Mode. Υπάρχουν πολλοί τρόποι για τη μορφοποίηση της εξόδου του Snort σε κατάσταση λειτουργίας NIDS. Η καταγραφή των πακέτων πραγματοποιείται σε ASCII ή δυαδική μορφή. Όσον αφορά τους συναγερμούς διακρίνονται έξι διαφορετικές καταστάσεις, οι οποίες παρουσιάζονται στον ακόλουθο πίνακα:

Επιλογή	Περιγραφή
- A fast	Fast alert mode. Γράφει το συναγερμό σε απλή μορφή με μία χρονοσφραγίδα (timestamp), με ένα μήνυμα συναγερμού και με τα IPs/ports πηγής και προορισμού.
- A full	Full alert mode. Είναι το default alert mode και χρησιμοποιείται αυτόματα όταν δεν έχουμε ορίσει διαφορετική κατάσταση συναγερμού.
- A unsock	Στέλνει το συναγερμό σε ένα UNIX socket όπου ένα άλλο πρόγραμμα μπορεί να το ακούσει.
- A none	Απενεργοποιεί το συναγερμό.
- A console	Στέλνει "fast style" συναγερμούς στην οθόνη.
- A cmg	Δημιουργεί "cmd style" συναγερμούς.

4.2.3 Snort Rules

Το Snort χρησιμοποιεί μία απλή και "lightweight" γλώσσα περιγραφής κανόνων που είναι ευέλικτη και αρκετά ισχυρή. Οι περισσότερες κανόνες του Snort γράφονται σε μία απλή γραμμή. Αν οι κανόνες εκτείνονται σε περισσότερες γραμμές χρησιμοποιούμε τον τελεστή \ στο τέλος της γραμμής. Οι κανόνες περιγράφουν τα χαρακτηριστικά ενός πακέτου που μπορεί να είναι μέρος μίας γνωστής επίθεσης και την ενέργεια που θα εκτελεστεί αν ένα τέτοιο πακέτο ανιχνευτεί. Το Snort εντοπίζει πακέτα και τα ελέγχει αν έχουν τα ίδια χαρακτηριστικά με αυτά που περιγράφονται τους κανόνες. Οι κανόνες αποτελείται από δύο λογικά μέρη, τον rule header και το rule options. Το τμήμα rule header περιέχει το πρωτόκολλο, την IP διεύθυνση της πηγής και του προορισμού, τις μάσκες δικτύου (netmasks) καθώς τα ports της πηγής και του προορισμού. Το τμήμα rule option περιέχει μηνύματα συναγερμού (alert messages) και πληροφορίες τις οποίες πρέπει να ελέγξει το πακέτο για να καθορίσει

αν θα κάνει την πράξη που ορίζει ο κανόνας (rule action). Στη συνέχεια δίνεται ένα παράδειγμα κανόνα του Snort:



4.2.3.1 Rules Headers

4.2.3.1.1 Rule Actions

Το τμήμα rule header περιέχει την πληροφορία που προσδιορίζει τα χαρακτηριστικά του πακέτου καθώς και την ενέργεια που πρέπει να εκτελεστεί αν εντοπιστεί ένα πακέτο που ικανοποιεί τα κριτήρια του κανόνα. Το πρώτο στοιχείο σε έναν κανόνα είναι το rule action. Το rule action είναι η ενέργεια που θα εκτελέσει το Snort όταν ταιριάξει κάποιο πακέτο με τα κριτήρια ενός κανόνα. Υπάρχουν πέντε διαθέσιμες default actions:

- **Alert** – δημιουργεί ένα συναγερμό (alert) για το γεγονός που εντόπισε και καταγράφει (log) το πακέτο
- **Log** – καταγράφει (log) το πακέτο στο δίσκο
- **Pass** – αγνοεί το πακέτο
- **Activate** – δημιουργεί ένα alert και ενεργοποιεί έναν άλλον dynamic rule
- **Dynamic** – περιμένει μέχρι να ενεργοποιηθεί από ένα activate rule και στη συνέχεια ενεργεί σαν ένα log rule.

Ο χρήστης μπορεί να δημιουργήσει δικούς του τύπους κανόνων και να συσχετίσει με αυτούς ένα ή περισσότερα output plug-ins. Στη συνέχεια μπορεί να χρησιμοποιήσει αυτούς τους τύπους κανόνων σαν action στους κανόνες του Snort.

4.2.3.1.2 Protocols

Το επόμενο τμήμα σε ένα κανόνα είναι το πρωτόκολλο. Το Snort αναγνωρίζει τέσσερα πρωτόκολλα, στα οποία ανήκουν τα πακέτα, για να εντοπίσει ύποπτη

συμπεριφορά. Το πρωτόκολλο μπορεί να είναι tcp, udp, icmp και ip. Μελλοντικά θα είναι παραπάνω, όπως ARP, IGRP, GRE, OSPF, RIP και IPX.

4.2.3.1.3 IP Addresses

Το επόμενο τμήμα στο rule header είναι η IP διεύθυνση και οι port πληροφορίες για ένα κανόνα. Η λέξη any χρησιμοποιείται για να δηλώσει οποιαδήποτε IP διεύθυνση. Οι διευθύνσεις είναι εκφρασμένες με μία αριθμητική IP διεύθυνση και ένα CIDR block. Το CIDR block δηλώνει τη μάσκα δικτύου (netmask) που θα έπρεπε να δοθεί στη διεύθυνση του κανόνα και σε όλα τα εισερχόμενα πακέτα τα οποία έχουν ελεγχθεί και αντιτίθενται σε αυτόν. Μία CIDR block μάσκα από /24 δηλώνει ένα δίκτυο τάξης C (Class C network), /16 ένα δίκτυο τάξης B και /32 δηλώνει διεύθυνση ενός συγκεκριμένου μηχανήματος. Για παράδειγμα, ο συνδυασμός διεύθυνσης/CIDR 192.168.1.0/24 δηλώνει το μπλοκ διευθύνσεων από 192.168.1.1 έως 192.168.1.255. Επομένως, η IP διεύθυνση αποστολέα μπορεί να ανήκει σε μία από τις διευθύνσεις 192.168.1.1 έως 192.168.1.255.

4.2.3.1.4 Port Numbers

Τα port numbers μπορούν να καθοριστούν με πολλούς τρόπους, ως any ports, static port definitions, ranges ή negation. Τα any ports χαρακτηρίζονται ριψοκίνδυνη τακτική. Τα static ports καθορίζονται από ένα μοναδικό port number, όπως 23 για telnet ή 80 για http. Τα port ranges δηλώνονται με τον τελεστή εύρους :. Τα port negation δηλώνονται χρησιμοποιώντας τον τελεστή άρνησης !. Ο τελεστής άρνησης μπορεί να εφαρμοστεί ενάντια σε όλους τους άλλους τύπους κανόνων (εκτός από το any). Αν για παράδειγμα κάποιος θέλει να καταγράψει όλα τα ports εκτός από τα X Windows ports, θα έπρεπε να γράψει τον ακόλουθο κανόνα:

log tcp any any -> 192.168.1.0/24 !6000:6010

4.2.3.1.5 The Director Operator

Ο direction operator δηλώνει την κατεύθυνση της κίνησης στην οποία εφαρμόζεται ο κανόνας. Η IP διεύθυνση και τα port numbers στο αριστερό μέρος του direction operator δηλώνουν την κίνηση που προέρχεται από τον υπολογιστή – πηγή ενώ το δεξί μέρος του direction operator δηλώνει τον υπολογιστή – προορισμό. Υπάρχει επίσης ένας διπλής κατεύθυνσης τελεστής (bi-directional operator), που συμβολίζεται

με το σύμβολο <>. Ο τελεστής αυτός υποδεικνύει στο Snort να λάβει υπόψιν του τα ζεύγη δεδομένων διεύθυνση/port τόσο για την πηγή όσο και για τον προορισμό. Αυτό είναι χρήσιμο για την καταγραφή και ανάλυση και των δύο μερών μίας συζήτησης, όπως telnet ή POP3 sessions. Ένα παράδειγμα καταγραφής ενός διπλής κατεύθυνσης τελεστή που χρησιμοποιείται για την καταγραφή των δύο μερών ενός telnet session είναι:

log tcp !192.168.1.0/24 any <> 192.168.1.0/24 23

4.2.3.1.6 Activate/Dynamic Rules

Τα ζεύγη κανόνων activate/dynamic rule δίνουν στο Snort μία ισχυρή δυνατότητα. Ένας κανόνας μπορεί να ενεργοποιεί (activate) ένα άλλο όταν η ενέργειά του εκτελεστεί για έναν αριθμό πακέτων. Αυτό είναι χρήσιμο όταν θέλει κάποιος να προγραμματίσει το Snort έτσι ώστε αυτό να συνεχίζει να καταγράφει την κίνηση των πακέτων όταν σταματά η εκτέλεση κάποιου κανόνα. Οι activate κανόνες συμπεριφέρονται σαν alert rules, με τη διαφορά ότι έχουν ένα απαιτούμενο πεδίο επιλογής "activates" και ειδοποιούν το Snort να προσθέσει ένα κανόνα όταν κάποιο συγκεκριμένο γεγονός πραγματοποιηθεί στο δίκτυο. Οι dynamic rules κανόνες συμπεριφέρονται σαν log rules, με τη διαφορά ότι έχουν ένα διαφορετικό πεδίο επιλογής: "activated_by" και ενεργοποιούνται δυναμικά όταν ένας activate rule id τελειώνει. Επίσης, οι dynamic κανόνες έχουν ένα ακόμη πεδίο επιλογής "count".

4.2.3.2 Rule Options

Τα rule options μορφοποιούν την καρδιά της μηχανής ανίχνευσης εισβολέων του Snort. Τα rule options περιέχουν πληροφορίες που αναφέρονται στα χαρακτηριστικά για τα οποία θα ελεγχθεί το πακέτο ώστε στη συνέχεια να καθοριστεί αν θα πραγματοποιηθεί η πράξη που ορίζει ο κανόνας.

Όλα τα Snort rule options διαχωρίζονται μεταξύ τους με το χαρακτήρα ;. Τα rule option keywords είναι τα λεκτικά που δηλώνουν τα είδη των options. Τα option arguments είναι οι παράμετροι που δέχονται τα options σε σχέση με τις οποίες θα ελεγχθεί το πακέτο. Τα option arguments για κάθε option διαχωρίζονται με το χαρακτήρα : από το αντίστοιχο option keyword.

Υπάρχουν τέσσερις βασικές κατηγορίες rule options.

- **Meta-data:** Αυτά τα options παρέχουν πληροφορίες για τον κανόνα μα δεν έχουν καμία επιρροή κατά την διάρκεια της ανίχνευσης.
- **Payload:** Αυτά τα options ελέγχουν τα δεδομένα των πακέτων για «ωφέλιμα δεδομένα» (payload).
- **Non-payload**: Αυτά τα options ελέγχουν για μη ωφέλιμα δεδομένα.
- **Post-detection:** Αυτά τα options είναι συγκεκριμένοι κανόνες που πραγματοποιούνται όταν ένας κανόνας σταματήσει να εκτελείται.

4.2.3.2.1 Meta-Data Rule Options

Στη συνέχεια αναλύονται τα Meta-Data Rule Options:

- **msg** – Το msg rule option δηλώνει ότι μαζί με την καταγραφή του πακέτου και το alert θα τυπωθεί και κάποιο μήνυμα το οποίο βρίσκεται μέσα σε " ", για παράδειγμα, msg: "<message text>";
- **reference** – Το reference keyword επιτρέπει στους κανόνες να περιέχουν αναφορές σε εξωτερικά συστήματα αναγνώρισης επιθέσεων. Το plug-in υποστηρίζει συστήματα αλλά και URLs και χρησιμοποιείται από εξόδους plug-in για την παροχή ενός συνδέσμου με επιπρόσθετες πληροφορίες για το alert που πραγματοποιήθηκε.
- **sid** – Το sid keyword χρησιμοποιείται για να προσδιορίσει μοναδικά τους κανόνες του Snort. Αυτή η πληροφορία επιτρέπει σε εξόδους plug-ins να αναγνωρίζουν εύκολα κανόνες. Αυτό το option πρέπει να χρησιμοποιείται με το rev keyword (το οποίο περιγράφεται παρακάτω).
- **rev** – Το rev keyword χρησιμοποιείται για να αναγνωρίσει επαναλήψεις των κανόνων του Snort. Οι επαναλήψεις, χρησιμοποιώντας το id του κανόνα, επιτρέπουν την αντικατάσταση των υπογραφών και των περιγραφών με νέες αναβαθμισμένες πληροφορίες. Αυτό το option χρησιμοποιείται μαζί με το sid keyword.
- **classtype** – Το classtype keyword κατηγοριοποιεί τα alerts ανάλογα με τον τύπο της επίθεσης. Ο χρήστης μπορεί να προσδιορίσει τι προτεραιότητα έχει κάθε τύπος κατηγοριοποίησης. Οι κανόνες που έχουν κατηγοριοποιηθεί έχουν ένα default priority set.

- **priority** – Η ετικέτα προτεραιότητας αναθέτει ένα επίπεδο ασφαλείας στους κανόνες. Ένας classtype rule αναθέτει μία default προτεραιότητα η οποία μπορεί να υπερβεί την προτεραιότητα του κανόνα.

4.2.3.2.2 Payload Detection Rule Options

Στη συνέχεια αναλύονται τα Payload Detection Rule Options:

- **content** – Το content keyword είναι ένα από τα πιο σημαντικά χαρακτηριστικά του Snort. Επιτρέπει στο χρήστη να θέτει κανόνες οι οποίοι αναζητούν συγκεκριμένο περιεχόμενο στο πακέτο. Όταν εντοπίζεται ταίριασμα περιεχομένου τότε καλείται η συνάρτηση Boyer-Moore για να επιβεβαιωθεί απολύτως το ταίριασμα περιεχομένου. Αν τα δεδομένα ταιριάζουν ακριβώς τότε το argument data string περιέχεται οπουδήποτε στο payload του πακέτου, το τεστ θεωρείται επιτυχές και στη συνέχεια εκτελούνται οι εναπομείναντες διαδικασίες από τα rule option tests. Το content keyword έχει έναν αριθμό από modifiers keywords. Τα modifiers keywords αλλάζουν τη συμπεριφορά το προηγούμενου περιεχόμενου. Τα modifiers keywords είναι τα ακόλουθα:
  1. **depth** – Επιτρέπει στο χρήστη που γράφει τον κανόνα να καθορίσει μέχρι πιο σημείο το Snort θα αναζητά σε ένα πακέτο ένα συγκεκριμένο πρότυπο (pattern). Για παράδειγμα, αν ο κανόνας δηλώνει ότι το βάθος είναι 5 το Snort θα ψάξει για το συγκεκριμένο πρότυπο μέχρι τα 5 πρώτα bytes του payload.
  2. **offset** – Επιτρέπει στο χρήστη που γράφει τον κανόνα να καθορίσει το σημείο έναρξης της αναζήτησης μέσα σε ένα συγκεκριμένο πακέτο. Ένα offset με τιμή 5 καθορίζει ότι η αναζήτηση θα ξεκινήσει μετά τα 5 πρώτα bytes του payload.
  3. **distance** – Επιτρέπει στο χρήστη που γράφει τον κανόνα να καθορίσει μέχρι πιο σημείο μέσα σε ένα πακέτο το Snort θα αγνοεί τα δεδομένα ώσπου να αρχίσει η αναζήτηση ενός συγκεκριμένου προτύπου. Η αναζήτηση ξεκινά αμέσως μετά την ολοκλήρωση του τελευταίου ταιριάσματος προτύπου που έλαβε χώρα στο πακέτο. Το distance μπορεί να θεωρηθεί όμοιο με το depth, με τη διαφορά ότι το distance σχετίζεται με το τέλος του τελευταίου ταιριάσματος προτύπου αντί να ξεκινά την αναζήτηση από την αρχή του πακέτου. Για παράδειγμα,

     alert tcp any any -> any 80 (content: "cgi-bin/phf"; offset:4; depth:20;)

4. **within** – Είναι ένα content modifier το οποίο εξασφαλίζει ότι υπάρχουν τουλάχιστον N bytes μεταξύ δύο ταιριασμάτων προτύπων. Έχει σχεδιαστεί να χρησιμοποιείται σε συνδυασμό με το distance rule option. Για παράδειγμα, alert tcp any any -> any any (content:"ABC"; content: "EFG"; within:10;)
5. **nocase** – Επιτρέπει στο χρήστη που γράφει τον κανόνα να καθορίσει ότι το Snort πρέπει να ψάξει για το συγκεκριμένο πρότυπο (pattern), αγνοώντας τις άλλες περιπτώσεις.
6. **rawbytes** – Επιτρέπει στους κανόνες να ψάξουν στα raw δεδομένα του πακέτου, αγνοώντας τυχόν αποκωδικοποιήσεις (decoding) που έγιναν από τους preprocessors.

4.2.3.2.3 Non-payload Detection Rule Options

Στη συνέχεια ακολουθεί η ανάλυση των Non-payload Detection Rule Options:

- **fragoffset** – Επιτρέπει τη σύγκριση του IP fragment offset με μία δεκαδική τιμή. Για να εντοπιστούν όλα τα πρώτα fragments ενός IP session, χρησιμοποιείται το keyword fragbits (το οποίο εξηγείται παρακάτω) και αναζητούνται τα More fragments option σε συνδυασμό με τιμή του fragoffset ίση με 0.
- **ttl** – Χρησιμοποιείται για τον έλεγχο της τιμής IP time-to-live. Αυτό το option keyword έχει ως σκοπό την ανίχνευση traceroute προσπαθειών.
- **tos** – Χρησιμοποιείται για τον έλεγχο του πεδίου IP TOS για μία συγκεκριμένη τιμή.
- **id** – Χρησιμοποιείται για τον έλεγχο του πεδίου IP ID για μία συγκεκριμένη τιμή.
- **ipopts** – Χρησιμοποιείται για να ελέγχει αν υπάρχει ένα συγκεκριμένο IP option. Μερικά IP options είναι τα ακόλουθα:
  - **rr** - Record route
  - **eol** - End of list
  - **nop** - No op
  - **ts** - Time Stamp
  - **sec** - IP security option
  - **lsrr** - Loose source routing
  - **ssrr** - Strict source routing
  - **satid** - Stream identifier
  - **any** - any IP options are set

- **fragbits** – Χρησιμοποιείται για να ελέγχει αν έχει γίνει fragmentation και αν έχουν ανατεθεί reserved bits στον IP header. Μπορούν να ελεγχθούν τα ακόλουθα bits:
  - **M** - More Fragments
  - **D** - Don' t Fragment
  - **R** - Reserved Bit
- **dsize** – Χρησιμοποιείται για να ελέγχει το μέγεθος του payload του πακέτου. Επίσης, χρησιμοποιείται για να εντοπίζει πακέτα που έχουν παράξενο μέγεθος. Σε πολλές περιπτώσεις, είναι χρήσιμο στην ανίχνευση υπερχείλισης μνήμης (buffer overflows).
- **flags** – Χρησιμοποιείται για να ελέγχει αν υπάρχουν συγκεκριμένα TCP flag bits. Μερικά TCP flag bits είναι:
  - **F** - FIN (LSB in TCP Flags byte)
  - **S** - SYN
  - **R** - RST
  - **P** - PSH
  - **A** - ACK
  - **U** - URG
  - **1** - Reserved bit 1 (MSB in TCP Flags byte)
  - **2** - Reserved bit 2
  - **0** - No TCP Flags Set
- **flow** – Επιτρέπει στους κανόνες να εφαρμόζονται σε συγκεκριμένες κατευθύνσεις της κίνησης. Τους επιτρέπει, επίσης, να εφαρμόζονται σε clients ή servers.
- **flowbits** – Επιτρέπει στους κανόνες να εντοπίζουν καταστάσεις (states) των πρωτοκόλλων μεταφοράς. Το flowbits είναι περισσότερο χρήσιμο για TCP sessions, καθώς επιτρέπει στους κανόνες να την κατάσταση ενός πρωτοκόλλου εφαρμογής. Υπάρχουν τα ακόλουθα έξι keywords που σχετίζονται με τα flowbits: *set, unset, toggle, isset, isnotset, noalert.*
- **seq** – Χρησιμοποιείται για τον έλεγχο μίας συγκεκριμένης TCP ακολουθίας.
- **ack** – Χρησιμοποιείται για τον έλεγχο ενός συγκεκριμένου TCP acknowledge αριθμού.
- **window** – Χρησιμοποιείται για τον έλεγχο ενός συγκεκριμένου μεγέθους TCP παραθύρου.

- **itype** – Χρησιμοποιείται για τον έλεγχο μίας συγκεκριμένης ICMP type τιμής.
- **icode** – Χρησιμοποιείται για τον έλεγχο μίας συγκεκριμένης ICMP code τιμής.
- **icmp id** – Χρησιμοποιείται για τον έλεγχο μίας συγκεκριμένης ICMP ID τιμής. Είναι χρήσιμο γιατί μερικά «κρυφά» covert channel προγράμματα χρησιμοποιούν στατικά ICMP πεδία όταν επικοινωνούν. Αυτό το plug-in αναπτύχθηκε για την ανίχνευση του stacheldraht DDoS agent.

4.2.3.2.4 Post Detection Rule Options

Ακολουθεί η ανάλυση των Non-payload Payload Detection Rule Options:

- **logto** – Ενημερώνει στο Snort να καταγράψει όλα τα πακέτα σε ένα συγκεκριμένο αρχείο καταγραφής (output log file). Είναι ιδιαίτερα χρήσιμο για τον συνδυασμό NMAP δραστηριοτήτων, HTTP CGI scans, και άλλα. Αυτό το option δεν λειτουργεί όταν το Snort είναι σε κατάσταση λειτουργίας binary logging.
- **session** – Εξάγει δεδομένα χρηστών από TCP sessions. Σε πολλές περιπτώσεις είναι χρήσιμο να γνωρίζει κάποιος τι τυπώνουν οι χρήστες στο telnet, rlogin, ftp, ή ακόμη στα web sessions. Υπάρχουν δύο διαθέσιμα argument keywords για το session rule option, printable ή all. Το printable keyword εκτυπώνει μόνο τα δεδομένα που ο χρήστης φυσιολογικά θα έβλεπε ή θα τύπωνε. Το all keyword αντικαθιστά τους μη printable χαρακτήρες με τις αντίστοιχες ισοδυναμίες τους σε δεκαεξαδικό σύστημα.
- **resp** – Προσπαθεί τον τερματισμό των sessions όταν ενεργοποιηθεί ένας συναγερμός. Στο Snort αυτό ονομάζεται «ευέλικτη απόκριση» (flexible response). Τα flexible response υποστηρίζουν τους ακόλουθους μηχανισμούς για τον τερματισμό των sessions:

Επιλογή	Περιγραφή
rst_snd	Στέλνει TCP-RST πακέτα στο socket πηγής
rst_rcv	Στέλνει TCP-RST πακέτα στο socket προορισμού
rst_all	Στέλνει TCP RST πακέτα και προς τις δύο κατευθύνσεις
icmp_net	Στέλνει a ICMP NET UNREACH στον αποστολέα
icmp_host	Στέλνει a ICMP HOST UNREACH στον αποστολέα
icmp_port	Στέλνει a ICMP PORT UNREACH στον αποστολέα
icmp_all	Στέλνει όλα τα παραπάνω ICMP πακέτα packets στον αποστολέα

- **react** – Το react keyword βασίζεται στις flexible response και υλοποιεί «ευέλικτη» αντίδραση (flexible reaction) στα πακέτα τα οποία ταιριάζουν σε ένα κανόνα του Snort. Η κύρια αντίδραση είναι το μπλοκάρισμα της κίνησης σε

ενδιαφέροντες δικτυακούς τόπους, όπως New York Times ή slashdot. Ο κώδικας των flexible response επιτρέπει στο Snort να διακόψει «παράνομες» συνδέσεις και/ή να στείλει μία προειδοποίηση στο browser. Η προειδοποίηση μπορεί να περιέχει το μήνυμα που επιθυμεί ο χρήστης (ο οποίος προγραμματίζει το Snort). Για τα παραπάνω μπορούν να χρησιμοποιηθούν τα ακόλουθα arguments (basic modifiers):

- block - διακοπή σύνδεσης και εμφάνιση μηνύματος προειδοποίησης
- warn - εμφάνιση μηνύματος προειδοποίησης (will be available soon)

✧ **tag** – Επιτρέπει στους κανόνες να καταγράφουν περισσότερα από ένα πακέτα και όχι μόνο αυτά που ταίριαζαν στα χαρακτηριστικά του κανόνα. Όταν ικανοποιείται κάποιος κανόνας, παρακολουθείται και καταγράφεται (tagged) επιπρόσθετη εξερχόμενη και εισερχόμενη κίνηση από το host. Αυτή η κίνηση καταγράφεται για να επιτρέψει την ανάλυση των response codes και της κίνησης που πραγματοποιείται μετά την επίθεση.

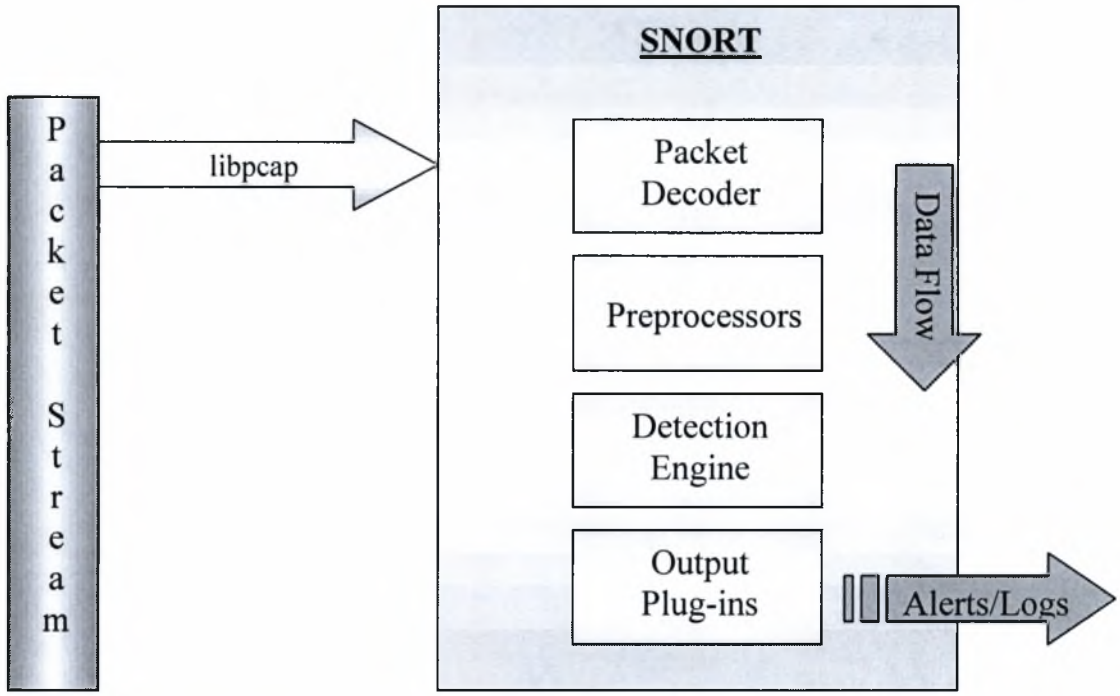
Περισσότερες πληροφορίες για τα παραπάνω μπορεί κάποιος να βρει στο δικτυακό τόπο http://www.snort.org/docs/snort_htmanuals/htmanual_233/ .

4.2.4 Αρχιτεκτονική Του Snort

Το Snort αποτελείται από τα ακόλουθα τέσσερα υποσυστήματα λειτουργίας τα οποία είναι σημαντικά για την ανίχνευση εισβολών [41]:

- Packet Decoder
- Preprocessors
- Detection Engine
- Output Plug-ins

Τα πακέτα που αναλύει το Snort περνούν και από τα τέσσερα παραπάνω υποσυστήματα.



4.2.4.1 Packet Decoder

Αρχικά υπάρχει ο μηχανισμός packet capturing, δηλαδή η συλλογή πακέτων. Η συλλογή πακέτων πραγματοποιείται χρησιμοποιώντας την βιβλιοθήκη libpcap (ή winpcap). Η βιβλιοθήκη δίνει την δυνατότητα της μεταφοράς των πακέτων από επίπεδο πυρήνα σε επίπεδο χρήστη. Στη συνέχεια ξεκινά η λειτουργία του υποσυστήματος Packet Decoder, όπου αποκωδικοποιούνται τα πακέτα και στη συνέχεια οργανώνονται σε μία εσωτερική δομή δεδομένων. Το Packet Decoder, που στην πραγματικότητα είναι μία σειρά από αποκωδικοποιητές, περιέχει συναρτήσεις που εξάγουν την απαιτούμενη πληροφορία για κάθε πακέτο, ξεκινώντας από τα πρωτόκολλα του επιπέδου Data Link Layer ανεβαίνοντας μέχρι και τα πρωτόκολλα του επιπέδου εφαρμογής. Μόλις το πακέτο αποθηκευτεί στη δομή δεδομένων είναι έτοιμο να αναλυθεί από τους preprocessors και τη detection engine.

4.2.4.2 Preprocessors

Τα preprocessors χωρίζονται σε δύο κατηγορίες:

- Εξετάζουν πακέτα για ύποπτη συμπεριφορά
- Τροποποιούν τα πακέτα ώστε να μπορεί να τα ερμηνεύσει η detection engine

Χρησιμοποιώντας τη detection engine, πολλές επιθέσεις δεν μπορούν να ανιχνευτούν, με τη μέθοδο αναγνώρισης υπογραφών. Για το λόγο αυτό τα preprocessors θεωρούνται απαραίτητα διότι έχουν τη δυνατότητα να ανιχνεύουν ύποπτη δραστηριότητα. Συνεπώς, το πρώτο είδος preprocessors είναι βασικό για την ανακάλυψη επιθέσεων που δεν βασίζονται στην αναγνώριση υπογραφών. Το δεύτερο είδος είναι υπεύθυνο για τη μετατροπή των πακέτων σε τέτοια μορφή ώστε να μπορεί η detection engine να χρησιμοποιήσει το ταίριασμα υπογραφών. Αυτά τα preprocessors σταματούν επιθέσεις οι οποίες προσπαθούν να ξεγελάσουν την detection engine παραποιώντας τα πρότυπα πακέτων.

Αφού δημιουργηθούν οι παράμετροι των preprocessors τοποθετούνται στο αρχείο *snort.conf*. Στο ίδιο αρχείο μπορεί κάποιος, ανάλογα με τις ανάγκες του, να προσθέσει ή να αφαιρέσει preprocessors. Παρακάτω αναφέρονται μερικά preprocessors:

- **frag2** – Το συγκεκριμένο preprocessor προστατεύει από επιθέσεις IP fragmentation. Έχει τη δυνατότητα να ανιχνεύει επιθέσεις που σχετίζονται με το fragmentation, είτε προέρχονται από τεχνικές IDS αποφυγής (IDS evasion) είτε από κακόβουλες DoS επιθέσεις. Το frag2 preprocessor είναι σημαντικό και δεν πρέπει ποτέ να απενεργοποιείται. Υπάρχουν οι πέντε ακόλουθες επιλογές:
  - timeout <seconds> – Δηλώνει το χρόνο που απαιτείται για να αποθηκευτεί ένα fragment. Αν δεν αποθηκευτεί μέσα σε αυτό το χρόνο, αγνοείται. Η default τιμή είναι 30 δευτερόλεπτα.
  - memcap <bytes> – Δηλώνει το αριθμό των bytes που τίθεται η memory cap. Η default τιμή είναι 4 MB.
  - detect_state_problems – Ενεργοποιεί συναγερμούς όταν πραγματοποιούνται γεγονότα όπως overlapping fragments.
  - min_ttl – Θέτει τον ελάχιστο χρόνο ζωής (minimum time to live - TTL). Η default τιμή είναι 0.
  - ttl_limit – Καθορίζει τη μέγιστη διαφορά στις τιμές TTL των fragmented πακέτων με το fragment ID που έχουν. Η default τιμή είναι 5.
- **stream4** – Το stream4 preprocessor χρησιμοποιείται για να συντηρεί την κατάσταση των TCP ροών και βοηθά στην ανίχνευση επιθέσεων που έχουν ως στόχο την συγκέντρωση πληροφοριών. Έχει δέκα διαφορετικές επιλογές.

- ✧ **Telnet_decode** – Ανήκει στους αποκωδικοποιητές preprocessors. Σχετίζεται με το Telnet και τα FTP πρωτόκολλα. Το Telnet_decode κωδικοποιεί ή αφαιρεί δυαδικούς κωδικούς έλεγχου (control codes) που έχουν εισαχθεί αυθαίρετα σε μία ροή Telnet ή FTP. Οι κακόβουλοι χρήστες εισάγουν κωδικούς έλεγχου στις επικοινωνίες για να παραπλανήσουν το Snort.
- ✧ **ARPspoof** – Σχεδιάστηκε για την ανίχνευση κακόβουλης κίνησης που χρησιμοποιεί το Address Resolution Protocol (ARP). Το ARP χρησιμοποιείται στα δίκτυα Ethernet για την αντιστοίχηση της IP address με βάση την MAC address του μηχανήματος. Υπάρχουν πολλές επιθέσεις που εμπλέκουν το ARP, με βασικότερη το ARP spoofing.

4.2.4.3 Detection Engine

Η detection engine συγκρίνει τα δεδομένα του κάθε πακέτου που έχει λάβει με τους κανόνες, για να ανιχνεύσει πιθανές εισβολές. Έχει, λοιπόν, δύο διαδικασίες:

- Την ανάλυση κανόνων (rules parsing)
- Την αναγνώριση υπογραφών (signature detection)

Η detection engine δημιουργεί τις υπογραφές επιθέσεων αναλύοντας τους κανόνες του Snort. Οι κανόνες του Snort διαβάζονται γραμμή προς γραμμή και φορτώνονται σε μία εσωτερική δομή δεδομένων. Οι κανόνες φορτώνονται μόνο όταν ξεκινά η λειτουργία του Snort, που σημαίνει ότι για να τροποποιηθεί, προστεθεί, ή διαγραφεί ένας κανόνας θα πρέπει να ανανεωθεί ο Snort daemon.

Η detection engine ελέγχει ένα μέρος του πακέτου ώστε να εντοπίσει αν περιέχει ένα συγκεκριμένο string ή μία τιμή που σχετίζεται με ένα κανόνα. Κάθε πακέτο συγκρίνεται με όλους τους κανόνες. Η detection engine αφού ολοκληρώσει τους ελέγχους ενός πακέτου προχωρά στην εξέταση του επόμενου. Όπως έχει αναφερθεί, ο κανόνας χωρίζεται σε δύο διαφορετικά τμήματα:

- το rule header και
- το rule option

Η detection engine επεξεργάζεται διαφορετικά τους rule headers και τους rule options. Δημιουργεί μία συνδεδεμένη λίστα που έχει τη μορφή ενός δέντρου απόφασης. Οι κόμβοι του δέντρου ελέγχουν κάθε εισερχόμενο πακέτο. Αν το πακέτο είναι TCP πηγαίνει στο τμήμα του δέντρου που περιέχει τους κανόνες για το TCP.

Στη συνέχεια το πακέτο ελέγχεται για το αν ταιριάζει με τη διεύθυνση πηγής ενός κανόνα. Αν αυτό ισχύει, συνεχίζεται ο έλεγχος με τα υπόλοιπα τμήματα του κανόνα. Η διαδικασία σταματά είτε όταν το πακέτο ταιριάζει με μία υπογραφή επίθεσης είτε όταν ο έλεγχος ολοκληρωθεί. Το σημαντικό είναι ότι το Snort ξεκινά τον έλεγχο ενός πακέτου αφού πρώτα έχει βρει μία υπογραφή για να το ταιριάσει με αυτό. Ακόμη και αν το πακέτο μπορούσε να ταιριάσει με άλλη υπογραφή, η detection engine συνεχίζει στο επόμενο πακέτο. Συνεπώς, είναι σημαντικό να έχουμε υπόψιν ότι κατά την οργάνωση των κανόνων οι πιο κακόβουλες υπογραφές φορτώνονται πρώτες.

4.2.4.4 Output Plug-ins

Τα Output Plug-ins εκτελούν λειτουργίες που σχετίζονται με την καταγραφή και τους συναγερμούς. Σκοπός τους είναι η γνωστοποίηση των αποτελεσμάτων του Snort στο χρήστη. Δηλαδή, αφού το Snort εντοπίζει ύποπτη δραστηριότητα, στη συνέχεια θα παράγει τις πληροφορίες που θα δει ο διαχειριστής. Μερικά γνωστά output plug-ins είναι τα:

- **alert_fast** – Εκτυπώνει τους συναγερμούς σε ένα αρχείο εξόδου. Είναι ο πιο γρήγορος τρόπος συναγερμού γιατί δεν εκτυπώνει τις επικεφαλίδες πακέτων. Συντάσσεται ως εξής: output alert_fast: alert.fast
- **alert_full** – Εκτυπώνει τους συναγερμούς μαζί με τις επικεφαλίδες πακέτων είτε σε ένα αρχείο εξόδου που θα ορίσει ο χρήστης στη γραμμή εντολών είτε σε default αρχείο καταγραφής (/var/log/snort). Συντάσσεται ως εξής: output alert_full: alert.full
- **log_tcpdump** – Καταγράφει τα πακέτα σε binary μορφή. Παίρνει ένα μόνο όρισμα, το όνομα του αρχείου εξόδου. Συντάσσεται ως εξής: output log_tcpdump: snort.log

4.2.5 Συμπέρασμα

Το Snort αποτελεί ένα από τα πιο δημοφιλή δικτυακά συστήματα εισβολέων. Είναι ένα open source λογισμικό, που ο καθένας έχει τη δυνατότητα να αναπτύξει τον κώδικά του. Αυτός είναι ένας από τους κυριότερους λόγους για τους οποίους προτιμάται ως NIDS σύστημα.

Ένα άλλο πλεονέκτημα του είναι ότι με τη βοήθεια της ομάδας Sourcefire Vulnerability Research Team, η οποία αναζητά συνεχώς νέες ευπάθειες, δημιουργούνται νέοι κανόνες για την προστασία από αυτές. Με τους κανόνες αυτούς μπορεί ο χρήστης να αναβαθμίζει την έκδοση του Snort που χρησιμοποιεί. Έτσι ο χρήστης έχει συνεχώς ένα αναβαθμισμένο σύστημα, το οποίο είναι εύκολο στη διαχείριση και τη διαμόρφωσή του.

4.3 Σενάρια Και Πειράματα Υλοποίησης

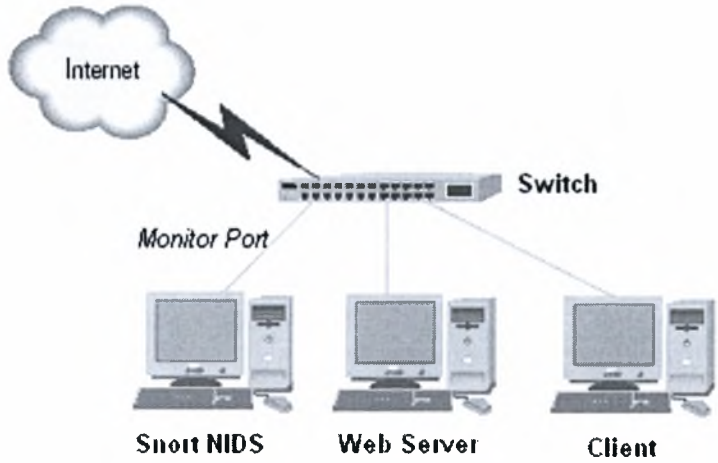
Όπως αναφέρθηκε στην υποενότητα 3.6.3 τα IDS πρέπει να ανιχνεύουν γεγονότα που υποδηλώνουν επιθέσεις κοινωνικής μηχανικής. Στόχος είναι να τροποποιηθεί το εργαλείο Snort ώστε να ανιχνεύει τέτοιες καταστάσεις. Εξαιτίας της δομής των εργαστηρίων, τα οποία δεν περιλαμβάνουν router και συνεπώς δεν μπορούσαμε να παρακολουθήσουμε την κίνηση όλου του δικτύου, τα πειράματα έγιναν ως εξής.

Χρησιμοποιήθηκαν τρεις υπολογιστές. Το Snort εγκαταστάθηκε σε έναν υπολογιστή, ένας δεύτερος υπολογιστής είχε το ρόλο του web server (όπου εγκαταστάθηκε ο Apache) και ένας τρίτος είχε το ρόλο του client. Οι τρεις αυτοί υπολογιστές ήταν συνδεδεμένοι με ένα switch. Συνεπώς, ο υπολογιστής που είχε εγκατεστημένο το Snort παρακολουθούσε και ανάλυε την κίνηση που μεταφέρονταν από τον client προς το web server και αντίστροφα. Για να επιτευχθεί αυτό, ακολουθήθηκε η διαδικασία του Port Mirroring, που αποτελεί μία μέθοδο παρακολούθησης της κίνησης σε δίκτυα που χρησιμοποιούν switch. Το port 19, που αντιστοιχεί τον υπολογιστή που έχει το Snort, παρακολουθεί την κίνηση ανάμεσα στα mirror ports 23 και 10, τα οποία αντιστοιχούν στον web server και στον client αντίστοιχα. Η εξερχόμενη και εισερχόμενη κίνηση που υπάρχει στα δύο port γίνεται duplicated στο monitoring port. Το Snort χρησιμοποιήθηκε σε κατάσταση λειτουργίας NIDS.

Οι IP διευθύνσεις των παραπάνω μηχανημάτων είναι:

Μηχάνημα	IP διεύθυνση
Snort	195.251.17.229
Web Server (Apache)	195.251.17.228
Client	195.251.17.230

Παρακάτω φαίνεται η αρχιτεκτονική του συστήματος που έγιναν τα πειράματα.



Στη συνέχεια αναφέρονται σενάρια που περιέχουν γεγονότα τα οποία σχετίζονται με επιθέσεις κοινωνικής μηχανικής καθώς και τα αποτελέσματα που προέκυψαν από τα πειράματα χρήσης και τροποποίησης του εργαλείου Snort.

4.3.1 1^{ο} Σενάριο

Ένα τέτοιο γεγονός είναι η αποστολή μη κρυπτογραφημένου κωδικού (password). Ο κοινωνικός μηχανικός μπορεί να εμφανίσει ένα pop-up παράθυρο που να υποδεικνύει στο χρηστή να δώσει το όνομά του (username) και τον κωδικό του ώστε να κάνει log on ή να ξανασυνδεθεί με το δίκτυο εξαιτίας διακοπής της σύνδεσης. Το παράθυρο αυτό είναι εικονικό, αφού έχει δημιουργηθεί από τον κακόβουλο χρήστη, και έχει ως στόχο την υποκλοπή του κωδικού.

Αλλά ακόμη και αν δεν έχει εμφανιστεί το παραπάνω εικονικό παράθυρο υπάρχει και ένας άλλος τρόπος υποκλοπής κωδικών. Οι χρήστες επισκέπτονται sites στα οποία απαιτείται να είναι εγγεγραμμένοι σε αυτά, ώστε να έχουν τη δυνατότητα να χρησιμοποιούν τις υπηρεσίες που αυτά παρέχουν. Δηλαδή, θα πρέπει να έχουν δώσει τα στοιχεία τους και να έχουν επιλέξει username και password. Υπάρχει περίπτωση κάποια από τα sites αυτά να μην αποστέλλουν κρυπτογραφημένα τα username και password που πληκτρολογεί κάθε φορά ο χρήστης προκειμένου να πραγματοποιηθεί η εξακρίβωση των στοιχείων του (αυθεντικοποίηση) ώστε να μπορεί να χρησιμοποιεί τις υπηρεσίες του συγκεκριμένου site. Αν, λοιπόν, κάποιος κακόβουλος χρήστης παρακολουθεί με ένα απλό εργαλείο (sniffer) την κίνηση του παραπάνω δικτύου, έχει τη δυνατότητα να υποκλέψει τον κωδικό που πληκτρολόγησε ο χρήστης.

Θα έλεγε κάποιος ότι αυτός ο κωδικός δεν είναι σημαντικός. Ας μην ξεχνάμε όμως αυτό που έχει αναφερθεί και σε προηγούμενες ενότητες. Οι χρήστες έχουν την τάση να χρησιμοποιούν έναν κωδικό για όλες τις συναλλαγές που πραγματοποιούν, είτε αυτές αφορούν την εταιρεία που εργάζονται, είτε τις προσωπικές τους συναλλαγές στο Διαδίκτυο. Αυτό το κάνουν για να μην χρειάζεται να θυμούνται μία πληθώρα από κωδικούς, αλλά δυστυχώς το μόνο που καταφέρνουν είναι να κάνουν ευκολότερη τη δουλειά των κοινωνικών μηχανικών. Σε περίπτωση που ο κοινωνικός μηχανικός καταφέρει να ανακαλύψει τον κωδικό, υπάρχει πιθανότητα να αποκτήσει πρόσβαση στις πληροφορίες ή το δίκτυο της εταιρείας που εργάζεται ο χρήστης.

Για τους παραπάνω λόγους, το Snort πρέπει να σηματοδοτεί συναγερμό όταν οι χρήστες δίνουν τους κωδικούς τους και αυτοί μεταφέρονται χωρίς να είναι κρυπτογραφημένοι. Δηλαδή, το Snort πρέπει να εξετάζει τα εξερχόμενα πακέτα που αποστέλλονται από τους δικτυακούς υπολογιστές σε εξωτερικούς servers και να ελέγχει αν σε αυτά περιέχονται κωδικοί χρηστών.

Αυτό επιτυγχάνεται γράφοντας κανόνες σε αρχεία τύπου rules. Τα αρχεία αυτά τοποθετούνται στο φάκελο rules του Snort. Για το συγκεκριμένο πείραμα δημιουργήθηκε το αρχείο password.rules και το τοποθετήθηκε στο φάκελο rules. Επίσης, στο αρχείο snort.conf κάναμε include το αρχείο αυτό ως εξής:
include $RULE_PATH/password.rules

Μία άλλη αλλαγή στο αρχείο snort.conf είναι ότι ορίσαμε τη μεταβλητή δικτύου HOME_NET ώστε να αντιστοιχεί την IP του υπολογιστή client. Αυτό έγινε ως εξής:
var HOME_NET 195.251.17.230

Το Snort για να ενεργοποιηθεί σε κατάσταση λειτουργίας NIDS και να καταγράψει τα πακέτα που εμείς επιθυμούμε, πληκτρολογούμε στο command prompt την ακόλουθη εντολή:
snort -dev -l ../log -h 195.251.17.230/32 -c ../etc/snort.conf

- Με την επιλογή -dev εξετάζονται τόσο οι επικεφαλίδες όσο και τα δεδομένα του κάθε πακέτου

- Με την επιλογή -l αποθηκεύονται στον κατάλογο log τα πακέτα που ικανοποιούν τα χαρακτηριστικά των κανόνων
- Με την επιλογή -h δηλώνεται το δίκτυο στο οποίο θα πραγματοποιηθεί η παρακολούθηση
- Με την επιλογή -c δηλώνεται το αρχείο κανόνων που θα χρησιμοποιηθεί και το οποίο έχει όνομα snort.conf (θα πρέπει να αναφέρεται ολόκληρο το μονοπάτι του αρχείου)

Οι κανόνες, που υπάρχουν στο αρχείο password.rules, εξετάζουν τα TCP πακέτα που εξέρχονται από τον υπολογιστή client και έχουν προορισμό το εξωτερικό δίκτυο. Επιπλέον, οι κανόνες ελέγχουν αν σε αυτά υπάρχει το περιεχόμενο password, χρησιμοποιώντας την εντολή, content: "password". Επειδή όμως, το password δεν εμφανίζεται μόνο σε αυτή τη μορφή συντάσσουμε κανόνες που περιέχουν όλες τις πιθανές μορφές που μπορεί αυτό να εμφανίζεται, όπως pass, password, passwd και άλλα. Όταν εντοπίζει τα περιεχόμενα αυτά, σηματοδοτεί συναγερμούς (alerts) και καταγράφει τα πακέτα στα οποία περιέχεται ο κωδικός. Επιπλέον, αναφέρει όλα τα χαρακτηριστικά που αφορούν το πακέτο.

Παρακάτω αναφέρονται οι ιδιαίτεροι κανόνες που χρησιμοποιούνται για να ανιχνεύσουν τη μεταφορά μη κρυπτογραφημένου κωδικού από τον προστατευόμενο host προς εσωτερικό ή εξωτερικό υπολογιστή του δικτύου. Περισσότερα για το ποιες αλλαγές και τροποποιήσεις απαιτούνται ώστε να πραγματοποιείται η ανίχνευση κωδικών δίνονται στο παράρτημα της συγκεκριμένης εργασίας.

alert tcp $HOME_NET any -> any any (msg: "unecrypted password discover attempt"; content: "password"; priority:1;))

alert tcp $HOME_NET any -> any any (msg: "unecrypted password discover attempt"; content: "passwd"; priority:1;))

alert tcp $HOME_NET any -> any any (msg: "unecrypted password discover attempt"; content: "pass"; priority:1;))

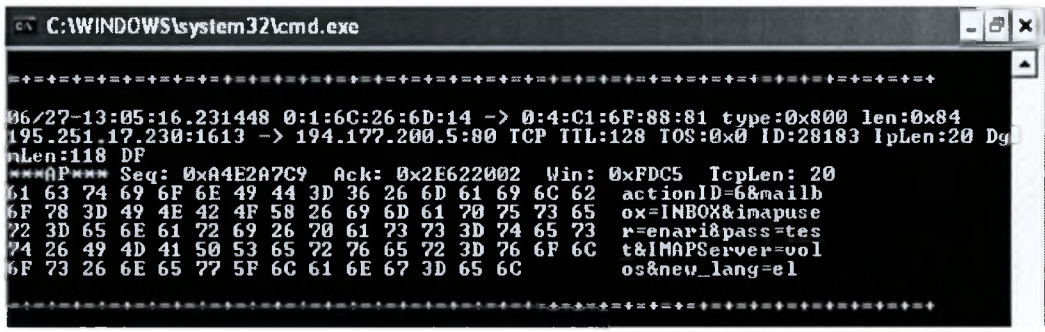
alert tcp $HOME_NET any -> any any (msg: "unecrypted password discover attempt"; content: "prd"; priority:1;))

alert tcp $HOME_NET any -> any any (msg: "unecrypted password discover attempt"; content: "pword"; priority:1;))

Ας υποθέσουμε ότι έχει εμφανιστεί στο χρήστη ένα παράθυρο και του ζητείται όνομα χρήστη και κωδικός.

Αφού ο χρήστης δώσει τα στοιχεία του το Snort θα παρακολουθήσει τα πακέτα που μεταφέρονται και αν ο κωδικός που μεταφέρεται είναι μη κρυπτογραφημένος θα συλλάβει το πακέτο.

Η παρακάτω εικόνα δείχνει πως βλέπει ο διαχειριστής στην οθόνη τα πακέτα που μεταφέρονται. Πιο συγκεκριμένα, δείχνει το πακέτο που περιέχει τον κωδικό αλλά και τα υπόλοιπα στοιχεία που έχει πληκτρολογήσει ο χρήστης. Ο κωδικός που πληκτρολόγησε ο χρήστης είναι test και τονίζεται με κόκκινη γραμμή. Όπως φαίνεται, αναφέρονται όλα τα χαρακτηριστικά του πακέτου, όπως IP πηγής και προορισμού, το είδος του πρωτοκόλλου και άλλα.



Όταν εντοπιστεί πακέτο που ικανοποιεί τα κριτήρια των παραπάνω κανόνων, καταγράφεται alert σε ένα αρχείο τύπου IDS με όνομα alert στον κατάλογο log, που έχει ορίσει ο χρήστης. Στη συνέχεια φαίνεται ένα τέτοιο alert για το πείραμα που πραγματοποιήθηκε.

```
[**] [1:0:0] unencrypted password discover attempt [**]
[Priority: 1]
06/27-13:05:16.231448 0:1:6C:26:6D:14 -> 0:4:C1:6F:88:81 type:0x800 len:0x84
195.251.17.230:1613 -> 194.177.200.5:80 TCP TTL:128 TOS:0x0 ID:28183 IpLen:20 DgmLen:118 DF
***AP*** Seq: 0xA4E2A7C9  Ack: 0x2E622002  Win: 0xFDC5  TcpLen: 20
```

Στο παραπάνω alert φαίνεται η ακολουθία [1:0:0].

- Ο πρώτος αριθμός αντιστοιχεί στο Generator ID, το οποίο δηλώνει στο χρήστη πιο συστατικό του Snort δημιούργησε το alert. Ο αριθμός 1 αντιστοιχεί σε γενικό συναγερμό του snort (snort general alert). Η λίστα με τους GIDs υπάρχει στο αρχείο gen-msg.txt στο φάκελο etc του Snort source.
- Ο δεύτερος αριθμός αντιστοιχεί στο Snort ID ή αλλιώς Signature ID. Η λίστα με τους SIDs υπάρχει στο παραπάνω αρχείο, gen-msg.txt.
- Ο τρίτος αριθμός αντιστοιχεί στο revision ID. Χρησιμοποιείται κυρίως όταν γράφονται υπογραφές.

Επίσης φαίνεται το μήνυμα συναγερμού που εμφανίζεται "unencrypted password discover attempt", το οποίο έχει επιλέξει ο χρήστης να εμφανίζεται όταν το Snort ανιχνεύει πακέτο που μεταφέρει μη κρυπτογραφημένο κωδικό.

Επιπλέον, στο παραπάνω alert, φαίνονται τα χαρακτηριστικά του πακέτου, όπως η ακριβής ημερομηνία (27/06) και ώρα (13:05:16.231448) που στάλθηκε, η διεύθυνση πηγής (195.251.17.230 που αντιστοιχεί στον client), το port πηγής (1613), η διεύθυνση προορισμού (194.177.200.5 που αντιστοιχεί στον DNS server), το port προορισμού (80), ο τύπος του πακέτου (TCP), το Time to Live - TTL (128) και άλλα χαρακτηριστικά που το αφορούν.

Εκτός από το alert, δημιουργείται ένας κατάλογος που έχει ως όνομα τη IP source του πακέτου που περιέχει τον κωδικό. Στον συγκεκριμένο κατάλογο δημιουργείται αρχείο που περιέχει όλα τα χαρακτηριστικά που πακέτου αυτού, όπως φαίνεται παρακάτω:

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[**] unencrypted password discover attempt [**]
06/27-13:05:16.231448 0:1:6C:26:6D:14 -> 0:4:C1:6F:88:81 type:0x800 len:0x84
195.251.17.230:1613 -> 194.177.200.5:80 TCP TTL:128 TOS:0x0 ID:28183 IpLen:20 DgmLen:118 DF
***AP*** Seq: 0xA4E2A7C9  Ack: 0x2E622002  Win: 0xFDC5  TcpLen: 20
61 63 74 69 6F 6E 49 44 3D 36 26 6D 61 69 6C 62  actionID=6&mailb
6F 78 3D 49 4E 42 4F 58 26 69 6D 61 70 75 73 65  ox=INBOX&imapuse
72 3D 65 6E 61 72 69 26 70 61 73 73 3D 74 65 73  r=enari&pass=tes
74 26 49 4D 41 50 53 65 72 76 65 72 3D 76 6F 6C  t&IMAPServer=vol
6F 73 26 6E 65 77 5F 6C 61 6E 67 3D 65 6C        os&new_lang=el

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Στη συνέχεια ο διαχειριστής, αφού ενημερωθεί για τη συγκεκριμένη μεταφορά κωδικού, αποφασίζει ποιες θα είναι οι επόμενες ενέργειές του. Το προτιμότερο είναι να διακόψει τη σύνδεση και να ειδοποιήσει τον χρήστη ώστε να αλλάξει εγκαίρως τον κωδικό του, προτού ο κοινωνικός μηχανικός εκμεταλλευτεί την πληροφορία αυτή.

4.3.2 2ο Σενάριο

Ένα ακόμη σενάριο που υποδηλώνει επίθεση κοινωνικής μηχανικής είναι το ακόλουθο. Θεωρείται φυσιολογικό να παρατηρείται μεγάλος όγκος εισερχόμενων πληροφοριών σε κάποιο host, γιατί αυτό σημαίνει ότι απλώς κατεβάζει (download) δεδομένα. Όταν όμως παρατηρείται μεγάλος όγκος εξερχόμενων πληροφοριών από ένα υπολογιστή ή ένα δίκτυο τότε συνήθως κάτι ύποπτο λαμβάνει χώρα. Το πιο πιθανό σενάριο είναι ότι κάποιος κοινωνικός μηχανικός εγκατέστησε σε κάποιο host ένα πρόγραμμα, για παράδειγμα ένα Δούρειο Ίππο, και εξάγει από αυτόν τις πληροφορίες που τον ενδιαφέρουν, χωρίς όμως να γίνεται αντιληπτός από τον νόμιμο χρήστη του host.

Για να αποφευχθεί αυτό, δημιουργήθηκε το αρχείο outcoming.rules και τοποθετήθηκε το φάκελο rules του Snort. Το αρχείο περιέχει κανόνες οι οποίοι συλλαμβάνουν τα TCP πακέτα που έχουν προορισμό από τον υπολογιστή client στο εξωτερικό δίκτυο. Στη συνέχεια, σηματοδοτούν τον αντίστοιχο συναγερμό και τα πακέτα καταγράφονται στο αρχείο log. Κατόπιν μετρούνται ώστε να βρεθεί ο ακριβής τους αριθμός. Ο διαχειριστής συγκρίνει τον αριθμό των εξερχόμενων TCP πακέτων με ένα κατώφλι που έχει ορίσει ο ίδιος. Το κατώφλι αυτό προκύπτει με στατιστικά στοιχεία, μετά από μετρήσεις των εξερχόμενων πακέτων υπό φυσιολογικές συνθήκες σε περίοδο, για παράδειγμα, ενός μηνός. Αφού μετρηθεί το μέγεθος των εξερχόμενων πακέτων, συγκρίνεται με το κατώφλι και αν το ξεπερνά ο διαχειριστής αποφασίζει τις ενέργειες που θα εκτελέσει ώστε να προστατεύσει το δίκτυο για το οποίο είναι υπεύθυνος.

Ο βασικός κανόνας που χρησιμοποιείται είναι ακόλουθος:

```
alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg: "outcoming information"; dsize: > 1; priority:1;)
```

Σύμφωνα με τον παραπάνω κανόνα ενεργοποιείται συναγερμός κάθε φορά που μεταφέρεται ένα πακέτο από το εσωτερικό δίκτυο (η μεταβλητή HOME_NET αντιστοιχεί στην IP διεύθυνση του client) στο εξωτερικό (EXTERNAL_NET) από οποιοδήποτε port πηγής σε οποιοδήποτε port προορισμού. Το μήνυμα που εκτυπώνεται όταν καταγράφεται ο συναγερμός είναι "outcoming information". Η μεταβλητή dsize, στο rule option του παραπάνω κανόνα, δηλώνει ότι το Snort συλλαμβάνει όλα τα πακέτα που έχουν μέγεθος payload μεγαλύτερο από 1 byte. Περισσότερα για το ποιες αλλαγές και τροποποιήσεις απαιτούνται ώστε να πραγματοποιείται η ανίχνευση των εξερχόμενων TCP πακέτων δίνονται στο παράρτημα της συγκεκριμένης εργασίας.

Στο αρχείο snort.conf κάναμε include το αρχείο outcoming.rules ως εξής:
include $RULE_PATH/outcoming.rules

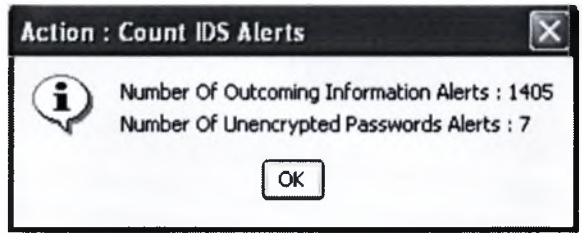
Όταν εντοπιστεί πακέτο που ικανοποιεί τα κριτήρια των κανόνων που αφορούν τα εξερχόμενα πακέτα, καταγράφεται alert σε ένα αρχείο τύπου IDS με όνομα alert στον κατάλογο log, που έχει ορίσει ο χρήστης. Στη συνέχεια φαίνεται ένα τέτοιο alert για το πείραμα που πραγματοποιήθηκε .

```
[**] [1:0:0] outcoming information [**]
[Priority: 1]
06/27-12:47:43.727645 0:1:6C:26:6D:14 -> 0:4:C1:6F:88:81 type:0x800 len:0x63
195.251.17.230:1336 -> 194.177.204.125:139 TCP TTL:128 TOS:0x0 ID:18463 IpLen:20 DgmLen:85 DF
***AP*** Seq: 0xEB51DBA6  Ack: 0x26316130  Win: 0xFC10  TcpLen: 20
```

Ο αριθμός τέτοιων alert είναι αρκετά μεγάλος γιατί και υπό φυσιολογικές συνθήκες τα εξερχόμενα TCP πακέτα που στέλνονται είναι πολλά. Όμως, το ερώτημα που προκύπτει είναι κάθε πόσο θα πρέπει το Snort να καταμετρά τα πακέτα που καταγράφονται και να τα ελέγχει με το κατώφλι. Αυτό θα οριστεί από τον ίδιο ανάλογα με τις ανάγκες του δικτύου για το οποίο είναι υπεύθυνος. Το διάστημα αυτό ποικίλει. Ο έλεγχος μπορεί να πραγματοποιείται κάθε μία ώρα για δίκτυα που περιέχουν σημαντικές πληροφορίες (όπως δίκτυα οργανισμών ή εταιρειών) αλλά μπορεί να πραγματοποιείται κάθε πέντε ώρες ή και περισσότερο για δίκτυα που περιέχουν λιγότερο σημαντικές πληροφορίες (δίκτυα σχολείων).

Για την καταμέτρηση του αριθμού των εξερχόμενων πακέτων στο πείραμα μας ακολουθήθηκε η παρακάτω διαδικασία. Υλοποιήθηκε πρόγραμμα σε γλώσσα

προγραμματισμού Java, με όνομα CntIDSAlert.java, το οποίο παίρνει ως είσοδο το αρχείο alert.IDS. Το αρχείο, αφού μετρήσει τους συναγερμούς, δίνει ως έξοδο τον αριθμό των alerts που αντιστοιχεί στην εξερχόμενη πληροφορία καθώς και τον αριθμό των alerts που αντιστοιχεί στα μη κρυπτογραφημένα passwords που στέλνονται από τον client (το σενάριο αυτό αναλύθηκε στην προηγουμένη υποενότητα). Η παρακάτω εικόνα δείχνει την έξοδο του προγράμματος, όπως τη βλέπει ο διαχειριστής, για ένα από τα πειράματα που πραγματοποιήθηκαν.



Να σημειωθεί ότι, για να ανιχνεύονται οι επιθέσεις κοινωνικής μηχανικής που μελετάμε, δηλαδή επιθέσεις που προσπαθούν να αντλήσουν πληροφορία από τα συστήματα, θα πρέπει να πραγματοποιείται συνεχής έλεγχος από το διαχειριστή του συστήματος και να ελέγχεται αν έχει ξεπεραστεί ο επιτρεπόμενος αριθμός εξερχόμενων πακέτων.

4.3.3 3^{ο} Σενάριο

Όπως έχει αναφερθεί τα προγράμματα backdoor και πιο συγκεκριμένα οι Δούρειοι Ίπποι μπορεί να αποτελούν προάγγελο των επιθέσεων κοινωνικής μηχανικής. Οι Δούρειοι Ίπποι είναι ιδιαίτερα επικίνδυνοι γιατί μπορούν να εκτελέσουν πολλές λειτουργίες, συμπεριλαμβανομένου την άντληση πληροφοριών που μελετήσαμε στο δεύτερο σενάριο ή ακόμα και την παρακολούθηση των πλήκτρων που πατά ο χρήστης με στόχο την υποκλοπή κωδικών. Για να αποφεύγουν τα γεγονότα αυτά τα συστήματα ανίχνευσης εισβολέων πρέπει να εντοπίζουν εγκαίρως τους Δούρειους Ίππους και να ειδοποιούν τους διαχειριστές σχετικά.

Το Snort χρησιμοποιεί υπογραφές Δούρειων Ίππων, όπως Subseven22, Dagger και Netbus και έχει γραμμένους κανόνες που ελέγχουν τα πακέτα και ανιχνεύουν προγράμματα backdoor. Το Snort στο φάκελο rules περιέχει το αρχείο backdoor.rules, το οποίο εξετάζει εισερχόμενα και εξερχόμενα tcp και udp πακέτα και αναζητά προγράμματα backdoor και Δούρειους Ίππους.

Στο αρχείο snort.conf κάναμε include το αρχείο backdoor.rules ως εξής:

include $RULE_PATH/ backdoor.rules

Παρακάτω αναφέρονται ενδεικτικά μερικοί κανόνες backdoor που περιέχονται στο αρχείο backdoor.rules.

alert tcp $EXTERNAL_NET 27374 -> $HOME_NET any (msg:"BACKDOOR subseven 22"; flow:to_server,established; content:"|0D 0A|[RPL]002|0D 0A|"; reference:arachnids,485; reference:url,www.hackfix.org/subseven/; classtype:misc-activity; sid:103; rev:7;)

alert tcp $HOME_NET 16959 -> $EXTERNAL_NET any (msg:"BACKDOOR subseven DEFCON8 2.1 access"; flow:from_server,established; content:"PWD"; classtype:trojan-activity; sid:107; rev:6;)

alert tcp $HOME_NET 12345:12346 -> $EXTERNAL_NET any (msg:"BACKDOOR netbus active"; flow:from_server,established; content:"NetBus"; reference:arachnids,401; classtype:misc-activity; sid:109; rev:5;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 12345:12346 (msg:"BACKDOOR netbus getinfo"; flow:to_server,established; content:"GetInfo|0D|"; reference:arachnids,403; classtype:misc-activity; sid:110; rev:4;)

alert tcp $HOME_NET 20034 -> $EXTERNAL_NET any (msg:"BACKDOOR NetBus Pro 2.0 connection established"; flowbits:isset,backdoor.netbus_2.connect; flow:from_server,established; content:"BN|10 00 02 00|"; depth:6; content:"|05 00|"; offset:8; depth:2; classtype:misc-activity; sid:115; rev:8;)

alert udp $HOME_NET 2140 -> $EXTERNAL_NET any (msg:"BACKDOOR DeepThroat 3.1 Server Response"; content:"Ahhhh My Mouth Is Open"; reference:arachnids,106; reference:mcafee,98574; reference:nessus,10053; classtype:misc-activity; sid:195; rev:7;)

alert tcp $HOME_NET 31785 -> $EXTERNAL_NET any (msg:"BACKDOOR HackAttack 1.20 Connect"; flow:established,from_server; content:"host"; classtype:misc-activity; sid:141; rev:5;)

Οι παραπάνω κανόνες είναι περισσότερο πολύπλοκοι από τους αυτούς που χρησιμοποιήθηκαν στα δύο προηγούμενα σενάρια, επειδή τα πακέτα που είναι μέρος backdoor προγραμμάτων και των Δούρειων Ίππων θα πρέπει να ικανοποιούν κριτήρια τα οποία βασίζονται σε υπογραφές γνωστών τέτοιων προγραμμάτων.

Επιπλέον, σε πολλούς κανόνες τους αρχείου backdoor.rules χρησιμοποιείται το keyword reference (όπως στον πρώτο από τους δοθέντες κανόνες) και έχει τη μορφή: reference:arachnids,485. Το keyword reference επιτρέπει στους κανόνες να περιέχουν αναφορές σε εξωτερικά συστήματα αναγνώρισης επιθέσεων. Στην ουσία το arachnids είναι μία από τις πολλές βάσεις δεδομένων που έχουν υπογραφές γνωστών backdoor προγραμμάτων.

Κάθε φορά που το Snort εντοπίζει ακολουθίες πακέτων που αποτελούν προγράμματα backdoor, ο διαχειριστής θα πρέπει να υποψιάζεται ότι υπάρχει ενδεχόμενο επίθεσης κοινωνικής μηχανικής και να αναζητά και άλλα γεγονότα που συνδέονται με αυτήν. Αν αυτό επιτευχθεί τότε εξαλείφεται ή μειώνεται σημαντικά ο κίνδυνος υποκλοπής πληροφοριών ή ζημιάς του συστήματος.

4.4 Συμπέρασμα

Το Snort, ως δικτυακό σύστημα ανίχνευσης εισβολέων, αποτυγχάνει με τον κλασικό τρόπο να προσδιορίσει μία επίθεση κοινωνικής μηχανικής. Ο λόγος είναι ότι μία τέτοια επίθεση αποτελείται από πολλές μικρότερες, το σύνολο των οποίων οδηγούν στην επίτευξη του στόχου του κοινωνικού μηχανικού. Το Snort μπορεί να εντοπίσει μεμονωμένα γεγονότα αλλά είναι δύσκολο να εντοπίσει το σύνολο όλων των μικρότερων επιθέσεων και να συμπεράνει την ολοκληρωμένη επίθεση κοινωνικής μηχανικής.

Επιπλέον, μία επίθεση κοινωνικής μηχανικής λαμβάνει χώρα σε πολλές φάσεις, δηλαδή σε διαφορετικές χρονικές στιγμές, χρησιμοποιώντας διαφορετικό IP source και ίσως διαφορετικό IP destination. Συνεπώς, ένα σύστημα IDS πρέπει να εφαρμόζει ευριστικούς αλγορίθμους που να ταιριάζει τα διαφορετικά κατά τα άλλα πακέτα για να συμπεραίνει ότι λαμβάνει χώρα μία επίθεση κοινωνικής μηχανικής.

Το Snort, όπως και τα περισσότερα δικτυακά συστήματα ανίχνευσης εισβολέων, θα πρέπει να τροποποιηθούν έτσι ώστε να είναι ικανά να αντιμετωπίσουν μία επίθεση κοινωνικής μηχανικής. Τα σενάρια που έχουν αναφερθεί στη συγκεκριμένη ενότητα αποτελούν ένα μέρος των τροποποιήσεων που μπορούν να γίνουν στο Snort προκειμένου να επιτευχθεί η ανίχνευση των μικρότερων επιθέσεων που συνθέτουν μία πλήρη επίθεση κοινωνικής μηχανικής.

Περισσότερα για τον τρόπο που πρέπει σχεδιαστεί και να υλοποιηθεί το Snort ώστε να ανιχνεύει επιθέσεις κοινωνικής μηχανικής θα αναφερθούν στην έκτη ενότητα «Μελλοντικές Επεκτάσεις», όπου αναφέρεται τι είδους μοντέλο θα πρέπει να σχεδιαστεί και να υλοποιηθεί για το σκοπό αυτό.

Ενότητα 5

ΣΥΜΠΕΡΑΣΜΑΤΑ

Παράλληλα με τη συνεχή ανάπτυξη που παρουσιάζει το Internet, παρατηρείται και αύξηση στον αριθμό των επιθέσεων που πραγματοποιούνται στα υπολογιστικά συστήματα. Τα συστήματα ανίχνευσης εισβολέων αποτελούν απαραίτητο εργαλείο για την προστασία των δικτύων και σε συνεργασία με τα υπόλοιπα εργαλεία ασφάλειας επιτρέπουν την πλήρη παρακολούθηση των δραστηριοτήτων που λαμβάνουν χώρα σε ένα δίκτυο.

Τα συστήματα ανίχνευσης εισβολέων είναι ικανά να προστατεύσουν τα συστήματα από μία πληθώρα επιθέσεων καθώς και από τις επιθέσεις που οργανώνονται από εξουσιοδοτημένους χρήστες των δικτύων. Για να επιτυγχάνεται η ανίχνευση μεγαλύτερου αριθμού επιθέσεων τα IDS πρέπει να χρησιμοποιούν τη μέθοδο αναγνώρισης υπογράφων σε συνδυασμό με την ανίχνευση κακής συμπεριφοράς. Με τον τρόπο αυτόν εντοπίζονται όχι μόνο οι γνωστές αλλά και οι άγνωστες επιθέσεις.

Όσον αφορά τις επιθέσεις κοινωνικής μηχανικής, όπως αναφέρθηκε και στις προηγούμενες ενότητες, αποτελούν μία νέα απειλή για τις πληροφορίες και τα συστήματα υπολογιστών. Η κοινωνική μηχανική βασίζεται στην αδυναμία της ανθρώπινης φύσης, εκμεταλλεύεται την εμπιστοσύνη που δείχνουν οι χρήστες και επιτυγχάνει με δόλιους τρόπους την παράνομη απόκτηση δεδομένων μέσω της εξαπάτησης. Συνήθως το θύμα είτε πείθεται να αποκαλύψει ευαίσθητες πληροφορίες, όπως κωδικούς, είτε προβαίνει σε ενέργειες που ανοίγουν την πόρτα στον εισβολέα. Στη συγκεκριμένη μελέτη ασχοληθήκαμε με την «ηλεκτρονική» κοινωνική μηχανική, όπου δηλαδή χρησιμοποιείται η τεχνολογία για την πραγματοποίηση των επιθέσεων και ο χρήστης, χωρίς να το γνωρίζει, εκτελεί ενέργειες οι οποίες διευκολύνουν το έργο του κακόβουλου χρήστη.

Η ανίχνευση τους είναι δύσκολη επειδή κάθε φορά παρουσιάζονται με διαφορετική μορφή ανάλογα με το σχέδιο που έχει καταστρώσει ο κοινωνικός μηχανικός. Όσον αφορά τις συνέπειες τους μπορεί να είναι από μηδαμινές, όπως απλή παρακολούθηση της κίνησης ενός host, έως και καταστροφικές για έναν οργανισμό ή ένα σύστημα, όπως υποκλοπή σημαντικών πληροφοριών και διαγραφή αρχείων του συστήματος. Η πιο αποτελεσματική μέθοδος ενάντια σε αυτές τις επιθέσεις είναι κυρίως η ενημέρωση των χρηστών για το φαινόμενο αυτό, οι απαραίτητες ρυθμίσεις στους μηχανισμούς ασφαλείας των δικτύων και η εφαρμογή των πολιτικών ασφαλείας που υπάρχουν σε κάθε σύστημα.

Ωστόσο, τα συστήματα ανίχνευσης εισβολέων, όπως και όλοι οι υπόλοιποι μηχανισμοί που είναι υπεύθυνοι για την ασφάλεια των δικτύων, δεν έχουν σχεδιαστεί κατάλληλα ώστε να ανιχνεύουν τέτοιες επιθέσεις. Αυτό αποδείχθηκε και μέσα από τα πειράματα που έγιναν χρησιμοποιώντας το εργαλείο Snort. Το Snort όπως έχει σχεδιαστεί, είναι ικανό να ανιχνεύει μία πληθώρα επιθέσεων, όμως δεν μπορεί να ανιχνεύει επιθέσεις κοινωνικής μηχανικής.

Μετά τη χρήση του εργαλείου Snort, έχουν προκύψει τα ακόλουθα συμπεράσματα:

- Το Snort είναι ένα lightweight δικτυακό σύστημα ανίχνευσης εισβολέων, εύκολο στη χρήση
- Αντιμετωπίζει πληθώρα επιθέσεων που απειλούν τα δίκτυα και τα πληροφοριακά συστήματα
- Είναι ευέλικτο, αφού δίνεται η δυνατότητα στο χρήστη να συντάξει νέους κανόνες οι οποίοι εκτελούν τις ενέργειες που επιθυμεί όταν ανιχνεύσει συγκεκριμένη δραστηριότητα στο δίκτυο
- Για να αντιμετωπίσει επιθέσεις κοινωνικής μηχανικής θα πρέπει να τροποποιηθεί, όπως τα σενάρια που αναφέρθηκαν σε προηγούμενη ενότητα

Ο εντοπισμός επιθέσεων κοινωνικής μηχανικής αποτελεί μία δύσκολη εργασία επειδή υπάρχει μία ιδιαιτερότητα σε αυτές. Οι επιθέσεις κοινωνικής μηχανικής μπορεί να αποτελούνται από πολλές μικρότερες, το σύνολο των οποίων θα οδηγήσει τελικά στην επίτευξη του στόχου που έχει θέσει ο κακόβουλος χρήστης. Επιπλέον, μία επίθεση κοινωνικής μηχανικής λαμβάνει χώρα σε πολλές φάσεις, δηλαδή σε διαφορετικές χρονικές στιγμές, χρησιμοποιώντας διαφορετικό IP source και ίσως

διαφορετικό IP destination. Συνεπώς, ένα σύστημα IDS πρέπει να εφαρμόζει ευριστικούς πλέον αλγορίθμους που να ταιριάζει τα διαφορετικά κατά τα άλλα πακέτα για να συμπεραίνει ότι λαμβάνει χώρα μία επίθεση κοινωνικής μηχανικής. Αυτό το θέμα αναλύεται διεξοδικότερα στην επόμενη ενότητα «Μελλοντικές Επεκτάσεις», όπου αναφέρεται τι είδους μοντέλο θα πρέπει να σχεδιαστεί και να υλοποιηθεί ώστε να ανιχνεύονται οι πολυάριθμες επιθέσεις που υποδηλώνουν μία ολοκληρωμένη επίθεση κοινωνικής μηχανικής.

Ένα γενικότερο συμπέρασμα είναι ότι ο τομέας της ασφάλειας απέκτησε πλέον, τόσο από τους χρήστες όσο και από τους μηχανικούς, το ενδιαφέρον που του αντιστοιχεί. Τα συστήματα ανίχνευσης εισβολέων παρουσιάζουν δυνατότητες βελτίωσης και εξέλιξης και στόχος των υπευθύνων, στο μέλλον, είναι να αντικαταστήσουν τους υπόλοιπους μηχανισμούς ασφαλείας, παρέχοντας πλήρη ασφάλεια στον υπολογιστή ή στο δίκτυο που προστατεύουν.

Ενότητα 6

ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

Το Snort δεν έχει σχεδιαστεί να εντοπίζει επιθέσεις κοινωνικής μηχανικής. Είναι δύσκολη η τροποποίηση του Snort ώστε να ανιχνεύει επιτυχώς τέτοιες επιθέσεις για τους ακόλουθους λόγους:

- Οι επιθέσεις αυτές παρουσιάζονται κάθε φορά με διαφορετική μορφή, ανάλογα με το σχέδιο που έχει καταστρώσει ο κοινωνικός μηχανικός.
- Το Snort αναγνωρίζει τις επιθέσεις χρησιμοποιώντας τη μέθοδο αναγνώρισης υπογράφων. Η ανίχνευση των επιθέσεων κοινωνικής μηχανικής δεν μπορεί να βασιστεί στη χρήση της μεθόδου αυτής, εξαιτίας του γεγονότος ότι κάθε φορά πραγματοποιούνται με διαφορετική μορφή.
- Οι επιθέσεις κοινωνικής μηχανικής αποτελούνται από ένα σύνολο από μικρότερες επιθέσεις, τα αποτελέσματα των οποίων οδηγούν στην επίτευξη του στόχου που έχει θέσει ο κακόβουλος χρήστης.
- Μία επίθεση κοινωνικής μηχανικής λαμβάνει χώρα σε πολλές φάσεις, δηλαδή σε διαφορετικές χρονικές στιγμές, χρησιμοποιώντας διαφορετικό IP source και ίσως διαφορετικό IP destination.

Χρησιμοποιώντας το εργαλείο Snort, στόχος των υπευθύνων ασφαλείας είναι να ανιχνεύονται, με μεγάλα ποσοστά επιτυχίας, οι επιθέσεις κοινωνικής μηχανικής. Στη συνέχεια αναλύεται ένα μοντέλο που μπορεί να υλοποιηθεί για το σκοπό αυτό. Το μοντέλο βασίζεται στο γεγονός ότι μία επίθεση κοινωνικής μηχανικής αποτελείται από πολλές μικρότερες.

Μία επίθεση κοινωνικής μηχανικής μπορεί να έχει την ακόλουθη μορφή.

1. Αρχικά ο κοινωνικός μηχανικός, αφού εντοπίσει το σύστημα-στόχο στέλνει στο χρήστη του ένα e-mail με το οποίο τον ενημερώνει ότι στον τάδε δικτυακό τόπο υπάρχει λογισμικό αναβάθμισης για κάποιο πρόγραμμα που χρησιμοποιεί.
2. Ο χρήστης επισκέπτεται τον δικτυακό τόπο, κατεβάζει το λογισμικό αναβάθμισης όμως παράλληλα κατεβάζει ένα κατασκοπευτικό λογισμικό, για παράδειγμα ένα Δούρειο Ίππο, που παρακολουθεί όλες τις κινήσεις του χρήστη.
3. Ο Δούρειος Ίππος από τη στιγμή που εγκατασταθεί στο μηχάνημα μπορεί να υποκλέψει σημαντικές πληροφορίες, όπως το όνομα χρήστη και τον κωδικό του και να ενημερώσει τον κακόβουλο χρήστη σχετικά. Η υποκλοπή του κωδικού μπορεί να προκαλέσει μεγάλη ζημιά στο χρήστη, γιατί όπως παρατηρείται οι χρήστες συνήθως χρησιμοποιούν τον ίδιο κωδικό σε πολλές δραστηριότητες και συναλλαγές που πραγματοποιούν. Όταν, λοιπόν, μάθει ο κοινωνικός μηχανικός, εν αγνοία του χρήστη, τον κωδικό του τότε αυτομάτως αποκτά πρόσβαση στις δραστηριότητες του χρήστη.
4. Επιπλέον, ο Δούρειος Ίππος μπορεί να κλέψει πληροφορίες από το σύστημα ή τα αρχεία του χρήστη και να αποστείλει ένα σημαντικό όγκο εξερχόμενων πληροφοριών στον κοινωνικό μηχανικό.
5. Θεωρούμε ότι με τα παραπάνω επετεύχθη ο στόχος του κοινωνικού μηχανικού και ολοκληρώθηκε με επιτυχία η επίθεση του.

Στο παραπάνω σενάριο παρατηρείται ένα αριθμός από μικρότερες επιθέσεις που η καθεμία από αυτές εκτελεί συγκεκριμένη λειτουργία. Θα πρέπει να γραφούν κανόνες ώστε το Snort να αναγνωρίζει ξεχωριστά καθεμία από αυτές τις επιθέσεις και να εμποδίζει την πραγματοποίησή τους, προτού λάβουν χώρα οι συνέπειες που μπορούν αυτές να προκαλέσουν αν συνδυαστούν κατάλληλα.

Το πρώτο βήμα, για παράδειγμα, είναι η αναγνώριση κατασκοπευτικού λογισμικού. Αν και υπάρχουν υπογραφές που αναγνωρίζουν μερικούς γνωστούς Δούρειους Ίππους, γενικότερα η αναγνώριση κατασκοπευτικού λογισμικού αποτελεί μία δύσκολη εργασία. Σε περίπτωση που έχει εντοπιστεί ο Δούρειος Ίππος του παραπάνω σεναρίου τότε η επίθεση αποτυγχάνει και έχει προστατευτεί επιτυχώς το σύστημα και οι πληροφορίες του χρήστη.

Σε αντίθετη περίπτωση, αν αποτύχει το Snort να αναγνωρίσει το Δούρειο Ίππο, θα πρέπει να ενημερώσει το χρήστη και να απαγορεύσει τη μεταφορά του κωδικού του με προορισμό την IP του κοινωνικού μηχανικού. Επιπλέον, μετρώντας τον αριθμό των εξερχόμενων πακέτων και συγκρίνοντας τον με το προκαθορισμένο κατώφλι, θα πρέπει να εμποδίσει τη μεταφορά αδικαιολόγητα μεγάλου όγκου εξερχόμενων πληροφοριών.

Επιπλέον, κάθε φορά που το Snort ανιχνεύει μία επίθεση η οποία μπορεί να αποτελεί μέρος μίας επίθεσης κοινωνικής μηχανικής, θα πρέπει να καταγράφει τα χαρακτηριστικά της. Αυτές οι καταγραφές θα χρησιμοποιούνται σαν υπογραφές επιθέσεων. Έτσι, κάθε φορά που λαμβάνει χώρα μία επίθεση θα ανατρέχει σε ιστορικά στοιχεία επιθέσεων που χαρακτηρίζονται ως επιμέρους επίθεση κοινωνικής μηχανικής. Με τον τρόπο αυτό θα εντοπίσει εγκαίρως επιθέσεις κοινωνικής μηχανικής που έχουν γίνει στο παρελθόν και τελικώς θα συμβάλει στην προστασία του συστήματος.

Συνεπώς, στόχος του μοντέλου είναι να αναγνωρίζει τις επιμέρους επιθέσεις και να τις εμποδίζει ώστε να αποτρέπει τις συνέπειες που μπορούν να προκύψουν από μία επίθεση κοινωνικής μηχανικής. Ακόμη και αν δεν καταφέρει να εντοπίσει μία επίθεση, το ενθαρρυντικό είναι ότι θα σπάσει αυτή την αλυσίδα επιθέσεων και θα προστατευτούν οι ευαίσθητες πληροφορίες των χρηστών.

ΠΑΡΑΡΤΗΜΑ

Στο συγκεκριμένο παράρτημα περιλαμβάνεται το αρχείο snort.conf, που περιέχεται στο φάκελο etc του Snort. Το αρχείο snort.conf είναι το configuration file και αποτελεί τον εγκέφαλο του Snort. Στο αρχείο αυτό ο χρήστης ορίζει τι πρέπει να κάνει το Snort ως δικτυακό σύστημα ανίχνευσης εισβολέων. Παρακάτω δίνονται όλες οι απαραίτητες ρυθμίσεις και τροποποιήσεις που έχουν γίνει ώστε το Snort να ελέγχει τα εξερχόμενα πακέτα που φεύγουν από τον υπολογιστή client και εξετάζει αν τα πακέτα αυτά ικανοποιούν τους κανόνες που έχουν γραφτεί στα αρχεία password.rules, outcoming.rules και backdoors.rules.

```
#--------------------------------------------------
#   http://www.snort.org     Snort 2.3.3 Ruleset
#     Contact: snort-sigs@lists.sourceforge.net
#--------------------------------------------------
# $Id: snort.conf,v 1.144.2.11 2005/04/22 19:15:49 jhewlett Exp $
#
##################################################
# This file contains a sample snort configuration.
# You can take the following steps to create your own custom
configuration:
#
#  1) Set the network variables for your network
#  2) Configure preprocessors
#  3) Configure output plugins
#  4) Customize your rule set
#
##################################################
# Step #1: Set the network variables:
#
# You must change the following variables to reflect your local
network.

var HOME_NET 195.251.17.230

# Set up the external network addresses as well.  A good start may be
"any"
var EXTERNAL_NET any

# Configure your server lists.  This allows snort to only look for
attacks to
# systems that have a service up.  Why look for HTTP attacks if you
are not
```

```
# running a web server?  This allows quick filtering based on IP
addresses
# These configurations MUST follow the same configuration scheme as
defined
# above for $HOME_NET.

# List of DNS servers on your network
var DNS_SERVERS $HOME_NET

# List of SMTP servers on your network
var SMTP_SERVERS $HOME_NET

# List of web servers on your network
var HTTP_SERVERS $HOME_NET

# List of sql servers on your network
var SQL_SERVERS $HOME_NET

# List of telnet servers on your network
var TELNET_SERVERS $HOME_NET

# List of snmp servers on your network
var SNMP_SERVERS $HOME_NET

# Configure your service ports.  This allows snort to look for
attacks destined
# to a specific application only on the ports that application runs
on.  For
# example, if you run a web server on port 8081, set your HTTP_PORTS
variable
# like this:
#
# var HTTP_PORTS 8081
#
# Port lists must either be continuous [eg 80:8080], or a single port
[eg 80].
# We will adding support for a real list of ports in the future.

# Ports you run web servers on
#
# Please note:  [80,8080] does not work.
# If you wish to define multiple HTTP ports,
#
## var HTTP_PORTS 80
## include somefile.rules
## var HTTP_PORTS 8080
## include somefile.rules
var HTTP_PORTS 80

# Ports you want to look for SHELLCODE on.
var SHELLCODE_PORTS !80

# Ports you do oracle attacks on
var ORACLE_PORTS 1521

# other variables
#
# AIM servers.  AOL has a habit of adding new AIM servers, so instead
of
# modifying the signatures when they do, we add them to this list of
servers.
```

```
var AIM_SERVERS
[64.12.24.0/23,64.12.28.0/23,64.12.161.0/24,64.12.163.0/24,64.12.200.
0/24,205.188.3.0/24,205.188.5.0/24,205.188.7.0/24,205.188.9.0/24,205.
188.153.0/24,205.188.179.0/24,205.188.248.0/24]

# Path to your rules files (this can be a relative path)
# Note for Windows users:  You are advised to make this an absolute
path,
# such as:  c:\snort\rules
var RULE_PATH ../rules

# Configure the snort decoder
# ============================
#
# Snort's decoder will alert on lots of things such as header
# truncation or options of unusual length or infrequently used tcp
options
#
#
# Stop generic decode events:
#
# config disable_decode_alerts
#
# Stop Alerts on experimental TCP options
#
# config disable_tcpopt_experimental_alerts
#
# Stop Alerts on obsolete TCP options
#
# config disable_tcpopt_obsolete_alerts
#
# Stop Alerts on T/TCP alerts
#
# In snort 2.0.1 and above, this only alerts when a TCP option is
detected
# that shows T/TCP being actively used on the network.  If this is
normal
# behavior for your network, disable the next option.
#
# config disable_tcpopt_ttcp_alerts
#
# Stop Alerts on all other TCPOption type events:
#
# config disable_tcpopt_alerts
#
# Stop Alerts on invalid ip options
#
# config disable_ipopt_alerts

# Configure the detection engine
# ===============================
#
# Use a different pattern matcher in case you have a machine with
very limited
# resources:
#
# config detection: search-method lowmem

# Configure Inline Resets
# ========================
#
```

```
# If running an iptables firewall with snort in InlineMode() we can
now
# perform resets via a physical device. We grab the indev from
iptables
# and use this for the interface on which to send resets. This config
# option takes an argument for the src mac address you want to use in
the
# reset packet.  This way the bridge can remain stealthy. If the src
mac
# option is not set we use the mac address of the indev device. If we
# don't set this option we will default to sending resets via raw
socket,
# which needs an ipaddress to be assigned to the int.
#
# config layer2resets: 00:06:76:DD:5F:E3

###################################################
# Step #2: Configure preprocessors
#
# General configuration for preprocessors is of
# the form
# preprocessor <name_of_processor>: <configuration_options>

# Configure Flow tracking module
# ------------------------------
#
# The Flow tracking module is meant to start unifying the state
keeping
# mechanisms of snort into a single place. Right now, only a portscan
detector
# is implemented but in the long term,  many of the stateful
subsystems of
# snort will be migrated over to becoming flow plugins. This must be
enabled
# for flow-portscan to work correctly.
#
# See README.flow for additional information
#
preprocessor flow: stats_interval 0 hash 2

# frag2: IP defragmentation support
# ------------------------------
# This preprocessor performs IP defragmentation.  This plugin will
also detect
# people launching fragmentation attacks (usually DoS) against hosts.
No
# arguments loads the default configuration of the preprocessor,
which is a 60
# second timeout and a 4MB fragment buffer.

# The following (comma delimited) options are available for frag2
#    timeout [seconds] - sets the number of [seconds] that an
unfinished
#                        fragment will be kept around waiting for
completion,
#                        if this time expires the fragment will be
flushed
#    memcap [bytes] - limit frag2 memory usage to [number] bytes
#                      (default:  4194304)
#
#    min_ttl [number] - minimum ttl to accept
```

```
#
#    ttl_limit [number] - difference of ttl to accept without
alerting
#                         will cause false positves with router flap
#
# Frag2 uses Generator ID 113 and uses the following SIDS
# for that GID:
#  SID     Event description
# -----   -------------------
#   1       Oversized fragment (reassembled frag > 64k bytes)
#   2       Teardrop-type attack

preprocessor frag2

# stream4: stateful inspection/stream reassembly for Snort
#---------------------------------------------------------------------
--
# Use in concert with the -z [all|est] command line switch to defeat
stick/snot
# against TCP rules.  Also performs full TCP stream reassembly,
stateful
# inspection of TCP streams, etc.  Can statefully detect various
portscan
# types, fingerprinting, ECN, etc.

# stateful inspection directive
# no arguments loads the defaults (timeout 30, memcap 8388608)
# options (options are comma delimited):
#   detect_scans - stream4 will detect stealth portscans and generate
alerts
#                  when it sees them when this option is set
#   detect_state_problems - detect TCP state problems, this tends to
be very
#                           noisy because there are a lot of crappy
ip stack
#                           implementations out there
#
#   disable_evasion_alerts - turn off the possibly noisy mitigation
of
#                            overlapping sequences.
#
#
#   min_ttl [number]       - set a minium ttl that snort will accept
to
#                            stream reassembly
#
#   ttl_limit [number]     - differential of the initial ttl on a
session versus
#                             the normal that someone may be playing
games.
#                             Routing flap may cause lots of false
positives.
#
#   keepstats [machine|binary] - keep session statistics, add
"machine" to
#                         get them in a flat format for machine
reading, add
#                         "binary" to get them in a unified binary
output
#                         format
#   noinspect - turn off stateful inspection only
```

```
#   timeout [number] - set the session timeout counter to [number]
seconds,
#                      default is 30 seconds
#   memcap [number] - limit stream4 memory usage to [number] bytes
#   log_flushed_streams - if an event is detected on a stream this
option will
#                         cause all packets that are stored in the
stream4
#                         packet buffers to be flushed to disk.  This
only
#                         works when logging in pcap mode!
#
# Stream4 uses Generator ID 111 and uses the following SIDS
# for that GID:
#  SID     Event description
# -----   -------------------
#   1       Stealth activity
#   2       Evasive RST packet
#   3       Evasive TCP packet retransmission
#   4       TCP Window violation
#   5       Data on SYN packet
#   6       Stealth scan: full XMAS
#   7       Stealth scan: SYN-ACK-PSH-URG
#   8       Stealth scan: FIN scan
#   9       Stealth scan: NULL scan
#   10      Stealth scan: NMAP XMAS scan
#   11      Stealth scan: Vecna scan
#   12      Stealth scan: NMAP fingerprint scan stateful detect
#   13      Stealth scan: SYN-FIN scan
#   14      TCP forward overlap

preprocessor stream4: disable_evasion_alerts

# tcp stream reassembly directive
# no arguments loads the default configuration
#   Only reassemble the client,
#   Only reassemble the default list of ports (See below),
#   Give alerts for "bad" streams
#
# Available options (comma delimited):
#   clientonly - reassemble traffic for the client side of a
connection only
#   serveronly - reassemble traffic for the server side of a
connection only
#   both - reassemble both sides of a session
#   noalerts - turn off alerts from the stream reassembly stage of
stream4
#   ports [list] - use the space separated list of ports in [list],
"all"
#                  will turn on reassembly for all ports, "default"
will turn
#                  on reassembly for ports 21, 23, 25, 53, 80, 143,
110, 111
#                  and 513

preprocessor stream4_reassemble

# http_inspect: normalize and detect HTTP traffic and protocol
anomalies
#
# lots of options available here. See doc/README.http_inspect.
```

```
# unicode.map should be wherever your snort.conf lives, or given
# a full path to where snort can find it.
preprocessor http_inspect: global \
    iis_unicode_map unicode.map 1252

preprocessor http_inspect_server: server default \
    profile all ports { 80 8080 8180 } oversize_dir_length 500

#
#  Example unique server configuration
#
#preprocessor http_inspect_server: server 1.1.1.1 \
#    ports { 80 3128 8080 } \
#    flow_depth 0 \
#    ascii no \
#    double_decode yes \
#    non_rfc_char { 0x00 } \
#    chunk_length 500000 \
#    non_strict \
#    oversize_dir_length 300 \
#    no_alerts


# rpc_decode: normalize RPC traffic
# ---------------------------------
# RPC may be sent in alternate encodings besides the usual 4-byte
encoding
# that is used by default. This plugin takes the port numbers that
RPC
# services are running on as arguments - it is assumed that the given
ports
# are actually running this type of service. If not, change the ports
or turn
# it off.
# The RPC decode preprocessor uses generator ID 106
#
# arguments: space separated list
# alert_fragments - alert on any rpc fragmented TCP data
# no_alert_multiple_requests - don't alert when >1 rpc query is in a
packet
# no_alert_large_fragments - don't alert when the fragmented
#                            sizes exceed the current packet size
# no_alert_incomplete - don't alert when a single segment
#                       exceeds the current packet size

preprocessor rpc_decode: 111 32771

# bo: Back Orifice detector
# -------------------------
# Detects Back Orifice traffic on the network.  Takes no arguments in
2.0.
#
# The Back Orifice detector uses Generator ID 105 and uses the
# following SIDS for that GID:
#  SID     Event description
# -----   ------------------
#   1       Back Orifice traffic detected

preprocessor bo

# telnet_decode: Telnet negotiation string normalizer
```

```
# ----------------------------------------------------
# This preprocessor "normalizes" telnet negotiation strings from
telnet and ftp
# traffic.  It works in much the same way as the http_decode
preprocessor,
# searching for traffic that breaks up the normal data stream of a
protocol and
# replacing it with a normalized representation of that traffic so
that the
# "content" pattern matching keyword can work without requiring
modifications.
# This preprocessor requires no arguments.
# Portscan uses Generator ID 109 and does not generate any SID
currently.

preprocessor telnet_decode

# Flow-Portscan: detect a variety of portscans
# ---------------------------------------
# Note:  The Flow preprocessor (above) must first be enabled for
Flow-Portscan to
# work.
#
# This module detects portscans based off of flow creation in the
flow
# preprocessors.  The goal is to catch one->many hosts and one->many
# ports scans.
#
# Flow-Portscan has numerous options available, please read
# README.flow-portscan for help configuring this option.

# Flow-Portscan uses Generator ID 121 and uses the following SIDS for
that GID:
#  SID     Event description
# -----   -------------------
#   1       flow-portscan: Fixed Scale Scanner Limit Exceeded
#   2       flow-portscan: Sliding Scale Scanner Limit Exceeded
#   3       flow-portscan: Fixed Scale Talker Limit Exceeded
#   4     flow-portscan: Sliding Scale Talker Limit Exceeded

# preprocessor flow-portscan: \
#     talker-sliding-scale-factor 0.50 \
#     talker-fixed-threshold 30 \
#     talker-sliding-threshold 30 \
#     talker-sliding-window 20 \
#     talker-fixed-window 30 \
#     scoreboard-rows-talker 30000 \
#     server-watchnet [10.2.0.0/30] \
#     server-ignore-limit 200 \
#     server-rows 65535 \
#     server-learning-time 14400 \
#     server-scanner-limit 4 \
#     scanner-sliding-window 20 \
#     scanner-sliding-scale-factor 0.50 \
#     scanner-fixed-threshold 15 \
#     scanner-sliding-threshold 40 \
#     scanner-fixed-window 15 \
#     scoreboard-rows-scanner 30000 \
#     src-ignore-net [192.168.1.1/32,192.168.0.0/24] \
#     dst-ignore-net [10.0.0.0/30] \
#     alert-mode once \
```

```
#     output-mode msg \
#     tcp-penalties on

# sfPortscan
# ----------
# Author: Dan Roelker
# Portscan detection module.  Detects various types of portscans and
# portsweeps.  For more information on detection philosophy, alert
types,
# and detailed portscan information, please refer to the
README.sfportscan.
#
# -configuration options-
#     proto { tcp udp icmp ip_proto all }
#       The arguments to the proto option are the types of protocol
scans that
#       the user wants to detect.  Arguments should be separated by
spaces and
#       not commas.
#     scan_type { portscan portsweep decoy_portscan
distributed_portscan all }
#       The arguments to the scan_type option are the scan types that
the
#       user wants to detect.  Arguments should be separated by
spaces and not
#       commas.
#     sense_level { low|medium|high }
#       There is only one argument to this option and it is the level
of
#       sensitivity in which to detect portscans.  The 'low'
sensitivity
#       detects scans by the common method of looking for response
errors, such
#       as TCP RSTs or ICMP unreachables.  This level requires the
least
#       tuning.  The 'medium' sensitivity level detects portscans and
#       filtered portscans (portscans that receive no response).
This
#       sensitivity level usually requires tuning out scan events
from NATed
#       IPs, DNS cache servers, etc.  The 'high' sensitivity level
has
#       lower thresholds for portscan detection and a longer time
window than
#       the 'medium' sensitivity level.  Requires more tuning and may
be noisy
#       on very active networks.  However, this sensitivity levels
catches the
#       most scans.
#     memcap { positive integer }
#       The maximum number of bytes to allocate for portscan
detection.  The
#       higher this number the more nodes that can be tracked.
#     logfile { filename }
#       This option specifies the file to log portscan and detailed
portscan
#       values to.  If there is not a leading /, then snort logs to
the
#       configured log directory.  Refer to README.sfportscan for
details on
#       the logged values in the logfile.
```

```
#     watch_ip { Snort IP List }
#     ignore_scanners { Snort IP List }
#     ignore_scanned { Snort IP List }
#       These options take a snort IP list as the argument.  The
'watch_ip'
#       option specifies the IP(s) to watch for portscan.  The
#       'ignore_scanners' option specifies the IP(s) to ignore as
scanners.
#       Note that these hosts are still watched as scanned hosts.
The
#       'ignore_scanners' option is used to tune alerts from very
active
#       hosts such as NAT, nessus hosts, etc.  The 'ignore_scanned'
option
#       specifies the IP(s) to ignore as scanned hosts.  Note that
these hosts
#       are still watched as scanner hosts.  The 'ignore_scanned'
option is
#       used to tune alerts from very active hosts such as syslog
servers, etc.
#
preprocessor sfportscan: proto  { all } \
                         memcap { 10000000 } \
                         sense_level { low }

# arpspoof
#----------------------------------------
# Experimental ARP detection code from Jeff Nathan, detects ARP
attacks,
# unicast ARP requests, and specific ARP mapping monitoring.  To make
use of
# this preprocessor you must specify the IP and hardware address of
hosts on
# the same layer 2 segment as you.  Specify one host IP MAC combo per
line.
# Also takes a "-unicast" option to turn on unicast ARP request
detection.
# Arpspoof uses Generator ID 112 and uses the following SIDS for that
GID:

#  SID     Event description
# -----   -------------------
#   1       Unicast ARP request
#   2       Etherframe ARP mismatch (src)
#   3       Etherframe ARP mismatch (dst)
#   4       ARP cache overwrite attack

#preprocessor arpspoof
#preprocessor arpspoof_detect_host: 192.168.40.1 f0:0f:00:f0:0f:00

# Performance Statistics
# ----------------------
# Documentation for this is provided in the Snort Manual.  You should
read it.
# It is included in the release distribution as doc/snort_manual.pdf
#
# preprocessor perfmonitor: time 300 file /var/snort/snort.stats
pktcnt 10000

# X-Link2State mini-preprocessor
# ------------------------------
```

```
# This preprocessor will catch the X-Link2State vulnerability
# (www.microsoft.com/technet/security/bulletin/MS05-021.mspx).
#
# Format:
# preprocessor xlink2state: ports { <port> [<port> <...>] } [drop]
#
# "drop" will drop the attack if in Inline-mode.

#  SID         Event description
# -----   -------------------
#   1      X-Link2State length greater than 1024

preprocessor xlink2state: ports { 25 691 }

###################################################################
# Step #3: Configure output plugins
#
# Uncomment and configure the output plugins you decide to use.
General
# configuration for output plugins is of the form:
#
# output <name_of_plugin>: <configuration_options>
#
# alert_syslog: log alerts to syslog
# ----------------------------------
# Use one or more syslog facilities as arguments.  Win32 can also
optionally
# specify a particular hostname/port.  Under Win32, the default
hostname is
# '127.0.0.1', and the default port is 514.
#
# [Unix flavours should use this format...]
# output alert_syslog: LOG_AUTH LOG_ALERT
#
# [Win32 can use any of these formats...]
# output alert_syslog: LOG_AUTH LOG_ALERT
# output alert_syslog: host=hostname, LOG_AUTH LOG_ALERT
# output alert_syslog: host=hostname:port, LOG_AUTH LOG_ALERT

# log_tcpdump: log packets in binary tcpdump format
# -------------------------------------------------
# The only argument is the output file name.
#
# output log_tcpdump: tcpdump.log

# database: log to a variety of databases
# ---------------------------------------
# See the README.database file for more information about configuring
# and using this plugin.
#
# output database: log, mysql, user=root password=test dbname=db
host=localhost
# output database: alert, postgresql, user=snort dbname=snort
# output database: log, odbc, user=snort dbname=snort
# output database: log, mssql, dbname=snort user=snort password=test
# output database: log, oracle, dbname=snort user=snort password=test

# unified: Snort unified binary format alerting and logging
# ---------------------------------------------------------
# The unified output plugin provides two new formats for logging and
generating
```

```
# alerts from Snort, the "unified" format.  The unified format is a
straight
# binary format for logging data out of Snort that is designed to be
fast and
# efficient.  Used with barnyard (the new alert/log processor), most
of the
# overhead for logging and alerting to various slow storage
mechanisms such as
# databases or the network can now be avoided.
#
# Check out the spo_unified.h file for the data formats.
#
# Two arguments are supported.
#    filename - base filename to write to (current time_t is
appended)
#    limit    - maximum size of spool file in MB (default: 128)
#
# output alert_unified: filename snort.alert, limit 128
# output log_unified: filename snort.log, limit 128

# You can optionally define new rule types and associate one or more
output
# plugins specifically to that type.
#
# This example will create a type that will log to just tcpdump.
# ruletype suspicious
# {
#   type log
#   output log_tcpdump: suspicious.log
# }
#
# EXAMPLE RULE FOR SUSPICIOUS RULETYPE:
# suspicious tcp $HOME_NET any -> $HOME_NET 6667 (msg:"Internal IRC
Server";)
#
# This example will create a rule type that will log to syslog and a
mysql
# database:
# ruletype redalert
# {
#   type alert
#   output alert_syslog: LOG_AUTH LOG_ALERT
#   output database: log, mysql, user=snort dbname=snort
host=localhost
# }
#
# EXAMPLE RULE FOR REDALERT RULETYPE:
# redalert tcp $HOME_NET any -> $EXTERNAL_NET 31337 \
#   (msg:"Someone is being LEET"; flags:A+;)

#
# Include classification & priority settings
# Note for Windows users:  You are advised to make this an absolute
path,
# such as:  c:\snort\etc\classification.config
#

include classification.config

#
# Include reference systems
```

```
# Note for Windows users:  You are advised to make this an absolute
path,
# such as:  c:\snort\etc\reference.config
#

include reference.config

#####################################################################
# Step #4: Configure snort with config statements
#
# See the snort manual for a full set of configuration references

config flowbits_size: 256

#####################################################################
# Step #5: Customize your rule set
#
# Up to date snort rules are available at http://www.snort.org
#
# The snort web site has documentation about how to write your own
custom snort
# rules.
#
# The rules included with this distribution generate alerts based on
on
# suspicious activity. Depending on your network environment, your
security
# policies, and what you consider to be suspicious, some of these
rules may
# either generate false positives ore may be detecting activity you
consider to
# be acceptable; therefore, you are encouraged to comment out rules
that are
# not applicable in your environment.
#
# The following individuals contributed many of rules in this
distribution.
#
# Credits:
#   Ron Gula <rgula@securitywizards.com> of Network Security Wizards
#   Max Vision <vision@whitehats.com>
#   Martin Markgraf <martin@mail.du.gtn.com>
#   Fyodor Yarochkin <fygrave@tigerteam.net>
#   Nick Rogness <nick@rapidnet.com>
#   Jim Forster <jforster@rapidnet.com>
#   Scott McIntyre <scott@whoi.edu>
#   Tom Vandepoel <Tom.Vandepoel@ubizen.com>
#   Brian Caswell <bmc@snort.org>
#   Zeno <admin@cgisecurity.com>
#   Ryan Russell <ryan@securityfocus.com>



#==================================
# Include all relevant rulesets here
#
# The following rulesets are disabled by default:
#
#   web-attacks, backdoor, shellcode, policy, porn, info, icmp-info,
virus,
#   chat, multimedia, and p2p
```

```
#
# These rules are either site policy specific or require tuning in
order to not
# generate false positive alerts in most enviornments.
#
# Please read the specific include file for more information and
# README.alert_order for how rule ordering affects how alerts are
triggered.
#========================================
include $RULE_PATH/attack-responses.rules
include $RULE_PATH/local.rules
include $RULE_PATH/bad-traffic.rules
include $RULE_PATH/exploit.rules
include $RULE_PATH/scan.rules
include $RULE_PATH/finger.rules
include $RULE_PATH/ftp.rules
include $RULE_PATH/telnet.rules
include $RULE_PATH/rpc.rules
include $RULE_PATH/rservices.rules
include $RULE_PATH/dos.rules
include $RULE_PATH/ddos.rules
include $RULE_PATH/dns.rules
include $RULE_PATH/tftp.rules
include $RULE_PATH/password.rules
include $RULE_PATH/outcoming.rules

include $RULE_PATH/web-cgi.rules
include $RULE_PATH/web-coldfusion.rules
include $RULE_PATH/web-iis.rules
include $RULE_PATH/web-frontpage.rules
include $RULE_PATH/web-misc.rules
include $RULE_PATH/web-client.rules
include $RULE_PATH/web-php.rules

include $RULE_PATH/sql.rules
include $RULE_PATH/x11.rules
include $RULE_PATH/icmp.rules
include $RULE_PATH/netbios.rules
include $RULE_PATH/misc.rules
include $RULE_PATH/oracle.rules
include $RULE_PATH/mysql.rules
include $RULE_PATH/snmp.rules

include $RULE_PATH/smtp.rules
include $RULE_PATH/imap.rules
include $RULE_PATH/pop2.rules
include $RULE_PATH/pop3.rules

include $RULE_PATH/nntp.rules
include $RULE_PATH/other-ids.rules
# include $RULE_PATH/web-attacks.rules
include $RULE_PATH/backdoor.rules
# include $RULE_PATH/shellcode.rules
# include $RULE_PATH/policy.rules
# include $RULE_PATH/porn.rules
# include $RULE_PATH/info.rules
# include $RULE_PATH/icmp-info.rules
include $RULE_PATH/virus.rules
# include $RULE_PATH/chat.rules
# include $RULE_PATH/multimedia.rules
# include $RULE_PATH/p2p.rules
```

```
include $RULE_PATH/experimental.rules

# Include any thresholding or suppression commands. See
threshold.conf in the
# <snort src>/etc directory for details. Commands don't necessarily
need to be
# contained in this conf, but a separate conf makes it easier to
maintain them.
# Note for Windows users:  You are advised to make this an absolute
path,
# such as:  c:\snort\etc\threshold.conf
# Uncomment if needed.
# include threshold.conf
```

ΒΙΒΛΙΟΓΡΑΦΙΑ

- **[1]** Ηλιούδης Χ., Ασφάλεια και Διαχείριση Δικτύων
- **[2]** Borders K., Prakash A., "Web Tap: Detecting Covert Web Traffic"
- **[3]** Presspoint.gr URL: http://www.presspoint.gr/release.asp?id=51220
- **[4]** Pathfiner URL: http://tech.pathfinder.gr/IT/security/1.html
- **[5]** Linux Security.com URL: http://www.linuxsecurity.com/content/view/117463/49/
- **[6]** Σοφοκλής Χ" Σοφοκλέους, Υπηρεσίες Πληροφορικής ΑΤΗΚ, «Οδός Επικοινωνιών», Απρίλιος 2001, Τεύχος 32
- **[7]** Mitnick K., Simon W., "The Art of Deception. Controlling the Human Element of Security"
- **[8]** Miller T., "Social Engineering: Techniques that can bypass Intrusion Detection Systems" URL: http://www.securityfocus.com/infocus/1229
- **[9]** SecurityWorld.com URL: http://www.securityworld.com/community/hottopics/lovebugvirus.html
- **[10]** BBC News http://news.bbc.co.uk/1/hi/sci/tech/744537.stm
- **[11]** FacilityCity.com URL: http://www.facilitycity.com/tfm/tfm_03_06_news1.asp
- **[12]** zZine Magazine URL: http://www.zzine.org/read.php?op=view&item=802
- **[13]** http://faculty.ncwc.edu/toconnor/426/426lect04.htm
- **[14]** O' Brien D., "Recognizing and Recovering from Rootkit Attacks", 1996
- **[15]** Γκρίτζαλης Σ., Συστήματα Ανίχνευσης Εισβολέων
- **[16]** Denning D., "An Intrusion Detection Model", IEEE Transactions on Software Engineering, Vol.13, No.2, pp.222-232, 1987
- **[17]** Bishop M., Computer Security: Art and Science, Addison Wesley, Boston, 2003
- **[18]** Gonzales J. M., "Defending Networks with Intrusion Detection Systems", June 2004
- **[19]** SecurityFocus.com Intrusion Detection, Theory and Practice URL: http://www.securityfocus.com/infocus/1203
- **[20]** InfoSysSsec.com http://www.infosyssec.net/infosyssec/intdet1.htm
- **[21]** http://www.robertgraham.com/pubs/network-intrusion-detection.html
- **[22]** Tan L., Jin Z., Mu Z., Tuominen O., "Intrusion Detection System (IDS)"
- **[23]** Staniford-Chen S., Cheung S., Crawford R., Dilger M., Frank J., Hoagland J., Levitt K., Wee C., Yip R., Zerkle D., "GrIDS – A Graph-Based Intrusion Detection System for Large Networks", in Proceedings of the 19th National Information Systems Security Conference, 1996

- **[24]** Northcutt S., Computer Security Incident Handling: Step by Step, The SANS Institute, Bethesda, 1998
- **[25]** Heberlein L., Levitt K., Mukherjee B., "Internetwork Security Monitor: An Intrusion-Detection System for Large-Scale Networks", in Proceedings of the 15th National Information Systems Security Conference, 1992
- **[26]** Staniford-Chen S., Heberlein L., "Holding Intruders Accountable on the Internet", in Proceedings of the 1995 IEEE Symposium on Security and Privacy, 1995
- **[27]** Savage S., Wetherall D., Karlin A., Anderson T., "Practical Network Support for IP Traceback", Computer Communication Review, 2000
- **[28]** Ptacek T., "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection"
  URL: http://secinf.net/info/ids/idspaper/idspaper.html
- **[29]** Egaas A., Gregorio A., Keto E., "Intrusion Detection Systems"
- **[30]** Tanase M., "The Future of IDS"
  URL: http://www.securityfocus.com/infocus/1518
- **[31]** Miller T., "Social Engineering: Techniques that can bypass Intrusion Detection Systems" URL: http://www.securityfocus.com/infocus/1229
- **[32]** Dolan A., "Social Engineering", SANS Institute 2004
- **[33]** Granger S., "Social Engineering Fundamentals, Part I: Hacker Tactics" URL: http://www.securityfocus.com/infocus/1527
- **[34]** Radha Gulati, "The Threat of Social Engineering and Your Defense Against It" SANS Institute 2003
- **[35]** Cohen F. & Associates "50 Ways to Defeat Your Intrusion Detection System"
- **[36]** Borders K., Prakash A. "Web Tap: Detecting Covert Web Traffic"
- **[37]** Dragon IDS, http://www.intrusion-detection-system-group.co.uk/dragon.htm
- **[38]** "Advanced Detection Technology", URL: www.fortinet.com
- **[39]** SNORT.ORG www.snort.org
- **[40]** Snort users manual
  http://www.snort.org/docs/snort_htmanuals/htmanual_233/
- **[41]** Koziol J., "Intrusion Detection with Snort"
- Παγκάλου Γ., Ασφάλεια Πληροφοριακών Συστημάτων και Δικτύων