Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών

Τμήμα Εφαρμοσμένης Πληροφορικής και Πολυμέσων

Μελέτη, σχεδιασμός και ενσωμάτωση συστήματος ανίχνευσης δικτυακών επιθέσεων σε εργαστηριακό περιβάλλον.

Πτυχιακή Εργασία

Παλαντάς Παναγιώτης Α.Μ.: 174

ΗΡΑΚΛΕΙΟ 2007

Περιεχόμενα

Πρόλογος – Περιγραφή της Πτυχιακής Εργασίας

Ο βασικός στόχος της πτυχιακής είναι η μελέτη, ο σχεδιασμός και η ενσωμάτωση λογισμικού συστήματος ανίχνευσης δικτυακών επιθέσεων (NIDS) ώστε να μπορούν να αναγνωρίζονται οι επιθέσεις που γίνονται στο πειραματικό μας δίκτυο, καθώς και να εντοπίζονται τα δίκτυα που τις προκαλούν.

Η επιλογή των συστημάτων θα γίνει μέσω μελέτης των υπαρχόντων συστημάτων λογισμικού ανίχνευσης με κριτήρια όπως το κόστος, οι δυνατότητες τους καθώς και η ευκολία που μπορούμε να τα προσαρμόσουμε στο πειραματικό μας περιβάλλον ώστε να μπορέσουμε να μετρήσουμε την αξιοπιστία τους. (Συστήματα που απαιτούν ειδικό Hardware δεν θα εξεταστούν)

Σημαντικό μέρος της πτυχιακής εργασίας είναι ο σχεδιασμός του εργαστηριακού μας περιβάλλοντος ώστε να μας παρέχει την δυνατότητα να συμπεριφέρεται σαν πραγματικό δίκτυο, πάνω στο οποίο θα εργαστούμε για να εξάγουμε τα συμπεράσματα μας.

Ευχαριστίες.

Για την πραγματοποίηση της πτυχιακής εργασίας, πολύτιμη η βοήθεια του Κέντρου Ελέγχου και Διαχείρισης Δικτύου του ΤΕΙ Κρήτης και συγκεκριμένα του Ph.D. Μ. Βούρκα για την βοήθεια του στον δικτυακό τομέα καθώς και του Αναπληρωτή καθηγητή Ph.D Κ. Βασιλακή για την βοήθεια του στην ανάπτυξη του όλου εγχειρήματος.

1. Ασφάλεια Υπολογιστών

Με τον όρο ασφάλεια υπολογιστών εννοούμε το πεδίο της επιστήμης υπολογιστών που αφορά τον έλεγχο των κινδύνων που σχετίζονται με την χρήση των υπολογιστών. Σκοπός της είναι να καταφέρει να δημιουργήσει μια ασφαλή υπολογιστική πλατφόρμα, σχεδιασμένη ώστε οι εξουσιοδοτημένοι χρήστες ή προγράμματα να μπορούν να εκτελούν τις εξουσιοδοτημένες τους λειτουργίες. Το οποίο περιλαμβάνει τον καθορισμό και την εγκαθίδρυση πολιτικών ασφαλείας (Security Policy). Η ασφάλεια υπολογιστών μπορεί να θεωρηθεί ως ένας υποτομέας της Security Engineering, η οποία ασχολείται με ένα ευρύτερο μέρος τον θεμάτων ασφαλείας σε αντιπαράθεση με την ασφάλεια υπολογιστών.

1.1 Τι εννοούμε όταν λέμε ασφάλεια.

Όταν μιλάμε για ασφάλεια υπάρχουν τρία βασικά σημεία που επεξηγούν τι είναι ασφάλεια:

1) Εμπιστευτικότητα (Confidentiality), η προστασία της πληροφορίας από μη εξουσιοδοτημένη πρόσβαση.
2) Ακεραιότητα (Integrity), η προστασία της πληροφορίας από μη εξουσιοδοτημένη τροποποίηση.
3) Διαθεσιμότητα (Availability), τα δεδομένα υπάρχουν στη θέση, στον χρόνο και στην μορφή που ο χρήστης τα χρειάζεται.

1.2 Γιατί χρειαζόμαστε ασφάλεια.

Με την πάροδο του χρόνου οι υπολογιστές χρησιμοποιούνται για να αποθηκεύσουν, να επεξεργαστούν, να προβάλουν και να μεταφέρουν ευαίσθητα δεδομένα. Για να μπορέσει να λειτουργήσει ένα τέτοιο σύστημα πρέπει να γίνει ανάλυση των επιπέδων ασφάλειας που θα πρέπει να έχει, ώστε να μπορεί να θεωρηθεί λειτουργικό. Αναφερόμαστε στο επίπεδο ασφάλειας που θα πρέπει να παρέχει γιατί κάθε σύστημα χρειάζεται διαφορετικά επίπεδα ασφάλειας τα οποία για να προκύψουν πρέπει να γίνει ανάλυση των κινδύνων του συγκεκριμένου συστήματος, να μετρηθούν

τα θετικά και τα αρνητικά σημεία του κάθε συστήματος ασφαλείας καθώς και να γίνει εκτίμηση του κόστους εγκατάστασης των συστημάτων ασφαλείας. Σκοπός της ασφάλειας υπολογιστών είναι να προστατέψει τα πολύτιμα στοιχεία του κάθε οργανισμού, δηλ. την πληροφορία, το υλικό και το λογισμικό, με την επιλογή και την εφαρμογή κατάλληλης περιφρούρησης. Τα τελευταία χρόνια με την τεράστια εξέλιξη της συνδεσιμότητας των συστημάτων και την χρήση του διαδικτύου (Internet) τα υπολογιστικά συστήματα τείνουν να επιβεβαιώνουν ότι οι τρεις αρχές της ασφάλειας δεν είναι εύκολο να διατηρηθούν.

1.3 Απειλές.

Τα υπολογιστικά συστήματα είναι ευάλωτα σε πολλές απειλές που μπορούν να προκαλέσουν διαφόρων τύπων ζημιές με αποτέλεσμα σημαντικές απώλειες. Η ζημιά μπορεί να είναι κάποιο λάθος που κατέστρεψε την ακεραιότητα μίας βάσης δεδομένων μέχρι μια φωτιά που κατέστρεψε ένα κέντρο υπολογιστών. Οι απώλειες μπορούν να προέλθουν, παραδείγματος χάριν, από τις ενέργειες των υποθετικά έμπιστων υπαλλήλων ή κακόβουλων χάκερ (hacker). Το να γίνει ακριβής υπολογισμός των απωλειών λόγω ελλιπής ασφάλειας δεν είναι πάντοτε δυνατή λόγω του ότι πολλές φορές δεν ανακαλύπτονται ενώ άλλες αποκρύβονται με σκοπό να μην υπάρξει δυσμενής δημοσιότητα του οργανισμού. Τα αποτελέσματα των διαφόρων απειλών ποικίλλουν αρκετά: μερικές έχουν επιπτώσεις στην εμπιστευτικότητα ή την ακεραιότητα των πληροφοριών ενώ άλλες στην διαθεσιμότητα του συστήματος.

Όπως γίνετε φανερό από τα παραπάνω οι απειλές για την ασφάλεια ενός υπολογιστικού συστήματος μπορούν να κατηγοριοποιηθούν σε:

α) κακόβουλες απειλές
β) μη σκόπιμες απειλές
γ) υλικές απειλές

Οι κατηγορίες αυτές μπορούν να αναπτυχθούν ακόμα περισσότερο χωρίζοντας τις σε υποκατηγορίες και παρουσιάζονται στο παράρτημα.

1.4 Ασφάλεια Δικτύου.

Είναι ένα από τα πολλά θέματα που ασχολείται η ασφάλεια υπολογιστών και αφορά την προστασία των δικτύων και των υπηρεσιών από μη εξουσιοδοτημένη τροποποίηση, καταστροφή, ή κοινοποίηση. Παρέχει εξασφάλιση ότι το δίκτυο εκτελεί τις κρίσιμες λειτουργίες του σωστά και δεν υπάρχουν επιβλαβής παρενέργειες.

1.4.1 Τρόποι αύξησης επίπεδου ασφάλειας.

Οι παρακάτω τρόποι μπορούν να χρησιμοποιηθούν για να κατασκευάσουμε ασφαλή συστήματα, παρ' όλα αυτά από μόνοι τους δεν μπορούν να εξασφαλίσουν την ασφάλεια.

- Κρυπτογραφικές τεχνικές μπορούν να χρησιμοποιηθούν για να ασφαλίσουν την πληροφορία που μεταδίδεται μεταξύ συστημάτων.
- Ισχυροί μηχανισμοί πιστοποίησης, ώστε να είναι εξασφαλισμένο ότι τα άκρα που συμμετέχουν στην επικοινωνία είναι αυτοί που ισχυρίζονται ότι είναι.
- Αντίγραφα ασφαλείας,
- Λογισμικό απομάκρυνσης ιών.(Anti-virus software)
- Τοίχος προστασίας (Firewall)
- Γνώση πρακτικών δημοσίων σχέσεων (Social Engineering), να μην δίνονται εμπιστευτικές πληροφορίες σε περιπτώσεις οπού δεν μπορεί να πιστοποιηθεί η ταυτότητα του αιτούντος.
- Να μην γίνετε χρήση λογισμικού που έχει γνωστοποιημένα προβλήματα ασφαλείας και δεν έχει γίνει ενημέρωση του.
- Συστήματα ανίχνευσης εισβολών (Intrusion Detection Systems).

2. Συστήματα Ανίχνευσης Εισβολών (Intrusion Detection Systems - IDS)

Είναι συστήματα λογισμικού (Software) ή υλικό (Hardware) που αυτοματοποιούν τη διαδικασία παρακολούθησης συμβάντων που συμβαίνουν σε υπολογιστικό σύστημα ή δίκτυο, αναλύοντας τα για σημάδια προβλημάτων ασφαλείας. Πιο συγκεκριμένα σκοπεύουν στο να εντοπίζουν επιθέσεις και/ή κατάχρηση σε υπολογιστικά συστήματα καθώς στη συνέχεια να ενημερώνουν τα κατάλληλα άτομα για το συμβάν που εντοπίστηκε. Με τον ρυθμό που έχουν αρχίσει να αυξάνονται πλέον οι δικτυακές επιθέσεις, σε αριθμούς αλλά και σε σοβαρότητα περιστατικών, τα IDS έχουν αρχίσει να καθίστανται αναγκαία.

2.1 Γιατί να χρησιμοποιήσουμε ένα σύστημα IDS.

Η ανίχνευση εισβολών επιτρέπει σε οργανισμούς να προστατέψουν τα συστήματα τους από απειλές που δημιουργούνται λόγω της αυξημένης ζήτησης για Διαδικτύωση. Σκεπτόμενοι την φύση των σημερινών δικτύων και τις απειλές που υπάρχουν, θα έπρεπε τέτοια συστήματα να είναι μέρος κάθε οργανισμού ή υπολογιστικού συστήματος. Ορισμένοι αρκετά σημαντικοί λόγοι είναι:

- Στο να ανιχνεύονται επίθεσης και άλλες παραβίασης ασφαλείας οι οποίες δεν μπορούν να εντοπιστούν από άλλα μέτρα ασφαλείας.
- Στην επιβολή των πολιτικών ασφαλείας ενός οργανισμού. Μπορεί να παρακολουθείται το εσωτερικό δίκτυο για συμπεριφορές που παρεκκλίνουν από τις πολιτικές ασφάλειας δικτύων του οργανισμού.
- Στο να ανιχνεύονται και να αντιμετωπίζονται πρώιμες επιθέσεις όπως σαρώσεις δικτύων (network scans).
- Στο να καταγραφούν οι υπάρχουσες απειλές.
- Στο να ενεργήσουν ως ποιοτικός έλεγχος για τον ασφαλή σχεδιασμό και έλεγχο των δικτύων.
- Στο να παρέχουν χρήσιμες πληροφορίες για εισβολές που συνέβησαν, προσφέροντας βελτιωμένη διάγνωση, επαναφορά και διόρθωση των αιτιολογικών παραγόντων.

2.2 Είδη συστημάτων ανίχνευσης εισβολών.

Ο πιο κλασσικός τρόπος ταξινόμησης τους είναι με βάση την πηγή των πληροφοριών. Έχουμε λοιπόν IDS που χρησιμοποιούν πληροφορίες που προέρχονται από την παρακολούθηση της δικτυακής κίνησης αναλύοντας πακέτα ώστε να βρεθεί αν ένα σύστημα γίνετε στόχος κάποιας επίθεσης. Ενώ υπάρχουν άλλα IDS που αναλύουν πληροφορίες που εξάγονται από εφαρμογές λογισμικού ή το ίδιο το λειτουργικό σύστημα.

2.2.1 Βασιζόμενα σε δίκτυο IDS (Network-Based IDS – NIDS)

Ένα σύστημα NIDS παρακολουθεί την δικτυακή κίνηση του δικτύου που βρίσκεται και την χρησιμοποιεί για πηγή πληροφοριών. Αυτό επιτυγχάνεται με το να τεθεί η κάρτα διεπαφής δικτύου σε λειτουργία promiscous για να μπορεί "αιχμαλωτίζει" όλα τα πακέτα που κυκλοφορούν στο κομμάτι του δικτύου που βρίσκεται ακόμα και αυτά που δεν προορίζονται για το συγκεκριμένο σύστημα. Στην συνέχεια το σύστημα εξετάζει σε πραγματικό χρόνο το πακέτο ή τα πακέτα με βάση γνωστές υπογραφές κακόβουλων πακέτων ή/και διενεργεί αποκωδικοποίηση πρωτοκόλλων ώστε να ανιχνεύσει ανωμαλίες. Η βάση δεδομένων με τις υπογραφές επιθέσεων που χρησιμοποιούν τα συστήματα αυτά είναι βασικό να αναβαθμίζονται σε καθημερινή βάση, για να έχουμε τα αναμενόμενα αποτελέσματα. Μόλις ένα IDS ανιχνεύσει ύποπτη λειτουργία, υπάρχει η δυνατότητα να στείλει ειδοποίηση στον διαχειριστή δικτύου για το συμβάν αλλά και να διακόψει την ύποπτη σύνδεση.

Παραδείγματα των λειτουργιών ενός NIDS είναι τα παρακάτω:

- Παρακολούθηση του δικτύου για σαρώσεις θυρών (port scans). Συνήθως πριν την επίθεση σε ένα σύστημα ο επιτιθέμενος ελέγχει ένα σύστημα για να αποκτήσει γνώσει των υπηρεσιών που προσφέρει και να ανακαλύψει ευάλωτα σημεία του.
- Παρακολούθηση έγκυρων συνδέσεων για γνωστές επιθέσεις. Το να συνδεθούμε για παράδειγμα σε ένα διακομιστή Ιστοσελίδων στην συνηθισμένη θύρα 80 αν και μπορεί να φανεί φυσιολογικό από μια πρώτη ματιά, πολλές φορές είναι μια εσκεμμένη επίθεση, όταν αυτή περιέχει εντολές όπως "GET /../../../etc/passwd HTTP/1.0"
- Αναγνώριση προσπαθειών εξαπάτησης του πρωτοκόλλου Internet (IP). Το πρωτόκολλο ARP είναι υπεύθυνο για μετατρέπει τις διευθύνσεις του IP σε διευθύνσεις MAC και πολλές φορές

γίνετε στόχος επίθεσης. Με το να σταλούν πλαστά πακέτα ARP μέσω της κάρτας δικτύου ένας εισβολέας που έχει αποκτήσει πρόσβαση σε ένα σύστημα, μπορεί να υποδυθεί ότι είναι ένα άλλο τελείως διαφορετικό σύστημα και να υποκλέψει τα πακέτα που προορίζονταν για το άλλο έμπιστο σύστημα. Ένα IDS μπορεί να παρακολουθεί τα πακέτα ARP και να αναγνωρίσει το σύστημα που επιτίθεται ενημερώνοντας το προσωπικό.

2.2.1.1 Πλεονεκτήματα των NIDS

- Μπορούμε αν τοποθετήσουμε στα σωστά σημεία ορισμένα NIDS να παρακολουθούμε ακόμα και μεγάλα δίκτυα.
- Η τοποθέτηση NIDS δεν επηρεάζει ιδιαίτερα το υπάρχον δίκτυο μας, καθώς συνήθως είναι συστήματα που απλά παρακολουθούν το δίκτυο χωρίς να εμπλέκονται στην φυσιολογική του λειτουργία.
- Μπορούν να τοποθετηθούν με τρόπο τέτοιο ώστε να μην μπορούν να ανακαλυφθούν ή επιτεθούν από κάποιον εισβολέα.

2.2.1.2 Μειονεκτήματα των NIDS

- Σε ορισμένα δίκτυα μεγάλων ταχυτήτων είναι πιθανό το NIDS να αποτύχει στο να αναγνωρίσει κάποια επίθεση που συμβαίνει σε ώρα αιχμής. Για αυτόν τον λόγο, αρκετοί κατασκευαστές NIDS δημιουργούν hardware συστήματα που είναι σαφώς γρηγορότερα.
- Τα σύγχρονα δίκτυα που λειτουργούν σε switch με αποτέλεσμα το NIDS να μην μπορεί να παρακολουθήσει όλα τα πακέτα που διέρχονται από το switch. Υπάρχουν βέβαια και switch που παρέχουν την δυνατότητα να στέλνουν όλη την κίνηση πακέτων σε μια θύρα τους, αν και συνήθως και εκεί υπάρχουν προβλήματα.
- Δεν μπορούν να αναλύσουν κρυπτογραφημένες πληροφορίες.
- Μερικά NIDS παρουσιάζουν προβλήματα με επιθέσεις που περιλαμβάνουν τεμαχισμένα πακέτα (fragmented packets).

2.2.2 Βασιζόμενα σε οικοδεσπότη IDS (Host-Based IDS – HIDS)

Τα συστήματα HID λειτουργούν παρακολουθώντας τα αρχεία, τις καταχωρήσεις του

συστήματος και τον συμβάντων, τα βασικά αρχεία του συστήματος και γενικότερα πηγές πληροφορίας που προέρχονται από το παρακολουθούμενο σύστημα με σκοπό να εντοπίσουν μη εξουσιοδοτημένες αλλαγές ή ύποπτες δραστηριότητες. Σε περίπτωση που βρεθεί κάτι ύποπτο ειδοποιούν αυτόματα τον διαχειριστή για τα προβλήματα.

Για παράδειγμα, ένα HIDS μπορεί να παρακολουθεί:

- τις εισερχόμενες ή/και εξερχόμενες συνδέσεις δικτύου που συμβαίνουν στο συγκεκριμένο σύστημα.
- Την λειτουργία σύνδεσης και αποσύνδεσης χρηστών (login – logout) στο επίπεδο δικτύου του υπολογιστή.
- Τις λειτουργίες του υπέρ-χρήστη (super-user) του συστήματος για μη συνηθισμένη συμπεριφορά.
- Την ακεραιότητα των αρχείων του συστήματος.

2.2.2.1 Πλεονεκτήματα των HIDS

- Επαλήθευση των επιθέσεων, μιας και τα γεγονότα που καταγράφονται έχουν ήδη συμβεί στο συγκεκριμένο σύστημα.
- Πλήρης εποπτεία συστήματος, ορίζοντας συγκεκριμένες πολιτικές ασφαλείας μπορούμε να ενημερωθούμε για όλες τις αλλαγές που προκύπτουν στο σύστημα, είτε είναι νόμιμες είτε όχι.
- Δεν επηρεάζονται από κρυπτογράφηση ή από switched περιβάλλον δικτύου. Τα πακέτα μπορούν να επεξεργαστούν αμέσως μετά την αποκρυπτογράφηση τους και πριν το λειτουργικό ή η εφαρμογή τα επεξεργαστεί.
- Δεν υπάρχει η ανάγκη για ειδικό υλικό (Hardware)

2.2.2.2 Μειονεκτήματα των HIDS

- Χρησιμοποιούν τους πόρους του συστήματος.
- Είναι πιθανό να απενεργοποιηθούν από ορισμένες επιθέσεις άρνησης λειτουργίας (Denial of Service Attack – DoS).
- Είναι συνήθως πιο δύσκολα στην διαχείριση λόγω της κατανεμημένης φύσης τους.

- Εξαρτώνται από το λειτουργικό σύστημα.

2.2.3 Συμπεράσματα

Σε καμία περίπτωση δεν θα πρέπει να μπούμε στο δίλημμα να επιλέξουμε μεταξύ των δύο ειδών συστημάτων ανίχνευσης εισβολών. Το κάθε σύστημα έχει τα δικά του δυνατά αλλά και αδύνατα σημεία Μια ολοκληρωμένη λύση αποτελείται από την ορθή χρήση και των δυο συστημάτων σε ένα δίκτυο ώστε να έχουμε την καλύτερη δυνατή επίβλεψη του. Θα πρέπει επίσης να γίνει κατανοητό ότι κανένα σύστημα από μόνο του δεν παρέχει επαρκή προστασία σε ένα υπολογιστικό περιβάλλον, αλλά θα πρέπει να υποστηρίζετε από έμπειρους διαχειριστές και σαφή πολιτική ασφαλείας που θα καλύπτει λεπτομερώς τα πιθανά σενάρια που μπορούν να συμβούν.

2.3 Μέθοδοι Ανίχνευσης των NIDS.

Κάθε σύστημα ανίχνευσης εισβολών προσφέρει στον χρήστη διαφορετικούς τρόπους για να ελέγξει αν το πακέτα που "κυκλοφορούν" στο δίκτυο είναι κακόβουλα ή όχι. Παρακάτω θα αναφερθούμε στις διάφορες μεθοδολογίες που έχουν αναπτυχθεί και χρησιμοποιούνται σήμερα στα διάφορα συστήματα ανίχνευσης εισβολών σύμφωνα με την Cisco "The science of Intrusion Detection System Attack Identification".

2.3.1 Ταίριασμα Προτύπων (Pattern Matching)

Η μεθοδολογία ταιριάσματος προτύπων βασίζεται στον έλεγχο μιας σειράς από bytes μέσα σε ένα και μόνο πακέτο και αντιπαραθέτοντας την με τα πρότυπα κακόβουλων πακέτων. Στις περισσότερες περιπτώσεις για να ελαττωθεί ο αριθμός των εξεταζόμενων πακέτων, ο έλεγχος γίνεται μόνο στην περίπτωση όπου το "ύποπτο" πακέτο προέρχεται ή κατευθύνεται σε συγκεκριμένη υπηρεσία. Βέβαια μια τέτοια προσέγγιση δημιουργεί προβλήματα στην περίπτωση που χρησιμοποιούνται πρωτόκολλα σε μη κλασσικές θύρες.

Ένα παράδειγμα ενός κανόνα ταιριάσματος προτύπων είναι:

➔ Αν το πακέτο είναι τύπου Ipv4 και TCP και η θύρα προορισμού η 3306 και το περιέχει το αλφαριθμητικό "root" τότε έκδωσε συναγερμό.

Το παραπάνω παράδειγμα είναι από τα πιο απλά που μπορούμε να έχουμε και βέβαια μπορούμε να ορίσουμε περισσότερες πληροφορίες για τον κανόνα της υπογραφής.

Πλεονεκτήματα.

- Είναι η πιο απλή μέθοδος ελέγχου εισβολών.
- Επιτρέπει την άμεση συσχέτιση ενός προβλήματος ασφάλειας (exploit) με ένα πρότυπο.
- Ειδοποιεί αξιόπιστα για το συγκεκριμένο πρότυπο.
- Είναι εφαρμόσιμη σε όλα τα πρωτόκολλα.

Μειονεκτήματα.

- Μπορεί να οδηγήσει σε υψηλά επίπεδα λάθος προειδοποιήσεων, στην περίπτωση που το πρότυπο δεν είναι όσο μοναδικό όσο θα έπρεπε.
- Αλλαγή στον τρόπο επίθεσης ενδέχεται να προκαλέσει χαμένα συμβάντα.
- Η συγκεκριμένη μέθοδος είναι πιθανόν να χρειάζεται περισσότερες από μια υπογραφές για να περιγραφεί το ίδιο πρόβλημα ασφάλειας.
- Συνήθως χρησιμοποιείται σε ένα και μόνο πακέτο, με αποτέλεσμα να μην έχει τα επιθυμητά αποτελέσματα σε δίκτυα βασισμένα σε ροή δεδομένων όπως η κίνηση HTTP.

2.3.2 Stateful Ταιριασμα Προτύπων (Stateful Pattern Matching)

Είναι μια πιο εξελιγμένη μέθοδος που προσθέτει στο Ταίριασμα Προτύπων την ιδέα ότι αφού μια δικτυακή ροή αποτελείται από περισσότερων του ενός πακέτου, θα πρέπει ληφθεί υπ' όψιν στην αντιστοίχιση. Προκύπτει επομένως, ότι τα συστήματα που χρησιμοποιούν την συγκεκριμένη μεθοδολογία ανίχνευσης πρέπει να εξετάζουν την σειρά άφιξης των πακέτων σε μια ροή TCP και να ελέγχουν για ταίριασμα προτύπων σε συνολικά τα πακέτα της ροής.

Στην συνέχεια ας αναλύσουμε την μέθοδο με βάση το παράδειγμα του ταιριάσματος προτύπων. Στην συγκεκριμένη περίπτωση το σύστημα μας αντί να ελέγχει για πρότυπα σε κάθε πακέτο, διατηρεί πληροφορίες κατάστασης για την ροή TCP που παρακολουθείται. Αν λοιπόν, χρησιμοποιώντας το ταίριασμα προτύπων, συμβεί μια επίθεση κατά την οποία ο επιτιθέμενος στείλει σε ένα πακέτο με θύρα προορισμού την 3306 και το αλφαριθμητικό “root” ο συναγερμός θα

σημάνει. Αν όμως ο επιτιθέμενος στείλει το αλφαριθμητικό “roo” στο πρώτο πακέτο και στη συνέχεια σε ένα δεύτερο πακέτο το “t” το σύστημα δεν δώσει ειδοποίηση. Λύση για το πρόβλημα μας δίνει το stateful ταίριασμα προτύπων το οποίο αποθηκεύει το “roo” και μόλις έρθει και το υπόλοιπο μέρος του αλφαριθμητικού κάνει τον έλεγχο προτύπου.

Πλεονεκτήματα

- Δεν απαιτεί πολύ περισσότερη δυσκολία για να εφαρμοστεί σε σχέση με το απλό ταίριασμα προτύπων
- Επιτρέπει την άμεση συσχέτιση ενός προβλήματος ασφάλειας (exploit) με ένα πρότυπο.
- Ειδοποιεί αξιόπιστα για το συγκεκριμένο πρότυπο.
- Είναι εφαρμόσιμη σε όλα τα πρωτόκολλα.
- Κάνει την προσπάθεια αποφυγής λίγο δυσκολότερη

Μειονεκτήματα

- Μπορεί να οδηγήσει σε υψηλά επίπεδα λάθος προειδοποιήσεων, στην περίπτωση που το πρότυπο δεν είναι όσο μοναδικό όσο θα έπρεπε.
- Αλλαγή στον τρόπο επίθεσης ενδέχεται να προκαλέσει χαμένα συμβάντα.
- Η συγκεκριμένη μέθοδος είναι πιθανόν να χρειάζεται περισσότερες από μια υπογραφές για να περιγραφεί το ίδιο πρόβλημα ασφάλειας.

2.3.3 Αποκωδικοποίηση Πρωτοκόλλου (Protocol Decode)

Η συγκεκριμένη μεθοδολογία εφαρμόζεται αποκωδικοποιώντας τα διάφορα στοιχεία με τρόπο ίδιο με αυτόν που ο πελάτης (client) ή ο εξυπηρετητής (server) θα έκαναν κατά την διάρκεια της “συνομιλίας” τους. Στην συνέχεια, τα στοιχεία του πρωτοκόλλου προσδιορίζονται και το σύστημα ανίχνευσης εισβολών εφαρμόζει τους κανόνες που προκαθορίζονται από το RFC (Request for Comments) ελέγχοντας για παραβιάσεις. Σε ορισμένες περιπτώσεις, οι παραβιάσεις αυτές βρίσκονται χρησιμοποιώντας το ταίριασμα προτύπων σε συγκεκριμένα πεδία του πρωτοκόλλου, ενώ συχνά γίνεται χρήση πιο εξελιγμένων τεχνικών που ελέγχουν το μέγεθος των μεταβλητών ενός πεδίου ή τον αριθμό των ορισμάτων.

Αν θεωρήσουμε για παράδειγμα ότι η επιτυχία της επίθεσης βασίζεται στην εκμετάλλευση

του υποθετικού πρωτοκόλλου ABΓ, και πιο συγκεκριμένα, η επίθεση απαιτεί να περάσουμε το όρισμα “foo” στον πεδίο τύπου (Field Type) του ABΓ. Για να γίνει ακόμα πιο πολύπλοκη η κατάσταση, θεωρούμε ότι πριν το πεδίο τύπου υπάρχει ένα πεδίο επιλογών κυμαινόμενου μήκους, με δυνατές τιμές τις επιλογές fooh, moo και urgent. Αν χρησιμοποιήσουμε τις προαναφερόμενες μεθόδους ανίχνευσης, θα παρατηρήσουμε ότι το σύστημα δημιουργεί πολλούς λανθασμένους συναγερμούς αφού το πεδίο fooh περιλαμβάνει το πρότυπο που ελέγχεται. Επίσης λόγω του ότι το πεδίο δεν έχει σταθερό αλλά έχει κυμαινόμενο μήκος μας αποτρέπει από το να ορίσουμε αρχικό και τελικό σημείο ελέγχου για το πρότυπο μας. Ο μόνος τρόπος, λοιπόν, για να ανιχνευτεί μια τέτοια επίθεση είναι να γίνει πλήρης αποκωδικοποίηση στο συγκεκριμένο πρωτόκολλο.

Πλεονεκτήματα

- Ελαχιστοποιούνται οι πιθανότητες λανθασμένων θετικών συναγερμών, αν το πρωτόκολλο είναι σωστά ορισμένο.
- Επιτρέπει την άμεση συσχέτιση ενός προβλήματος ασφάλειας (exploit) με ένα πρότυπο.
- Είναι πιο γενική με αποτέλεσμα να ανιχνεύει παραλλαγές της επίθεσης.
- Ενημερώνει αξιόπιστα για παραβιάσεις των κανόνων ενός πρωτοκόλλου.

Μειονεκτήματα

- Μπορεί να οδηγήσει σε υψηλά επίπεδα λανθασμένων συναγερμών, αν το RFC δεν είναι σωστά ορισμένο και επιτρέπει στους προγραμματιστές που το αναπτύσσουν να το εφαρμόζουν όπως είναι πιο βολικό για αυτούς.
- Η συγκεκριμένη μέθοδος παρουσιάζει μεγαλύτερους χρόνους ανάπτυξης συστημάτων ώστε να υλοποιηθεί σωστά ο αναλυτής πρωτοκόλλου (Protocol Parser).

2.3.4 Ευρετική Ανάλυση (Heuristic-Based Analysis)

Οι υπογραφές που βασίζονται στην ευρετική ανάλυση χρησιμοποιούν διαφόρων ειδών λογικούς αλγόριθμους για να βασίσουν την επιλογή συναγερμών. Οι αλγόριθμοι αυτοί είναι πολλές φορές στατιστικές εκφράσεις του τύπου της κίνησης που παρουσιάζεται. Για να γίνει πιο κατανοητό, ένα παράδειγμα τέτοιας υπογραφής είναι μια υπογραφή που χρησιμοποιείται για να αναγνωριστεί η σάρωση θυρών σε ένα σύστημα. Η συγκεκριμένη υπογραφή ελέγχει για την ύπαρξη

ενός κατώτατου αριθμού μοναδικών θυρών που ζητούνται σε ένα συγκεκριμένο σύστημα. Μια τέτοια υπογραφή μπορεί να γίνει ακόμα πιο ακριβής καθορίζοντας το είδος των πακέτων, τον χρόνο ή αν τα πακέτα έχουν κοινή αρχή. Τέτοιου είδους υπογραφές χρειάζονται τροποποιήσεις στον έλεγχο κατώτερων τιμών, ώστε να αντικατοπτρίζουν την χρήση του δικτύου που παρακολουθούν. Η ευρετική ανάλυση έχει την δυνατότητα να επεξεργάζεται πολύπλοκες σχέσεις καθώς και απλά στατιστικά παραδείγματα όπως το παραπάνω.

Πλεονεκτήματα

- Ανακαλύπτει ύποπτη δραστηριότητα που δεν θα μπορούσε να ανιχνευτεί με άλλους τρόπους.

Μειονεκτήματα

- Οι αλγόριθμοι συνήθως χρειάζονται αλλαγές για να ελαχιστοποιηθούν τα λανθασμένα θετικά αποτελέσματα, ανάλογα με το δίκτυο.
- Περισσότερη επεξεργαστική ισχύς.

2.3.5 Ανάλυση Ανωμαλιών (Anomaly Analysis)

Οι υπογραφές βασισμένες στην ανάλυση ανωμαλιών χαρακτηρίζονται από τον έλεγχο της δικτυακής κίνησης που αποκλίνει από αυτό που θεωρείται “φυσιολογική” κίνηση. Ένα από τα μεγαλύτερα προβλήματα της συγκεκριμένης προσέγγισης είναι αρχικά το να οριστεί το τι θεωρείται “φυσιολογικό”. Συστήματα τα οποία έχουν προγραμματιστεί να συγκεκριμένους ορισμούς του “φυσιολογικού” πρέπει να χαρακτηρίζονται ως συστήματα βασιζόμενα στην ευρετική ανάλυση. Υπάρχουν όμως συστήματα που σχεδιάζονται με την δυνατότητα να μαθαίνουν τι θεωρείται “φυσιολογική” δικτυακή κίνηση και τι όχι. Η πρόκληση σε αυτά τα συστήματα είναι να εξαλειφθεί η πιθανότητα λάθους κατά την ταξινόμηση της φυσιολογικής δραστηριότητας, καθώς επίσης και το πως το σύστημα μπορεί να διαφοροποιηθεί μεταξύ των επιτρεπτών αποκλίσεων, αυτών που δεν επιτρέπονται και αυτών που αποτελούν κίνηση πληροφοριών λόγω επίθεσης.

Η συγκεκριμένη μεθοδολογία έχει ακόμη αρκετά προβλήματα να λύσει μέχρι να θεωρηθεί ώριμη, αν και υπάρχουν εμπορικά πακέτα που υποστηρίζουν ότι την χρησιμοποιούν. Δυστυχώς όμως ακαδημαϊκές έρευνες δείχνουν ότι δεν είναι ακόμα έτοιμη για γενικότερη χρήση.

Υποκατηγορία της ανάλυσης ανωμαλιών είναι η ανίχνευση με την χρήση προφίλ (Profile-based Detection) η οποία ειδοποιεί στην περίπτωση που συμβεί αλλαγή στον τρόπο που οι χρήστες ή ένα σύστημα αλληλεπιδρά με το δίκτυο. Κληρονομεί με την σειρά της όμως τις ελλείψεις και τα προβλήματα της κατηγορίας της.

Πλεονεκτήματα

- Αν η μέθοδος εφαρμοστεί σωστά, έχει την δυνατότητα να ανιχνεύσει άγνωστες επιθέσεις.
- Προσφέρει χαμηλότερες επιβαρύνσεις, αφού δεν χρειάζεται να αναπτύσσονται συνέχεια νέες υπογραφές.

Μειονεκτήματα

- Συνήθως τέτοια συστήματα δεν έχουν την δυνατότητα να παράσχουν τα δεδομένα της εισβολής.
- Στις περισσότερες περιπτώσεις ο σηματοθορυβικός λόγος (Signal-to-noise ratio – SNR) είναι πολύ χαμηλός
- Η μέθοδος εξαρτάται σε μεγάλο βαθμό από το περιβάλλον όπου το σύστημα εκπαιδεύτηκε για το τι θεωρείται φυσιολογικό.

3. Επιλέγοντας ένα Σύστημα Ανίχνευσης Εισβολών.

Σε αυτήν την ενότητα θα επικεντρωθούμε στο να επιλέξουμε ένα σύστημα το οποίο θα υλοποιηθεί στο πειραματικό περιβάλλον μας, όπου μέσω μιας σειράς δοκιμών και μετρήσεων θα προσπαθήσουμε να βγάλουμε ένα συμπέρασμα για το πόσο αποδοτικό μπορεί να είναι για την ασφάλεια ενός δικτύου.

3.1 Υλοποιήσεις Συστημάτων.

Στην σημερινή εποχή η διακίνηση της πληροφορίας έχει γίνει ένα από τα πιο σημαντικά θέματα που απασχολούν τον άνθρωπο. Το Διαδίκτυο, προσφέρει την δυνατότητα να μεταφέρουμε σε οποιοδήποτε μέρος της γης, οποιαδήποτε πληροφορία επιθυμούμε και σε οποιαδήποτε μορφή αυτή βρίσκεται. Καθώς ο όγκος της διακινούμενης πληροφορίας μεγαλώνει και θέτοντας σαν στόχο την εύκολη αλλά και την γρήγορη μεταφοράς της, παρατηρούμε μια συνεχή αύξηση στην χωρητικότητα των δικτύων που την μεταφέρουν. Πλέον, μιλάμε για δίκτυα της τάξης των 100 Mb και ζούμε την μεταγωγή τους στην τάξη των 1000 Mb ανά δευτερόλεπτο.

Η αύξηση αυτή στην ταχύτητα των δικτύων συνεπάγεται αύξηση του φόρτου των συστημάτων ανίχνευσης δικτυακών επιθέσεων αφού πρέπει να επεξεργαστούν σε πραγματικό χρόνο 10 ή 100 φορές μεγαλύτερη ποσότητα πακέτων και να αποφανθούν πια συνιστούν απειλή. Ευτυχώς με την ταυτόχρονη αύξηση της χωρητικότητας των δικτύων υπάρχει και ταυτόχρονη αύξηση της υπολογιστικής ισχύς των ηλεκτρονικών συστημάτων, με αποτέλεσμα να είναι δυνατός ο έλεγχος της δικτυακής κίνησης για εισβολείς.

Πλέον μπορούμε να βρούμε αρκετά συστήματα ανίχνευσης δικτυακών εισβολών καθώς και συστήματα πρόληψης δικτυακών εισβολών σε δύο βασικές μορφές, με την χρήση λογισμικού (software) και η νέα στροφή με ηλεκτρονικά (hardware) προϊόντα. Στην παρούσα εργασία, αποκλείουμε τις λύσεις hardware που υπάρχουν σε πληθώρα με αρκετά υψηλή τιμή και υποσχόμενη κορυφαία απόδοση προστασίας του δικτύου μας και μελετούμε μόνο τις λύσεις λογισμικού.

3.2 Επιλογή λογισμικού NIDS.

Οι περισσότερες εταιρίες, όπως αναφέραμε και παραπάνω, έχουν αρχίσει να αφήνουν την

αγορά των βασιζόμενων σε λογισμικό NIDS και να προσφέρουν ολοκληρωμένες λύσεις σε επίπεδο hardware. Επίσης, εταιρίες που διαθέτουν ακόμα προϊόντα λογισμικού, παρατηρούμε ότι τα έχουν αφήσει στάσιμα σε αναβαθμίσεις, προσπαθώντας να προωθήσουν τις λοιπές λύσεις τους.

Αφήνοντας το στρατόπεδο των ακριβών εμπορικών προϊόντων, επόμενη λύση μας είναι ο χώρος του ανοιχτού λογισμικού (Open Source) όπου μπορούμε να βρούμε διάφορες λύσεις. Το ανοιχτό λογισμικό, πλέον θεωρείται αξιόπιστη λύση για πολλές εφαρμογές και η υποστήριξη που παρέχει τόσο σε επίπεδο αναβαθμίσεων αλλά και επίλυσης προβλημάτων θεωρείται ικανοποιητικό. Βέβαια, σε καμία περίπτωση δεν έχει ακόμα φτάσει στα επίπεδα των εμπορικών εφαρμογών μιας και προσφέρεται από την κοινότητα που υποστηρίζει το προϊόν και τους προγραμματιστές που το ανέπτυξαν, χωρίς αυτοί να έχουν στην διάθεση τους της δυνατότητες μιας μεγάλης εταιρίας κατασκευής λογισμικού.

Για την υλοποίηση του συστήματος NID που θα μας απασχολήσει παρακάτω θα χρησιμοποιηθεί το Snort, παρόλο που η κοινότητα έχει να επιδείξει και άλλα πακέτα λογισμικών όπως το Bro-IDs και το Prelude IDS. Οι λόγοι που καταλήξαμε στην επιλογή μας αναλύονται παρακάτω.

3.3 Γιατί να επιλέξουμε το Snort.

Παρακάτω παραθέτονται οι λόγοι που μας οδήγησαν στην επιλογή του εν λόγω πακέτου λογισμικού αντί των ανταγωνιστών του.

- Μπορεί να διαμορφωθεί στις ανάγκες μας. Όλα τα στοιχεία που απαρτίζουν το Snort, τα αρχεία διαμόρφωσης του αλλά και οι κανόνες που χρησιμοποιεί είναι διαθέσιμα ώστε να το προσαρμόσουμε στο δίκτυο μας.
- Είναι πολύ διαδεδομένο. Χρησιμοποιείται από πολλούς χρήστες και έχει μεγάλη κοινότητα που το υποστηρίζει, με αποτέλεσμα να είναι υπάρχει αρκετό υλικό τεκμηρίωσης για την εγκατάσταση αλλά και για την λειτουργία του.
- Είναι σχεδιασμένο για να λειτουργεί διαφορετικά λειτουργικά συστήματα. Μπορεί να λειτουργήσει τόσο σε διανομές Unix όσο και σε βασισμένες στο Unix καθώς επίσης και στα Windows.
- Αναβαθμίζεται συνεχώς. Η συντήρηση του γίνεται συνεχώς και μπορούμε να βρούμε διορθώσεις του ή προσθήκες χαρακτηριστικών σε τακτά χρονικά διαστήματα. Επίσης μπορούμε να κατεβάσουμε τις νέες υπογραφές επιθέσεων από την ιστοσελίδα του.

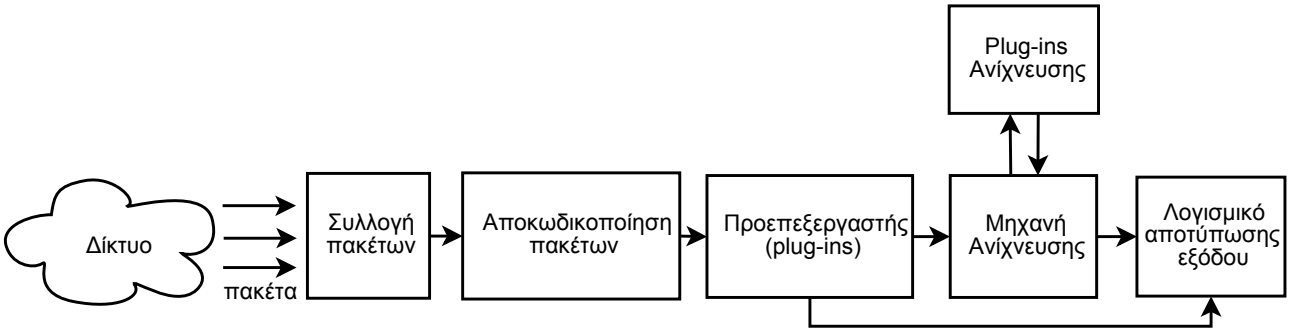
- Είναι φιλικό στην χρήση. Μόνο σε σχέση με άλλα NIDS ανοικτού κώδικα όπως το Bro.
- Είναι δωρεάν. Το Snort διανέμεται υπό την άδεια GNU GPL, που σημαίνει ότι μπορεί να χρησιμοποιηθεί δωρεάν. Αν και για να αποκτηθεί πρόσβαση στους πιο πρόσφατους κανόνες θα πρέπει να πληρώσουμε ένα ποσό ανάλογα με την χρήση και το πλήθος τον συστημάτων.

4. Η αρχιτεκτονική του Snort.

Πριν προχωρήσουμε με την τοποθέτηση του λογισμικού μας στο πειραματικό μας περιβάλλον θα πρέπει να μελετήσουμε την αρχιτεκτονική του ώστε να κατανοήσουμε την λειτουργία του και τον τρόπο που μας παρέχει τα αποτελέσματα του. Το Snort αποτελείται από διάφορα στοιχεία όπως προεπεξεργαστές και πακέτα λογισμικού ειδοποίησης (Alert Plugins). Η αρχιτεκτονική του Snort αποτελείται από πέντε βασικά συστατικά μέρη:

- Την βιβλιοθήκη συλλογής πακέτων (Packet Capture Library)
- Τον αποκωδικοποιητή πακέτων (Packet Decoder)
- Τον προεπεξεργαστή (Preprocessor)
- Την μηχανή ανίχνευσης (Detection Engine)
- Πακέτα λογισμικού αποτύπωσης της εξόδου (Output Plug-ins)

Στο παρακάτω σχήμα παρουσιάζεται η διαδρομή ενός πακέτου μέσω της αρχιτεκτονικής του Snort καθώς και τα επιμέρους τμήματα που το αποτελούν.



Σχήμα 1. Διαδρομή ενός δικτυακού πακέτου που ελέγχεται από το Snort.

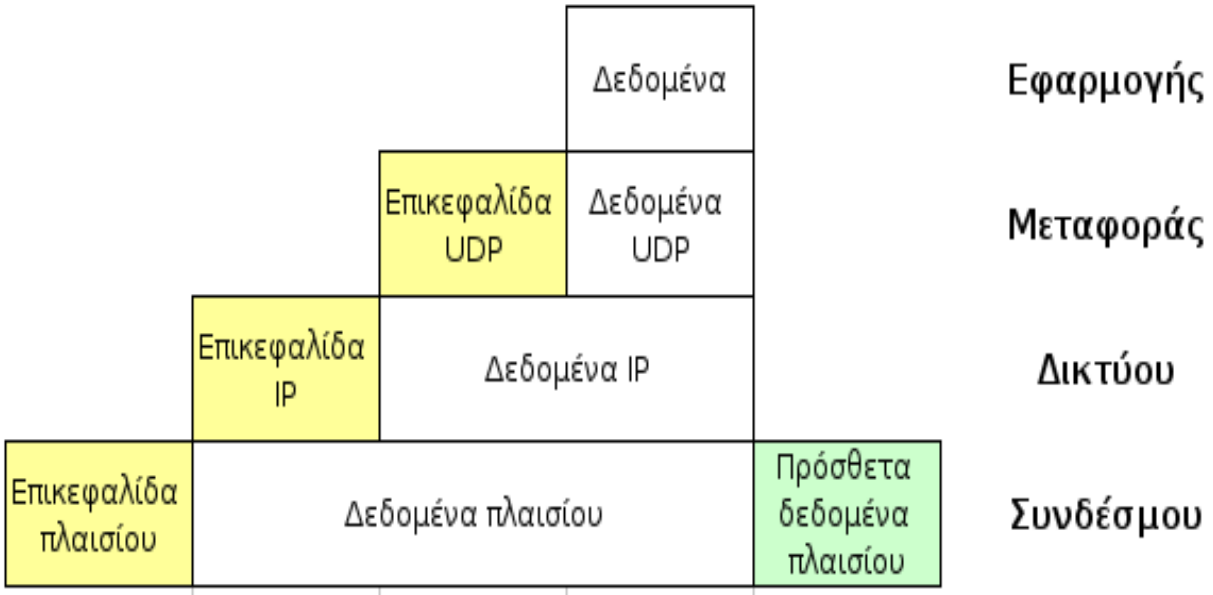
4.1 Βιβλιοθήκη Συλλογής Πακέτων (Packet Capture Library - Libpcap)

Στο Snort η συλλογή των πακέτων που διακινούνται στο δικτυακό τομέα που παρακολουθείται γίνεται μέσω της κάρτας δικτύου του υπολογιστή με την βοήθεια της βιβλιοθήκης pcap. Η συγκεκριμένη βιβλιοθήκη είναι μια διασύνδεση προγραμματισμού εφαρμογών (Application Programming Interface – API) που μπορεί να χρησιμοποιηθεί από μια εφαρμογή για την συλλογή πακέτων από το δίκτυο, ενώ στις νέες εκδόσεις έχει προστεθεί και η δυνατότητα να μεταδίδει πακέτα στο επίπεδο συνδέσμου (Link Layer) του TCP/IP. Το συγκεκριμένο API είναι σχεδιασμένο για να επικοινωνεί με τις εφαρμογές μέσω της γλώσσα προγραμματισμού C/C++. Η επιλογή της ως εξωτερικό πρόγραμμα συλλογής πακέτων για τον Snort έγινε καταρχήν λόγω του ότι είναι ανεξάρτητη του υπολογιστικού περιβάλλοντος, που σημαίνει οτι μπορεί να χρησιμοποιηθεί σε διαφορετικά λειτουργικά συστήματα (υπάρχει έκδοση και για Windows), ενισχύοντας έτσι την ανεξαρτησία του ίδιου του Snort.

Η βιβλιοθήκη επιτρέπει την συλλογή πακέτων σε ακατέργαστη μορφή όπως αυτά μεταφέρονται στο δίκτυο, χωρίς το εκάστοτε λειτουργικό σύστημα να έχει επιφέρει αλλαγές σε αυτά. Συνήθως μια εφαρμογή δεν μεταχειρίζεται ακατέργαστα πακέτα αλλά επαφύεται στο λειτουργικό σύστημα για “διαβάσει” τις πληροφορίες του πρωτοκόλλου και να μεταφέρει τα δεδομένα που χρειάζεται η εφαρμογή. Στην περίπτωση του Snort όμως συμβαίνει το ακριβώς αντίθετο, αφού απαιτούνται όλες οι πληροφορίες που περιέχονται σε ένα πακέτο για να μπορεί να ανιχνεύσει ορισμένες επιθέσεις.

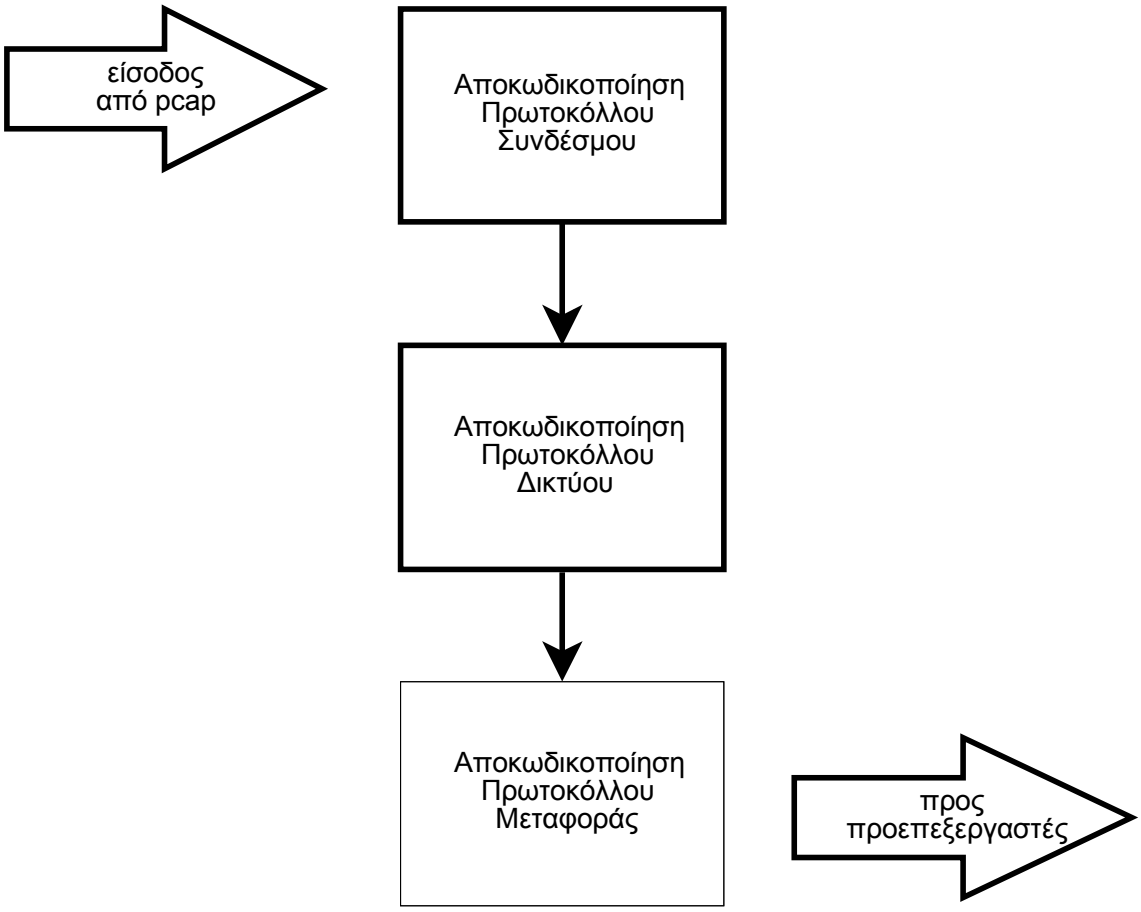
4.2 Αποκωδικοποιητής Πακέτων (Packet Decoding)

Η λειτουργία του είναι να λαμβάνει τα δεδομένα του επιπέδου συνδέσμου που συλλέγονται από την βιβλιοθήκη pcap και να τα αναλύει στα επιμέρους στοιχεία που το αποτελούν. Όπως είναι γνωστό τα πρωτόκολλα Διαδικτύου κάνουν χρήση της ενθυλάκωσης (encapsulation) για να παρέχουν γενικά πρωτόκολλα και υπηρεσίες. Ένα πρωτόκολλο υψηλού στρώματος χρησιμοποιεί τα χαμηλότερα για να μπορέσει να λειτουργήσει. Τα πακέτα, λοιπόν που μας παρέχονται από την pcap ανήκουν στο επίπεδο συνδέσμου, που σημαίνει ότι εμπεριέχουν τις πληροφορίες όλων των ανώτερων πρωτοκόλλων.



Σχήμα 2. Παράδειγμα ενθυλάκωσης δεδομένων σε ένα διάγραμμα UDP μέσα σε ένα πακέτο IP.

Στην πραγματικότητα, επομένως, ο αποκωδικοποιητής πακέτων είναι μια σειρά αποκωδικοποιητών όπου ο καθένας με την σειρά του αποκωδικοποιεί συγκεκριμένα στοιχεία των πρωτοκόλλων. Η σειρά που αναλύεται το πακέτο φαίνεται στο παρακάτω σχήμα.



Σχήμα 3. Διαδρομή του δικτυακού πακέτου μέσα στον αποκωδικοποιητή πακέτων.

Καθώς το πακέτο περνάει από τα διαδοχικά στάδια αποκωδικοποίησης, μια δομή δεδομένων γεμίζει με τα χρήσιμα πλέον δεδομένα του πακέτου με σκοπό την ανάλυση τους από τους προεπεξεργαστές και την μηχανή ανίχνευσης.

4.3 Προεπεξεργαστές.

Επιτρέπουν στο Snort να επεκτείνει την λειτουργικότητα του, δίνοντας την δυνατότητα σε χρήστες και προγραμματιστές να εισαγάγουν αρθρωτά 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. Θα πρέπει να αναφέρουμε ότι υπάρχουν είδη επιθέσεων τα οποία δεν θα μπορούσαν να ανιχνευθούν από το σύστημα Snort χωρίς την επιπλέον επεξεργασία των προεπεξεργαστών.

Οι προεπεξεργαστές είναι ένα πολύ σημαντικό χαρακτηριστικό του IDS λόγω του ότι τα plug-ins μπορούν να ενεργοποιηθούν ή να απενεργοποιηθούν κατά βούληση του διαχειριστή του. Αν για παράδειγμα, δεν επιθυμούμε να ελέγχουμε το δίκτυο μας για σαρώσεις θυρων, μπορούμε να απενεργοποιήσουμε το εν λόγω plug-in χωρίς να επηρεαστεί καθόλου το υπόλοιπο σύστημα μας. Οι παράμετροι που αφορούν τους προεπεξεργαστές διαμορφώνονται μέσω του αρχείου snort.conf όπως θα δούμε και αργότερα.

Η τελευταία σταθερή έκδοση του Snort 2.8.0 που χρησιμοποιούμε περιλαμβάνει τους εξής προεπεξεργαστές:

- Frag3

Ο συγκεκριμένος προεπεξεργαστής επιτρέπει την αποκατάτμηση του IP με βάση τον αποδέκτη. Έχει δημιουργηθεί για να αντικαταστήσει τον παλαιότερο frag2 προσφέροντας γρηγορότερη εκτέλεση , απλούστερη διαχείρηση δεδομένων και αντιμετώπιση τεχνικών αποφυγής ανίχνευσης. Η ανάλυση βασισμένη στον αποδέκτη (Target-based Analysis) που χρησιμοποιεί είναι μια σχετικά καινούργια έννοια στα NIDS, όπου αντί να βασιζόμαστε στα μοντέλα των πρωτοκόλλων και να αναζητούμε επιθέσεις, βασιζόμαστε στα συστήματα που είναι αποδέκτες.

- Stream4

Παρέχει τη δυνατότητα επανασυγκρότησης μιας ροής δεδομένων TCP καθώς και ανάλυση με βάση την κατάσταση της ροής. Ο Stream4 έχει την δυνατότητα να παρακολουθεί πολλές ταυτόχρονες ροές TCP. Συγκεκριμένα στην αρχική του διαμόρφωση είναι ρυθμισμένος να χειρίζεται 8192 ταυτόχρονες συνδέσεις TCP, ενώ μπορεί να ρυθμιστεί ώστε να χειρίζεται περισσότερες από 100.000 ταυτόχρονες συνδέσεις. Τέλος επίσης παρέχει την δυνατότητα παρακολούθησης συνεδριών UDP.

- Flow

Έχει ως σκοπό την ενοποίηση των μηχανισμών ελέγχου κατάστασης του Snort σε ένα μοναδικό σύστημα.

- Stream5

Ο προεπεξεργαστής Stream5 επιτρέπει την επανασυγκρότηση της ροής δεδομένων TCP με βάση τον αποδέκτη. Στόχος του είναι να αντικαταστήσει τον Stream4 αλλά και τον Flow.

- SfPortScan

Ο SfPortscan, έχει αναπτυχθεί από την Sourcefire, και έχει σχεδιαστεί για να ανιχνεύει την πρώτη φάση σε μια δικτυακή επίθεση: την φάση της αναγνώρισης. Στην φάση της αναγνώρισης, ο επιτιθέμενος προσπαθεί να ανακαλύψει τι είδους δικτυακά πρωτόκολλα και υπηρεσίες υποστηρίζει ο διακομιστής. Μιας και ο επιτιθέμενος δεν έχει γνώση του στόχου του, τα περισσότερα ερωτήματα που στέλνει στον διακομιστή θα απαντιούνται αρνητικά, αφού πολλές από τις υπηρεσίες δεν θα υπάρχουν. Γνωρίζοντας ότι όταν πρόκειται για "νόμιμη" δικτυακή επικοινωνία, οι αρνητικές απαντήσεις από διακομιστές είναι σπάνιες και πόσο μάλλον πολλαπλές αρνητικές απαντήσεις σε ένα δεδομένο χρόνο. Σκοπός, λοιπόν, του SfPortscan είναι να ανιχνεύσει αυτές τις αρνητικές απαντήσεις.

Ένα από τα γνωστότερα εργαλεία σάρωσης θυρών που χρησιμοποιείτε σήμερα είναι το Nmap. Το Nmap υποστηρίζει πολλές αν όχι όλες τις γνωστές τεχνικές σάρωσης θυρών, και το SfPortscan έχει σχεδιαστεί ώστε να μπορεί να ανιχνεύει τις διαφορετικές τεχνικές του Nmap.

- RPC Decode

Ο σκοπός του συγκεκριμένου προεπεξεργαστή είναι να κανονικοποιήσει τις πολλαπλές κατατμημένες εγγραφές σε μια ολοκληρωμένη εγγραφή, ωστέ να είναι δυνατή η αναγνώριση της υπογραφής μιας κακόβουλης εγγραφής από την μηχανή ανίχνευσης.

- Perfomance Monitor

Η ενεργοποίηση του Performance Monitor επιτρέπει την μέτρηση της πραγματικής και θεωρητικής απόδοσης του Snort. Έχει την δυνατότητα να τυπώνει τα στατιστικά στοιχεία είτε στην κονσόλα είτε σε ένα αρχείο για μετέπειτα επεξεργασία. Μερικά από τα στοιχεία που μας παρέχει είναι το ποσοστό χαμένων πακέτων, η χρήση του δικτύου, η χρήση της CPU και πολλά στατιστικά για τις συνδέσεις που υπάρχουν στο δίκτυο μας.

- HTTP Inspect

Ο προεπεξεργαστής HTTP Inpect είναι υπεύθυνος για την ανίχνευση αφύσικης κίνησης HTTP και να την κανονικοποιεί ώστε να μπορεί να ερμηνευτεί σωστά από την μηχανή ανίχνευσης. Με τον όρο κανονικοποίηση της κίνησης εννοούμε την διαδικασία “μετάφρασης” μιας ασαφούς συλλογής χαρακτήρων, όπως οι Unicode, σε μια συλλογή χαρακτήρων που το Snort δύναται να αναγνωρίσει.

Η κωδικοποίηση των δεδομένων HTTP είναι μια γνωστή μέθοδος που χρησιμοποιούν οι κράκερς για να κρύψουν μια επίθεση από τα IDS. Χωρίς την χρήση του HTTP Inspect μια επίθεση είναι εύκολο να μεταμφιεστεί ώστε να μην ταιριάζει με τις υπάρχουσες υπογραφές ανίχνευσης και ο διακομιστής ιστοσελίδων θα το θεωρήσει ως έγκυρο αλφαριθμητικό URL.

- SMTP Preprocessor

Ο προεπεξεργαστής αυτός χρησιμοποιείται για την αποκωδικοποιήση 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 συμπεριφορές που δεν είναι ορισμένες στα RFC.

- FTP/Telnet Preprocessor

Ο FTP/Telnet είναι μια βελτιωμένη έκδοση του παρωχημένου αποκωδικοποιητή Telnet, ο

οποίος παρέχει την 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 επίσης και να κανονικοποιήσει τα πεδία. Ο προεπεξεργαστής ελέγχει τόσο τις αιτήσεις του πελάτη όσο και τις απαντήσεις του διακομιστή.

- SSH

Ο SSH προεπεξεργαστής έχει σχεδιαστεί για να ανιχνεύει τα ακόλουθα προβλήματα ασφαλείας (Exploits): Goggles, CRC 32, Secure CRT, και το Protocol Mismatch.

- DCE/RPC

Ο προεπεξεργαστής ανιχνεύει και αποκωδικοποιεί SMB και DCE/RPC (Distributed Computing Environment/Remote Procedure Call) κίνηση, αν και ο μόνος λόγος που αποκωδικοποιεί το SMB είναι για να αποκτήσει τα δεδομένα DCE/RPC που περιέχονται στο επίπεδο του SMB.

- DNS

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

4.4 Μηχανή Ανίχνευσης.

Επόμενο βήμα στην διαδρομή των πακέτων στο Snort, αφού "περάσουν" από όλους τους ενεργοποιημένους προεπεξεργαστές είναι η μηχανή ανίχνευσης. Σκοπός της μηχανής ανίχνευσης είναι να ελέγξει τα δεδομένα που έρχονται από τους προεπεξεργαστές με ένα σύνολο κανόνων. Αν οι κανόνες ταιριάζουν με τα δεδομένα των πακέτων τότε αυτά στέλνονται στο άρθρωμα αποτύπωσης εξόδου, όπως φαίνεται και στο σχήμα 4.

Πιο συγκεκριμένα, η μηχανή ανίχνευσης έχει δυο σημαντικούς ρόλους: την ανάλυση των κανόνων και την ανίχνευση των υπογραφών. Η μηχανή ανίχνευσης είναι υπεύθυνη για την δημιουργία των υπογραφών αφού επεξεργαστεί τους κανόνες. Οι κανόνες διαβάζονται με την σειρά που βρίσκονται στο κάθε αρχείο και τοποθετούνται σε μια εσωτερική δομή δεδομένων. Η

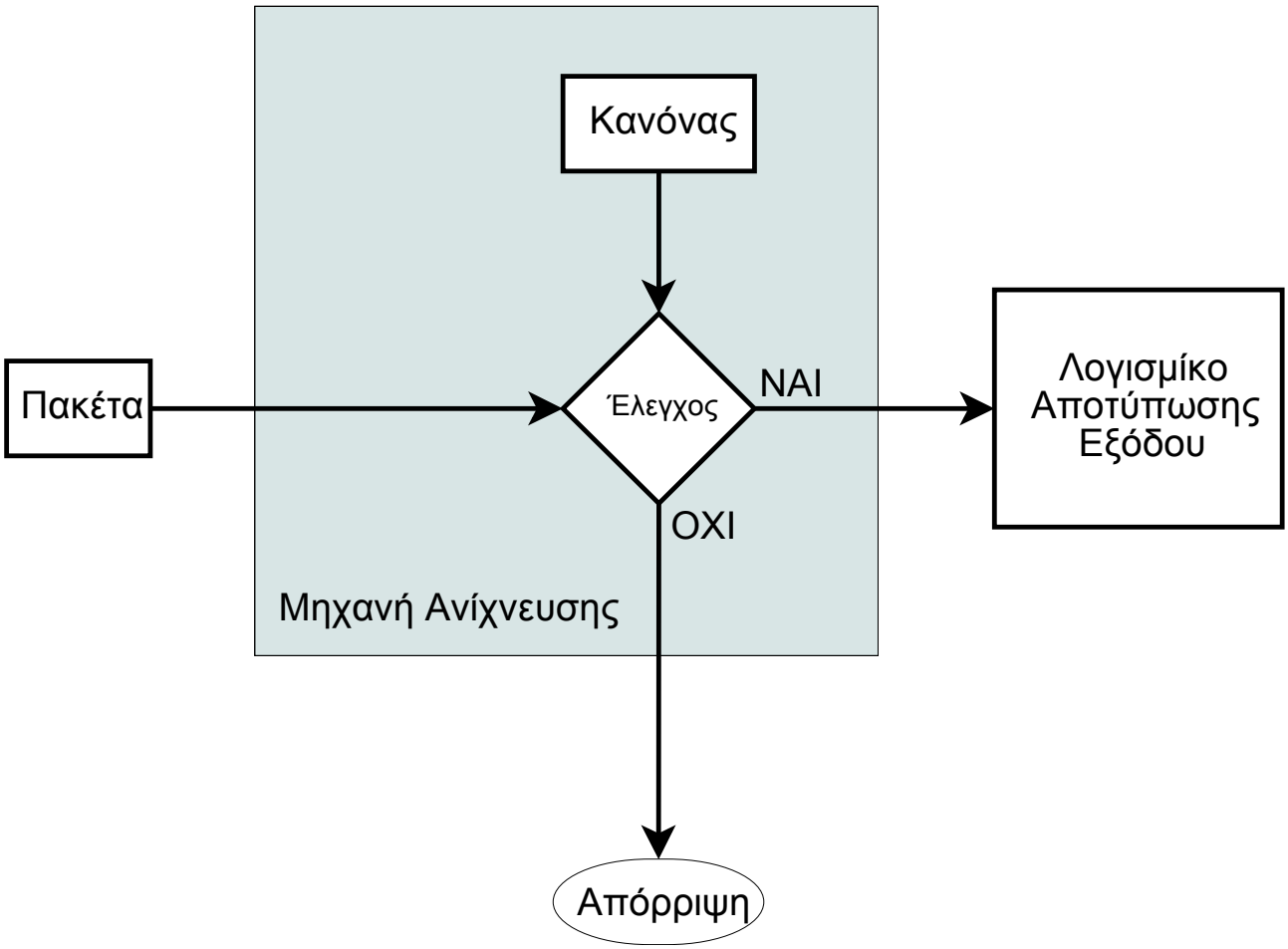
διαδικασία αυτή συμβαίνει στην όταν εκκινούμε το Snort, πράγμα που σημαίνει ότι αν τροποποιήσουμε τους κανόνες θα πρέπει να γίνει επανεκκίνηση του Snort για να εισαχθούν οι αλλαγές. Θα πρέπει να αναφέρουμε, ότι οι κανόνες χωρίζονται σε δυο τμήματα: τον κανόνα της κεφαλίδα και τον κανόνα της επιλογής. Η κεφαλίδα περιέχει πληροφορίες για τις συνθήκες που πρέπει να υπάρχουν ώστε να ισχύει η υπογραφή.

Τέτοιες πληροφορίες είναι:

- Ο τύπος καταχώρησης (Ειδοποίηση/Καταγραφή)
- Το πρωτόκολλο
- Το εύρος πεδίου των διευθύνσεων του αποστολέα και του παραλήπτη.
- Η θύρα επικοινωνίας.

Ο κανόνας επιλογής περιέχει:

- την υπογραφή
- το επίπεδο προτεραιότητας
- και ορισμένες πληροφορίες για την επίθεση



Σχήμα 4. Διάγραμμα λειτουργίας της μηχανής ανίχνευσης του Snort.

Η μηχανή ανίχνευσης επεξεργάζεται με διαφορετικό τρόπο τα δυο αυτά τμήματα του κανόνα Αρχικά δημιουργεί μια δενδροειδή διασυνδεδεμένη λίστα απόφασης με τα επιμέρους δεδομένα των κανόνων και στην συνέχεια ελέγχει τα δικτυακά δεδομένα με τους κόμβους του δένδρου για συγκεκριμένα στοιχεία. Έτσι ένα πακέτο αρχικά ελέγχεται για να προσδιοριστεί αν ανήκει στο πρωτόκολλο TCP και αν είναι τότε συνεχίζει να ελέγχεται από το μέρος του δένδρου που αφορά το TCP. Στην συνέχεια, ελέγχεται η διεύθυνση από την οποία προήλθε ώστε να συνεχίσει ο έλεγχος των κανόνων που το αφορούν. Η διαδικασία επαναλαμβάνεται μέχρι να βρεθεί μια υπογραφή που να το περιγράφει ή να αποδειχθεί ότι είναι έγκυρο και να απορριφθεί.

4.5 Λογισμικό αποτύπωσης εξόδου.

Αν τα δεδομένα που θα έχουν περάσει από την μηχανή ανίχνευσης ταιριάζουν με κάποιον κανόνα της, τότε θα πρέπει να εκδοθεί μια προειδοποίηση για το συμβάν. Το λογισμικό αποτύπωσης εξόδου είναι υπεύθυνο για να μεταφέρει τις προειδοποιήσεις αυτές στην μορφή που επιθυμούμε. Το Snort παρέχει πολλούς διαφορετικούς τρόπους για την αποτύπωση των δεδομένων και επιπρόσθετα υπάρχουν προγράμματα που έχουν γραφτεί για να τον συγκεκριμένο λόγο. Θα πρέπει επίσης να αναφέρουμε ότι το Snort δεν μας περιορίζει σε μια μόνο λύση αποτύπωσης εξόδου αλλά μπορούμε να ενεργοποιήσουμε πλέον του ενός τρόπου.

Στην παρούσα έκδοση του το Snort περιλαμβάνει τις εξής μεθόδους:

- alert_syslog

Μας παρέχει την δυνατότητα να αποτυπώνουμε την έξοδο του Snort σύστημα του syslog. Η υπηρεσία syslog μπορεί επίσης να χρησιμοποιηθεί ώστε να καταγράφει πληροφορίες από πολλές διαφορετικές συσκευές εκτός του Snort, όπως firewalls, διακομιστές ιστοσελίδων κ.α.

- alert_fast

Τυπώνει τις ειδοποιήσεις του Snort σε διάταξη μιας σειράς ανά εγγραφή στο αρχείο που θα επιλέξουμε. Είναι γρηγορότερος τρόπος ειδοποίησης από την δυνατότητα alert full που θα δούμε στην συνέχεια, αφού δεν γράφει τις κεφαλίδες των πακέτων στο αρχείο εξόδου.

- alert_full

Είναι ένας απαρχαιωμένος τρόπος καταγραφής, αλλά μπορεί να χρησιμοποιηθεί σε χαμηλής χωρητικότητας δίκτυα. Το συγκεκριμένο plug-in δημιουργεί έναν κατάλογο για κάθε διεύθυνση που παράγει έναν συναγερμό και μέσα σε αυτόν αποθηκεύει το αποκωδικοποιημένο περιεχόμενο των πακέτων, συμπεριλαμβανομένων και των κεφαλίδων του.

- alert_unixsock

"Ανοιγει" μια θύρα επικοινωνίας σε ένα Unix σύστημα και αποστέλλει σε αυτήν τις ειδοποιήσεις. Για να πάρουμε τα δεδομένα, θα πρέπει να χρησιμοποιήσουμε κάποιο εξωτερικό πρόγραμμα ή διαδικασία, η οποία θα συνδεθεί στην συγκεκριμένη θύρα ώστε να λάβει τις ειδοποιήσεις του Snort. Όπως είναι φανερό ο συγκεκριμένος τρόπος λειτουργίας δεν είναι εφικτός στο λειτουργικό περιβάλλον Windows.

- log_tcpdump

Χρησιμοποιείται για να καταγράψει τα συμβάντα σε ένα αρχείο σύμφωνα με την διαμόρφωση που χρησιμοποιεί το πρόγραμμα tcpdump. Η συγκεκριμένη μέθοδος καταγραφής συνηθίζεται σε περιπτώσεις όπου θέλουμε να επεξεργαστούμε περαιτέρω τις πληροφορίες των πακέτων με την πληθώρα προγραμμάτων που υποστηρίζουν την συγκεκριμένη μορφή.

- 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.

- csv

Επιτρέπει την αποθήκευση των συμβάντων σε αρχεία τύπου CSV. Ο συγκεκριμένος τύπος αρχείου διευκολύνει την εισαγωγή των στοιχείων σε βάσεις δεδομένων και υπολογιστικά φύλλα

λόγω της οριοθέτησης τους από (,).

- unified

Έχει σχεδιαστεί ειδικά για να παρέχει την μεγαλύτερη δυνατή ταχύτητα. Το unified plug-in αποθηκεύει τα δεδομένα των εισβολών σε ένα ειδικά διαμορφωμένο δυαδικό φορμάτ. Η συγκεκριμένη έξοδος δημιουργεί δυο διαφορετικά αρχεία, το αρχείο καταχωρήσεως δεδομένων και το αρχείο ειδοποιήσεων. Στον πρώτο, αποθηκεύονται όλες οι πληροφορίες του περιέχει το εν λόγω πακέτο, ενώ στο δεύτερο γράφεται μια περίληψη του συμβάντος που περιλαμβάνει τις διευθύνσεις IP του αποστολέας και του παραλήπτη, το εμπλεκόμενο πρωτόκολλο επικοινωνίας, τις θύρες επικοινωνίας που χρησιμοποιήθηκαν καθώς και την ταυτότητα του συμβάντος.

Όπως αναφέραμε 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.

- unified 2

Είναι ο αντικαταστάτης της διαμόρφωσης unified και έχει σχεδιαστεί με τα ίδια χαρακτηριστικά αποδοτικότητας αλλά διαφορετικό φορμάτ αποθήκευσης.

- alert_prelude

Επιτρέπει στην έξοδο να αποθηκευτεί σε μια βάση δεδομένων του υβριδικού συστήματος IDS Prelude. Το συγκεκριμένο plug-in δεν εγκαθίσταται με την συνηθισμένη εγκατάσταση του Snort αλλά πρέπει να επιλεχτεί, αν επιθυμούμε την χρήση του.

- log null

Πολλές φορές είναι χρήσιμη η δυνατότητα να δημιουργήσουμε κανόνες που θα ειδοποιούν για ορισμένους τύπους διαδικτυακής κίνησης χωρίς όμως να θέλουμε να την αποθήκευση των πακέτων που την προκάλεσαν. Για την συγκεκριμένη περίπτωση έχει δημιουργηθεί το plug-in εξόδου log null.

- alert_arube_action

Το plug-in αυτό έχει την δυνατότητα να επικοινωνεί με ένα ασύρματο ελεγκτή δικτύων Aruba και να αλλάζει την κατάσταση των εξουσιοδοτημένων σε αυτό χρηστών. Επιτρέπετε έτσι στο Snort να λαμβάνει μέτρα προστασίας προς τους χρήστες ενός τέτοιου δικτύου, ελέγχοντας την δικτυακά τους δικαιώματα. Το συγκεκριμένο plug-in δεν εγκαθίσταται με την συνηθισμένη εγκατάσταση του Snort αλλά πρέπει να ενεργοποιηθεί κατά την εγκατάσταση.

5. Υλοποιώντας ένα σύστημα ανίχνευσης εισβολών.

Η τοποθέτηση ενός NIDS σε ένα δίκτυο θα πρέπει να γίνει με πολύ προσοχή, έχοντας κατά νου τις ιδιαιτερότητες του εκάστοτε δικτύου καθώς και τα αποτελέσματα που θέλουμε να επιτύχουμε. Θα πρέπει να ληφθεί υπ όψιν ότι εγκαθιστούμε ένα λογισμικό ασφάλειας το οποίο έχει κάποιες απαιτήσεις, αλλά επίσης και το γεγονός ότι το συγκεκριμένο λογισμικό θα πρέπει να αλληλεπιδρά με το δίκτυο. Από μόνο του το γεγονός αυτό δημιουργεί προβλήματα μιας και θα πρέπει οποιαδήποτε υλοποίηση σχεδιαστεί προσεκτικά.

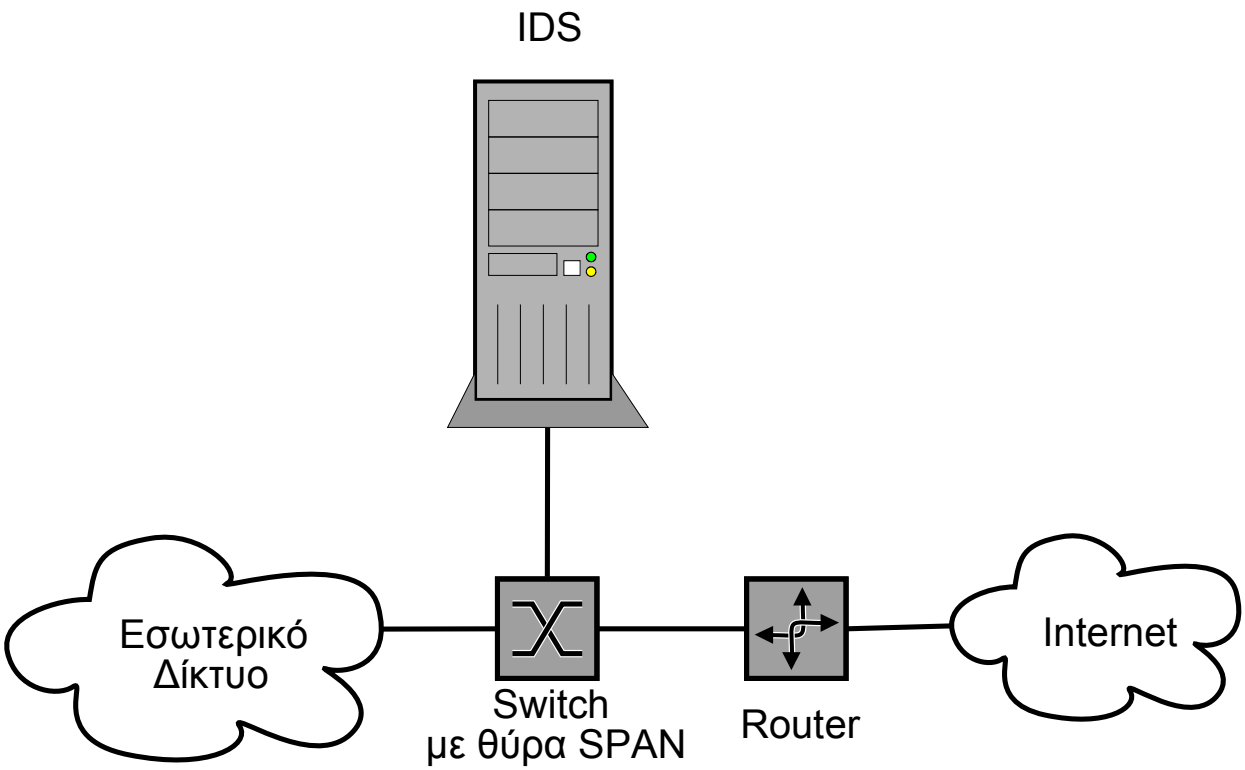
Τα περισσότερα δίκτυα, πλέον, βασίζουν την υποδομή τους σε διαμεταγωγείς (switch) σε αντίθεση με το παλιότερα όπου το μερίδιο του λέοντος κατείχαν τα hub. Η αλλαγή αυτή περιπλέκει την ευκολία ενσωμάτωσης του συστήματος στο δίκτυο μας λόγω των ιδιαιτεροτήτων των switches. Στην συνέχεια θα αναφερθούμε στους τρόπους με τους οποίους μπορούμε να παρακολουθήσουμε ένα δίκτυο που χρησιμοποιεί switches καθώς και τα θετικά και αρνητικά σημεία της κάθε συνδεσμολογίας.

5.1 Συνδεσμολογία για την παρακολούθηση ενός δικτύου βασισμένο σε switch.

Για να ξεπεράσουμε το εγγενές "πρόβλημα" των switch, όπου τα δεδομένα πλέον δεν αντιγράφονται σε κάθε θύρα όπως στην περίπτωση των hubs έχουμε τρεις βασικές μεθόδους. Στις επόμενες παραγράφους θα δούμε πως μπορούμε να παρακολουθήσουμε την κίνηση μεταξύ ενός δρομολογητή (Router) και ενός switch.

5.1.2 Χρήση SPAN (Switch Port Analyzer) θύρας.

Η θύρα SPAN δίνει την δυνατότητα σε ένα switch να μπορεί να αντιγράφει την εισερχόμενη/ή και εξερχόμενη κίνηση από μία θύρα ή VLAN σε μία άλλη θύρα. Συνδέοντας τον δρομολογητή (Router) και το σύστημα ανίχνευσης εισβολών στο switch και ενεργοποιώντας την αντιγραφή των δεδομένων (RX/TX) της θύρας του δρομολογητή στην θύρα του IDS μας, μπορούμε να παρακολουθούμε όλα τα δεδομένα που διακινούνται. Εκτός από την παραπάνω δικτυακή κίνηση που αντιγράφετε στην θύρα του IDS, η ίδια θύρα μπορεί να χρησιμοποιηθεί και για τον χειρισμό του συστήματος μας. Στο σχήμα 5 παρουσιάζεται ο συγκεκριμένος τύπος σύνδεσης.



Σχήμα 5. Παρακολούθηση δικτύου με την χρήση θύρας SPAN.

Τα πλεονεκτήματα της χρήσης της συγκεκριμένης μεθόδου είναι:

- Ευκολία εγκατάστασης. Το IDS μπορεί να συνδεθεί απευθείας στο switch χωρίς να είναι αναγκαίες τροποποιήσεις στην δομή του δικτύου μας.
- Η διαχείριση του IDS μπορεί να γίνει εύκολα χωρίς επιπλέον υλικό.
- Ο τερματισμός συνεδριών και επαναρύθμιση του τείχου προστασίας δεν επηρεάζονται.

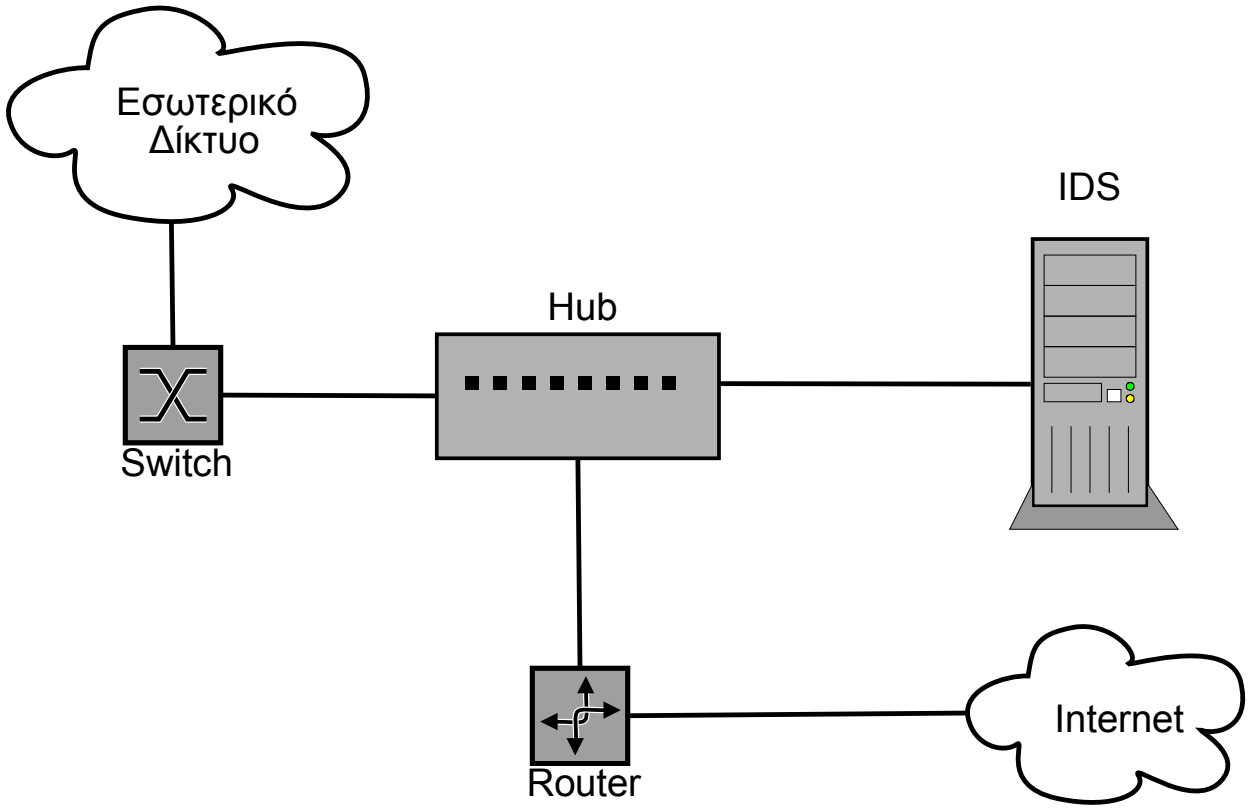
Τα μειονεκτήματα είναι:

- Μπορεί να υπάρχει μόνο μια θύρα SPAN σε ένα switch. Εκτός αν γίνει SPAN σε VLAN ή μια σειρά από θύρες σε μια (θυρες από 1 – 4 στην 5)
- Με το αντιγράφουμε πολλές θύρες σε μια θα υπερφορτώσουμε την θύρα SPAN.
- Αν δεν κάνουμε αλλαγές στο IDS είναι ευάλωτο σε επιθέσεις.
- Δεν υπάρχει η δυνατότητα να αντιγραφούν προβληματικά πακέτα. (CRC errors)

5.1.3 Χρήση hub.

Όπως αναφέραμε και παραπάνω δίκτυα που στηρίζονται σε hub είναι ευάλωτα σε παρακολούθηση, αφού έχουν την ιδιότητα ως διαμοιραζόμενα μέσα όταν λάβουν ένα πακέτο (πλαίσιο – Frame) να το μεταδίδουν σε όλες της θύρες του. Συνήθως η συγκεκριμένη μέθοδος παρακολούθησης δικτύου δεν χρησιμοποιείτε μιας και οι πιθανότητες να δημιουργήσουμε προβλήματα λόγω του σχετικά μικρού εύρους για μεταφορά δεδομένων και την ύπαρξη συγκρούσεων στα πλαίσια που μεταδίδονται όταν η δικτυακή κίνηση αυξάνει.

Στην περίπτωση που χρησιμοποιηθεί hub η υλοποίηση της παρακολούθησης των δεδομένων που διέρχονται από τον δρομολογητή γίνεται απλά προσθέτοντας πριν το διαμεταγωγέα το hub, όπως φαίνεται στο σχήμα 6.



Σχήμα 6. Παρακολούθηση δικτύου με την χρήση Hub.

Τα πλεονεκτήματα χρήσης της συγκεκριμένης μεθόδου είναι:

- Εύκολη υλοποίηση που δεν απαιτεί λεπτομερή γνώση των δικτύων.
- Η διαχείριση του IDS μπορεί να γίνει εύκολα χωρίς επιπλέον υλικό.
- Μπορούμε να τερματίζουμε συνεδρίες και δεν χρειάζεται επιπλέον ρύθμιση του τείχους προστασίας.

- Τα hubs είναι πολύ οικονομικά.

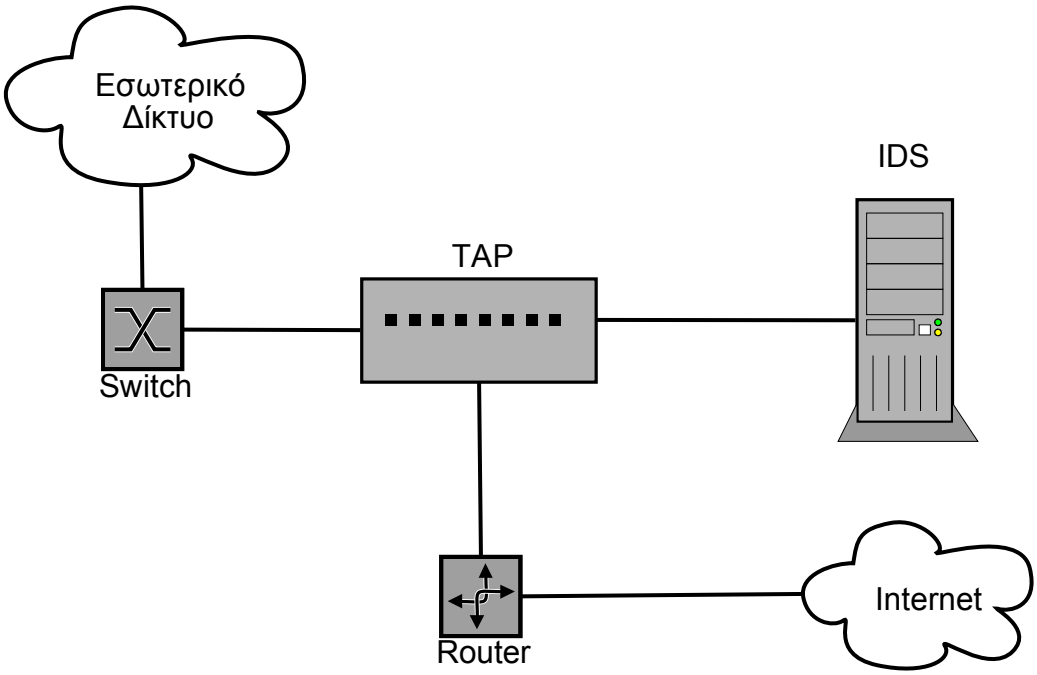
Τα μειονεκτήματα της είναι:

- Αν η σύνδεση μεταξύ του διαμεταγωγέα και του δρομολογητή είναι τύπου full-duplex ο αριθμός των συγκρούσεων θα μειώσει την συνολική απόδοση.
- Τα hubs είναι πολύ πιο ευάλωτα σε μηχανικά προβλήματα.

5.1.4 Με την χρήση δικτυακών "κοριών" (TAP)

Είναι συσκευές που επιτρέπουν την παρακολούθηση της δικτυακής κίνησης χωρίς να προκαλούν οποιαδήποτε αλλαγή στην ροή δεδομένων. Ένας δικτυακός "κοριός" έχει τουλάχιστον τρεις θύρες – την θύρα Α την Β και την θύρα παρακολούθησης. Σύμφωνα με το παράδειγμα μας για παρακολουθούμε την κίνηση μεταξύ του δρομολογητή και του διαμεταγωγέα αρκεί να παρεμβάλλουμε τον "κοριό" ανάμεσά τους, συνδέοντας το καλώδιο δικτύου από τον δρομολογητή στην θύρα Α του κοριού και την θύρα Β με τον διαμεταγωγέα. Τέλος στον θύρα παρακολούθησης τοποθετούμε το σύστημα IDS το οποίο με τη σειρά του επεξεργάζεται την κίνηση που παρακολουθεί. Συνήθως ένας δικτυακός "κοριός" επιτρέπει δικτυακή κίνηση μίας κατεύθυνσης, με αποτέλεσμα αν θέλουμε να παρακολουθούμε και τα εισερχόμενα και τα εξερχόμενα πακέτα να γίνεται χρήση περισσότερων του ενός "κοριού".

Στο εμπόριο υπάρχει πληθώρα τέτοιων συστημάτων με διάφορα χαρακτηριστικά και δυνατότητες, τα οποία δεν θα μελετηθούν στην συγκεκριμένη πτυχιακή εργασία.



Σχήμα 7. Παρακολούθηση δικτύου με την χρήση δικτυακού "κοριού".

Τα πλεονεκτήματα χρήσης της συγκεκριμένης μεθόδου:

- Δεν δημιουργεί προβλήματα στην κίνηση των δεδομένων.
- Αφού γίνει η τοποθέτηση του, οποιαδήποτε αλλαγή στην δομή του δικτύου ανάμεσα στην θύρα παρακολούθησης και το IDS δεν επηρεάζει το συνολικό δίκτυο.
- Επιτρέπει στο IDS να παρακολουθεί "προβληματικά" πακέτα πχ. CRC errors.
- Στην περίπτωση που διακοπή η παροχή ρεύματος στον "κοριό" η σύνδεση των θυρών A και B συνεχίζει να λειτουργεί.
- Μπορούν να παρακολουθηθούν full-duplex συνδέσεις.

Τα μειονεκτήματα της είναι:

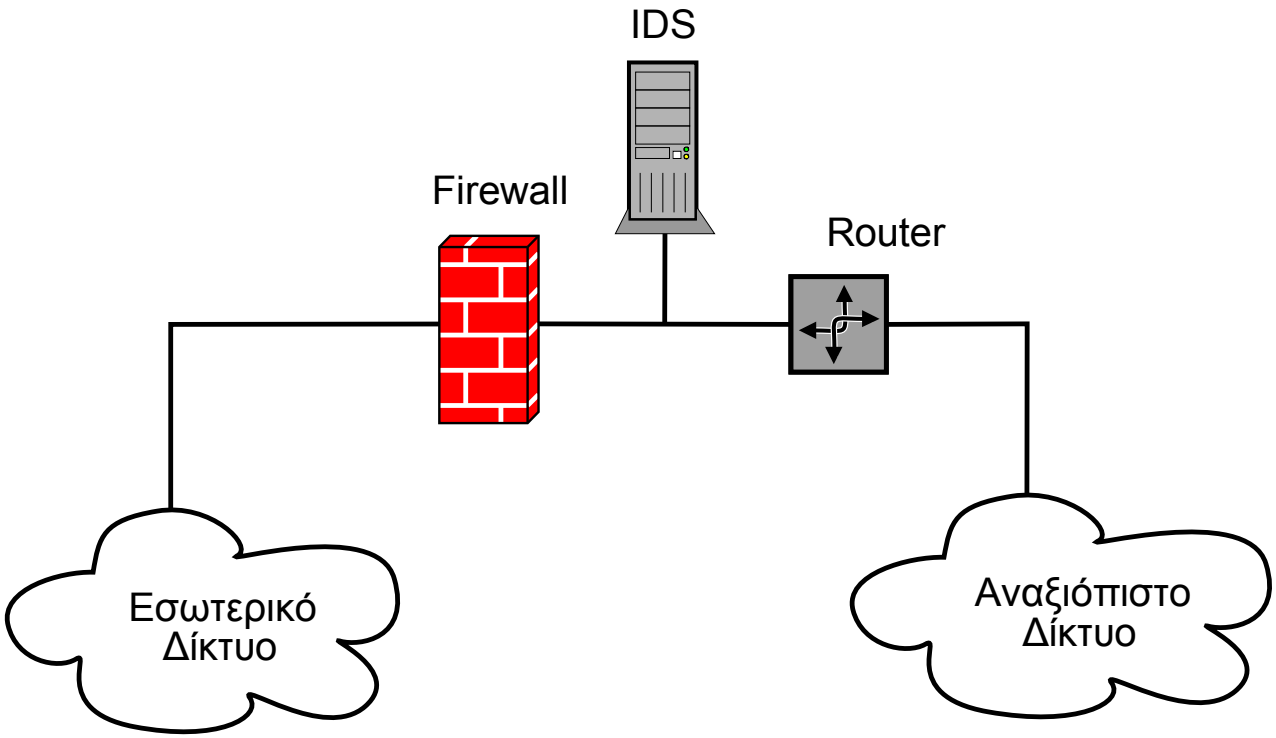
- Η τιμή των συσκευών παγίδευσης.
- Η παρακολούθηση μεγάλων δικτύων απαιτεί πολλούς "κοριούς" και IDS.
- Χρειάζονται ειδικές ρυθμίσεις για full-duplex συνδέσεις (συνδυασμός καναλιών στο IDS – channel-bonding)

5.2 Τοποθέτηση Αισθητήρων σε ένα δίκτυο.

Έχοντας ήδη αναλύσει τον τρόπο σύνδεσης των δικτύων, θα πρέπει τώρα να μελετήσουμε τα σημεία του δικτύου όπου η τοποθέτηση αισθητήρων IDS θα παρέχει αξιόπιστη παρακολούθηση των επιθέσεων που συμβαίνουν σε αυτό. Τα σημεία που κρίνονται αναγκαία για παρακολούθηση είναι τα σημεία όπου διαφορετικά δικτυακά μέρη ενώνονται, καθώς και στα υποδίκτυα που θέλουμε να προστατέψουμε. Βέβαια, η όλη υποδομή ενός NIDS είναι ανάλογη με τους πόρους μπορούμε να διαθέσουμε για τον κάθε ξεχωριστό αισθητήρα καθώς και την ιδιαίτερη τοπολογία του δικτύου. Παρακάτω θα δούμε μερικά παραδείγματα τοποθέτησης IDS αισθητήρων σε δίκτυα.

5.2.1 Τοποθέτηση ενός μόνο IDS μετά τον δρομολογητή.

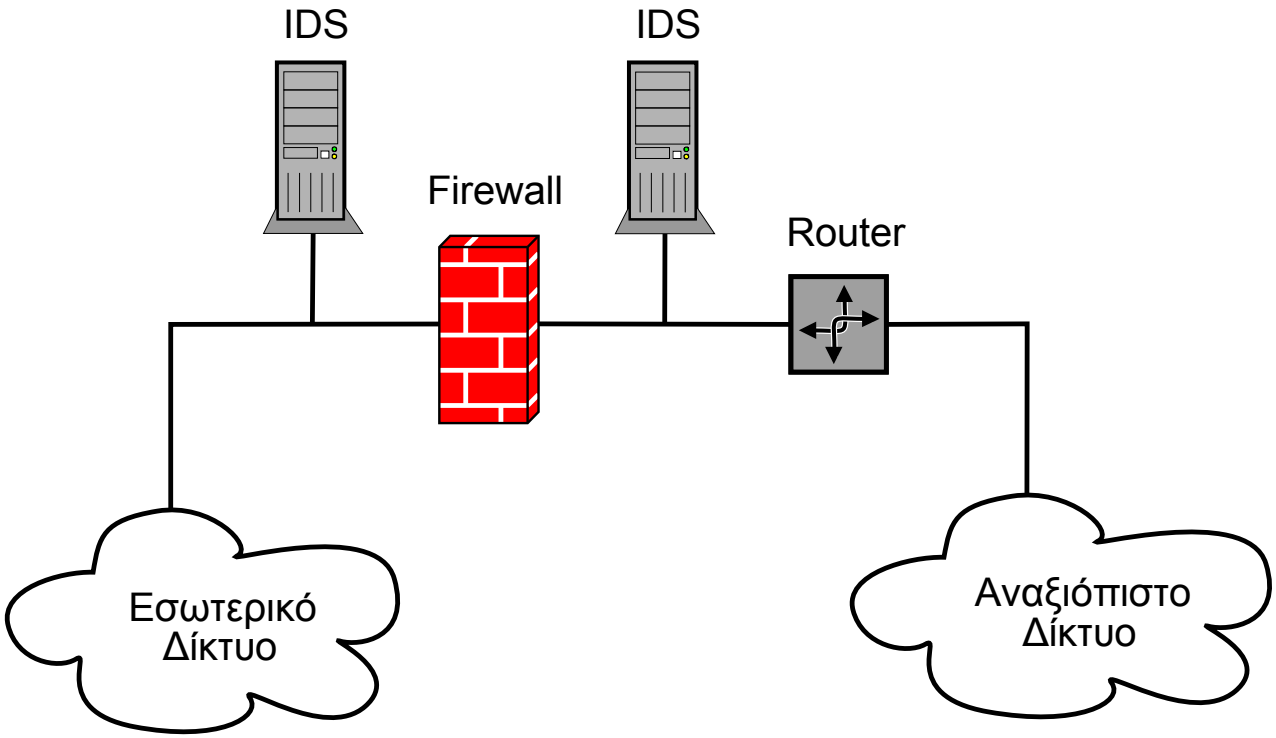
Είναι ίσως η πιο συνηθισμένη λύση τοποθέτησης IDS για μικρά δίκτυα. Το IDS τοποθετείτε αμέσως μετά τον δρομολογητή ώστε να έχει πρόσβαση σε όλο την κίνηση που εισέρχεται στο δίκτυο και πριν από το Firewall ώστε να παρέχει μια γενικότερη εικόνα των προσπαθειών επίθεσης στο δίκτυο, πριν αυτές πιθανόν απορριφθούν από αυτό. Ωστόσο η δικτυακή κίνηση που προσλαμβάνει ο αισθητήρας σε αυτήν την συνδεσμολογία δεν είναι εντελώς αναλλοίωτη, αφού οι περισσότεροι διαχειριστές δικτύων χρησιμοποιούν λίστες ελέγχου πρόσβασης (Access Control Lists) για να φιλτράρουν την κίνηση προς το δίκτυο τους. Το IDS μπορεί να χρησιμοποιηθεί επίσης και για τον έλεγχο των λιστών αυτών, δίνοντας μας πληροφορίες για να τις βελτιώσουμε. Ο σχεδιασμός του δικτύου φαίνεται στο σχήμα 8.



Σχημα 8. Παρακολούθηση όλης της δικτυακής κίνησης με την χρήση ενός IDS.

5.2.2 Προσθέτοντας ακόμα ένα IDS μετά το Firewall.

Η τοποθέτηση ενός δεύτερου IDS μετά το Firewall, μας παρέχει μια σαφώς καλύτερη εικόνα του τι συμβαίνει στην πραγματικότητα στο εσωτερικό του δικτύου. Η βασική εργασία ενός Firewall σε ένα δίκτυο είναι να ρυθμίζει την δικτυακή κίνηση μεταξύ δυο ζωνών που έχουν διαφορετικό επίπεδο ασφάλειας απορρίπτοντας τα πακέτα δεδομένων που δεν πρέπει να εισέρχονται στο "ασφαλές" δίκτυο. Επομένως, το IDS σε αυτήν την περίπτωση ελέγχει την δικτυακή κίνηση που θεωρείται ασφαλής, ενημερώνοντας μας για επιθέσεις που έχουν περάσει το Firewall και έχουν εισχωρήσει στο εσωτερικό δίκτυο.

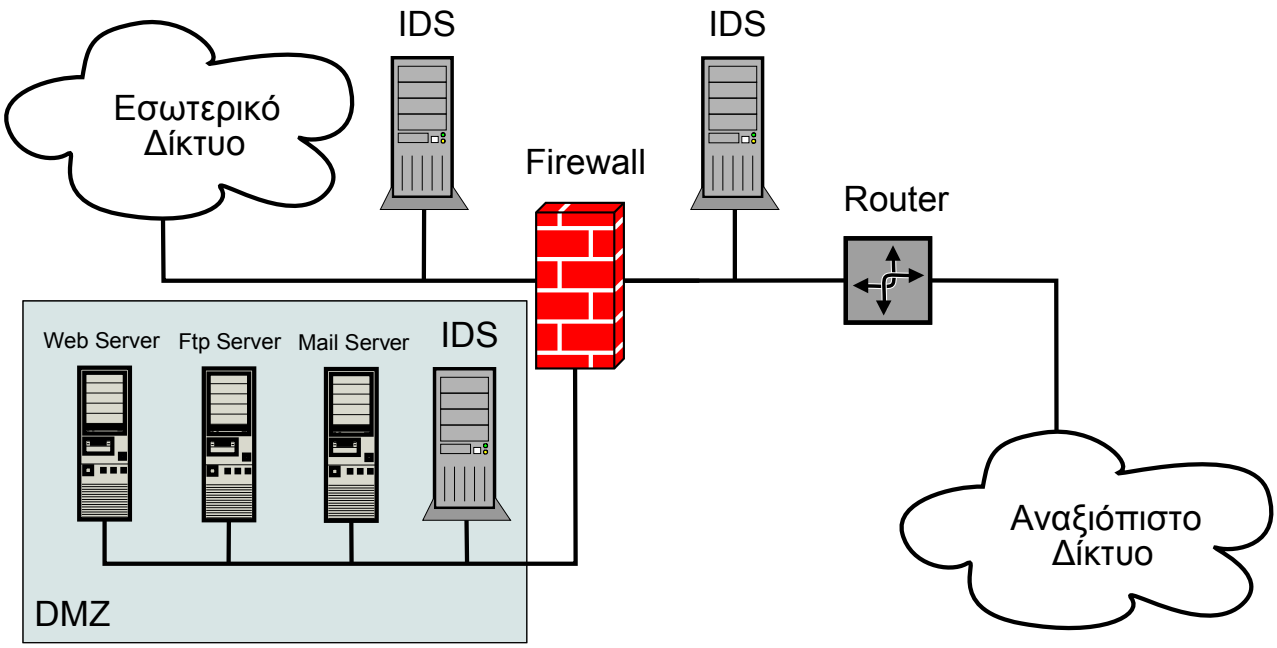


Σχήμα 9. Χρήση δυο IDS για καλύτερη εποπτεία του δικτύου.

5.2.3 Υπερασπίζοντας την DMZ (Demilitarized Zone).

Πολλές αρχιτεκτονικές δικτύων χρησιμοποιούν την "αποστρατικοποιημένη" ζώνη (DMZ) για να παρέχουν τις δημόσιες υπηρεσίες τους, διακομιστές ιστοσελίδων, διακομιστές FTP. Σε ένα δίκτυο οι πιο ευάλωτοι σε επιθέσεις υπολογιστές,είναι αυτοί που προσφέρουν υπηρεσίες στο εξωτερικό δίκτυο. Για τον παραπάνω λόγω οι υπολογιστές αυτοί τοποθετούνται σε ένα ξεχωριστό υποδίκτυο, ώστε αν κάποιος εισβολέας αποκτήσει πρόσβαση σε αυτούς να μην μπορεί να επιτεθεί στους υπολογιστές του εσωτερικού δικτύου. Για να επιτευχθεί φυσικά το παραπάνω, πρέπει οι διακομιστές που βρίσκονται στην DMZ να μην έχουν την δυνατότητα απευθείας επικοινωνίας με το εσωτερικό δίκτυο.

Σε ένα τέτοιο περιβάλλον δικτύου θα χρειαστούμε τρία IDS, ένα πριν το Firewall, ένα μετά και ένα μέσα στην DMZ. Ο λόγος που χρησιμοποιούμε ένα παραπάνω IDS είναι το νέο υποδίκτυο DMZ που θέλουμε να προστατέψουμε. Η σχηματική αναπαράσταση παρουσιάζεται στο σχήμα 10.



Σχήμα 10. Παρακολούθηση της DMZ με την χρήση ενός επιπλέον IDS.

6. Σχεδίαση-Υλοποίηση του εργαστηριακού μας περιβάλλοντος.

Στο συγκεκριμένο κεφάλαιο θα ορίσουμε τις προδιαγραφές του εργαστηριακού περιβάλλοντος που θα μας επιτρέψουν να έχουν αμερόληπτες και σωστές μετρήσεις, βασιζόμενοι πάντα στον εξοπλισμό που μπορεί να μας διατεθεί. Αν και οι προτεινόμενες λύσεις ορισμένες φορές δεν μας παρέχουν τα καλύτερα αποτελέσματα, έχουν γίνει προσπάθειες ώστε τα συγκεκριμένα στοιχεία να αφήνουν ανεπηρέαστο το αποτέλεσμα.

6.1 Προδιαγραφές του εργαστηριακού περιβάλλοντος.

Για την συγκεκριμένη πτυχιακή εργασία, το εργαστηριακό περιβάλλον που θα χρησιμοποιηθεί για να μελετηθεί η συμπεριφορά του συστήματος ανίχνευσης δικτυακών εισβολών είναι το πιο σημαντικό τμήμα, αφού σε αυτό θα στηριχτούμε για να εξάγουμε τα συμπεράσματα μας. Το εργαστηριακό περιβάλλον, θέλουμε να μας παρέχει τις ακόλουθες απαιτήσεις ώστε να μπορεί να εξομοιώνει όσο το δυνατόν καλύτερα ένα πραγματικό δίκτυο. Θα πρέπει λοιπόν:

- Να έχει διαφορετικά λειτουργικά συστήματα.

Ένα δίκτυο συνήθως, αποτελείται από πολλά διαφορετικά υπολογιστικά συστήματα όπου το καθένα επιτελεί διαφορετική εργασία σε αυτό. Ο κάθε υπολογιστής ενός δικτύου επικοινωνεί με την χρήση δικτυακών πρωτοκόλλων που είναι σαφώς ορισμένα και υπάρχουν προδιαγραφές για την λειτουργία τους με τα υπόλοιπα συστήματα του δικτύου. Δυστυχώς όμως ο κάθε κατασκευαστής του εκάστοτε πρωτοκόλλου ενσωματώνει στην υλοποίηση του την "προσωπική" ερμηνεία των κανόνων που το περιβάλουν. Αποτέλεσμα των παραπάνω είναι να υπάρχουν διαφοροποιήσεις στον τρόπο ανασύνθεσης τον δικτυακών ροών στα διάφορα λειτουργικά σύστημα.

- Να υπάρχει διαφοροποίηση εσωτερικού-εξωτερικού δικτύου.

Έχοντας ως βασικό στόχο την δημιουργία ενός ρεαλιστικού μοντέλου, το περιβάλλον μας θα πρέπει να αποτελείται από δύο τουλάχιστον διαφορετικά δίκτυα. Κάθε διαφορετική ζώνη ασφαλείας θα μεταφραστεί σε ένα διαφορετικό υποδίκτυο, δημιουργώντας την ζώνη έμπιστου

δικτύου, όπου θα βρίσκονται οι διακομιστές που θα προσφέρουν τις υπηρεσίες του, και την ζώνη εξωτερικού δικτύου (μη ασφαλής), όπου προσομοιώνει ένα δίκτυο αμφιβόλου ασφάλειας όπως το Διαδίκτυο. Το IDS θα πρέπει να ελέγχει την δικτυακή κίνηση που ανταλλάσσεται μεταξύ αυτών των δυο ζωνών και να αναφέρει τις προσπάθειες για εκμετάλλευση του δικτύου μας.

- Να ρυθμιστεί κατάλληλα το IDS ώστε να παρέχει γρήγορα και αξιόπιστα αποτελέσματα.

Στα σύγχρονα δίκτυα όπου οι ταχύτητες πλέον είναι της τάξης του 1Gb/s ένα IDS θα πρέπει να έχει την δυνατότητα να μπορεί να παρακολουθεί την δικτυακή κίνηση χωρίς να χάνει πακέτα λόγω του μεγάλου όγκου δεδομένων που διακινούνται. Αν ένα IDS δεν μπορεί να ανεχτεί τις μεγάλες ταχύτητες διακίνησης των δεδομένων τότε δεν θεωρείται αξιόπιστο αφού δεν θα είναι σε θέση να ανιχνεύσει όλες τις επιθέσεις που συμβαίνουν.

- Να υπάρχει η δυνατότητα νόμιμης δικτυακής κίνησης.

Το μεγαλύτερο μέρος της δικτυακής κίνησης προς ένα δίκτυο υπολογιστών,ευτυχώς, είναι νόμιμη κίνηση προς τις υπηρεσίες που προσφέρει και δεν αποτελεί κίνδυνο για αυτές. Ένα IDS θα πρέπει να είναι σε θέση να διακρίνει την νόμιμη κίνηση και να μην ενημερώνει από λάθος τους διαχειριστές ότι το δίκτυο τους είναι υπό επίθεση εάν κάτι τέτοιο δεν είναι πραγματικότητα (False Positives).

- Το IDS να είναι ασφαλές από επιθέσεις.

Θα πρέπει να εξασφαλιστεί η ακεραιότητα του IDS και να γίνουν προσπάθειες ώστε να μην είναι δυνατή η ανίχνευση του. Αν το IDS είναι μέρος του δικτύου και υπάρχει πρόσβαση σε αυτό τότε ο εισβολέας θα προσπαθήσει να επιτεθεί σε αυτό είτε για να το απενεργοποιήσει είτε για να καλύψει τις επιθέσεις του αφήνοντας λειτουργικό.

- Ο έλεγχος του IDS να γίνεται από ξεχωριστό υπολογιστή.

Η πρόσβαση του IDS θα πρέπει να είναι εφικτή μόνο από το κατάλληλα εξουσιοδοτημένο προσωπικό και να γίνεται μέσω ενός ξεχωριστού υποδικτύου διαχείρισης. Οι διαχειριστές δικτύων συνήθως χρησιμοποιούν ένα ξεχωριστό υποδίκτυο όπου λειτουργούν προγράμματα ελέγχου του δικτύου και στο οποίο η πρόσβαση είναι δυνατή μόνο από συγκεκριμένους υπολογιστές.

- Να παρέχονται υπηρεσίες στο εξωτερικό δίκτυο.

Το δίκτυο που θα κατασκευάσουμε θα πρέπει να προσφέρει στους εξωτερικούς χρήστες υπηρεσίες, όπως ιστοσελίδες, FTP. Το σημαντικό είναι να έχουμε ένα ενεργό δίκτυο όπου θα υπάρχει έντονη δικτυακή κίνηση και όχι ένα σταθερό μοντέλο όπου η μοναδική κίνηση θα είναι αυτή που κατασκευάζουμε για να μετρήσουμε την αξιοπιστία του IDS.

- Να παρέχεται ασφάλεια στο εσωτερικό δίκτυο.

Το υπολογιστικό σύστημα που θα χρησιμοποιήσουμε ως δρομολογητή για τα δυο δίκτυα θα πρέπει να προστατεύει το δίκτυο φιλτράροντας την εισερχόμενη κίνηση. Για να γίνει αυτό θα χρησιμοποιηθούν λίστες πρόσβασης που θα αποκόπτουν την κίνηση προς τις υπηρεσίες των διακομιστών που δεν χρησιμοποιούνται.

6.2 Υλοποιώντας το πειραματικό περιβάλλον μας.

Έχοντας προσδιορίσει τις διάφορες απαιτήσεις του συστήματος μας, επόμενο βήμα είναι ή ανάπτυξη του δικτύου με βάση αυτές. Στην συγκεκριμένη ενότητα θα περιγράψουμε το δίκτυο, τα υπολογιστικά συστήματα που το αποτελούν καθώς και το λογισμικό που χρησιμοποιήθηκε σε αυτά ώστε να τα καταστήσει λειτουργικά.

6.2.1 Υλοποιώντας το εσωτερικό δίκτυο.

Σύμφωνα με τις απαιτήσεις που έχουμε ήδη αναφέρει παραπάνω, θα σχεδιάσουμε το εσωτερικό μας δίκτυο.

6.2.1.1 Διακομιστές υπηρεσιών.

Στο δίκτυο μας εγκαταστήσαμε δυο υπολογιστές για να προσφέρουν υπηρεσίες ιστοσελίδων και FTP στο εξωτερικό δίκτυο σύμφωνα με τις απαιτήσεις σχεδίασης του εργαστηριακού μας περιβάλλοντος. Ο κάθε ένας από τους διακομιστές φέρει διαφορετικό λειτουργικό σύστημα, ο ένας

χρησιμοποιεί λειτουργικό Ubuntu 7.04 Feisty Fawn την έκδοση για διακομιστές (Server Edition) και ο δεύτερος Windows XP Professional.

Στον Linux διακομιστή εγκαταστήσαμε τον πρόγραμμα διακομιστή ιστοσελίδων Apache2 2.2.3, την γλώσσα προγραμματισμού για δημιουργία δυναμικών ιστοσελίδων PHP 5.2.1, τον διακομιστή βάσης δεδομένων MySQL 5.0.38 και εγκαταστήσαμε την πλατφόρμα ασύγχρονης τηλε-εκπαίδευσης GUNET eClass 2.0 όπου δημιουργήσαμε ψηφιακό περιεχόμενο. Για την παροχή υπηρεσιών FTP χρησιμοποιήσαμε τον διακομιστή ProftpD.

Στον διακομιστή με λειτουργικό περιβάλλον Windows χρησιμοποιήσαμε ως πρόγραμμα διακομιστή ιστοσελίδων το IIS 6.0 (Internet Information Server), για γλώσσα προγραμματισμού δημιουργίας δυναμικών ιστοσελίδων καθώς και διακομιστή βάσης δεδομένων επιλέξαμε επίσης την PHP και MySQL αντίστοιχα, λόγω προηγούμενης εμπειρίας. Επίσης εγκαταστήσαμε το πρόγραμμα διαχείρισης περιεχομένου Joomla! 1.0.13 για την δημιουργία και παροχή υπηρεσιών ιστοσελίδων.

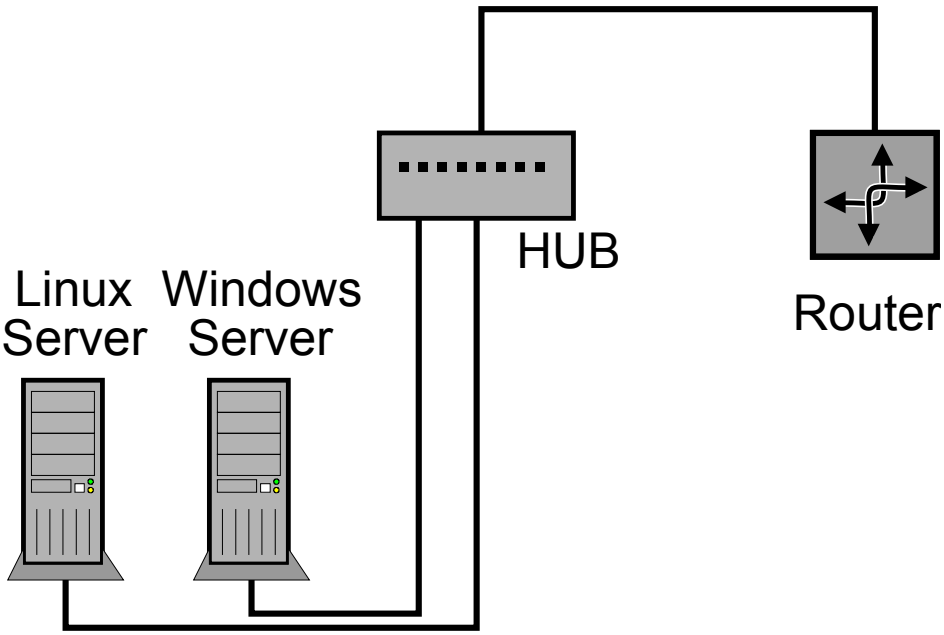
Κάνοντας χρήση δυο διαφορετικών λειτουργικών συστημάτων επιτυγχάνεται η τήρηση των προδιαγραφών και δημιουργείται ένα αληθοφανές δίκτυο

6.2.1.2 Δρομολογητής.

Για να μεταφέρονται να πακέτα μεταξύ του εσωτερικού και του εξωτερικού δικτύου που αποτελούν το πειραματικό μας περιβάλλον, χρησιμοποιήσαμε έναν υπολογιστή με δυο κάρτες δικτύου των 100 Mb/s. Στο εσωτερικό μας δίκτυο έχουμε αναθέσει τις διευθύνσεις IP 192.168.1.0/24 ενώ το εξωτερικό δίκτυο μπορεί να πάρει διευθύνσεις IP στο υποδίκτυο 172.16.1.0/24. Στον υπολογιστή που χρησιμοποιήθηκε για να εκτελεί χρέη δρομολογητή εγκαταστάθηκε το λειτουργικό Linux Slackware έκδοση 12.0 στο οποίο και δημιουργήθηκαν οι κανόνες δρομολόγησης για την σωστή μεταφορά των πακέτων ανάμεσα στο δύο δίκτυα.

Στην κατάσταση που βρίσκεται το δίκτυο όλα τα δικτυακά πακέτα που στέλνονται σε κάποιον από τους υπολογιστές που ανήκουν στο υποδίκτυο, παραδίδονται σε αυτούς χωρίς κανένα φιλτράρισμα. Αποτέλεσμα αυτού είναι οποιοσδήποτε από το εξωτερικό δίκτυο να μπορεί να επιτεθεί εύκολα σε αυτούς τους διακομιστές. Για να αυξήσουμε το επίπεδο ασφαλείας του εσωτερικού μας δικτύου σύμφωνα με τις απαιτήσεις του κεφαλαίου 7.1 και να περιορίσουμε την πρόσβαση σε αυτό μόνο στις υπηρεσίες που προσφέρει, εφαρμόσαμε λίστες ελέγχου πρόσβασης με τυπική συμπεριφορά άρνησης όλων των εισερχόμενων συνδέσεων εκτός από αυτές που αφορούν τις προσφερόμενες υπηρεσίες.

Στο σχήμα 11 φαίνεται το εργαστηριακό δίκτυο μας στην παρούσα κατάσταση με τους διακομιστές και τον δρομολογητή.



Σχήμα 11. Το εσωτερικό δίκτυο του εργαστηριακού περιβάλλοντος.

6.2.1.3 Σύστημα ανίχνευσης δικτυακών εισβολών Snort.

Η υλοποίηση του Snort, σύμφωνα με τις προδιαγραφές, έγινε σε ένα Pentium Core 2 Duo 6550 (2.33 GHz) με 2GB μνήμης και δύο κάρτες δικτύου. Στον συγκεκριμένο σύστημα εγκαταστήσαμε το λειτουργικό σύστημα Ubuntu 7.04 Feisty Fawn την έκδοση για διακομιστές. Επίσης χρησιμοποιήσαμε την βάση δεδομένων MySQL έκδοση 5.0.38, την γλώσσα προγραμματισμού PHP έκδοση 5.2.1 και για πρόγραμμα διακομιστή ιστοσελίδων τον Apache 2 έκδοση 2.2.3.

Η εγκατάσταση του Snort έγινε με την έκδοση 2.8.0 του λογισμικού, το οποίο μεταγλωττίστηκε στο σύστημα μας καθώς και έγινε η αναγκαία παραμετροποίηση ώστε να προσαρμοστεί στο δίκτυο μας. Η διαδικασία εγκατάστασης του λογισμικού IDS καθώς και η παραμετροποίηση του παρουσιάζεται στο παράρτημα “Β”.

Έχοντας ως προϋπόθεση την γρήγορη και απρόσκοπτη λειτουργία του συστήματος μας, επιλέξαμε για plug-in εξόδου την λειτουργία unified, η οποία όπως έχουμε ήδη αναφέρει παραπάνω παρέχει στο πρόγραμμα μας την δυνατότητα να αποθηκεύσει τα δεδομένα με τον γρηγορότερο τρόπο.

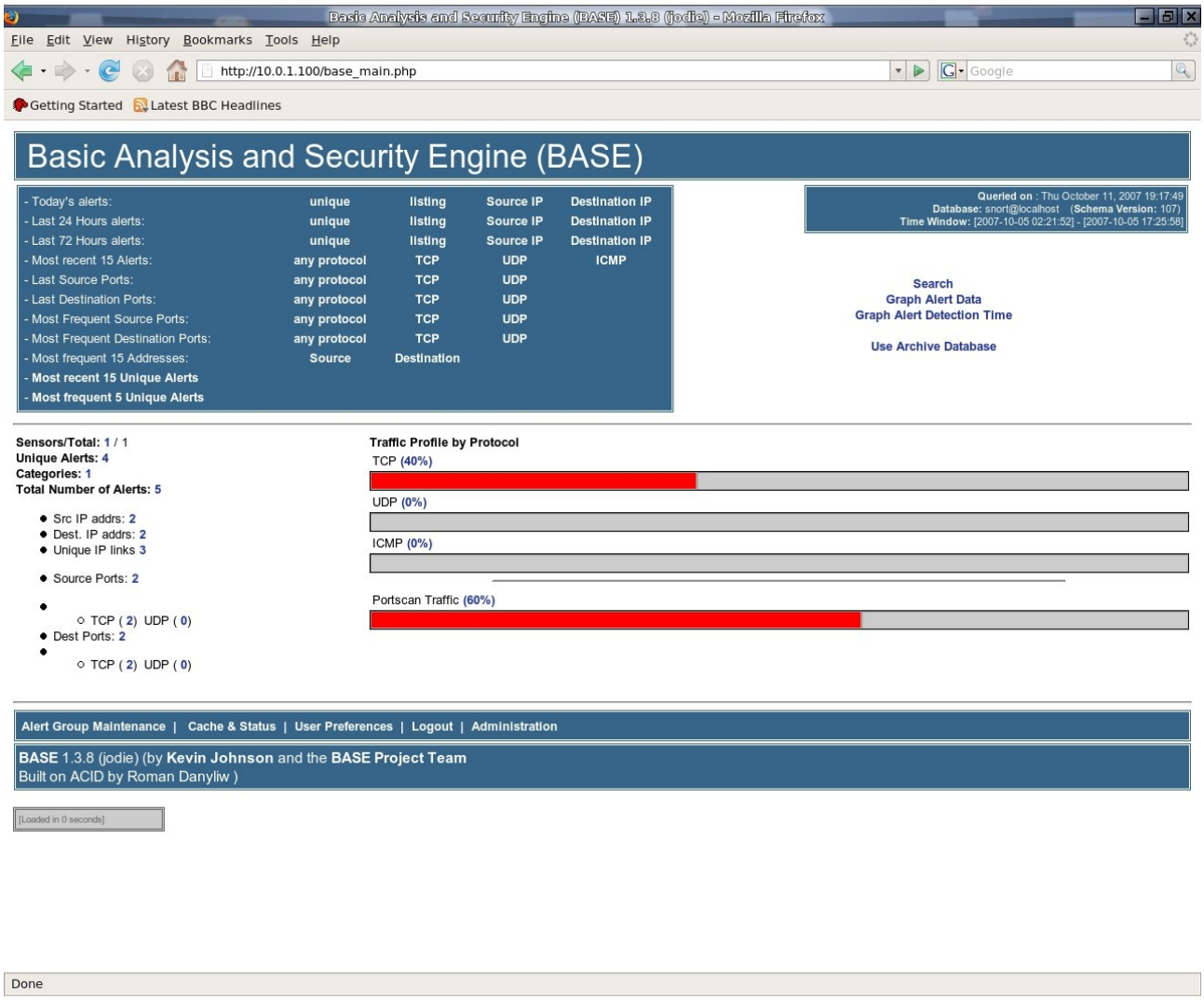
Χρησιμοποιώντας την συγκεκριμένη προσέγγιση αποτύπωσης εξόδου του Snort κρίθηκε επιτακτική η χρήση ενός εξωτερικού προγράμματος για την αποθήκευση των δεδομένων αυτών σε βάση δεδομένων. Το πρόγραμμα που ανέλαβε αυτή την υποχρέωση είναι το Barnyard στην έκδοση

0.2.0 που βρίσκεται αυτή την στιγμή επίσης έχει την δυνατότητα να αναγνωρίζει την δυαδική μορφή που χρησιμοποιεί το Snort και να το μετατρέπει σε μορφή ASCII που είναι ευκολότερο να αναγνωσθεί από τους ανθρώπους. Τέλος πρέπει να αναφέρουμε ότι το Barnyard έχει γνώση της κατάστασης που βρίσκεται η MySQL βάση δεδομένων, με αποτέλεσμα να αποθηκεύει τα δεδομένα σίγουρα και χωρίς απώλειες.

Σε αυτό σημείο, έχουμε ένα λειτουργικό περιβάλλον ανίχνευσης δικτυακών απειλών το οποίο είναι σε θέση να ανιχνεύσει και να αποθηκεύσει τα δικτυακά δεδομένα που χρίζουν προσοχής σε μια βάση δεδομένων. Αν και λειτουργικό το συγκεκριμένο σύστημα χρειάζεται ένα γραφικό περιβάλλον χρήσης που να έχει την δυνατότητα να επεξεργάζεται τα δεδομένα που υπάρχουν αποθηκευμένα στην βάση και να μας τα προβάλει σύμφωνα με τα κριτήρια που θα επιλέξουμε. Την λύση έρχεται να μας την δώσει το BASE (Basic Analysis and Security Engine), το οποίο είναι ο διάδοχος του ACID (Analysis Console for Intrusion Databases), που είναι μια μηχανή ανάλυσης των συμβάντων ασφάλειας που είναι αποθηκευμένα σε μια βάση δεδομένων και είναι γραμμένη στην γλώσσα προγραμματισμού PHP. Στο σχήμα 12 βλέπουμε ένα στιγμιότυπο από την δοκιμαστική λειτουργία του BASE πριν αρχίσει η διαδικασία μετρήσεων. Η χρήση του BASE μας προσφέρει τα ακόλουθα πλεονεκτήματα:

- μια διασύνδεση για την αναζήτηση στην βάση δεδομένων καθώς και τη δημιουργία ερωτημάτων Η αναζήτηση μπορεί να γίνει με παραμέτρους που αφορούν το δίκτυο, όπως η διεύθυνση IP, ή με χρονικούς παραμέτρους.
- έναν περιηγητή πακέτων με την δυνατότητα αποκωδικοποίησης των εμφάνισης των πληροφοριών του επιπέδου 3 και 4 των αποθηκευμένων πακέτων.
- Δυνατότητες διαχείρισης δεδομένων, συμπεριλαμβανομένων της ομαδοποίησης των συμβάντων ασφαλείας, της διαγραφής συμβάντων ασφαλείας και ειδοποίησης μέσω μηνυμάτων ηλεκτρονικού ταχυδρομείου.
- Παρέχει την δυνατότητα δημιουργίας διαφόρων γραφημάτων και στατιστικών με βάση τις παραμέτρους που θα καθορίσουμε.

Η εγκατάσταση και παραμετροποίηση του BASE παρουσιάζεται στο Παράρτημα “Β”.

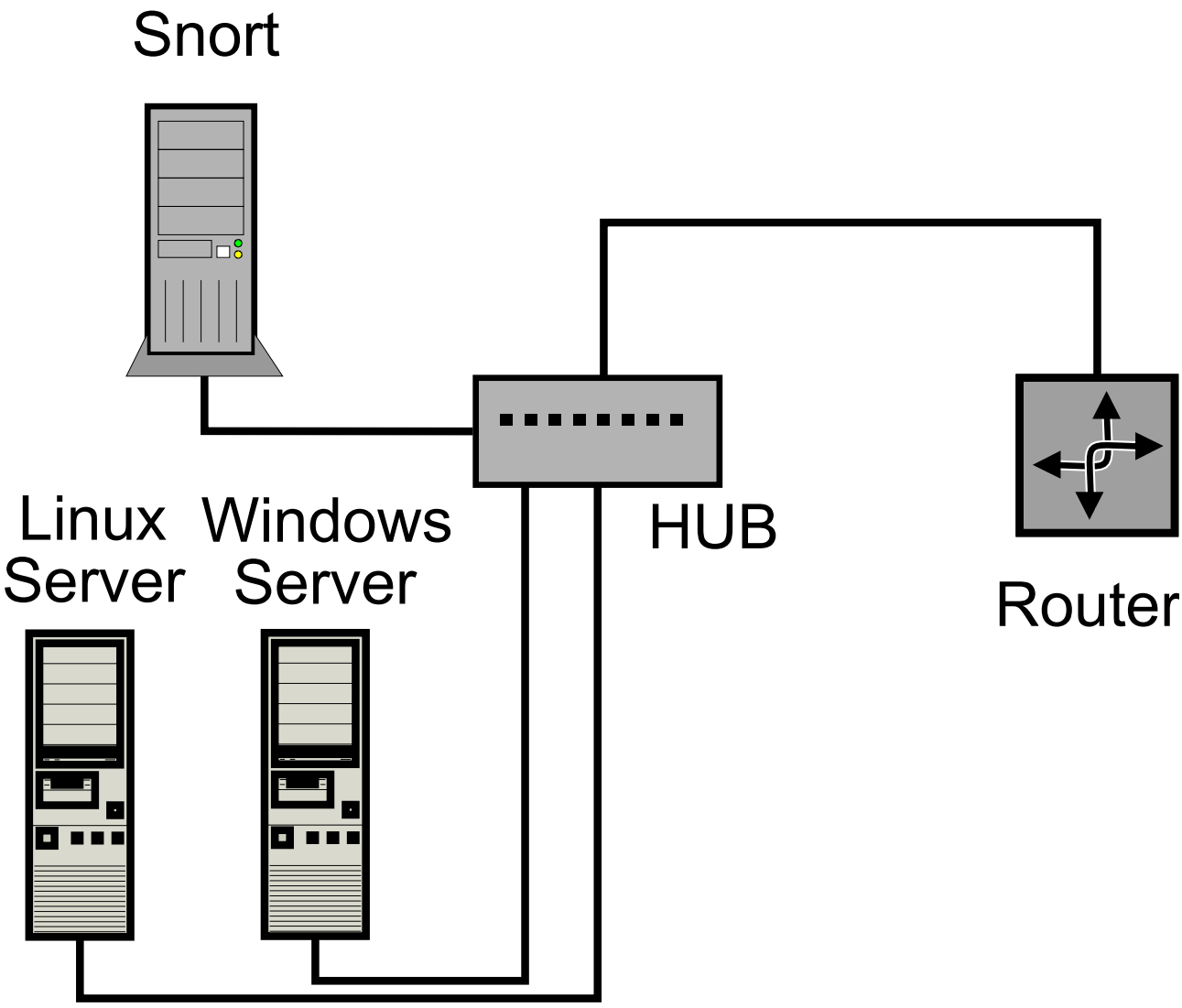


Σχήμα 12. Στιγμιότυπο από την κονσόλα διαχείρισης BASE.

Όπως έχει αναφερθεί ήδη στις προδιαγραφές του δικτύου, βασική προϋπόθεση είναι η προστασία του Snort από κακόβουλους χρήστες του εσωτερικού ή εξωτερικού δικτύου. Για να επιλύσουμε το συγκεκριμένο πρόβλημα δεν αναθέτουμε διεύθυνση IP στην κάρτα δικτύου που είναι υπεύθυνη για την σύλληψη της δικτυακής κίνησης. Σε αυτή την περίπτωση ο υπολογιστής δεν έχει την δυνατότητα να επικοινωνήσει με τα υπόλοιπα συστήματα του δικτύου, μπορεί όμως να παρακολουθεί την δικτυακή κίνηση μπαίνοντας σε λειτουργία promiscuous.

Στον ίδιο υπολογιστή έχουμε τοποθετήσει και μια δεύτερη κάρτα δικτύου, η οποία μας παρέχει την δυνατότητα επικοινωνίας με το IDS μέσω ενός υποδικτύου που χρησιμοποιείτε αποκλειστικά για την διαχείριση του. Επομένως, υπηρεσίες που επιτρέπουν πρόσβαση στο σύστημα μας, όπως ο Secure Shell διακομιστής ο οποίος μας δίνει την δυνατότητα απομακρυσμένης πρόσβασης στο Linux ή ο διακομιστής ιστοσελίδων Apache 2, συνδέονται αποκλειστικά με την δεύτερη κάρτα δικτύου.

Η προσθήκη του Snort στο εσωτερικό μας δίκτυο φαίνεται στο παρακάτω σχήμα.



Σχήμα 13. Ολοκληρωμένη αποτύπωση του εσωτερικού δικτύου.

6.2.2 Υλοποιώντας το εξωτερικό δίκτυο.

Είναι το τμήμα του δικτύου που θα δημιουργεί την κίνηση προς τις υπηρεσίες που προσφέρουν οι διακομιστές μας. Χρησιμοποιείτε για να προσομοιωθούν οι χρήστες του πειραματικού περιβάλλοντος μας δίνοντας μας την ευκαιρία να μελετήσουμε το δίκτυο μας με πραγματική κίνηση.

6.2.2.1 Προσομοίωση χρήσης.

Έχοντας ως σκοπό την μελέτη απόδοσης του Snort σε διαφορετικές συνθήκες δικτυακής κίνησης είναι απαραίτητο να διακινούνται δεδομένα από και προς τους διακομιστές υπηρεσιών. Για την επίτευξη του συγκεκριμένου εγχειρήματος επιστρατεύτηκε ένα υπολογιστικό σύστημα με λειτουργικό σύστημα Windows XP Pro καθώς και κατάλληλο λογισμικό για την δημιουργία της δικτυακής κίνησης.

Στην προσπάθεια μας να προσομοιώσουμε ένα πραγματικό περιβάλλον και να ικανοποιήσουμε την προδιαγραφή για ύπαρξη νόμιμης κίνησης στο δίκτυο μας, χρησιμοποιήσαμε λογισμικό το οποίο δημιουργούσε αιτήσεις προς την υπηρεσία ιστοσελίδων των διακομιστών ανά τακτά χρονικά διαστήματα. Λόγω ότι στο συγκεκριμένο λογισμικό (Nsauditor) δεν ήταν εφικτή η είσοδος πολλαπλών διευθύνσεων ιστοσελίδων με την χρήση εξωτερικού αρχείου χρησιμοποιήθηκαν πολλαπλά στιγμιότυπα του προγράμματος. Ταυτόχρονα με τις συνδέσεις πρωτοκόλλου HTTP, το σύστημα μας δημιουργούσε συνδέσεις στην υπηρεσία FTP των διακομιστών όπου και κατέβαζε δεδομένα.

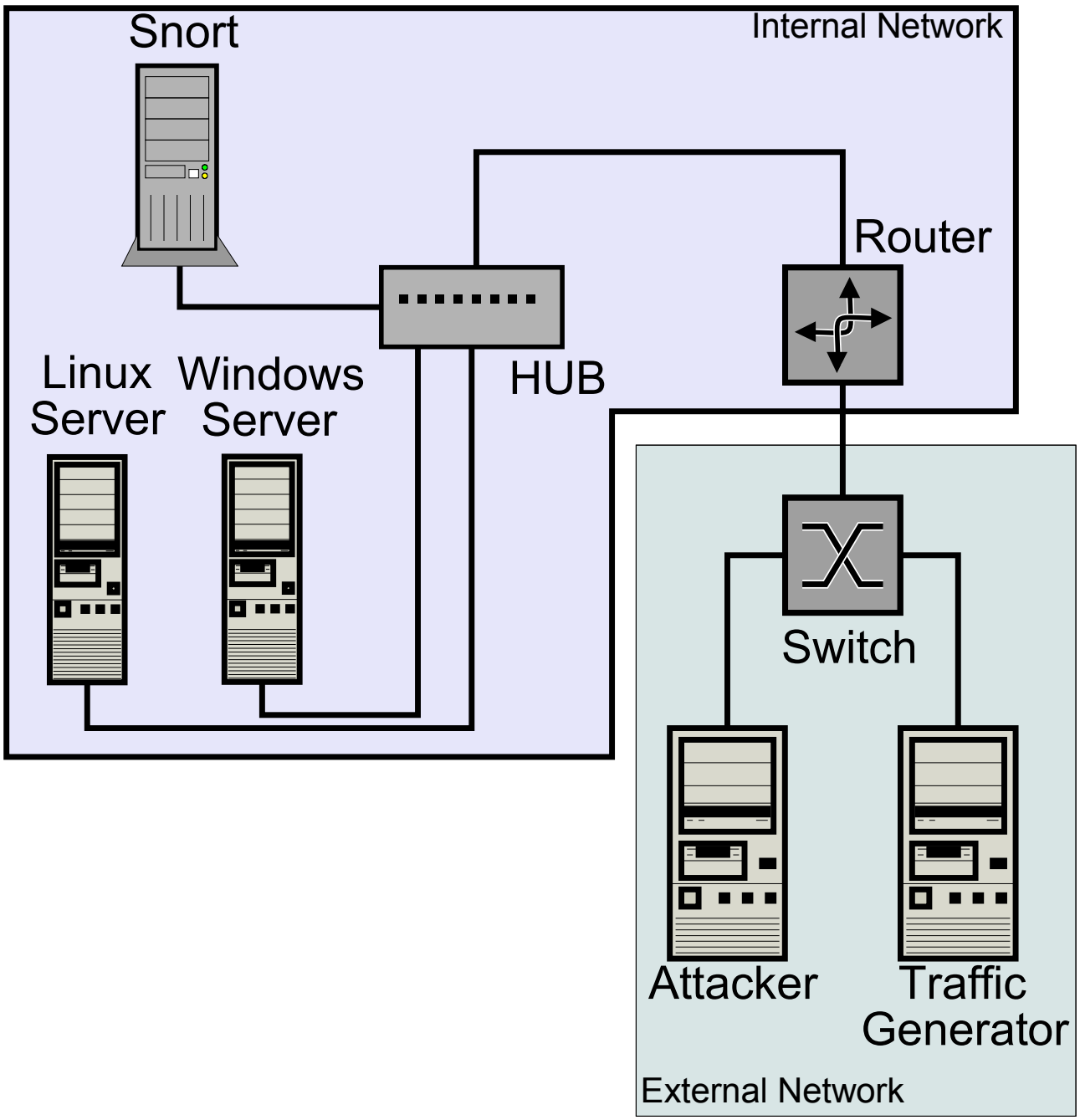
6.2.2.2 Προσομοίωση επίθεσης

Τελευταίο στοιχείο του πειραματικού μας περιβάλλοντος είναι το υπολογιστικό σύστημα που ήταν υπεύθυνο για τις επιθέσεις στο δίκτυο μας. Στον συγκεκριμένο υπολογιστή είχαν εγκατασταθεί τα Windows XP Pro, το Ubuntu Linux έκδοση 7.04 καθώς και αρκετά διαφορετικά πακέτα λογισμικού που παρέχουν την δυνατότητα δικτυακών επιθέσεων. Από τα πολλά προγράμματα που χρησιμοποιήθηκαν, τα περισσότερο χρήσιμα ήταν τα:

- Nmap. Είναι ένα δωρεάν πρόγραμμα που επιτελεί σαρώσεις ασφαλείας σε υπολογιστές, με σκοπό να ανιχνεύσει τις διαθέσιμες υπηρεσίες των διακομιστών καθώς και τις εκδόσεις των εφαρμογών των υπηρεσιών και του λειτουργικού συστήματος.
- Nessus. Είναι ένα περιεκτικό πρόγραμμα ελέγχου προβλημάτων ασφαλείας. Σκοπός του είναι να αναφέρει πιθανές ή επιβεβαιωμένες αδυναμίες στους υπολογιστές που εξετάζει.
- Nikto. Είναι ένα ανοικτού κώδικα πρόγραμμα ελέγχου προβλημάτων ασφαλείας διακομιστών ιστοσελίδων το οποίο ενσωματώνει με την βοήθεια της βιβλιοθήκης LibWhisker πολυάριθμες τεχνικές αποφυγής ανίχνευσης από IDS.

6.3 Συνολικό δίκτυο.

Ενώνοντας τα επιμέρους δίκτυα ολοκληρώνουμε το πειραματικό μας περιβάλλον όπως φαίνεται στο σχήμα 14. Στο επόμενο κεφάλαιο θα ασχοληθούμε με την διαδικασία των μετρήσεων που έγιναν στο δίκτυο που κατασκευάσαμε.



Σχήμα 14. Αναπαράσταση της τελικής μορφής του εργαστηριακού περιβάλλοντος.

7. Μετρήσεις Απόδοσης.

Με την ολοκλήρωση του δικτύου μας, είμαστε πλέον σε θέση να περιγράψουμε την διαδικασία των μετρήσεων καθώς και εκθέσουμε τα αποτελέσματα του προγράμματος ανίχνευσης δικτυακών απειλών Snort. Πρέπει να αναφέρουμε ότι δεν υπάρχουν σαφής οδηγίες διεξαγωγής δοκιμών στο πεδίο λογισμικού των IDS αν και κατά καιρούς έχουν γίνει προσπάθειες στον συγκεκριμένο τομέα. Στις δοκιμές, έγινε προσπάθεια να ελεγχθούν διάφοροι τομείς του Snort ώστε να παρέχουμε ένα αντικειμενικό αποτέλεσμα.

Η διαδικασία μετρήσεων περιείχε τρεις διαφορετικές δοκιμασίες, όπου η κάθε μια στοχεύει στο να παρέχει αποτελέσματα για τους τρεις τομείς που ελέγξαμε. Η πρώτη δοκιμή έχει σκοπό την αξιολόγηση του Snort στο τομέα ανίχνευσης εισβολών, η δεύτερη στην απρόσκοπτη λειτουργία του υπό φορτίο και η τρίτη επικεντρώνεται στην δυνατότητα του να ανιχνεύει επιθέσεις που χρησιμοποιούν τεχνικές αποφυγής IDS.

7.1 1η μέτρηση: Δυνατότητες ανίχνευσης επιθέσεων.

Στην συγκεκριμένη δοκιμή στόχος είναι ο έλεγχος της αποτελεσματικότητας του Snort στην ανίχνευση επιθέσεων. Δεδομένου αυτού όλες οι επιθέσεις έγιναν χωρίς να υπάρχει επιπλέον κίνηση στο δίκτυο που θα μπορούσε να επηρεάσει το IDS, καθώς και οι υπηρεσίες που ήταν στόχος επιθέσεων λειτουργούσαν κανονικά. Βασικός κανόνας αναγνώρισης για μια επίθεση έχει τεθεί η σωστή αναγνώριση του από το πρόγραμμα με τρόπο που να ορίζει συγκεκριμένα ποια επίθεση συνέβη.

Οι επιθέσεις που χρησιμοποιήθηκαν καλύπτουν τους εξής τομείς:

- Σάρωση δικτυακών θυρών (Port Scans)
- Άρνηση υπηρεσιών (Denial of Service)
- Διακομιστές Ιστοσελίδων (Web)
- Διακομιστές FTP (FTP)

Τα προγράμματα που χρησιμοποιήθηκαν για τις επιθέσεις ήταν το Nikto, το Nessus, το Nmap καθώς και διάφορα exploits που μπορούν να βρεθούν εύκολα στο διαδίκτυο.

Θα πρέπει να αναφέρουμε ότι σε καμία περίπτωση δεν μπορεί να ελεγχθεί το σύνολο των υπογραφών του Snort καθώς και επίσης και να οριστεί ένα υποσύνολο επιθέσεων που με βάση αυτό να μπορέσει να μετρηθεί η συνολική απόδοση ενός IDS. Για τον λόγο αυτό επιλέξαμε επιθέσεις που

είναι εύκολο να βρεθούν στο διαδίκτυο (ως ιδέα, ως πηγαίος κώδικας ή πρόγραμμα) και μπορεί κάποιος να τις χρησιμοποιήσει ενάντια σε υπολογιστικά δίκτυα.

7.2 2η μέτρηση: Λειτουργία υπό φορτίο.

Σε αυτή την μέτρηση σκοπός μας είναι να δοκιμάσουμε το Snort σε διάφορες συνθήκες δικτυακής κίνησης ώστε να εκτιμήσουμε την απόδοση του σε ένα πραγματικό δίκτυο. Για να γίνουν οι μετρήσεις χρησιμοποιήσαμε μια τροποποιημένη έκδοση του προγράμματος Nikto, το οποίο εκτελεί δικτυακές επιθέσεις σε διακομιστές ιστοσελίδων. Στην τροποποιημένη αυτή έκδοση επιλέξαμε μια επίθεση την οποία επαναλάβαμε 10.000 φορές κατά την διάρκεια της δοκιμής και μετά την ολοκλήρωση ελέγξαμε τον αριθμό επιθέσεων που αναγνώρισε το Snort. Το πείραμα επαναλήφθηκε σε τέσσερα διαφορετικά επίπεδα δικτυακής κίνησης, με μηδενικό φόρτο, με φόρτο 25Mbit, με φόρτο 50Mbits και με 75Mbits ανά δευτερόλεπτο.

Η δικτυακή κίνηση που δημιουργείται αποτελείται από πακέτα των πρωτοκόλλων HTTP και FTP καθώς και ένα μικρό μέρος από πακέτα ICMP. Τα πακέτα αυτά δημιουργούνται κυρίως από τον υπολογιστή που είναι επιφορτισμένος με την δημιουργία της δικτυακής κίνησης και έχουν ως αποδέκτες και τους δύο διακομιστές που έχουν τοποθετηθεί στο εσωτερικό δίκτυο.

7.3 3η μέτρηση: Τεχνικές αποφυγής εντοπισμού επιθέσεων.

Σκοπός της συγκεκριμένης δοκιμής είναι η εξέταση των δυνατοτήτων του Snort στο εντοπισμό των επιθέσεων όταν αυτές χρησιμοποιούν τεχνικές αποφυγής εντοπισμού. Για την εν λόγω δοκιμή χρησιμοποιήθηκε το πρόγραμμα ελέγχου συμβάντων ασφαλείας Nikto, του οποία η συνεργασία με την βιβλιοθήκη LibWhisker του επιτρέπει την χρήση δυνατοτήτων αποφυγής εντοπισμού των επιθέσεων που εκτελεί.

Στο πειραματικό μας περιβάλλον εκτελέσαμε ένα υποσύνολο των αρχικών επιθέσεων που χρησιμοποιήθηκε στην 1η μέτρηση επιχειρώντας να “κρύψουμε” τις επιθέσεις με την χρήση των εξής μεθόδων:

- Random URI Encoding (Τυχαία κωδικοποιήση του URI)
- Directory Self-reference /./ (Δήλωση του καταλόγου /./)
- Premature URL Ending (Πρόωρη τερματισμός του URL)

- Fake Parameters to files (Ψεύτικοι παράμετροι σε αρχεία)
- Session Spliting (Χωρισμός των συνεδριών)

Στην συνέχεια επαναλάβαμε τις επιθέσεις στον διακομιστή κατακερματίζοντας τα δικτυακά πακέτα με την βοήθεια του fragrouter 1.6-2.2. Το συγκεκριμένο πρόγραμμα έχει την δυνατότητα να λαμβάνει τα δικτυακά πακέτα και στην συνέχεια αφού τα επεξεργαστεί σύμφωνα με τις επιλογές μας τα δρομολογεί στο δίκτυο. Για την δοκιμή, το fragrouter εκτελούταν στον υπολογιστή που εκτελεί χρέη δρομολογητή και πραγματοποιήθηκαν οι εξής αλλαγές στα πακέτα:

- Ordered 8-byte IP fragments. (Κομμάτια IP των 8-byte σε σειρά)
- Ordered 8-byte IP fragments, one out of order. (Κομμάτια IP των 8-byte σε σειρά, ένα εκτός σειράς)
- Ordered 8-byte IP fragments, one duplicate. (Κομμάτια IP των 8-byte σε σειρά, ένα αντίγραφο)
- Out of order 8-byte IP fragments, one duplicate. (Κομμάτια IP των 8-byte εκτός σειράς, ένα αντίγραφο)

8 Αποτελέσματα μετρήσεων.

Σε αυτήν την ενότητα συνοψίζονται τα αποτελέσματα των μετρήσεων απόδοσης για το Snort υπό την μορφή πινάκων. Όπως γίνεται αντιληπτό και από τους παρακάτω πίνακες το λογισμικό μας κατάφερε να επικρατήσει σε όλες τις δοκιμές.

1η Μέτρηση : Δυνατότητες ανίχνευσης επιθέσεων.	**Επιθέσεις που πραγματοποιήθηκαν.**	**Επιθέσεις που αναγνωρίστηκαν.**
Σαρώσεις δικτυακών θυρών (Port Scans)	6	6
Αρνηση Υπηρεσιών (Denial of Service - DoS)	12	12
Διακομιστές Ιστοσελίδων (WWW exploits)	80	80
Διακομιστές FTP (FTP exploits)	22	22
Σύνολο	**120**	**120**

2η μέτρηση: Λειτουργία υπό φορτίο.	**0%**	**25%**	**50%**	**75%**
Υποσύνολο επιθέσεων.	100	100	100	100

3η μέτρηση: Τεχνικές αποφυγής εντοπισμού επιθέσεων.	**Επιθέσεις που πραγματοποιήθηκαν.**	**Επιθέσεις που αναγνωρίστηκαν.**
Nikto με χρήση LibWhisker	20	20
Fragrouter	20	20
Σύνολο	**40**	**40**

Όπως φαίνεται και στους πίνακες των αποτελεσμάτων το Snort κατάφερε να φέρει εις πέρας όλες τις δοκιμασίες του με άριστα αποτελέσματα παρόλο τον χαρακτηρισμό του ως “δωρεάν” πρόγραμμα. Είναι μια πολύ αξιόλογη λύση για δίκτυα ταχυτήτων μέχρι 100Mbit που είχαμε την δυνατότητα να μετρήσουμε και είναι πολύ πιθανόν να μπορεί να σταθεί επάξια και σε γρηγορότερα ακόμα περιβάλλοντα.

Η απόδοση του όπως φαίνεται δεν επηρεάζεται από τεχνικές αποφυγής επιθέσεων μίας και κατάφερε να αναγνωρίσει όλες τις επιθέσεις που δοκιμάσαμε και φαίνεται ότι έχει διορθώσει παλαιότερες αδυναμίες του στο συγκεκριμένο τομέα.

Σύμφωνα, επομένως, με όλα τα παραπάνω το Snort θεωρείται μια συμφέρουσα επιλογή για τον έλεγχο ενός δικτύου από τους κακόβουλους χρήστες. Είναι ένα εργαλείο που παρέχει πολλές δυνατότητες που όταν εγκατασταθεί και παραμετροποιηθεί σωστά, παρέχει στους διαχειριστές δικτύων πολύτιμη βοήθεια για την προστασία του δικτύου.

9 Συμπεράσματα.

Η εποχή μας μαστίζεται από μια αυξανόμενη τάση για δικτύωση που προσφέρει τρομερή ευκολία στην μετάδοση των πληροφοριών δημιουργώντας ταυτόχρονα σημαντικούς κινδύνους ασφαλείας των δεδομένων που υπάρχουν στα δικτυωμένα συστήματα. Η διαχείριση τόσο σε επίπεδο λειτουργίας αλλά και ασφάλειας του δικτύου είναι μια χρονοβόρα διαδικασία η οποία απασχολεί το μεγαλύτερο μέρος του χρόνου ενός διαχειριστή δικτύου. Τα συστήματα ανίχνευσης δικτυακών εισβολών έχουν δημιουργηθεί για να παρακολουθούν σε 24ώρη βάση το δίκτυο που πρέπει να προστατέψουν και αναφέρουν την ύποπτη δραστηριότητα στους διαχειριστές δικτύων για περαιτέρω ανάλυση. Η αδιάλειπτη λειτουργία τους δίνει την δυνατότητα για ουσιαστικότερο έλεγχο αλλά και γνώση των αδυναμιών του δικτύου.

Το Snort λειτούργησε στο εργαστηριακό περιβάλλον πέραν των προσδοκιών μας και κατάφερε να εντοπίσει όλες τις επιθέσεις μας. Ένα σύστημα ανίχνευσης δικτυακών απειλών σαν το Snort μπορεί αναμφίβολα να προστατέψει ένα δίκτυο καθώς και να παρέχει σημαντικά στοιχεία για την λειτουργία του. Θα πρέπει επίσης να αναφέρουμε ότι το Snort εκτός από την χρήση του ως IDS μπορεί να λειτουργήσει και ως σύστημα πρόληψης δικτυακών εισβολών (Intrusion Prevention System – IPS). Σε αυτή την περίπτωση το Snort έχει την δυνατότητα να σταματήσει όποιες συνδέσεις θεωρεί επιθέσεις αυξάνοντας έτσι την συνολική ασφάλεια του δικτύου.

Με την βοήθεια του Κέντρου Ελέγχου και Διαχείρισης Δικτύου του ΤΕΙ Κρήτης, ενσωματώσαμε στο δίκτυο του το σύστημα Snort σε μια προσπάθεια να αποκτήσουμε βαθύτερη γνώση για την λειτουργία του δικτύου. Σύμφωνα με τις δοκιμές που έχουν γίνει μέχρι σήμερα, το Snort αν και μπορεί να παρακολουθήσει το δίκτυο του ΤΕΙ (ταχύτητα δικτύου 100 Mbit/s με το Internet) έχει παρατηρηθεί μια αργοπορία του BASE στην εμφάνιση των ειδοποιήσεων. Σύμφωνα με τον έλεγχο που έγινε στο μηχάνημα πιθανότερος λόγος για το συγκεκριμένο πρόβλημα είναι τα τεχνικά χαρακτηριστικά του υπολογιστή καθώς και ο πολύ μεγάλος αριθμός από επιθέσεις που ανιχνεύονται και αποθηκεύονται στην βάση κάνοντας την mySQL να αργεί να ανταποκριθεί στις αιτήσεις μας.

Με την εισαγωγή του συστήματος στο δίκτυο του ΤΕΙ Κρήτης πλέον έχει γίνει εφικτή η αναγνώριση εξωτερικών επιθέσεων προς το Ίδρυμα μας προσφέροντας την δυνατότητα πρόληψης ορισμένων επιθέσεων. Επίσης είναι πλέον εφικτή η αναγνώριση υπολογιστικών συστημάτων που ανήκουν στο ΤΕΙ Κρήτης και κάνουν κακή χρήση των υπηρεσιών που αυτό τους προσφέρει, δίνοντας στο ΚΕΔΔ την δυνατότητα να ενημερώνει τους κατόχους τους για την παραβίαση της πολιτικής ασφαλείας του. Τέλος η δυνατότητα αποθήκευσης των πακέτων που αποτελούν

κακόβουλη κίνηση παρέχει στοιχεία για την μετέπειτα ανάλυση των επιθέσεων από το προσωπικό του ΚΕΔΔ δίνοντας τους βαθύτερη γνώση των κινδύνων που ελλοχεύουν στο Διαδίκτυο.

Συστήματα όπως το Snort παρέχουν χρήσιμες πληροφορίες για το δίκτυο, αλλά αν δεν υπάρχει κατάλληλα ενημερωμένο προσωπικό με γνώση στο πεδίο της ασφάλειας υπολογιστών και ασφάλειας δικτύων, κανένα πρόγραμμα δεν μπορεί να παρέχει επαρκεί ασφάλεια σε ένα δίκτυο. Η βασικότερη λύση για την αποφυγή δυσάρεστων καταστάσεων έλλειψης ασφάλειας είναι η γνώση και η εκπαίδευση.

10 Βιβλιογραφία.

[1] Network Intrusion Detection 3rd Edition Aug 27, 2002. Stephen Northcutt, Judy Novak. ISBN-10: 0735712654 ISBN-13: 978-0735712652

[2] Intrusion Detection with Snort 2Rev Edition May 20, 2003. Jack Koziol ISBN-10: 157870281X ISBN-13: 978-1578702817

[3] Advanced IDS Techniques Using Snort, Apache, MySQL, PHP and ACID 1st Edition May 8, 2003. Rafeeq Rehman ISBN-10: 0131407333 ISBN-13: 978-0131407336

[4] Snort for Dummies Pap/Cdr Edition July 9, 2004. Charlie Scott, Paul Wolfe, Bert Hayes. ISBN-10: 0764568353 ISBN-13: 978-0764568350

[5] Snort Intrusion Detection and Prevention Toolkit Pap/Cdr Edition February 1, 2007. Brian Caswell, Jay Beale, Andrew Baker. ISBN-10: 1597490997 ISBN-13: 978-1597490993

[6] Implementing Intrusion Detection Systems: A Hands-On Guide for Securing the Network December 11,2002. Tim Crothers. ISBN-10: 0764549499 ISBN-13: 978-0764549496

[7] Intrusion Alert: An Ethical Hacking Guide to Intrusion Detection. 1st Edition July 1, 2007. Ankit Fadia. ISBN-10: 1598634143 ISBN-13: 978-1598634143

[8] The Science of Intrusion Detection System Attack Identification. Cisco Systems 2002

[9] An Overview of Issues in Testing Intrusion Detection. June 2003. National Institute of Standards and Technology ITL: Peter Mell, Vincet Hu. Massachusetts Institute of Technology Lincoln Laboratory: Richard Lippmann, Josh Haines, Marc Zissman.

[10] A Methodology for Testing Intrusion Detection Systems. 2nd Revision 27 September 1996. Nicholas J. Puketza, Kui Zhang, Mandy Chung, Biswanath Mukherjee, Ronald A. Olsson.

[11] How to Guide – Implementing a Network Based Intrusion Detection System. Internet Security Systems 2000, Brian Laing.

[12] Deploying Network – Based Intrusion Detection. March 2004, Cisco Systems.

[13] Guide to Intrusion Detection and Prevention Systems (IDPS) February 2007. National Institute of Standards and Technology. Karen Scarfone, Peter Mell.

[14] Access Control Lists: http://en.wikipedia.org/wiki/Access_control_list

[15] Address Resolution Protocol: http://en.wikipedia.org/wiki/Address_resolution_protocol

[16] Antivirus Software: http://en.wikipedia.org/wiki/Antivirus

[17] Comma-Separated Values: http://en.wikipedia.org/wiki/Comma-separated_values

[18] Cyclic Redudancy Check: http://en.wikipedia.org/wiki/Cyclic_redundancy_check

[19] Demilitarized Zone: http://en.wikipedia.org/wiki/Demilitarized_zone_(computing)

[20] Denial of Service Attack: http://en.wikipedia.org/wiki/Denial_of_service

[21] Distributed Computing Environment: http://en.wikipedia.org/wiki/Distributed_Computing_Environment

[22] Domain name System: http://en.wikipedia.org/wiki/Dns

[23] Encapsulation: http://en.wikipedia.org/wiki/Encapsulation_(networking)

[24] Firewall: http://en.wikipedia.org/wiki/Firewall

[25] File Transfer Protocol: http://en.wikipedia.org/wiki/Ftp

[26] Full Duplex:http://en.wikipedia.org/wiki/Full_duplex

[27] GNU General Public License: http://en.wikipedia.org/wiki/GNU_GPL

[28] Hypertext Transfer Protocol: http://en.wikipedia.org/wiki/Http

[29] IP Fragmentation Attacks: http://en.wikipedia.org/wiki/IP_fragmentation_attacks

[30] Ipv4: http://en.wikipedia.org/wiki/Ipv4

[31] Open Source: http://en.wikipedia.org/wiki/Open_source

[32] PHP: http://en.wikipedia.org/wiki/Php

[33] Request for Comments: http://en.wikipedia.org/wiki/Request_for_comments

[34] Remote Procedure Call: http://en.wikipedia.org/wiki/Rpc

[35] Security Policy: http://en.wikipedia.org/wiki/Security_policy

[36] Server Message Block: http://en.wikipedia.org/wiki/Server_message_block

[37] Simple Mail Transfer Protocol: http://en.wikipedia.org/wiki/Smtp

[38] Slackware Snort Installation Guide, 2007 Jeffrey Denton: http://www.snort.org/docs/setup_guides/slackware-snort-0.2.txt

[39] Social Engineering: http://en.wikipedia.org/wiki/Social_engineering

[40] Sound-to-Noise Ratio: http://en.wikipedia.org/wiki/Signal-to-noise_ratio

[41] Secure Shell: http://en.wikipedia.org/wiki/Ssh

[42] Syslog: http://en.wikipedia.org/wiki/Syslog

[43] Transmission Control Protocol: http://en.wikipedia.org/wiki/Tcp

[44] Unix: http://en.wikipedia.org/wiki/Unix

[45] Virtual LAN: http://en.wikipedia.org/wiki/Vlan

[46] Computer Security Threats: http://www.caci.com/business/ia/threats.html

11. Γλωσσάρι.

Security Policy	Ορίζει το τι σημαίνει ασφάλεια για ένα σύστημα, οργανισμό ή άλλη οντότητα.
Firewall	Υλικό ή λογισμικό η παραμετροποίηση του οποίου επιτρέπει, απαγορεύει ή αναδρομολογεί δεδομένα μέσω ενός δικτύου υπολογιστών που αποτελείται από διαφορετικά επίπεδα εμπιστοσύνης.
Antivirus	Λογισμικό το οποίο προσπαθεί να αναγνωρίσει, περιορίσει και να εξαλείψει ιούς υπολογιστών και άλλο κακόβουλο λογισμικό.
Social Engineering	Είναι ένα σύνολο τεχνικών που χρησιμοποιούνται με σκοπό την χειραγώγηση των ανθρώπων ώστε να δώσουν απόρρητες πληροφορίες.
ARP – Address Resolution Protocol	Η μέθοδος που χρησιμοποιείται για την εύρεση της hardware διεύθυνσης ενός συστήματος όταν είναι γνωστή μόνο η διεύθυνση του επιπέδου δικτύου. Στο IP χρησιμοποιείται κυρίως για την μετάφραση της διεύθυνσης IP σε Ethernet MAC.
IP fragmentation	Η διαδικασία του χωρισμού ενός αυτοδύναμου πακέτου IP σε δύο ή περισσότερα πακέτα IP μικρότερου μεγέθους.
DoS attack – Denial of Service attack	Επίθεση που έχει σκοπό να καταστήσει μη διαθέσιμο ένα υπολογιστικό σύστημα προς τους νόμιμους χρήστες του.
IPv4	Είναι η τέταρτη επανάληψη του Internet Protocol (IP) και η πρώτη έκδοση που είναι ευρέως διαδεδομένη. Το IPv4 κυρίαρχο πρωτόκολλο στο επίπεδο δικτύου του Διαδικτύου.
TCP – Transmission Control Protocol	Είναι ένα από τα βασικά πρωτόκολλα της σουίτας πρωτοκόλλων Διαδικτύου. Το TCP παρέχει αξιόπιστη και με σειρά παράδοση των ροών δεδομένων, κάνοντας το κατάλληλο για εφαρμογές μεταφοράς αρχείων και ηλεκτρονικού ταχυδρομείου.
RFC – Request for Comments	Τα έγγραφα RFC είναι μια σειρά από υπηρεσιακά σημειώματα που αφορούν έρευνα, καινοτομίες και μεθοδολογίες που αφορούν τεχνολογίες Internet. Ορισμένες από αυτές τις προτάσεις υιοθετούνται ως Internet στάνταρ.
Sound-Noise-Ratio	
Open Source	Είναι μια σειρά αρχών και πρακτικών που προωθούν την πρόσβαση στην σχεδίαση και παραγωγή υλικών και γνώσης. Ο όρος χρησιμοποιείται συνήθως για τον κώδικα του λογισμικού που είναι διαθέσιμος στο κοινό με ελαφριές ή καθόλου απαγορεύσεις πνευματικής ιδιοκτησίας.
Unix	Είναι ένα λειτουργικό σύστημα υπολογιστών το οποίο αρχικά αναπτύχθηκε το 1969 από μια ομάδα εργαζομένων της AT&T στα Bell Labs.

GNU GPL	Είναι μια διαδεδομένη άδεια ελεύθερου λογισμικού σύμφωνα με την οποία το λογισμικό καθώς και προγράμματα που προκύπτουν από αυτό είτε με αλλαγές ή προσθήκες συνεχίζουν να είναι ελεύθερα.
Encapsulation	Είναι η διαδικασία κατά την οποία τα δεδομένα ενός πρωτοκόλλου ανώτερου επίπεδου ενσωματώνονται στο πρωτόκολλο ενός χαμηλότερου επιπέδου.
RPC – Remote Procedure Call	Είναι μια τεχνολογία που επιτρέπει σε ένα πρόγραμμα να εκτελέσει μια υπορουτίνα ή διαδικασία σε διαφορετικό διάστημα διεύθυνσης (συνήθως σε ένα διαφορετικό υπολογιστή) χωρίς ο προγραμματιστής να προγραμματίζει τις λεπτομέρειες τις απομακρυσμένης αλληλεπίδρασης.
SMTP – Simple Mail Transfer Protocol	Είναι ο de facto τρόπος για την αποστολή ηλεκτρονικού ταχυδρομείου μέσω του Διαδικτύου. Το πρωτόκολλο που χρησιμοποιείται σήμερα είναι γνωστός ως ESMTP.
SSH – Secure Shell	Είναι ένα δικτυακό πρωτόκολλο που επιτρέπει την ανταλλαγή δεδομένων ανάμεσα σε δύο υπολογιστές μέσω ασφαλούς καναλιού.
DCE/RCP – Distributed Computing Environment/Remote Procedure Calls	Είναι ένα σύστημα RPC που επιτρέπει στο λογισμικό να εκτελείτε σε διαφορετικά υπολογιστικά συστήματα σαν να εκτελούταν στο ίδιο.
SMB – Server Message Block	Είναι ένα δικτυακό πρωτόκολλο του επιπέδου εφαρμογής το οποίο χρησιμοποιείται για τον διαμοιρασμό αρχείων, εκτυπωτών και επικοινωνία μεταξύ κόμβων του δικτύου. Χρησιμοποιείται κυρίως από τα Miscrosoft Windows όπου είναι γνωστό ως “Microsoft Windows Network”.
DNS – Domain Name System	Στο Διαδίκτυο ο DNS συσχετίζει διάφορες πληροφορίες με τα ονόματα τομέων. Η σημαντικότερη λειτουργία του είναι η χρήση του ως “τηλεφωνικός κατάλογος” του Διαδικτύου, μεταφράζοντας τα ονόματα ιστοσελίδων (www.teicrete.gr) σε διευθύνσεις IP (193.92.11.2) που υποστηρίζει ο δικτυακός εξοπλισμός.
Syslog	Είναι το πρωτόκολλο που χρησιμοποιείται για την αποστολή αρχείων καταγραφής σε ένα δίκτυο IP. Ο όρος συχνά χρησιμοποιείται για το πρωτόκολλο καθώς και την εφαρμογή που στέλνει τα μυνήματα.
CSV – Comma-Separated Values	Είναι ένα είδος διαμόρφωσης αρχείου το οποίο περιέχει εγγραφές, όπου η κάθε εγγραφή αποτελεί μια μόνο γραμμή και διαχωρίζονται μεταξύ τους με κόμμα (,).
Aruba Networks	Εταιρία που προσφέρει κινητές δικτυακές υπηρεσίες.
VLAN – Virtual LAN	Είναι μια ομάδα υπολογιστών οι οποίοι επικοινωνούν μεταξύ τους σαν να ήταν συνδεδεμένοι στον ίδιο δίαυλο επικοινωνίας ανεξαρτήτως της φυσικής τους τοποθεσίας.
CRC Errors – Cyclic	Το CRC είναι ένα είδος συνάρτησης η οποία έχει ως όρισμα

Redundancy Check Errors	μια ροή δεδομένων οποιουδήποτε μήκους και παράγει ως έξοδο μια τιμή ορισμένου μήκους. Το CRC χρησιμοποιείται για την αναγνώριση τυχαίων αλλαγών κατά την διάρκεια μετάδοσης των δεδομένων.
Network TAP	Είναι μια συσκευή hardware η οποία παρέχει την δυνατότητα πρόσβασης στα δεδομένα που διακινούνται σε ένα δικτυακό περιβάλλον.
Full Duplex	Είναι κάθε σύστημα που επιτρέπει την ταυτόχρονη αμφίδρομη επικοινωνία.
ACL – Access Control List	Στην ασφάλεια υπολογιστών η ACL είναι μια λίστα “αδειών” που αντιστοιχούν σε ένα αντικείμενο. Η λίστα καθορίζει ποίος ή τι επιτρέπεται να έχει πρόσβαση στο αντικείμενο και ποίες λειτουργίες επιτρέπεται να συμβούν σε αυτό.
DMZ – Demilitarized Zone	Είναι ένα φυσικό ή λογικό υποδίκτυο το οποίο περιέχει τις εξωτερικές υπηρεσίες ενός οργανισμού σε ένα εξωτερικό μη ασφαλές δίκτυο όπως το Διαδίκτυο.
PHP – Hypertext Preprocessor	Είναι μια γλώσσα προγραμματισμού για την δημιουργία δυναμικών ιστοσελίδων.
HTTP – Hypertext Transfer Protocol	Είναι ένα πρωτόκολλο επικοινωνίας που χρησιμοποιείται για την μεταφορά πληροφοριών σε εσωτερικά δίκτυα ή το Διαδίκτυο. Αρχικός σκοπός του ήταν να παρέχει ένα τρόπο δημοσιοποίησης και επανάκτησης σελίδων υπερκειμένου.
FTP – File Transfer Protocol	Χρησιμοποιείται για την μεταφορά δεδομένων μεταξύ δυο υπολογιστικών συστημάτων μέσω του Διαδικτύου ή μέσω εσωτερικού δικτύου. Ειδικότερα χρησιμοποιείται για την ανταλλαγή αρχείων σε οποιοδήποτε δίκτυο υποστηρίζει το πρωτόκολλο TCP/IP.

ΠΑΡΑΡΤΗΜΑ Α

Κακόβουλες απειλές			
Κατηγορία	**Απειλή**	**Επίπεδο OSI**	**Ορισμός**
Κακόβουλο Λογισμικό – Malicious Software	Ιός - Virus	Εφαρμογής	Κακόβουλο λογισμικό που έχει την δυνατότητα να ενσωματώνεται σε άλλο λογισμικό.
	Σκουλήκι - Worm	Εφαρμογής Δικτύου	Κακόβουλο λογισμικό το οποίο αποτελεί ξεχωριστή εφαρμογή.
	Δούρειος Ίππος – Trojan Horse	Εφαρμογής	Ένα Worm το οποίο προσποιείται ότι είναι μια χρήσιμη εφαρμογή ή ένα ιός που έχεο ενσωματωθεί επίτηδες σε ένα πρόγραμμα πριν την διανομή του.
	Ωρολογιακή Βόμβα – Time Bomb	Εφαρμογής	Ένας ιός ή ένα σκουλήκι σχεδιασμένο να ενεργοποιείται μια συγκεκριμένη ημερομηνία/ώρα
	Λογική Βόμβα – Logic Bomb	Εφαρμογής	Ένας ιός ή ένα σκουλήκι σχεδιασμένο να ενεργοποιείται υπό συγκεκριμένες συνθήκες.
	Λαγός - Rabbit	Εφαρμογής Δικτύου	Ένα σκουλήκι σχεδιασμένο να αντιγράφει τον εαυτό του μέχρι να εξαντλήσει όλα τα αποθέματα του υπολογιστή.
	Βακτήριο - Bacterium	Εφαρμογής	Ένας ιός σχεδιασμένος να μπορεί να ενσωματωθεί από μόνος του στο λειτουργικό σύστημα και να εξαντλεί τα αποθέματα του υπολογιστή.
Εξαπάτηση - Spoofing	Εξαπάτηση - Spoofing	Δικτύου Διασύνδεσης Δεδομένων	Ένας υπολογιστής που ανήκει σε ένα δίκτυο παριστάνει την ταυτότητα ενός άλλος υπολογιστή, με σκοπό την απόκτηση πρόσβασης στους υπόλοιπους υπολογιστές του δικτύου.
	Μεταμφίεση - Masquerade		Σύνδεση σε ένα υπολογιστή προσποιούμενοι νόμιμη

			ταυτότητα χρήστη.
Ανίχνευσης - Scanning	Σειριακή ανίχνευση	Μεταφοράς Δικτύου	Σειριακή δοκιμή συνθηματικών μέχρι να επιτευχθεί πρόσβαση.
	Ανίχνευση με λεξικό	Εφαρμογής	Ανίχνευση με την χρήση λεξικού κοινά χρησιμοποιούμενων συνθηματικών μέχρι να επιτευχθεί πρόσβαση.
Κατασκοπείας – Snooping (Eavesdropping)	Ψηφιακή κατασκοπεία	Δικτύου	Ηλεκτρονική επιτήρηση ψηφιακών δικτύων για την εύρεση κωδικών ή άλλων χρήσιμων δεδομένων.
	Βλέποντας πάνω από τον ώμο.	Φυσικό	Απευθείας οπτική παρατήρηση του υπολογιστή με σκοπό την απόκτηση πρόσβασης.
Ψάξιμο σκουπιδιών – Scavenging.	Έλεγχος απορριμμάτων	Όλα	Απόκτηση πρόσβασης σε πεταμένα απορρίμματα με σκοπό την απόκτηση κωδικών ή άλλων δεδομένων.
	Αναζήτηση	Εφαρμογής Δικτύου	Συνήθως αυτοματοποιημένη ανίχνευση μεγάλων ποσοτήτων μη προστατευμένων δεδομένων με σκοπό την εύρεση στοιχείων για την απόκτηση πρόσβασης.
Ανεπιθύμητα μηνύματα - Spamming	Ανεπιθύμητα μηνύματα	Εφαρμογής Δικτύου	Υπερφόρτωση ενός συστήματος με εισερχόμενα μυνήματα ή άλλου είδους κίνηση με σκοπό την κατάρρευση του συστήματος.
Σήραγγα - Tunneling	Σήραγγα	Δικτύου	Οποιαδήποτε ψηφιακή επίθεση που προσπαθεί να παραβλέψει έναν μηχανισμό ασφαλείας χρησιμοποιώντας χαμηλού επιπέδου λειτουργίες.

Μη σκόπιμες απειλές			
Κατηγορία	**Απειλή**	**Επίπεδο OSI**	**Ορισμός**
Βλάβη	Βλάβη εξοπλισμού	Όλα	Το υλικό λειτουργεί με λανθασμένο, απροσδόκητο τρόπο.
	Βλάβη λογισμικού	Εφαρμογής	Η συμπεριφορά του λογισμικού έρχεται σε αντίθεση με την αναμενόμενη συμπεριφορά του.
Ανθρώπινο λάθος	Κερκόπορτα	Εφαρμογής	Πρόσβαση στο σύστημα για την φάση της ανάπτυξης που ακούσια παρέμεινε στην τελική διανομή.
	Λάθος χρήστη	Όλα	Ακούσια αλλαγή, χειρισμός ή καταστροφή προγραμμάτων, αρχείων δεδομένων ή hardware.

Φυσικές Απειλές			
Κατηγορία	**Απειλή**	**Επίπεδο OSI**	**Ορισμός**
Φυσικό περιβάλλον	Ζημιά από φωτιά	Δ/Υ	Φυσική καταστροφή του εξοπλισμού λόγω φωτιάς.
	Ζημιά από νερό	Δ/Υ	Φυσική καταστροφή του εξοπλισμού λόγου νερού.
	Απώλεια ρεύματος	Δ/Υ	Αποτυχία των υπολογιστικών συστημάτων ή του ζωτικού εξοπλισμού υποστήριξης λόγω έλλειψης ρεύματος.
	Κοινωνικές Αναταραχές Βανδαλισμός	Δ/Υ	Φυσική καταστροφή κατά την διάρκεια επιχειρήσεων εκτός πολέμου.
	Μάχη	Δ/Υ	Φυσική καταστροφή κατά την διάρκεια στρατιωτικών επιχειρήσεων.

ΠΑΡΑΡΤΗΜΑ Β

Στο συγκεκριμένο παράρτημα θα περιγραφεί η εγκατάσταση και η παραμετροποίηση των παρακάτω προγραμμάτων:

- Snort
- Barnyard
- BASE

Εγκατάσταση του Snort.

Αρχικά κατεβάζουμε την τελευταία έκδοση του Snort ελέγχουμε την αυθεντικότητα του και στην συνέχεια το αποσυμπιέζουμε:

fantasma@snortserver:~$ mkdir download
fantasma@snortserver:~$ cd download
fantasma@snortserver:~/download$ wget http://www.snort.org/dl/current/snort-2.8.0.tar.gz
fantasma@snortserver:~/download$ wget http://www.snort.org/dl/current/snort-2.8.0.tar.gz.md5
fantasma@snortserver:~/download$ md5sum -c snort-2.8.0.tar.gz.md5
snort-2.8.0.tar.gz: OK
fantasma@snortserver:~/download$ tar xvfz snort-2.8.0.tar.gz
fantasma@snortserver:~/download$ cd snort-2.8.0

Στην συνέχεια προχωράμε στην εγκατάστασή του:

fantasma@snortserver:~/download/snort-2.8.0$ ./configure –enable-dynamicplugins –enable-timestamps –enable-perfprofiling –with-mysql
fantasma@snortserver:~/download/snort-2.8.0$ make
fantasma@snortserver:~/download/snort-2.8.0$ sudo make install

Αφού τελειώσει η εγκατάσταση του Snort θα πρέπει να δημιουργήσουμε ένα λογαριασμό χρήστη τον οποίο θα χρησιμοποιεί το πρόγραμμα για την λειτουργία του. Ο λογαριασμός χρήστη που θα χρησιμοποιήσουμε θα είναι κλειδωμένος και το shell θα δείχνει στο /bin/false ώστε να είναι

αδύνατη η χρήση του για είσοδο στο σύστημα:

fantasma@snortserver:~/download/snort-2.8.0$ sudo groupadd snort
fantasma@snortserver:~/download/snort-2.8.0$ sudo useradd -g snort snort -s /bin/false
fantasma@snortserver:~/download/snort-2.8.0$ sudo passwd -S snort

Στη συνέχεια θα πρέπει να δημιουργήσουμε τους καταλόγους που θα χρησιμοποιήσει το Snort για την αποθήκευση των κανόνων και των αρχείων καταγραφής, καθώς επίσης και θα ρυθμίσουμε τα δικαιώματα πρόσβασης στον κατάλογο /var/log/snort :

fantasma@snortserver:~/download/snort-2.8.0$ sudo mkdir -p /etc/snort/rules
fantasma@snortserver:~/download/snort-2.8.0$ sudo mkdir -p /var/log/snort/archive
fantasma@snortserver:~/download/snort-2.8.0$ sudo chown -R snort.snort /var/log/snort

Για να μπορέσουμε να κατεβάσουμε από την ιστοσελίδα του Snort τους τελευταίους διαθέσιμους κανόνες πρέπει να εγγραφούμε (δωρεάν) ως χρήστες του προγράμματος. Στην συνέχεια μεταγλωττίζουμε τους κανόνες που κατεβάσαμε και τους αντιγράφουμε στις τοποθεσίες που θα ορίσουμε το Snort να τις διαβάζει:

fantasma@snortserver:~/download/snort-2.8.0$ md5sum -c snortrules-snapshot-CURRENT.tar.gz.md5
fantasma@snortserver:~/download/snort-2.8.0$ tar xvfz snortrules-snapshot-CURRENT.tar.gz
fantasma@snortserver:~/download/snort-2.8.0$ cd so_rules
fantasma@snortserver:~/download/snort-2.8.0/so_rules$ make
fantasma@snortserver:~/download/snort-2.8.0/so_rules$ cat *.rules > so.rules
fantasma@snortserver:~/download/snort-2.8.0/so_rules$ sudo cp so.rules /etc/snort/rules
fantasma@snortserver:~/download/snort-2.8.0/so_rules$ sudo mkdir /usr/local/lib/snort_dynamicrule
fantasma@snortserver:~/download/snort-2.8.0/so_rules$ sudo cp *.so /usr/local/lib/snort_dynamicrule

Παραμετροποίηση του Snort ώστε να χρησιμοποιεί τους κανόνες των δυναμικών plug-ins:

fantasma@snortserver:~/download/snort-2.8.0/so_rules$ sudo vi /etc/snort/snort.conf

```
dynamicdetection directory /usr/local/lib/snort_dynamicrule/
include $RULE_PATH/so.rules
```

Αντιγράφουμε τους υπόλοιπους κανόνες στον κατάλογο παραμετροποίησης του Snort:

fantasma@snortserver:~/download/snort-2.8.0/so_rules$ cd ../rules
fantasma@snortserver:~/download/snort-2.8.0/rules$ sudo cp * /etc/snort/rules

Στη συνέχεια, δημιουργούμε το αρχείο εκκίνησης του:

fantasma@snortserver:~/download/snort-2.8.0/rules$ sudo vi /etc/init.d/snort

```
#!/bin/sh
#
# Start/Stop/Restart Snort NIDS
#

# Specify network interface
INTERFACE="eth1"
CONF="/etc/snort/snort.conf"

snort_start() {
if ! /sbin/ifconfig $INTERFACE | grep "RUNNING" 1> /dev/null; then
                echo "Bringing up interface $INTERFACE..."
                /sbin/ifconfig $INTERFACE up -arp
                /usr/bin/touch /var/run/snort.$INTERFACE
fi
echo "Starting Snort..."
/usr/local/bin/snort -u snort -g snort -i $INTERFACE -c $CONF \
        -D -F /etc/snort/excludes.conf
}

snort_stop() {
echo "Stopping Snort..."
/bin/killall snort
```

```
if [ -e /var/run/snort.$INTERFACE ]; then
        echo "Shutting down interface $INTERFACE..."
        /sbin/ifconfig $INTERFACE down
        /usr/bin/rm -f /var/run/snort.$INTERFACE
fi
}

snort_restart() {
snort_stop
/usr/bin/sleep 2
snort_start
}

case "$1" in
'start')
snort_start
;;
'stop')
snort_stop
;;
'restart')
snort_restart
;;
*)
echo "usage $0 start|stop|restart"
esac
```

Παραμετροποίηση του Snort.

Μετά την ολοκλήρωση της εγκατάστασης του Snort επόμενο βήμα είναι η παραμετροποίηση του, όπου θα ορίσουμε την συμπεριφορά ελέγχου του δικτύου, τις τοποθεσίες των αναγκαίων αρχείων για την σωστή λειτουργία του και θα δώσουμε κάποιες πληροφορίες για το δίκτυο μας ώστε να γνωρίζει το Snort την τοπολογία του και να παρέχει σωστότερα αποτελέσματα. Αρχικά ορίζουμε τις τοποθεσίες των αρχείων δίνοντας την πλήρη διαδρομή:

fantasma@snortserver:~$ sudo vi /etc/snort/snort.conf

var RULE_PATH /etc/snort/rules

include /etc/snort/classification.config

include /etc/snort/reference.conf

include /etc/snort/threshold.conf

Θα πρέπει να ρυθμίσουμε την μεταβλητή HOME_NET στις αντίστοιχες διευθύνσεις δικτύου που αποτελούν το δίκτυο μας. Η αρχική της τιμή είναι “any”.

var HOME_NET [192.168.1.0/24]

Ενεργοποιούμε την δυνατότητα ειδοποίησης για υπερμεγέθη πακέτα:

config enable_decode_oversized_alerts

Παραμετροποιούμε την μηχανή ανίχνευσης (ως προς τον τύπο) καθώς και την σειρά με την οποία γίνεται η επεξεργασία των κανόνων:

config detection: search-method ac-bnfa

config order: pass alert log activation

Ενεργοποιούμε την δυνατότητα ελέγχου επιδόσεων ώστε να μας ενημερώνει για κακογραμμένους κανόνες οι οποίοι καταναλώνουν πολύ χρόνο επεξεργασίας μειώνοντας έτσι την απόδοση του Snort:

config profile_rules: print 10, sort total_ticks

Ενεργοποιούμε τον προεπεξεργαστή frag3, ο οποίος είναι υπεύθυνος για τον τρόπο επανένωσης των κατετμημένων πακέτων σύμφωνα με το λειτουργικό σύστημα που εκτελεί ο κάθε διακομιστής. Στο πειραματικό μας δίκτυο έχουμε ένα σύστημα Microsoft Windows και ένα Linux για τα οποία θα πρέπει να χρησιμοποιήσουμε διαφορετικές ρυθμίσεις:

preprocessor frag3_global: max_frags 65536, prealloc_frags 65536

preprocessor frag3_engine: policy linux \

bind_to [192.168.1.1/32] \

```
                                    detect_anomalies
preprocessor frag3_engine: policy windows \
                                    bind_to [192.168.1.2/32] \
                                    detecy anomalies
```

Στην συνέχεια θα πρέπει να ενεργοποιήσουμε και να παραμετροποιήσουμε τον προεπεξεργαστή stream5, ο οποίος εξαρτάται για την σωστή λειτουργία του επίσης από το λειτουργικό σύστημα του διακομιστή:

```
preprocessor stream5_global: max_tcp 8192, track_tcp yes, track_udp yes
preprocessor stream5_tcp: policy linux, bind_to [192.168.1.1/32]
preprocessor stream5_tcp: policy windows, bind_to [192.168.1.2/32]
preprocessor stream5_udp:
```

Στη συνέχεια παραμετροποιούμε τον preprocessor των στατιστικών:

```
preprocessor perfmonitor: time 300 file /var/log/snort/snort.stats pktcnt 10000
```

Τέλος ορίζουμε την λειτουργία των προεπεξεργαστών SSH και DCE/RPC:

```
preprocessor ssh: server_ports {22} \
        max_client_bytes 19600 \
        max_encrypted_packets 20 \
        disable_protomismatch \
        disable_paysize

preprocessor dcerpc: \
        ports smb { 139 445 } ports dcerpc { 135 } \
        max_frag_size 3000 \
        memcap 100000 \
        alert_memcap
```

Τέλος το plug-in εξόδου του Snort που θα χρησιμοποιήσουμε για να συνεργαστεί με το πρόγραμμα Barnyard:

output log_unified: filename snort.log, limit 512

Εγκατάσταση του BASE.

Η εγκατάσταση του BASE έχει κάποιες απαιτήσεις σε προγράμματα της PHP τα οποία πρέπει να εγκατασταθούν για να λειτουργήσει. Η εγκατάσταση τους γίνεται με την εντολή pear:

fantasma@snortserver:~$ sudo pear install –alldeps Image_Graph-alpha Image_Canvas-alpha Image_Color Numbers_Roman

Στη συνέχεια κατεβάζουμε το ADODB, το οποίο είναι μια αφαιρετική βιβλιοθήκη για βάσεις δεδομένων, καθώς και το BASE:

fantasma@snortserver:~/download$ wget http://easynews.dl.sourceforge.net/sourceforge/adodb/adodb480.tgz

fantasma@snortserver:~/download$ wget http://easynews.dl.sourceforge.net/sourceforge/secureideas/base-1.3.8.tar.gz

Αποσυμπιέζουμε το ADODB και διορθώνουμε τα δικαιώματα χρήσης των αρχείων του:

fantasma@snortserver:~/download$ tar xvf adodb480.tgz

fantasma@snortserver:~/download$ sudo mv adodb /var/www

fantasma@snortserver:~/download$ sudo chmod -R o-w /var/www/adodb

Στην συνέχεια αποσυμπιέζουμε το BASE και το μετακινούμε στον κατάλογο που χρησιμοποιεί ο διακομιστής ιστοσελίδων, στην περίπτωση μας /var/www/apache2-default, και ορίζουμε τις απαραίτητες παραμέτρους για την λειτουργία του:

fantasma@snortserver:~/download$ tar xvfz base-1.3.8.tar.gz

fantasma@snortserver:~/download$ sudo mv base-1.3.8 /var/www/apache2-default/base

fantasma@snortserver:~/download$ cd /var/www/apache2-default/base

fantasma@snortserver:~/var/www/apache2-default/base$ sudo cp base_conf.php.dist base_conf.php

fantasma@snortserver:~/var/www/apache2-default/base$ sudo vi base_conf.php

```
$BASE_url = '/';
$DBlib_path = '/var/www/adodb';
$DBtype = 'mysql';
$alert_dbname = 'snort';
$alert_host = 'localhost';
$alert_port = '';
$alert_user = 'snort';
$alert_password = 'sn0rtp2sswd';
$show_rows = 90;
$show_expanded_query = 1;
$colored_alerts = 1;
```

Ορισμένοι κανόνες του Snort παρέχονται μαζί με ένα αρχείο βοήθειας το οποίο περιγράφει την επίθεση που ανιχνεύει. Το BASE έχει την δυνατότητα να εμφανίζει αυτά τα αρχεία αρκεί να αντιγραφούν στον κατάλογο του.

```
fantasma@snortserver:~/var/www/apache2-default/base$ cd
fantasma@snortserver:~$ cd download
fantasma@snortserver:~/download$ cd snort-2.8.0/doc
fantasma@snortserver:~/download/snort-2.8.0/doc$ sudo cp -r signatures
/var/www/apache2-default/base
```

Δημιουργία πινάκων στη βάση δεδομένων.

Για να είναι εφικτή η λειτουργία του BASE πρέπει να δημιουργήσουμε τους πίνακες που θα χρησιμοποιεί η βάση δεδομένων για την αποθήκευση των ειδοποιήσεων του Snort, καθώς και να δημιουργήσουμε τους χρήστες που θα έχουν πρόσβαση σε αυτήν:

fantasma@snortserver:~/download/snort-2.8.0$ mysql -uroot -p < schemas/create_mysql snort

fantasma@snortserver:~/download/snort-2.8.0$ mysql -uroot -p

mysql> grant create,insert,select,delete,update on snort.* to snort@localhost identified by 'sn0rtp2sswd';

mysql> flush privileges;

mysql> exit;

Στη συνέχεια πρέπει το BASE να δημιουργήσει στη βάση δεδομένων snort κάποιους δικούς του πίνακες για τις εσωτερικές του λειτουργίες. Για να το επιτύχουμε αυτό ανοίγουμε ένα φυλλομετρητή ιστοσελίδων (web browser) εισάγουμε την διεύθυνση του BASE και επιλέγουμε τον σύνδεσμο “Setup page”. Στην συνέχεια ακολουθούμε τις οδηγίες που μας εμφανίζει το πρόγραμμα για την δημιουργία των πινάκων.

Εγκατάσταση του Barnyard.

Το Barnyard χρησιμοποιείται για τον διαχωρισμό του Snort από την βάση δεδομένων. Μιας και το Snort χρησιμοποιεί μια διαδικασία για την επεξεργασία των πακέτων και την αποθήκευση τους είναι προτιμότερο να χρησιμοποιείται ένα εξωτερικό πρόγραμμα που να αναλαμβάνει την αποθήκευση στην βάση δεδομένων, δεδομένου ότι αυτή η διαδικασία είναι κάπως αργή. Στην συνέχεια θα δούμε την εγκατάσταση και ρύθμιση του Barnyard στο εργαστηριακό μας περιβάλλον. Αρχικά κατεβάζουμε την τελευταία έκδοση του από τον δικτυακό τόπο του Snort και πιστοποιούμε την αυθεντικότητα του:

fantasma@snortserver:~/download/snort-2.8.0$ cd ..

fantasma@snortserver:~/download$ wget http://www.snort.org/dl/barnyard/barnyard-0.2.0.tar.gz

fantasma@snortserver:~/download$ wget http://www.snort.org/dl/barnyard/barnyard-0.2.0.tar.gz.md5

fantasma@snortserver:~/download$ md5sum barnyard-0.2.0.tar.gz

be3283028cf414b52b220308ceb411e9 barnyard-0.2.0.tar.gz

fantasma@snortserver:~/download$ cat barnyard-0.2.0.tar.gz.md5

md5 : be3283028cf414b52b220308ceb411e9 barnyard-0.2.0.tar.gz

sha1 : 4adfcabb2702def5a9a6c68cbde1b90a70f7e67a barnyard-0.2.0.tar.gz

fantasma@snortserver:~/download$ tar xvfz barnyard-0.2.0.tar.gz

fantasma@snortserver:~/download$ cd barnyard-0.2.0

fantasma@snortserver:~/download/barnyard-0.2.0$ ./configure –enable-mysql

fantasma@snortserver:~/download/barnyard-0.2.0$ make

fantasma@snortserver:~/download/barnyard-0.2.0$ sudo make install

Το barnyard κατά την εγκατάσταση του δεν δημιουργεί μια απαραίτητη εγγραφή στον πίνακα sensor στην βάση δεδομένων του Snort. Η έλλειψη αυτής της εγγραφής δεν επιτρέπει στο BASE να ανανεώσει τους πίνακες του με αποτέλεσμα να μην εμφανίζονται οι ειδοποιήσεις. Για να λυθεί το παραπάνω πρόβλημα δημιουργούμε από μόνοι μας την εγγραφή:

fantasma@snortserver:~/download/barnyard-0.2.0$ mysql -uroot -p

mysql> use snort;

mysql> insert into snort.sensor (sid,hostname,interface,filter,detail,encoding,last_cid) values

(1,”localhost”,”eth0”,””,1,0,0);

mysql> exit;

Για την παρακολούθηση των ειδοποιήσεων που έχουν επιτυχώς αποθηκευτεί στην βάση δεδομένων, το Barnyard χρησιμοποιεί ένα αρχείο το οποίο πρέπει να το δημιουργήσουμε:

fantasma@snortserver:~/download/barnyard-0.2.0$ sudo vi /var/log/snort/barnyard.waldo

/var/log/snort

snort.log

0

0

Στην συνέχεια ρυθμίζουμε τις παραμέτρους λειτουργία του Barnyard μεταβάλλοντας το αρχείο barnyard.conf:

fantasma@snortserver:~/download/barnyard-0.2.0$ sudo vi /etc/snort/barnyard.conf

config daemon

config localtime

config hostname: localhost

config interface: eth0

config sid-msg-map: /etc/snort/sid-msg.map

config gen-msg-map: /etc/snort/gen-msg.map

config class-file: /etc/snort/classification.config

output log_acid_db: mysql, sensor_id 1, database snort, server localhost, user snort, detail full, password sn0rtp2sswd

Τέλος δημιουργούμε το αρχείο εκκίνησης του προγράμματος και αλλάζουμε τα δικαιώματα χρήσης ώστε να γίνει εκτελέσιμο:

fantasma@snortserver:~/download/barnyard-0.2.0$ sudo vi /etc/init.d/barnyard

#!/bin/sh

#

Start/Stop/Restart Barnyard

```
#

 CONF="/etc/snort/barnyard.conf"

barnyard_start() {
        echo "Starting Barnyard..."
        /usr/local/bin/barnyard -v -c $CONF \
        -d /var/log/snort \
        -f snort.log \
        -w /var/log/snort/barnyard.waldo \
        -a /var/log/snort/archive \
        -X /var/run/barnyard.pid
}

barnyard_stop() {
        echo "Stopping Barnyard..."
        /bin/killall barnyard
}

barnyard_restart() {
        barnyard_stop
        /usr/bin/sleep 2
        barnyard_start
}

case "$1" in
'start')
        barnyard_start
;;
'stop')
        barnyard_stop
;;
'restart')
        barnyard_restart
```

```
;;
*)
echo "usage $0 start|stop|restart"

esac
```

fantasma@snortserver:~/download/barnyard-0.2.0$ chmod 0700 /etc/init.d/barnyard

Για να εκκινήσουμε το Snort και το Barnyard αρκεί να εκτελέσουμε τα αρχεία εκκίνησης που έχουμε δημιουργήσει για το κάθε πρόγραμμα:

fantasma@snortserver:~/download/barnyard-0.2.0$ sudo /etc/init.d/snort start
fantasma@snortserver:~/download/barnyard-0.2.0$ sudo /etc/init.d/barnyard start