ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΣΕΡΡΩΝ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΕΠΙΚΟΙΝΩΝΙΩΝ

Πτυχιακή Εργασία

με τίτλο:

« ΣΥΣΤΗΜΑΤΑ ΑΝΙΧΝΕΥΣΗΣ ΕΙΣΒΟΛΩΝ ΣΕ ΔΙΚΤΥΑ_ ΜΙΑ ΕΦΑΡΜΟΓΗ ΤΟΥ ΛΟΓΙΣΜΙΚΟΥ SNORT ΣΤΟ ΠΕΡΙΒΑΛΛΟΝ ΤΟΥ ΔΙΚΤΥΟΥ ΤΟΥ ΤΕΙ ΣΕΡΡΩΝ»

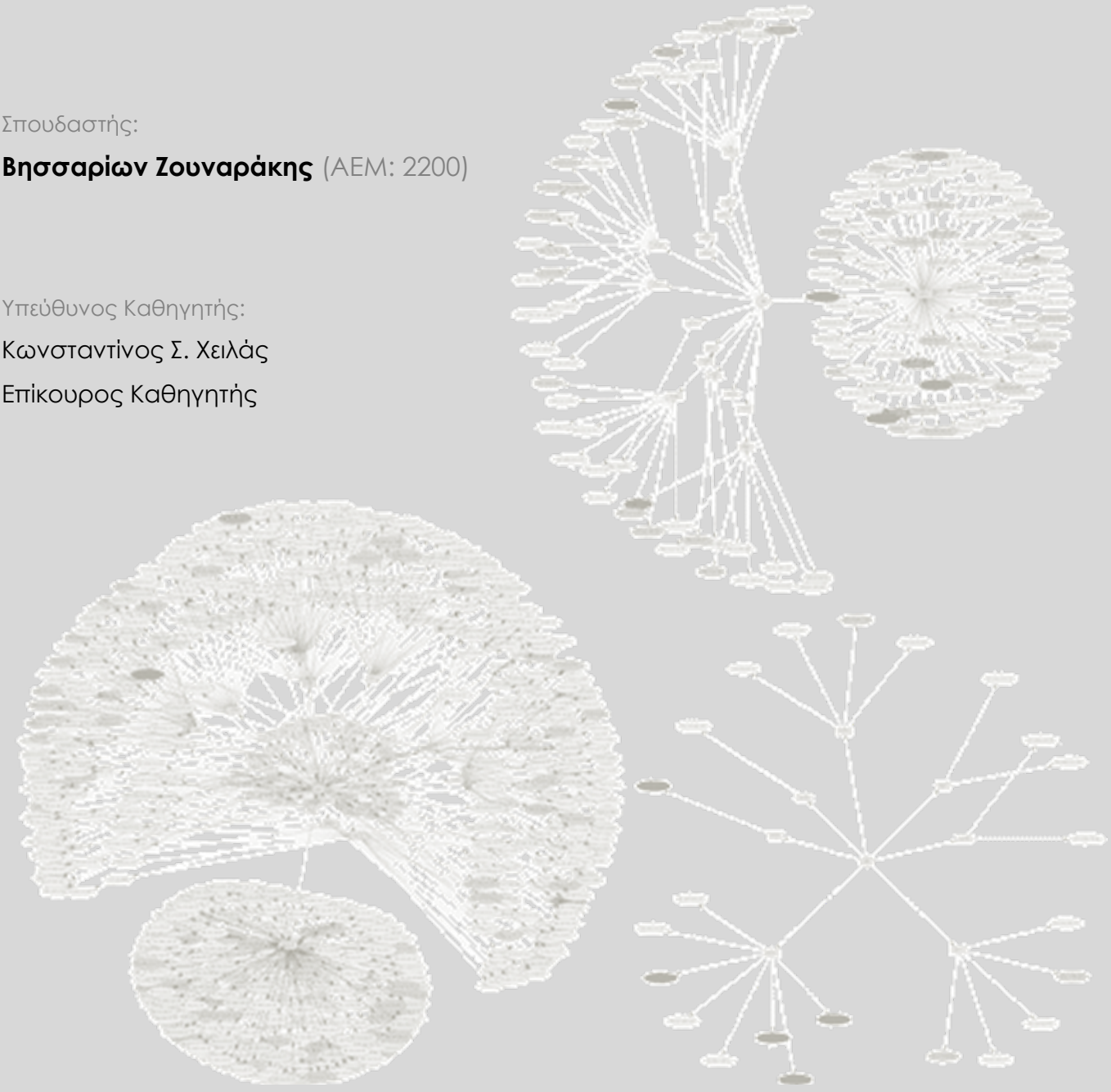
Σπουδαστής:

Βησσαρίων Ζουναράκης (ΑΕΜ: 2200)

Υπεύθυνος Καθηγητής:

Κωνσταντίνος Σ. Χειλάς

Επίκουρος Καθηγητής



ΜΑΙΟΣ 2013
ΣΕΡΡΕΣ

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΣΕΡΡΩΝ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΕΠΙΚΟΙΝΩΝΙΩΝ

Πτυχιακή Εργασία

με τίτλο:

« ΣΥΣΤΗΜΑΤΑ ΑΝΙΧΝΕΥΣΗΣ ΕΙΣΒΟΛΩΝ ΣΕ ΔΙΚΤΥΑ_ ΜΙΑ ΕΦΑΡΜΟΓΗ ΤΟΥ ΛΟΓΙΣΜΙΚΟΥ SNORT ΣΤΟ ΠΕΡΙΒΑΛΛΟΝ ΤΟΥ ΔΙΚΤΥΟΥ ΤΟΥ ΤΕΙ ΣΕΡΡΩΝ»

Σπουδαστής:

Βησσαρίων Ζουναράκης (ΑΕΜ: 2200)

Υπεύθυνος Καθηγητής:

Κωνσταντίνος Σ. Χειλάς

Επίκουρος Καθηγητής

ΜΑΙΟΣ 2013
ΣΕΡΡΕΣ

ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ:

Βεβαιώνω ότι είμαι συγγραφέας αυτής της πτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην πτυχιακή εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η πτυχιακή εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τις απαιτήσεις του προγράμματος σπουδών του Τμήματος Πληροφορικής & Επικοινωνιών του Τ.Ε.Ι. Σερρών.

ΕΥΧΑΡΙΣΤΙΕΣ

Στο σημείο αυτό, θέλω να ευχαριστήσω τον καθηγητή μου, κ. Κωνσταντίνο Σ. Χειλά , Επίκουρο Καθηγητή του Τμήματος Πληροφορικής & Επικοινωνιών, για την κριτική και τις συμβουλές που μου έδωσε κατά την διάρκεια εκπόνησης της πτυχιακής μου εργασίας. Οφείλω επίσης ευγνωμοσύνη στην οικογένειά μου, για την υπομονή και την συμπαράσταση όλο αυτό το διάστημα.

ΠΕΡΙΕΧΟΜΕΝΑ

1 Εισαγωγή 1

2 Ασφάλεια Δικτύων 3

2.1 Γενικά.....3
2.2 Αδύναμα Σημεία της Ασφάλειας των Δικτύων 4
2.3 Τύποι Επιθέσεων 5
2.4 Πολιτικές Ασφάλειας Δικτύων 7
2.5 Αρχιτεκτονική Ασφάλειας Δικτύων 8

3 Συστήματα Ανίχνευσης Εισβολών (Intrusion Detection Systems – IDS) 11

3.1 Στάδια Ανάπτυξης 11
3.2 Γενικά 14
3.3 Τρόποι Λειτουργίας των IDS 14
3.4 Υποκατηγορίες των IDS 16
3.5 Πρωτόκολλα που Ανιχνεύουν τα IDS.....19
3.5.1 TCP (Transmission Control Protocol).....20
3.5.2 UDP (User Datagram Protocol).....22
3.5.3 ICMP (Internet Control Message Protocol).....22

4 Το Λογισμικό Snort 24

4.1 Γενικά 24
4.2 Λόγοι Επιλογής του Snort 25
4.3 Τρόποι Λειτουργίας 26
4.4 Κανόνες (Rules) ή Υπογραφές (Signatures) των IDS 27
4.4.1 Βασικές Διαφορές 27
4.5 Οι Κανόνες (Rules) 28
4.6 Η Αρχιτεκτονική του Snort 33
4.7 Βιβλιοθήκη Συλλογής Πακέτων (Packet Capture Library – Libpcap) 34
4.8 Αποκωδικοποιητής Πακέτων (Packet Decoding) 35
4.9 Προ-επεξεργαστές (Preprocessors) 35
4.10 Μηχανή Ανίχνευσης (Detection Engine) 40
4.11 Πακέτα Λογισμικού Αποτύπωσης Εξόδου (Output Plug – INS) 41

5 Εργαστηριακό Μέρος 44

5.1 Εγκατάσταση και Ρύθμιση του SNORT IN IDS MODE σε Windows 7 45
5.2 Ρύθμιση Κανόνων για Ανίχνευση IDS 52
5.3 Υλοποίηση Επιθέσεων με το Λογισμικό NMAP - ZENMAP GUI 53
5.4 Αποτελέσματα Επιθέσεων ανά Κατηγορία 54
5.4.1 Επίθεση από το Εσωτερικό του Δικτύου 54
5.4.2 Επίθεση με Σάρωση Δικτυακών Θυρών TCP 54
5.4.3 Επίθεση με Σάρωση Πακέτων UDP 55
5.4.4 Επίθεση με Σάρωση Πακέτων ICMP μέσω PING 56
5.4.5 Επίθεση Εκτός του Εσωτερικού Δικτύου 57
5.5 Συμπεράσματα 57

6 Παράρτημα 59

7 Βιβλιογραφία 72

ΕΙΣΑΓΩΓΗ

Το θέμα της παρούσας εργασίας σχετίζεται με τα συστήματα ανίχνευσης εισβολών (intrusion detection system) σε δίκτυα. Στη σημερινή εποχή η ανάγκη προστασίας των δικτύων υπολογιστών είναι αυξημένη και επιτακτική. Στόχος της εργασίας είναι να αναδείξει τα συστήματα ανίχνευσης εισβολών και να διερευνήσει τους μηχανισμούς εκείνους που θα μπορούσαν να οδηγήσουν στην ομαλή λειτουργία των δικτύων.

Αυτό επιχειρείται να γίνει μέσα από την εφαρμογή του λογισμικού Snort στο περιβάλλον του δικτύου του ΤΕΙ Σερρών, το οποίο θα είναι και το βασικό εργαλείο για την επίτευξη του στόχου αυτού. Η εργασία θα κινηθεί, αρχικά, σε ιστορική αναδρομή των συστημάτων ανίχνευσης εισβολών, και έπειτα σε περιγραφή και μελέτη της συμπεριφοράς του λογισμικού Snort σε ρεαλιστικό περιβάλλον, στο βαθμό όπου αυτό είναι εφικτό.

Τα συστήματα ανίχνευσης δικτυακών εισβολών έχουν σαν κύρια αποστολή την έγκαιρη ανίχνευση αλλά και την αποτροπή των διάφορων επιθέσεων που απειλούν στις μέρες μας όλα τα δίκτυα υπολογιστών. Η ανάπτυξη της τεχνολογίας αυτών των συστημάτων έχει καταγράψει μεγάλη πρόοδο τα τελευταία χρόνια και τείνουν να γίνουν απαραίτητο κομμάτι κάθε ολοκληρωμένης αρχιτεκτονικής ασφάλειας δικτύων.

Εστιάζοντας λοιπόν, στην ανάγκη αυτή, επιχειρείται η διεξοδική μελέτη του λογισμικού Snort, το οποίο αποτελεί ένα μέρος –το πρωταρχικό σε επίπεδο ασφάλειας μονάδας- του συστήματος ασφάλειας σε δίκτυα, στα πλαίσια της απλούστερης μορφής του, ώστε να είναι όσο το δυνατόν ακριβέστερα και λεπτομερή τα συμπεράσματα της μελέτης. Η έρευνα πραγματώνεται επίσης και για να προσεγγιστεί εργαστηριακά ο βαθμός αξιοπιστίας των συστημάτων ανίχνευσης εισβολών (IDS), στη μικρότερη δυνατή μονάδα δικτύου.

ΔΙΚΤΥΑ

2. ΑΣΦΑΛΕΙΑ ΔΙΚΤΥΩΝ

2.1 ΓΕΝΙΚΑ

Η ανάπτυξη των δικτύων στους υπολογιστές ήταν ραγδαία της τελευταίες δεκαετίες παγκοσμίως, για την κάλυψη αναγκών μεγάλων εταιριών, την συνεργασία πανεπιστημίων, άλλων φορέων, ακόμα και του απλού πολίτη. Αυτό έχει σαν αποτέλεσμα την προσπάθεια και τη δημιουργία μίας αρχιτεκτονικής ασφάλειας δικτύων-καίριας σημασίας-για την κάλυψη, όσο το δυνατόν πιο επιτυχημένα, των αδυναμιών των δικτύων, προς αποφυγή παραβίασής τους.

Με την έννοια ασφάλεια δικτύων, αναφερόμαστε σ' ένα πολυσύνθετο κομμάτι, το οποίο αποτελείται από διάφορους κανόνες, πολιτικές ασφάλειας, ενέργειες αλλά και εξοπλισμό(software/hardware). Με τον όρο «ασφάλεια», υποδηλώνονται ένα πλήθος εννοιών, όπως είναι η εμπιστευτικότητα (Confidentiality), δηλαδή η προστασία της πληροφορίας από μη εξουσιοδοτημένη πρόσβαση, η ακεραιότητα (Integrity), δηλαδή η προστασία της πληροφορίας από μη εξουσιοδοτημένη τροποποίηση και τέλος η διαθεσιμότητα (Availability), όπου τα δεδομένα υπάρχουν στη θέση, στον χρόνο και στην μορφή που ο χρήστης τα χρειάζεται. Άλλες εκφάνσεις της εμπιστευτικότητας είναι: η ιδιωτικοποίηση(privacy), δηλαδή η προστασία των προσωπικών δεδομένων καθώς και η μυστικότητα(secrecy), δηλαδή η προστασία δεδομένων που ανήκουν σε μία εταιρεία. Δίνοντας λοιπόν, μεγάλη βαρύτητα στην ασφάλεια των δικτύων αποτρέπουμε τυχόν επιθέσεις που έχουν ως αντίκτυπο οικονομικές απώλειες, πιθανή σπατάλη πολύτιμου χρόνου για την αποκατάσταση του συστήματος που δέχτηκε επίθεση, όπως και την αποφυγή ενδεχόμενου καίριου πλήγματος στο κύρος και την αξιοπιστία του συστήματος.

2.2 ΑΔΥΝΑΜΑ ΣΗΜΕΙΑ ΤΗΣ ΑΣΦΑΛΕΙΑΣ ΤΩΝ ΔΙΚΤΥΩΝ

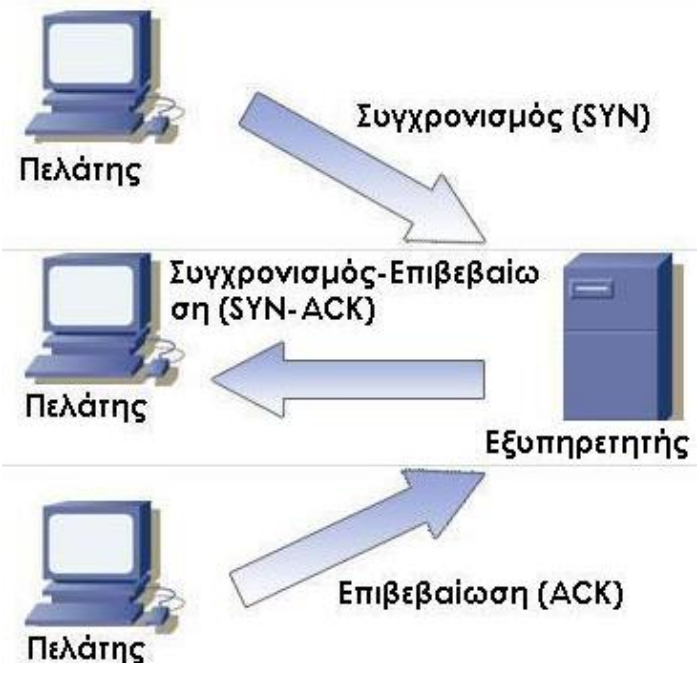
Τα υπολογιστικά συστήματα είναι αρκετά ευάλωτα στις μέρες μας για τον λόγο ότι, οι χρήστες των συστημάτων αυτών αγνοούν τους παράγοντες που πολλαπλασιάζουν τις πιθανότητες μίας επίθεσης. Κυριότεροι παράγοντες εξ' αυτών είναι οι ακόλουθοι: Πρώτον, οι αδύναμοι κωδικοί πρόσβασης που επιλέγονται ως «κλειδιά ασφαλείας» είτε από τους διαχειριστές είτε από τους ίδιους τους χρήστες στο εκάστοτε σύστημα. Μπορεί αρχικά να χρησιμοποιούνται ως συνθηματικά εισόδου σε κάποιο σύστημα, τα οποία να μπορεί να αναγνωρίσει μόνο ο αντίστοιχος χρήστης ή διαχειριστής, όμως λόγω του αναμενόμενου της «κρυφής-απόρρητης» πληροφορίας, παραδείγματος χάριν ημερομηνίες γεννήσεως, ονομασίες συγγενικών προσώπων και άλλα παρόμοια, οι κωδικοί αυτοί είναι ευάλωτοι στην αποκρυπτογράφησή τους. Δεύτερον, η λανθασμένη εγκατάσταση και ρύθμιση δικτυακών συσκευών, όπως δρομολογητές(routers), servers κτλ. είναι πιθανό να δώσουν πάτημα στον εισβολέα να εισέρθει στο σύστημα με σχετική άνεση, χωρίς πολλές φορές να έχουμε καμία ένδειξη για αυτή την εισβολή. Η ζημιά που μπορεί να προκαλέσει κάτι τέτοιο, μπορεί να είναι από πολύ μικρή και σχετικά αθώα, όπως εκμετάλλευση του δικτύου του χρήστη χωρίς την άδειά του, μέχρι και πολύ μεγάλη με ανεπανόρθωτες συνέπειες, όπως κλοπή και διακίνηση απόρρητων πληροφοριών. Τρίτον, η παλαιότητα του λογισμικού (software) ενός συστήματος, ή πιθανά λάθη στον κώδικά του (bugs), ή κάποια λανθασμένη παραμετροποίηση, ή ακόμα και μη σωστά διορθωτικά προγράμματα(patches), μπορούν με τη σειρά τους να αποτελέσουν «κερκόπορτες» για τους επίδοξους εισβολείς. Τέλος, το ίδιο το ανθρώπινο δυναμικό που χρησιμοποιεί το εκάστοτε σύστημα, είτε από μη σωστή εκπαίδευση είτε από σφάλμα της πολιτικής ασφάλειας είτε από ελλιπής ενημέρωσή του, αποτελεί έναν από τους κατεξοχήν λόγους που ένα σύστημα γίνεται ευάλωτο.

2.3 ΤΥΠΟΙ ΕΠΙΘΕΣΕΩΝ

Στο σημείο αυτό είναι καλό να προσδιοριστούν και ορισμένοι από τους βασικούς τύπους επιθέσεων που μπορούν να λάβουν χώρα στα διάφορα πληροφοριακά συστήματα δικτύων.

Αρχικά, μια από τις πλέον γνωστές και διαδεδομένες μορφές επιθέσεων είναι η εισβολή ιών (viruses) στο σύστημα. Στόχος τους είναι να μεταβάλουν το λειτουργικό σύστημα του συστήματος ή του λογισμικού, με απώτερο σκοπό συνήθως την καταστροφή του. Μια άλλη μορφή επίθεσης είναι οι Δούρειοι Ίπποι (Trojans). Αυτοί δίνουν την ψευδαίσθηση της εκτέλεσης νόμιμων και χρήσιμων λειτουργιών στο σύστημα χωρίς να γίνονται αντιληπτοί, ενώ στην ουσία εκτελούν επιβλαβείς ενέργειες με τις οποίες καταστρέφουν δεδομένα ή εγκαθιστούν λογισμικό για την επίτευξη της επίθεσης.

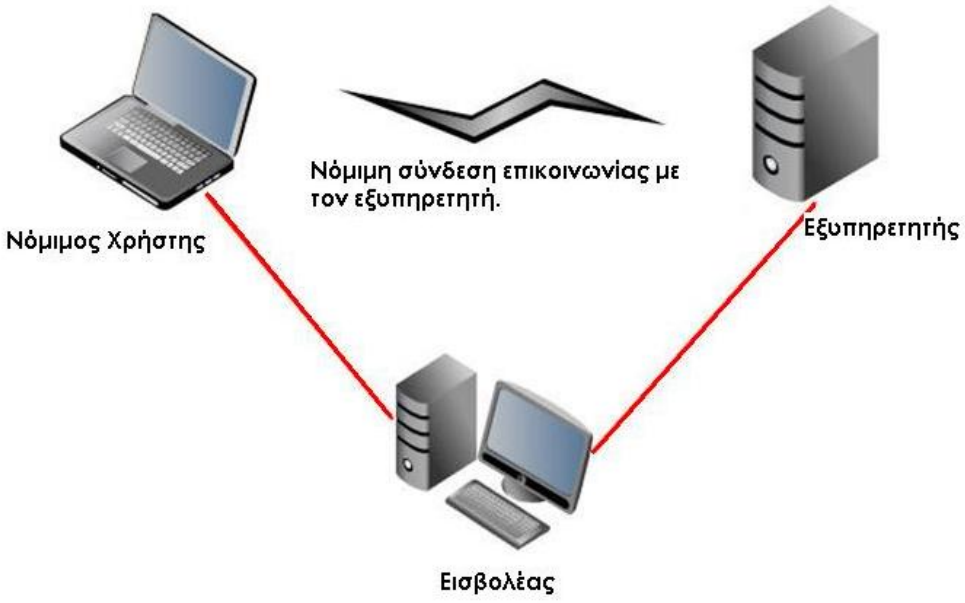
Για την επίτευξη μίας επικοινωνίας ενός εξυπηρετητή μ' έναν πελάτη με την χρήση του πρωτοκόλλου TCP/IP απαιτείται μία διαδικασία τριών βημάτων (3-way handshake). Αρχικά ο πελάτη (client) στέλνει ένα μήνυμα SYN(synchronize) στον εξυπηρετητή (server) και αναμένει την απόκρισή του. Ο εξυπηρετητής με την σειρά του μόλις λάβει το μήνυμα από τον πελάτη έχοντας δεδομένη την ολοκλήρωση της συνόδου δεσμεύει πόρους για την ομαλή υπό-εγκατάσταση της συνόδου αυτής και στέλνει μήνυμα SYN/ACK στον πελάτη για να του ανακοινώσει πως είναι σε αναμονή για την εξυπηρέτηση της συνόδου. Τέλος ο πελάτης στέλνει ένα μήνυμα επιβεβαίωσης ACK (acknowledgment) για να ξεκινήσει η σύνδεση.



Εικόνα 1 TCP/IP 3-way handshake

Οι hackers βασιζόμενοι σε αυτό τον τρόπο λειτουργίας του πρωτοκόλλου TCP/IP, στέλνουν μεγάλο αριθμό από SYN μηνύματα από μη υπαρκτές (spoofed) IP διευθύνσεις με αποτέλεσμα οι διαθέσιμοι πόροι των συστημάτων εξυπηρέτησης να δεσμεύονται, ενώ τα SYN/ACK μηνύματα να μην καταλήγουν σε κάποιο υπαρκτό προορισμό. Αυτοί οι πόροι βάση των κανόνων λειτουργίας αυτού του πρωτοκόλλου παραμένουν δεσμευμένοι για κάποιο χρονικό διάστημα. Σε αυτό το διάστημα αν κάποιος νόμιμος χρήστης-πελάτης θέλει να κάνει χρήση αυτού του συγκεκριμένου εξυπηρετητή δεν θα έχει την δυνατότητα λόγω ΑΡΝΗΣΗΣ εξυπηρέτησης του εξυπηρετητή. Από εκεί προέρχεται και η ονομασία της επίθεσης, DoS (Denial of Service).

Άλλος τρόπος επίθεσης επίσης είναι και η μέθοδος της κλοπής πληροφοριών, με την πιο διαδεδομένη μέθοδο, αυτήν του «μεσάζοντα» (Man in the middle). Σε αυτή την περίπτωση ο εισβολέας μπαίνει ανάμεσα στη ζεύξη του νόμιμου χρήστη με τον εξυπηρετητή(server) και προσποιούμενος τον server εξαπατά τον χρήστη ο οποίος νομίζει ότι οι πληροφορίες που δίνει όπως τον αριθμό της πιστωτικής του κάρτας είναι ασφαλείς. Από εκεί και πέρα είναι θέμα του εισβολέα το πώς θα χρησιμοποιήσει τις πληροφορίες αυτές.



Εικόνα 2 Μέθοδος του «μεσάζοντα» (Man in middle)

Επίσης, με τα προγράμματα σκουλήκια (worms), πέρα από υποκλοπή ή καταστροφή των δεδομένων ενός συστήματος, είναι πολύ επικίνδυνα και λόγω του χαοτικού τους εύρους. Τα εν λόγω προγράμματα αυτό-αναπαράγονται και εξαπλώνονται με αποτέλεσμα να αυξάνουν τον επεξεργαστικό φόρτο του συστήματος και τελικά να το κάνουν να υπολειτουργεί.

Τέλος, ένας τρόπος επίθεσης είναι όταν ο εισβολέας προσπαθεί να αποσπάσει πληροφορίες που χρειάζεται ερχόμενος σε επαφή με ανυποψίαστους χρήστες του συστήματος που θέλει να επιτύχει την επίθεση μέσω των «κοινωνικών μέσων» (social engineering). Οι συνήθεις τρόποι επικοινωνίας είναι μέσω τηλεφωνικής συζήτησης, ηλεκτρονικού ταχυδρομείου (email) κτλ.

2.4 ΠΟΛΙΤΙΚΕΣ ΑΣΦΑΛΕΙΑΣ ΔΙΚΤΥΩΝ

Βασικός παράγοντας για την σωστή ασφάλεια οποιουδήποτε δικτύου είναι η χάραξη μίας προκαθορισμένης πολιτικής που θα καθορίζει τον τρόπο με τον οποίο θα λειτουργεί αυτή. Η πολιτική αυτή (policy Security) στις μέρες μας είναι πλέον διεθνών προδιαγραφών. Υπάρχουν δηλαδή πρότυπα που μπορεί μία εταιρία ή ένας οργανισμός ν' ακολουθήσει εφαρμόζοντας στο δίκτυό του, όπως είναι το ISO 27002(update 17799) πρότυπο, που αναφέρει λεπτομερώς τα σημεία που πρέπει να καλυφθούν ώστε να πιστοποιείται η ασφάλεια δικτύου του. Κρίσιμα σημεία που χρήζουν προσοχής είναι τα ακόλουθα:

Πρώτον, ο έλεγχος ασφαλούς λειτουργίας, τόσο σε επίπεδο δικτύου όσο και σε επίπεδο υπολογιστών που το απαρτίζουν (host). Είναι σημαντικό να υπάρχει τέτοια μέριμνα, αφού κάτι τέτοιο ουσιαστικά θα λειτουργεί σαν δεύτερο τμήμα άμυνας, προστατεύοντας από κάποια επίθεση ακόμα και αν αυτή επιτύχει να ξεπεράσει τα όποια μέτρα ασφάλειας στα όρια του δικτύου.

Δεύτερον, το άρτια οργανωμένο σύστημα ανίχνευσης επιθέσεων (Intrusion Detection System). Αυτό επιδιώκει έγκαιρα, επακριβώς και αποτελεσματικά να

ενημερώνει για την ύπαρξη επιθέσεων στο σύστημα και να λαμβάνει τα κατάλληλα μέτρα για, την καταρχήν, αποτροπή τους.

Τρίτον, τα σωστά καταρτισμένα συστήματα εξουσιοδότησης (Authorization) και ελέγχου πρόσβασης (Access control) που αξιόπιστα θα τσεκάρουν καθέναν που προσπαθεί να μπει στο δίκτυο και τις όποιες υπηρεσίες αυτό παρέχει.

Τέταρτον, τα πλήρη εφεδρικά (backup) συστήματα αποθήκευσης και αποκατάστασης. Αυτά λειτουργούν σε περίπτωση επιτυχούς επίθεσης από την οποία μπορεί να καταρρεύσει το δίκτυο.

Πέμπτον, η σύγχρονη και αξιόπιστη τεχνολογία κρυπτογράφησης. Σκοπός της είναι να καταστήσει σαφώς ασφαλέστερες τις επικοινωνίες ανάμεσα σε κόμβους του εσωτερικού δικτύου, αλλά και σε αυτές που πραγματοποιούνται μέσω του διαδικτύου(Internet).

Και έκτον, η αποτελεσματική φυσική φύλαξη του χώρου. Διότι όσο άρτια και αν είναι η δικτυακή ασφάλεια, σε περίπτωση που δεν συνδυάζεται από εξίσου ισχυρή φυσική ασφάλεια, είναι σαφές ότι δεν θα έχει καμία αξία.

2.5 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΑΣΦΑΛΕΙΑΣ ΔΙΚΤΥΩΝ

Μια σωστά δομημένη αρχιτεκτονική ασφάλειας δικτύου υπολογιστών, πέραν των παραπάνω ενεργειών, εμπεριέχει και άλλες συνιστώσες που αυξάνουν το επίπεδο ασφάλειας, με την ορθή εγκατάσταση και τον σωστό χειρισμό τους.

Οι κρυπτογραφικές τεχνικές, που είναι ένα εξαιρετικής σημασίας μέσο, μπορούν να χρησιμοποιηθούν για την ασφαλή μεταφορά της πληροφορίας που μεταδίδονται μεταξύ συστημάτων μιας και καθιστά την υποκλοπή αρκετά αδύνατη. Η κρυπτογραφία μπορεί να εφαρμοστεί σε πολλά επίπεδα, όπως στο επίπεδο μεταφοράς, στο επίπεδο δικτύου (π.χ. μεταξύ πελάτη-εξυπηρετητή, πελάτη με πελάτη κ.α.).

Τα αντίγραφα ασφαλείας, τα οποία σε περίπτωση επίθεσης και κατάρρευσης του συστήματος, καθιστούν μη επίπονη και χρονοβόρα την αποκατάσταση και την δημιουργία, πάλι, του όγκου των αποθηκευμένων πληροφοριών στην προτεραία του κατάσταση.

Το λογισμικό απομάκρυνσης ιών (Anti-virus software), το οποίο ελέγχει σε πραγματικό χρόνο κάθε πληροφορία που εισέρχεται στο σύστημα για κακόβουλα στοιχεία.

Το τείχος προστασίας (Firewall), είναι συσκευή ή λογισμικό όπου με την σωστή ρύθμιση επιτρέπει ή απορρίπτει πακέτα δεδομένων που περνούν από ένα δίκτυο σ' ένα άλλο[1]. Σκοπός του τοίχους προστασίας είναι να προλαμβάνει μία πιθανή επίθεση. Αυτό μπορεί να γίνει απορρίπτοντας όλες τις συνδέσεις στο δίκτυο, εκτός αυτών που ο διαχειριστής του δικτύου επιτρέπει. Το αντίθετο μπορεί να κάνει το δίκτυο ευάλωτο σε επιθέσεις από εξωτερικούς χρήστες. Σήμερα, η 4ης γενιάς τείχος προστασίας είναι ενσωματωμένος στο λειτουργικό σύστημα.

Εκτός όμως από το τείχος προστασίας είναι αναγκαίο και ένα σύστημα όπου θα έχει την δυνατότητα να αξιολογεί μία τυχόν επίθεση, αφού έχει πραγματοποιηθεί και έχει ενεργοποιηθεί ο συναγερμός (alert) σε πραγματικό χρόνο. Μια εσωτερική, δηλαδή, επίβλεψη επιθέσεων στο ίδιο το δίκτυο. Αυτό το ρόλο αναλαμβάνουν τα συστήματα ανίχνευσης εισβολών (Intrusion Detection systems).

Σε αυτό το σημείο αξίζει να επισημανθεί ότι, οι παραπάνω συνιστώσες (λογισμικό ή υλικό)που αναφέρθηκαν, σε μεμονωμένη εφαρμογή τους, δεν επιφέρουν το επιθυμητό αποτέλεσμα ασφάλειας. Είναι αναγκαίο να λειτουργούν όλες παράλληλα, ώστε να παρέχουνε στο σύστημα το καλύτερο δυνατό αποτέλεσμα ασφάλειας.

[1] http://el.wikipedia.org/wiki/Firewall

IDS

3. ΣΥΣΤΗΜΑΤΑ ΑΝΙΧΝΕΥΣΗΣ ΕΙΣΒΟΛΩΝ
(Intrusion Detection Systems – IDS)

3.1 ΣΤΑΔΙΑ ΑΝΑΠΤΥΞΗΣ

Στα πρώτα στάδιά τους τα συστήματα ανίχνευσης εισβολών αποτελούνταν από ένα σύνολο εργαλείων με σκοπό να βοηθήσουν τους διαχειριστές στον έλεγχο των συστημάτων, όπως τα αρχεία καταγραφής πρόσβασης χρήστη (user access logs), τα αρχεία καταγραφής πρόσβασης σε αρχεία (file access logs) και αρχεία καταγραφής συμβάντων του συστήματος (system event logs).

Στην διάρκεια της ανάπτυξής τους, ο Fred Cohen το 1984 εξέφρασε την δυσπιστία του για την δυνατότητα των συστημάτων να ανιχνεύουν μία εισβολή σε κάθε περίπτωση και πως οι πόροι που απαιτούνταν για την ανίχνευση αυξάνονταν με το ποσό της χρήσης[2].

Δύο χρόνια αργότερα, το 1986, μία άλλη επιστήμονας, η Dorothy E. Denning, επικουρούμενη από τον Peter G. Neumann, δημοσίευσε ένα μοντέλο ενός συστήματος ανίχνευσης εισβολών που αποτέλεσε την βάση για την σημερινή εξέλιξή τους[3]. Το μοντέλο της αυτό χρησιμοποιούσε στατιστικές για την ανίχνευση ανωμαλιών, στατιστικές με βάση τα προφίλ των χρηστών, των συστημάτων υποδοχής και τα συστήματα στόχων, καθώς και ένα κανόνα με βάση το έμπειρο σύστημα για την ανίχνευση γνωστών τύπων εισβολών σε συστήματα. Στο ινστιτούτο έρευνας SRI International της Καλιφόρνια, του δόθηκε η ονομασία Intrusion Detection Expert System (IDES). Το μοντέλο αυτό έτρεξε σε σταθμούς εργασίας της εταιρείας Sun Microsystems, όπου θα μπορούσε να εξετάζει και το χρήστη και τα δεδομένα σε επίπεδο δικτύου. Η Teresa F. Lunt, πρότεινε την προσθήκη τεχνικού νευρωνικού δικτύου ως ένα

[2] Cohen, F., 1987. "Computer Viruses Theory and Experiments," Computers and Security, vol. 6, pp. 22--35.
[3] Denning, Dorothy E., "An Intrusion Detection Model," Proceedings of the Seventh IEEE Symposium on Security and Privacy, May 1986, pages 119–131.

τρίτο εργαλείο[4]. Έτσι η SRI προχώρησε το 1993 στην υλοποίηση της επόμενης γενιάς IDES, τα Next-generation Intrusion Detection Expert System (NIDES)[5].

Με βάση των παραπάνω , το 1988 δημιουργήθηκε το Multics Intrusion Detection and Alerting System (MIDAS), ένα έμπειρο σύστημα με P-Best και Lisp, όπου χρησιμοποιούσε στατιστικά για την μείωση διαδρομών ελέγχου[6].

Το Wisdom & Sense (W&S) ήταν και αυτό ένα σύστημα όπου δημιουργώντας κανόνες με βάση τις στατιστικές ανάλυσης, τους χρησιμοποίησε για την ανίχνευση ανωμαλιών. Αναπτύχθηκε το 1989 στο Los Alamos National Laboratory[7] των Η.Π.Α.

Μέσα στον χρόνο έγιναν πολλές απόπειρες βελτίωσης των συστημάτων ανίχνευσης. Μία εξ' αυτών είναι και το Information Security Officer's Assistant (ISOA) μία πρωτοτυπία για το 1990, διότι εμπεριείχε μία ποικιλία στρατηγικών ανίχνευσης, όπως στατιστικές, ένα προφίλ χρήστη και ένα έμπειρο σύστημα[8]. Επίσης και το ComputerWatch, όπου αναπτύχθηκε στο εργαστήριο AT&T Bell Labs,στο Murray Hill του New Jersey και χρησιμοποιούσε στατιστικές και κανόνες για την μείωση ελέγχου των δεδομένων και την ανίχνευση εισβολών[9]. Στη συνέχεια, το 1991, οι ερευνητές του Davis πανεπιστημίου της Καλιφόρνια, δημιούργησαν ένα πρωτότυπο πρότυπο κατανεμημένου συστήματος ανίχνευσης εισβολών (DIDS), το οποία ήταν επίσης ένα έμπειρο σύστημα[10].

---

[4] Lunt, Teresa F., "IDES: An Intelligent System for Detecting Intruders," Proceedings of the Symposium on Computer Security; Threats, and Countermeasures; Rome, Italy, November 22–23, 1990, pages 110–121.

[5] Lunt, Teresa F., "Detecting Intruders in Computer Systems," 1993 Conference on Auditing and Computer Technology, SRI International.

[6] Sebring, Michael M., and Whitehurst, R. Alan., "Expert Systems in Intrusion Detection: A Case Study," The 11th National Computer Security Conference, October, 1988.

[7] Vaccaro, H.S., and Liepins, G.E., "Detection of Anomalous Computer Session Activity," The 1989 IEEE Symposium on Security and Privacy, May, 1989.

[8] Winkeler, J.R., "A UNIX Prototype for Intrusion and Anomaly Detection in Secure Networks," The Thirteenth National Computer Security Conference, Washington, DC., pages 115–124, 1990.

[9] Dowell, Cheri, and Ramstedt, Paul, "The ComputerWatch Data Reduction Tool," Proceedings of the 13th National Computer Security Conference, Washington, D.C., 1990.

[10] Snapp, Steven R, Brentano, James, Dias, Gihan V., Goan, Terrance L., Heberlein, L. Todd, Ho, Che-Lin, Levitt, Karl N., Mukherjee, Biswanath, Smaha, Stephen E., Grance, Tim, Teal,

Την ίδια χρονιά, στο εργαστήριο Los Alamos National Laboratory των Η.Π.Α. αναπτύχθηκε ένα πρωτότυπο σύστημα ανίχνευσης εισβολών, επονομαζόμενο ως Network Anomaly Detection and Intrusion Reporter (NADIR). Ήταν βασισμένο σε μεγάλο βαθμό στο έργο των Dennnig και Lunt[11].

Το Lawrence Berkeley National Laboratory της Καλιφόρνια, το 1998, ανακοίνωσε το λογισμικό Bro. Αυτό χρησιμοποιούσε την δική του γλώσσα κανόνων για την ανάλυση των πακέτων από την βιβλιοθήκη συλλογής Libpcap δεδομένων[12]. Τον Νοέμβρη του ίδιου έτους το APE αναπτύχθηκε ως ένας αναλυτής κίνησης πακέτων δικτύου(sniffer), χρησιμοποιώντας την βιβλιοθήκη libpcap. Ένα μήνα αργότερα το APE μετονομάστηκε σε Snort και ξεκίνησε η ανοδική πορεία του, φτάνοντας να είναι στις μέρες μας από το μεγαλύτερο στον κόσμο IDS/ IPS σύστημα με περισσότερους από 300.000 ενεργούς χρήστες[13].

Το 2001 το σύστημα Audit Data Analysis and Mining (ADAM) χρησιμοποιεί tcpdump αναλυτή πακέτων για την δημιουργία προφίλ των κανόνων για την ταξινόμησή τους[14].

Τέλος, το 2003 από τους Δρ. Yongguang Zhang και Δρ. Wenke Lee υποστηρίζεται η τεράστια σημασία των συστημάτων ανίχνευσης εισβολών[15] σε υπολογιστικά συστήματα δικτύων και πιο συγκεκριμένα σε δίκτυα με κινούμενους κόμβους.

Daniel M. and Mansur, Doug, "DIDS (Distributed Intrusion Detection System) -- Motivation, Architecture, and An Early Prototype," The 14th National Computer Security Conference, October, 1991, pages 167–176.

[11] Jackson, Kathleen, DuBois, David H., and Stallings, Cathy A., "A Phased Approach to Network Intrusion Detection," 14th National Computing Security Conference, 1991.

[12] Paxson, Vern, "Bro: A System for Detecting Network Intruders in Real-Time," Proceedings of The 7th USENIX Security Symposium, San Antonio, TX, 1998.

[13] Kohlenberg, Toby (Ed.), Alder, Raven, Carter, Dr. Everett F. (Skip), Jr., Esler, Joel., Foster, James C., Jonkman Marty, Raffael, and Poor, Mike, "Snort IDS and IPS Toolkit," Syngress, 2007, ISBN 978-1-59749-099-3.

[14] Barbara, Daniel, Couto, Julia, Jajodia, Sushil, Popyack, Leonard, and Wu, Ningning, "ADAM: Detecting Intrusions by Data Mining," Proceedings of the IEEE Workshop on Information Assurance and Security, West Point, NY, June 5–6, 2001.

[15] Yongguang Zhang and Wenke Lee, "Intrusion detection in wireless adhoc networks", MobiCom '00 Proceedings of the 6th annual international conference on Mobile computing and networking, pp 275-28, 2000.

3.2 ΓΕΝΙΚΑ

Όπως προαναφέρθηκε ένα τείχος προστασίας εφόσον ρυθμιστεί σωστά αποτελεί την πρώτη γραμμή άμυνας ενός δικτύου. Παρ' όλη την ορθή λειτουργία του όμως, υπάρχει περίπτωση το τείχος προστασίας να μην ανταποκριθεί σε μία πιθανή ασυνήθιστη αίτηση. Έτσι υπάρχουν και άλλα μέτρα ασφάλειας που χρησιμοποιούνται στην πρώτη γραμμή άμυνας, όπως οι Λίστες Ελέγχου Πρόσβασης (Access Control Lists – ACLs). Ο συνδυασμός αυτών των δύο συμβάλλουν στην θωράκιση της ασφάλειας του δικτύου, χωρίς όμως να μας καθησυχάζει ότι είναι ο βέλτιστος και αποτελεσματικός τρόπος προστασίας. Γι' αυτό, θεωρείται απαραίτητη η ύπαρξη μίας δεύτερης γραμμής άμυνας.

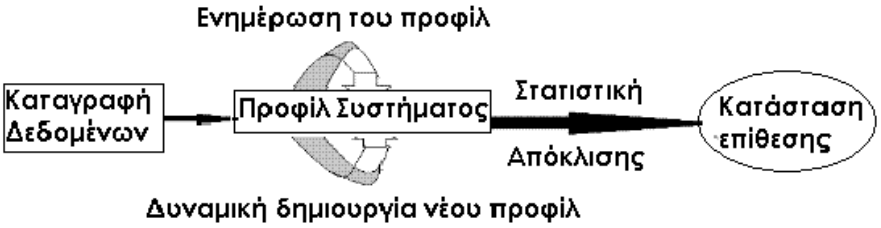
Αυτό το μοντέλο ασφάλειας ενός δικτύου, δηλαδή η χρήση πολλών επιπέδων προστασίας στο δίκτυο, ορίζεται ως «Defense of Depth» (Άμυνα σε βάθος). Στη δεύτερη γραμμή άμυνας, μία καλή λύση είναι η εφαρμογή ένας ανιχνευτικού συστήματος που θα αναγνωρίζει τυχόν ανεπιθύμητα στοιχεία, που εξαπάτησαν το τείχος προστασίας και εισήλθαν στο εσωτερικό του δικτύου.

3.3 ΤΡΟΠΟΙ ΛΕΙΤΟΥΡΓΙΑΣ ΤΩΝ IDS

Ένα σύστημα ανίχνευσης εισβολών είναι λογισμικό ή συνδυασμός λογισμικού και υλικού το οποίο πραγματοποιεί την άμεση ανίχνευση περίεργης δικτυακής κίνησης καθώς και την γρήγορη αντίδραση σ' ένα υπολογιστή ή στο δίκτυο. Εξ ορισμού ένα IDS λειτουργεί παθητικά, δηλαδή, δεν εφαρμόζει τεχνικές για να περιορίσει την εξάπλωση μίας επίθεσης, αλλά μένει μόνο στην αναγνώριση και την υπόδειξη επιθέσεων στον διαχειριστή ασφάλειας του δικτύου.

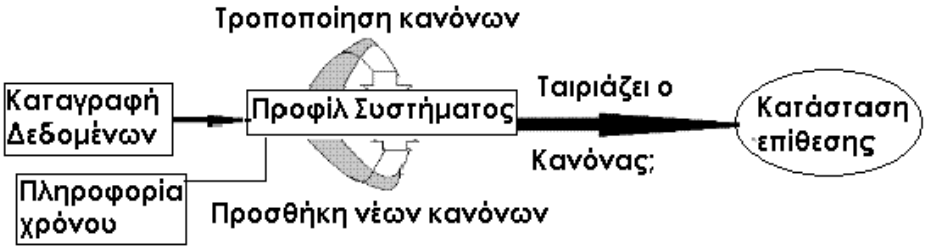
Τα συστήματα ανίχνευσης επιθέσεων χρησιμοποιούν για την ανίχνευση είτε τον τρόπο στατιστικής διαταραχών είτε την παρατήρηση κακής συμπεριφοράς. Η ανίχνευση με βάση στατιστική διαταραχών (statistical anomaly), καθιερώνει ένα «συνηθισμένο προφίλ δραστηριότητας» με

στατιστικό τρόπο. Ο τρόπος αυτός ανίχνευσης βασίζεται στο γεγονός ότι, όλες οι επιθετικές δραστηριότητες είναι ανωμαλίες και, πως, αν κάτι παρεκκλίνει από το σύνηθες προφίλ θεωρείται επίθεση.



Εικόνα 3 Ανίχνευση με βάση τη στατιστική διαταραχών (statistical anomaly)

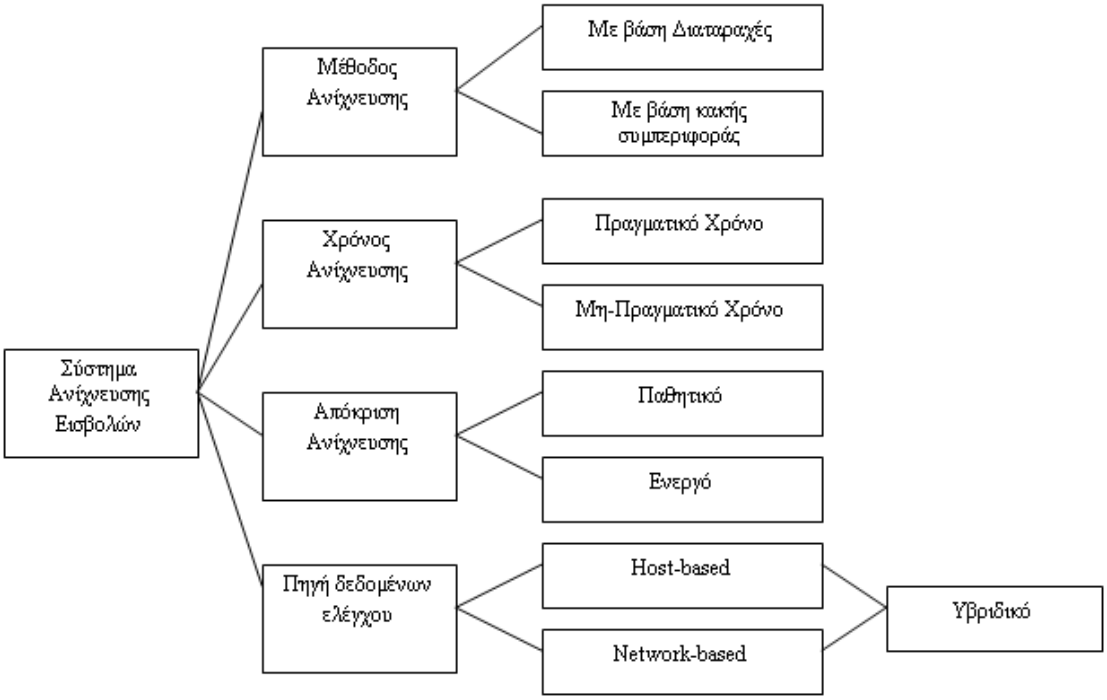
Η ανίχνευση κακής συμπεριφοράς (Misuse detection), από την άλλη, βασίζεται στο ότι υπάρχουν κάποια προκαθορισμένα σχέδια επίθεσης, ονομαζόμενα ως πρότυπα ή υπογραφές (signatures). Αν το σύστημα ανιχνεύσει κάποιο από τα πρότυπα, τότε συμπεραίνει ότι δέχεται επίθεση. Ο τρόπος αυτός είναι ο πιο σύνηθες στα περισσότερα IDS.



Εικόνα 4 Η ανίχνευση κακής συμπεριφοράς (Misuse detection)

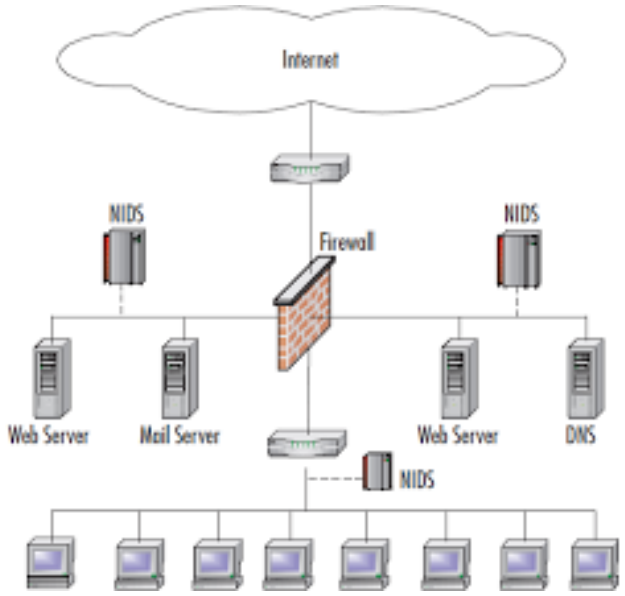
3.4 ΥΠΟΚΑΤΗΓΟΡΙΕΣ ΤΩΝ IDS

Τα συστήματα ανίχνευσης εισβολών σύμφωνα με τον τρόπο επιλογής τοποθέτησης-εγκατάστασης στο δίκτυο, χωρίζονται σε τρεις κατηγορίες. Αυτές παρουσιάζονται αναλυτικότερα παρακάτω:



Εικόνα 5 Κατηγοριοποίηση των IDS

1. ***Network-based IDS (NIDS)*** – Τα συγκεκριμένα συστήματα είναι τα πιο διαδεδομένα και παρακολουθούν την κίνηση συνολικά του δικτύου ή ένα κομμάτι του για ίχνη εισβολής. Συνήθως λειτουργούν με τον τρόπο ανίχνευσης κακής συμπεριφοράς.

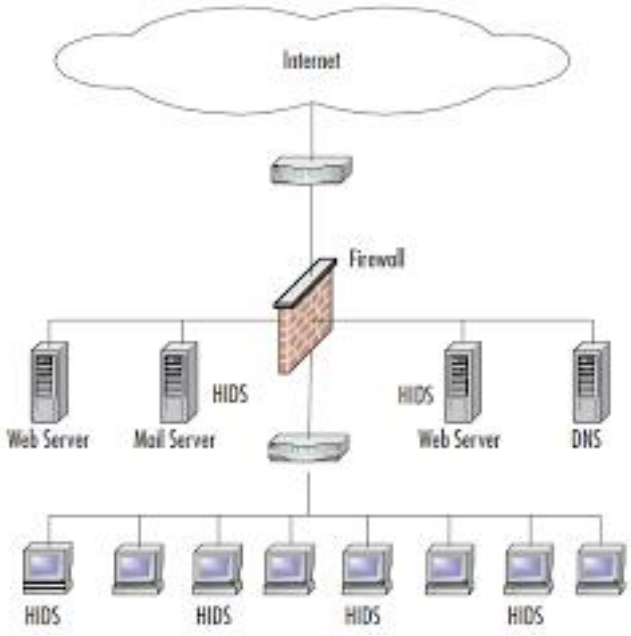


Εικόνα 6 Παράδειγμα ενός NIDS συστήματος

Αυτά έχουν την ιδιότητα να διατηρούνται καλύτερα απ' ότι τα host-based, διότι η εγκατάστασή τους σε μία τοποθεσία του δικτύου είναι απλή και δίνει την δυνατότητα να παρακολουθούν την κίνηση ευαίσθητων δεδομένων του δικτύου, χωρίς εξουσιοδότηση ή κάποιου είδους πρόσβασης, με συνέπεια την κατάχρηση προνομίων εξουσιοδότησης. Ένα άλλο στοιχείο που χαρακτηρίζει τα network-based συστήματα είναι πως η εγκατάστασή τους δεν απαιτεί μετατροπές στους εξυπηρετητές (servers) ή στους κόμβους του δικτύου. Συνήθως οι εξυπηρετητές έχουν μικρές δυνατότητες όσο αφορά την CPU και την χωρητικότητα του δίσκου. Επομένως, μία πιθανή μετατροπή θα επέφερε πιθανά προβλήματα λειτουργικότητας στο δίκτυο. Για τον λόγο αυτό, η τοποθέτηση των NIDS δεν επηρεάζει το δίκτυο, καθώς δεν αποτελεί κρίσιμο παράγοντα για την λειτουργικότητά του, όπως οι δρομολογητές ή κάποια άλλη δικτυακή συσκευή.

Τα συστήματα NIDS παρουσιάζουν και κάποια μειονεκτήματα, όπως ότι δεν έχουν την δυνατότητα ανίχνευσης μίας επίθεσης που γίνεται σε διαφορετικό δίκτυο εκτός αυτού που προστατεύουν, επιφέροντας μεγάλο κόστος αγοράς για την εγκατάστασή τους σ' ένα περιβάλλον με πολλαπλές δικτυώσεις Ethernet. Η συνδεσμολογία ενός σύγχρονου δικτύου βασίζεται σε μεταγωγείς (switches) με αποτέλεσμα τα NIDS να μην μπορούν να παρακολουθήσουν όλα τα πακέτα που κινούνται στο δίκτυο. Υπάρχουν βέβαια switch με δυνατότητα συγκέντρωσης της κίνησης του δικτύου σε μία θύρα SPAN, όπου κι εκεί εντοπίζονται προβλήματα. Τέλος, τα network-based δεν μπορούν να διαχειριστούν εύκολα επιθέσεις με κρυπτογραφημένες πληροφορίες. Βέβαια είναι λίγες οι περιπτώσεις τέτοιων επιθέσεων, στις μέρες μας, εκτός των επιθέσεων σε web servers, διότι αυτή την στιγμή χρησιμοποιούνται οι IP διευθύνσεις έκδοσης τέσσερα (IPv4) από τους διαχειριστές για την επικοινωνία των συστημάτων δικτύου και το πρόβλημα αναμένεται να γίνει πιο εμφανές με την μετάβαση στο IPv6.

2. ***Host-based IDS (HIDS)*** – Είναι συστήματα που εξετάζουν κάθε υπολογιστή (host) του δικτύου που είναι εγκατεστημένα, ξεχωριστά. Πιο συγκεκριμένα, λειτουργούν παρακολουθώντας για ασυνήθη δραστηριότητα που περιορίζεται στον τοπικό υπολογιστή, όπως παράξενη πρόσβαση στα αρχεία ή μετατροπές σε δικαιώματα του συστήματος. Επειδή υπάρχει κάθε χρονική στιγμή εικόνα των συμβάντων του συστήματος θεωρούνται πιο ισχυρά από τις άλλες κατηγορίες IDS. Έτσι αν γίνει μία δραστηριότητα που έχει σκοπό την δυσλειτουργία του υπολογιστή, τότε το σύστημα ανίχνευσης θα το αναγνώριζε ως επίθεση. Σε αντίστοιχη περίπτωση, σ' ένα Network-IDS θα είχε περάσει απαρατήρητο από το σύστημα ανίχνευσης.



Εικόνα 7 Παράδειγμα ενός HIDS συστήματος

Τα host-based έχουν μειωμένο ρυθμό λανθασμένων ειδοποιήσεων (false positive) συγκριτικά με τα NIDS, γιατί το εύρος των εντολών που εκτελούν είναι πολύ μικρότερο, παρά τα είδη των πακέτων που κινούνται στο δίκτυο, με αποτέλεσμα να ελαττώνεται η πολυπλοκότητα των συστημάτων αυτών. Τέλος, δεν επηρεάζονται από κρυπτογραφημένα πακέτα ή συνδεσμολογίες με βάση switch, διότι τα πακέτα επεξεργάζονται μετά την αποκρυπτογράφησή τους και αφού εισέλθουν στο σύστημα.

Υπάρχουν όμως και κάποιες αδυναμίες των συστημάτων host-based, όπως είναι η απαίτηση εγκατάστασής τους σε κάθε συσκευή που επιθυμείται η προστασία της, δηλαδή αν η συσκευή είναι ένας server του δικτύου, τότε όπως αναφέραμε και παραπάνω, θα προκαλέσει

προβλήματα χωρητικότητας η εγκατάσταση του λογισμικού IDS ή προβλήματα ασφάλειας αφού μπορεί να μην είναι δυνατή η πρόσβαση του προσωπικού στο σημείο που είναι ο server όταν χρειαστεί. Ακόμα, είναι δυσκολότερα στην διαχείρισή τους λόγω της κατανεμημένης φύσης τους. Τέλος, τα συστήματα HIDS αγνοούν εντελώς το περιβάλλον του δικτύου, πράγμα που τα καθιστά ευάλωτα σε επιθέσεις, γιατί ο χρόνος ανάλυσης των βλαβών αυξάνει γραμμικά με τον αριθμό των υπολογιστών που προστατεύονται στο δίκτυο. Άρα, αν χρειάζεται t χρόνος για την ανάλυση ενός συμβάντος στο σύστημα, θα χρειαστεί 2t για δύο συστήματα, 3t για τρία κοκ.

3. ***Υβριδικά IDS (Hybrid IDS)*** – Η κατηγορία αυτή των συστημάτων ανίχνευσης χρησιμοποιεί χαρακτηριστικά και των δύο προηγούμενων κατηγοριών. Λειτουργεί με τον τρόπο των NIDS, με την διαφορά πως εξετάζουν μόνο την κίνηση του εκάστοτε κόμβου. Αυτό έχει ως αποτέλεσμα την αντιμετώπιση πολλών προβλημάτων των NIDS, όπως είναι η σωστή λειτουργία κάτω από μεγάλο όγκο εξέτασης πακέτων, σε συστήματα με κρυπτογραφημένες επικοινωνίες (VPNs) και switched δίκτυα.

3.5 ΠΡΩΤΟΚΟΛΛΑ ΠΟΥ ΑΝΙΧΝΕΥΟΥΝ ΤΑ IDS

Καθώς τα συστήματα ανίχνευσης εισβολών εξαπλώνονται υπάρχουν περιπτώσεις όπου παράγονται ειδοποιήσεις (alerts) για συγκεκριμένες διευκρινήσεις (scans) και επιθέσεις, ασυνήθιστων IP πακέτων, μη μπορώντας ο διαχειριστής του συστήματος του δικτύου να καταλάβει την σημασία τους. Ως ασυνήθιστα πακέτα ορίζουμε εκείνα που παραβιάζουν τα IP protocol standards όπως αυτά ορίζονται από σύνολο των προδιαγραφών RFC

(Requests for comments)[16]. Είναι πακέτα, όπου σκοπός τους είναι να καταφέρουν να περάσουν τους ελέγχους των τειχών προστασίας ή τα συστήματα ανίχνευσης εισβολών ενός δικτύου, με αποτέλεσμα την κακόβουλη δράση τους εντός αυτού. Υπάρχουν πολλά είδη IP πρωτοκόλλων, τρία εκ των οποίων είναι τα πιο συνηθισμένα σε ένα δίκτυο και τα οποία ελέγχουν σίγουρα τα IDS: το Transmission Control Protocol (TCP), το User Datagram (UDP) και το Internet Message (ICMP). Υπάρχουν και άλλα πολλά όπως είναι τα IGRP, EIGRP, OSPF κτλ. Κάθε ένα από αυτά έχει τον δικό του κωδικό αριθμό (Internet protocol number), όπου χρησιμοποιούν για την αναγνώρισή τους κάποια συστήματα ανίχνευσης.

3.5.1 TCP (Transmission Control Protocol)

Είναι ένα πρωτόκολλο που βασίζεται στις αμφίδρομες συνδέσεις, όπως αναφέρθηκε περιληπτικά και στο προηγούμενο κεφάλαιο. Στην επικεφαλίδα του πρωτοκόλλου υπάρχουν κάποιες σημαίες (flags) για να δείχνουν την κατάσταση μίας σύνδεσης. Δηλαδή, αν μία σύνδεση έχει αρχίσει ή τελείωσε, αν τα δεδομένα είναι υψηλής προτεραιότητας, κ.α. Πολλές από τις επιθέσεις εκμεταλλεύονται αυτό το πεδίο, αλλοιώνοντας τις πληροφορίες των flags, για να μην γίνουν αντιληπτές από το firewall ή το IDS ενός συστήματος, καθώς και για να προσδιορίσουν το είδος του λειτουργικού που χρησιμοποιεί μία συσκευή ή ένα σύστημα. Οι προδιαγραφές του πρωτοκόλλου αυτού ορίζονται στο RFC 793. Τα flags του TCP είναι έξι:

1. URG (Urget) – Προκαλείται από ένα διακόπτη και είναι η ένδειξη άμεσης προτεραιότητας.
2. ACK (Acknowledgment) – Ορίζεται η εγκυρότητα της σύνδεσης.
3. PSH (Push) – Ενημερώνει τον δέκτη του πακέτου να ολοκληρώσει με τα δεδομένα το δυνατόν γρηγορότερα.

[16] Θόδωρος Κομνηνός – Παύλος Σπυράκης, Ασφάλεια Δικτύων και Υπολογιστικών Συστημάτων Αναχαιτίστε τους εισβολείς, Εκδόσεις Ελληνικά Γράμματα, Αθήνα 2002, σελίδα 205.

4. RST (Reset) – Απότομη λήξη μιας σύνδεσης.
5. SYN (Synchronization) – Εκκίνηση μίας σύνδεσης TCP.
6. FIN (Finish) – Ομαλή λήξη μιας σύνδεσης.

Υπάρχουν αρκετοί συνδυασμοί των παραπάνω flags όπου χρησιμοποιούνται άλλοτε για τις κανονικές(νόμιμες) λειτουργίες του πρωτοκόλλου, όπως ο '**SYN, SYN ACK, ACK**' που χρησιμοποιείται κατά την διαδικασία three-way handshake για την σύνδεση μίας TCP συνεδρίας. Άλλος συνδυασμός είναι ο '**FIN ACK ACK**' όπου γίνεται χρήση σε μία σύνοδο λήξης μιας υπάρχουσας σύνδεσης. Ακόμα είναι ο '**RST ACK**' συνδυασμός που χρησιμοποιείται για την άμεση διακοπή μίας σύνδεσης.

Υπάρχουν όμως και άλλοι συνδυασμοί όπου χρησιμοποιούνται από κακόβουλους χρήστες, ορίζονται ως ανώμαλα (flags) και είναι γνωστοί στον τομέα ασφάλειας δικτύων. Ενδεικτικά, ο '**SYN FIN'**, είναι ο πιο γνωστός παράνομος συνδυασμός. Χρησιμοποιείται από πολλά εργαλεία σάρωσης θυρών, γιατί πολλά συστήματα ανίχνευσης εισβολών αδυνατούσαν να τον εντοπίσουν στο παρελθόν. Τα σημερινά IDS θεωρούν δεδομένη απειλή όταν εντοπίζουν τον συγκεκριμένο συνδυασμό flags. Έτσι δημιουργήθηκαν παραλλαγές του '**SYN FIN'**, όπως είναι οι '**SYN FIN PSH'**, '**SYN FIN RST'** και '**SYN FIN RST PSH'**, όπου χρησιμοποιούνται από τους εισβολείς επωφελούμενοι από το γεγονός ότι, τα συστήματα ανίχνευσης για την βελτίωση του χρόνου απόκρισης ψάχνουν για πακέτα μόνο με '**SYN FIN'** bit να έχουν ενεργοποιημένο, μη δίνοντας σημασία στα υπόλοιπα flags. Ακόμα ένας ανώμαλος συνδυασμός είναι η απενεργοποίηση όλων των flags. Είναι παράνομη η μορφή αυτών των άδειων «null» πακέτων. Επίσης. είναι ύποπτη η χρήση πακέτων που έχουν μόνο ένα FIN flag, γιατί χρησιμοποιούνται για σάρωση θυρών, χαρτογράφηση δικτύων και άλλες δραστηριότητες. Εκτός από τα flags όμως, υπάρχουν και δύο επιπλέον bits στα πακέτα του πρωτοκόλλου αυτού, για μελλοντική χρήση. Αυτά ονομάζονται «δεσμευμένα bits» και πρέπει να είναι πάντα απενεργοποιημένα. Τέλος, μπορούν να ανιχνεύσουν ένα ύποπτο πακέτο TCP, όταν διαπιστωθεί πως η θύρα (port) της

πηγής ή του προορισμού είναι μηδέν ή ο αριθμός βεβαίωσης λήψης είναι διάφορος του μηδενός όταν το ACK flag είναι ενεργοποιημένο.

3.5.2 UDP (User Datagram Protocol)

Το πρωτόκολλο αυτό δεν βασίζεται στην αμφίδρομη σύνδεση μεταξύ δύο κόμβων, όπως το TCP, καθώς δεν υπάρχουν τα flags και τα δεσμευμένα bits. Το κοινό τους σημείο είναι πως βασίζονται στα ports της πηγής και του παραλήπτη. Έτσι, και στο πρωτόκολλο αυτό, όπως και στο TCP, δεν πρέπει να υπάρχει port με τιμή μηδενική. Οι σύνηθες επιθέσεις που δέχονται τα UDP πακέτα χρησιμοποιούν την τεχνική του θρυμματισμού (fragmentation), δηλαδή τον χωρισμό ενός πακέτου σε μικρότερα τμήματα-κομμάτια (fragment).

3.5.3 ICMP (Internet Control Message Protocol)

Χρησιμοποιείται για την μεταβίβαση ενός μηνύματος λάθους μεταξύ δύο υπολογιστικών συστημάτων ή από ένα σύστημα σε μία δικτυακή συσκευή, όπως ένα δρομολογητή. Το συγκεκριμένο πρωτόκολλο δεν έχει αριθμούς θυρών (ports), γι' αυτό και κάνει χρήση τύπων μηνυμάτων και κωδικών. Αξίζει να σημειωθεί πως χρησιμοποιείται συνδυαστικά με άλλα πρωτόκολλα, όπως το UDP για την μεταφορά μηνυμάτων λάθους. Υποστηρίζει επίσης κίνηση εκπομπής. Λόγω της απλότητάς του ICMP είναι ευάλωτο. Τα περισσότερα πακέτα αυτού αποτελούνται από ένα header και την πληροφορία. Υπάρχουν όμως και ICMP όπως τα echo request που δεν περιέχουν καμία πληροφορία. Αυτό το εκμεταλλεύονται κάποιες εφαρμογές, όπως τα προγράμματα denial of service και tunneling, για την απόκρυφη μεταφορά δεδομένων που θα περάσουν απαρατήρητα από το δίκτυο. Τα IDS πρέπει να αντιλαμβάνονται από το μέγεθος τις περισσότερες φορές πως είναι ύποπτα τα πακέτα ICMP αυτά.

Snort IDS

4. ΤΟ ΛΟΓΙΣΜΙΚΟ SNORT

4.1 ΓΕΝΙΚΑ

Το Snort είναι ένα δωρεάν λογισμικό ανοιχτού κώδικα (Open Source), αποτροπής εισβολών (network intrusion prevention system -NIPS)και ανίχνευσης εισβολών (network intrusion detection system -NIDS) σε δίκτυα. Μέχρι την έκδοση 1.9 χρησιμοποιούνταν σε δίκτυα μικρού μεγέθους και με μικρό εύρος ζώνης(bandwidth), δηλαδή μέχρι της τάξης των 100Mbps. Από την έκδοση 2.0 και έπειτα άλλαξε ο μηχανισμός εντοπισμού (detection engine) με την νέα «Hi-performance Multi-rule inspection engine» τεχνική και έτσι το Snort μπορεί να χρησιμοποιείται σε δίκτυα με εύρος ζώνης της τάξης Gigabit.

Δημιουργήθηκε από τον Marty Roesch το 1998 και σήμερα έχει περάσει στην εταιρία Sourcefire, όπου ο Roesch είναι ο ιδρυτής και προϊστάμενος στο τμήμα της τεχνολογικής ανάπτυξης της εταιρείας. Το 2009 αναρτήθηκε στο InfoWorld's Open Source Hall of Fame ως μία από τις «μεγαλύτερες σε τμήματα λογισμικού ανοιχτού πηγαίου κώδικα όλων των εποχών». Ο κώδικας είναι γραμμένος στην γλώσσα προγραμματισμού C και τρέχει σε όλα σχεδόν τα λειτουργικά συστήματα υπολογιστών (Cross-platform).

4.2 ΛΟΓΟΙ ΕΠΙΛΟΓΗΣ ΤΟΥ SNORT

Υπάρχει πληθώρα πακέτων λογισμικών συστημάτων ανίχνευσης ή αποτροπής εισβολών σε δίκτυα. Ο λόγος που επιλέχτηκε το Snort είναι για την ικανότητά του να εκτελεί σε πραγματικό χρόνο ανάλυση της κίνησης και την καταγραφή πακέτων σε Internet protocol (IP) δίκτυα. Ικανότητα ανίχνευσης μεγάλου εύρους επιθέσεων όπως buffer overflows, σάρωση θυρών (port scans), επιθέσεις που εκμεταλλεύονται σφάλματα των λειτουργικών συστημάτων, αδυναμίες των CGI κ.α. Δεν είναι απαιτητικό σε πόρους για την λειτουργία του. Επίσης είναι εύκολα διαμορφώσιμο και ευέλικτο ανάλογα με τις εκάστοτε ανάγκες του δικτύου, διότι όλα τα αρχεία διαμόρφωσης αλλά και των κανόνων που χρειάζονται για την ρύθμιση των παραμέτρων είναι στη διαθεσιμότητα του χρήστη/διαχειριστή του δικτύου.

Η εταιρεία προβαίνει σε συχνές αναβαθμίσεις του λογισμικού, έτσι μπορεί να ενημερώνεται για διορθώσεις ή προσθήκες χαρακτηριστικών όπως και μέσω της ιστοσελίδας του για νέες υπογραφές επιθέσεων. Δίνει την δυνατότητα στον χρήστη να δημιουργεί τους δικούς του κανόνες(rules) και να αλλάζει την βάση μέσα από την λειτουργία plug-ins, δηλαδή κώδικας που προαιρετικά εμπεριέχεται κατά την εγκατάσταση του λογισμικού και προσφέρει δυνατότητες όπως η ενεργός ανταπόκριση σε κακόβουλη κίνηση (malicious traffic). Τέλος, είναι φιλικό στην χρήση του σε σύγκριση με άλλα ανοικτού κώδικα λογισμικά, και λόγω της δωρεάν διανομής του υπό την άδεια της GNU GPL, είναι αρκετά διαδεδομένο στο χώρο με αποτέλεσμα την ύπαρξη ικανοποιητικού υλικού τεκμηρίωσης για την εγκατάσταση καθώς και για την λειτουργία του.

4.3 ΤΡΟΠΟΙ ΛΕΙΤΟΥΡΓΙΑΣ

Το Snort εκτός από την λειτουργία του σαν ανιχνευτής εισβολών μπορεί να ρυθμιστεί και σε άλλες λειτουργίες που περιγράφονται παρακάτω:

- Sniffer mode (Αναλυτής κίνησης δικτύου), το πρόγραμμα διαβάζει τα πακέτα του δικτύου και τα εμφανίζει στην κονσόλα (οθόνη) σε φιλική μορφή προς τον χρήστη, δηλαδή εκτελεί μία απλή καταγραφή κίνησης του δικτύου. Με διάφορα φίλτρα (Berkley packet filter-BPF) που μπορεί να χρησιμοποιηθούν από τον χρήστη, δίνεται η δυνατότητα να οριστεί το είδος των πακέτων που θα εμφανίζονται ως προς το πρωτόκολλο, τον αποστολέα, τον παραλήπτη και διάφορα άλλα χαρακτηριστικά ενός πακέτου. Δηλαδή, αν ο χρήστης πληκτρολογήσει την λέξη κλειδί icmp στο snort τότε θα εμφανίζονται μόνο τα πακέτα αυτού του πρωτόκολλου.
- Packet logger mode (Καταγραφικό πακέτων), το πρόγραμμα καταγράφει τα πακέτα που διαβάζει από το δίκτυο στο δίσκο, αντί να τα εμφανίζει απλά στην οθόνη. Αυτή η λειτουργία βοηθά σε περιπτώσεις όπου απαιτείται η λεπτομερής εξέταση των πακέτων που αναγιγνώσκονται. Το Snort μπορεί ν' αποθηκεύει τα πακέτα σε διάφορες μορφές, όπως για παράδειγμα σε binary μορφή (tcpdump format) με την οποία μπορούν να χρησιμοποιηθούν σαν είσοδο σε διάφορα άλλα προγράμματα ανάλυσης πακέτων και πρωτοκόλλων, σε ASCII μορφή ώστε να είναι δυνατή η ανάγνωσή τους, σε XML μορφή ή και να οργανωθούν σε βάσεις δεδομένων. Η συγκεκριμένη λειτουργία, δεν λειτουργεί συνήθως ανεξάρτητα από τις λειτουργίες του sniffer ή την NIDS, αλλά παράλληλα με αυτά.
- Network Intrusion Detection System – NIDS mode (ανίχνευση εισβολής σε δίκτυο), συγκρίνει την κίνηση του δικτύου μ' ένα προκαθορισμένο σύνολο υπογραφών που είναι γνωστές ως κανόνες, όπου ορίζονται από τον χρήστη και εκτελεί διάφορες ενέργειες με βάση ότι έχει εντοπίσει. Είναι η πιο κοινή λειτουργία (common mode) που τρέχει το Snort και χρειάζεται και στο εργαστηριακό κομμάτι της εργασίας αυτής. Συνήθως εκτελείται από

την γραμμή εντολών (command line) σε κάθε λειτουργικό σύστημα (Unix ή Windows). Υπάρχουν βέβαια λογισμικά που προσφέρουν γραφικό περιβάλλον, όπως είναι το IDScenter στα windows και το Demarc Puresecure για windows και για Unix. Τα λογισμικά αυτά όμως δεν θα μπορέσουν να χρησιμοποιηθούν, διότι είναι συμβατά για παλαιότερες εκδόσεις των λειτουργικών συστημάτων. Η τεχνική που χρησιμοποιεί το Snort για την διαδικασία ανίχνευσης εισβολών είναι κατά κύριο λόγο η μέθοδος ανίχνευσης κακής συμπεριφοράς (Misuse Detection) με την χρήση των υπογραφών(Signatures) ενός βλαβερού(malicious) πακέτου. Το snort όμως ειδικά μετά την έκδοση 2.0 συνδυάζει στην λειτουργία της ανάλυσης των γεγονότων για την ανίχνευση πιθανών επιθέσεων και κάποιες από τις μεθόδους του πρωτόκολλου ανίχνευσης διαταραχών (Protocol Anomaly Detection) και του πρωτοκόλλου κακής συμπεριφοράς (Misuse Detection). Οι μηχανισμοί αυτοί υλοποιούνται κατά κύριο λόγο από τους προεπεξεργαστές (preprocessors) που εξηγούνται αναλυτικά παρακάτω, αλλά και από το νέο μηχανισμό του snort 2.0 που συντάσσει τους κανόνες (rules) σε κατηγορίες.

4.4 ΚΑΝΟΝΕΣ (RULES) Ή ΥΠΟΓΡΑΦΕΣ (SIGNATURES) ΤΩΝ IDS

4.4.1 ΒΑΣΙΚΕΣ ΔΙΑΦΟΡΕΣ

Οι κανόνες (rules) και οι υπογραφές (signatures) είναι ισοδύναμα σαν έννοιες και συνήθως χρησιμοποιούνται σαν συνώνυμες λέξεις. Υπάρχουν όμως κάποιες διαφορές[16] μεταξύ τους, που είναι οι εξής:

- Υπογραφές (signatures), είναι τα ειδικά χαρακτηρίστηκα του πακέτου που το χαρακτηρίζουν σαν ύποπτο ή βλαβερό(malicious). Τα χαρακτηριστικά αυτά είναι στο payload ή την επικεφαλίδα (header) του πακέτου και είναι

[16]Δημήτρης Πρίτσος, ISLAB HACK: Βασικές Έννοιες & Προγραμματισμός του Snort 2.0, Αθήνα 2003 σελίδα 5.

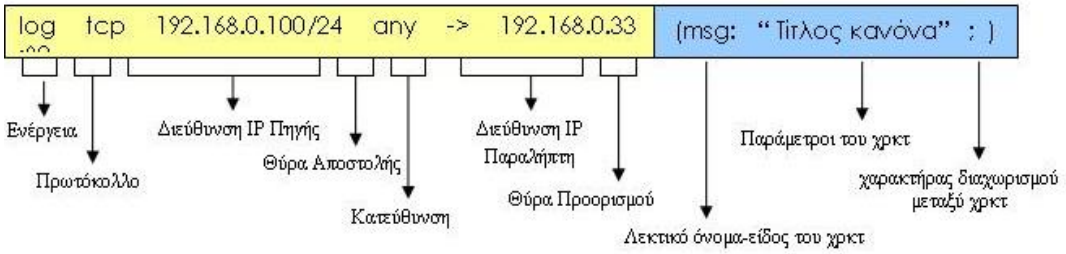
πρότυπα (patterns) από συμβολοσειρές (strings) που χαρακτηρίζονται σαν υπογραφή (Signature) ενός «κακού» πακέτου. Γενικά η περιγραφή ενός πακέτου που είναι malicious όταν γίνεται με ένα signature είναι στατική. Δηλαδή, ένα signature περιγράφει ένα υπαρκτό χαρακτηριστικό (positive pattern much) στο payload και μερικά χαρακτηριστικά στην επικεφαλίδα (header)του πακέτου.

- Κανόνες (rules), είναι κανόνες οι οποίοι περιγράφουν στο snort ή άλλο IDS τα χαρακτηριστικά ενός πακέτου που μπορεί να είναι μέρος μίας γνωστής επίθεσης καθώς και την ενέργεια που θα εκτελεστεί κατά τον εντοπισμό του. Η περιγραφή ενός πακέτου με ένα rule είναι αρκετά πιο δυναμική. Αφενός σε ένα rule μπορεί να περιγράφονται περισσότερα του ενός υπαρκτά χαρακτηριστικά στο payload αφετέρου μπορούν να περιγράφονται χαρακτηριστικά που δεν πρέπει να έχει ένα πακέτο για να θεωρηθεί ύποπτο. Τέλος ένα rule μπορεί να περιγράφει ένα ολόκληρο stream και όχι ένα πακέτο μόνο για τις περιπτώσεις που γίνεται statefull intrusion detection.

4.5 ΟΙ ΚΑΝΟΝΕΣ (RULES)

Οι κανόνες είναι ένα από το σημαντικότερο τμήμα των συστημάτων ανίχνευσης εισβολών. Όπως αναφέραμε και στον προσδιορισμό της έννοιας παραπάνω, πρόκειται για ένα πρότυπο με το οποίο γίνεται αναζήτηση στα διακινούμενα πακέτα του δικτύου. Με την εύρεση κάποιου πακέτου που φέρει χαρακτηριστικά ίδια με αυτά του πρότυπου, θεωρείται από το λογισμικό ως επίθεση. Έτσι μπορεί να ανιχνεύει μία προσπάθεια σύνδεσης από μία IP «ύποπτη», τον αντικανονικό συνδυασμό των πακέτων TCP κάνοντας ένα έλεγχο της διεύθυνσης της πηγής. Επίσης μπορεί να ανιχνεύει, μία προσπάθεια επίθεσης με Denial of Service χρησιμοποιώντας την μέθοδο της πολλαπλής αποστολής ίδιας εντολής, κάτι που αντιμετωπίζεται με τον έλεγχο του αριθμού που μία εντολή εκτελείται και να παράγεται ειδοποίηση όταν ξεπεραστεί το όριο που έχει οριστεί. Ακόμα μπορεί να ανιχνεύει μία επίθεση σε ένα FTP server,

δημιουργώντας μία υπογραφή που θα στηρίζεται σε διαδοχή καταστάσεων (stage tracking), ειδοποιώντας όταν κάποιος προσπαθήσει να κάνει κάποια κίνηση χωρίς να έχει περάσει από την απαιτούμενη διαδικασία. Δημιουργώντας τον κατάλληλο κανόνα μπορεί να ανιχνεύει ένα email με ιό, απλώς ελέγχοντας το όνομα του θέματος ή των συνημμένων αρχείων. Τέλος παρατηρείται από τα παραπάνω παραδείγματα υπογραφών και τους τρόπους που αυτές λειτουργούν για την ανίχνευση των επιθέσεων, πως μπορούν να είναι από αρκετά απλές, ελέγχοντας κάποιο πεδίο των πακέτων, ως σύνθετες όπου αναλύουν την σύνδεση με βάση το χρησιμοποιούμενο πρωτόκολλο. Σύμφωνα με την εικόνα που εμφανίζεται, θα κάνουμε μία σύντομη ανάλυση ενός κανόνα (rule).



Εικόνα 8 Κανόνας

Οι Κανόνες(rules) του Snort μπορούν να γραφτούν σε απλή περιγραφική γλώσσα σε ASCII μορφή και κάθε ένα από αυτά αποτελείται από δύο λογικά μέρη, τον Rule Header (επικεφαλίδα κανόνα) και τα Rule Options (ιδιότητες/χαρακτηριστικά κανόνα). Ο Rule Header περιέχει τις εξής πληροφορίες:

- **Action**: Είναι η ενέργεια που θα εκτελέσει το Snort όταν ταιριάξει κάποιο πακέτο με ένα *Rule*. Η ενέργεια αυτή έχει να κάνει με την αντίδραση (Response) του Snort κατά την ανίχνευση μίας πιθανής επίθεσης. Η ενέργεια (action) μπορεί να είναι :
    - ***Alert***, η οποία θα δημιουργήσει μία ειδοποίηση για το γεγονός που εντόπισε και στη συνέχεια θα καταγράψει το πακέτο. Οι ειδοποιήσεις είναι ο τρόπος με τον οποίο το Snort επισημαίνει το γεγονός της ανίχνευσης μίας επίθεσης.

- **Log**, η οποία θα καταγράψει το πακέτο στον δίσκο.
- ***Pass***, η οποία θα αγνοήσει το πακέτο.
- ***Activate***, η οποία θα προκαλέσει μία ειδοποίηση και στη συνέχεια θα ενεργοποιήσει ένα δυναμικό κανόνα(*dynamic Rule)*.
- ***Dynamic***, η οποία θα περιμένει μέχρι να ενεργοποιηθεί από ένα activate *Rule* και στη συνέχεια θα ενεργήσει σαν ένα *log Rule*. Ο χρήστης έχει την δυνατότητα να ορίσει και δικούς του τύπους από ενέργειες (actions).

- **Protocol**: Είναι το είδος του πρωτοκόλλου στο οποίο ανήκει το πακέτο που θα εξεταστεί. Το πρωτόκολλο μπορεί να είναι ip, tcp, icmp, udp.
- **Source IP**: Είναι η IP διεύθυνση αποστολέα που βρίσκεται στον IP header του πακέτου, σε συνδυασμό με την μάσκα του δικτύου (netmask) στο οποίο μπορεί να ανήκει, εκφρασμένη με CIDR τρόπο γραφής. Με τον CIDR τρόπο γραφής γίνεται δυνατό να ορισθεί μία ομάδα (block) από συνεχείς IP διευθύνσεις.
- **Source Port**: Είναι η πόρτα αποστολής που έχει νόημα στα tcp και udp πακέτα. Στο συγκεκριμένο παράδειγμα με το λεκτικό any εννοείται οποιαδήποτε πόρτα.
- **Destination IP**: Είναι η IP διεύθυνση του παραλήπτη του πακέτου. Ο τρόπος γραφής της είναι ο ίδιος με αυτόν που ισχύει για την Source IP και στο συγκεκριμένο παράδειγμα η IP διεύθυνση του παραλήπτη με την netmask που έχει ορισθεί, αντιπροσωπεύει έναν μόνο αριθμό τον 192.168.0.33.
- **Destination Port**: Είναι η πόρτα προορισμού του πακέτου.

Τα χαρακτηριστικά των κανόνων(Rule Options) περιέχουν πληροφορία που αναφέρεται στα χαρακτηριστικά για τα οποία θα ελεγχθεί το πακέτο. Επίσης στα Rule Options μπορούν να ορισθούν και κάποιες επιπρόσθετες ενέργειες που θα εκτελεστούν για κάποιο πακέτο που θα ταιριάξει με τους κανόνες. Το

κάθε option που περιέχεται στο Rule Options τμήμα του κανόνα, αποτελείται από τα Option Keyword και τα Option Arguments.

i. **Option Keyword**: Είναι το λεκτικό που υποδηλώνει το όνομα-είδος του option. Το λεκτικό από την έκδοση snort 2.0 και μετά έχει την δυνατότητα να εκφραστεί σαν Regular Expression του UNIX. Αυτό δίνει την δυνατότητα να περιγραφούν οι κανόνες με μία γενική μορφή ώστε να μπορούν να περιγράψουν γενικές επιπτώσεις επιθέσεων, όπως το Buffer Overflow. Για παράδειγμα, στην τελευταία περίπτωση, μπορεί να περιγράφει ένας κανόνας με τέτοια μορφή ώστε όταν συναντάτε ένας μεγάλος αριθμός από NOPs σε ένα πακέτο, αυτό να θεωρείται σαν ύποπτο για buffer overflow exploit. Έτσι, δίνεται η δυνατότητα να εκφραστεί, μέσα από ένα κανόνα, ένας τρόπος ανάλυσης βασισμένος στις ανωμαλίες (Anomaly based) ενός πακέτου ή stream. Χαρακτηριστική είναι η περίπτωση του Buffer Overflow, το οποίο παραμένει αρκετά δύσκολο στον εντοπισμό του με αυτό τον τρόπο, και είναι μία πολύ χρήσιμη εναλλακτική για να πιάνονται όλες οι παραλλαγές ενός είδη γνωστού Buffer Overflow Exploit.
ii. **Option Argument**: Είναι οι παράμετροι που δέχεται το option σε σχέση με τις οποίες θα ελεγχθεί το πακέτο. Τα Option Arguments για κάθε option, πρέπει να διαχωρίζονται με τον χαρακτήρα '**:**' από το αντίστοιχο Option Keyword.
iii. **Option Separator**: Είναι ο χαρακτήρας διαχωρισμού '**;**' μεταξύ δύο χαρακτηριστικών.

Ο συγκεκριμένος κανόνας στην **Εικόνα 8** έχει ένα χαρακτηριστικό. Αυτό είναι το όνομα msg, το οποίο δηλώνει ότι αν ταιριάξει κάποιο πακέτο με αυτό τον κανόνα, τότε μαζί με την ειδοποίηση (ή το πακέτο που θα καταγραφεί) θα τυπωθεί και κάποιο μήνυμα, το οποίο είναι αυτό που ακολουθεί μέσα στα εισαγωγικά και το οποίο αποτελεί το Option Argument αυτού του χαρακτηριστικού.

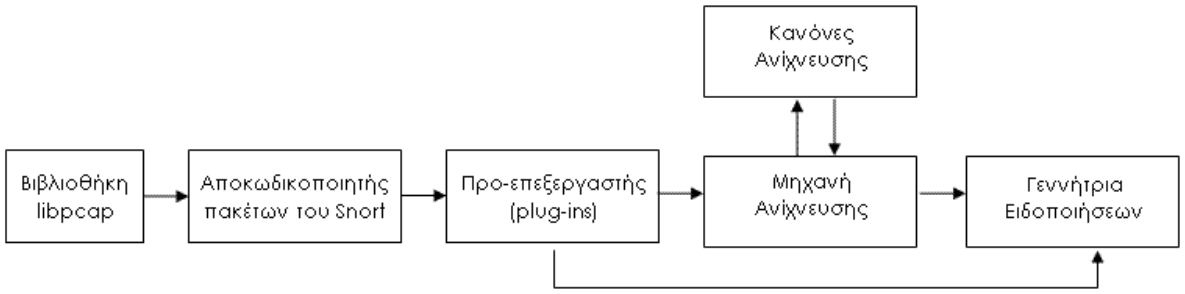
Το Snort διανέμεται με πάνω από 2500 έτοιμους κανόνες, για χρήση στην ανίχνευση γνωστών επιθέσεων, ενώ για την δημιουργία νέων κανόνων προσφέρει μία μεγάλη γκάμα από χαρακτηριστικά που μπορεί ο χρήστης να χρησιμοποιήσει, τα οποία του δίνουν την ευελιξία να εκτελεί λεπτομερής και σε βάθος περιγραφή των χαρακτηριστικών του κάθε πακέτου, για το οποίο θέλει να γίνει έλεγχος για τον εντοπισμό μίας επίθεσης. Τα Rules χωρίζονται σε δύο κατηγορίες στα **Generic Rules** και στα **Unique Rules**. Αυτά ορίζονται αυτομάτως από το snort αναλόγως με της πληροφορίες που φέρει το header του Rule.

- **Generic**, είναι οι κανόνες που στον ορισμό των IP δικτύων ή πόρτων έχουν την λέξη κλειδί "**any**". Αυτό σημαίνει ότι ο κανόνας είναι έτσι ορισμένος ώστε να ελέγχει οποιοδήποτε πακέτο που έχει υπογραφή στο payload του αυτό που περιγράφεται στα χαρακτηριστικά του κανόνα.
- **Unique Rules**, είναι αυτά που έχουν συγκεκριμένη έκταση δικτύων και πορτών που εξετάζουν.

4.6 Η ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ SNORT

Στην ενότητα αυτή, θα μελετηθεί η λειτουργία του λογισμικού και ο τρόπος όπου παράγονται τα αποτελέσματά του. Το Snort αποτελείται από πέντε βασικά στάδια επεξεργασίας των δεδομένων που λαμβάνει και είναι:

- Η βιβλιοθήκη συλλογής πακέτων (Packet capture libraty-libpcap)
- Τον αποκωδικοποιητή πακέτων (Packet decoder)
- Τον προ-επεξεργαστή (Preprocessor)
- Την μηχανή ανίχνευσης (Detection Engine)
- Πακέτα λογισμικού αποτύπωσης εξόδου (Output Plug-ins)



Εικόνα 9 Στάδια επεξεργασίας ενός δικτυακού πακέτου που ελέγχει το Snort

4.7 ΒΙΒΛΙΟΘΗΚΗ ΣΥΛΛΟΓΗΣ ΠΑΚΕΤΩΝ (PACKET CAPTURE LIBRARY – LIBPCAP)

Το Snort για την συλλογή των πακέτων που διακινούνται στο δίκτυο που παρακολουθεί, μέσω της κάρτας δικτύου του υπολογιστή, χρησιμοποιεί την βιβλιοθήκη pcap (packet capture library). Αυτή είναι μία διασύνδεση προγραμματισμού εφαρμογών (Application Programming Interface – API), που εκτός από την συλλογή πακέτων από το δίκτυο, στις τελευταίες εκδόσεις, δίνεται η δυνατότητα να μεταδίδει και πακέτα στο επίπεδο συνδέσμου (link layer) του TCP/IP. Η συγκεκριμένη βιβλιοθήκη είναι σχεδιασμένη να επικοινωνεί με τις υπόλοιπες εφαρμογές στην γλώσσα προγραμματισμού C/C++.

Το γεγονός ότι η βιβλιοθήκη είναι ανεξάρτητο τμήμα του λογισμικού snort δίνει την δυνατότητα να χρησιμοποιείται σε διαφορετικά λειτουργικά , ενισχύοντας την ανεξαρτησία του snort. Το λειτουργικό σύστημα των Windows χρησιμοποιεί την βιβλιοθήκη pcap με την ονομασία Winpcap. Χαρακτηριστικό της βιβλιοθήκης αυτής είναι ότι συλλέγει τα πακέτα σε ακατέργαστη μορφή, δηλαδή όπως αυτά μεταφέρονται στο δίκτυο, μη επιτρέποντας στο εκάστοτε λειτουργικό σύστημα την αλλαγή σε αυτά. Το snort επωφελείται από αυτήν την ιδιότητα, αφού χρειάζονται όλες οι πληροφορίες που περιέχονται σ' ένα πακέτο για να ανιχνεύσει ορισμένες επιθέσεις.

4.8 ΑΠΟΚΩΔΙΚΟΠΟΙΗΤΗΣ ΠΑΚΕΤΩΝ (PACKET DECODING)

Η λειτουργία του είναι να λαμβάνει τα διαφορετικού τύπου πακέτα του δικτύου (Ethernet, SLIP, PPP κτλ) με την χρήση της βιβλιοθήκης libpcap. Έτσι κάθε πακέτο που φτάνει στο δίκτυο το πιάνει και το δίνει στην μηχανή αποκωδικοποίησης. Στην συνέχεια επιλέγεται τι είδους πακέτο είναι με βάση το δεύτερο επίπεδο του OSI (OSI-layer2) και από εκεί καλείται η κατάλληλη συνάρτηση που θα κάνει την πρώτη αποκωδικοποίηση του πακέτου. Για παράδειγμα, αν είναι ένα πακέτο Ethernet, θα κληθεί η κατάλληλη συνάρτηση για Ethernet decoding. Η διαδικασία αυτή συνεχίζεται για κάθε επίπεδο του OSI μέχρι το πακέτο να αποκωδικοποιηθεί πλήρως. Αυτό πετυχαίνεται εκμεταλλευόμενο την ιδιότητα της ενθυλάκωσης που χρησιμοποιούν τα πρωτόκολλα του διαδικτύου. Στην πραγματικότητα, ο αποκωδικοποιητής πακέτων είναι μία σειρά από αποκωδικοποιητές όπου, ο καθένας αποκωδικοποιεί συγκεκριμένα στοιχεία των πρωτοκόλλων. Τα αποτελέσματα της αποκωδικοποίησης για κάθε πακέτο δομούνται σε μία δομή την struct _Packet και στην συνέχεια προωθούνται στο επόμενο στάδιο επεξεργασίας.

4.9 ΠΡΟ-ΕΠΕΞΕΡΓΑΣΤΕΣ (PREPROCESSORS)

Οι προεπεξεργαστές είναι plug-ins του Snort που εκτελούνται μετά την αποκωδικοποίση πακέτων και πριν την μηχανή ανίχνευσης. Επιτρέπει στο λογισμικό να επεκτείνει την λειτουργικότητά του, δίνοντας την δυνατότητα σε χρήστες και προγραμματιστές να εισάγουν plug-ins αρκετά εύκολα. Μπορούν να χρησιμοποιηθούν είτε για να ελέγξουν τα πακέτα για ύποπτη δραστηριότητα είτε για να τα επεξεργαστούν έτσι, ώστε η μηχανή ανίχνευσης να μπορεί να τα αξιοποιήσει αποδοτικότερα. Οι προεπεξεργαστές καλούνται προς εκτέλεση μία μόνο φορά για κάθε πακέτο. Θα πρέπει να σημειωθεί ότι υπάρχουν είδη επιθέσεων τα οποία δεν θα μπορούσαν να ανιχνευθούν από το σύστημα του Snort χωρίς την επιπλέον επεξεργασία των προεπεξεργαστών, όπως για παράδειγμα ο frag2 που ανασυνθέτει τα κατακερματισμένα πακέτα σε ένα νέο

πακέτο ώστε να μπορούν να εφαρμόζονται οι κανόνες στο νέο πακέτο και όχι στο κάθε κομμάτι του «παλιού» πακέτου.

Οι προεπεξεργαστές είναι πολύ σημαντικό χαρακτηριστικό του IDS λόγω του ότι τα plug-ins μπορούν να ενεργοποιηθούν ή ν' απενεργοποιηθούν κατά βούληση του διαχειριστή του λογισμικού. Αν για παράδειγμα, δεν επιθυμείται ο έλεγχος του δικτύου για σαρώσεις θυρών, δίνεται η δυνατότητα απενεργοποίησης του εν λόγω plug-in χωρίς να επηρεαστεί το υπόλοιπο σύστημα επεξεργασίας. Οι παράμετροι που αφορούν στους προεπεξεργαστές βρίσκονται και μπορούν να διαμορφωθούν μέσω του αρχείου snort.conf ανάλογα με τις ανάγκες του δικτύου.

Η τελευταίες εκδόσεις του Snort 2.9 και έπειτα, χρησιμοποιούν τους κάτωθι προεπεξεργαστές:

- **Frag3**

  Έχει δημιουργηθεί για να αντικαταστήσει τον παλαιότερο frag2 προσφέροντας γρηγορότερη εκτέλεση, απλούστερη διαχείριση δεδομένων και αντιμετώπιση τεχνικών αποφυγής ανίχνευσης. Ο συγκεκριμένος προεπεξεργαστής έχει σαν στόχο τα πακέτα που περνούν κατακερματισμένα από το δίκτυο που παρακολουθείται από το snort, να τα επαναδομεί στην αρχική τους μορφή (ένα πακέτο). Με αυτό το μηχανισμό δύναται η δυνατότητα στο λογισμικό να δοκιμάζει τους κανόνες και να καταλαβαίνει αν γίνεται προσπάθεια επίθεσης.

- **Stream5**

  Έχει αντικαταστήσει τον Stream4 αλλά και τον Flow που υπήρχαν σε προηγούμενες εκδόσεις του snort. Ο προεπεξεργαστής Stream5 επιτρέπει την ανασυγκρότηση της ροής δεδομένων TCP καθώς και την ανάλυση με βάση την κατάσταση της ροής. Έχει την δυνατότητα να παρακολουθεί πολλές ταυτόχρονες ροές TCP πακέτων. Τέλος, έχει την δυνατότητα παρακολούθησης πακέτων UDP.

- **sfPortscan**

  Ο προεπεξεργαστής αυτός έχει αναπτυχθεί από την Sourcefire και έχει σχεδιαστεί για να ανιχνεύει την πρώτη φάση σε μία δικτυακή επίθεση, την φάση της αναγνώρισης. Στην φάση αυτή, ο επιτιθέμενος προσπαθεί να ανακαλύψει τι είδος δικτυακά πρωτόκολλα και υπηρεσίες υποστηρίζει ο διακομιστής του δικτύου. Μίας και ο επιτιθέμενος δεν έχει γνώση του στόχου του, τα περισσότερα ερωτήματα που στέλνει στον διακομιστή θα έχουν αρνητική απάντηση, αφού πολλές από τις υπηρεσίες δεν υπάρχουν. Λαμβάνοντας υπόψη πως πρόκειται για «νόμιμη» δικτυακή επικοινωνία, οι αρνητικές απαντήσεις από διακομιστές είναι σπάνιες και πόσο μάλλον πολλαπλές αρνητικές απαντήσεις σε ένα δεδομένο χρόνο. Με αυτό τον τρόπο ο sfportscan προσπαθεί να βρει αν συμβαίνει επίθεση με σάρωση θυρών του δικτύου.

Ένα από τα γνωστότερα εργαλεία σάρωσης θυρών που χρησιμοποιείται σήμερα είναι το Nmap, όπου θα χρησιμοποιηθεί και για τις ανάγκες του εργαστηριακού μέρους.

- **RPC Decode**

  Ο σκοπός του προεπεξεργαστή είναι να κανονικοποιήσει τις πολλαπλές κατατμημένες εγγραφές σε μία ολοκληρωμένη εγγραφή, ώστε να είναι δυνατή η αναγνώριση της υπογραφής μιας κακόβουλης εγγραφής από την μηχανή ανίχνευσης. Αν είναι ενεργοποιημένος ο Stream5, θα επεξεργαστεί μόνο την κίνηση του πελάτη (client).

- **Performance Monitor**

  Ο συγκεκριμένος προεπεξεργαστής επιτρέπει την μέτρηση της πραγματικής και θεωρητικής απόδοσης του Snort. Όταν είναι ενεργοποιημένος, μπορεί να τυπώνει τα στατιστικά στοιχεία είτε στην κονσόλα είτε σε ένα αρχείο για μετέπειτα επεξεργασία. Μερικά από την πληθώρα στοιχείων που παράγει είναι το ποσοστό χαμένων πακέτων, η

χρήση του δικτύου, η χρήση της CPU και πολλά στατιστικά όσο αφορά τις συνδέσεις του δικτύου.

- **HTTP Inspect**

  Είναι ένας γενικά αποκοδικοποιητής HTTP για εφαρμογές χρήστη. Ο προεπεξεργαστής βρίσκει στον buffer του δικτύου τα πεδία των HTTP δεδομένων και τα κανονικοποιεί. Με τον όρο κανονικοποίηση εννοείται η διαδικασία «μετάφρασης» μιας ασαφούς συλλογής χαρακτηριστικών, όπως οι Unicode, σε μια συλλογή χαρακτηριστικών που είναι αναγνωρίσιμη από το Snort. Η κωδικοποίηση των δεδομένων HTTP είναι μέθοδος που χρησιμοποιείται ευρέως από τους crackers για να καλύψουν τα ίχνη μίας επίθεσης από το σύστημα ανίχνευσης εισβολής. Χωρίς τον προεπεξεργαστή, αυτό με μία μεταμφίεση των πακέτων ώστε να μην ταιριάζει στις υπάρχουσες υπογραφές ανίχνευσης, ο διακομιστής ιστοσελίδων θα θεωρούσε έγκυρο ένα τέτοιο URL.

- **SMTP Preprocessor**

  Ο προεπεξεργαστής αυτός χρησιμοποιείται για την αποκωδικοποίηση SMTP κίνησης. Σε έναν δεδομένο προσωρινό χώρο αποθήκευσης δεδομένων, μπορεί να αποκωδικοποιήσει το πρωτόκολλο και να εντοπίσει τις εντολές SMTP καθώς και τις απαντήσεις τους. Έχει την δυνατότητα, εκτός από την κανονικοποίηση της ροής δεδομένων SMTP, να ελέγχει για αδυναμίες υπερχείλισης της μνήμης (Buffer Overflow) και συμπεριφοράς που δεν είναι ορισμένες στα RFC.

- **POP**

  Είναι ένας αποκωδικοποιητής POP3 για τις εφαρμογές του χρήστη. Λαμβάνει υπόψη την ροή δεδομένων όπου και αποκωδικοποιεί εντολές POP3 και τις απαντήσεις αυτών. Ο προεπεξεργαστής, αποθηκεύει την κατάσταση των μεμονωμένων πακέτων. Ωστόσο, η διατήρηση της σωστής κατάστασης των πακέτων αυτών, εξαρτάται από την επανασυναρμολόγηση του διακομιστή. Θα πρέπει να είναι

ενεργοποιημένος ο προεπεξεργαστής Stream5 και οι θύρες του POP να εμπεριέχονται στις παραμέτρους του stream5 για την λειτουργία και σωστή επανασυναρμολόγηση του POP. Τέλος ο POP χρησιμοποιεί GID 142 για την καταγραφή γεγονότων.

- **IMAP**

  Ο προεπεξεργαστής αυτός είναι αποκωδικοποιητής IMAP4 για εφαρμογές του χρήστη. Βρίσκει και αποκωδικοποιεί στο δίαυλο δεδομένων, εντολές και απαντήσεις IMAP4. Μαρκάρει τις εντολές, τα δεδομένα τις επικεφαλίδας των πακέτων και εξάγει τα συνημμένα IMAP4 αποκωδικοποιώντας τα κατάλληλα. Για την λειτουργία αυτού του προεπεξεργαστή χρειάζεται η ενεργοποίηση του stream5 και χρησιμοποιεί GID 141 για την καταγραφή γεγονότων.

- **FTP/Telnet Preprocessor**

  Αυτός ο προεπεξεργαστής είναι μια βελτιωμένη έκδοση του παρωχημένου αποκωδικοποιητή Telnet, ο οποίος παρέχει τη δυνατότητα statefull ελέγχου ροών δεδομένων FTP και Telnet. Είναι ικανός να αποκωδικοποιήσει την ροή δεδομένων, να αναγνωρίσει τις εντολές και τις απαντήσεις FTP και Telnet καθώς και να κανονικοποιήσει τα πεδία αυτών. Ο προεπεξεργαστής ελέγχει τόσο τις αιτήσεις του πελάτη όσο και τις απαντήσεις του διακομιστή.

- **SHH**

  Ο προεπεξεργαστής SHH έχει σχεδιαστεί για να ανιχνεύει τα ακόλουθα προβλήματα ασφαλείας (exploits): Challenge-Response Buffer Overflow, CRC 32, Secure CRT και το Protocol Mismatch.

- **DNS**

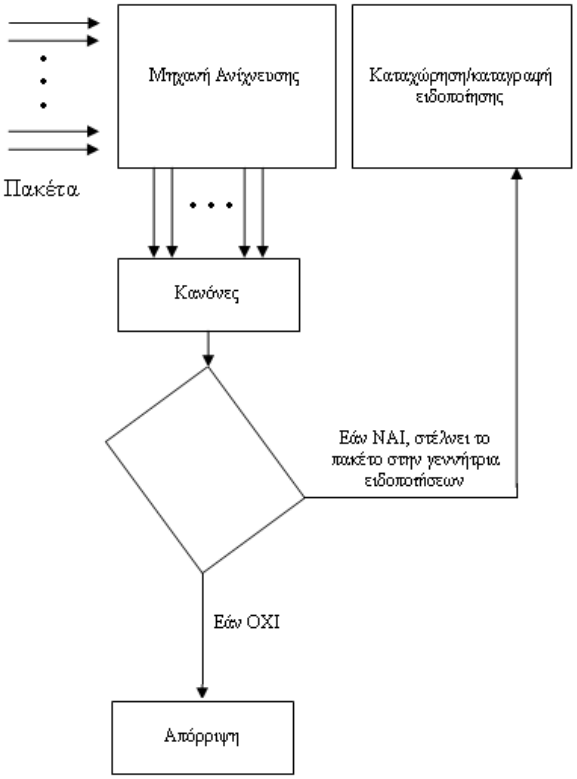
  Χρησιμοποιείται για την αποκωδικοποίηση απαντήσεων DNS και έχει την δυνατότητα να ανιχνεύει τα ακόλουθα προβλήματα ασφαλείας: DNS Client Rdata Overflow, Obsolete Record Types, και Experimental Record Types.

- **DCE/RPC 2**

  Ο κύριος σκοπός του προεπεξεργαστή είναι εκτελεί SMB ανακατάτμηση και DCE/RFC ανασυγκρότηση για να αποφύγει τους κανόνες αποφυγής αυτών των μεθόδων.

4.10 Μηχανή Ανίχνευσης (Detection Engine)

Αφού τα πακέτα θα ελεγχθούν από τους προ-επεξεργαστές που είναι ενεργοποιημένοι, τον λόγο έχει η μηχανή ανίχνευσης. Αυτή η διαδικασία θα μπορούσε να οριστεί ως η «καρδιά» του Snort. Σκοπός της είναι η συλλογή πληροφοριών από τον αποκωδικοποιητή πακέτων και τους προεπεξεργαστές, συγκρίνοντας τα περιεχόμενα των πακέτων με ένα σύνολο κανόνων με βάση την ανίχνευση του plug-in. Αν διαπιστωθεί κάποια ομοιότητα, στέλνει τα πακέτα στην διαδικασία καταχώρησης ή αποτύπωσης εξόδου αυτών και παραγωγής ειδοποίησης.



Εικόνα 10 Διάγραμμα διαδικασίας μηχανής ανίχνευσης του Snort

Οι κανόνες αυτοί περιέχουν υπογραφές για τις επιθέσεις. Δηλαδή η μηχανή ανίχνευσης είναι υπεύθυνη για την δημιουργία των υπογραφών, αφού επεξεργαστεί τους κανόνες. Η επεξεργασία των κανόνων γίνεται με την σειρά που βρίσκονται στο αρχείο και τοποθετούνται σε μια εσωτερική δομή δεδομένων. Η διαδικασία αυτή συμβαίνει κατά την εκκίνηση του snort, γεγονός που σημαίνει ότι, αν τροποποιηθεί κάποιος από τους κανόνες, θα πρέπει να γίνει επανεκκίνηση του λογισμικού.

4.11 ΠΑΚΕΤΑ ΛΟΓΙΣΜΙΚΟΥ ΑΠΟΤΥΠΩΣΗΣ ΕΞΟΔΟΥ (OUTPUT PLUG – INS)

Όταν στον έλεγχο της μηχανής ανίχνευσης ένα πακέτο αναγνωριστεί πως ταιριάζει με κάποιον από τους κανόνες, ενεργοποιείται μία ειδοποίηση, η οποία δημιουργεί και καταγράφει το συμβάν σε επιθυμητή μορφή. Το snort υποστηρίζει μια ποικιλία plug-ins, για την αποτύπωση των δεδομένων. Έτσι μερικές από τις υπηρεσίες που χρησιμοποιεί η έκδοση 2.9 του snort είναι:

- **Alert_syslog**

  Αυτή η υπηρεσία αποτυπώνει την έξοδο των αποτελεσμάτων του συστήματος του λογισμικού snort. Μπορεί να χρησιμοποιηθεί και για την καταγραφή πληροφοριών από άλλα διαφορετικά λογισμικά, όπως firewalls, server http κ.α.

- **Alert_fast**

  Τυπώνει τις ειδοποιήσεις που παράγονται σε διάταξη μίας σειράς ανά εγγραφή στο αρχείο που θα επιλεγεί. Είναι ο γρηγορότερος τρόπος αποτύπωσης ειδοποιήσεων συγκριτικά με την υπηρεσία alert full που θα ορίσουμε αμέσως μετά, αφού δεν εγγράφει τις κεφαλίδες των πακέτων στο αρχείο.

- **Alert_full**

  Το συγκεκριμένο plug-in δημιουργεί έναν κατάλογο για κάθε διεύθυνση που παράγεται ειδοποίηση και μέσα σε αυτόν αποθηκεύει σε αρχείο το αποκωδικοποιημένο περιεχόμενο των πακέτων, συμπεριλαμβανομένων και των κεφαλίδων του. Είναι ένας απαρχαιωμένος τρόπος καταγραφής, γιατί δεσμεύει μεγάλο χώρο για την αποθήκευση των δεδομένων. Μπορεί να χρησιμοποιηθεί σε χαμηλής χωρητικότητας δίκτυα.

- **Log_tcpdump**

  Είναι μία υπηρεσία που καταγράφει τις ειδοποιήσεις σε ένα αρχείο σύμφωνα με την διαμόρφωση που υποστηρίζει το πρόγραμμα tcpdump.

Χρησιμοποιείται σε περιπτώσεις που επιθυμείται η περαιτέρω επεξεργασία των συμβάντων.

- **Database**

  Έχει την δυνατότητα να καταγράφει τα συμβάντα στις εξής σχεσιακές βάσεις δεδομένων: MySQL, MSSQL, PostgreSQL, Oracle καθώς και σε συμβατές με το ODBC του Unix. Η επιλογή του να έχουμε τα στοιχεία που μας παρέχει το Snort αποθηκευμένα σε μια σχεσιακή βάση δεδομένων αυξάνει τις δυνατότητες επεξεργασίας των δεδομένων, όμως στην περίπτωση που μιλάμε για δίκτυα με αρκετή κίνηση είναι πιθανό να δημιουργηθεί συμφόρηση λόγω των αυξημένων εγγραφών.

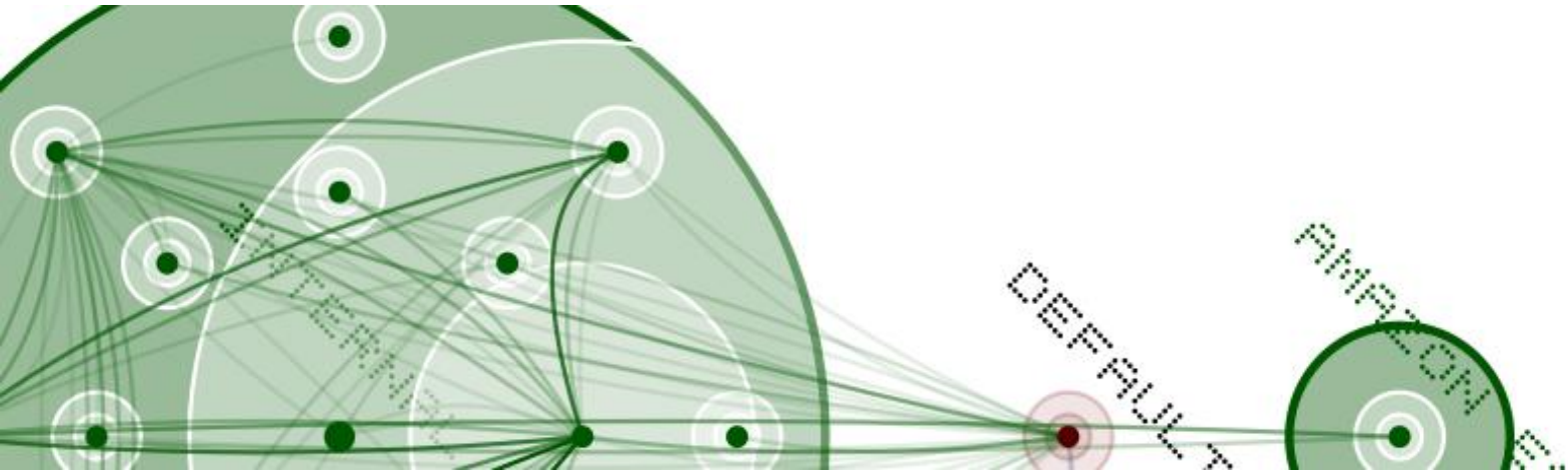
- **Alert_prelude**

  Η συγκεκριμένη υπηρεσία επιτρέπει την αποθήκευση των εξαγόμενων αποτελεσμάτων σε μια βάση δεδομένων του συστήματος IDS prelude. Σε παλαιότερες εκδόσεις του snort αυτό το plug-in αποτελούσε επιπρόσθετη επιλογή.

- **Unified2**

  Είναι ο αντικαταστάτης της διαμόρφωσης unified και έχει σχεδιαστεί με τα ίδια χαρακτηριστικά αποδοτικότητας, αλλά διαφορετικές μορφές αποθήκευσης. Σκοπός του συγκεκριμένου λογισμικού αποτύπωσης εξόδου είναι να επιτρέψει στο Snort να εξάγει τα δεδομένα του με τον πιο γρήγορο τρόπο, για να επεξεργαστούν στην συνέχεια με κάποιο εξωτερικό πρόγραμμα που γνωρίζει τη συγκεκριμένη διαμόρφωση. Η πιο συχνά χρησιμοποιούμενη εφαρμογή είναι το Barnyard, το οποίο αναλύει τα δεδομένα και στέλνει την έξοδο του σε κάποια βάση δεδομένων για διαχείριση.

ЕФАРМОГН



5. ΕΡΓΑΣΤΗΡΙΑΚΟ ΜΕΡΟΣ

Για τις ανάγκες της εργασίας και για την κατανόηση του τρόπου λειτουργίας των συστημάτων ανίχνευσης εισβολών, πραγματοποιήθηκε εγκατάσταση του λογισμικού Snort σ' έναν εργαστηριακό υπολογιστή (host) ***(Παράρτημα 1)***, ο οποίος εντάσσεται σ' ένα από τα υπό-δίκτυα της σχολής. Αυτός ο τρόπος εγκατάστασης του συστήματος ανίχνευσης ανήκει στην κατηγορία των Host-IDS, όπως αναφέρθηκε σε παραπάνω κεφάλαιο αναλυτικά.

Το HIDS σύστημα αυτό, είναι ενταγμένο σε ένα δίκτυο που η συνδεσμολογία του είναι βασισμένη σε μεταγωγέα (switch). Το switch έχει σαν κύριο χαρακτηριστικό ότι κάθε θύρα του δεν έχει το ίδιο εύρος ζώνης μεταφοράς δεδομένων, όπως συμβαίνει στα hubs, όπου όλες οι θύρες μοιράζονται το ίδιο εύρος ζώνης του μέσου μεταφοράς δεδομένων. Αυτό καθιστά αδύνατη την επίβλεψη από το HIDS όλων των θυρών του δικτύου, εκτός και αν το σύστημα ανίχνευσης έχει συνδεθεί στην θύρα SPAN (switch port analyzer) του μεταγωγού, όπου τότε μπορεί να επιβλέψει όλους τους υπολογιστές του δικτύου και επομένως μετατρέπεται σε NIDS.

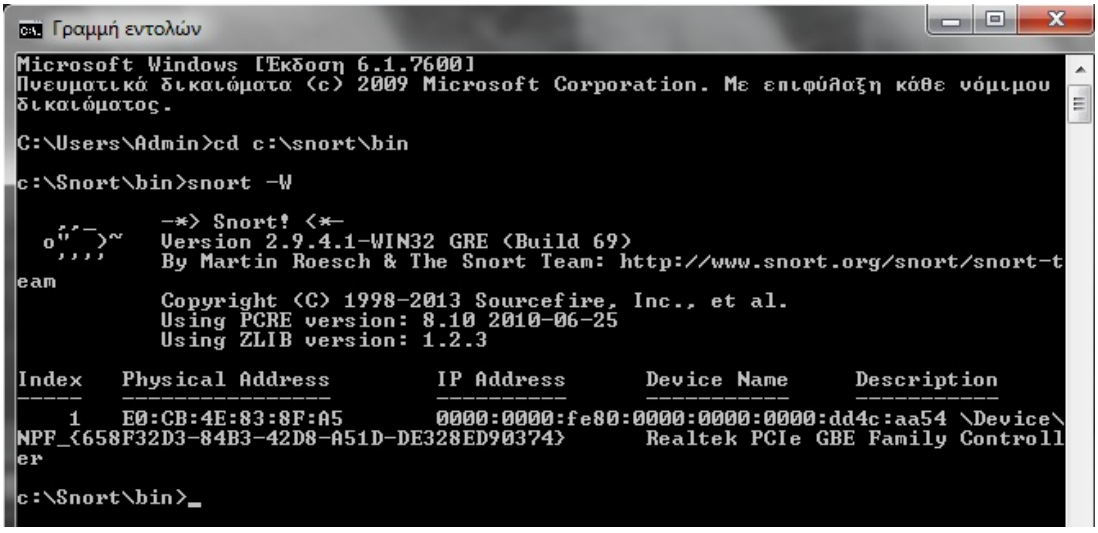
Με τις παραπάνω τεχνικές εφαρμογής των NIDS όμως, δεν μπορούμε να έχουμε μία πλήρη και ακριβή εικόνα των επιθέσεων που λαμβάνουν χώρα στο δίκτυό που εξετάζεται κάθε χρονική στιγμή, γι' αυτό επιλέχτηκε η τοποθέτηση και η μελέτη της συμπεριφοράς ενός Host based - IDS στο δίκτυο.

5.1 ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΡΥΘΜΙΣΗ ΤΟΥ SNORT IN IDS MODE ΣΕ WINDOWS 7

Όπως έχει αναφερθεί, το λογισμικό που θα χρησιμοποιηθεί και θα μελετηθεί διανέμεται δωρεάν στο εμπόριο με λίγες μόνο ιδιότητες κλειδωμένες (επί πληρωμή) και μπορεί να βρει κανείς το αρχείο εγκατάστασης, όπως και το φάκελο με τους κανόνες (rules) που χρησιμοποιεί το λογισμικό αυτό, στην επίσημη ιστοσελίδα[4]. Η έκδοση που χρησιμοποιείται είναι η 2.9.4.1 του Snort.

Αφού έχει γίνει εγκατάσταση της βιβλιοθήκης Winpcap και του αρχείου εγκατάστασης, καθώς και έχει κατεβεί το συμπιεσμένο αρχείο με τους κανόνες του Snort (που είναι: Snortrules-snapshot-2931.tar), ακολουθούνται τα παρακάτω βήματα για την ρύθμισή του:

1. Γίνεται αποσυμπίεση του αρχείο των κανόνων μέσα στον κατάλογο Snort, επιλέγοντας αντικατάσταση χωρίς διατήρηση των προηγουμένων.
2. Ανοίγεται την γραμμή εντολών (command prompt) από τον διαχειριστή και πηγαίνει στο φάκελο που βρίσκεται το snort, όπου είναι: C:\snort\bin>
3. Για να ειδωθεί ο αριθμός της κάρτας δικτύου που θέλει να ρυθμιστεί για να παρακολουθεί το Snort, πληκτρολογείται η εντολή C:\snort\bin> snort –W . Έτσι εμφανίζεται στην οθόνη η ακόλουθη εικόνα:



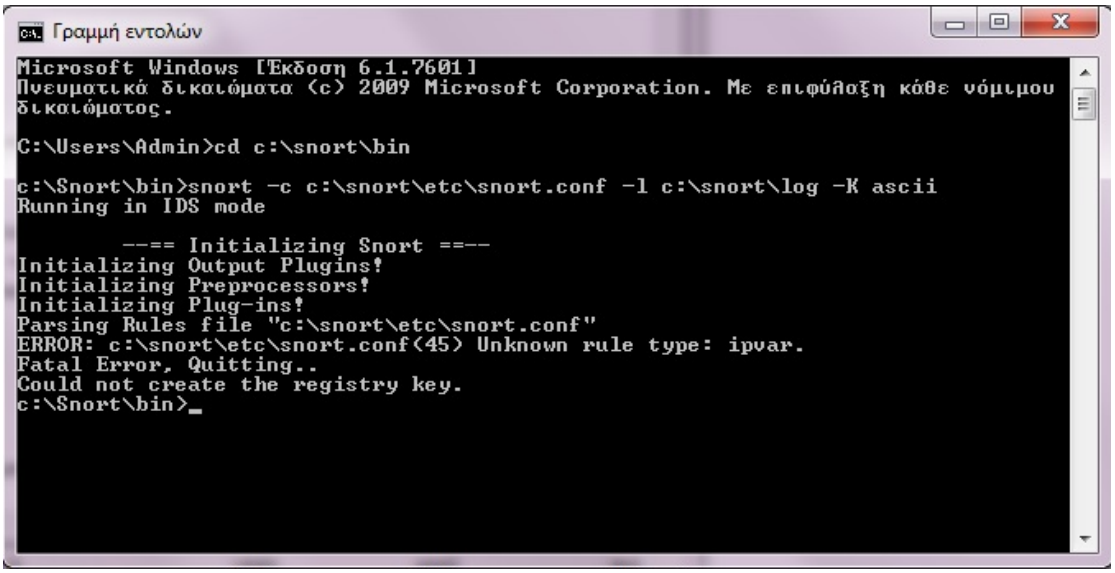
Εικόνα 11 Στιγμιότυπο 1ο

[4] http://www.snort.org/

4. Πληκτρολογείται η εντολή:

   C:\snort\bin>snort –c c:\snort\etc\snort.conf –l c:\snort\log –K ASCII, όπου:

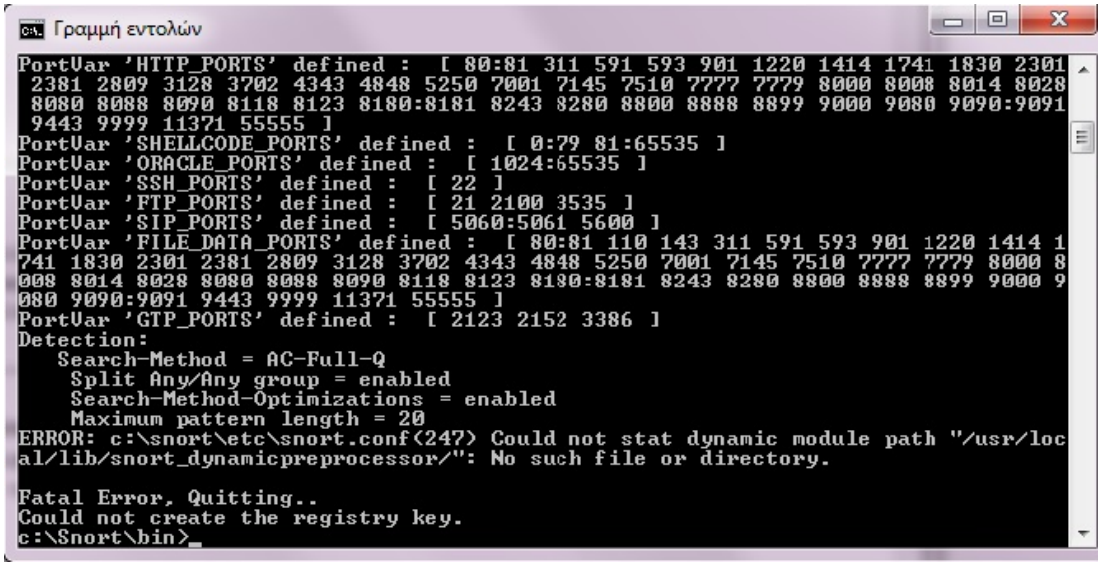
   - **–c** δηλώνεται το αρχείο, στην περίπτωση αυτή το snort.conf, όπου όποια διαμόρφωση εμφανίζεται, συμπεριλαμβανομένων ανάθεση μεταβλητών που χρησιμοποιούνται στις τιμές των κανόνων, ενημερώνουν το snort για το ποιά επιλογή προ-επεξεργαστή να χρησιμοποιήσει. Υποδεικνύει στο σύστημα τους κανόνες που θα συμπεριλάβει στην ανάλυση της κίνησης. Οι κανόνες αυτοί είναι είτε οι προκαθορισμένοι είτε ο χρήστης μπορεί να δημιουργήσει κάθε φορά δικούς του ανάλογα με τις απαιτήσεις του δικτύου.
   - **-l** δηλώνεται η διεύθυνση του καταλόγου που θα αποθηκεύει το λογισμικό τα αποτελέσματα, στην συγκεκριμένη περίπτωση του υποδεικνύεται ο κατάλογο log.
   - **-K** δηλώνεται η μορφή όπου θα εξάγει και θα αποθηκεύει στην συνέχεια τα αποτελέσματα το πρόγραμμα. Η προεπιλεγμένη ρύθμιση είναι σε pcap, εμείς όμως του ορίζουμε σε ASCII .



Εικόνα 12 Στιγμιότυπο 2°

Ανοίγεται το αρχείο snort.conf για να διορθωθούν τα σφάλματα όπου εντοπίζονται κατά την εκτέλεση της εντολής του προγράμματος. Το αρχείο βρίσκεται στον δίσκο \snort\etc.

5. Αλλάζονται όλες οι μεταβλητές με όνομα ipvar σε var, επειδή δεν αναγνωρίζονται ως τύποι κανόνων. Εκτελείται πάλι η εντολή του βήματος 2.



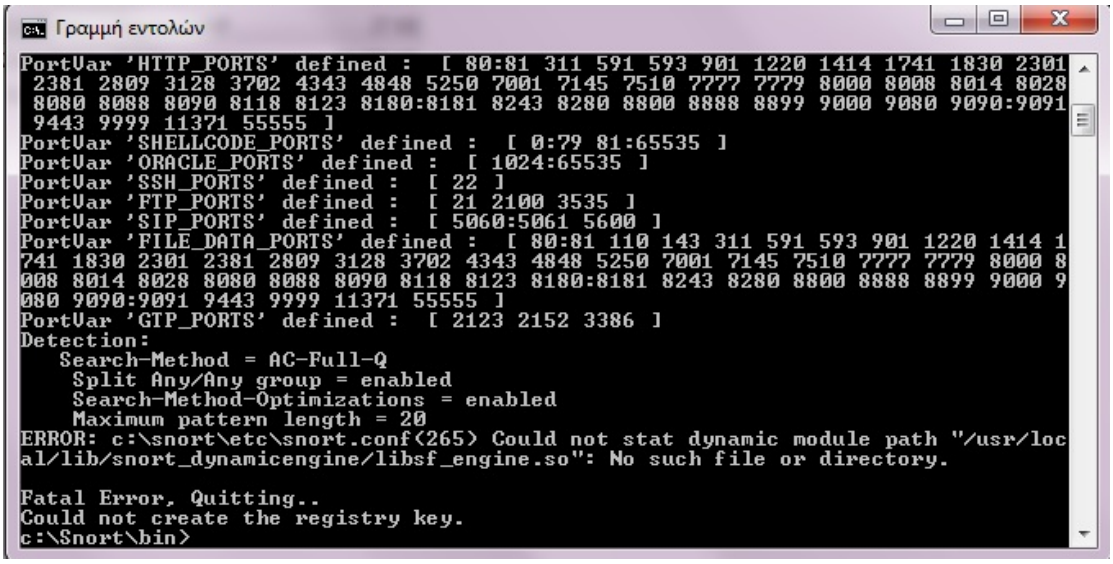
Εικόνα 13 Στιγμιότυπο 3°

6. Πρέπει ν' αλλάξει η διεύθυνση της βιβλιοθήκης σε C:\snort\lib\snort_dynamicpreprocessor που χρησιμοποιεί το πρόγραμμα και να προστεθούν οι διευθύνσεις των περιεχομένων του καταλόγου αυτού όπως προβάλλονται στο παρακάτω, στιγμιότυπο 4.

```
246 # path to dynamic preprocessor libraries
247 dynamicpreprocessor directory C:\Snort\lib\snort_dynamicpreprocessor
248
249 dynamicpreprocessor C:\Snort\lib\snort_dynamicpreprocessor\sf_dce2.dll
250 dynamicpreprocessor C:\Snort\lib\snort_dynamicpreprocessor\sf_dnp3.dll
251 dynamicpreprocessor C:\Snort\lib\snort_dynamicpreprocessor\sf_dns.dll
252 dynamicpreprocessor C:\Snort\lib\snort_dynamicpreprocessor\sf_ftptelnet.dll
253 dynamicpreprocessor C:\Snort\lib\snort_dynamicpreprocessor\sf_gtp.dll
254 dynamicpreprocessor C:\Snort\lib\snort_dynamicpreprocessor\sf_imap.dll
255 dynamicpreprocessor C:\Snort\lib\snort_dynamicpreprocessor\sf_modbus.dll
256 dynamicpreprocessor C:\Snort\lib\snort_dynamicpreprocessor\sf_pop.dll
257 dynamicpreprocessor C:\Snort\lib\snort_dynamicpreprocessor\sf_reputation.dll
258 dynamicpreprocessor C:\Snort\lib\snort_dynamicpreprocessor\sf_sdf.dll
259 dynamicpreprocessor C:\Snort\lib\snort_dynamicpreprocessor\sf_sip.dll
260 dynamicpreprocessor C:\Snort\lib\snort_dynamicpreprocessor\sf_smtp.dll
261 dynamicpreprocessor C:\Snort\lib\snort_dynamicpreprocessor\sf_ssh.dll
262 dynamicpreprocessor C:\Snort\lib\snort_dynamicpreprocessor\sf_ssl.dll
```

Εικόνα 14 Στιγμιότυπο 4°

Εκτελείται πάλι η εντολή του βήματος 2.



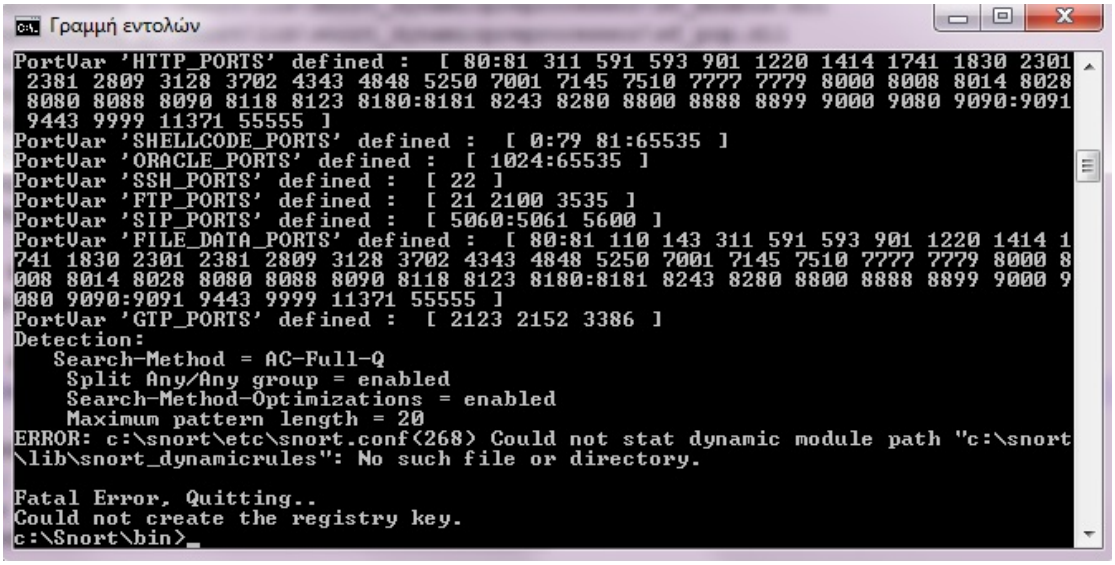
Εικόνα 15 Στιγμιότυπο 5°

7. Το ίδιο σφάλμα εντοπίζεται και για τις διευθύνσεις των δύο ακόμα βιβλιοθηκών. Οι αλλαγές ορίζονται στο στιγμιότυπο 5.

```
264  # path to base preprocessor engine
265  dynamicengine C:\Snort\lib\snort_dynamicengine\sf_engine.dll
266
267  # path to dynamic rules libraries
268  dynamicdetection directory C:\Snort\lib\snort_dynamicrules
```

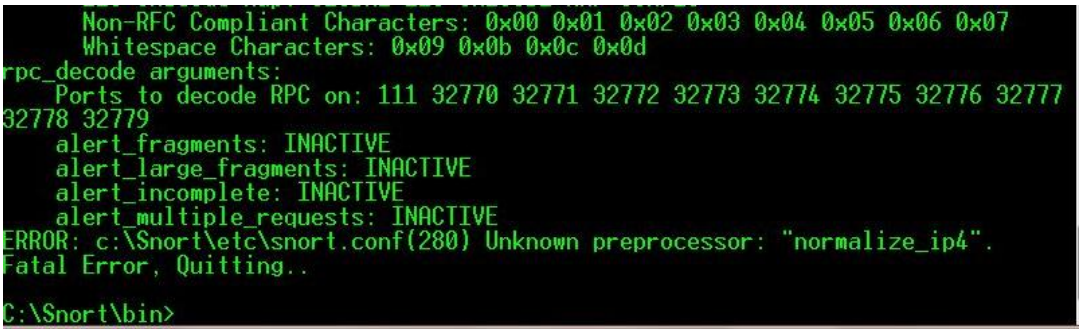
Εικόνα 16 Στιγμιότυπο 6°

Εκτελείται πάλι η εντολή του βήματος 2.



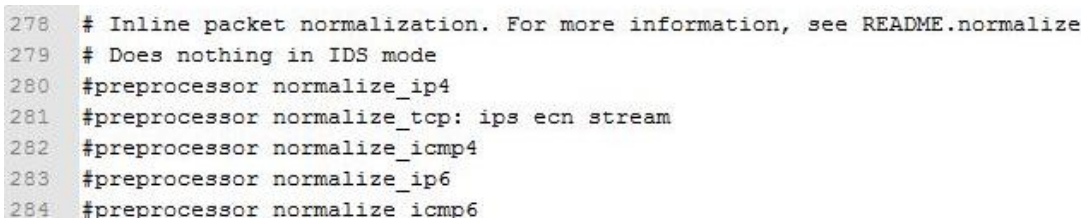
Εικόνα 17 Στιγμιότυπο 7°

8. Δημιουργείται ένας φάκελος στον C:\snort\lib , με όνομα snort_dynamicrules. Εκτελείται πάλι η εντολή του βήματος 2.



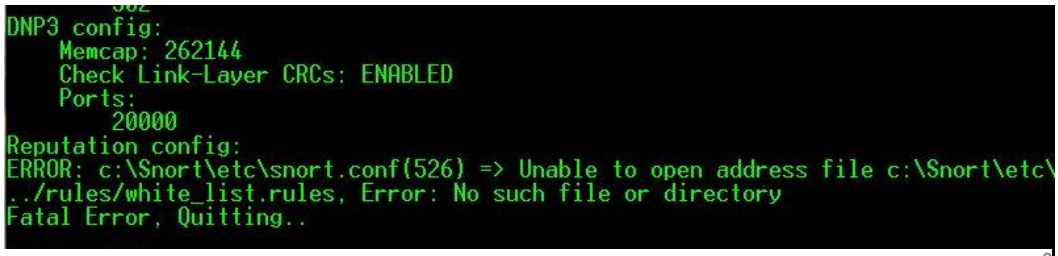
Εικόνα 18 Στιγμιότυπο 8°

9. Γίνονται σχόλια όλες οι γραμμές (preprocessor normalize lines) με την προσθήκη της δίεσης (#).



Εικόνα 19 Στιγμιότυπο 9°

Εκτελείται πάλι η εντολή του βήματος 2.



Εικόνα 20 Στιγμιότυπο 10°

10. Δημιουργείται μέσα στον κατάλογο c:\snort\rules\ ένα txt αρχείο με όνομα "white_list.rules" και ένα "Black_list.rules" και στο αρχείο snort.conf αλλάζεται το slash "/" σε backslash "\" των μεταβλητών RULE_PATH (Στιγμιότυπο 11ο), καθώς και στις γραμμές 525,526 αντίστοιχα (Στιγμιότυπο 12ο).

```
# 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 c:\snort\rules
var SO_RULE_PATH c:\snort\so_rules
var PREPROC_RULE_PATH c:\snort\preproc_rules

# If you are using reputation preprocessor set these
# Currently there is a bug with relative paths, they are relative to where snort is
# not relative to snort.conf like the above variables
# This is completely inconsistent with how other vars work, BUG 89986
# Set the absolute path appropriately
var WHITE_LIST_PATH ..\rules
var BLACK_LIST_PATH ..\rules
```

Εικόνα 21 Στιγμιότυπο 11°

```
preprocessor reputation: \
   memcap 500, \
   priority whitelist, \
   nested_ip inner, \
   whitelist $WHITE_LIST_PATH\white_list.rules, \
   blacklist $BLACK_LIST_PATH\black_list.rules

##################################################
```

Εικόνα 22 Στιγμιότυπο 12°

11. Στην γραμμή 573 αλλάζεται το όνομα "blacklist" σε "black_list".

```
include $RULE_PATH/attack-responses.rules
include $RULE_PATH/backdoor.rules
include $RULE_PATH/bad-traffic.rules
include $RULE_PATH/black_list.rules
include $RULE_PATH/botnet-cnc.rules
include $RULE_PATH/chat.rules
```

Εικόνα 23 Στιγμιότυπο 13°

12. Τέλος, εκτελείται η εντολή:

    C:\snort\bin>snort –c c:\snort\etc\snort.conf –l c:\snort\log –K ASCII –T

    όπου: -T είναι παράμετρος όπου δοκιμάζει και δηλώνει τυχόν σφάλματα πάνω στις τρέχουσες ρυθμίσεις του snort.

```
Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
Preprocessor Object: SF_DCERPC2  Version 1.0  <Build 3>

Snort successfully validated the configuration!
Snort exiting

C:\Snort\bin>
```

Εικόνα 24 Στιγμιότυπο 14°

Το πρόγραμμα τρέχει επιτυχώς με την τρέχουσα διαμόρφωση. Αφού έγινε η εγκατάσταση του λογισμικού Snort έγινε η ρύθμιση των δικτυακών παραμέτρων για την σωστή λειτουργία του συστήματος ανίχνευσης, όπως η καταχώρηση της διεύθυνσης IP του υπολογιστή στο δίκτυο με την μάσκα υπό-δικτύωσης καθώς και την διεύθυνση του διακομιστή DNS του δικτύου. Αυτές οι παράμετροι βρίσκονται στο αρχείο snort.conf του λογισμικού Snort.

```
40  ##################################################
41  # Step #1: Set the network variables.  For more information, see README.variables
42  ##################################################
43
44  # Setup the network addresses you are protecting
45  ipvar HOME_NET 83.212.59.188/24
46
47  # Set up the external network addresses. Leave as "any" in most situations
48  ipvar EXTERNAL_NET any
49
50  # List of DNS servers on your network
51  ipvar DNS_SERVERS 195.130.67.4
52
53  # List of SMTP servers on your network
54  ipvar SMTP_SERVERS $HOME_NET
55
56  # List of web servers on your network
57  ipvar HTTP_SERVERS $HOME_NET
```

Εικόνα 25 Στιγμιότυπο 15^{ο}

5.2 ΡΥΘΜΙΣΗ ΚΑΝΟΝΩΝ ΓΙΑ ΑΝΙΧΝΕΥΣΗ IDS

Έχοντας κάνει σωστά όλα τα παραπάνω βήματα εγκατάστασης του λογισμικού, χρειάζεται να προσαρμοστούν οι κανόνες (υπογραφές) που χρησιμοποιεί το snort, καθώς και να δημιουργηθεί ο κανόνας για την ανίχνευση επίθεσης στο τοπικό δίκτυο.

Για τον εντοπισμό επιθέσεων, όσο αφορά τα πρωτόκολλα icmp και udp, είναι αρκετοί οι υπάρχοντες κανόνες που εμπεριέχονται στο πακέτο εγκατάστασης. Ενεργοποιούνται, όπως φαίνεται και στο στιγμιότυπο 13 και 14 αντίστοιχα, ανοίγοντας το αρχείο icmp.rules και exploit.rules που βρίσκονται στον κατάλογο rules. Αφαιρείται έπειτα η δίεση (#) από την αρχή του κανόνα.

```
# Potentially "BAD" ICMP rules are included in icmp.rules

# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP-INFO IRDP router advertisement"; itype:9; reference:bugtraq,578; refer
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP-INFO IRDP router selection"; itype:10; reference:bugtraq,578; referenc
 alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP-INFO PING *NIX"; itype:8; content:"|10 11 12 13 14 15 16 17 18 19 1A 1E
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP-INFO PING BSDtype"; itype:8; content:"|08 09 0A 0B 0C 0D 0E 0F 10 11 1
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP-INFO PING BayRS Router"; itype:8; content:"|01 02 03 04 05 06 07 08 09
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP-INFO PING BeOS4.x"; itype:8; content:"|00 00 00 00 00 00 00 00 00 00 0
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP-INFO PING Cisco Type.x"; itype:8; content:"|AB CD AB CD AB CD AB CD AB
 alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP-INFO PING Delphi-Piette Windows"; itype:8; content:"Pinging from Del";
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP-INFO PING Flowpoint2200 or Network Management Software"; itype:8; cont
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP-INFO PING IP NetMonitor Macintosh"; itype:8; content:"|A9| Sustainable
 alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP-INFO PING LINUX/*BSD"; dsize:8; id:13170; itype:8; classtype:misc-activ
 alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP-INFO PING Microsoft Windows"; itype:8; content:"0123456789abcdefghijkln
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP-INFO PING Network Toolbox 3 Windows"; itype:8; content:"==============
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP-INFO PING Ping-O-MeterWindows"; itype:8; content:"OMeterObeseArmad"; d
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP-INFO PING Pinger Windows"; itype:8; content:"Data|00 00 00 00 00 00 00
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP-INFO PING Seer Windows"; itype:8; content:"|88 04|                ";  dep
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP-INFO PING Oracle Solaris"; dsize:8; itype:8; classtype:misc-activity;
 alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP-INFO PING Windows"; itype:8; content:"abcdefghijklmnop"; depth:16; clas
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP-INFO traceroute"; itype:8; ttl:1; classtype:attempted-recon; sid:385;
```

Εικόνα 26 Στιγμιότυπο 16°

```
# alert tcp $EXTERNAL_NET 1863 -> $HOME_NET any (msg:"EXPLOIT Microsoft MSN Messenger png overflow"; flow:to_client,establ
alert udp $EXTERNAL_NET any -> $HOME_NET 42 (msg:"EXPLOIT Microsoft Windows WINS name query overflow attempt UDP"; flow:to_
# alert tcp $EXTERNAL_NET any -> $HOME_NET 617 (msg:"EXPLOIT Arkeia backup client type 84 overflow attempt"; flow:to_serve
# alert tcp $EXTERNAL_NET any -> $HOME_NET 5001 (msg:"EXPLOIT Bontago Game Server Nickname buffer overflow"; flow:to_serve
#alert tcp $EXTERNAL_NET any -> $HOME_NET 42 (msg:"EXPLOIT Microsoft Windows WINS name query overflow attempt TCP"; flow:es
# alert tcp $EXTERNAL_NET any -> $HOME_NET 617 (msg:"EXPLOIT Arkeia backup client type 77 overflow attempt"; flow:to_serve
# alert tcp $EXTERNAL_NET any -> $HOME_NET 41523 (msg:"EXPLOIT ARCserve backup TCP slot info msg client domain overflow"; f
# alert udp $EXTERNAL_NET any -> $HOME_NET 41524 (msg:"EXPLOIT ARCserve backup UDP product info msg 0x9c client domain over
# alert tcp $EXTERNAL_NET any -> $HOME_NET 41523 (msg:"EXPLOIT ARCserve backup TCP product info msg 0x9c client name overf
# alert udp $EXTERNAL_NET any -> $HOME_NET 41524 (msg:"EXPLOIT ARCserve discovery service overflow"; flow:to_server; dsize:
# alert udp $EXTERNAL_NET any -> $HOME_NET 41524 (msg:"EXPLOIT ARCserve backup UDP product info msg 0x9c client name overf
```

Εικόνα 27 Στιγμιότυπο 17°

Στην συνέχεια για την ανίχνευση των επιθέσεων tcp θυρών θα χρειαστεί να δημιουργηθούν κανόνες, που θα λειτουργούν μόνο τοπικά του δικτύου για λόγους μόνο δοκιμών. Ανοίγεται το αρχείο local.rules που βρίσκεται στον κατάλογο rules και γράφονται οι κανόνες, όπως εμφανίζονται παρακάτω.

```
# ------------
# LOCAL RULES
# ------------
# This file intentionally does not come with signatures.  Put your local
# additions here.

alert tcp $HOME_NET any -> $EXTERNAL_NET 80 (msg:"Test for web traffic"; sid: 1000011;)
alert tcp $HOME_NET any -> $EXTERNAL_NET 135 (msg:"Test for web traffic"; sid: 1000001;)
```

Εικόνα 25 Στιγμιότυπο 15^{ο}

Οι δύο κανόνες καταγράφουν ειδοποιήσεις για επιθέσεις στις θύρες 80 και 135 tcp πακέτων, για την διεύθυνση του δικτύου που έχει οριστεί από τον διαχειριστή, για κάθε διεύθυνση που παράγει την επίθεση εντός του δικτύου ($HOME_NET any -> $EXTERNAL_NET). Όταν ανιχνεύεται μία τέτοια επίθεση, το λογισμικό θα ονομάζει την επίθεση "test for web traffic" που έχει οριστεί από το λεκτικό (εντολή) "msg". Δίνεται και ένας σειριακός αριθμός για την επίθεση από τον δημιουργό, με την εντολή "sid".

5.3 Υλοποίηση Επιθέσεων με το Λογισμικό NMAP - ZENMAP GUI

Ανοίγοντας την γραμμή εντολών των windows, τρέχεται το Snort σε IDS mode, με την εντολή του βήματος 2 παραπάνω. Με την βοήθεια του προγράμματος Nmap, έκδοση 6.25, δημιουργούνται εικονικές νόμιμες επιθέσεις στον υπολογιστή του δικτύου που έχει εγκατασταθεί το σύστημα ανίχνευσης εισβολών, από εσωτερικούς ή απομακρυσμένους υπολογιστές. Το Nmap ***(Παράρτημα 2)*** είναι ένα δωρεάν λογισμικό που εκτελεί σαρώσεις ασφαλείας σε υπολογιστές, με σκοπό να ανιχνεύει τις διαθέσιμες υπηρεσίες των διακομιστών καθώς και τις εκδόσεις των εφαρμογών των υπηρεσιών και του λειτουργικού συστήματος. Υπάρχουν κάποιες επιλογές τεχνικών στις εντολές που εφαρμόζονται για την δημιουργία των επιθέσεων ***(Παράρτημα 3)***, όπου μερικές εξ αυτών θα χρησιμοποιηθούν.

Ο τρόπος που λειτουργεί το λογισμικό Nmap, είναι ο ακόλουθος: Στέλνοντας υπερβολικό όγκο πακέτων δεδομένων, το επιτιθέμενο σύστημα IDS αντιλαμβάνεται μία απότομη αύξηση της κίνησης του δικτύου, με αποτέλεσμα

να την μεταφράζει ως επίθεση. Γι' αυτό κι ο τρόπος αυτός είναι νόμιμος. Οι επιθέσεις που πραγματοποιήθηκαν στο σύστημα με αυτό τον τρόπο είναι οι εξής:

- Σάρωση δικτυακών θυρών TCP (TCP port Scan)
- Σάρωση πακέτων UDP (UDP Scan)
- Σάρωση με πακέτα ICMP μέσω ping (Ping scan)
- Έντονη σάρωση χωρίς ping (Intense scan, no ping)

5.4 ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΠΙΘΕΣΕΩΝ ΑΝΑ ΚΑΤΗΓΟΡΙΑ

5.4.1 ΕΠΙΘΕΣΗ ΑΠΟ ΤΟ ΕΣΩΤΕΡΙΚΟ ΤΟΥ ΔΙΚΤΥΟΥ

Δοκιμάστηκαν επιθέσεις από το εσωτερικό του δικτύου, δηλαδή από άλλους κόμβους που είναι στο ίδιο switch, και διαπιστώθηκε ότι το snort αντιλαμβάνεται όλες τις μορφές των επιθέσεων, εκτός από τις TCP επιθέσεις. Αυτό πιθανόν να οφείλεται στο ότι δεν εμπίπτει σε κάποιο από τους κανόνες (rules) του Snort. Έτσι δημιουργήθηκε ο προαναφερόμενος νέος τοπικός κανόνας (local rule), με αποτέλεσμα τον εντοπισμό πλήρως των επιθέσεων.

5.4.2 ΕΠΙΘΕΣΗ ΜΕ ΣΑΡΩΣΗ ΔΙΚΤΥΑΚΩΝ ΘΥΡΩΝ TCP

Για την σάρωση της διαδικτυακής θύρας 80 εκτελείται η εντολή στο Nmap: nmap –sV –sT –p T:80 83.212.59.188. Παρατηρείται στο αρχείο alert.ids που βρίσκεται στον κατάλογο log η καταγραφή ειδοποίησης όπως φαίνεται παρακάτω.

```
128  [**] [1:1000011:0] Test for web traffic [**][Priority: 0]
129  03/27-18:27:00.488342 83.212.59.189:57131 -> 83.212.59.188:80
130  TCP TTL:38 TOS:0x0 ID:21693 IpLen:20 DgmLen:44
131  ******S* Seq: 0x80DE44D8  Ack: 0x0  Win: 0x400  TcpLen: 24
132  TCP Options (1) => MSS: 1460
```

Εικόνα 26 Στιγμιότυπο 16°

Παρατηρείται ότι δίνεται ο αριθμός sid του κανόνα που ανήκει η επίθεση. Δίπλα αναγράφεται το μήνυμα που έχει οριστεί να εμφανίζεται για τον συγκεκριμένο κανόνα. Εδώ επισημαίνεται ότι, πάντα αναγράφεται η ιδιότητα ή

το όνομα του πρωτοκόλλου. Έπειτα, κάνει μία εκτίμηση της επικινδυνότητας της επίθεσης με μεγαλύτερης σημασίας τον Αριθμό 0. Στην συνέχεια καταγράφεται η ημερομηνία και η ώρα που ανιχνεύτηκε η επίθεση καθώς και την ip διεύθυνση του αποστολέα (εισβολέας) και του παραλήπτη (επιτιθέμενου). Για το πρωτόκολλο TCP καταγράφει και την θύρα που εντόπισε την επίθεση. Ακολουθεί το όνομα του πρωτοκόλλου και διάφορα χαρακτηριστικά του πακέτου και του πρωτοκόλλου που στάλθηκε από τον εισβολέα. Αντίστοιχα και για την θύρα 135, την εντολή: nmap –sV –sT –p T:135 83.212.59.188.

```
302  [**] [1:1000001:0] Test for web traffic [**][Priority: 0]
303  03/29-18:02:59.586493 83.212.59.6:62595 -> 83.212.59.188:135
304  TCP TTL:57 TOS:0x0 ID:28206 IpLen:20 DgmLen:44
305  ******S* Seq: 0xBF027EBA  Ack: 0x0  Win: 0x400  TcpLen: 24
306  TCP Options (1) => MSS: 1460
```

Εικόνα 27 Στιγμιότυπο 17ο

5.4.3 ΕΠΙΘΕΣΗ ΜΕ ΣΑΡΩΣΗ ΠΑΚΕΤΩΝ UDP

Εκτελείται η εντολή στο Nmap: nmap –sS –sU –T4 –A –v 83.212.59.188 . Παρατηρείται στο αρχείο alert.ids που βρίσκεται στον κατάλογο log η καταγραφή ειδοποίησης όπως φαίνεται παρακάτω.

```
218  [**] [1:3200:10] EXPLOIT Microsoft Windows WINS name query overflow attempt UDP
219  [Classification: Attempted Administrator Privilege Gain] [Priority: 1]
220  03/27-18:38:33.337779 83.212.59.189:58181 -> 83.212.59.188:42
221  UDP TTL:128 TOS:0x0 ID:4345 IpLen:20 DgmLen:257 DF
222  Len: 229
223  [Xref => http://technet.microsoft.com/en-us/security/bulletin/MS04-006][Xref =>
```

Εικόνα 28 Στιγμιότυπο 18ο

Η ειδοποίηση για το UDP πρωτόκολλο διαφέρει στο ότι ο κανόνας που εμπεριέχεται σαν προεπιλογή, κατηγοριοποιεί την επίθεση εκτός της επικινδυνότητας και στο τέλος προτείνει τρόπους για την επίλυση, προσκομίζοντας ιστότοπους για την ενίσχυση ή ενημέρωση της ασφάλειας του δικτύου.

5.4.4 Επίθεση με Σάρωση Πακέτων ICMP μέσω PING

Εκτελείται η εντολή στο Nmap: nmap -sn 83.212.59.188 . Παρατηρείται στο αρχείο alert.ids που βρίσκεται στον κατάλογο log η καταγραφή ειδοποίησης όπως φαίνεται παρακάτω.

```
390  [**] [1:382:9] ICMP-INFO PING Windows [**]
391  [Classification: Misc activity] [Priority: 3]
392  03/29-18:08:13.582328 83.212.59.183 -> 83.212.59.188
393  ICMP TTL:128 TOS:0x0 ID:1280 IpLen:20 DgmLen:60
394  Type:8  Code:0  ID:1   Seq:1  ECHO
```

Εικόνα 29 Στιγμιότυπο 19ο

Πρόκειται για την παλαιότερη και πιο διαδεδομένη μορφή επίθεσης, επονομαζόμενη ως «Ping of denial». Η εντολή ping χρησιμοποιεί το πρωτόκολλο ICMP, το οποίο χρησιμοποιείται για την επικοινωνία μεταξύ υπολογιστών, χωρίς να χρειάζεται η υλοποίηση ενός ισχυρότερου και πιο περίπλοκου (συνεπώς και πιο αργού) πρωτοκόλλου όπως το TCP. Το ICMP μεταφέρει πολύ λίγες πληροφορίες οι οποίες ενημερώνουν κάθε υπολογιστή για την κατάσταση της σύνδεσής του με άλλα μηχανήματα. Δυστυχώς, η άμυνα από το Ping of Death είναι εξαιρετικά δυσχερής, καθώς το Ping αποτελεί τη μέθοδο με την οποία ένας Η/Υ δηλώνει ότι είναι ενεργός μέσα στο δίκτυο. Δεν είναι, λοιπόν, δυνατόν, το σύστημα να αρνηθεί να απαντήσει σε όποιο Ping δέχεται, με αποτέλεσμα αν τα μηνύματα αυτά είναι πάρα πολλά, ο επεξεργαστικός φόρτος να μεγαλώνει και πιθανόν να διακοπεί τελείως η λειτουργία του συστήματος. Θεωρητικά, αυτό το πρόβλημα μπορεί να αντιμετωπιστεί αν απενεργοποιηθεί (κλείσει) η ICMP port (στα windows αυτή είναι η default επιλογή). Δυστυχώς, με τον τρόπο αυτό μειώνεται η ταχύτητα σύνδεσης ενός Η/Υ με το δίκτυο. Έτσι, πολλοί διαχειριστές προτιμούν να διατηρούν αυτή τη δυνατότητα εν λειτουργία, παρ' όλους τους κινδύνους που συνεπάγεται κάτι τέτοιο, επιλέγοντας ένα σύστημα ανίχνευσης εισβολών, όπως το λογισμικό που μελετάται.

5.4.5 ΕΠΙΘΕΣΗ ΕΚΤΟΣ ΤΟΥ ΕΣΩΤΕΡΙΚΟΥ ΔΙΚΤΥΟΥ

Εν συνεχεία δοκιμάστηκαν επιθέσεις εκτός του εσωτερικού του δικτύου στον κόμβο που έχει εγκατασταθεί το σύστημα ανίχνευσης εισβολών και διαπιστώθηκε πως δεν εντόπισε καμία επίθεση. Αυτό οφείλεται στην σωστή αρχιτεκτονική ασφάλειας του δικτύου, όπου έχει σωστά ρυθμισμένη την πρώτη γραμμή άμυνάς του, που είναι το τείχος προστασίας(firewall) και αποτρέπει τις ανεπιθύμητες ενέργειες.

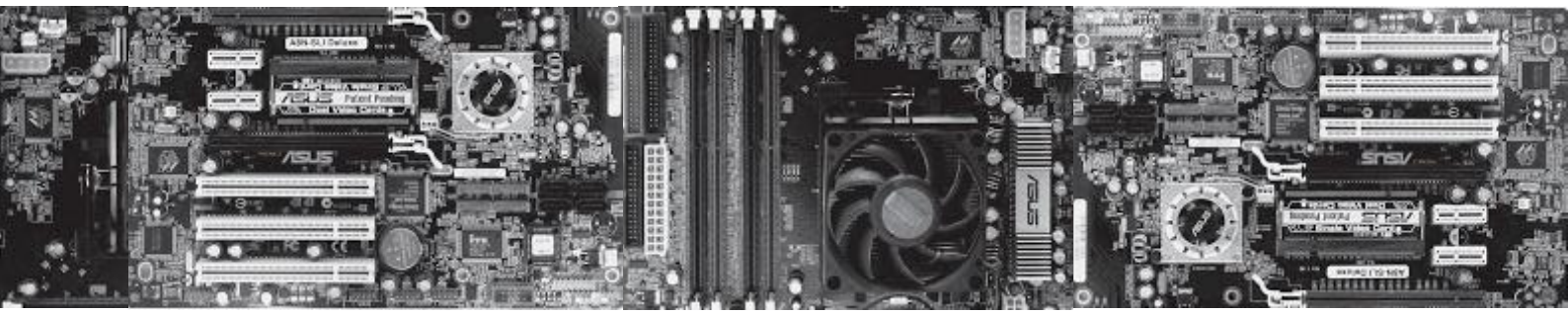
5.5 ΣΥΜΠΕΡΑΣΜΑΤΑ

Το Snort, στο εργαστηριακό περιβάλλον, λειτούργησε άριστα, ανιχνεύοντας όλες τις επιθέσεις που πραγματοποιήθηκαν. Έτσι, σύμφωνα με τα στατιστικά αποτελέσματα ***(Παράρτημα 4)***, μετά το πέρας της εκτέλεσης του προγράμματος, μπορεί να ειπωθεί ότι το λογισμικό Snort, με την σωστή εγκατάσταση, μπορεί να προσφέρει αρκετά υψηλή προστασία σ' ένα δίκτυο. Το Snort μπορεί να προλαμβάνει επιθέσεις εκτός του να τις ανιχνεύει, δηλαδή να λειτουργήσει ως ISP. Στην περίπτωση αυτή θα μπορεί να διακόπτει όποιες συνδέσεις θεωρεί επιθέσεις και έτσι να δημιουργηθεί ένα «άτρωτο» σύστημα ασφάλειας ενός δικτύου.

Στο σημείο αυτό, θα πρέπει να αναφερθεί ότι σε καμία περίπτωση δεν μπορεί να ελεγχθεί το σύνολο των κανόνων του Snort και πόσο μάλλον να βρεθεί τρόπος για την συνολική δοκιμή μέσω επιθέσεων της απόδοσης ενός IDS. Γι' αυτό επιλέχθηκαν ενδεικτικά τρόποι επιθέσεων που είναι εύκολο να βρεθούν στο διαδίκτυο και μπορεί κανείς να τις χρησιμοποιήσει σε δίκτυα υπολογιστών.

Με βάση τ' αποτελέσματα της εργαστηριακής προσομοίωσης του Snort μπορεί να συμπεράνει κανείς πως με τις κατάλληλες ρυθμίσεις του λογισμικού στην εγκατάσταση θα μπορούσε να θωρακίσει την ασφάλεια του δικτύου του ΤΕΙ Σερρών, προστατεύοντάς το από υπηρεσίες που δεν χρησιμοποιούν τους κανόνες ασφάλειας ή πιθανούς εισβολείς, εξωτερικούς και γιατί όχι και

εσωτερικούς. Τέλος θα πρέπει να τονιστεί πως, χωρίς προσωπικό με άριστη γνώση και κατάλληλη εκπαίδευση στο πεδίο της ασφάλειας των δικτύων και υπολογιστών, κανένα λογισμικό δεν είναι ικανό να παρέχει την μέγιστη απόδοση ασφάλειας σε ένα δίκτυο.



ΠΑΡΑΡΤΗΜΑ 1

ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ ΥΠΟΛΟΓΙΣΤΗ ΚΑΙ ΤΟΥ ΜΕΤΑΓΩΓΕΑ

Τα βασικότερα χαρακτηριστικά του εργαστηριακού υπολογιστή (host), στον οποίο έγινε η εγκατάσταση του Snort είναι τα ακόλουθα:

i. Desktop (ΕΠΕΞΕΡΓΑΣΤΗ:Intel Core (TM) i7 CPU 920 @ 2.67Ghz 2.6 Ghx, ΜΝΗΜΗ : 3Gb)
ii. Realtek PCIe GBE Family Controller

Τα χαρακτηριστικά του μεταγωγέα (switch) που χρησιμοποιήθηκε για τη δημιουργία του υποδικτύου είναι της εταιρείας Level One, FSW-2421 24-port Fast Ethernet Switch.

ΠΑΡΑΡΤΗΜΑ 2

ΘΥΡΕΣ (PORTS) ΕΠΙΚΟΙΝΩΝΙΑΣ ΕΝΟΣ ΣΥΣΤΗΜΑΤΟΣ

Το nmap είναι ένα πρόγραμμα όπου κάνει σάρωση των θυρών επικοινωνίας ενός συστήματος, οι οποίες χρησιμοποιούνται από πολλά δικτυακά πρωτόκολλα, όπως είναι το TCP και το UDP. Τα συστήματα ανίχνευσης είναι σε θέση να ανιχνεύουν τα πακέτα που θέλουν να χρησιμοποιήσουν τις θύρες επικοινωνίας με σκοπό την επίθεση στο σύστημα. Είναι λοιπόν σημαντικό να γνωρίζει ο διαχειριστής των συστημάτων ανίχνευσης εισβολών τον κωδικό, τον αριθμό και την λειτουργία της κάθε θύρας. Στις μέρες μας αυτό καθίσταται δύσκολο, διότι ο αριθμός των καταγεγραμμένων θυρών φτάνει στις 65.535 εκ των οποίων οι 1024 πρώτες ελέγχονται από την IANA.

Παραθέτονται ορισμένες από τις βασικότερες θύρες επικοινωνίας ενός συστήματος, για την ανταλλαγή πακέτων TCP και UDP, όπως αυτές ορίζονται στο RFC 1700 τον Οκτώβριο του 1994.

Κωδικός	Αριθμός	Περιγραφή Λειτουργίας
	0/tcp	Reserved
	0/udp	Reserved
tcpmux	1/tcp	TCP Port Service Multiplexer
tcpmux	1/udp	TCP Port Service Multiplexer
compressnet	2/tcp	Management Utility
compressnet	2/udp	Management Utility
compressnet	3/tcp	Compression Process
compressnet	3/udp	Compression Process
#	4/tcp	Unassigned
#	4/udp	Unassigned
Rje	5/tcp	Remote Job Entry
rje	5/udp	Remote Job Entry
#	6/tcp	Unassigned
#	6/udp	Unassigned
echo	7/tcp	Echo
echo	7/udp	Echo
#	8/tcp	Unassigned
#	8/udp	Unassigned
discard	9/tcp	Discard
discard	9/udp	Discard

www-http	80/tcp	World Wide Web HTTP
www-http	80/udp	World Wide Web HTTP
hosts2-ns	81/tcp	HOSTS2 Name Server
hosts2-ns	81/udp	HOSTS2 Name Server
xfer	82/tcp	XFER Utility
xfer	82/udp	XFER Utility
cisco-fna	130/tcp	cisco FNATIVE
cisco-fna	130/udp	cisco FNATIVE
cisco-tna	131/tcp	cisco TNATIVE
cisco-tna	131/udp	cisco TNATIVE
cisco-sys	132/tcp	cisco SYSMAINT
cisco-sys	132/udp	cisco SYSMAINT
statsrv	133/tcp	Statistics Service
statsrv	133/udp	Statistics Service
ingres-net	134/tcp	INGRES-NET Service
ingres-net	134/udp	INGRES-NET Service
loc-srv	135/tcp	Location Service
loc-srv	135/udp	Location Service
profile	136/tcp	PROFILE Naming System
profile	136/udp	PROFILE Naming System
snmp	161/udp	SNMP
snmptrap	162/tcp	SNMPTRAP
snmptrap	162/udp	SNMPTRAP
cmip-man	163/tcp	CMIP/TCP Manager
cmip-man	163/udp	CMIP/TCP Manager
cmip-agen	t164/tcp	CMIP/TCP Agent
smip-agent	164/udp	CMIP/TCP Agent
xns-courier	165/tcp	Xerox
xns-courier	165/udp	Xerox
s-net	166/tcp	Sirius Systems
s-net	166/udp	Sirius Systems
namp	167/tcp	NAMP
namp	167/udp	NAMP
rsvd	168/tcp	RSVD
rsvd	168/udp	RSVD
send	169/tcp	SEND
send	169/udp	SEND
print-srv	170/tcp	Network PostScript
print-srv	170/udp	Network PostScript
multiplex	171/tcp	Network Innovations Multiplex
multiplex	171/udp	Network Innovations Multiplex
cl/1	172/tcp	Network Innovations CL/1
cl/1	172/udp	Network Innovations CL/1
xyplex-mux	173/tcp	Xyplex
xyplex-mux	173/udp	Xyplex
mailq	174/tcp	MAILQ

ΠΑΡΑΡΤΗΜΑ 3
ΕΠΙΛΟΓΕΣ ΓΙΑ ΤΗΝ ΔΗΜΙΟΥΡΓΙΑ ΕΠΙΘΕΣΕΩΝ ΣΤΟ NMAP

Παραθέτονται οι επιλογές ως προς την δημιουργία των εντολών στο λογισμικό Nmap για τον τρόπο των επιθέσεων σ' ένα δίκτυο ή σύστημα. Εάν πληκτρολληθεί στην γραμμή εντολών nmap –h , εμφανίζονται:

1) –sV: Ανιχνεύει ανοιχτές θύρες για τον προσδιορισμό πληροφοριών για τις υπηρεσίες / εκδόσεις.
2) -sT : Είναι η default επιλογή στο nmap για την σάρωση θυρών.Η τεχνική αυτή όνομαζεται TCP Connect και είναι εύκολα ανιχνεύσιμη από τον στόχο.
3) -sS :Με την τεχνική αυτή το nmap σαρώνει όπως παραπάνω τις επιλεγμένες πόρτες με την διαφορά ότι δεν υλοποιεί μία πλήρης TCP σύνδεση.Η τεχνική αυτή ονομάζεται TCP SYN ή μισάνοιχτη σάρωση(half-open scanning) και έχει το πλεονέκτημα ότι δεν είναι τόσο εύκολα ανιχνεύσιμη όσο η TCP Connect.
   Με την εντολή αυτή το nmap κάνει ότι και στην παραπάνω αλλά με διαφορετική τεχνική και ποιο "αθόρυβα".
4) -sU : Με την επιλογή αυτή το nmap μπορεί να σαρώσει τις θύρες που χρησιμοποιούν το προτόκωλλο UDP ή απλά τις UDP θύρες. Το nmap μας έβγαλε τις θύρες UDP που είναι ανοιχτές στο σύστημα-στόχος.
5) Επιλογή -sP :Με την επιλογή αυτή το nmap σαρώνει ένα δίκτυο με την τεχνική ping για να βρει ποια συτήματα είναι σε λειτουργία. Πρόκειται δηλ. για σάρωση ping με την οποία στέλνονται σε κάθε σύστημα ICMP πακέτα ECHO έτσι ώστε να απαντήσει το σύστημα με ένα άλλο ICMP πακέτο ECHO_REPLY το οποίο υποδεικνύει ότι το σύστημα είναι σε λειτουργία. Ωστόσο μπορείτε να σαρώσετε παραπάνω από ενα σύστημα και να βρείτε ποια είναι τα συστήματα σε λειτουργία.
6) -sF : Με αυτή την επιλογή στέλνετε στις θύρες του στόχου ένα πακέτο FIN. Η τεχνική αυτή δουλεύει συνήθως σε μόνο σε UNIX συστήματα.
7) -sX :Η τεχνική αυτή στέλνει ένα πακέτο FIN,URG και PUSH.

8) -sN : Με την τεχνική αυτή το απενεργοποιεί όλες τις σημαίες(ή σημάνσεις) περιμένοντας από το σύστημα-στόχο ένα πακέτο RST για όλες τις κλειστές θύρες.
9) -O :Με την επιλογή αυτή το nmap μπορεί να εξακριβώσει την ταυτότητα του λειτουργικού συτήματος του υπολογιστή στόχου.
10) -p :Όπως θα προσέξατε και παραπάνω με τον διακόπτη -p λέμε στο nmap ποιες πόρτες θέλουμε ακριβώς να σαρώσει.
11) -F :Με την επιλογή αυτή το nmap σαρώνει το στόχο μόνο για τις πόρτες που στις οποίες τρέχουν κάποια services(υπηρεσίες).
12) -v: Χρησιμοποιήστε αυτή την για να έχετε μία πιο λεπτομερή έξοδο από το nmap. Μπορείτε να χρησιμοποιήσετε και -vv για ακόμα πιο λεπτομερή.
13) -P0 :Η επιλογή αυτή χρησιμοποιείτε για να σαρώσετε ενα domain χωρίς όμως να κάνετε ping.
14) -D :Με την επιλογή αυτή μπορούμε να μπερδέψουμε το σύστημα-στόχο από που έρχεται η σάρωση μπερδεύοντας την αληθινή IP μας με άλλες έγκυρες.
15) -I :Με την επιλογή αυτή μπορούμε να σαρώσουμε τον στόχο βρίσκοντας τον κάτοχο(Owner) της υπηρεσίας που τρέχουν στον συγκεκριμένο στόχο. Αυτή η τεχνική είναι περισσότερο χρήσιμη σε συστήματα UNIX.
16) -oN :Με την επιλογή αυτή μπορείτε να αποθηκεύσετε τα αποτελέσματα του nmap σε ένα αρχείο της επιλογή σας.

ΠΑΡΑΡΤΗΜΑ 4

ΑΠΟΤΕΛΕΣΜΑΤΑ ΤΟΥ SNORT

Παραθέτονται τα στατιστικά στοιχεία όπως καταγράφονται από το λογισμικό, μετά την ολοκλήρωση της εκτέλεσής του. Τα κυριότερα είναι τα στατιστικά χρόνου, τα συνολικά πακέτα I/O, τα στατιστικά με βάση τα πρωτόκολλα και τέλος στατιστικά για το IDS mode.

193 out of 1024 flowbits in use.

[Port Based Pattern Matching Memory]
+- [Aho- Corasick Summary] ----------
| Storage Format : Full-Q
| Finite Automaton: DFA
| Alphabet Size : 256 Chars
| Sizeof State : Variable (1,2,4 bytes)
| Instances : 149
| 1 byte states : 138
| 2 byte states : 11
| 4 byte states : 0
| Characters : 46305
| States : 35868
| Transitions : 3370467
| State Density : 36.7%
| Patterns : 3119
| Match States : 2992
| Memory (MB) : 17.73
| Patterns : 0.23
| Match Lists : 0.34
| DFA
| 1 byte states : 0.78
| 2 byte states : 16.24
| 4 byte states : 0.00
+----------
[Number of patterns truncated to 20 bytes: 367]
pcap DAQ configured to passive.
The DAQ version does not support reload.
Acquiring network traffic from "\Device\NPF_{658F32D3-84B3-42D8-A51D-DE328ED90374}".
Decoding Ethernet

--== Initialization Complete ==--

```
   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.4.1-WIN32 GRE (Build 69)
   ""      By Martin Roesch & The Snort Team: http://www.snort.org/snort/snort-t
eam
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using PCRE version: 8.10 2010-06-25
           Using ZLIB version: 1.2.3

           Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 1.16  <Build 18>
           Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
           Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
           Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
           Preprocessor Object: SF_SIP  Version 1.1  <Build 1>
           Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
           Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
           Preprocessor Object: SF_POP  Version 1.0  <Build 1>
           Preprocessor Object: SF_MODBUS  Version 1.1  <Build 1>
           Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
           Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
           Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
           Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
           Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
           Preprocessor Object: SF_DCERPC2  Version 1.0  <Build 3>
Commencing packet processing (pid=6408)
*** Caught Int-Signal
===============================================================
================
Run time for packet processing was 76833.454000 seconds
Snort processed 2739194 packets.
Snort ran for 0 days 21 hours 20 minutes 33 seconds
   Pkts/hr:       130437
 Pkts/min:         2139
 Pkts/sec:           35
===============================================================
================
Packet I/O Totals:
    Received:      2740892
    Analyzed:      2739194 ( 99.938%)
     Dropped:         1693 (  0.062%)
      Filtered:          0 (  0.000%)
Outstanding:         1698 (  0.062%)
     Injected:           0
===============================================================
================
Breakdown by protocol (includes rebuilt packets):
```

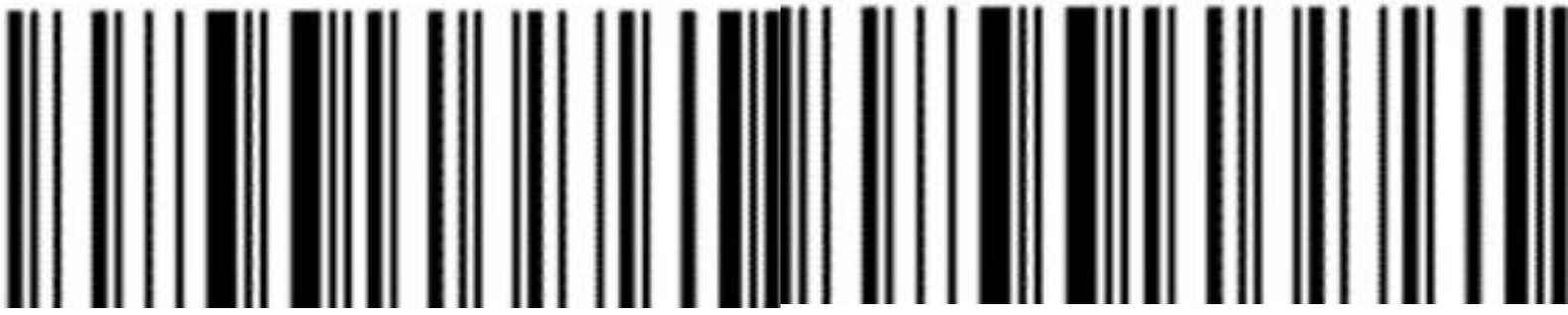
```
        Eth:      2739194 (100.000%)
       VLAN:            0 (  0.000%)
        IP4:      2606669 ( 95.162%)
       Frag:            0 (  0.000%)
       ICMP:            8 (  0.000%)
        UDP:       162879 (  5.946%)
        TCP:      2442266 ( 89.160%)
        IP6:        40210 (  1.468%)
    IP6 Ext:            0 (  0.000%)
   IP6 Opts:            0 (  0.000%)
      Frag6:            0 (  0.000%)
      ICMP6:            0 (  0.000%)
       UDP6:            0 (  0.000%)
       TCP6:            0 (  0.000%)
     Teredo:            0 (  0.000%)
   ICMP-IP:             0 (  0.000%)
      EAPOL:            0 (  0.000%)
   IP4/IP4:             0 (  0.000%)
   IP4/IP6:             0 (  0.000%)
   IP6/IP4:             0 (  0.000%)
   IP6/IP6:             0 (  0.000%)
       GRE:             0 (  0.000%)
   GRE Eth:             0 (  0.000%)
  GRE VLAN:             0 (  0.000%)
   GRE IP4:             0 (  0.000%)
   GRE IP6:             0 (  0.000%)
GRE IP6 Ext:            0 (  0.000%)
   GRE PPTP:            0 (  0.000%)
    GRE ARP:            0 (  0.000%)
    GRE IPX:            0 (  0.000%)
   GRE Loop:            0 (  0.000%)
       MPLS:            0 (  0.000%)
        ARP:        48821 (  1.782%)
        IPX:            9 (  0.000%)
   Eth Loop:            0 (  0.000%)
   Eth Disc:            0 (  0.000%)
   IP4 Disc:            0 (  0.000%)
   IP6 Disc:            0 (  0.000%)
   TCP Disc:            0 (  0.000%)
   UDP Disc:            0 (  0.000%)
  ICMP Disc:            0 (  0.000%)
All Discard:            0 (  0.000%)
      Other:        45001 (  1.643%)
Bad Chk Sum:       757273 ( 27.646%)
    Bad TTL:            0 (  0.000%)
     S5 G 1:            0 (  0.000%)
```

```
    S5 G 2:           0 (  0.000%)
     Total:      2739194
===================================================================
================
Action Stats:
     Alerts:           4 (  0.000%)
     Logged:           4 (  0.000%)
     Passed:           0 (  0.000%)
Limits:
      Match:           0
      Queue:           0
        Log:           0
      Event:           0
      Alert:           0
Verdicts:
      Allow:      2735433 ( 99.801%)
      Block:           0 (  0.000%)
    Replace:           0 (  0.000%)
  Whitelist:        3761 (  0.137%)
  Blacklist:           0 (  0.000%)
     Ignore:           0 (  0.000%)
===================================================================
================
Frag3 statistics:
        Total Fragments: 0
      Frags Reassembled: 0
              Discards: 0
         Memory Faults: 0
              Timeouts: 0
              Overlaps: 0
             Anomalies: 0
                Alerts: 0
                 Drops: 0
     FragTrackers Added: 0
    FragTrackers Dumped: 0
FragTrackers Auto Freed: 0
    Frag Nodes Inserted: 0
     Frag Nodes Deleted: 0
===================================================================
================
Stream5 statistics:
           Total sessions: 208040
             TCP sessions: 197945
             UDP sessions: 10095
            ICMP sessions: 0
              IP sessions: 0
```

TCP Prunes: 0
UDP Prunes: 0
ICMP Prunes: 0
IP Prunes: 0
TCP StreamTrackers Created: 197944
TCP StreamTrackers Deleted: 197944
TCP Timeouts: 2
TCP Overlaps: 0
TCP Segments Queued: 0
TCP Segments Released: 0
TCP Rebuilt Packets: 0
TCP Segments Used: 0
TCP Discards: 15317
TCP Gaps: 0
UDP Sessions Created: 10095
UDP Sessions Deleted: 10095
UDP Timeouts: 0
UDP Discards: 0
Events: 194
Internal Events: 0
TCP Port Filter
Dropped: 0
Inspected: 0
Tracked: 1696619
UDP Port Filter
Dropped: 0
Inspected: 0
Tracked: 10095
==========
==========
HTTP Inspect - encodings (Note: stream-reassembled packets included):
POST methods: 0
GET methods: 0
HTTP Request Headers extracted: 0
HTTP Request cookies extracted: 0
Post parameters extracted: 0
HTTP Response Headers extracted: 247
HTTP Response cookies extracted: 40
Unicode: 0
Double unicode: 0
Non-ASCII representable: 0
Directory traversals: 0
Extra slashes ("//"): 0
Self-referencing paths ("./"): 0
HTTP Response Gzip packets extracted: 0
Gzip Compressed Data Processed: n/a

```
   Gzip Decompressed Data Processed:     n/a
   Total packets processed:              1467019
===============================================================
================
SMTP Preprocessor Statistics
  Total sessions                        : 0
  Max concurrent sessions               : 0
===============================================================
================
dcerpc2 Preprocessor Statistics
  Total sessions: 0
===============================================================
================
SSL Preprocessor:
   SSL packets decoded: 124
         Client Hello: 0
         Server Hello: 42
          Certificate: 21
          Server Done: 59
   Client Key Exchange: 0
   Server Key Exchange: 0
        Change Cipher: 43
             Finished: 0
    Client Application: 0
    Server Application: 29
                Alert: 15
  Unrecognized records: 38
  Completed handshakes: 0
       Bad handshakes: 0
     Sessions ignored: 29
    Detection disabled: 15
===============================================================
================
SIP Preprocessor Statistics
  Total sessions: 0
===============================================================
================
Reputation Preprocessor Statistics
  Total Memory Allocated: 0
===============================================================
================
Snort exiting
```

ΑΝΑΦΟΡΕΣ



ΒΙΒΛΙΟΓΡΑΦΙΑ

- Stephen Northcutt-Judy Novak, Network Intrusion Detection Third Edition, Sept 2002. ISBN: 0-73571-265-4.
- Γ. Πάγκαλος- Ι. Μαυρίδης, Ασφάλεια Πληροφοριακών συστημάτων και δικτύων, Εκδόσεις ΑΝΙΚΟΥΛΑ ΘΕΣΣΑΛΟΝΙΚΗ 2002. ISBN: 960-516-018-8
- Θόδωρος Κομνηνός- Παύλος Σπυράκης, Ασφάλεια Δικτύων & Υπολογιστικών Συστημάτων Αναχαιτίστε τους εισβολείς, Εκδόσεις Ελληνικά Γράμματα Αθήνα 2002. ISBN: 960-406-225-5
- Karen Scarfone, Peter Mell, Guide to Intrusion Detection and Prevention System (IDPS) February 2007. National Institute of Standards and Technology.
- Στέφανος Γκρίτζαλης- Σωκράτης Κ. Κάτσικας- Δημήτρης Γκρίτζαλης, Ασφάλεια Δικτύων Υπολογιστών, Εκδόσεις Παπασωτηρίου Αθήνα 2003. ISBN: 960-7530-45-4.
- Jack Koziol, Intrusion Detection with Snort 2Rev Edition May 20, 2003. ISBN: 978-1578702817.
- Sourcefire, Inc, Snort Users Manual 2.9.2 The Snort project, December 7, 2011.
- Brian Caswell-Jay Beale-Andrew Baker, Snort Intrusion Detection and Prevention Toolkit Pcap/Cdr Edition February 1, 2007. ISBN: 1-59749-099-7.
- Δημήτρης Πρίτσος, SLAB HACK: Βασικές Έννοιες & Προγραμματισμός του Snort 2.0, Αθήνα 2003.
- http://nmap.org/
- http://harrykar.blogspot.gr/2009/05/snort-ids-ips-nsm-andbeyond.html
- http://www.powertech.com/guides/compliance/iso_27002_(17799).html

- http://en.wikipedia.org/wiki/Intrusion_detection_system
- http://www.cse.wustl.edu/~jain/cse571-07/ftp/ids/index.html
- http://www.combofix.org/what-is-host-based-intrusion-detection.php
- http://en.wikipedia.org/wiki/Snort_(software)
- http://el.wikipedia.org/wiki/Firewall
- http://en.wikipedia.org/wiki/Pcap
- http://www.eeei.gr/interbiz/articles/dos.htm
- http://www.islab.demokritos.gr/gr/html/Snort2_dpritsos/Snort2&Snort_Preprocessors.pdf
- http://harrykar.blogspot.gr/2009/05/snort-ids-ips-nsm-andbeyond.html
- http://www.rfc-editor.org/
- http://www.snort.org/