Πανεπιστήμιο Πειραιώς

Σχολή Βιομηχανικών Σπουδών

Τμήμα Ψηφιακών Συστημάτων

«Διδακτική της Τεχνολογίας και Ψηφιακά Συστήματα»

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΘΕΜΑ

Honeypots & Honeyd

Υλοποίηση δικτυακής υπηρεσίας για το honeyd

Διπλωματική Εργασία : *Τσιρεπλή Ισμήνη*

Μ.Ε. : *07075*

Υπεύθυνος Καθηγητής : *Δρ. Χρήστος Ξενάκης*

Αθήνα,

Σεπτέμβριος 2011

Πανεπιστήμιο Πειραιώς

Αθήνα, 2011

Η τριμελής επιτροπή διδασκόντων που επικυρώνει τη διπλωματική εργασία της Τσιρεπλή Ισμήνη είναι οι :

.....	.....	.....
Χρήστος Ξενάκης	***Σωκράτης Κ. Κάτσικας***	***Κων. Λαμπρινουδάκης***
Επίκουρος Καθηγητής	Καθηγητής	Επίκουρος Καθηγητής
Πανεπιστημίου Πειραιώς	Πανεπιστημίου Πειραιώς	Πανεπιστημίου Πειραιώς

Περίληψη

Άμεσα συνδεδεμένα με το διαδίκτυο είναι τα υπολογιστικά συστήματα όπου διαρκώς δέχονταν και δέχονται επιθέσεις από εισβολείς και αυτοματοποιημένες επιθέσεις. Μέχρι τα τέλη της δεκαετίας του ’90 υπήρχαν πολύ λίγες πληροφορίες για τις δράσεις και τις τεχνικές των επιτιθέμενων στα πληροφοριακά συστήματα και αυτό γιατί όλες οι προσπάθειες ήτανε επικεντρωμένες κυρίως στο να εμποδίσουν τους εισβολείς να εισβάλουν στο εσωτερικό του συστήματος. Η κατάσταση αυτή άρχισε να αλλάζει λίγο πριν από το 2000 και με την προσφορά του «The Honeynet Project, ενός μη κερδοσκοπικού οργανισμού από εθελοντές, ο οποίος ανέπτυξε εργαλεία ανοικτού κώδικα ασφαλείας για τη βελτίωση της ασφάλειας στο Διαδίκτυο. Ένα από τα πιο πρόσφατα εργαλεία για την αντιμετώπιση των δικτυακών επιθέσεων από κακόβουλους χρήστες και αυτοματοποιημένες επιθέσεις είναι τα **honeypots και τα honeynets.**

Παράλληλα η συνεχής εξάπλωση των worms και των κακόβουλων χρηστών είχε ως αποτέλεσμα την αύξηση του όγκου της κακόβουλης διαδικτυακής κίνησης. Άμεσος στόχος λοιπόν ήτανε η επίτευξη συγκέντρωσης και ανάλυσης αυτής της κίνησης, η οποία θα βοηθούσε στο να εντοπίσουμε τις ενέργειες των κακόβουλων εισβολέων, να ανακαλύψουμε καινούργια είδη επιθέσεων, να δούμε ποιοί τύποι επιθέσεων και ποια worms είναι ενεργά, αλλά και να μας ευαισθητοποιήσει σε θέματα ασφάλειας δικτύων. Ένα τέτοιο εργαλείο που προσομοιώνει σχεδόν οποιοδήποτε γνωστό λειτουργικό σύστημα στο επίπεδο του δικτύου είναι το το **honeyd**. Το honeyd προσομοιώνει τη στοίβα δικτύου, network stack, από τα περισσότερα λειτουργικά συστήματα και έτσι καταφέρνει να δημιουργήσει virtual hosts ,εικονικά συστήματα, τα οποία φαίνονται σαν φυσικά συστήματα.

Στα πλαίσια της διπλωματικής αυτής θα εγκατασταθεί ένα honeypot το οποίο θα βασίζεται στο λογισμικό honeyd. Έπειτα, στο κύριος μέρος της διπλωματικής αυτής θα υλοποιηθεί μια καινούργια δικτυακή υπηρεσία για το honeyd. Στη συνέχεια θα πραγματοποιηθούν διάφορες επιθέσεις στην υπηρεσία αυτή στις οποίες το honeyd θα πρέπει να τις εντοπίζει.

Στο πρώτο κεφάλαιο παρουσιάζονται οι βασικές πληροφορίες σχετικά με τα δίκτυα, την αρχιτεκτονική των δικτύων την ασφάλεια , τις επιθέσεις και την προστασία στα δίκτυα Υπολογιστών. Στο δεύτερο κεφάλαιο δίνονται οι βασικοί ορισμοί, οι κατηγοριοποιήσεις και οι δομές των Honeypot, Honeynet. Στο τρίτο κεφάλαιο αναλύεται ο ορισμός του Honeyd και υλοποιείται μια καινούργια δικτυακή υπηρεσία για το Honeyd. Τέλος στο τέταρτο κεφάλαιο παρουσιάζονται τα συνολικά συμπεράσματα της εργασίας.

Λέξεις κλειδιά: Δίκτυο, IP, TCP/IP, Μάσκα Υποδικτύου, Ασφάλεια, Επιθέσεις, Antivirus, Firewall, IDS, Honeypot, Honeynet, honeytokens, Honeyd.

Abstract

The computational systems which are directly connected to the Internet were always and still are vulnerable to intruders and automated attacks. Little information was available as for the actions taken and techniques used by the attackers of the information systems till the end of ‘90’s, due to the fact that all efforts focused on preventing the attackers from intruding the system. This situation started changing since 2000 when ‘’The Honeynet Project’’, a non-profit organization made of volunteers, was founded and developed open-code security tools aiming to improve the Internet security. Honeypots and honeynets are two of the most recent tools used for preventing network attacks by malicious users and automated attacks.

The concurrent and continuous spread of worms and malicious users resulted in increasing the malicious network traffic. The concentration and analysis of this traffic was an urgent target and would help detect the actions of the malicious intruders, discover new attacks, detect which types of attack and which worms are active, but also make us sensitive to network security issues. Honeyd is a tool which simulates almost every known operating system on the network level. The honeyd simulates the network stack from most of the operating systems and in this way manages to create virtual hosts which appear as natural systems.

In the framework of this diploma thesis, a honeypot based on the honeyd software will be installed. In the main body of this thesis a new network service for honeyd will be implemented. Several attacks will be performed to the service afterwards which should normally be detected by honeyd.

The first chapter presents the basic information about the networks and the architecture of networks, the concerning attacks and computer networks security. The second chapter provides the basic definitions, categorizations and structures of Honeypot, and Honeynet. The third chapter analyze the definition of Honeyd and a new network service for Honeyd is implemented. Finally, the forth chapter presents the overall conclusions reached.

Key words: Network, IP, TCP/IP, Subnet Mask, security, attacks, antivirus, firewall, IDS, Honeypot, Honeynet, honeytokens, Honeyd.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου Δρ. Χρήστο Ξενάκη για την ευκαιρία που μου έδωσε να ασχοληθώ με το συγκεκριμένο θέμα, την εμπιστοσύνη που μου έδειξε, καθώς επίσης για την υποστήριξή του σε κάθε ζήτημα που αφορούσε τη διπλωματική μου εργασία.

Επίσης θα ήθελα να ευχαριστήσω τον Πάνο Χριστόφορο, υποψήφιο λέκτορα του Τμήματος Πληροφορικής και Τηλεπικοινωνιών του Καποδιστριακού Πανεπιστημίου Αθηνών, για την πολύτιμη συνεργασία του και καθοδήγηση του στις κάθε είδους δυσκολίες που συνάντησα καθ' όλη την διάρκεια αυτής της εργασίας.

Τέλος, ένα μεγάλο «ευχαριστώ» στα αγαπημένα μου πρόσωπα, τους φίλους μου και ιδιαίτερα στους γονείς μου, που αποδέχθηκαν όλες τις επιλογές μου και μου παρείχαν την αμέριστη συμπαράστασή τους, και υποστήριξή τους όλο αυτό το διάστημα ακόμη και τις στιγμές όπου τα πράγματα δεν πήγαιναν καλά.

Περιεχόμενα

Κατάλογος Εικόνων

Κατάλογος Πινάκων

1 Κεφάλαιο 1 - Εισαγωγή

Δομή του Κεφαλαίου

Το κεφάλαιο αυτό παρουσιάζει βασικές πληροφορίες σχετικά με τον ορισμό και την αρχιτεκτονική των δικτύων υπολογιστών καθώς επίσης πληροφορίες σχετικά με τις επιθέσεις που μπορούν να δεχτούν και τον τρόπο προστασία τους.

1.1 Εισαγωγή

1.2 Ορισμός Δικτύων Υπολογιστών

Ως δίκτυο υπολογιστών ορίζεται ο τρόπος με τον οποίο δύο ή περισσότεροι αυτόνομοι και ανεξάρτητοι υπολογιστές ή περιφερειακές συσκευές μπορούν να συνδεθούν ώστε να ανταλλάσσουν μεταξύ τους πληροφορίες.

Τα δίκτυα δημιουργήθηκαν για να εξυπηρετήσουν τις ανάγκες που προέκυψαν από την εξάπλωση της χρήσης των υπολογιστών. Βασικός σκοπός της ύπαρξής τους, είναι ο διαμερισμός των πόρων του συστήματος και η ανταλλαγή πληροφοριών κάθε μορφής (προγράμματα, αρχεία, δεδομένα). Πόροι του συστήματος μπορεί να είναι είτε υλικό (hardware), π.χ. υπολογιστές, εκτυπωτές, plotters, σκληροί δίσκοι είτε λογισμικό (software), π.χ. δεδομένα, προγράμματα εφαρμογών, υπηρεσίες. Τα προγράμματα, τα δεδομένα και οι συσκευές (σκληροί δίσκοι, εκτυπωτές, κλπ) είναι διαθέσιμα σε οποιονδήποτε είναι συνδεδεμένος στο δίκτυο, ανεξάρτητα από τη φυσική του θέση. Με τον τρόπο αυτό επιτυγχάνεται εξοικονόμηση χρημάτων, αύξηση της απόδοσης του συστήματος, κεντρικός έλεγχος και εύκολη επεκτασιμότητα. Συνοπτικά λοιπόν οι δυνατότητες των δικτύων είναι οι εξής:

- ☞ Διαμοιρασμός ψηφιακών πόρων του συστήματος, δηλαδή προγραμμάτων, φακέλων, αρχείων κ.λπ. Αυτό πρακτικά σημαίνει ότι συγκροτείται ένας εικονικός κοινόχρηστος χώρος, όπου όλοι οι χρήστες, ανάλογα και με τα προνόμια - δικαιώματα που τους έχουν δοθεί από το διαχειριστή του δικτύου, έχουν πρόσβαση από τον υπολογιστή τους και μπορούν να χρησιμοποιήσουν τα ίδια αρχεία, τους ίδιους φακέλους και τις ίδιες εφαρμογές, ανεξάρτητα από το ποιος έχει δημιουργήσει το αρχείο ή σε ποιον υπολογιστή έχει εγκατασταθεί η εφαρμογή.
- ☞ Κοινή χρήση περιφερειακών συσκευών. Αυτό σημαίνει ότι τα μέλη του δικτύου μπορούν να χρησιμοποιήσουν από κοινού τις ίδιες περιφερειακές συσκευές. Έτσι, αν για παράδειγμα ένας χρήστης έχει τέσσερις υπολογιστές, δεν χρειάζεται να έχει και τέσσερις εκτυπωτές και τέσσερις σαρωτές. Αρκεί μία συσκευή από το κάθε είδος, η οποία θα χρησιμοποιείται από όλους τους υπολογιστές. Η δυνατότητα αυτή μεταφράζεται ξεκάθαρα σε εξοικονόμηση κεφαλαίων αλλά και χώρου.

- ☞ Διαμοιρασμός μιας σύνδεσης Internet σε όλους τους υπολογιστές του δικτύου. Αυτό σημαίνει ότι η ύπαρξη μιας και μοναδικής σύνδεσης με το διαδίκτυο αρκεί για να παράσχει πρόσβαση σε όλους τους υπολογιστές του τοπικού δικτύου. Η ταχύτητα σύνδεσης του κάθε υπολογιστή με το Internet εξαρτάται από το είδος της σύνδεσης (PSTN, ISDN, ADSL κ.λπ.) καθώς και από τον αριθμό των PC που βρίσκονται συνδεδεμένα στο διαδίκτυο την ίδια στιγμή.
- ☞ Αξιοποίηση υπολογιστών περιορισμένων δυνατοτήτων ή παλαιότερης τεχνολογίας. Αυτό σημαίνει ότι υπολογιστές που ως αυτόνομες μονάδες δεν μπορούσαν να χρησιμεύσουν σε κάτι αξιόλογο (λ.χ. επειδή δεν διέθεταν συσκευή ανάγνωσης CD-ROM ή επειδή ο σκληρός τους δίσκος είχε περιορισμένο αποθηκευτικό χώρο), μπορούν τώρα να ενταχθούν σε ένα μικρό δίκτυο και να παίξουν κάποιο ρόλο μέσα σ' αυτό.

1.3 Αρχιτεκτονική και Είδη Δικτύων Υπολογιστών.

Η αρχιτεκτονική των δικτύων καθορίζει τον τρόπο με τον οποίο οι υπολογιστές και οι λοιπές συσκευές συνδέονται μεταξύ τους για να σχηματίσουν ένα σύστημα επικοινωνίας που θα επιτρέπει στους χρήστες να διαμοιράζουν πληροφορίες και συσκευές του δικτύου. Σε ένα δίκτυο δεδομένων περιλαμβάνονται:

- ☞ Τερματικοί Κόμβοι, οι οποίοι ελέγχουν τους πόρους του δικτύου (λογισμικό και υλικό).
- ☞ Υποδίκτυα, φυσικά μέσα μετάδοσης, πρωτόκολλα επικοινωνίας, τοπολογία, τερματικοί κόμβοι, πόροι που μπορούν να διαφέρουν πολύ ανά υποδίκτυο.
- ☞ Συσκευές Διασύνδεσης, οι οποίες διασυνδέουν τα ετερογενή υποδίκτυα έτσι ώστε να εξασφαλίζεται η επικοινωνία τερματικών κόμβων που βρίσκονται σε διαφορετικά υποδίκτυα.

Τα είδη των δικτύων ποικίλουν και :

- Ανάλογα με την γεωγραφική ανάπτυξή τους διακρίνονται σε :
  - ☞ **Δίκτυα ευρείας περιοχής (Wide Area Networks,WAN),** που καλύπτουν αποστάσεις μερικών χιλιομέτρων (συνήθως άνω των 5 km) στην ίδια πόλη ή μέχρι χιλιάδων χιλιομέτρων σε διαφορετικές πόλεις - κράτη - ηπείρους.

Αποτελούνται από υπολογιστές, τηλεπικοινωνιακές συσκευές και γραμμές. Παραδείγματα τέτοιων δικτύων είναι τα δίκτυα των αεροπορικών εταιρειών, τα τραπεζικά δίκτυα, τα δημόσια δίκτυα δεδομένων κλπ.

- ☞ **Δίκτυα μικρών αποστάσεων ή τοπικά δίκτυα (Local Area Networks, LAN**) που καλύπτουν μικρές αποστάσεις (μερικών εκατοντάδων μέτρων ή λίγων χιλιομέτρων) και περιορίζονται στα πλαίσια μιας επιχείρησης. Ο διαχωρισμός τους από τα δίκτυα ευρείας περιοχής οφείλεται στο ότι χρησιμοποιούν διαφορετικές τεχνικές λειτουργίας. Τα τοπικά δίκτυα έχουν μικρό κόστος ανά χρήστη, μεγάλη ταχύτητα μεταφοράς πληροφοριών, επεκτασιμότητα, βελτιστοποίηση της χρήσης των μηχανημάτων, υψηλό επίπεδο παρερχομένων υπηρεσιών στους χρήστες του δικτύου και συμβατότητα με συσκευές κατασκευασμένες με συγκεκριμένα πρότυπα.
- ☞ **Αστικά Δίκτυα (Metropolitan Area Networks, MAN),** που καλύπτουν δίκτυα που δεν ξεπερνούν τα σύνορα μιας πόλης. Είναι ταχύτερα από τα τοπικά δίκτυα και μπορούν να μεταδώσουν εικόνα, φωνή και δεδομένα αποδοτικότερα.

- Ανάλογα τον τηλεπικοινωνιακό φορέα εξυπηρέτησης διακρίνονται σε :

  - ☞ **Ιδιωτικά δίκτυα (Private Networks).** Ανήκουν εξ ολοκλήρου σε ιδιωτικούς οργανισμούς και χρησιμοποιούν αποκλειστικές γραμμές επικοινωνίας δημόσιων τηλεπικοινωνιακών φορέων (leased lines) χωρίς να τις μοιράζονται με άλλους χρήστες ή ιδιόκτητες γραμμές επικοινωνίας.
  - ☞ **Δημόσια δίκτυα (Public Networks),** που εξυπηρετούν τις διασυνδέσεις μεταξύ απομακρυσμένων σημείων. Χρησιμοποιούνται όταν η απόσταση είναι μεγάλη και καθίσταται απαγορευτική, λόγω κόστους, η χρήση αποκλειστικών γραμμών.

- Ανάλογα με την τεχνική προώθησης της πληροφορίας διακρίνονται σε :

  - ☞ **Δίκτυα μεταγωγής και Δίκτυα Ακρόασης.**

- Ανάλογα με το λειτουργικό σύστημα των υπολογιστών που τα αποτελούν διακρίνονται σε :

  - ☞ **Δίκτυο Windows**
  - ☞ **Δίκτυο Novel (Dos)**
  - ☞ **Δίκτυο apletalk (για υπολογιστές τύπου Macintosh)**

- Ανάλογα με το πρωτόκολλο επικοινωνίας που χρησιμοποιούν διακρίνονται σε :

  - ☞ **Δίκτυα TCP/IP**
  - ☞ **Δίκτυα NET/BEUI**

- Και ανάλογα με την σχεδίαση τους διακρίνονται σε :

  - **Client-Server (πελάτης-διακομιστής)**
  - **Peer to Peer (ομότιμα δίκτυα).**

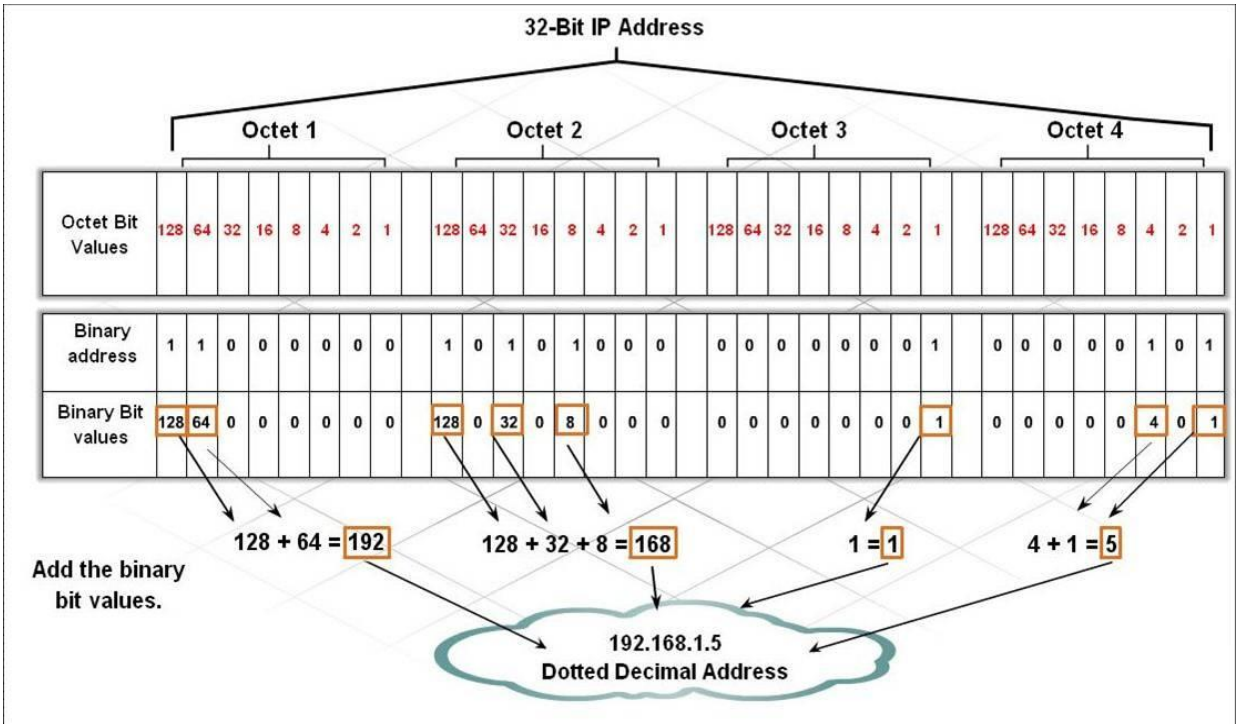
1.4 Διευθύνση IP και Μάσκα υποδικτύου

Η σύνδεση τοπικών δικτύων μεταξύ τους αλλά και με δίκτυα ευρείας περιοχής, δημιουργεί ένα διαδίκτυο. Οι υπολογιστές που συνδέονται σ' ένα διαδίκτυο συνήθως δεν διαθέτουν το ίδιο λειτουργικό σύστημα. Για να αλληλεπιδράσουν μεταξύ τους χρησιμοποιούν το πρωτόκολλο IP (Internet Protocol). Πρωτόκολλο IP λοιπόν ονομάζεται η μέθοδος, κατά την οποία υπολογιστές με διαφορετικό λειτουργικό σύστημα μπορούν να ανταλλάσουν μεταξύ τους πληροφορίες. Κάθε υπολογιστής ο οποίος διαθέτει το κατάλληλο λογισμικό για την υποστήριξη αυτού του πρωτοκόλλου, μπορεί να επικοινωνεί με οποιονδήποτε άλλο υπολογιστή που υποστηρίζει το πρωτόκολλο IP.

Στο πρωτόκολλο IP κάθε υπολογιστής που είναι συνδεδεμένος στο δίκτυο έχει μια **σταθερή μοναδική διεύθυνση** που επιτρέπει σε κάθε άλλο υπολογιστή που είναι συνδεδεμένος να το προσδιορίσει απόλυτα. Η εξέλιξη του πρωτοκόλλου IP είναι το πρωτόκολλο TCP/IP το οποίο εξασφάλισε την ακριβή μεταφορά των πληροφοριών από και προς έναν άλλο υπολογιστή. Η επιτυχία του πρωτοκόλλου TCP/IP ως του πρωτοκόλλου δικτύου του Internet οφείλεται κυρίως στη δυνατότητά του να συνδέει δίκτυα διαφόρων μεγεθών και συστήματα διαφορετικών τύπων.

Μία διεύθυνση IP (Ip address - Internet Protocol address), είναι ένας μοναδικός αριθμός που χρησιμοποιείται από συσκευές για τη μεταξύ τους αναγνώριση και συνεννόηση σε ένα δίκτυο υπολογιστών Κάθε συσκευή που ανήκει στο δίκτυο, όπως δρομολογητές (routers), υπολογιστές, εκτυπωτές, μηχανές για fax μέσω Internet, και ορισμένα τηλέφωνα, πρέπει να έχει τη δική της μοναδική διεύθυνση. Μία διεύθυνση IP μπορεί να θεωρηθεί το αντίστοιχο μιας διεύθυνσης κατοικίας . Όπως κάθε διεύθυνση κατοικίας και αριθμός τηλεφώνου αντιστοιχούν σε ένα και μοναδικό κτίριο ή τηλέφωνο, μια IP διεύθνση χρησιμοποιείται για τη μοναδική αναγνώριση ενός υπολογιστή ή άλλης συσκευής που συνδέεται στο δίκτυο.

Έτσι λοιπόν μια διεύθυνση IP είναι ένας αριθμός 32 bit που προσδιορίζει μοναδικά έναν κεντρικό υπολογιστή (υπολογιστή ή άλλη συσκευή, όπως εκτυπωτή ή δρομολογητή) σε ένα δίκτυο TCP/IP.Οι διευθύνσεις IP εκφράζονται κανονικά σε μορφή δεκαδικών διευθύνσεων με τελείες, με τέσσερις αριθμούς να χωρίζονται από τελείες, όπως 192.168.1.5

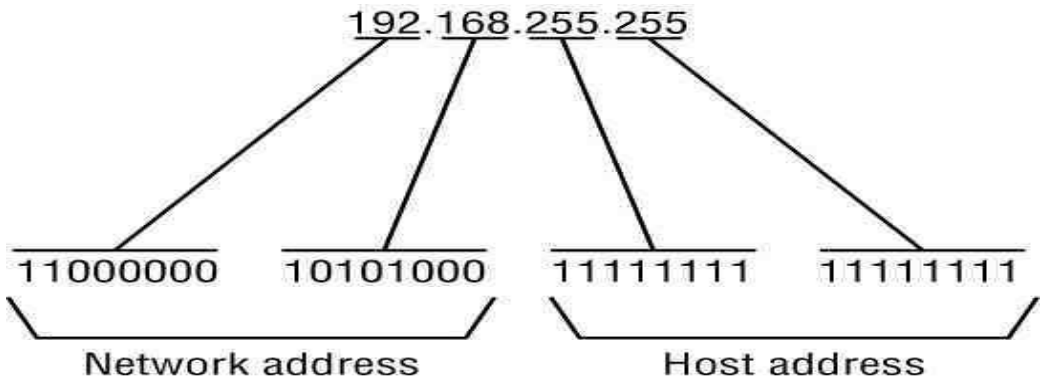


Εικόνα 1: Δεκαδική Παράσταση διευθύνσεων IP

Για να λειτουργήσει αποδοτικά ένα ευρύτερο δίκτυο (wide area network - WAN) TCP/IP ως συλλογή δικτύων, οι δρομολογητές που μεταβιβάζουν πακέτα δεδομένων μεταξύ δικτύων δεν γνωρίζουν την ακριβή θέση του κεντρικού υπολογιστή για τον οποίο προορίζεται ένα πακέτο πληροφοριών. Οι δρομολογητές γνωρίζουν μόνο σε ποιο δίκτυο ανήκει ο κεντρικός υπολογιστής και χρησιμοποιούν τις πληροφορίες που είναι αποθηκευμένες στον πίνακα διαδρομών τους, για να καθορίσουν τον τρόπο μεταβίβασης του πακέτου στο δίκτυο κεντρικού υπολογιστή προορισμού. Έτσι μία διεύθυνση IP έχει δύο συστατικά μέρη :

- Το ένα προσδιορίζει την ταυτότητα του δικτύου του οποίου μέλος είναι αυτός ο υπολογιστής
- και το άλλο την ταυτότητα του υπολογιστή.

Έτσι η IP του 192.168.255.255 δηλώνει



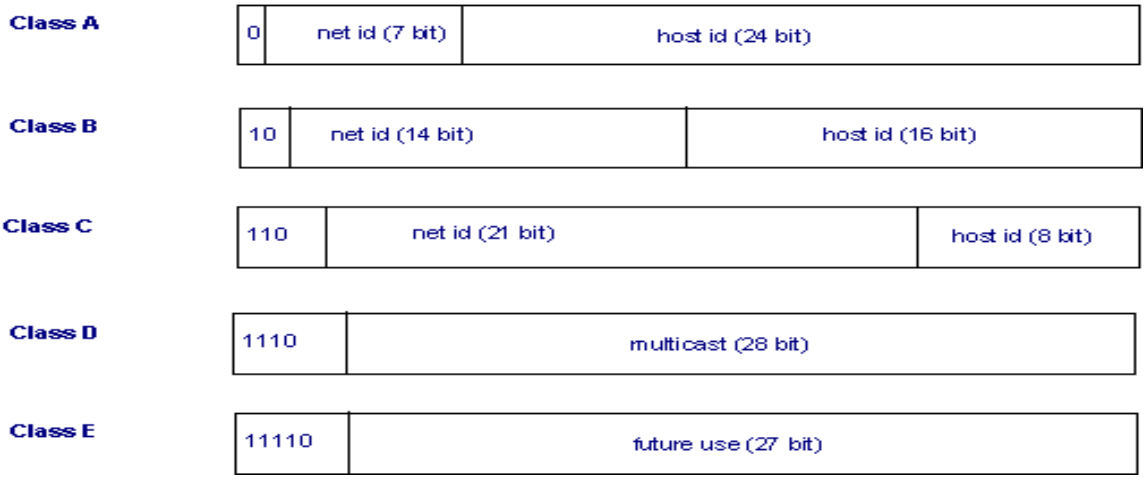
Εικόνα 2: Συστατικά μέρη διεύθυνσης IP

Η χρησιμότητα αυτού του διαχωρισμού φαίνεται στην λειτουργία των routers που δρομολογούν τα πακέτα IP βάση της διεύθυνσης του δικτύου και μόνο, αγνοώντας την διεύθυνση του συγκεκριμένου υπολογιστή. Με τον τρόπο αυτό απλοποιούνται οι πίνακες δρομολόγησης καθώς ένα ολόκληρο LAN με πολλούς υπολογιστές φαίνεται στους routers με μία μόνο διεύθυνση.

Από τα 32 bit της διεύθυνσης δεν είναι σταθερού μήκους ούτε το τμήμα της διεύθυνσης δικτύου ούτε το τμήμα της διεύθυνσης υπολογιστή, Ανάλογα με το πόσο μεγάλο είναι το τμήμα της IP διεύθυνσης που αφιερώνεται ως διεύθυνση δικτύου οι διευθύνσεις διακρίνονται σε τρεις κλάσεις και τύπους :

Κλάση A : 8 bit διεύθυνση δικτύου / 24 bit διεύθυνση υπολογιστή
Κλάση B : 16 bit διεύθυνση δικτύου / 16 bit διεύθυνση υπολογιστή
Κλάση C : 24 bit διεύθυνση δικτύου / 8 bit διεύθυνση υπολογιστή

Οι κλάσεις D και E υπάρχουν, αλλά γενικά δεν χρησιμοποιούνται από τελικούς χρήστες.



Εικόνα 3: Κλάσεις διευθύνσεων IP

Ο εικόνα 3 μας δείχνει το πλήθος δικτύων και υπολογιστών που μπορεί κατά μέγιστο να εξυπηρετήσει κάθε κλάση διευθύνσεων λαμβανομένων υπόψη και των bit που δεσμεύονται για τον προσδιορισμό της κλάσης.

ΤΥΠΟΣ	ΤΙΜΗ ΠΡΩΤΟΥ ΠΕΔΙΟΥ	ΧΡΗΣΗ ΠΕΔΙΩΝ ΔΙΕΥΘΥΝΣΗΣ
A	0...127	Network . Host . Host . Host
B	128...191	Network . Network . Host . Host
C	192...223	Network . Network . Network . Host
D	224...239	Multicast
E	240...254	Για μελλοντική χρήση

Εικόνα 4: Πλήθος δικτύων και υπολογιστών ανά κλάση

Κάθε κλάση διεύθυνσης έχει διαφορετική προεπιλεγμένη μάσκα υποδικτύου .Η μάσκα υποδικτύου χρησιμοποιείται από το πρωτόκολλο TCP/IP για να προσδιορίσει αν ένας κεντρικός υπολογιστής βρίσκεται στο τοπικό υποδίκτυο ή σε απομακρυσμένο δίκτυο.

Σαν **μάσκα υποδικτύου** εννοούμε μια σειρά από 32 bit όπου τα bit της διεύθυνσης δικτύου έχουν τιμή 1 και τα bit της διεύθυνσης υπολογιστή τιμή 0. Η τιμή της μάσκας παριστάνεται στη συνέχεια σε δεκαδική μορφή με τον ίδιο τρόπο που χρησιμοποιείται στις IP διευθύνσεις. Στο σχήμα 2 φαίνονται οι εξ ορισμού (default) μάσκες για τις κλάσεις διευθύνσεων A, B και C

Κλάση	Βασική Μάσκα (Δυαδικό)	Βασική Μάσκα (Δεκαδικό)
A	11111111.00000000.00000000.00000000	255.0.0.0
B	11111111.11111111.00000000.00000000	255.255.0.0
C	11111111.11111111.11111111.00000000	255.255.255.0

Πίνακας 1: Δεκαδική και Δυαδική παράσταση μάσκας υποδικτύου

Έτσι:

- Τα δίκτυα της κλάσης A χρησιμοποιούν μια προεπιλεγμένη μάσκα υποδικτύου 255.0.0.0 και έχουν την περιοχή 0-127 ως την πρώτη οκτάδα τους.
- Τα δίκτυα της κλάσης B χρησιμοποιούν μια προεπιλεγμένη μάσκα υποδικτύου 255.255.0.0 και έχουν την περιοχή 128-191 ως την πρώτη οκτάδα τους.
- Τα δίκτυα της κλάσης C χρησιμοποιούν μια προεπιλεγμένη μάσκα υποδικτύου 255.255.255.0 και έχουν την περιοχή 192-223 ως την πρώτη οκτάδα τους.

1.5 Πρωτόκολλο TCP/IP

Τα πρωτόκολλα δικτύωσης, καθορίζουν τους κοινά αποδεκτούς κανόνες και διαδικασίες, βάσει των οποίων οι υπολογιστές ενός δικτύου επικοινωνούν και ανταλλάσσουν δεδομένα μεταξύ τους, φροντίζοντας ταυτόχρονα για τη διασφάλιση

της ακεραιότητας των μεταφερόμενων «πακέτων». Το αρχικό επικοινωνιακό πρωτόκολλο (σύνολο κανόνων επικοινωνίας) ήταν το **NCP (Network Control Protocol)** το οποίο όμως αντικαταστάθηκε με το πέρασμα του χρόνου, διότι αποδείχθηκε αργό κι ανασφαλές, από ένα υψηλών προδιαγραφών και μεγαλύτερης πολυπλοκότητας πρωτόκολλο, το γνωστό σήμερα σαν **TCP/IP (Transmission Control Protocol/Internet Protocol).**

Το πρωτόκολλο TCP/IP (Transmission Control Protocol/Internet Protocol Πρωτόκολλο Ελέγχου Μετάδοσης/ Πρωτόκολλο Internet) είναι μια συλλογή πρωτοκόλλων επικοινωνίας το οποίο επιτρέπει την επικοινωνία μεταξύ υπολογιστών διαφορετικών τύπων, καθορίζοντας τον τρόπο με τον οποίο τα μέλη ενός δικτύου θα αλληλεπιδράσουν μεταξύ τους και παρέχοντας ταυτόχρονα πρόσβαση στο Διαδίκτυο και τους πόρους του. Η ονομασία TCP/IP προέρχεται από τις συντομογραφίες των δυο κυριότερων πρωτοκόλλων που περιέχει το TCP ή Transmission Control Protocol (Πρωτόκολλο Ελέγχου Μετάδοσης) και το IP ή Internet Protocol (Πρωτόκολλο Διαδικτύου).

1.6 Ασφάλεια στα Δίκτυα Υπολογιστών

Η ασφάλεια στα δίκτυα υπολογιστών έχει να κάνει με την πρόληψη και ανίχνευση μη εξουσιοδοτημένων ενεργειών των χρηστών του δικτύου καθώς και τη λήψη μέτρων. Ποιο συγκεκριμένα, η ασφάλεια στα δίκτυα υπολογιστών σχετίζεται με την **πρόληψη** (prevention) τη λήψη δηλαδή μέτρων για να προληφθούν επιθέσεις και φθορές στις μονάδες ενός δικτύου υπολογιστών, την **ανίχνευση** (detection) δηλαδή την λήψη μέτρων για την ανίχνευση του πότε, πώς και από ποιον προκλήθηκε φθορά σε μία από τις παραπάνω μονάδες και την **αντίδραση** (reaction) την λήψη δηλαδή μέτρων για την αποκατάσταση ή ανάκτηση των συστατικών ενός δικτύου.

Σήμερα η έννοια της ασφάλειας των δικτύων υπολογιστών αλλά και των πληροφοριακών συστημάτων γενικότερα, συνδέεται στενά με τρεις βασικές έννοιες:

- ☞ Διαθεσιμότητα (Availability)
- ☞ Εμπιστευτικότητα (Confidentiality)
- ☞ Ακεραιότητα (Integrity)

☞ **Διαθεσιμότητα :**

Διαθεσιμότητα ονομάζεται η ικανότητα ενός δικτύου να είναι προσπελάσιμες και χωρίς αδικαιολόγητη καθυστέρηση οι υπηρεσίες ενός ή περισσότερων υπολογιστών όταν τις χρειάζεται μια εξουσιοδοτημένη μονάδα. Με τον όρο διαθεσιμότητα εννοούμε ότι τα δεδομένα είναι προσβάσιμα και οι υπηρεσίες λειτουργούν, παρά τις όποιες τυχόν διαταραχές του δικτύου, όπως διακοπή τροφοδοσίας, φυσικές καταστροφές, ατυχήματα ή επιθέσεις. Αυτό σημαίνει ότι οι εξουσιοδοτημένοι χρήστες των υπολογιστικών συστημάτων και των υπολογιστών του δικτύου δεν αντιμετωπίζουν προβλήματα άρνησης εξυπηρέτησης (denial of service) όταν επιθυμούν να προσπελάσουν τους πόρους του δικτύου.

☞ **Εμπιστευτικότητα :**

Με τον όρο εμπιστευτικότητα ορίζουμε την αποφυγή αποκάλυψης πληροφοριών σε μη εξουσιοδοτημένους χρήστες. Συγκεκριμένα, οι πληροφορίες και τα δεδομένα δικτύου είναι προσπελάσιμα μόνο από εξουσιοδοτημένους χρήστες καθιστώντας αδύνατο σε εξωτερικούς χρήστες να αποκτήσουν οποιαδήποτε πρόσβαση. Κοινές εκφάνσεις της εμπιστευτικότητας είναι η **ιδιωτικότητα-Privacy,** δηλαδή το δικαίωμα απομόνωσης και ελευθερίας ενός χρήστη από αδικαιολόγητες παρεμβάσεις και "εισβολές" και η **μυστικότητα** δηλαδή η προστασία δεδομένων προσωπικού χαρακτήρα που αφορούν συγκεκριμένα πρόσωπα

☞ **Ακεραιότητα :**

Πρόκειται για την επιβεβαίωση ότι τα δεδομένα που έχουν αποσταλεί, παραληφθεί ή αποθηκευτεί είναι πλήρη και δεν έχουν υποστεί αλλοίωση. Στην πληροφορική, ακεραιότητα σημαίνει πρόληψη μη εξουσιοδοτημένης μεταβολής πληροφοριών, δηλαδή, πρόληψη από μη εξουσιοδοτημένη εγγραφή ή διαγραφή, συμπεριλαμβανομένης και της μη εξουσιοδοτημένης δημιουργίας δεδομένων. Επομένως, η μετατροπή, διαγραφή και δημιουργία των δεδομένων ενός υπολογιστικού συστήματος, γίνεται μόνο από εξουσιοδοτημένα μέρη.

1.7 Επιθέσεις

Στις μέρες μας οι επιθέσεις εναντίον δικτύων και υπολογιστικών συστημάτων έχουν πάρει μεγάλη έκταση και προκαλούν σοβαρά προβλήματα σε οργανισμούς, εταιρείες και χρήστες. Συνήθως στόχος ενός κακόβουλου χρήστη είναι η κατάργηση της κανονικής λειτουργίας ενός συστήματος και η υποκλοπή απορρήτων δεδομένων προς όφελός του. Οι κατηγορίες των πιθανών επιθέσεων που μπορεί να δεχθεί ένα δίκτυο ή υπολογιστικό σύστημα είναι οι εξής:

1.7.1 Απόκτηση πληροφοριών για το σύστημα

Σκοπός της επίθεσης αυτής είναι η απόκτηση μιας καλής εικόνας του υπό εξέταση δικτύου (αρχιτεκτονική και συστήματα), η εύρεση των υπηρεσιών που λειτουργούν και του μέρους που αυτές εκτελούνται και το είδος του λογισμικού που χρησιμοποιείται (πχ. λειτουργικό σύστημα, mail servers, DNS servers κτλ.). Η επίθεση μπορεί να γίνει με εντελώς *παθητικό* τρόπο (παρακολούθηση της κίνησης προς και από το δίκτυο) αλλά για να επιτευχθεί αυτό χρειάζεται παρουσία του κακόβουλου σε ένα σημείο διέλευσης της κίνησης. Επίσης μέσω κοινών εργαλείων δικτύου (DNS zone transfers, ping, traceroute, εξέταση σελίδων web) επιτυγχάνεται και *ενεργητική* διερεύνηση των δικτύων. Άλλα γνωστά εργαλεία που μπορούν να χρησιμοποιηθούν για να γίνει οποιαδήποτε επίθεση είναι τα IPscanners και το Nmap, το οποίο χρησιμοποιείται για Finger Printing (ανακάλυψη του λειτουργικού συστήματος) και για Port Scanning (ανακάλυψη των υπηρεσιών που είναι συνδεδεμένες στα ports).

1.7.2 Μη εξουσιοδοτημένη πρόσβαση

Σκοπός της παραπάνω επίθεσης είναι η υποκλοπή κωδικών. Η υποκλοπή μπορεί να γίνει κατ' αρχήν μέσω «Social Engineering». Το Social Engineering είναι ο πιο εύκολος τρόπος υποκλοπής και στηρίζεται συνήθως στην αφέλεια των διαχειριστών. Η υποκλοπή κωδικών οφείλεται και σε διαχειριστικές τακτικές όπως η χρήση «αδύναμων» κωδικών (πχ. λέξεις με νόημα και όχι τυχαία χρήση γραμμάτων,

αριθμών και συμβόλων) και η κακή προστασία των κωδικών (αποστολή κωδικού χωρίς κρυπτογράφηση). Επίσης μη εξουσιοδοτημένη πρόσβαση μπορεί να προέλθει και από μη προβλεπόμενα σημεία εξόδου του δικτύου (πχ. modems). Τέλος, κυρίως τα τελευταία χρόνια με τη ραγδαία ανάπτυξη και χρήση των ασυρμάτων δικτύων η μη εξουσιοδοτημένη πρόσβαση είναι πολύ πιο εύκολη σε σχέση με τα ενσύρματα μέσα. Η ασύρματη μετάδοση παρουσιάζει αρκετά κενά ασφαλείας σε συνδυασμό με την χρήση αδύναμων πρωτοκόλλων ασφαλείας (πχ. WEP - Wired Equivalent Privacy). Το WEP είναι ένα πρωτόκολλο ασφάλειας του MAC επιπέδου, το οποίο βασίζεται στη χρήση του αλγόριθμου κρυπτογράφησης RC4 με κάποιο μυστικό κλειδί 64 ή 128 bit. Το μυστικό κλειδί είναι κοινό ανάμεσα στους χρήστες που επικοινωνούν.

1.7.3 Υποκλοπή και παραποίηση

Η επίθεση αυτή χωρίζεται σε δυο κατηγορίες το **Packet sniffing** και το **“Man-in-the-Middle attacks”**. Το packet sniffing είναι η διαδικασία κατά την οποία γίνεται παρακολούθηση μέσω ενός sniffer (πχ. Snort) των πακέτων που κυκλοφορούν στο δίκτυο και εκμαίευση διαφόρων λειτουργικών πληροφοριών. Μπορεί να συμβεί σε δίκτυα με Hub, σε μη ασφαλισμένα ασύρματα δίκτυα. Κάθε πληροφορία που κυκλοφορεί μη κρυπτογραφημένη είναι διαθέσιμη σε αυτόν που παρακολουθεί (πχ. telnet passwords, web passwords, email, αριθμοί πιστωτικών καρτών κτλ.).

Οι επιθέσεις “Man-in-the-Middle” προκύπτουν όταν κάποιος παρεμβληθεί σε μια επικοινωνία και προσπαθήσει είτε να υποκλέψει τα στοιχεία είτε να υποκριθεί ότι είναι κάποιος τρίτος φορέας (πχ. ARP poisoning, TCP session hijacking, DNS cache poisoning). Το ARP poisoning είναι τύπος παραβίασης σε δίκτυο υπολογιστών το οποίο βασίζεται στο πρωτόκολλο ARP. Ο κακόβουλος χρήστης μπορεί, μεταδίδοντας λανθασμένα πακέτα ARP, να μπερδέψει άλλους host ώστε να στείλουν τα πλαίσια δεδομένων (data frames) τους σε άλλον υπολογιστή χωρίς να το αντιληφθούν και παράλληλα έχει τη δυνατότητα να αποκλείσει έναν host από ένα δίκτυο. Το Session Hijacking είναι η διαδικασία της κατάληψης μιας ενεργής συνεδρίας. Ένας από τους κύριους λόγους που είναι χρήσιμη μία τέτοια επίθεση είναι η παράκαμψη της διαδικασίας ελέγχου ταυτότητας ώστε να αποκτήσουμε πρόσβαση σε έναν υπολογιστή. Δεδομένου ότι η συνεδρία είναι ήδη ενεργή, δεν υπάρχει λόγος για εκ νέου έλεγχο της ταυτότητας και έτσι ο επιτιθέμενος μπορεί να έχει εύκολη πρόσβαση

σε πόρους και ευαίσθητες πληροφορίες, όπως κωδικούς πρόσβασης, στοιχεία της τράπεζας και πολλά άλλα. Η επίθεση DNS cache poisoning (που πολλές φορές αναφέρεται και ως cache pollution - μόλυνση) είναι μια τεχνική επίθεσης που επιτρέπει στον επιτιθέμενο να εισαγάγει ψεύτικες πληροφορίες DNS στον εξυπηρετητή.

Ο κυριότερος τρόπος αντιμετώπισης των επιθέσεων αυτών είναι η χρήση κρυπτογραφίας σε κάποιες εφαρμογές όπως η αντικατάσταση του telnet με ssh και η χρήση της κρυπτογραφημένης εκδοχής του IMAP για email (https). Επίσης απαραίτητη είναι η χρήση ψηφιακών πιστοποιητικών και υπογραφών τα οποία προσφέρουν κρυπτογραφία και επιβεβαίωση ταυτότητας.

1.7.4 Malware - Κακόβουλο λογισμικό

Η παραπάνω επίθεση είναι ίσως η πιο γνωστή μιας και αντιμέτωποι με αυτή έρχονται όλοι οι χρήστες του διαδικτύου σχεδόν καθημερινά. Χωρίζεται σε δύο κατηγορίες. Μια κατηγορία είναι οι **ιοί δούρειοι ίπποι** - Trojans. Χαρακτηριστικό των Δούρειων ίππων είναι η αργή διάδοση τους σε ένα δίκτυο υπολογιστών και το γεγονός ότι είναι εγκατεστημένοι σε εκτελέσιμα προγράμματα. Η δεύτερη κατηγορία είναι οι **αυτόματα διαδιδόμενοι ιοί** - worms. Τα worms εκμεταλλεύονται προβλήματα λογισμικού σε λειτουργικά συστήματα ή εφαρμογές για να μεταδοθούν στο διαδίκτυο. Επίσης γίνεται χρήση παραπλανητικών email που παρασύρουν το χρήστη στο να εκτελέσει συγκεκριμένες ενέργειες του υπολογιστή του, ενώ παράλληλα γίνεται χρήση του καταλόγου διευθύνσεων του χρήστη για τη μετάδοση σε νέους χρήστες.

1.7.5 Επιθέσεις Άρνησης Υπηρεσίας (DoS – Denial of Service) και DDOS επιθέσεις (Κατανεμημένες D.O.S. Επιθέσεις - Distributed D.O.S. Attacks)

Οι DoS (Denial of Service) επιθέσεις αποτελούν μια από τις σημαντικότερες απειλές και μεταξύ των δυσκολότερων προβλημάτων ασφάλειας στο σημερινό διαδίκτυο. Ιδιαίτερου ενδιαφέροντος είναι οι DDOS (Distributed D.O.S. Attacks) επιθέσεις, των οποίων οι επιπτώσεις μπορούν να είναι ιδιαίτερα σημαντικές. Με

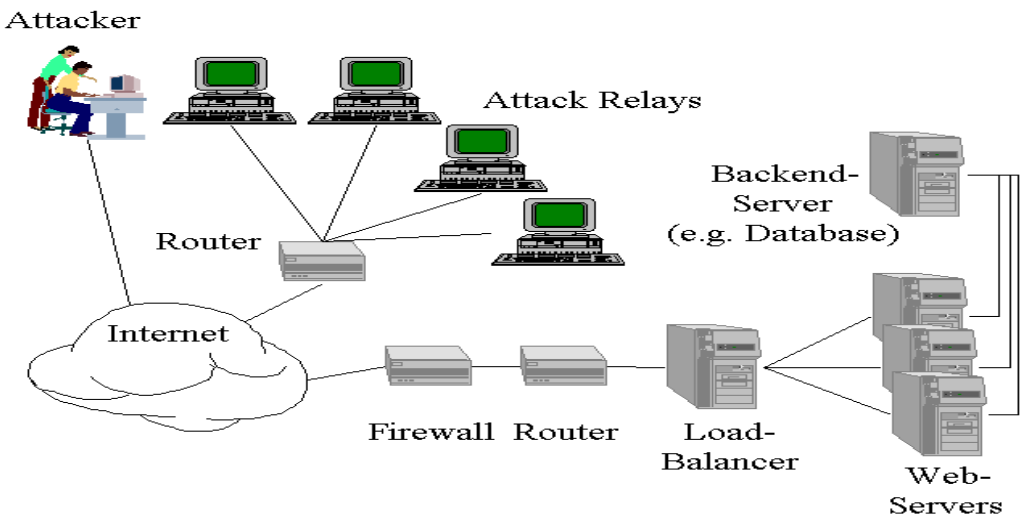
ελάχιστη ή καμία προειδοποίηση, μια επίθεση DDoS μπορεί εύκολα να εξαντλήσει τους υπολογιστικούς και επικοινωνιακούς πόρους του θύματός της μέσα σε ένα μικρό χρονικό διάστημα.

1.7.5.1 DoS Επιθέσεις – Denial of Service Attacks

Οι **DoS** επιθέσεις αποσκοπούν όχι την υποκλοπή κωδικών και προγραμμάτων, αλλά να θέσουν εκτός λειτουργίας τον υπολογιστή/σύστημα στον οποίο επιτίθενται. Σκοπός αυτών των επιθέσεων είναι η υπερφόρτωση του συστήματος – στόχος, καταναλώνοντας μνήμη και bandwidth έτσι ώστε να μην είναι σε θέση πλέον να εξυπηρετήσει τους κανονικούς χρήστες. Αυτό επιτυγχάνεται μέσω της αποστολής πακέτων δεδομένων (data packets) σε υπερβολικά μεγάλο ρυθμό έτσι ώστε το σύστημα-στόχος να αδυνατεί να τα επεξεργαστεί με αποτέλεσμα τις περισσότερες φορές να πρέπει να γίνει επανεκκίνηση (reboot) του συστήματος.

Οι πιο συνηθισμένοι τύποι Denial Of Service είναι οι επιθέσεις που εκμεταλλεύονται αδυναμίες του πρωτοκόλλου **TCP/IP,** οι επιθέσεις που εκμεταλλεύονται αδυναμίες του **IPv4** και οι επιθέσεις που προσπαθούν να **εξαντλήσουν όλους τους πόρους** (resources) - μνήμη, CPU-RAM, Bandwidth - του συστήματος στόχου με αποτέλεσμα την διακοπή της λειτουργίας του.

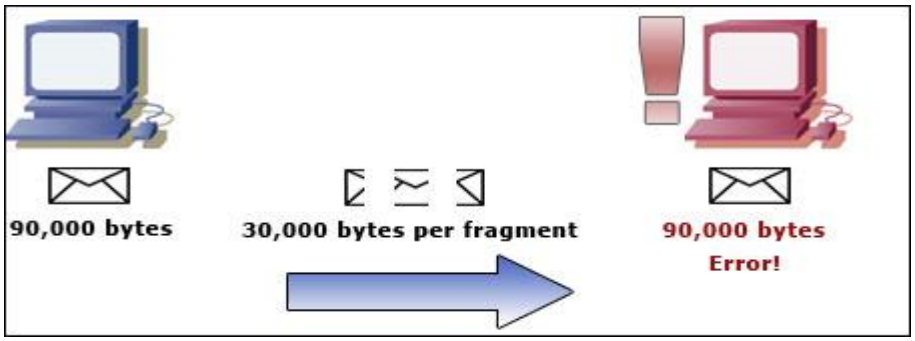
Μερικές DoS επιθέσεις που σχετίζονται με το πρωτόκολλο TCP/IP είναι οι **Ping of Death, Teardrop, SYN Attack, Land Attack και Smurf Attack**.



Εικόνα 5: Denial of service attack

- **Ping of Denial** (γνωστή και ως Ping of Death). Αυτή η επίθεση είναι πολύ γνωστή και χρησιμοποιούταν παλαιότερα για να κάνει απομακρυσμένα συστήματα να "παγώνουν" (hang) ή ακόμα και να κάνουν αυτόματη επανεκκίνηση (reboot), έτσι ώστε οι χρήστες να μην μπορούν να τα χρησιμοποιήσουν. Αυτό πλέον δεν είναι εφικτό μια και στις μέρες μας σχεδόν όλοι οι διαχειριστές των συστημάτων (Systems Administrators) έχουν αναβαθμίσει τα συστήματά τους κάνοντάς τα ασφαλή από τέτοιες επιθέσεις.

Ο τρόπος για να γίνει αυτή η επίθεση είναι να στείλει κάποιος ένα πακέτο data (data packet) που υπερβαίνει το μέγιστο επιτρεπόμενο όριο bytes του πρωτοκόλλου TCP/IP, που είναι 65536. Στέλνοντας ένα πακέτο data (data packet) μεγαλύτερο από αυτό, αμέσως το σύστημα-στόχος παθαίνει κατάρρευση (crash) ή "παγώνει" (hang) ή κάνει επανεκκίνηση (reboot). Το Ping Of Death έγινε πολύ δημοφιλές λόγω της ευκολίας στην υλοποίηση του. Επίσης αυτού του τύπου η επίθεση ήταν και είναι ακόμα πολύ δημοφιλής στο IRC (Internet Relay Chat).

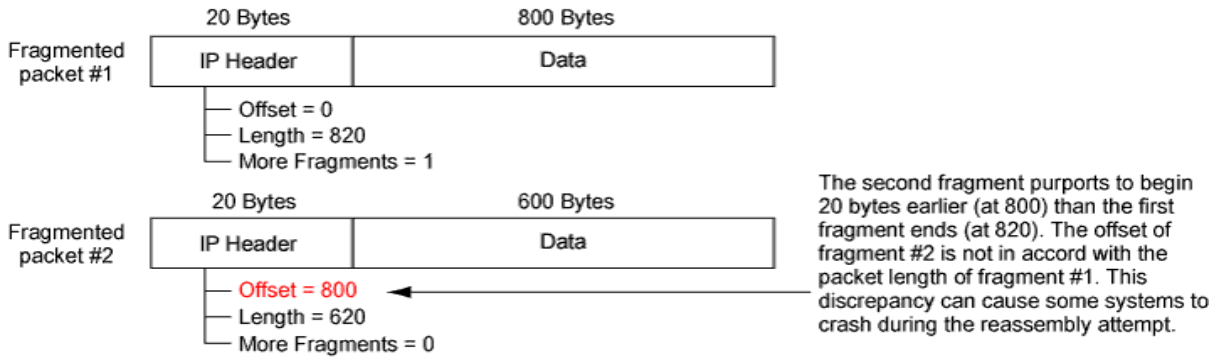


Εικόνα 6 : Ping of Denial Attack

- **Teardrop:** Αυτή η επίθεση εκμεταλλεύεται την αδυναμία του πρωτοκόλλου TCP/IP στην επανασύνδεση (reassembly) των πακέτων δεδομένων (data packets) κατά την λήψη τους. Όταν στέλνονται data στο Internet αυτά κατανέμονται σε μικρότερα κομμάτια στην υπολογιστή που κάνει την μετάδοση και συναρμολογούνται πάλι στον υπολογιστή που λαμβάνει. Ας υποθέσουμε ότι θέλουμε να στείλουμε 8000 bytes από έναν υπολογιστή σε έναν άλλον. Δεν θα τα στείλουμε όλα μαζί με μία μετάδοση (transmission) αλλά θα κοπούν σε μικρότερα πακέτα data (data packets) και κάθε πακέτο θα έχει συγκεκριμένο κομμάτι από τα 8000 bytes όπως : το 1ο πακέτο θα έχει byte 1 έως byte 1500,

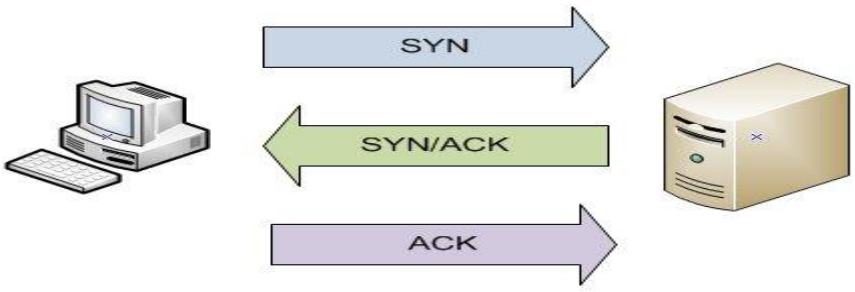
το 2ο πακέτο θα έχει byte 1501 έως byte 3000, το 3ο πακέτο θα έχει byte 3001 έως byte 4000 κ.λ.π.

Αυτά τα πακέτα έχουν στο αρχικό τους κομμάτι (TCP header) ένα πεδίο (offset) που περιγράφει πως θα γίνει η συναρμολόγηση στο σύστημα που θα λάβει τα πακέτα. Στην επίθεση αυτή τα πακέτα που στέλνονται υπερκαλύπτουν το ένα το άλλο με αποτέλεσμα όταν το σύστημα που τα λαμβάνει προσπαθεί να τα συναρμολογήσει (reassembly) να καταρρεύσει (crash) ή να "παγώσει" (hang) ή να κάνει επανεκκίνηση (reboot).



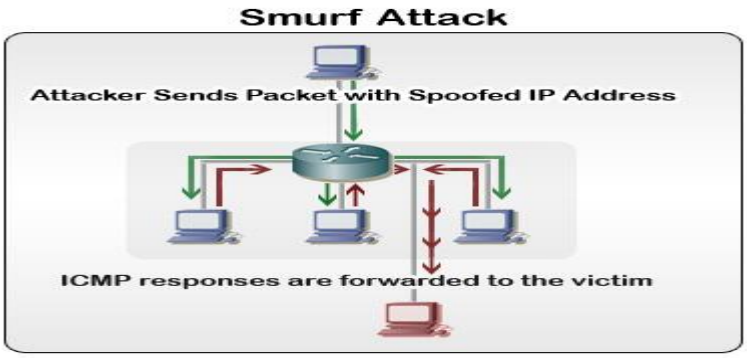
Εικόνα 7: Teardrop Επίθεση

- **SYN Flood Attack ή Land Attack:** Αυτές οι δύο επιθέσεις είναι ίδιες με τη διαφορά ότι η SYN attack (Synchronise Flooding Attack) χρησιμοποιεί ανύπαρκτες διευθύνσεις IP για να αναγκάσει το σύστημα στόχο να περιμένει μια απάντηση από ανύπαρκτη IP διεύθυνση ενώ η Land Attack αντί για ανύπαρκτες IP διευθύνσεις χρησιμοποιεί την IP του συστήματος-στόχου και αυτό έχει ως αποτέλεσμα μια ατελείωτη σειρά (endless loop) επεξεργασίας για το σύστημα. Και οι δύο επιθέσεις, όταν γίνονται μαζικά, έχουν ως αποτέλεσμα την κατάρρευση (crash) ή το "πάγωμα" (hang) ή την επανεκκίνηση (reboot) του συστήματος.



Εικόνα 8: SYN Flood Attack

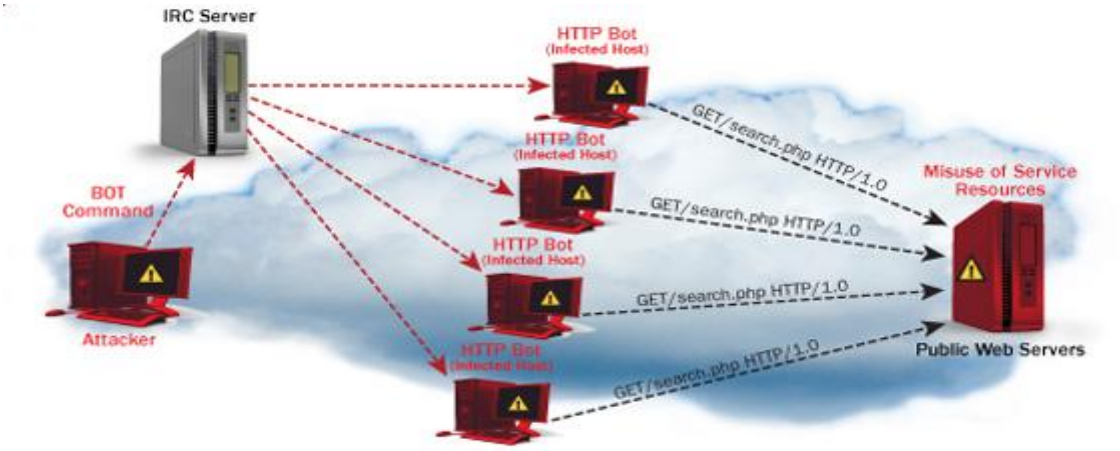
- **Smurf Attack:** Η Smurf attack είναι μια επίθεση όπου στέλνεται ένας υπέρογκος αριθμός από ping requests συνήθως στον router του δικτύου, χρησιμοποιώντας ψεύτικες (spoofed) IP διευθύνσεις μέσα από το ίδιο το δίκτυο. Κάθε φορά που ο router δέχεται ένα τέτοιο Ping request είτε το διανέμει (route it) είτε απαντάει σε αυτό (echo back), με αποτέλεσμα να υπερφορτωθεί το δίκτυο και να σταματήσει να λειτουργεί.



Εικόνα 9: Smurf Attack

1.7.5.2 Κατανεμημένες D.O.S. Επιθέσεις - Distributed D.O.S. Attacks

Οι **DDoS επιθέσεις** αποτελούνται από ροές πακέτων από ανόμοιες πηγές. Αυτές χρησιμοποιούν την ισχύ ενός μεγάλου αριθμού συντονισμένων διαδικτυακών hosts για να καταναλώσουν κάποιο κρίσιμο πόρο στο θύμα τους και έτσι αυτό δεν μπορεί να εξυπηρετήσει τους νόμιμους πελάτες του. Η κίνηση συνήθως είναι τόσο συγκεντρωτική ώστε να είναι δύσκολο να διακρίνει κανείς τα νόμιμα πακέτα από τα πακέτα της επίθεσης.



Εικόνα 10: Distributed D.O.S. Attacks

Μια επίθεση DDoS χρησιμοποιεί πολλούς υπολογιστές για να προωθήσει μια συντονισμένη επίθεση DoS ενάντια σε έναν ή περισσότερους στόχους. Χρησιμοποιώντας την τεχνολογία πελατών / εξυπηρετητών, ο επιτιθέμενος είναι σε θέση να πολλαπλασιάσει την αποτελεσματικότητα μιας DoS επίθεσης εκμεταλλευόμενος τους πόρους πολλών υπολογιστών, οι οποίοι χρησιμεύουν σαν πλατφόρμες για την επίτευξη της επίθεσης. Η επίθεση DDoS είναι η πιο εξελιγμένη μορφή DoS επίθεσης. Ξεχωρίζει από τις άλλες επιθέσεις λόγω της δυνατότητάς της να αναπτύσσεται με έναν διανεμημένο τρόπο μέσω του διαδικτύου και να αθροίζει αυτές τις δυνάμεις για να δημιουργήσει μεγάλη διαστάσεων κίνηση. Οι DDoS επιθέσεις δεν προσπαθούν να παραβιάσουν το σύστημα του στόχου. Ο κύριος στόχος τους είναι να προκληθεί ζημιά σε ένα θύμα είτε για προσωπικούς λόγους, είτε για το υλικό κέρδος, είτε για τη δημοσιότητα.

Μια DDoS επίθεση αποτελείται από τέσσερα στοιχεία:

- **Τον πραγματικό επιτιθέμενο (attacker)**.
- **Τους χειριστές (handlers) ή τους κύριους**, οι οποίοι αποτελούνται από hosts στους οποίους τρέχει ένα ειδικό πρόγραμμα ικανό να ελέγχει πολλαπλούς πράκτορες. Τους πράκτορες (agents) ή zombie hosts επίθεσης, οι οποίοι είναι υπονομευμένοι hosts που τρέχουν ένα ειδικό πρόγραμμα που είναι αρμόδιο για την παραγωγή μιας ροής πακέτων προς το προοριζόμενο θύμα.
- **Το θύμα(victim) ή το στόχο (host)**.

Εικόνα 11: Architecture of a DDoS Attack

1.8 Προστασία στα Δίκτυα Υπολογιστών

Η προστασία των δικτύων και των υπολογιστικών συστημάτων είναι τεχνικά δύσκολο και οικονομικά ασύμφορο να επιτευχθεί για όλα τα είδη των επιθέσεων. Για να μπορέσει κάποιος να πει ποιες είναι κάθε φορά οι κατάλληλες ενέργειες ώστε να επιτευχθεί ένα ικανοποιητικό επίπεδο ασφάλειας πρέπει να εξετάσει αρκετές παραμέτρους, οι οποίες έχουν να κάνουν με τα συστήματα που θέλει να προστατέψει, με το είδος των υπηρεσιών που παρέχει καθένα από αυτά, με την δομή του δικτύου και τις απαιτήσεις των χρηστών του. Ας δούμε όμως κάποια βασικά συστατικά μιας υλοποίησης ασφαλείας σε ένα δίκτυο.

1.8.1 Antivirus

Τα antivirus είναι προγράμματα τα οποία τρέχουν σε ένα σύστημα και ελέγχουν κάθε διεργασία που εκτελείται για να δουν αν πρόκειται για μια διεργασία η οποία δεν κάνει κάτι παράνομο, ή για μια διεργασία που έχει χαρακτηριστεί ως ιός, σκουλήκι ή δούρειος ίππος και η οποία είναι κακόβουλη. Σήμερα υπάρχουν αρκετά τέτοια προγράμματα, εμπορικά ή μη, και τα οποία έχουν διάφορους τρόπους λειτουργίας. Ο συνηθέστερος είναι με την χρήση Signatures δηλαδή μια λίστα με ακολουθίες ενεργειών που θεωρεί ύποπτες ή ενδεικτικές επίθεσης. Όσα λειτουργούν με τον τρόπο αυτό ελέγχουν για διεργασίες που εκτελούν συγκεκριμένη ακολουθία ενεργειών, η οποία έχει αναγνωριστεί και χαρακτηριστεί ως παράνομη. Μόλις μια τέτοια ακολουθία ενεργειών εκτελεστεί στο σύστημα, τότε αυτομάτως το πρόγραμμα που την εκτέλεσε θεωρείται ιός και η εκτέλεσή του εμποδίζεται από το antivirus.

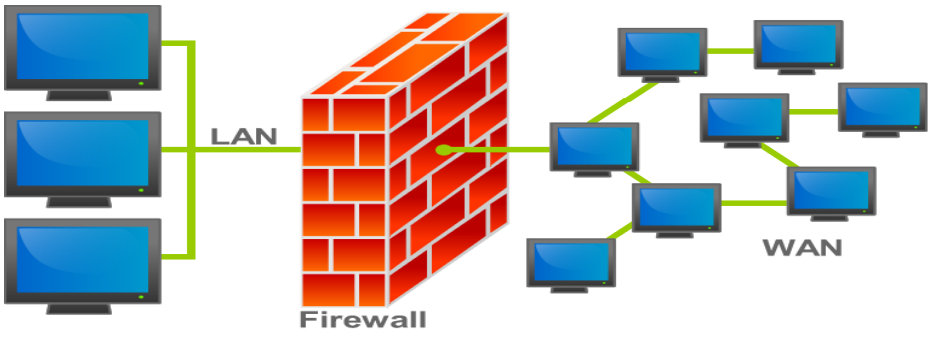
Η χρήση antivirus προγραμμάτων θεωρείται βασικό συστατικό ενός πλάνου ασφαλείας και δεν πρέπει να λείπει σχεδόν ποτέ από κανένα σύστημα. Επειδή μάλιστα είναι και από τις πρώτες τεχνολογίες που εμφανίστηκαν για την προστασία των συστημάτων, μπορεί να θεωρηθεί ότι βρίσκεται σε αρκετά ώριμο στάδιο, σε σχέση πάντα με τις άλλες λύσεις που έχουν προταθεί ή αναπτυχθεί για την προστασία ενός συστήματος από κακόβουλες διεργασίες.

Ωστόσο τα antivirus προγράμματα έχουν αρκετά μειονεκτήματα το μεγαλύτερο από τα οποία εντοπίζεται στο ότι υπάρχει άμεση εξάρτηση από τον κατασκευαστή του

προγράμματος. Επειδή ιοί και άλλα κακόβουλα προγράμματα αναπτύσσονται συνεχώς, πρέπει να βγαίνουν συνεχώς signatures για αυτά, ώστε να μπορεί το antivirus να τα αναγνωρίσει. Όσο πιο γρήγορα ανακοινωθούν τα νέα signatures, τόσο πιο γρήγορα θα προστατευτεί ο υπολογιστής. Έχει παρατηρηθεί πως το βασικότερο πρόβλημα των antivirus προγραμμάτων δεν έχει να κάνει με τον τρόπο λειτουργίας τους, αλλά με τους χρήστες τους. Αν ο χρήστης δεν ενημερώσει το πρόγραμμά του, τότε αυτομάτως αφήνει τον υπολογιστή του εκτεθειμένο σε κίνδυνο από τον οποίο θα μπορούσε να έχει προστατευτεί. Επομένως η προστασία που τους παρέχει είναι υποδεέστερη από αυτή που θα έπρεπε ή θα μπορούσε, απλά και μόνο επειδή ο χρήστης αμελεί να εκτελέσει μια απλή διαδικασία. Τέτοιες περιπτώσεις είναι πολύ συχνές και για το λόγο αυτό είναι περισσότερο από θεμιτή η ύπαρξη ενός μηχανισμού αυτόματης ενημέρωσης του προγράμματος. Αν μάλιστα υπάρχει η δυνατότητα ύπαρξης ενός εξυπηρετητή στο δίκτυο, ο οποίος θα αναλαμβάνει να ενημερώνει τους πελάτες του, τότε έχουμε ακόμη καλύτερη κατάσταση.

1.8.2 Firewall

Το firewall ή τείχος προστασίας, χρησιμοποιείται για να δηλώσει κάποια συσκευή ή πρόγραμμα που είναι έτσι ρυθμισμένο ώστε να επιτρέπει ή να απορρίπτει πακέτα δεδομένων που περνούν από ένα δίκτυο υπολογιστών σε ένα άλλο [1]. Η κύρια λειτουργία ενός firewall είναι η ρύθμιση της κυκλοφορίας δεδομένων ανάμεσα σε δύο δίκτυα υπολογιστών. Συνήθως τα δύο αυτά δίκτυα είναι το Διαδίκτυο και το τοπικό/εταιρικό δίκτυο τα οποία έχουν διαφορετικό επίπεδο εμπιστοσύνης. Το διαδίκτυο έχει μικρό βαθμό εμπιστοσύνης (low level of trust), ενώ το εταιρικό δίκτυο ή το δίκτυο ενός σπιτιού διαθέτει τον μέγιστο βαθμό εμπιστοσύνης.



Εικόνα 12: Firewall

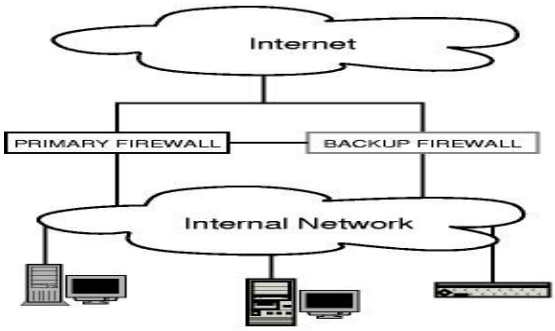
Ο σκοπός της τοποθέτησης ενός firewall είναι η πρόληψη επιθέσεων στο τοπικό δίκτυο και η αντιμετώπισή τους. Παρόλα αυτά όμως, ένα firewall μπορεί να αποδειχθεί μη αποδοτικό εάν δεν ρυθμιστεί σωστά. Η σωστή πρακτική είναι το firewall να ρυθμίζεται με τρόπο ώστε να απορρίπτει όλες τις συνδέσεις εκτός αυτών που επιτρέπει ο διαχειριστής του δικτύου (default-deny). Για να ρυθμιστεί σωστά ένα firewall θα πρέπει ο διαχειριστής του δικτύου να έχει μία ολοκληρωμένη εικόνα για τις ανάγκες του δικτύου και επίσης να διαθέτει πολύ καλές γνώσεις πάνω στα δίκτυα υπολογιστών.

Η τεχνολογία του firewall εμφανίστηκε στα τέλη της δεκαετίας του 1980, όταν ακόμη το Διαδίκτυο ήταν σε πρώιμα στάδια. Σήμερα ο όρος αυτός έφτασε να σημαίνει το λογισμικό ή υλικό που παρεμβάλλεται μεταξύ δικτύων υπολογιστών ούτως ώστε να αποτρέψει την διάδοση ιών, δούρειων ίππων και τις επιθέσεις από κακόβουλους χρήστες. Το firewall αναπτύχθηκε ως εξής :

1.8.2.1 1η γενιά - Φίλτρα πακέτων

Το πρώτο ερευνητικό δημοσίευμα πάνω στην τεχνολογία firewall προέκυψε το 1988 όταν οι μηχανικοί της DEC (Digital Equipment Corporation) ανέπτυξαν φίλτρα πακέτων δεδομένων (data packet filters). Τα φίλτρα αυτά θεωρούνται ως η πρώτη γενιά firewall. Η λειτουργία του είναι να διαβάζουν τα πακέτα δεδομένων που διακινούνται από το ένα δίκτυο στο άλλο και, εάν κάποιο πακέτο ταιριάζει με κάποιο συγκεκριμένο κανόνα, τότε το απορρίπτουν. Ο διαχειριστής του δικτύου είναι σε θέση να ορίσει τους κανόνες βάσει των οποίων θα απορρίπτονται τα πακέτα.

Αυτός ο τύπος firewall δεν ενδιαφέρεται για το εάν κάποιο πακέτο ανήκει σε μία σύνδεση, δηλαδή δεν αποθηκεύει πληροφορίες σχετικά με την κατάσταση των διαφόρων συνδέσεων από το ένα δίκτυο στο άλλο (stateless packet filtering). Αντιθέτως, φιλτράρει κάθε πακέτο με βάση την πληροφορία που περιέχεται στο ίδιο το πακέτο (π.χ. διεύθυνση IP προέλευσης, διεύθυνση IP προορισμού, πρωτόκολλο, αριθμός θύρας κοκ). Επειδή τα πρωτόκολλα TCP και UDP χρησιμοποιούν τις ευρέως διαδεδομένες θύρες (Well known ports), ένα firewall πρώτης γενιάς μπορεί να ξεχωρίσει τα πακέτα που αφορούν διάφορες λειτουργίες, όπως για παράδειγμα το email, την μεταφορά αρχείων, την περιήγηση στο Διαδίκτυο κοκ.

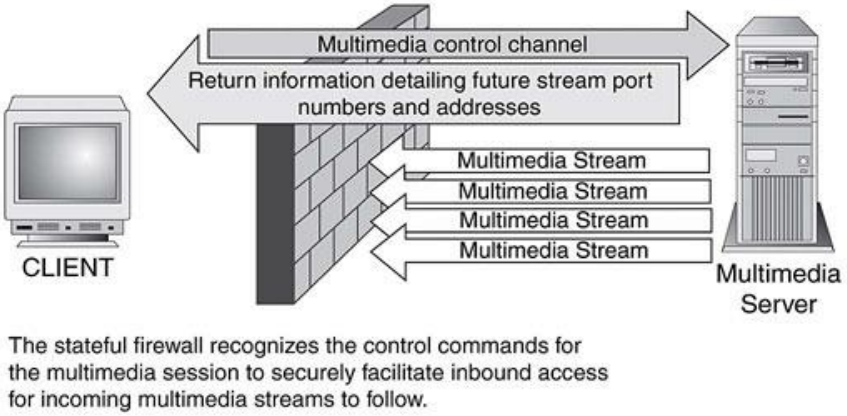


Εικόνα 13: 1η γενιά - Φίλτρα πακέτων

1.8.2.2 2η γενιά - Φίλτρα κατάστασης

Η δεύτερη γενιά firewall αναπτύχθηκε από τρεις ερευνητές στα εργαστήρια της AT&T Bell: Dave Presetto, Howard Trickey και Kshitij Nigam. Τα firewall της δεύτερης γενιάς δρουν όπως τα firewall πρώτης γενιάς με κάποιες επιπρόσθετες λειτουργίες. Μία από αυτές είναι ότι πλέον εξετάζουν και την κατάσταση (state) του κάθε πακέτου, δηλαδή την σύνδεση από την οποία προήλθε. Για τον λόγο αυτό και αναφέρονται ως φίλτρα κατάστασης (stateful firewalls). Τα φίλτρα αυτά κρατούν ανά πάσα στιγμή πληροφορίες για τον αριθμό και το είδος των συνδέσεων μεταξύ των δύο δικτύων και επιπλέον μπορούν να ξεχωρίσουν εάν ένα πακέτο αποτελεί την αρχή ή το τέλος μία νέας σύνδεσης ή μέρος μίας ήδη υπάρχουσας.

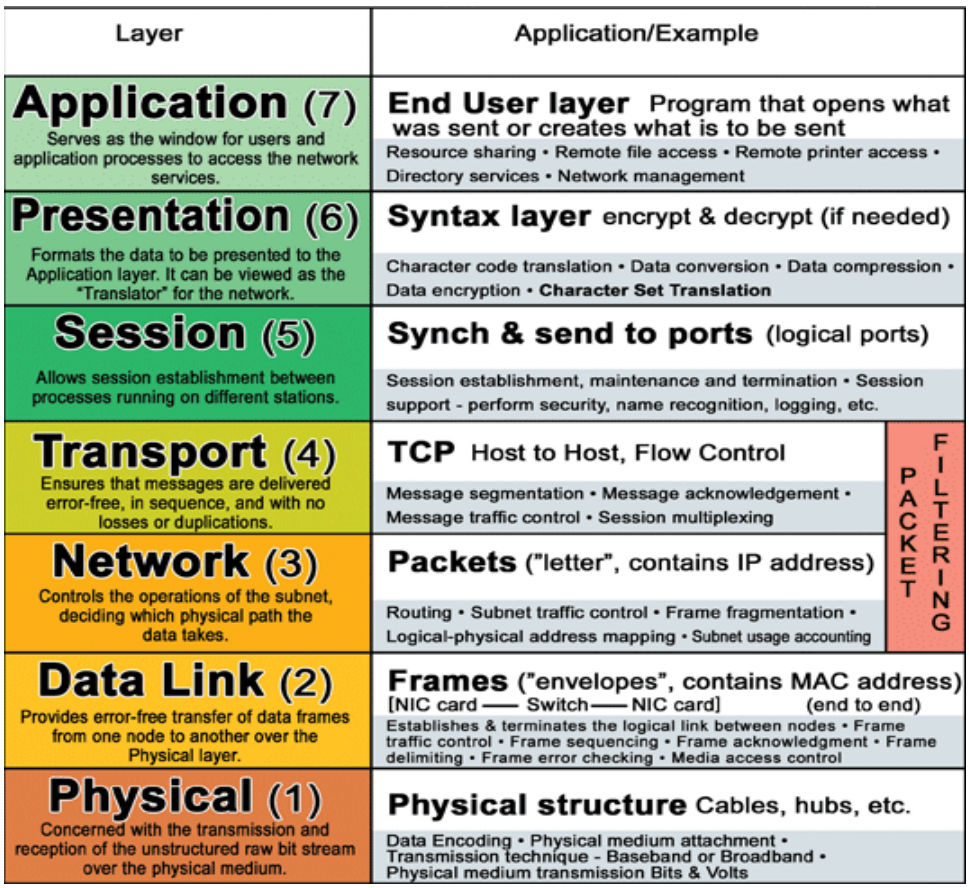
Οι διαχειριστές τέτοιων firewalls μπορούν να ορίσουν τους κανόνες βάσει των οποίων θα επιτρέπεται η δημιουργία συνδέσεων από το εξωτερικό δίκτυο (Διαδίκτυο) προς το τοπικό/εταιρικό δίκτυο. Με τον τρόπο αυτό γίνεται πιο εύκολη η πρόληψη διαφόρων ειδών επιθέσεων, όπως για παράδειγμα ή επίθεση SYN flood.



Εικόνα 14: 2η γενιά - Φίλτρα κατάστασης

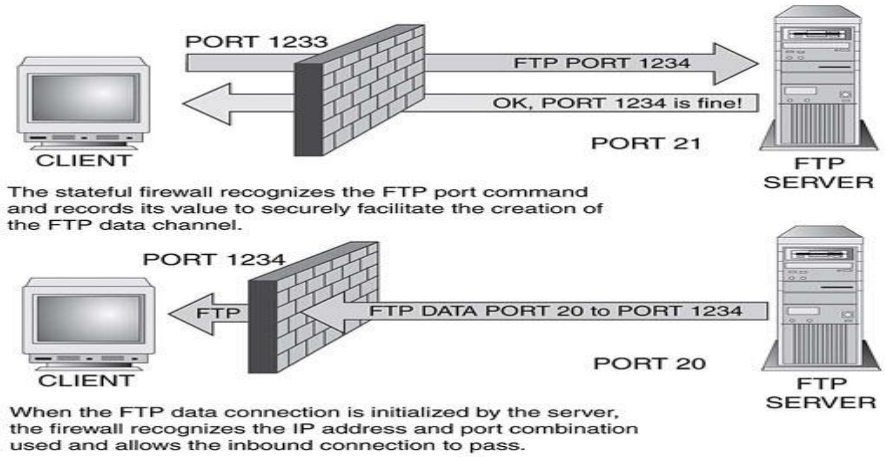
1.8.2.3 3η γενιά – Firewall

Η τρίτη γενιά firewall βασίζεται πλέον στο επίπεδο εφαρμογών σύμφωνα με το μοντέλο αναφοράς OSI (Open Systems Interconnection). Το μοντέλο αναφοράς OSI είναι ένα πρότυπο αναφοράς διασύνδεσης ανοικτών συστημάτων το οποίο χρησιμοποιούμε για να μας βοηθήσει να καταλάβουμε, πώς τα διάφορα δικτυακά "πρωτόκολλα" συνεργάζονται μεταξύ τους.



Εικόνα 15: Open Systems Interconnection

Το κύριο χαρακτηριστικό αυτής της γενιάς firewall είναι ότι μπορεί να αντιλαμβάνεται ποια προγράμματα και πρωτόκολλα προσπαθούν να δημιουργήσουν μία νέα σύνδεση (πχ FTP - File Transfer Protocon, DNS - Domain Name System, περιήγηση στο Διαδίκτυο κοκ). Με τον τρόπο αυτό μπορούν να εντοπιστούν εφαρμογές που προσπαθούν να δημιουργήσουν ανεπιθύμητες συνδέσεις ή καταχρήσεις ενός πρωτοκόλλου ή μίας υπηρεσίας.



Εικόνα 16: 3η γενιά – Firewall

1.8.2.4 Σήμερα (4η γενιά – Firewall):

Σήμερα εδραιώνονται τα firewall 4ης γενιάς, τα οποία διαθέτουν γραφικό περιβάλλον μέσω του οποίου μπορεί ο χρήστης να κάνει τις επιλογές του όσον αφορά την ασφάλεια του δικτύου του και να θέσει τους κανόνες βάσει τον οποίων θα απορρίπτονται κάποια πακέτα ή συνδέσεις. Τα firewall 4ης γενιάς μπορούν πλέον να ενσωματωθούν στο λειτουργικό σύστημα και συνεργάζονται στενά με άλλα συστήματα ασφαλείας, όπως για παράδειγμα το IPS - Intrusion Prevention System.

Με την χρήση firewalls μπορούν να αντιμετωπιστούν αρκετές επιθέσεις και κυρίως επιθέσεις DoS ή DDoS. Επίσης με τα firewalls απαγορεύεται η πρόσβαση σε θύρες και άρα σε εφαρμογές οι οποίες είναι ευάλωτες σε επιθέσεις.

1.8.3 IDS - Σύστημα Ανίχνευσης Επιθέσεων

Το IDS είναι ένα σύστημα λογισμικού, το οποίο πραγματοποιεί την ανίχνευση περίεργης δικτυακής κίνησης σε έναν υπολογιστή ή στο δίκτυο [2]. Εξ' ορισμού ένα IDS απλά ανιχνεύει επιθέσεις και τις παρουσιάζει στο διαχειριστή ασφαλείας του δικτύου. Δεν αναλαμβάνει δράση, υπό την έννοια ότι δεν κάνει κάτι για να περιορίσει την εξάπλωση της επίθεσης. Τα περισσότερα συστήματα ανίχνευσης επιθέσεων λειτουργούν με την χρήση Signatures, δηλαδή ανιχνεύουν παράνομες ακολουθίες ενεργειών, όπως κάνουν και τα περισσότερα antivirus προγράμματα που περιγράψαμε νωρίτερα. Εκτός από την χρήση των Signatures τα IDS συστήματα

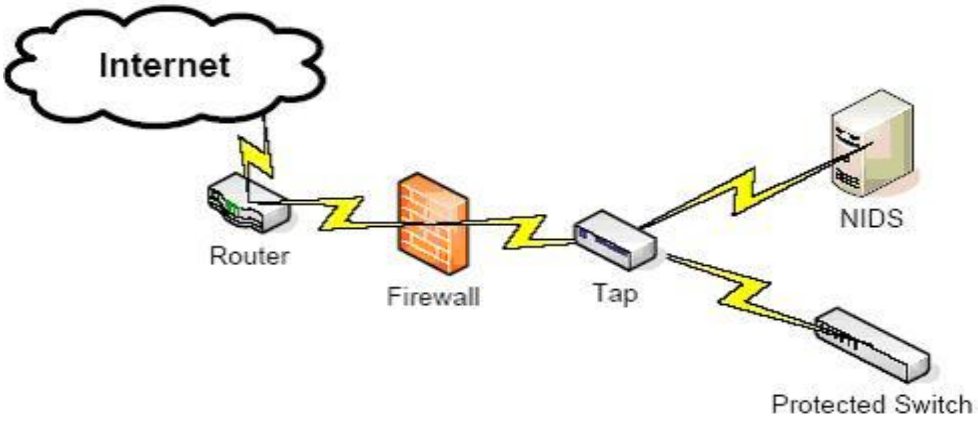
ανιχνεύουν και αναλύουν περίεργη δικτυακή κίνηση και αποφασίζουν αν πρόκειται για κάποιου είδους επίθεση ή όχι.

Τα συστήματα ανίχνευσης επιθέσεων συναντώνται σε δύο τύπους:

- **Network – Based (NIDSs)**
- **Host – Based (HIDSs)**.

1.8.3.1 Network – Based IDS Συστήματα

Τα **Network – Based IDSs** είναι τα πιο διαδεδομένα και εξετάζουν τη διερχόμενη δικτυακή κίνηση (traffic) για ίχνη εισβολής. Στην (εικόνα 18) παρουσιάζεται η διάταξη ενός Network – Based IDS (στην οποία περιέχονται δύο αισθητήρες σε διαφορετικά σημεία ο καθένας, που επικοινωνούν με ένα σταθμό παρακολούθησης δεδομένων στο εσωτερικό δίκτυο).



Εικόνα 17: Διάταξη Network – Based IDS

Το Network – Based IDS συνήθως αποτελείται από δύο μέρη: τους αισθητήρες και τον σταθμό διαχείρισης / ανάλυσης. Ο αισθητήρας βρίσκεται σε ένα τομέα του δικτύου και παρακολουθεί για ύποπτη κίνηση. Ο σταθμός διαχείρισης λαμβάνει τις ενδείξεις κινδύνου από τους αισθητήρες και τις μεταβιβάζει στον διαχειριστή του συστήματος, δηλαδή στον διαχειριστή ασφαλείας του δικτύου. Οι αισθητήρες είναι

συνήθως συστήματα που υπάρχουν μόνο για να παρακολουθούν το δίκτυο. Έχουν ένα δικτυακό interface που αναλύει τα πάντα, δηλαδή λαμβάνουν όλη την δικτυακή κίνηση, όχι μόνο ότι προορίζεται για τη δική τους IP διεύθυνση, αλλά και το διερχόμενο από αυτούς traffic με σκοπό την περαιτέρω ανάλυση. Αν ανιχνεύσουν κάτι ύποπτο το μεταβιβάζουν στον σταθμό διαχείρισης / ανάλυσης. Ο σταθμός διαχείρισης / ανάλυσης μπορεί να δείξει τα σήματα κινδύνου, που έλαβε από τους αισθητήρες ή να πραγματοποιήσει επιπλέον ανάλυση.

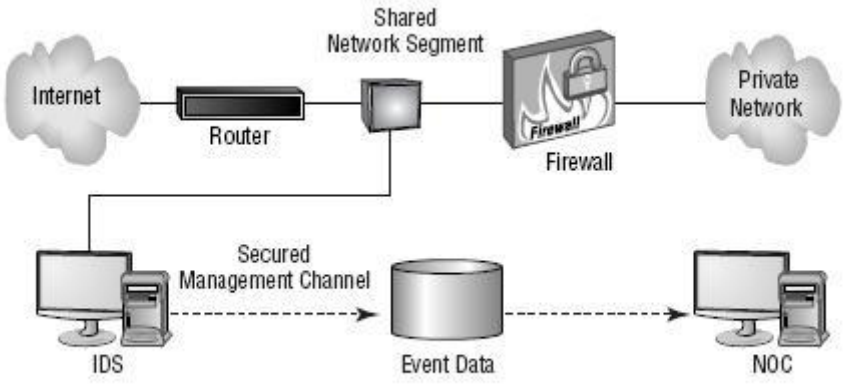
Τα πλεονεκτήματα των **Network – Based IDSs συστημάτων** είναι αρκετά. Μερικά από αυτά είναι ότι μπορούν να ανιχνεύσουν κάποιες από τις επιθέσεις που χρησιμοποιούν το δίκτυο. Επιπλέον έχουν την τάση να είναι καλύτερα διατηρούμενα από ότι τα host – based συστήματα. Αυτό σημαίνει ότι τρέχουν σε ένα συγκεκριμένο σύστημα και η εγκατάστασή τους είναι απλή και πραγματοποιείται σε μια τοποθεσία στο δίκτυο που δίνει τη δυνατότητα παρακολούθησης ευαίσθητης κίνησης δεδομένων, χωρίς εξουσιοδότηση ή κάποιων ειδών πρόσβασης με κατάχρηση προνομίων εξουσιοδότησης. Ένα Network – Based IDS σύστημα δεν απαιτεί μετατροπές στους servers μιας επιχείρησης ή στους hosts για να εγκατασταθεί. Αυτό είναι μεγάλο όφελος, γιατί συνήθως οι servers έχουν κλειστές ανοχές όσο αφορά τη CPU, το I/O και την χωρητικότητα του δίσκου. Τέλος ένα Network – Based IDS σύστημα δεν αποτελεί κρίσιμο παράγοντα για την λειτουργικότητα του δικτύου, και αυτό γιατί δεν λειτουργεί ως δρομολογητής ή ως κάποια άλλη κρίσιμη συσκευή. Άρα, τυχόν αποτυχία στο σύστημα του IDS δε θα έχει σημαντική επίδραση στην επιχείρηση.

Πλην όμως των πλεονεκτημάτων ένα **Network – Based IDSs σύστημα** έχει και κάποια μειονεκτήματα. Μερικά από αυτά είναι ότι απλά εξετάζουν τη δικτυακή σύνδεση στον τομέα που είναι συνδεδεμένο και μόνο εκεί. Δηλαδή δεν μπορούν να ανιχνεύσουν μία επίθεση που γίνεται σε διαφορετικό τμήμα του δικτύου. Το πρόβλημα αυτό γίνεται μεγαλύτερο σε ένα περιβάλλον με πολλαπλές δικτυώσεις ethernet όπου για να καλύψει τις ανάγκες του σε δικτυακή κάλυψη, ένας μεγάλος οργανισμός θα πρέπει να αγοράσει πολλούς αισθητήρες κάτι που σημαίνει επιπλέον κόστος. Επίσης τα Network – Based IDS συστήματα συνήθως χρησιμοποιούν ανάλυση signatures για να καλύψουν τις προδιαγραφές απόδοσης. Έτσι ανιχνεύονται κοινές προγραμματισμένες επιθέσεις από εξωτερικές πηγές, αλλά αυτή η μέθοδος δεν είναι επαρκής για πιο πολύπλοκα είδη επιθέσεων. Επιπλέον ένα σύστημα ανίχνευσης επιθέσεων μπορεί να χρειαστεί να μεταδώσει μεγάλες ποσότητες

δεδομένων στο κεντρικό σύστημα ανάλυσης. Κάποιες φορές αυτό σημαίνει ότι οποιοδήποτε εξεταζόμενο πακέτο παράγει μία μεγαλύτερη ποσότητα κίνησης δεδομένων. Το μειονέκτημα εδώ είναι ότι παρέχεται ελάχιστος συντονισμός μεταξύ των αισθητήρων, δηλαδή οποιοσδήποτε αισθητήρας δεν γνωρίζει αν κάποιος άλλος έχει ανιχνεύσει μια επίθεση. Τέλος ένα Network – Based IDS σύστημα πιθανόν να αντιμετωπίσει δυσκολίες στο χειρισμό επιθέσεων στη διάρκεια κρυπτογραφημένων συνόδων. Ευτυχώς, είναι πολύ λίγες οι επιθέσεις που πραγματοποιούνται εντός μιας κρυπτογραφημένης συνόδου, εκτός από τις επιθέσεις εναντίον ευπαθών Web Servers. Αυτό το γεγονός θα γίνει περισσότερο εμφανές με την μετάβαση στο IPv6.

1.8.3.2 Host – Based IDS Συστήματα

Τα **Host – Based IDSs συστήματα** παρακολουθούν τη δραστηριότητα χρηστών και εφαρμογών στο τοπικό μηχάνημα για ίχνη εισβολής. Αυτού του είδους τα IDSs παρέχουν πιο ακριβή πληροφορία για την ύπαρξη ή μη κάποιας επίθεσης και αυτό γιατί μπορούν να καταλάβουν τι συμβαίνει κάθε φορά στο σύστημα. Έτσι αν συμβεί μια άγνωστης μορφής επίθεση κατά την οποία γίνεται προσπάθεια να επιτευχθεί η δυσλειτουργία του υπολογιστή, το Host – Based IDS θα το αναγνωρίσει ως επίθεση, ενώ αντίθετα το Network – Based IDS δεν θα αντιληφθεί την επίθεση. Γενικά θεωρείται πως η χρήση Host – Based IDS έχει καλύτερα αποτελέσματα από την χρήση Network – Based IDS.



Εικόνα 18: Διάταξη Host – Based IDS

Τα Host – Based IDSs ψάχνουν για ίχνη εισβολής στο τοπικό σύστημα του host. Χρησιμοποιούν συχνά το μηχανισμό ελέγχου και καταγραφής του host σαν πηγή πληροφοριών για ανάλυση. Πιο συγκεκριμένα ψάχνουν για ασυνήθη δραστηριότητα που περιορίζεται στον τοπικό host, όπως logins, παράξενη πρόσβαση σε αρχεία, μη εγκεκριμένη αύξηση δικαιωμάτων ή μετατροπές σε δικαιώματα του συστήματος. Η συγκεκριμένη αρχιτεκτονική χρησιμοποιεί μηχανισμούς βασισμένους σε κανόνες για την ανάλυση της δραστηριότητας. Για παράδειγμα, ένας τέτοιος κανόνας μπορεί να είναι ο εξής: δυνατότητα για πρόσβαση στο λογαριασμό root (διαχειριστή) είναι δυνατή μόνο μέσω της εντολής su. Συνεπώς, επιτυχημένες προσπάθειες πρόσβασης στο λογαριασμό root θα μπορούσαν να θεωρηθούν ως επίθεση.

Τα πλεονεκτήματα ενός Host – Based IDS συστήματος είναι ότι αποτελούν ένα πολύ δυνατό εργαλείο ανάλυσης πιθανών επιθέσεων. Για παράδειγμα, είναι σε θέση μερικές φορές να πουν τι ακριβώς έκανε ο εισβολέας, ποιες εντολές εκτέλεσε, ποια αρχεία έτρεξε και ποιες ρουτίνες του συστήματος κάλεσε αντί για μια αόριστη υπόθεση ότι προσπάθησε να εκτελέσει μια επικίνδυνη εντολή. Άρα τα Host – Based IDSs συνήθως παρέχουν πολύ πιο λεπτομερείς και σχετικές πληροφορίες από ότι τα Network – Based IDSs. Επιπλέον τα Host – Based IDSs συστήματα έχουν μικρότερους false positive ρυθμούς από ότι τα network based. Αυτό συμβαίνει γιατί το εύρος των εντολών που εκτελούνται σε ένα συγκεκριμένο host είναι πολύ πιο εστιασμένο, παρά τα είδη της κίνησης πακέτων που ρέουν σε ένα δίκτυο. Αυτή η ιδιότητα μπορεί να μειώσει την πολυπλοκότητα των Host – Based μηχανισμών. Τα Host – Based IDS συστήματα μπορούν να χρησιμοποιηθούν σε περιβάλλοντα όπου δεν χρειάζεται πλήρης ανίχνευση εισβολών ή όταν δεν υπάρχει διαθέσιμο bandwidth για επικοινωνία αισθητήρα – σταθμού ανάλυσης. Είναι πλήρως αυτοσυντηρούμενα, κάτι που τους επιτρέπει, σε κάποιες περιπτώσεις, να εκτελούνται από read-only μέσα. Έτσι, οι εισβολείς δύσκολα μπορούν να εξουδετερώσουν το IDS. Τέλος, σε ένα Host – Based σύστημα είναι ευκολότερο να σχηματιστεί μία ενεργή αντίδραση σε περίπτωση επίθεσης, όπως ο τερματισμός μιας υπηρεσίας ή το logging off ενός επιτιθέμενου χρήστη.

Ένα σημαντικό μειονέκτημα των Host – Based IDSs συστημάτων είναι ότι απαιτούν εγκατάσταση στο σύστημα που θέλουμε να προστατεύσουμε. Αν για παράδειγμα, έχουμε έναν server που πρέπει να τον προστατέψουμε θα πρέπει να εγκατασταθεί το IDS στον server αυτόν. Όπως αναφέραμε και παραπάνω, αυτό μπορεί να προκαλέσει προβλήματα χωρητικότητας. Σε μερικές περιπτώσεις, αυτό

μπορεί να προκαλέσει και προβλήματα ασφαλείας μιας και το προσωπικό που είναι υπεύθυνο για την ασφάλεια του συστήματος ίσως να μην έχει πρόσβαση στον server όταν χρειαστεί. Ένα άλλο σημαντικό πρόβλημα είναι ότι έχουν την τάση να εξαρτώνται από το υπάρχον σύστημα καταγραφής (logging system) και ελέγχου του server. Εάν ο server δεν λειτουργεί έτσι ώστε η καταγραφή και ο έλεγχος να είναι σε ικανοποιητικό επίπεδο, θα πρέπει να γίνει αλλαγή στις ρυθμίσεις του. Αυτό αποτελεί τεράστιο πρόβλημα αλλαγής στη διαχείριση του server. Επιπλέον αυτά τα συστήματα είναι σχετικά ακριβά. Πολλοί οργανισμοί δεν έχουν την οικονομική δυνατότητα να προστατέψουν ολόκληρα δικτυακά τμήματα με τη χρήση Host – Based IDSs. Έτσι επιλέγουν ποια συστήματα θα προστατέψουν και ποια όχι. Αυτό το γεγονός αφήνει μεγάλα κενά στην κάλυψη της ανίχνευσης εισβολών στο δίκτυο, αφού ένας εισβολέας σε ένα γειτονικό αλλά απροστάτευτο σύστημα μπορεί να υποκλέψει authentication πληροφορίες ή άλλο πολύτιμο υλικό από το δίκτυο. Τέλος, τα Host – Based IDSs είναι πιο ευάλωτα, σε μεγαλύτερο ακόμα βαθμό από τοπικούς περιορισμούς. Αγνοούν εντελώς το περιβάλλον του δικτύου, άρα ο χρόνος ανάλυσης που απαιτείται για την εκτίμηση ζημιών από πιθανή εισβολή αυξάνει γραμμικά με τον αριθμό των host που προστατεύονται.

Όπως αναφέραμε παραπάνω τα συστήματα ανίχνευσης επιθέσεων συναντώνται σε δύο τύπους αλλά υπάρχουν και τρία είδη μηχανισμών με τους οποίους μπορεί κάποιο IDS να αποφασίσει αν υπάρχει επίθεση ή όχι. Οι μηχανισμοί αυτοί είναι οι εξής:

- **Ανάλυση με βάση τα γεγονότα ή υπογραφές (events ή signatures)**

Τα συστήματα που βασίζονται σε events ή signatures λειτουργούν με τρόπο αντίστοιχο με τον τρόπο λειτουργίας των antivirus προγραμμάτων. Η εταιρεία ανάπτυξης του συστήματος IDS παράγει κάθε φορά μια λίστα από «Signatures», δηλαδή μια λίστα με ακολουθίες ενεργειών που θεωρεί ύποπτες ή ενδεικτικές επίθεσης. Το IDS ερευνά και αναλύει το περιβάλλον ελέγχοντας για γνωστές ακολουθίες ενεργειών. Μόλις ανιχνευτεί μια τέτοια ακολουθία, τότε το IDS ενημερώνει τον σταθμό ελέγχου για το συμβάν.

- **Στατιστική ανάλυση**

Τα συστήματα που βασίζονται στην στατιστική ανάλυση κατασκευάζουν στατιστικά πρότυπα για το περιβάλλον τους. Τέτοια πρότυπα μπορεί να είναι η μέση διάρκεια μιας συνόδου, ο μέσος αριθμός βημάτων περιήγησης σε ένα web site, η μέση συχνότητα εμφάνισης μιας IP Διεύθυνσης και άλλα τέτοια. Τα πρότυπα αυτά καθορίζουν αυτό που αποκαλείται «Συνήθης Συμπεριφορά» και κάθε απόκλιση από αυτήν θεωρείται ως ένδειξη περίεργης συμπεριφοράς.

- **Προσαρμόσιμα συστήματα**

Τα προσαρμόσιμα συστήματα ξεκινούν από γενικούς κανόνες για το περιβάλλον και στη συνέχεια μαθαίνουν ή προσαρμόζονται σε τοπικές καταστάσεις που διαφορετικά θα τις θεωρούσαν ασυνήθιστες. Μετά από την αρχική περίοδο μάθησης, το σύστημα καταλαβαίνει την αλληλεπίδραση ανθρώπων – περιβάλλοντος και προειδοποιεί τους υπεύθυνους για ασυνήθιστες δραστηριότητες. Η έρευνα σε αυτόν τον τομέα συνεχίζεται διαρκώς.

1.8.4 Τεχνικές Ανίχνευσης Επιθέσεων

Μια άλλη κατηγοριοποίηση των IDS γίνεται με βάση την τεχνική που χρησιμοποιούν για να ανιχνεύσουν τις εισβολές. Οι εισβολές χωρίζονται σε 6 κατηγορίες:

- Προσπάθεια εισόδου στο σύστημα, που ανιχνεύεται από τυπικά προφίλ συμπεριφοράς ή παραβιάσεις περιορισμών ασφαλείας.
- Κρυφή επίθεση, που ανιχνεύεται επίσης από τα τυπικά προφίλ συμπεριφοράς.
- Διείσδυση στο σύστημα ελέγχου ασφαλείας, η οποία ανιχνεύεται με συνεχή παρακολούθηση συγκεκριμένων προτύπων δραστηριότητας.
- Διαρροή, που γίνεται αντιληπτή με μια τυπική χρήση των πόρων του συστήματος.
- Denial of Service (άρνηση εκτέλεσης εφαρμογής), που επίσης γίνεται αντιληπτή από χρήση πόρων του συστήματος.

- Κακόβουλη χρήση, που ανιχνεύεται μέσω τυπικής συμπεριφοράς προφίλ, παραβιάσεων κανόνων ασφαλείας, ή με χρήση ειδικών προνομίων.

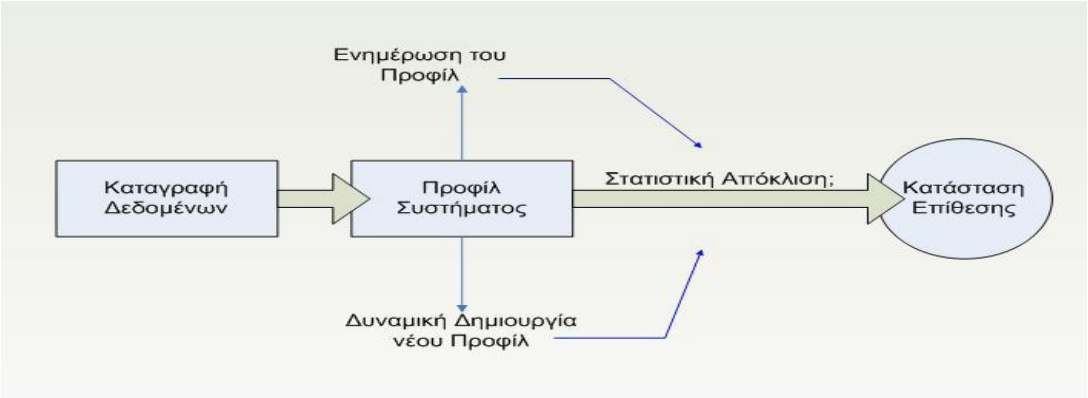
Οι τεχνικές που χρησιμοποιούνται στην ανίχνευση εισβολών χωρίζονται αντίστοιχα σε δύο είδη:

- **Ανίχνευση Διαταραχών (Anomaly Detection)**

Οι τεχνικές ανίχνευσης διαταραχών χαρακτηρίζονται σαν επιθετικές δραστηριότητες με ανωμαλίες. Αυτό σημαίνει ότι αν ορίσουμε ένα “σύνηθες προφίλ δραστηριότητας” για ένα σύστημα, θα μπορούσαμε θεωρητικά να σημαδέψουμε (flag) όλες τις καταστάσεις του συστήματος που αποκλίνουν από το καθιερωμένο προφίλ. Αυτό μπορεί να γίνει με βάση ένα, στατιστικά, σημαντικό νούμερο προσπαθειών εισβολής. Παρόλα αυτά αν το σύνολο των επιθετικών δραστηριοτήτων αλλάξει την κατάσταση του σύνολο των δραστηριοτήτων διαταραχής καταλήγουμε στις

- Ασυνήθεις δραστηριότητες που δεν έχουν χαρακτήρα εισβολής και χαρακτηρίζονται ως επιθετικές και στις
- Επιθετικές δραστηριότητες που δεν είναι ασυνήθεις και καταλήγουν σε false negatives δηλαδή γεγονότα δεν χαρακτηρίζονται ως επιθέσεις, ενώ στην πραγματικότητα είναι.

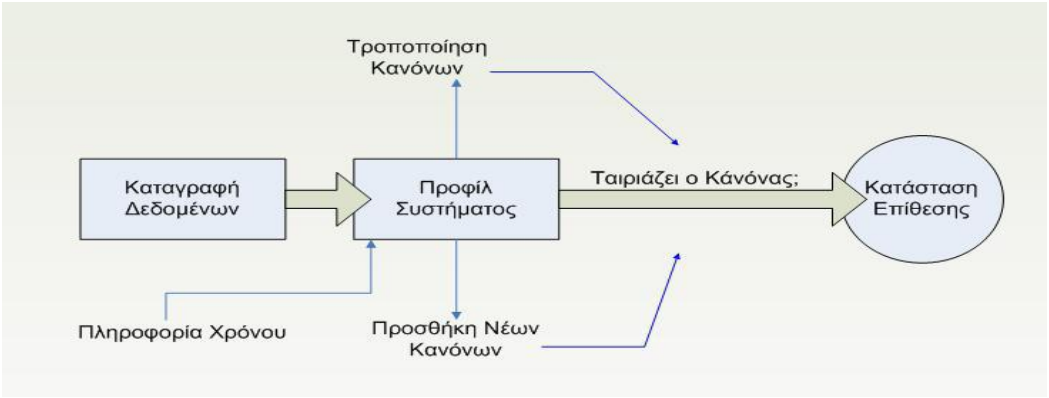
Τα κυριότερα λοιπόν ζητήματα στην ανίχνευση διαταραχών σε συστήματα ανίχνευσης επιθέσεων, είναι να γίνονται οι επιλογές στα επίπεδα των ορίων, ώστε καμία από τις δύο παραπάνω δραστηριότητες να μην μεγιστοποιείται. Σημαντική είναι, επίσης και η επιλογή των χαρακτηριστικών στην παρακολούθηση δεδομένων. Τα συστήματα ανίχνευσης διαταραχών είναι υπολογιστικά ακριβά, λόγω του κόστους του ελέγχου και του της συνεχούς ανανέωσης (updating) των μετρικών του προφίλ ενός συστήματος. Ένα σχηματικό παράδειγμα ενός τυπικού anomaly detection συστήματος είναι το παρακάτω:



Εικόνα 19: Παράδειγμα συστήματος ανίχνευσης διαταραχών

☞ Ανίχνευση Κακής Συμπεριφοράς (Misuse Detection)

Οι τεχνικές ανίχνευσης κακής συμπεριφοράς (misuse detection) σημαίνει ότι υπάρχουν τρόποι αναπαράστασης επιθέσεων με τη μορφή ενός προτύπου ή ενός signature, ώστε ακόμα και παραλλαγές της επίθεσης να μπορούν να ανιχνευτούν. Τα συστήματα αυτά μοιάζουν πολύ με τα antivirus προγράμματα αλλά δεν είναι αποτελεσματικά σε άγνωστες τεχνικές επίθεσης. Τα anomaly detection συστήματα προσπαθούν να μαντέψουν το συμπλήρωμα της «κακής» συμπεριφοράς, ενώ τα misuse detection συστήματα προσπαθούν να αναγνωρίσουν γνωστές «κακές» συμπεριφορές. Το σημαντικότερο ζήτημα στα misuse detection συστήματα είναι το πώς θα δημιουργήσουμε signatures που να περιγράφουν όλες τις πιθανές παραλλαγές μιας σχετικής επίθεσης και πώς θα δημιουργήσουμε signatures που αγνοούν την μη-επιθετική δραστηριότητα. Ένα σχηματικό παράδειγμα ενός τυπικού misuse detection συστήματος είναι το παρακάτω:



Εικόνα 20: Παράδειγμα συστήματος ανίχνευσης «κακής συμπεριφοράς»

1.8.5 Χρησιμότητα IDS

Η χρήση ενός IDS στο δίκτυο παρέχει ένα δεύτερο επίπεδο προστασίας, κάτι που μειώνει ακόμη περισσότερο την πιθανότητα να παραβιαστεί η ασφάλεια ενός συστήματος του δικτύου. Το IDS έχει τη δυνατότητα να ανιχνεύσει 3 ειδών συμβάντα ασφαλείας. Αυτά είναι :

- Τα «κακά» συμβάντα, αυτά που συμβαίνουν κατά τη διάρκεια μιας επίθεσης,
- Συμβάντα κακής ρύθμισης του συστήματος, δηλαδή αυτά που έχουν ως αποτέλεσμα την δυσλειτουργία των συστημάτων και οφείλονται σε λανθασμένη ρύθμιση των παραμέτρων λειτουργίας τους, και
- Τα συμβάντα αναποτελεσματικότητας, όπως αναποτελεσματικής δικτυακής κίνησης

Τα συμβάντα που είναι πιο συχνά και σοβαρά, όπως για παράδειγμα το σκουλήκι (worm), είναι πιο σημαντικό να εντοπιστούν άμεσα από αυτά που είναι πιο σπάνια ή έχουν μικρότερη επίδραση, όπως για παράδειγμα μια σάρωση θυρών από έναν «περίεργο» εργαζόμενο της εταιρείας. Οι πληροφορίες που λαμβάνονται για καθένα από τους παραπάνω τύπους συμβάντων έχουν διαφορετική σημασία για τον διαχειριστή ασφαλείας του δικτύου. Για συμβάντα του πρώτου τύπου λαμβάνονται πληροφορίες που έχουν να κάνουν με τον τρόπο αντιμετώπισης των απειλών σε συστήματα που έχουν ή μπορούν να κυριευτούν από κάποιον κακόβουλο χρήστη. Για συμβάντα του δεύτερου τύπου, ο διαχειριστής λαμβάνει πληροφορίες σχετικές με λανθασμένες ρυθμίσεις συστημάτων και μπορεί έτσι να βοηθηθεί στην σωστή ρύθμιση των παραμέτρων λειτουργίας τους. Για τον τρίτο τύπο συμβάντων ο διαχειριστής λαμβάνει πληροφορίες που τον βοηθούν να βελτιστοποιήσει τον τρόπο λειτουργίας του δικτύου για το οποίο είναι υπεύθυνος.

Συνεπώς με τον μηχανισμό καταγραφής συμβάντων και ειδοποίησης του υπευθύνου ασφαλείας δίνεται η δυνατότητα για άμεση ή έμμεση αντίδραση σε κάθε ειδοποίηση. Αν υπάρχει αυτόματος μηχανισμός αντίδρασης έναντι της όποιας απειλής, για παράδειγμα με την χρήση κάποιου Intrusion Prevention System, τότε η επίθεση αντιμετωπίζεται άμεσα, πριν καν εξαπλωθεί τόσο ώστε να προκαλέσει επιπλέον και μεγαλύτερης έκτασης ζημιές. Σε περίπτωση που δεν υπάρχει τέτοιος μηχανισμός διαθέσιμος, η αντιμετώπιση της απειλής γίνεται από τον υπεύθυνο

ασφαλείας ο οποίος αναλαμβάνει να εκτελέσει τις κατάλληλες ενέργειες για απομάκρυνση του κινδύνου που εμφανίστηκε στο δίκτυο.

Επιπλέον με την χρήση ενός IDS μπορούμε άμεσα να αναγνωρίσουμε λάθη στις ρυθμίσεις ενός δικτύου ή ενός εξυπηρετητή. Με την χρήση ενός IDS μπορούμε να ανιχνεύσουμε περιπτώσεις λανθασμένης δικτυακής κίνησης που οφείλεται σε λανθασμένη παραμετροποίηση των συστημάτων και επομένως να διορθώσουμε τις ρυθμίσεις των συστημάτων. Με τον τρόπο αυτό βελτιστοποιείται η συνολική λειτουργία του δικτύου και προσφέρονται αποδοτικότερες υπηρεσίες. Για παράδειγμα, μπορεί μια συσκευή να έχει αποθηκευμένο ένα λάθος κωδικό πρόσβασης σε αρχεία. Το IDS θα ανιχνεύσει τις αποτυχημένες προσπάθειες σύνδεσης και θα ειδοποιήσει τον διαχειριστή για το συμβάν. Αυτός με τη σειρά του θα δει που οφείλεται αυτό και θα ρυθμίσει πλέον κατάλληλα την συσκευή. Πολλά αντίστοιχα λάθη μπορούν να διορθωθούν επειδή το IDS θα ανιχνεύσει την μη φυσιολογική δικτυακή κίνηση και τα συμβάντα που αυτή προκαλεί.

Ένα τρίτο όφελος από την χρήση ενός IDS στο δίκτυο είναι ότι μπορεί να βοηθήσει στην βελτιστοποίηση της δικτυακής κίνησης ή τουλάχιστον να προσφέρει στον διαχειριστή του δικτύου μια καλύτερη παρουσίαση του τρόπου λειτουργίας του δικτύου του. Με χρήση της τεχνικής «Ανίχνευσης Διαταραχών» που περιγράφηκε παραπάνω, το IDS δημιουργεί προφίλ «κανονικής» κίνησης και αντιδρά σε ότι αποκλίνει από το προφίλ αυτό. Αυτό, εκτός από το ότι του δίνει την δυνατότητα ανίχνευσης διαταραχών, επιτρέπει στο διαχειριστή να έχει μια άποψη της χρήσης των πόρων και της γενικότερης συμπεριφοράς του δικτύου σε περιόδους κανονικής λειτουργίας του δικτύου. Αυτό επιτρέπει στον διαχειριστή να προβεί σε κατάλληλες ενέργειες βελτίωσης της συνολικής λειτουργίας του δικτύου.

Τέλος ένα βασικό χαρακτηριστικό της χρήσης IDS σε ένα δίκτυο, είναι ότι λειτουργεί ως αποτρεπτικό στοιχείο για τους κακόβουλους χρήστες. Από την στιγμή που γίνει γνωστό πως οι όποιες κακόβουλες ενέργειες ανιχνεύονται και τιμωρούνται, οι χρήστες το σκέφτονται διπλά πριν προβούν σε τέτοιες ενέργειες. Με τον τρόπο αυτό αυξάνεται η ασφάλεια του δικτύου, καθώς θα μειωθεί ο αριθμός των χρηστών που θα «δοκιμάσουν» τις δυνατότητες του συστήματος IDS, απλά και μόνο από φόβο μη γίνουν αντιληπτοί.

Από τα παραπάνω γίνεται σαφές πως παρά το κόστος εγκατάστασης και χρήσης ενός συστήματος IDS στο δίκτυο, τα πλεονεκτήματα είναι τόσα ώστε αντισταθμίζεται το κόστος αυτό και αξίζει να γίνει συχνότερη η εμφάνιση τέτοιων

συστημάτων προστασίας στα δίκτυα. Ωστόσο με την πάροδο του χρόνου, εμφανίστηκαν ολοένα και πιο δυναμικές μέθοδοι ασφαλείας. Μία από αυτές είναι και τα **honeypots**, που με τους κλασικούς όρους θα μπορούσαν να περιγραφούν ως παραπλανητικά συστήματα ανίχνευσης εισβολών.

2 Κεφάλαιο 2: Honeypot – Honeytokens - Honeynet

“If you know the enemy and know yourself you need not fear the results of a hundred battles. If you know yourself but not the enemy, for every victory gained you will also suffer a defeat.”

— Sun Tzu, The Art of War

Δομή του Κεφαλαίου

Σ’ αυτό το κεφάλαιο ορίζεται το honeypot ως έννοια και ακολουθεί μία σύντομη ιστορική αναδρομή. Αναφέρονται οι κατηγοριοποιήσεις γύρω από την τεχνολογία των honeypots και τα βασικά πλεονεκτήματα και μειονεκτήματα της χρήσης των honeypots. Ακολουθεί μία σύντομη αλλά περιεκτική περιγραφή του Honeynet και της δομής του καθώς και των Honeytokens .

2.1 Τι είναι το honeypot

*"**Honeypot** ονομάζεται ένας πόρος πληροφοριακών συστημάτων του οποίου η αξία έγκειται στην μη εξουσιοδοτημένη ή παράνομη χρήση του πόρου αυτού"* [3].

Πιο απλά το honeypot είναι ένα σύνολο πόρων το οποίο εκθέτουμε μέσα στο δίκτυό μας για να του επιτεθούν οι "κακοί hackers", εισβολείς. Μπορούμε να το δούμε σαν ένα σύστημα "θύμα", το οποίο υποστηρίζει συγκεκριμένες υπηρεσίες και σκοπός του είναι να παραπλανήσει τον εισβολέα, υποκρινόμενο ένα υπολογιστή θύμα,

- ☞ έτσι ώστε, οι εισβολείς να του επιτεθούν για να αλληλεπιδράσουν αρνητικά πάνω σε αυτό
- ☞ παράλληλα με την επίθεση των κακών, το honeypot επιχειρεί να εντοπίσει τον εισβολέα/επιτιθέμενο, να καταγράψει τις ενέργειες του και να γνωστοποιήσει τις τεχνικές και τα εργαλεία που χρησιμοποιήθηκαν για την εισβολή.

Ακόμη πιο απλά τα Honeypots, «βάζα με μέλι» είναι υπολογιστές οι οποίοι παρακολουθούν ένα σύνολο από IP διευθύνσεις ενός subnet (μία διεύθυνση IP χρησιμοποιείται ως ταυτότητα για έναν ηλεκτρονικό υπολογιστή), οι οποίες δεν χρησιμοποιούνται από χρήστες (π.χ. όταν κάποια εταιρία «αγοράζει» ένα subnet αλλά χρησιμοποιεί πολλή λιγότερες από τις συνολικά 256 διαφορετικές IP διευθύνσεις που αναλογούν σε αυτό). Στην ορολογία των Honeypots, το σύνολο αυτών των διευθύνσεων λέγεται **dark space**. Το dark space υπό κανονικές συνθήκες δεν θα έπρεπε να λαμβάνει κίνηση. Επειδή όμως οι επιτιθέμενοι δεν γνωρίζουν ποιες διευθύνσεις χρησιμοποιούνται και ποιες όχι, επιτίθενται στα τυφλά, ακόμα και στο dark space. Εξ ορισμού λοιπόν ότι κίνηση έρχεται στο dark space είναι ύποπτη και αποκαλύπτει στοιχεία για τη φύση και προέλευση των διαδικτυακών επιθέσεων. Δουλειά των Honeypots είναι να παρακολουθούν το dark space, να αναλύουν την κίνηση που έρχεται σε αυτό και να προσπαθούν να έχουν φυσική αλληλεπίδραση με τους επιτιθέμενους. Να απαντάνε δηλαδή σε αυτά που ζητάνε οι επιτιθέμενοι , όπως ακριβώς θα γινότανε με έναν πραγματικό υπολογιστή. Συνήθως τα Honeypots

τρέχουν ευάλωτες υπηρεσίες, π.χ. παλιές εκδόσεις ενός λειτουργικού συστήματος ώστε να αποτελούν καλύτερο «δόλωμα» για τους επιτιθέμενους και να προσελκύουν ακόμα περισσότερες επιθέσεις.

Τα Honeypots κατά γενική ομολογία είναι δύσκολο να διαμορφωθούν και να συντηρηθούν. Λίγοι οργανισμοί, ερευνητικά κέντρα και ελάχιστες ιδιωτικές εταιρίες τρέχουν Honeypots για το δικό τους dark space. Επιπρόσθετα, τα Honeypots είναι τόσα όσα και το πλήθος των αχρησιμοποίητων διευθύνσεων τις οποίες αναλαμβάνουν. Αν το Honeypot παρακολουθεί 2-3 διευθύνσεις τότε θα δει ελάχιστες επιθέσεις και κατά πάσα πιθανότητα αρκετά αργά σε βάθος χρόνου. Αν όμως παρακολουθεί χιλιάδες ή και εκατομμύρια αχρησιμοποίητες διευθύνσεις, τότε θα βλέπει χιλιάδες επιθέσεις και θα μπορεί να συλλέγει πληρέστερα και γρηγορότερα στοιχεία για το τύπο και τη σοβαρότητα των επιθέσεων αυτών.

2.2 Ιστορία των honeypots

Όπως προαναφέραμε, στα τέλη του 20ου αιώνα δημοσιεύονται τα πρώτα επιστημονικά συγγράμματα και άρθρα τα οποία έθεσαν τα θεμέλια για την ανάπτυξη των honeypots. Η ιστορία των honeypots αρχίζει στα μέσα της δεκαετίας του 1980 και παρουσιάζει ιδιαίτερο ενδιαφέρον. Σημαντικές ερευνητικές και αναπτυξιακές δραστηριότητες πραγματοποιήθηκαν από στρατιωτικούς, κυβερνητικούς και επιχειρηματικούς οργανισμούς. Ωστόσο, ελάχιστες πληροφορίες κοινοποιήθηκαν πριν το 1990 και μόλις πρόσφατα δημοσιεύθηκαν άρθρα και αναπτύχθηκε λογισμικό, τα οποία υποστηρίζουν τη συγκεκριμένη ιδέα. Στα τέλη του 20ου αιώνα, δημοσιεύσεις σχετικές με γεγονότα, ιδέες και έννοιες έθεσαν τα θεμέλια για την ανάπτυξη των honeypots.

Αρχικά το 1989 ο αστρονόμος Clifford Stoll στο βιβλίο του “The Cuckoo’s Egg: Tracking a Spy Through the Maze of Computer Espionage” [4] διηγείται, πώς αντιλήφθηκε την ύπαρξη ενός επιτιθέμενου σε σύστημα της αστρονομικής κοινότητας κατά τη διάρκεια της εργασίας του. Επικεντρώθηκε στη συστηματική παρακολούθησή του και στη συγκέντρωση στοιχείων ικανών για την αναγνώριση της ταυτότητάς του, προστατεύοντας συγχρόνως το σύστημα. Ο Clifford Stoll δε δημιούργησε ένα honeypot αλλά χρησιμοποίησε παραγωγικά συστήματα της ακαδημαϊκής και ερευνητικής κοινότητας με στόχο να δελεάσει τον επιτιθέμενο με τρόπο που

προσέγγιζε την τεχνολογία των honeypots. Η ιδιαιτερότητα του βιβλίου του και επίσης η συμβολή του στην ιστορία των honeypots έγκειται στις ιδέες τις οποίες ανέπτυξε.

Στη συνέχεια το 1992 ο Bill Cheswick στο άρθρο του “An Evening with Berferd in Which a Cracker Is Lured, Endured, and Studied” [5] περιγράφει τη δημιουργία ενός εξειδικευμένου συστήματος προσέλκυσης επιτιθέμενων. Στην ουσία, το άρθρο του αποτελεί την πρώτη τεκμηριωμένη παρουσίαση ενός πραγματικού honeypot. Ο Bill Cheswick δεν αναλύει αποκλειστικά τεχνολογικά θέματα που αφορούν τη δημιουργία ενός honeypot, αλλά αναφέρεται διεξοδικά στην μελέτη της δραστηριότητας του επιτιθέμενου και επίσης στις συνέπειες της δραστηριότητάς του στο σύστημα.

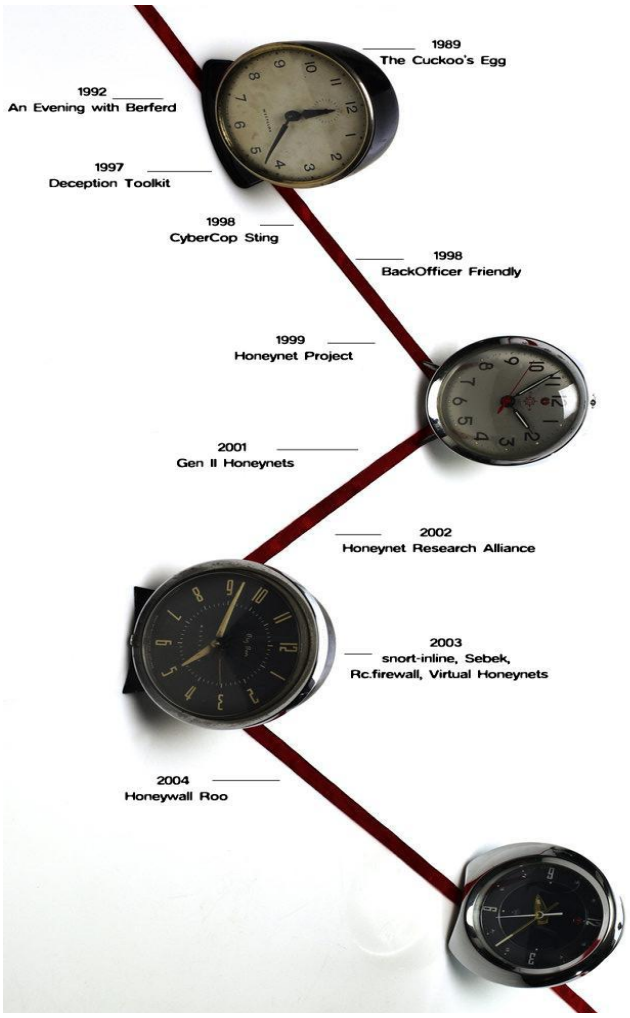
Οι εξελίξεις στο πεδίο της ασφάλειας των δικτύων συνεχίστηκαν με έντονο ρυθμό στη διάρκεια της δεκαετίας του ’90. Πέρα από το θεωρητικό υπόβαθρο το 1997 ήρθε η πρώτη σχετική υλοποίηση με το Deception toolkit (DTK) του Fred Cohen. Αυτό θεωρείται σήμερα και το πρώτο honeypot.

Στην πράξη είναι μία συλλογή από Perl scripts [6] σχεδιασμένα για Unix συστήματα που προσομοιώνουν μία πληθώρα γνωστών αδυναμιών. Η ιδέα του deception toolkit είναι η παραπλάνηση του επιτιθέμενου. Συγκεκραμένα με το toolkit μπορεί να στηθεί ένα σύστημα, το οποίο φέρεται να έχει πολλές γνωστές αδυναμίες. Αυτό επιτυγχάνεται με την αποστολή εξόδου προς τον επιτιθέμενο που μοιάζει αληθινή. Έτσι π.χ. όταν ο κακόβουλος χρήστης στέλνει το γνωστό sendmail exploit, το DTK απαντάει αντίστοιχα κάνοντας τον να νομίζει πως η επίθεση ήταν πράγματι επιτυχής. Με αυτό τον τρόπο ο επιτιθέμενος από τη μία, χάνει πολύτιμο χρόνο ενώ συνάμα οι διαχειριστές καταφέρνουν να καταγράψουν την επίθεση και να απαντήσουν προτού το σύστημα τους δεχτεί κάποια επίθεση που πραγματικά θα λειτουργήσει.

Το 1998-1999 ήρθε και η πρώτη εμπορική λύση στο χώρο των honeypots με το Cybercopsting το οποίο μπορούσε να προσομοιώνει ποικίλες διαφορετικές δικτυακές συσκευές. Την ίδια περίοδο δημιουργήθηκε και το NetFacade το οποίο προσομοίωνε και αυτό δικτυακές συσκευές αλλά σε πολύ μεγαλύτερη κλίμακα (μπορούσε να δημιουργήσει ένα ολόκληρο class-C δίκτυο, 256 συστημάτων). Αν και δε σημείωσε μεγάλη επιτυχία, άφησε πίσω του ένα debugging δικτυακό εργαλείο (του Marty Roesch) το οποίο ουσιαστικά αποτέλεσε και τη βάση για την μετέπειτα δημιουργία του Snort IDS. Το BackOfficer Friendly ήταν το πρώτο Windows honeypot το οποίο αν και δεν ήταν πολύ λειτουργικό διανεμόταν δωρεάν, δημοσιοποιώντας σε μεγάλο βαθμό την άγνωστη μέχρι τότε τεχνολογία των honeypots.

Ιδιαίτερα σημαντική υπήρξε η πραγμάτωση του Honeynet Project από τον Lance Spitzner το 1999 [7]. Ο Lance Spitzner συντέλεσε στη συγκρότηση μίας ομάδας επαγγελματιών της ασφάλειας, οι οποίοι επικεντρώθηκαν στην μελέτη και τη συγκέντρωση εξαιρετικά χρήσιμων πληροφοριών με στόχο τη γνωστοποίησή τους στους ενδιαφερόμενους. Το 2001 δημοσίευσαν το βιβλίο “Know Your Enemy: Learning about Security Threats” [8], του οποίου το περιεχόμενο αναφέρεται στην έρευνά τους και τα αποτελέσματά της.

Η ερευνητική δραστηριότητα στα πλαίσια του Honeynet Project υποστηρίχθηκε από ένα βελτιωμένο και εξελιγμένο είδος honeypot, το οποίο αποτελεί ένα δίκτυο honeypots, ένα honeynet. Η ανάπτυξη των honeypots συνεχίζεται τον 21° αιώνα με γνώμονα τις εξειδικευμένες γνώσεις που αποκτούνται σταδιακά από την παρακολούθηση των επιτιθέμενων όπως πχ το 2003 με το Sebek, το Snort-Inline κ.α. καθώς και την γενικότερη αναβάθμιση των honeypots από σχετικά πολύ απλά εργαλεία σε αρκετά προηγμένα συστήματα [9] [10] [11]. Στην εικόνα 22 θα δούμε και σχηματικά την ιστορία των Honeypots που περιγράψαμε παραπάνω.



Εικόνα 21: Η πορεία των honeypots μέχρι το 2004

Ο σχεδιασμός και η υλοποίησή τους απαιτούν ιδιαίτερη προσοχή, ώστε να συντελέσουν αποτελεσματικά στην προστασία των πληροφοριακών συστημάτων. Είμαστε λοιπόν έτοιμοι να δούμε από τί αποτελείται ένα honeypot.

2.3 Δομή του honeypot

Ένα σύστημα honeypot αποτελείται από **έναν υπολογιστή** (physical device**), το λειτουργικό του** και μια **εφαρμογή** που παίζει τον ρόλο του honeypot. Το λειτουργικό που τρέχει στο physical device μπορεί να είναι οτιδήποτε π.χ. linux, windows 98, όμως η εφαρμογή honeypot να το κάνει να μοιάζει με μια συσκευή win xp ή unix κ.α. . Με απλά λόγια διευθετούμε το honeypot να μοιάζει και να συμπεριφέρεσαι σαν ένα σύστημα με λειτουργικό π.χ. windows 2003 ή να υποκρίνεσαι έναν υπολογιστή unix. Πριν επιχειρηθεί μια επίθεση, ο εισβολέας εξετάζει το στόχο και περισυλλέγει πληροφορίες. Μια από τις απαιτούμενες διαδικασίες είναι και η διαπίστωση του λειτουργικού που τρέχει μια συσκευή. Και αυτό γιατί πριν επιτεθεί σε μια μηχανή θα πρέπει να γνωρίζει με τι έχει να κάνει. Αυτή η διαδικασία ονομάζεται **operating system finger printing**. Αν λοιπόν ο υπολογιστής μας τρέχει linux (πραγματικά) και εμείς έχουμε διευθετήσει το honeypot να υποκρίνεται π.χ μια συσκευή που τρέχει windows srv 2003 τότε στις επιθέσεις τύπου os finger printing το honeypot θα αντιδρά σαν μια συσκευή με win srv 2003. Έτσι το honeypot μας βοηθάει να προβάλουμε την εικόνα οποιοδήποτε λειτουργικού επιθυμούμε εμείς.

Ένα σύστημα honeypot διαθέτει επιπλέον έναν **μηχανισμό καταγραφής δεδομένων**. Στόχος του μηχανισμού καταγραφής δεδομένων είναι αφενός ο εισβολέας όταν διεισδύει μέσα στο honeypot να πέσει πάνω σε αυτόν τον μηχανισμό αφετέρου τα δεδομένα που κυλούν προς το εσωτερικό του honeypot, να καταγραφούν από τον μηχανισμό.

Τέλος διαθέτει ένα **data control** . Το To data control είναι ένα διαφανές στον εισβολέα αντικείμενο το οποίο στηρίζεται στις πληροφορίες που κατέγραψε ο μηχανισμός καταγραφής δεδομένων και αποφασίζει τον τρόπο με τον οποίο θα αντιδράσει, δηλαδή :

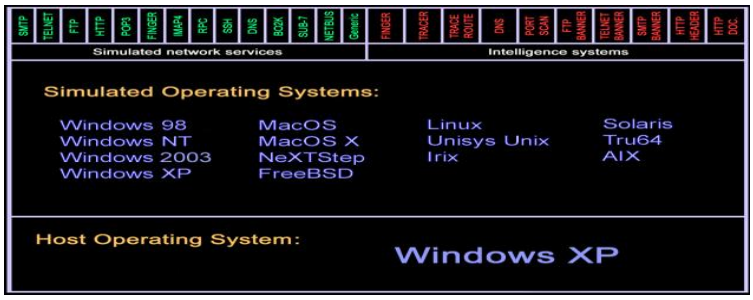
- ποιόν θα ενημερώσει για την επίθεση,
- ποιές πληροφορίες να καταγράψει στα logs,
- και να επιχειρήσει να βρει πληροφορίες για την ταυτότητα του εισβολέα

Η αντίδραση του honeypot εξαρτάται από το πώς το έχουμε ρυθμίσει εμείς προηγουμένως καθώς και η θέση του honeypot εξαρτάται από τον σκοπό που θέλουμε να πετύχουμε, π.χ. αν υποθέσουμε πως έχουμε προβλήματα με τους χρήστες τις δικής μας επιχείρησης, διότι μερικοί από αυτούς παίζουν με port scanning ή ακόμα χειρότερα επιχειρούν να υποκλέψουν δεδομένα τότε θα πρέπει να το τοποθετήσουμε μέσα στο εσωτερικό δίκτυο σε μια θέση που θα είναι προσβάσιμη από όλους τους τοπικούς χρήστες. Έτσι αν το βρουν π.χ. με ένα global scan και το ρωτήσουν αν αυτό διαθέτει ftp υπηρεσία να πάρουν την απάντηση “ναι”, διαθέτω ftp υπηρεσία, και είναι διαθέσιμη. Όταν ρυθμίζουμε ένα honeypot το κάνουμε με τέτοιο τρόπο, επιδιώκοντας να παροτρύνουμε τον εισβολέα να ασχοληθεί αρκετό χρόνο με τον χαρακτηρισμό του “θύματος υπολογιστή” με απώτερο σκοπό την περισυλλογή των δεδομένων, όπως την ταυτότητα του επιτιθέμενου, το πώς σκέφτεται, τί εργαλεία χρησιμοποιεί, για τί πληροφορίες ενδιαφέρεται κ.α..

2.4 Χαρακτηριστικά των Honeypots

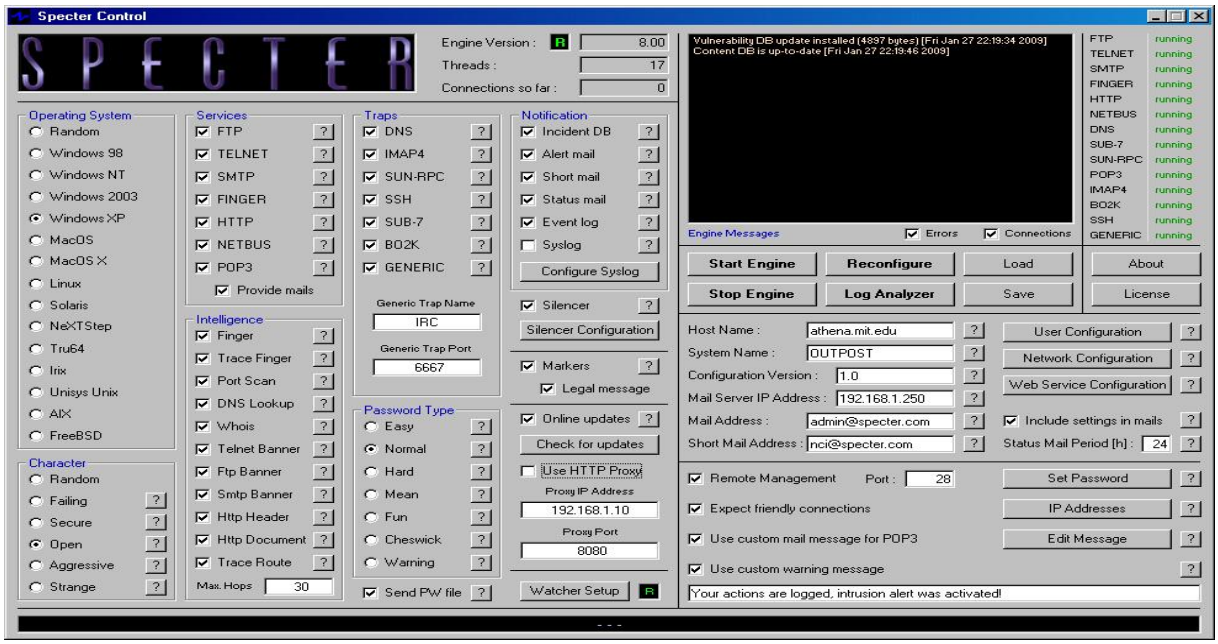
Όπως είπαμε και παραπάνω ανεξάρτητα με το λειτουργικό που τρέχει ένα σύστημα, το honeypot μπορεί να το κάνει να υποκρίνεται με διαφορετικό λειτουργικό. Οι εικόνες που παραθέτω είναι από το www.specter.com. [12].

Το specter (εικόνα 23) είναι μια έξυπνη honeypot εφαρμογή που βασίζεται στην ανίχνευση της εισβολής των επιτιθέμενων. Προσομοιώνει έναν ευάλωτο υπολογιστή, προσφέροντας ένα ενδιαφέρον περιβάλλον με στόχο να παραπλανήσει τους επιτιθέμενους μακριά από τα μηχανήματα παραγωγής. Το Specter προσφέρει κοινές υπηρεσίες Internet, όπως SMTP, FTP, POP3, HTTP και TELNET που φαίνεται απόλυτα φυσιολογικές για τους επιτιθέμενους, αλλά στην πραγματικότητα αποτελούν παγίδα γι αυτούς, μιας και καταγράφει τις κινήσεις και τις ενέργειές τους με σκοπό να ενημερώσει τους διαχειριστές του honeypot.



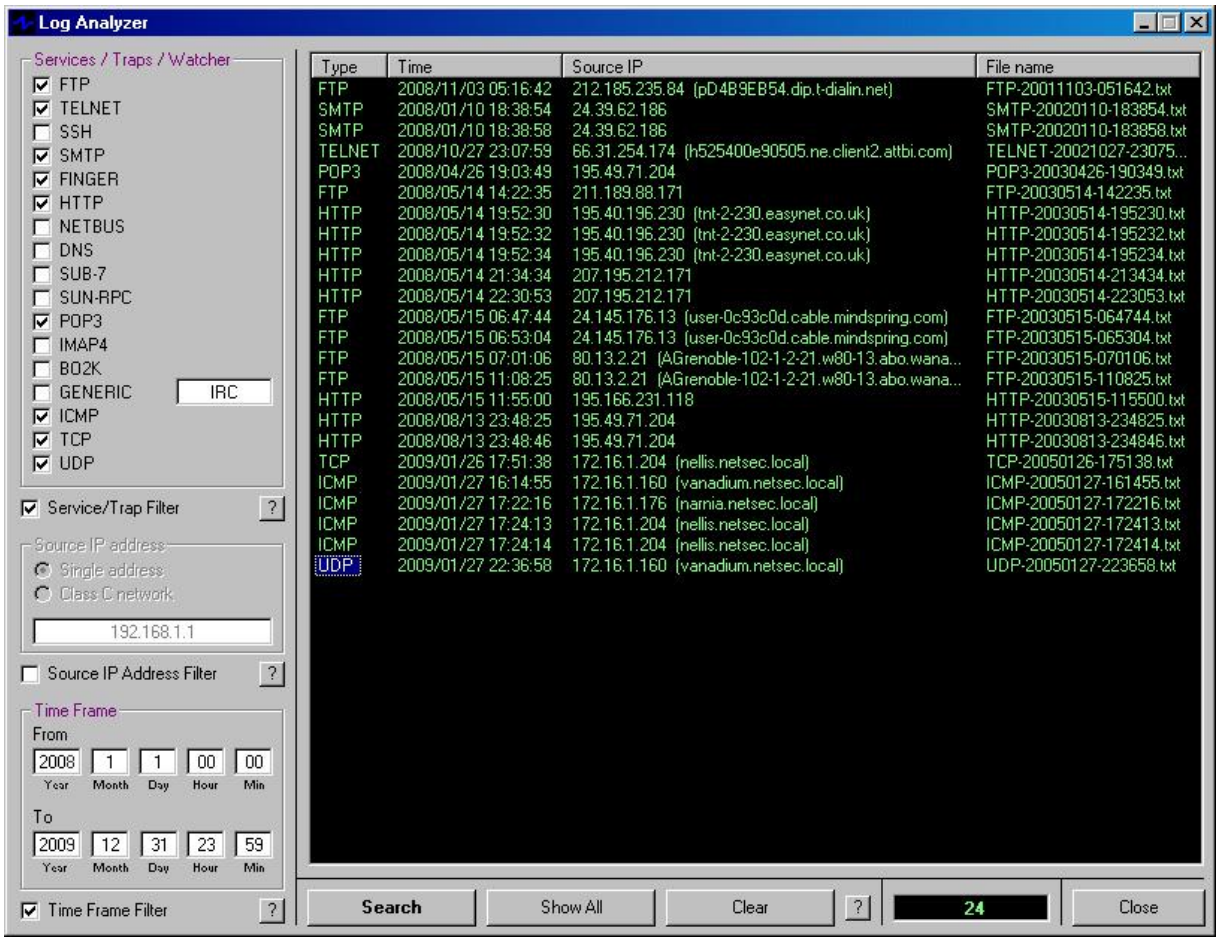
Εικόνα 22: Screenshot of Specter Control

Στην **(εικόνα 24)** βλέπουμε την επιλογή Random του συστήματος



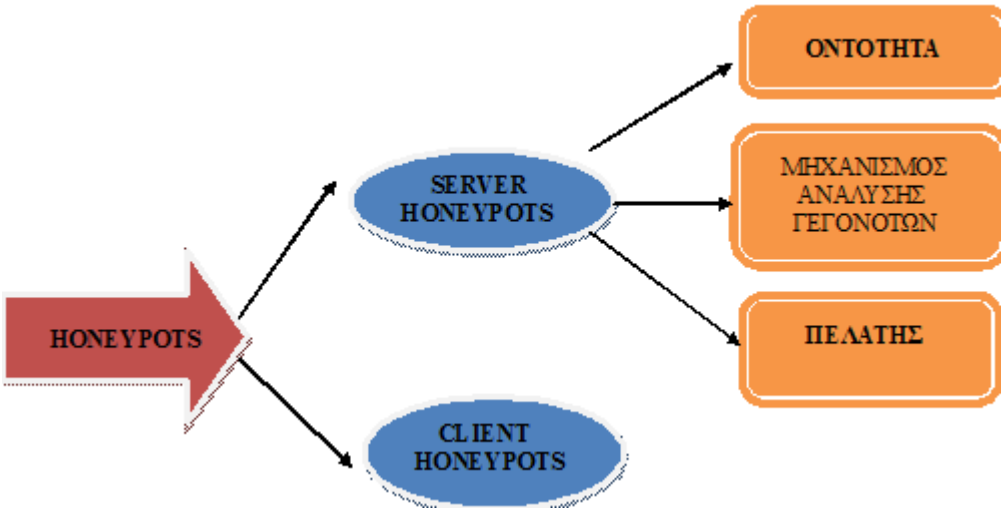
Εικόνα 23: Screenshot of Specter Control

Και κατόπιν πως ρυθμίζετε ο χαρακτήρας του honeypot



Εικόνα 24: Screenshot of Specter Control

2.5 Διακρίσεις Honeypots



Εικόνα 25: Διακρίσεις Honeypots

Τα honeypots (όπως βλέπουμε και παραπάνω εικόνα 26) χωρίζονται σε 2 κατηγορίες :

- **Τα server honeypots** και
- **Τα client honeypots**

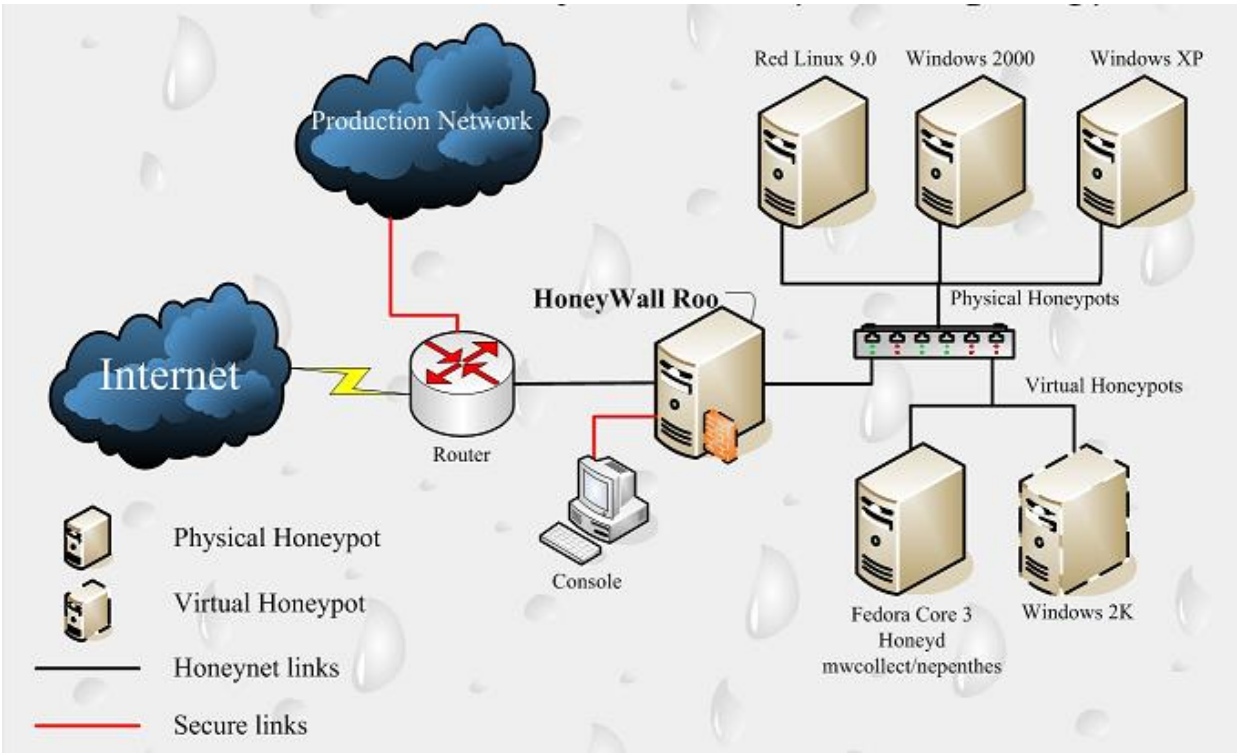
Ανάλογα με το είδος της επίθεσης που δέχονται.

Τα **server honeypots,** βοηθούν στην προστασία των εξυπηρετητών από τις επιθέσεις των εισβολέων και εκμεταλλευόμενοι τις ευπάθειες των υπηρεσιών των εξυπηρετητών, προσπαθούν να παραβιάσουν το σύστημά. Τα **client honeypots** αποτελούν μία σύγχρονη τεχνολογία ασφάλειας, η οποία επιτρέπει την αναγνώριση κακόβουλων εξυπηρετητών που επιτίθενται εναντίον των χρηστών. Ένα client honeypot σύστημα απαιτεί ενεργητική αλληλεπίδραση με έναν εξυπηρετητή. Τα client-honeypots επικοινωνούν με διάφορους εξυπηρετητές και τους διακρίνουν ανάλογα με τη φύση της δραστηριότητάς τους σε καλόβουλους(benign servers) και κακόβουλους εξυπηρετητές (malicious servers).

Τα **client honeypots** αποτελούνται από τρία βασικά συστατικά στοιχεία.

- ☞ Την **οντότητα** που είναι υπεύθυνη για τη δημιουργία μίας λίστας εξυπηρετητών με τους οποίους πρόκειται να επικοινωνήσει ένας πελάτης,
- ☞ τον **πελάτη** που επικοινωνεί με τους εξυπηρετητές του συστήματος και
- ☞ τον **μηχανισμό ανάλυσης γεγονότων** του συστήματος του πελάτη, στη διάρκεια της αλληλεπίδρασης που διακρίνει τους εξυπηρετητές σε καλόβουλους και κακόβουλους.

Επίσης, στα client honeypots εφαρμόζεται κάποια στρατηγική προστασίας, η οποία αποτρέπει οποιαδήποτε επίθεση εκτός των ορίων του. Συνήθως, αυτό επιτυγχάνεται με τη χρήση ενός τοίχους προστασίας (firewall) ή ενός εξειδικευμένου εικονικού συστήματος (virtual machine). Έτσι λοιπόν έχουμε ένα φυσικό (physical) honeypot και ένα εικονικό (virtual) honeypot. Η (εικόνα 27) είναι από το «The Artemis Project» [13] και μας δείχνει τα δύο είδη του honeypot.

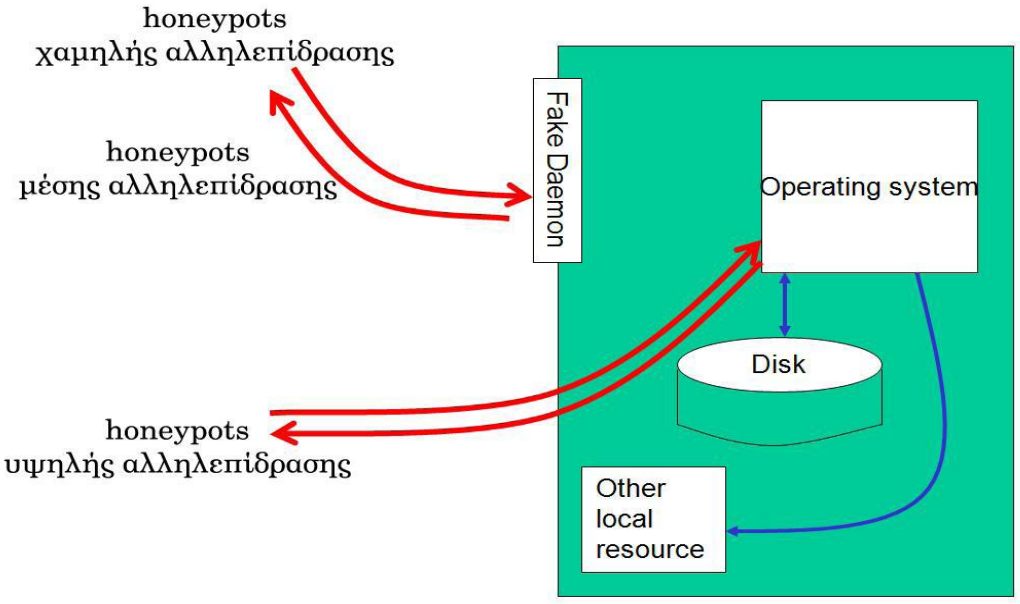


Εικόνα 26: The Artemis Project: Honeynet Topology

Φυσικό (physical), ονομάζεται ένα honeypot που είναι πραγματικό μηχάνημα με τη δική του ip διεύθυνση. Μπορεί να τρέχει οποιοδήποτε λειτουργικό σύστημα -linux, -unix, -windows, -Mac Os, κτλ και οποιαδήποτε υπηρεσία του ορίσουμε πχ -www,- mysql, -ftp. Ένα υπολογιστικό σύστημα μπορεί να φιλοξενεί και μερικά εικονικά (virtual) μηχανήματα, δεν πρόκειται δηλαδή για πραγματικά μηχανήματα αλλά για προσομοίωση συστημάτων σε κάποιον υπολογιστή. Αυτό προσφέρει πολύ ευκολότερη συντήρηση και λιγότερες φυσικές απαιτήσεις.

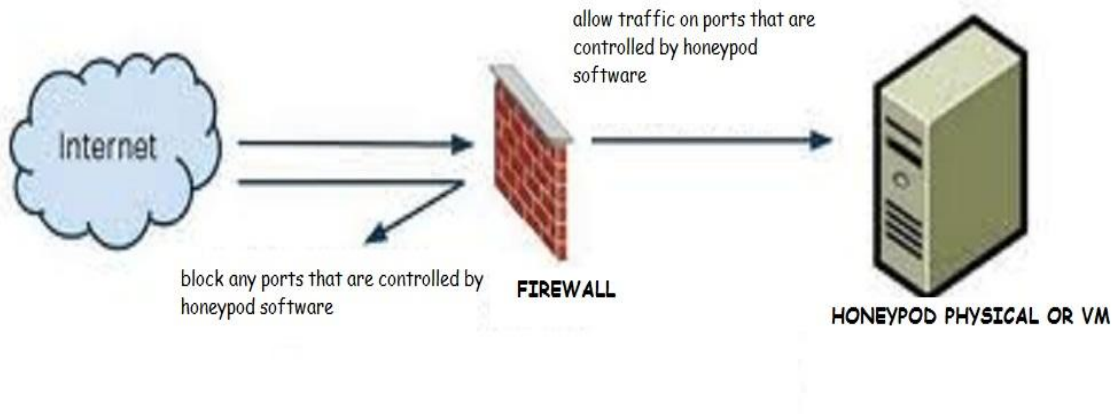
Σαν **εικονικά honeypots** χρησιμοποιούνται το Vmware [14] και το -user-mode linux [15]. Πρόκειται για λογισμικό που επιτρέπει να τρέχουν περισσότερα από ένα λειτουργικά συστήματα σε ένα μηχάνημα και με ένα δυνατό σε ισχύ μηχάνημα μπορεί να τρέχουν αρκετά διαφορετικά λειτουργικά συστήματα, το καθένα από τα οποία θα έχει τη δική του ip και και παράλληλα να μπορεί να δημιουργήσει ακόμα και αυθαίρετες δικτυακές τοπολογίες.

Τέλος το honeypot διακρίνεται ανάλογα με τις τεχνικές εφαρμογές του. Υπάρχουν 3 διαφορετικές εφαρμογές στα honeypots και ο ουσιαστικός παράγοντας που το διακρίνει σε 3 κατηγορίες είναι “το επίπεδο συμμετοχής τους”, δηλαδή τι βαθμό συμμετοχής επιτρέπεται να έχει ένας επιτιθέμενος σε ένα honeypot. Οι κατηγορίες αυτές είναι τα **χαμηλής, μέσης** και **υψηλής αλληλεπίδρασης honeypots** (εικόνα 28).



Εικόνα 27: Επίπεδα αλληλεπίδρασης honeypots

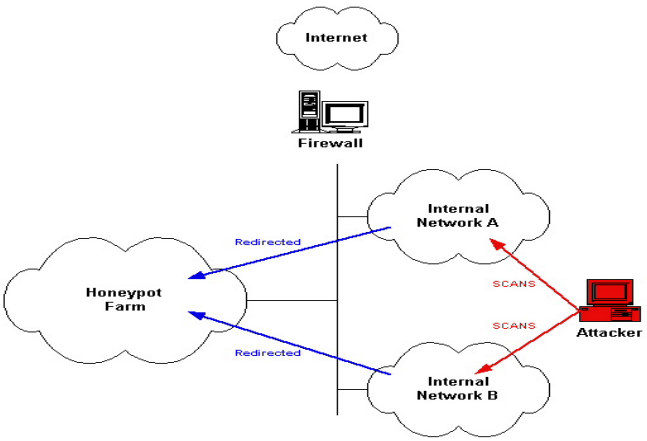
Ένα **χαμηλής αλληλεπίδρασης honeypot** (εικόνα 29) έχει περιορισμένες δυνατότητες, καθώς προσομοιώνει μερικά μόνο μέρη ,π.χ. τη στοίβα δικτύου. Αυτό που κάνει είναι να εξομοιώνει συστήματα και οι δραστηριότητες των επιτιθέμενων περιορίζονται, ως αποτέλεσμα να εκτελούν μόνο κάποιες βασικές εντολές. Δεν μπορεί να γίνει πλήρης συμβιβασμός ”compromise” καθώς δεν πρόκειται για πραγματικό σύστημα με πλήρης εφαρμογές. Οι εξομοιωμένες υπηρεσίες περιορίζονται στο πλήθος των πληροφοριών που μπορούν να συλλάβουν, δεδομένου ότι οι επιτιθέμενοι έχουν όρια ως προς αυτό που μπορούν να κάνουν. Επίσης, οι εξομοιωμένες υπηρεσίες δουλεύουν καλύτερα με γνωστή συμπεριφορά ή αναμενόμενες επιθέσεις. Όταν οι επιτιθέμενοι κάνουν κάτι άγνωστο ή απροσδόκητο, τα χαμηλής-αλληλεπίδρασης honeypots δεν μπορούν να καταλάβουν τις ενέργειες του επιτιθέμενου, ώστε να αποκριθούν κατάλληλα, ή να καταγράφουν τη δραστηριότητα του. Μερικά παραδείγματα χαμηλής-αλληλεπίδρασης **honeypots** είναι το **Honeyd** (στο οποίο θα αναφερθούμε παρακάτω), το **Specter** (που αναφέραμε παραπάνω), και **KFSensor** [16].



Εικόνα 28: Low interaction honeypot

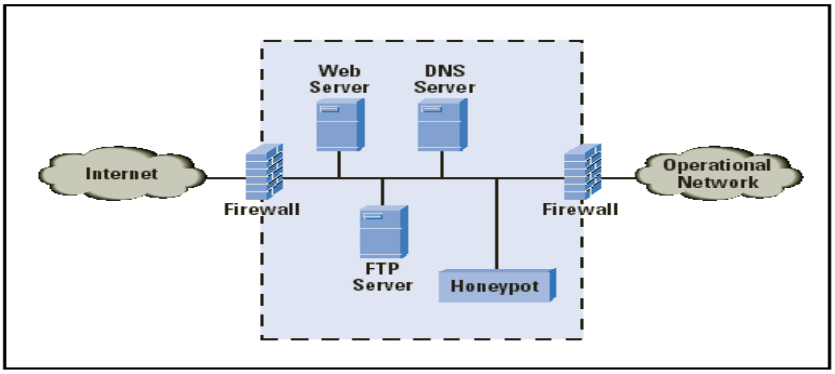
Ένα **μέσης αλληλεπίδρασης honeypot** (εικόνα 30), σε αντίθεση με τα χαμηλής αλληλεπίδρασης honeypot, στέλνει πληροφορίες πίσω στον επιτιθέμενο. Ένα αίτημα λαμβάνει απάντηση από το σύστημα και ο επιτιθέμενος έχει τη δυνατότητα να χρησιμοποιήσει / δημοσιεύσει τις εντολές. Τα μέσης αλληλεπίδρασης honeypots συμμετέχουν απλά και δεν χρησιμοποιούν πραγματικούς δαίμονες “real daemons”, αλλά σενάρια “scripts” ή μικρά προγράμματα για να μιμηθούν τη συμπεριφορά του συστήματος. Η λειτουργικότητα τους εξαρτάται από το σενάριο

“script” και στις περισσότερες περιπτώσεις, οι εντολές είναι πολύ περιορισμένες. Το μεγάλο πλεονέκτημα της χρήσης τέτοιων σεναρίων είναι η καταγραφή των δυνατοτήτων τους και η παράκαμψη πιθανών αδυναμιών των πραγματικών υπηρεσιών.



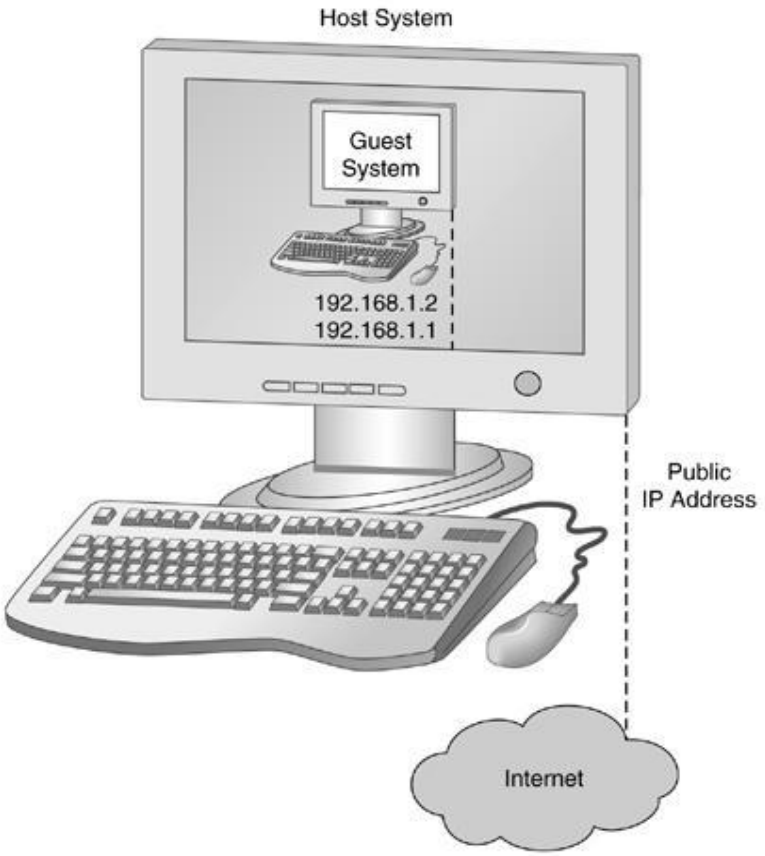
Εικόνα 29: Medium interaction honeypot

Τα **υψηλής δραστηριότητας honeypots** (εικόνα 31) είναι πραγματικοί υπολογιστές με πραγματικές εφαρμογές που δίνουν στους επιτιθέμενους τη δυνατότητα να τα παραβιάσουν και να αποκτήσουν πρόσβαση στο λειτουργικό τους σύστημα. Έτσι έχουν τη δυνατότητα να καταγράψουν μεγάλο όγκο πληροφοριών. Μέσα από ένα παραβιασμένο σύστημα μπορούμε να αποκαλύψουμε τα **rootkits** (μηχανισμοί και τεχνικές κακόβουλων προγραμμάτων) των επιτιθεμένων διότι τα φορτώνουν επάνω στο σύστημα, να αναλύσουμε τις πληκτρολογήσεις τους όταν αλληλεπιδρούν με το σύστημά μας , όταν μιλούν με άλλους επιτιθέμενους και οποιαδήποτε άλλη αλληλεπίδραση/λειτουργία έχουν. Μπορούμε επίσης να μάθουμε τα κίνητρα των επιτιθεμένων, τα επίπεδα ικανότητας, την οργάνωση, και άλλες τέτοιες πληροφορίες. Επίσης τα υψηλής αλληλεπίδρασης honeypots σε σχέση με τα χαμηλής και μέσης αλληλεπίδρασης δεν εξομοιώνουν ως αποτέλεσμα να έχουν ως σκοπό να εντοπίσουν τη νέα , άγνωστη συμπεριφορά.



Εικόνα 30: High interaction honeypot

Μία κλασική λύση στην υλοποίηση υψηλής αλληλεπίδρασης honeypots είναι αυτή των virtual machines (πχ virtual box, vmware κ.α), όπως φαίνεται και στην εικόνα 32.



Εικόνα 31: Παράδειγμα υψηλής αλληλεπίδρασης honeypot μέσω VM

Παρ' όλα αυτά τα υψηλής αλληλεπίδρασης honeypots θέτουν κάποιους κινδύνους αν σκεφτούμε ότι στους επιτιθέμενους παρέχονται πραγματικά λειτουργικά συστήματα για να αλληλεπιδράσουν πάνω σ' αυτά. Αρχικά μπορούν εύκολα να εξαπατηθούν και να καταστρέψουν άλλα μη - honeypots συστήματα θεωρώντας τα ως κακόβουλα. Δεύτερον είναι πολύ σύνθετα. Για να εγκαταστήσουμε ένα honeypot δεν εγκαθιστούμε απλά το λογισμικό αλλά θα πρέπει να χτίσουμε και να διαμορφώσουμε τα πραγματικά συστήματα για τους επιτιθέμενους με σκοπό να αλληλεπιδράσουν με αυτά.

Τέλος διαθέτουν υψηλό βαθμό πολυπλοκότητας γιατί θα πρέπει να ελαχιστοποιήσουμε τον κίνδυνο ζημίας όταν οι επιτιθέμενοι πραγματοποιούν επιθέσεις με άλλα συστήματα.

Ο ακόλουθος πίνακας (πίνακας 1) [17] παρέχει μια επισκόπηση του διαφορετικού τύπου honeypot και του επιπέδου συμμετοχής τους (low- medium- high interaction honeypot) που αναφερθήκαμε παραπάνω.

	Low interaction Honeypot	Medium interaction Honeypot	High interaction Honeypot
Degree of involvement	Low	Mid	High
Real operating system	-	-	x
Risk	Low	Mid	High
Information gathering	Connections	Requests	All
Compromised wished	-	-	x
Knowledge to run	Low	Low	High
Knowledge to develop	Low	High	High
Maintenance time	Low	Low	Very high

Πίνακας 2: "Honeypots" by R. Baumann, C. Plattner

Αν και η ιδέα των honeypots δεν είναι καινούργια, τα honeypots είναι ακόμα σε επίπεδο ανάπτυξης. Ειδικά ο συνδυασμός των honeypots με τα συστήματα ανίχνευσης κινήσεων και τα firewalls μπορούν να δώσουν νέες προοπτικές στον αγώνα κατά της "κοινότητας των hackers/crackers","blackhats community". Είναι πιθανό ότι ο συνδυασμός αυτών των δύο θα μπορούσε να χρησιμοποιηθεί για τη δημιουργία δυναμικών ρόλων ώστε να αποτρέψουν ενεργά τις επιθέσεις πριν αυτές χτυπήσουν / επιτεθούν σε ένα παραγωγικό σύστημα.

2.6 Πλεονεκτήματα και Μειονεκτήματα χρήσης των honeypots

Για να δούμε την σπουδαιότητα της χρήσης των honeypots θα προσπαθήσουμε να παραθέσουμε τα βασικά πλεονεκτήματα και μειονεκτήματα της τεχνολογίας των honeypots [18].

2.6.1 Πλεονεκτήματα

- Χαμηλή ανάγκη πόρων: όπως αναφέραμε και παραπάνω η πλειοψηφία των honeypots (ιδίως τα low και medium interaction) έχουν πολύ χαμηλές απαιτήσεις πόρων. Για παράδειγμα το honeyd (θα αναφερθούμε παρακάτω) μπορεί να δημιουργήσει τεράστια εικονικά δίκτυα (με χιλιάδες διαφορετικές διευθύνσεις IP).
- Απλότητα: Τα περισσότερα εργαλεία είναι απλά και δυναμικά, χωρίς να χρησιμοποιούν τους πολύπλοκους και υψηλούς σε κατανάλωση πόρων αλγορίθμους άλλων τεχνολογιών (πχ IDS).
- Ανακάλυψη νέων απειλών (και μείωση των false negatives): Τα honeypots μπορούν να ανιχνεύσουν νέα είδη επιθέσεων και απειλών. Η οποιαδήποτε δραστηριότητα στο honeypot θεωρείται ανωμαλία, και καταγράφεται.
- False positives: Ένα κλασικό πρόβλημα σε παρόμοιες τεχνολογίες (πχ IDS) είναι αυτό των αυξημένων false positives. Αντίθετα, μιας και η οποιαδήποτε δραστηριότητα ή επικοινωνία με το honeypot θεωρείται μη θεμιτή, ο αριθμός των false positives μειώνεται δραματικά.
- Μικρή ποσότητα όγκου δεδομένων: Τα honeypots μελετούν μόνο την κίνηση που γίνεται προς αυτά, δίχως να λαμβάνουν υπόψη τους πιθανές αυξομειώσεις της δικτυακής κίνησης, ή αν ένα πακέτο είναι legitimate κτλ. Με αυτόν τον τρόπο δεν συλλέγονται τεράστιες ποσότητες δεδομένων, ούτε δημιουργούνται εκατοντάδες alerts, βοηθώντας έτσι κατά πολύ το έργο του διαχειριστή.
- Κρυπτογράφηση: Ακόμη και αν μία επίθεση είναι κρυπτογραφημένη, το honeypot θα την καταγράψει.
- Εσωτερικές απειλές (insiders): Honeypots και honeytokens (θα αναφερθούμε παρακάτω) αποτελούν μία πολύ καλή λύση σε περιπτώσεις οργανισμών που θεωρούν πως έχουν αυξημένη πιθανότητα τέτοιου είδους απειλών.

2.6.2 Μειονεκτήματα

- ☞ Ρίσκο: Ίσως είναι το βασικότερο μειονέκτημα των honeypots. Παρόλο που σε πολλές περιπτώσεις το να καταληφθεί ένα μηχάνημα είναι το ζητούμενο, η πιθανότητα να χρησιμοποιηθεί το σύστημα ως πλατφόρμα επίθεσης προς άλλα δίκτυα παραμένει.
- ☞ Μικρή ποσότητα όγκου δεδομένων: Παρόλο που κατά βάση το γεγονός ότι τα honeypots καταγράφουν μικρή ποσότητα δικτυακών δεδομένων είναι θετικό, σε περιπτώσεις που πιθανόν χρειαζόμαστε μια πιο αναλυτική εικόνα του τι έχει συμβεί κάτι τέτοιο δεν είναι εφικτό. Για το λόγο αυτό συνίσταται η χρήση και άλλων προγραμμάτων καταγραφής της κίνησης (πχ tcpdump, wireshark κ.α).
- ☞ Τεκμηρίωση (Documentation): Η πλειοψηφία των εργαλείων έχουν χαμηλού επιπέδου τεκμηρίωση. Αυτό έχει ως αποτέλεσμα τη μειωμένη ενασχόληση των χρηστών με τέτοιου είδους προγράμματα και τεχνολογίες.
- ☞ Fingerprinting και crackers: Στις περισσότερες των περιπτώσεων ένας επιτιθέμενος με αρκετή εμπειρία μπορεί σχετικά εύκολα να κατανοήσει πως το δίκτυο στο οποίο επιτίθεται δεν είναι πραγματικό.

2.7 Εφαρμογές των Honeypots

Οι χρήσεις των honeypots ποικίλουν. Μερικές από τις πιο σημαντικές εφαρμογές τους είναι οι εξής.

- ☞ Αποκλεισμός επιθέσεων: Αρχικά μπορούν να εμποδίσουν συγκεκριμένες επιθέσεις. Παραδείγματος χάρη υπάρχουν αυτοματοποιημένες επιθέσεις από «σκουλήκια» (worms) οι οποίες σαρώνουν ένα εύρος διευθύνσεων ψάχνοντας για συστήματα με συγκεκριμένες αδυναμίες ασφαλείας (security holes). Όταν βρεθούν τέτοια συστήματα, τα σκουλήκια (worms) επιτίθενται καταλαμβάνοντας το σύστημα αυτό και αφού αντιγραφούν σε αυτό συνεχίζουν την αναζήτηση για άλλα τέτοιου είδους συστήματα. Τα honeypots μπορούν να μειώσουν την ταχύτητα εξάπλωσης αυτών των επιθέσεων με το να καθυστερήσουν όσο το

δυνατόν περισσότερο το σάρωμα που κάνει το σκουλήκι (worm). Κάτι τέτοιο μπορεί να γίνει με το να ρυθμιστεί ένα χαμηλής αλληλεπίδρασης honeypot να ακούει σε ένα μεγάλο αριθμό αχρησιμοποίητων «IP» διευθύνσεις του «Lan» και να απαντάει στις διάφορες σαρώσεις με «Window size of zero» στο «TCP». Χαρακτηριστικό παράδειγμα τέτοιου συστήματος είναι το «LaBrea Tarpit» [19]. Επιπλέον τα honeypots μπορούν να αποτρέψουν επιθέσεις από «hackers». Μπορεί δηλαδή ένα honeypot σωστά εγκατεστημένο να μπερδέψει έναν hacker έτσι ώστε να επιτεθεί σε αυτό και όχι στον πραγματικό υπολογιστή παραγωγής (production server).

- ☞ Μείωση του spamming: Τα honeypots μπορούν να χρησιμοποιηθούν για την μείωση του spamming. Κάτι τέτοιο μπορεί να επιτευχθεί με την χρησιμοποίηση χαμηλής αλληλεπίδρασης honeypots ως «open mail relays» που να καταγράφουν τα συγκεκριμένα «emails» και στην συνέχεια να ενημερώνουν το «spam filter» του «email server». Αντίστοιχα μπορούν να ανιχνεύσουν μια επίθεση. Επειδή, όπως προαναφέραμε, κάθε δικτυακή κίνηση στα honeypots είναι εξορισμού ύποπτη τότε η ανίχνευση οποιασδήποτε κακόβουλης κίνησης ακόμη και επιθέσεων με άγνωστες τεχνικές είναι πρακτικά ανιχνεύσιμη. Τα honeypots συμβάλουν στο να ανταποκριθεί ο διαχειριστής συστημάτων πιο αποτελεσματικά σε μια ενδεχόμενη επίθεση αφού μέσω των honeypots (όταν γίνεται σε αυτά) μπορεί να προσδιορίσει την επίθεση που δέχεται, χωρίς να σταματήσει κάποιο «service» βγάζοντας το από το δίκτυο για να αναλύσει την επίθεση.
- ☞ Ερευνητικό σκοπό: Τέλος μπορούν να χρησιμοποιηθούν για ερευνητικούς σκοπούς. Οι τεχνικές των «hackers» εξελίσσονται συνεχώς και γίνονται πιο καλές και πιο αποτελεσματικές. Το να γνωρίζουμε τις τεχνικές τους είναι αναγκαίο αν θέλουμε να προστατεύσουμε το σύστημά μας αποτελεσματικά. Ακριβώς σε αυτό το σημείο έρχονται τα honeypots τα οποία δίνουν μια λύση για την πιο στενή παρακολούθηση των καινούργιων τεχνικών επίθεσης.

2.8 Honeytokens

Ο όρος των honeytokens [20] εμφανίστηκε το 2003 στο χώρο των honeypots. Παραπάνω αναφέραμε ότι τα honeypots είναι υπολογιστικοί πόροι με τους οποίους

μπορεί να αλληλεπιδράσει ο επιτιθέμενος. Στην πραγματικότητα τα honeypots είναι κάτι γενικότερο.

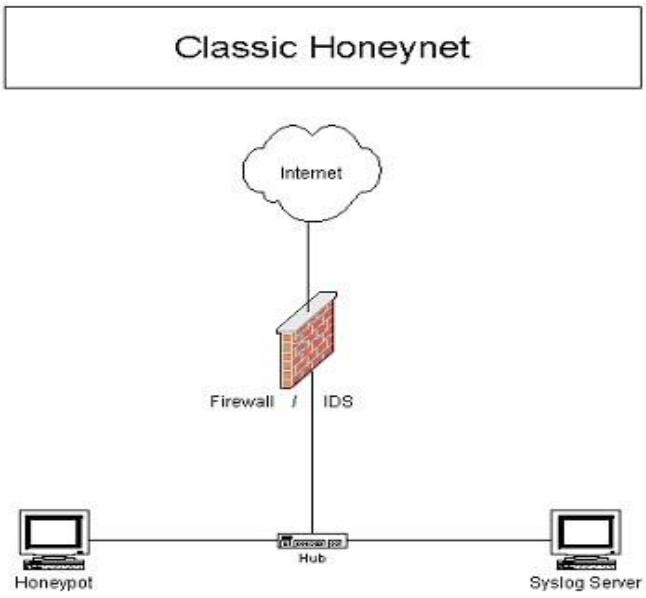
Έτσι λοιπόν ως honeytoken ορίζετε ένα honeypot το οποίο δεν είναι ένας υπολογιστικός πόρος, αλλά οποιουδήποτε είδους ψηφιακή οντότητα. Αν και ως όρος αναφέρθηκε όπως είπαμε για πρώτη φορά το 2003, στην πραγματικότητα δεν πρόκειται για κάτι καινούργιο, αλλά για μία παλιά μέθοδο. Για παράδειγμα οι εκδότες βιβλίων με μαθηματικούς πίνακες (πχ. πίνακες λογαρίθμου και αστρονομικοί ημεροδείκτες) για αιώνες εισήγαγαν μικρά λάθη στους πίνακες τους ώστε να προσδιορίζονται εύκολα τα αντίγραφα. Επίσης ένα παράδειγμα, που θυμίζει honeytoken, είναι αυτό της Google, και της γνωστής της εφαρμογής της Google Earth, όπου υφίσταται μία ολόκληρη πόλη (με το όνομα Argleton), η οποία στην πραγματικότητα δεν υπάρχει (copyright trap) [21].

Εικόνα 32: Παράδειγμα honeytoken

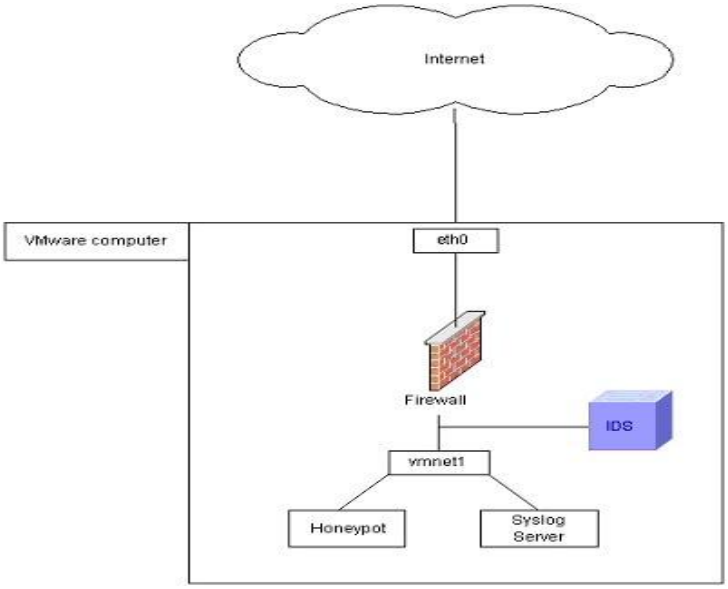
Τέλος τα honeytokens όπως και τα honeypots δεν λύνουν κάποιο πρόβλημα ασφάλειας. Ωστόσο μπορούν να γίνουν ένας έξυπνος τρόπος ελέγχου της ακεραιότητας, να μας προστατέψουν από κακόβουλους insiders/εισβολείς, ή να βοηθήσουν στην έγκαιρη ανίχνευση μη εξουσιοδοτημένης πρόσβασης. Για παράδειγμα η δημιουργία ενός honeytoken λογαριασμού για πρόσβαση (login) μπορεί να βοηθήσει στην παρακολούθηση κακόβουλων ενεργειών.

2.9 Τι είναι το Honeynet

Honeynet ονομάζεται το σύνολο πολλών υψηλής αλληλεπίδρασης honeypots που έχουν τοποθετηθεί σε ένα δίκτυο. Αυτά τα honeypots είναι δυνατό να υπάρχουν είτε ως φυσικά μηχανήματα είτε ως εικονικά. Για παράδειγμα ένα honeynet δίκτυο με φυσικά μηχανήματα φαίνεται στην εικόνα 34 και ένα δίκτυο με εικονικά μηχανήματα στην εικόνα 35.



Εικόνα 34: honeynet δίκτυο με φυσικά μηχανήματα



Εικόνα 35: honeynet δίκτυο με εικονικά μηχανήματα

Ένα honeynet συνήθως αποτελείται από διαφορετικού τύπου honeypots, δηλαδή συστήματα με διαφορετικές υπηρεσίες και λειτουργικά συστήματα, ώστε να συγκεντρώνει ταυτόχρονα δεδομένα από διαφορετικά συστήματα αλλά και να αποτελεί ένα περισσότερο αληθοφανές δίκτυο. Μερικές φορές μάλιστα σχεδιάζεται με τέτοιο τρόπο ώστε να αποτελεί ολοκληρωμένο αντίγραφο δικτύων ή παραγωγικών συστημάτων. Ο στόχος του honeynet είναι να συλλέγει δεδομένα από κάθε δυνατή πηγή, ενώ ταυτόχρονα να προστατεύει το δίκτυο με το να περιορίζει τις κακόβουλες κινήσεις από τα δεσμευμένα honeypots.

Η συλλογή αυτών των δεδομένων μπορεί να γίνει σε διαφορετικά επίπεδα , με σκοπό την καταγραφή περισσότερης πληροφορίας, χωρίς να γίνει αντιληπτή από τον επιτιθέμενο. Τα επίπεδα στα οποία συλλέγουμε την πληροφορία είναι τρία:

- **firewall** και δίνει σαν δεδομένα τα **firewall logs**
- δεδομένα που παίρνουμε από το **IDS** (alerts, warnings , detections) και
- τα δεδομένα από τα honeypots.

Το σύνολο των τεχνικών που υλοποιούνται για την συλλογή των δεδομένων ονομάζεται Data Capture.

Επιπλέον το honeynet εκτός από τη συλλογή δεδομένων συλλέγει και data δηλαδή πραγματοποιεί ελέγχους στο εσωτερικό του. Ο λόγος που το κάνει αυτό είναι

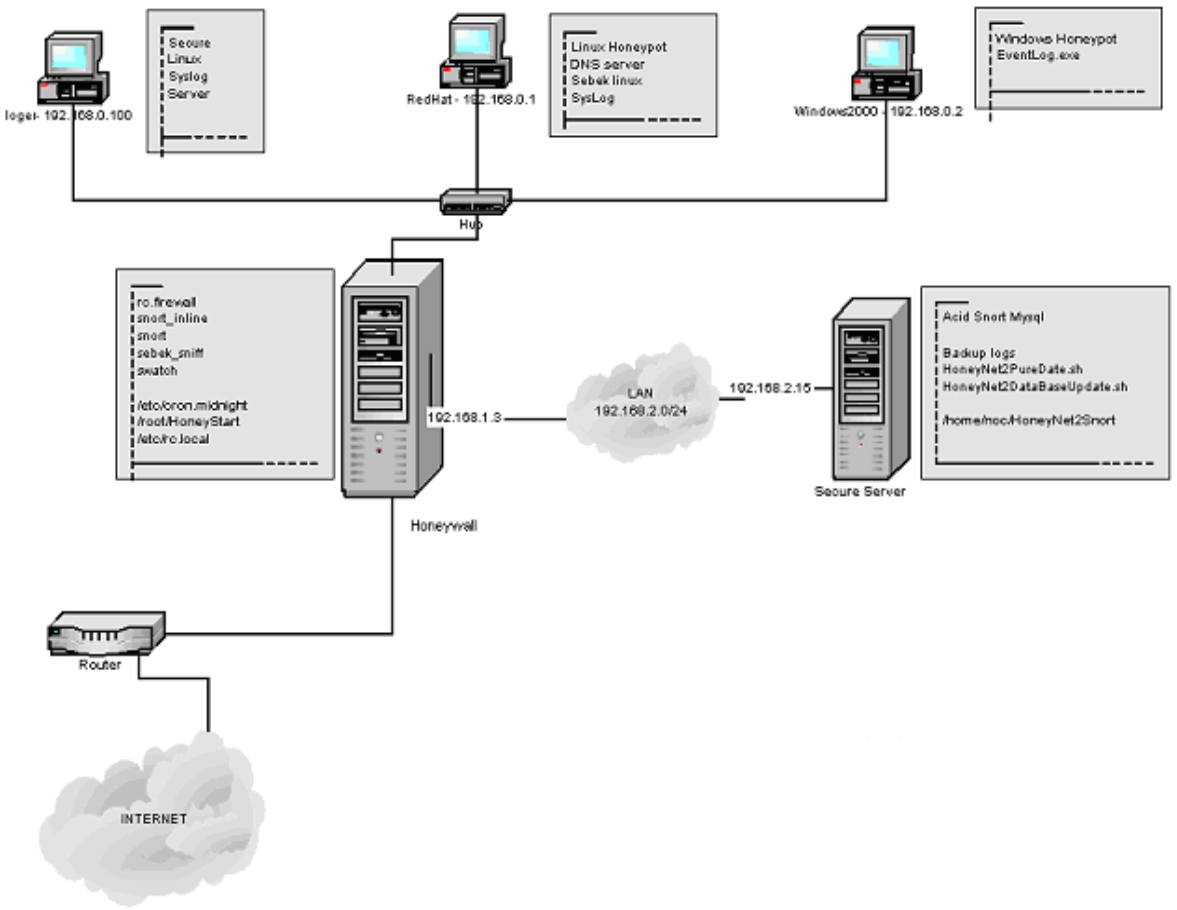
γιατί όταν παραβιαστεί, ο επιτιθέμενος μπορεί να το χρησιμοποιήσει για να επιτεθεί προς το παραγωγικό δίκτυο.Το σύνολο των τεχνικών που υλοποιούνται για τον έλεγχο των δεδομένων ονομάζεται Data Control.

2.10 Δομή του honeynet

Όπως προαναφέραμε σκοπός του honeynet είναι να δημιουργηθεί ένα ελεγχόμενο δίκτυο, δηλαδή ένα δίκτυο που να μπορεί να ελέγχει και να καταγράφει ότι δραστηριότητα γίνεται μέσα σ' αυτό.

Το βασικό στοιχείο της δομής του honeynet είναι η πύλη-honeynet, δηλαδή το honeywall. Σκοπός του honeywall είναι να διασύνδεει το honeynet με το παραγωγικό δίκτυο μας και το διαδίκτυο, αλλά και να το απομονώσει από αυτά. Τα διασύνδεει για να επιτρέψει τις επιθέσεις προς το Honeynet ενώ τα απομονώνει όταν υπάρχει ένδειξη ότι μια επίθεση ξεκινάει από το Honeynet προς άλλα δίκτυα ώστε να τα προστατέψει. Μπροστά από το honeywall είναι τα παραγωγικά συστήματα ενός οργανισμού και ότι υπάρχει πίσω από το honeywall είναι τα συστήματα στόχοι (honeypots). Τα honeypots είναι τα συστήματα με τα οποία αλληλεπιδρούν οι επιτιθέμενοι και το honeywall είναι αυτό που φροντίζει για την καταγραφή των δεδομένων (data capture) από και προς τα συστήματα “θύματα” αλλά και για τον έλεγχο και περιορισμό (data control) των επιθέσεων από τα συστήματα “θύματα” προς το παραγωγικό δίκτυο, έτσι ώστε να μικραίνει η πιθανότητα να επιτευχθεί επίθεση προς το παραγωγικό δίκτυο μέσω των honeypots.

Στην εικόνα 36 περιγράφεται ένα **honeynet**, το οποίο αποτελείται από 3 honeypots, το honewall, ένα hub, ένα router, έναν κεντρικό υπολογιστή, το δίκτυο διαχείρισης του honeynet και φυσικά το Internet.



Εικόνα 36: Παράδειγμα Honeynet

Τα τρία honeypots συνδέονται μέσω του hub. Το hub με τη σειρά του επικοινωνεί και με το honeywall. Δεξιά από το honeywall βρίσκεται το δίκτυο διαχείρισης του honeynet και ο κεντρικός υπολογιστής ο οποίος συγκεντρώνει τα δεδομένα από το honeywall. Στο κάτω μέρος βρίσκεται ο router ο οποίος παρέχει την πρόσβαση στο internet. Στην παρούσα περίπτωση το honeywall χρησιμοποιείται για να χωρίζει τα honeypots από τα παραγωγικά συστήματα και επίσης να καταγράφει την κίνηση των honeypots.

Γενικότερα το Honeywall έχει πολλαπλή χρησιμότητα. Κατ' αρχήν, λειτουργεί σαν bridge (συσκευή OSI επιπέδου 2) ανάμεσα στα honeypots και στο δρομολογητή (router) που παρέχει σύνδεση με το διαδίκτυο. Φυσικά ότι περνάει από το διαδίκτυο προς και από τα honeypots, καταγράφεται. Η σύλληψη των δεδομένων επιτυγχάνεται από το IDS που είναι εγκατεστημένο πάνω στο honeywall. Το honeywall επίσης χρησιμοποιείται ως firewall για να αποτρέπει τις τυχών επιθέσεις από κάποιο παραβιασμένο honeypot προς το υπόλοιπο δίκτυο ή προς το διαδίκτυο.

Τέλος παρέχει δυνατότητα απομακρυσμένης διαχείρισης του Honeywall δυνατότητα αποστολής των alerts μέσω e-mail, να επιτρέπει μεταφορά των δεδομένα που έχουν συλληφθεί σε άλλο σημείο για ανάλυση και λοιπά άλλες όμοιες λειτουργίες.

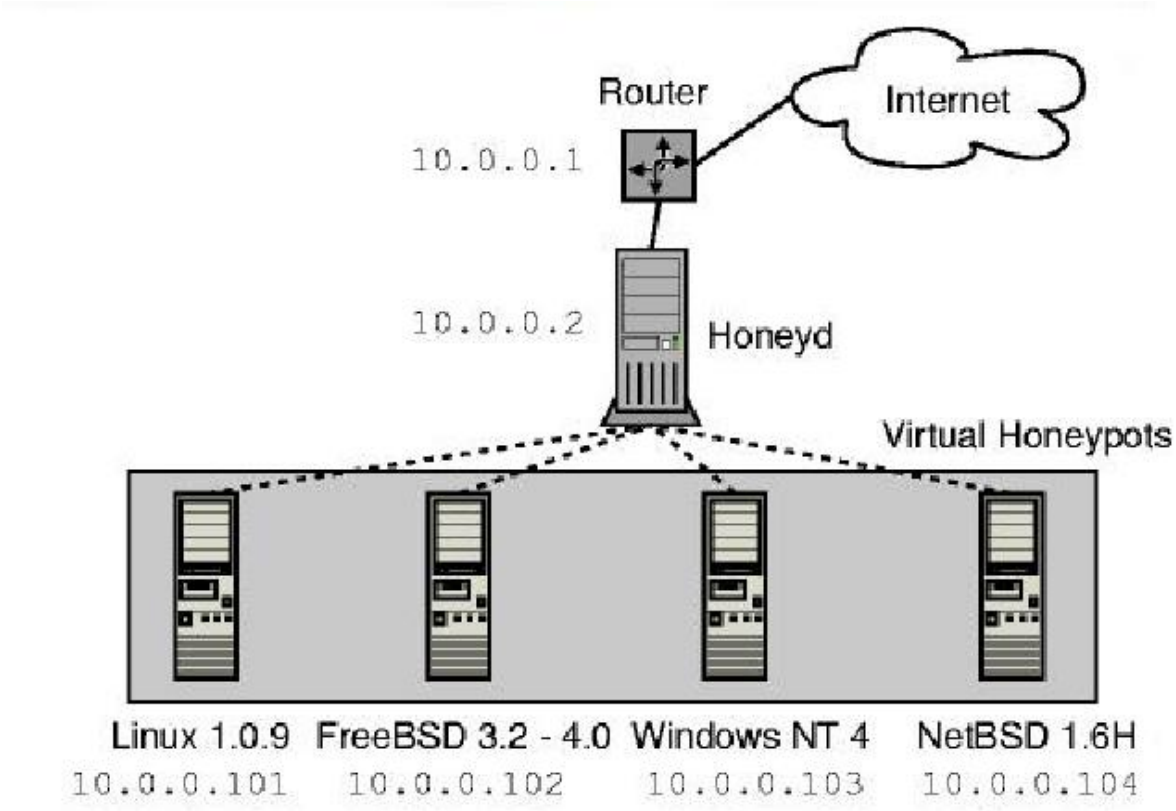
3 Κεφάλαιο 3: Honeyd - Υλοποίηση Honeyd

Δομή του Κεφαλαίου

Σ' αυτό το κεφάλαιο ορίζεται το honeyd ως έννοια και ακολουθεί μία σύντομη περιγραφή της αρχιτεκτονικής σχεδίασής του. Αναλύονται οι βασικές ρυθμίσεις του και η υλοποίηση του πρακτικού μέρους της παρούσας διπλωματικής εργασίας.

3.1 Τι είναι το Honeyd

Το honeyd είναι ένα low interaction honeypot το οποίο έχει τη δυνατότητα να αξιοποιεί χιλιάδες μη δεσμευμένες διευθύνσεις IP, αντιστοιχίζοντας τις σε εικονικά honeypots [22] [23]. Το honeyd είναι δηλαδή ένα μικρό πρόγραμμα το οποίο δημιουργεί virtual hosts σε ένα δίκτυο με το να προσομοιώνει την TCP/IP στοίβα διαφόρων λειτουργικών συστημάτων και μπορεί να ρυθμιστεί για να τρέχει διάφορες υπηρεσίες. Αυτές οι υπηρεσίες είναι συνήθως μικρά scripts που προσομοιώνουν πραγματικές υπηρεσίες όπως το POP3, FTP, HTTP ή SMTP. Έτσι, για κάθε διαφορετική IP έχουμε τη δυνατότητα να ορίσουμε τι είδους σύστημα θέλουμε να προσομοιάσουμε (πχ ένα Windows web server στην πόρτα 80).



Εικόνα 33: Παράδειγμα χρήσης honeyd

Εκτός από virtual hosts το honeyd δημιουργεί και δικτυακές τοπολογίες virtual networks [24] δίνοντάς μας τη δυνατότητα να ορίσουμε χαρακτηριστικά όπως το latency, το packet loss, τα hops έτσι ώστε το virtual network να μοιάζει περισσότερο με ένα πραγματικό δίκτυο.

Έτσι λοιπόν με το honeyd μπορούμε να προσομοιώσουμε πάρα πολλά εικονικά συστήματα, όπως (Linux, Windows NT, FreeBSD κ.α.) λαμβάνοντας μία συμπεριφορά, ανάλογη των ρυθμίσεων που έχουμε κάνει. Επίσης μπορούμε να ρυθμίσουμε αυθαίρετες υπηρεσίες με απλό τρόπο, δηλαδή να ρυθμίσουμε προγράμματα (μέσω ενός απλού αρχείου) που θα αλληλεπιδρούν με τον επιτιθέμενο ανάλογα με το πώς θέλουμε εμείς. Κάθε φορά που το honeyd θα δέχεται μία νέα σύνδεση, να ξεκινά το πρόγραμμα που έχουμε ορίσει για να επικοινωνήσει με τον επιτιθέμενο και ταυτόχρονα να υπάρχει η δυνατότητα να κάνουμε proxy συνδέσεις σε άλλα μηχανήματα ή passive fingerprinting ώστε να αντλήσουμε πληροφορίες για τα απομακρυσμένα μηχανήματα του επιτιθέμενου. Ένα επιπλέον χαρακτηριστικό του honeyd είναι ότι μας επιτρέπει να προσομοιώνουμε λειτουργικά συστήματα στο TCP/IP επίπεδο.

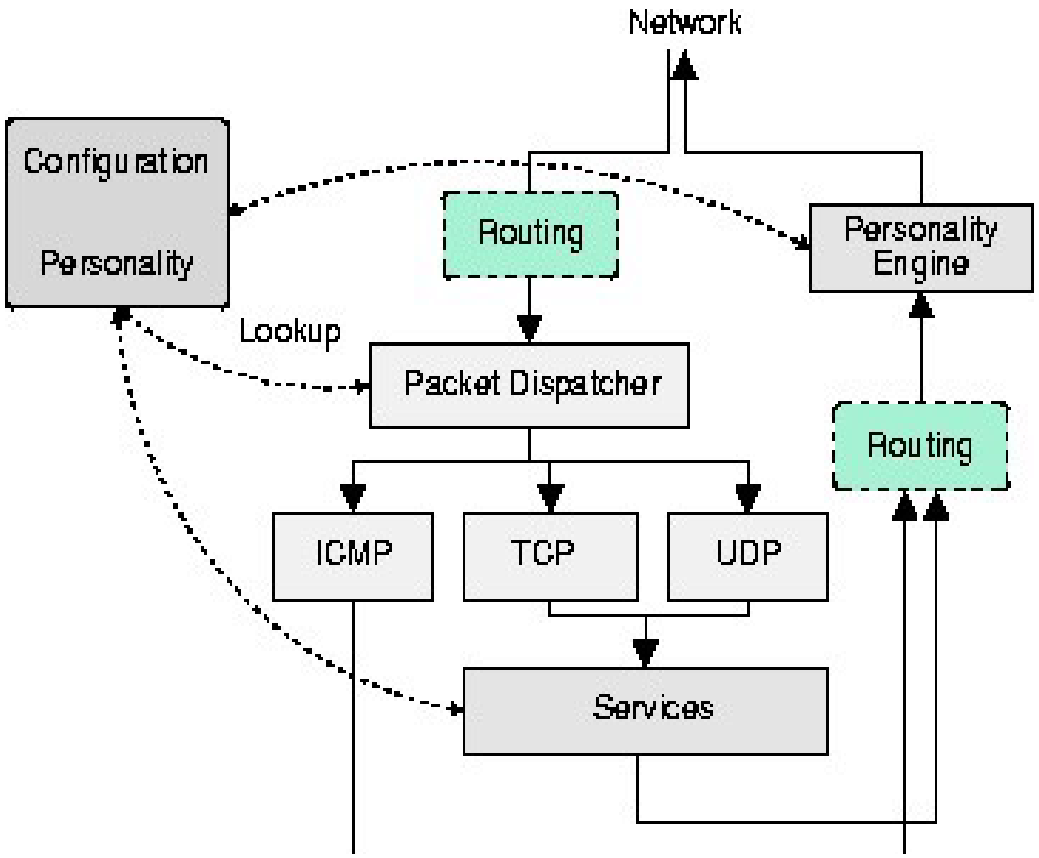
Με αυτό τον τρόπο μπορούμε να εξαπατήσουμε προγράμματα όπως το Nmap και το Xprobe τα οποία (από τη πλευρά δηλαδή του επιτιθέμενου) επιστρέφουν ψευδή αποτελέσματα σε σχέση με το λειτουργικό σύστημα που πράγματι τρέχει στο σύστημα μας.

Με το honeyd μπορούμε επίσης να δημιουργήσουμε εικονικές και πολύπλοκες τοπολογίες δικτύου, με πολλά εικονικά hubs, routers και switches. Μπορούμε να ρυθμίσουμε χαρακτηριστικά όπως το latency, to packet loss και το bandwidth του δικτύου μας. Επιπλέον μιας και το honeyd υποστηρίζει το ασύμμετρο routing μπορούμε να κάνουμε προσθήκη πραγματικών συστημάτων σε μία εικονική τοπολογία δικτύου αλλά και προσθήκη κατανεμημένων λειτουργιών μέσω GRE τούνελ [25].

Τέλος μας δίνει τη δυνατότητα να κάνουμε προσομοίωση υποσυστημάτων δηλαδή να εκτελεστούν πραγματικά Unix προγράμματα στον εικονικό χώρο του honeypot (πχ web servers, ftp servers κα.).

3.2 Αρχιτεκτονική σχεδίασης Honeyd

Η βασική αρχιτεκτονική σχεδίασης του honeyd περιγράφεται στην (εικόνα 35) και αποτελείται από τα εξής στοιχεία: **configuration database, central packet dispatcher, protocol handlers, personality engine**, και τέλος ένα **optional routing component** [26]



Εικόνα 34: αρχιτεκτονική του honeyd

Τον πιο σημαντικό ρόλο τον έχει η personality engine, η οποία τροποποιεί το περιεχόμενο του πακέτου πριν σταλεί στο δίκτυο ώστε να φαίνεται ότι προέρχεται από τη δικτυακή στοίβα του λειτουργικού συστήματος το οποίο έχει ρυθμιστεί να προσομοιώνει ο virtual host. Έτσι λοιπόν η personality engine κάνει τη δικτυακή στοίβα των virtual hosts να μοιάζει αληθινή με το να πραγματοποιεί αλλαγές στις κεφαλίδες των πρωτοκόλλων κάθε εξερχόμενου πακέτου ώστε να μοιάζουν με τα χαρακτηριστικά του ρυθμισμένου λειτουργικού. Η βάση για αυτές τις αλλαγές είναι το fingerprint database του πιο γνωστού port scanner, του nmap [27]. Ένα από τα πρώτα βήματα για κάθε επιτιθέμενο είναι να κάνει ένα port scan στον host στον οποίο θα επιτεθεί, ώστε να δει τι υπηρεσίες τρέχει και ποιο λειτουργικό σύστημα. Οπότε είναι ιδιαίτερα σημαντικό οι honeyd hosts να ξεγελούν έναν επιτιθέμενο και να νομίζει ότι οι hosts είναι κανονικά συστήματα.

3.3 Ρύθμιση του honeyd

Το Honeyd όπως προαναφέραμε είναι ένας μικρός open source daemon για τη δημιουργία virtual hosts σε ένα δίκτυο. Τα virtual hosts μπορούν να ρυθμιστούν να τρέχουν διάφορες υπηρεσίες και η συμπεριφορά τους μπορεί να ρυθμιστεί έτσι ώστε να προσημειώνουν διάφορα λειτουργικά συστήματα. Αυτό που κάνει το honeyd είναι να παίρνει τα network stacks διάφορων λειτουργικών συστημάτων από το fingerprint database του nmap και του Xprobe και με αυτό τον τρόπο καταφέρνει να προσομοιώσει τα αντίστοιχα λειτουργικά σε κάποιον virtual host.

Για τη ρύθμιση του Honeyd πραγματοποιήθηκαν αλλαγές στο αρχείο «honeyd.conf», το οποίο βρίσκεται στον φάκελο εγκατάστασης του. Σε αυτό το αρχείο καταγράφουμε όλη τη δομή που θέλουμε να δημιουργήσουμε σύμφωνα με το συντακτικό και τις εντολές του honeyd που θα περιγραφούν παρακάτω. Επίσης έχουμε τη δυνατότητα να δημιουργήσουμε δεκάδες εικονικά honeypots προσδίδοντας τους διάφορες προσωπικότητες όπως λειτουργικά συστήματα (Windows, Linux κ.α.) ή διάφορες συσκευές (modems, routers, printers κ.α.) και επιπλέον στα honeypots που δημιουργούμε καθορίζουμε τις ανοικτές πόρτες που επιθυμούμε να μελετήσουμε και την απόκριση της συμπεριφορά τους.

Το Honeyd έχει τη δυνατότητα να εξαπατά προγράμματα που βασίζονται στο signature scanning όπως είναι το Nmap και το Xprobe, στέλνοντας κατάλληλες απαντήσεις. Επίσης διαβάζει ένα ειδικό αρχείο με signatures το οποίο καθορίζει την τροποποίηση της TCP/IP στοίβας ανά honeypot και συνεπώς της προσωπικότητας του. Ο χρήστης έχει τη δυνατότητα να επιλέξει το αρχείο signatures υπό την μορφή που ορίζει το NMAP, είτε υπό την μορφή που ορίζει το Xprobe, είτε από τον συνδυασμό και των δύο ταυτοχρόνως. Η επιλογή γίνεται κατά την εκτέλεση της εντολής εκκίνησης του Honeyd. Όταν λοιπόν το Honeyd καταλάβει πως κάποια σάρωση είναι σε εξέλιξη, θα αποκριθεί σύμφωνα με τα signatures που του ορίσαμε και ανάλογα με την προσωπικότητα που έχουμε προσδώσει στο κάθε honeypot. Αντίθετα, ο επιτιθέμενος που εκτελεί το Nmap ή το Xprobe με στόχο κάποιο από τα εικονικά honeypots θα πάρει παρόμοια απάντηση με τον αν στόχευε ένα πραγματικό σύστημα.

Προκειμένου να λειτουργήσουν τα εικονικά honeypots που δημιουργούμε, το Honeyd χρησιμοποιεί μη δεσμευμένες διευθύνσεις IP. Για την εύρεση αυτών των IP

απαραίτητη είναι η λειτουργία της ARPD. Δουλειά της είναι να παρατηρεί την κίνηση για το δίκτυο ή υποδίκτυο που ορίζουμε και να απαντά με τη διεύθυνση MAC του host στον οποίο τρέχει για τις αιτήσεις που γίνονται. Δηλαδή απαντάει στα πακέτα που προορίζονται για ανενεργές IP's σαν να προορίζονταν για τον host στον οποίο τρέχει το arpd [28].

Οι πιο βασικές του εντολές για τη ρύθμιση του honeyd είναι οι εξής:

- **Create:** Η εντολή create είναι το πρώτο βήμα για την δημιουργία εικονικών honeypot στο Honeyd. Με την εντολή αυτή δημιουργούμε ένα template, το οποίο στη συνέχεια συσχετίζουμε με μία διεύθυνση IP αφού του δώσουμε κάποια προσωπικότητα. Η BNF μορφή της εντολής έχει ως εξής:

```
creation := "create" <template-name> | "create default" | "dynamic"
<template-name>
```

Όπως παρατηρούμε διακρίνονται τρεις διαφορετικές χρήσεις της create.

- *create template*: Με την εντολή αυτή δημιουργούμε ένα απλό template. Πχ. Create windows.
- *create default*: Με αυτή την εντολή επιλέγουμε το default template σύστημα που θα χρησιμοποιείται. Έτσι, όλες οι ελεύθερες και μη συσχετιζόμενες με άλλο honeypot, IP διευθύνσεις, να τρέχουν το default template. Για παράδειγμα όταν το honeyd δεχτεί ένα πακέτο για την διεύθυνση 192.168.2.153 να αναζητήσει την αντίστοιχη εγγραφή (template) για το 192.168.2.153. Αν δεν έχουμε ορίσει κάτι συγκεκριμένο θα χρησιμοποιήσει το default.
- *create dynamic*: Εδώ μπορούμε να δημιουργήσουμε ένα δυναμικό template το οποίο και θα λειτουργεί για παράδειγμα συγκεκριμένες ημέρες (ή ώρες) ή που θα δέχεται κίνηση μόνο από συγκεκριμένες διευθύνσεις.

- **Set:** Με την εντολή set μπορούμε να παραμετροποιήσουμε το template το οποίο δημιουργήσαμε με την εντολή create, δηλαδή να συσχετίσουμε το template με κάποια προσωπικότητα από το αρχείο που περιέχει τα Nmap signatures, το nmap.prints. Αφού συσχετίσουμε την προσωπικότητα με το template, το Honeyd αναλαμβάνει να τροποποιήσει όλα τα εξερχόμενα μηνύματα από αυτό το template κατά τέτοιο τρόπο, ώστε να ταιριάζουν με την προσωπικότητα αυτή. Με την εντολή set καθορίζουμε επίσης και την συμπεριφορά των δικτυακών πρωτοκόλλων για το συγκεκριμένο template. Τα πρωτόκολλα αυτά μπορεί να είναι είτε το TCP, είτε το UDP είτε το ICMP. Η BNF μορφή της εντολής έχει ως εξής:

```
set ::= "set" <template-name> "default" <proto> "action" <action> |
        "set" <template-name> "personality" <personality-name> |
        "set" <template-name> "personality" "random" |
        "set" <template-name> "ethernet" <cmd-string> |
        "set" <template-name> "uptime" <seconds> |
        "set" <template-name> "droprate in" <percent> |
        "set" <template-name> "uid" <number> ["gid" <number>]
```

Για να καθορίσουμε την συμπεριφορά τους προσθέτουμε μετά το set μία από τις παρακάτω επιλογές ως εξής:

- **Open**, για να καθορίσουμε πως όλες οι πόρτες θα είναι ανοικτές εξ ορισμού. Αυτή η εντολή επηρεάζει μόνο τα πρωτόκολλα TCP και UDP.
- **Block**, για να καθορίσουμε πως όλα τα πακέτα του αντίστοιχου πρωτοκόλλου θα γίνονται drop εξ ορισμού. Θα λαμβάνονται δηλαδή, αλλά δεν θα στέλνεται καμία απάντηση πίσω στον αποστολέα.
- **Reset**, για να καθορίσουμε πως όλες οι πόρτες θα είναι κλειστές εξ ορισμού. Έτσι όταν το εικονικό honeypot θα λαμβάνει κάποιο SYN πακέτο θα απαντάει με κάποιο TCP RST μήνυμα αν το πακέτο αυτό προορίζεται για κάποια TCP πόρτα και με ένα ICMP port unreachable αν το πακέτο αυτό προορίζεται για κάποια UDP πόρτα.

- **Add:** Η εντολή add παραμετροποιεί τα εικονικά honeypots καθορίζοντας ποιες πόρτες θα μείνουν ανοικτές και προσθέτοντας υπηρεσίες μέσω κάποιον scripts. Η BNF μορφή της εντολής έχει ως εξής:

```
addition ::= "add" template-name proto "port" port-number action |
      "add" template-name "subsystem" cmd-string ["shared"] |
      "add" template-name "use" template-name "if" condition
```

Με την εντολή add:

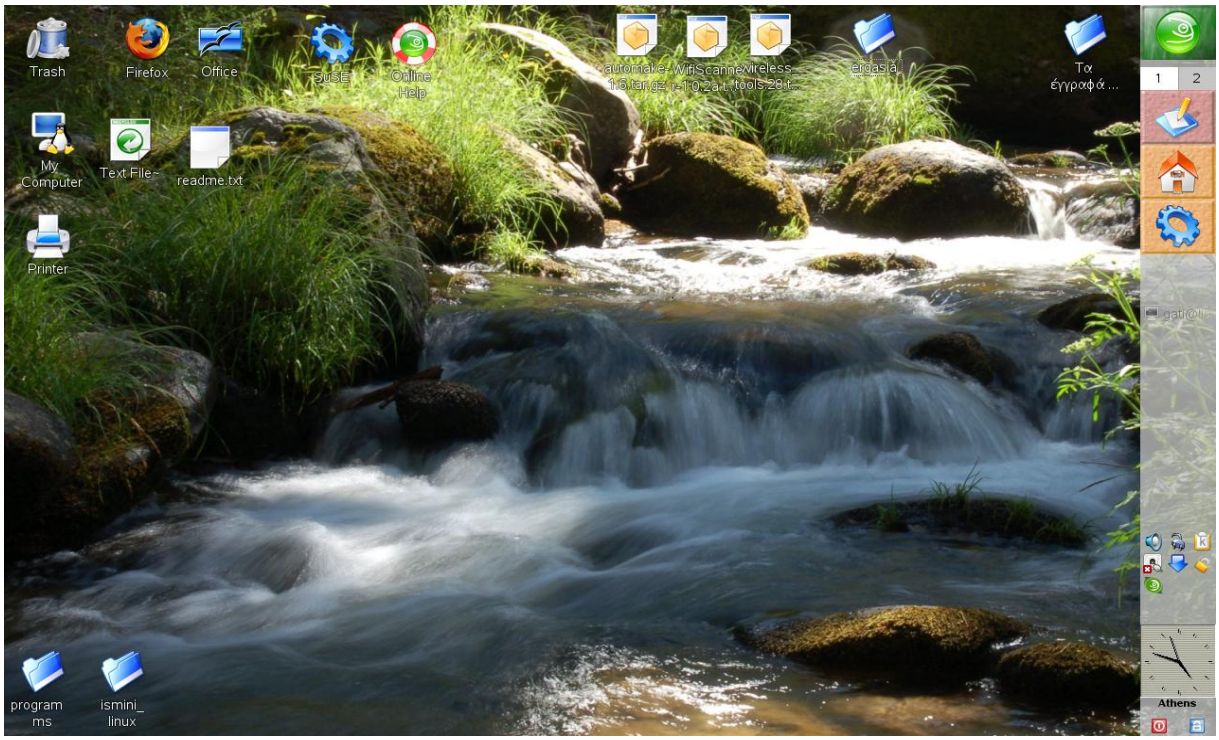
- μπορούμε να καθορίσουμε τη συμπεριφορά που θα έχουν διάφορες πόρτες όπως το ποιες θα παραμείνουν ανοικτές και ποιες κλειστές. π.χ. add default tcp port 25 open
- μπορούμε να καθορίσουμε τα διάφορα scripts που θα υπάρχουν «πίσω» από κάθε πόρτα και θα «απαντάνε» στην εισερχόμενη κίνηση εξομοιώνοντας διάφορες υπηρεσίες. π.χ add linux tcp port 80 /linux/suse8.0/apache.sh
- μπορούμε να ορίσουμε μια TCP ή UDP πόρτα να λειτουργεί ως proxy και να ανακατευθύνει την κίνηση σε κάποιο άλλο honeypot ή ακόμα και στον ίδιο τον αποστολέα. π.χ `add proxy default port 139 proxy $ipsrc:139`

- **bind:** Με την εντολή bind συσχετίζουμε τα εικονικά honeypot τα οποία δημιουργήσαμε με κάποια διεύθυνση IP. Η BNF μορφή της εντολής έχει ως εξής:

```
binding ::= "bind" ip-address template-name |
      "bind" ip-address "to" interface-name |
      "bind" condition ip-address template-name |
      "dhcp" template-name "on" interface-name
      ["ethernet" cmd-string] |
      "clone" template-name template-name
```

3.4 Υλοποίηση του Honeyd

Αρχικά η υλοποίηση του Honeyd πραγματοποιήθηκε σε ένα πραγματικό μηχάνημα, το οποίο έτρεχε με λειτουργικό σύστημα Linux Suse 10.3



Εικόνα 35: Linux Suse 10.3

και ήτανε σχεδιασμένο για χρήση σε διακομιστές αλλά και δυνατότητα γραφικού περιβάλλοντος. Στη συνέχεια εγκαταστάθηκαν οι βιβλιοθήκες libevent, libdnet και libpcap, δίνοντας σαν root

linuxsystem:~ # tar xzf library.tar.gz, για να αποσυμπιέσουμε το zip-αρισμένο αρχείο
linuxsystem:~ # cd directory/, για να μπούμε στο directory που αποσυμπιέστηκαν τα αρχεία
linuxsystem:~ # ./configure; make; make install, για να κάνουμε την εγκατάσταση

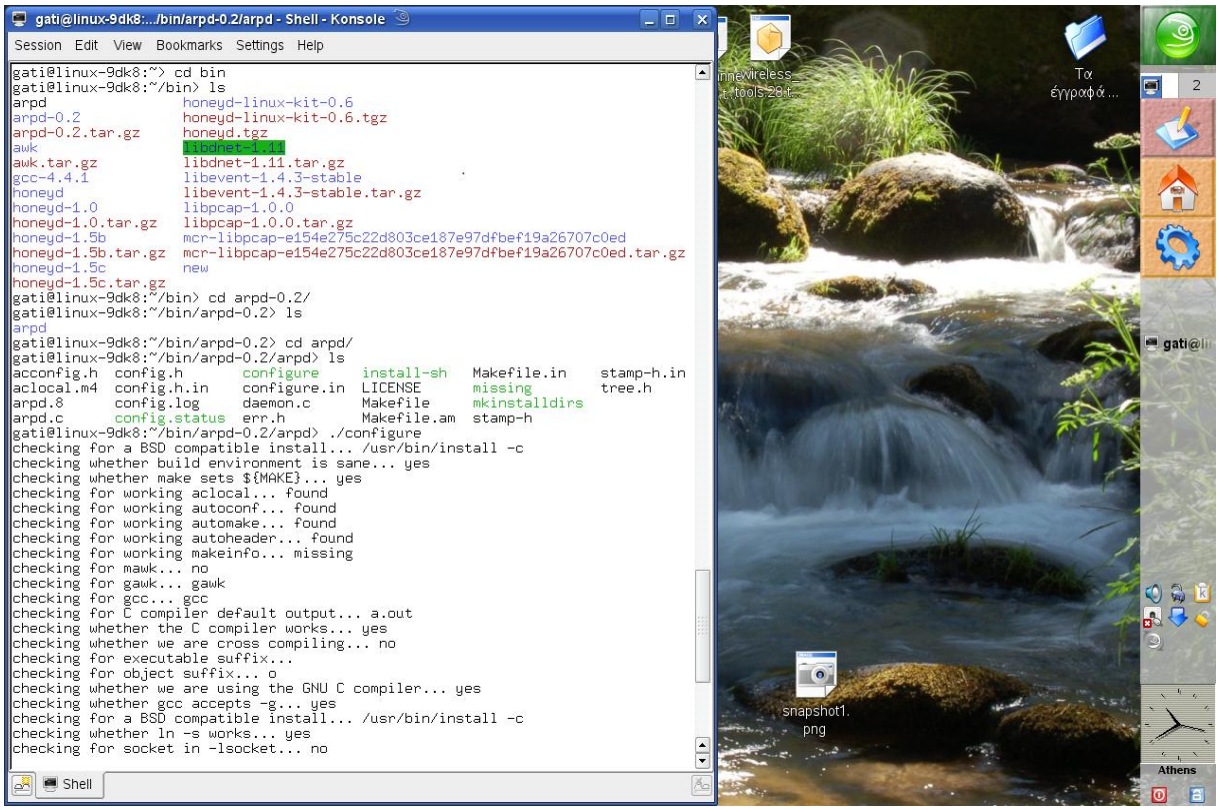
linuxsystem:~ # **ldconfig -m /usr/local/lib,** για να σιγουρευτούμε οτι οι καινούργιες βιβλιοθήκες έχουν προστεθεί στο σύστημα

και έπειτα έγινε εγκατάσταση του Honeyd έκδοσης 1.0 ως εξής

linuxsystem:~ # tar xzf honeyd-1.0.tar.gz

linuxsystem:~ # cd honeyd-1.0/

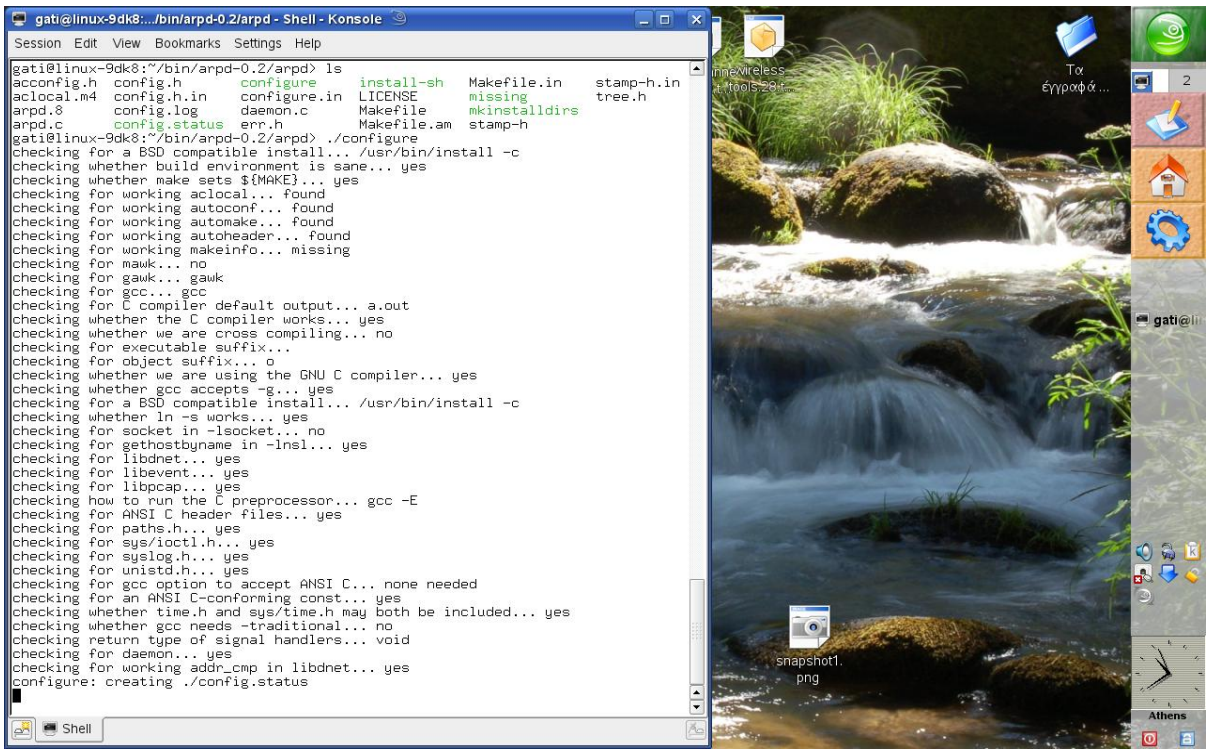
linuxsystem:~ # ./configure; make; make install.



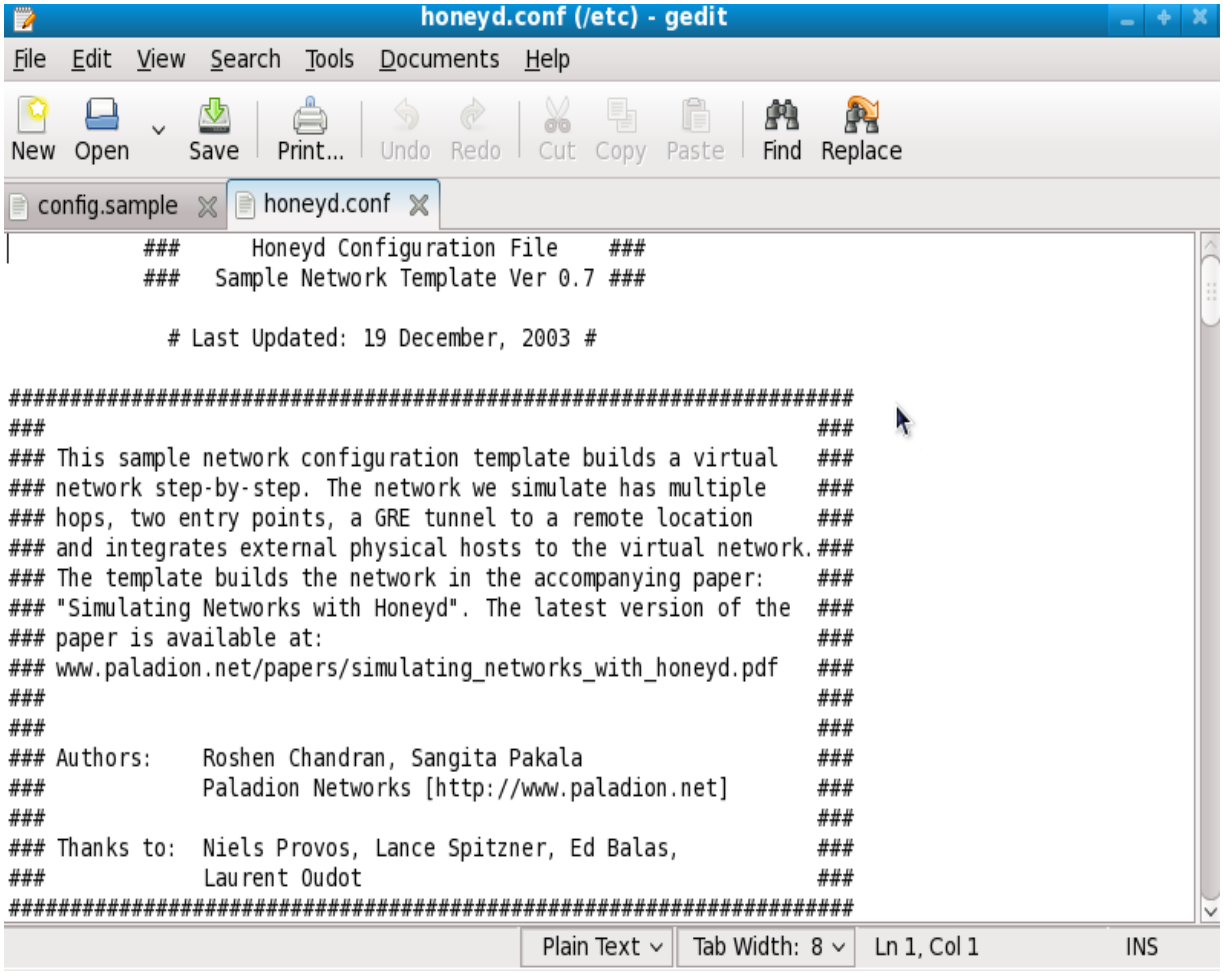
Εικόνα 36: Εγκατάσταση βιβλιοθηκών

Μετά την εγκατάσταση του Honeyd απαραίτητη ήτανε η εγκατάσταση της βιβλιοθήκης arpd η οποία θα αναλάμβανε την δρομολόγηση των πακέτων στους virtual hosts.

Η εγκατάστασή της έγινε από τις αποθήκες του λογισμικού του Suse 10.3 και αντίστοιχα έγινε configuration του honeyd από το αρχείο παραμετροποίησης του Honeyd το “Honeyd.conf” για να πραγματοποιηθούν οι επιθέσεις.



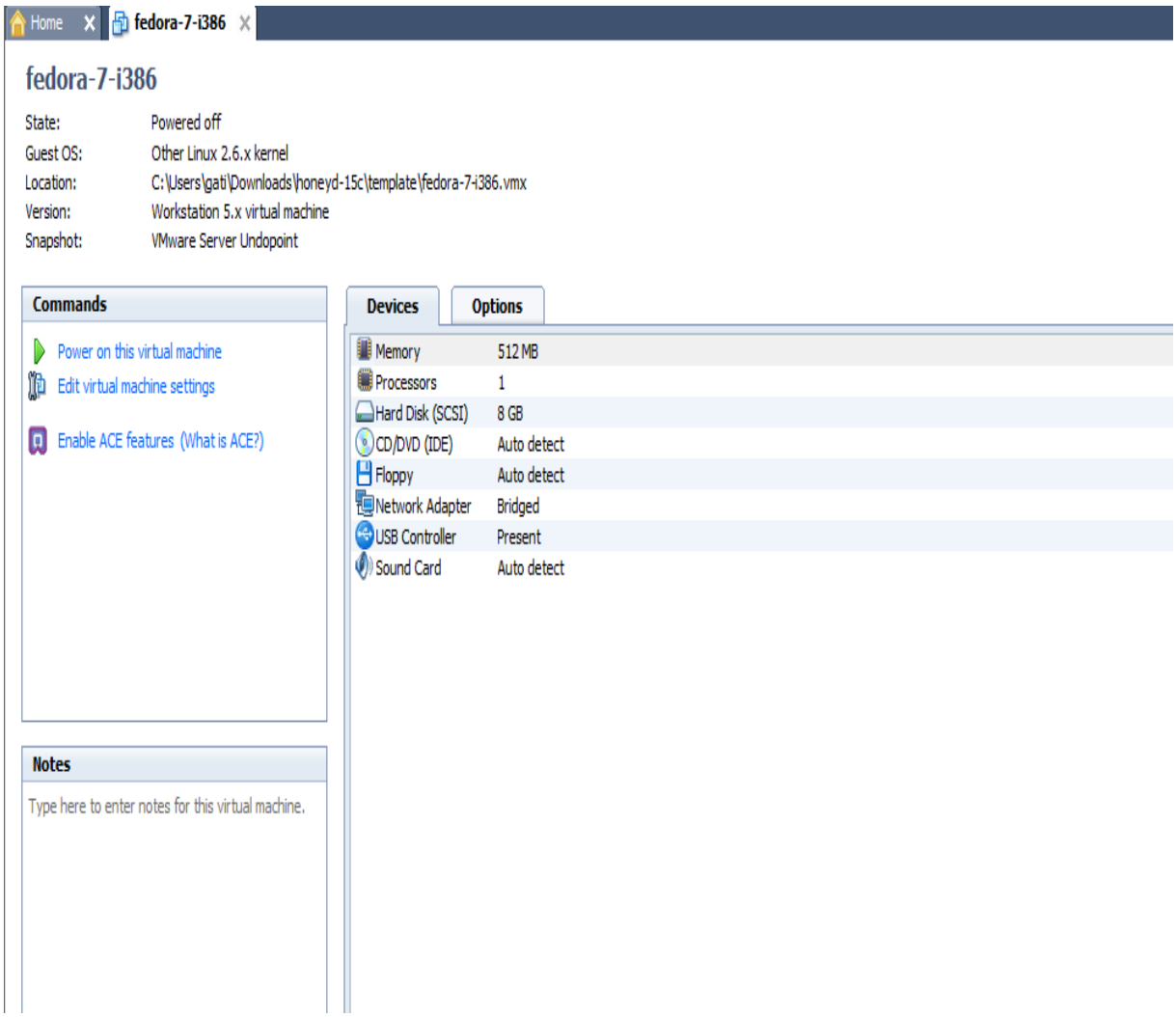
Εικόνα 37: Εγκατάσταση arpd



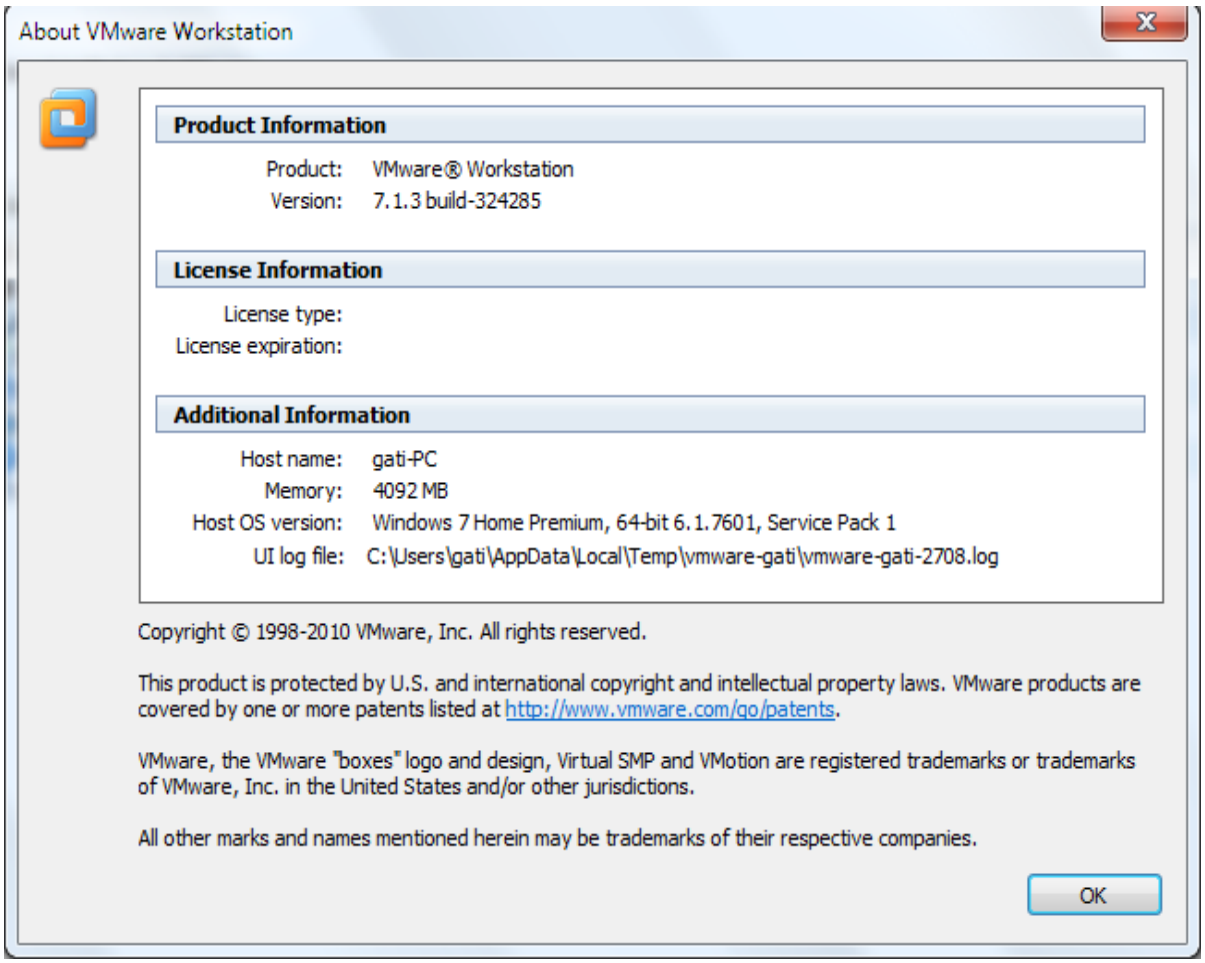
Εικόνα 38: Configuration του honeyd

Λόγω των περιορισμένων δυνατοτήτων του μηχανήματος (περιορισμένη υπολογιστική ισχύ και μικρού μεγέθους μνήμης RAM) και για λόγους ευκολίας έγινε εγκατάσταση του Honeyd εικονικά μέσω VMware σε ένα μηχάνημα με περισσότερες δυνατότητες με λειτουργικό σύστημα Windows 7.

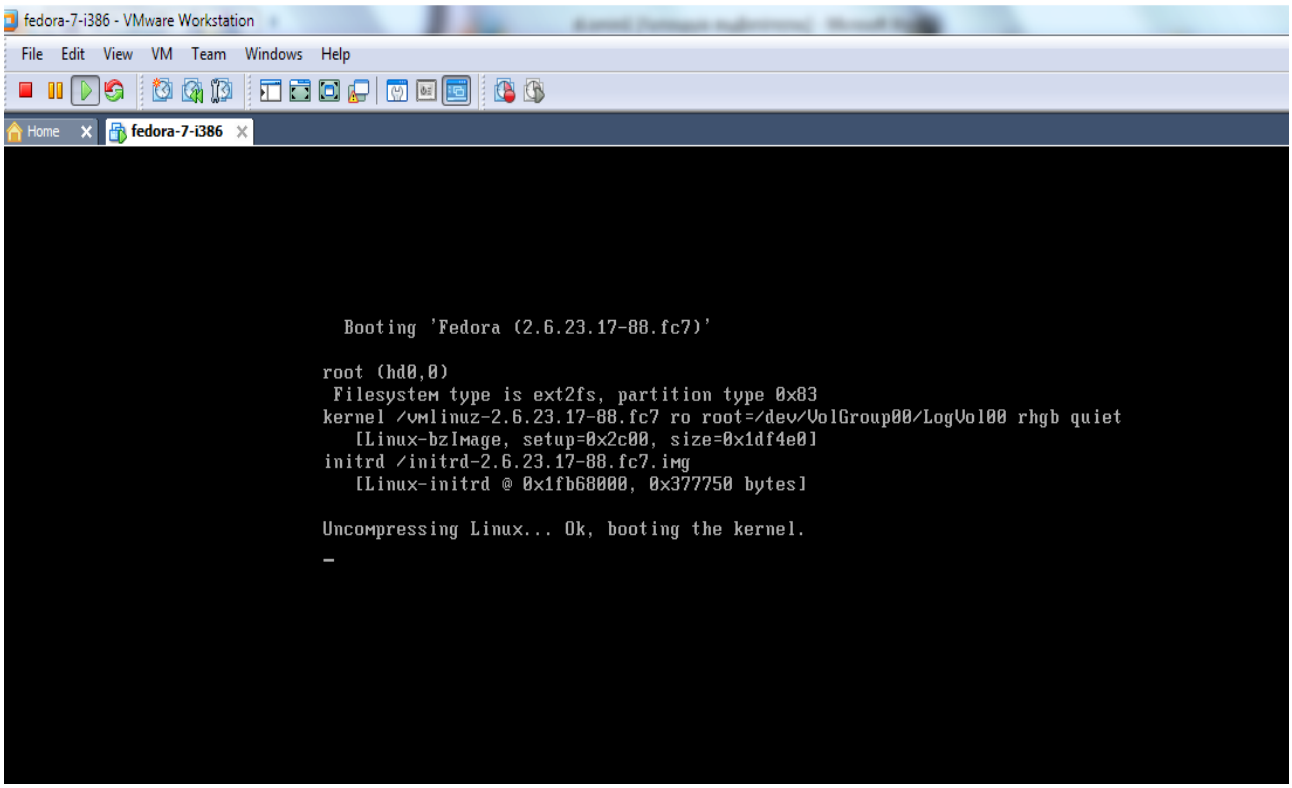
Συγκεκριμένα έγινε εγκατάσταση του linux-fedora-core7, εικόνα 40, στο vmware (7.1.3 build), εικόνα 41, στο οποίο αντίστοιχα έγιναν εγκατάσταση οι βιβλιοθήκες που απαιτούνταν (arpd, libevent, libnet και libpcap) από τις αποθήκες του λογισμικού του linux-fedora-core7, και όλο το απαραίτητο software. Μετά την εγκατάσταση έγινε de-bugging στις παραπάνω βιβλιοθήκες έτσι ώστε να προσαρμοστούν και να λειτουργήσουν σωστά με τις εκδόσεις vmware και honeyd που εγκαταστάθηκαν και έγινε και configuration του honeyd από το αρχείο παραμετροποίησης του Honeyd το “honeyd.conf” για να γίνουν οι εικονικές επιθέσεις.



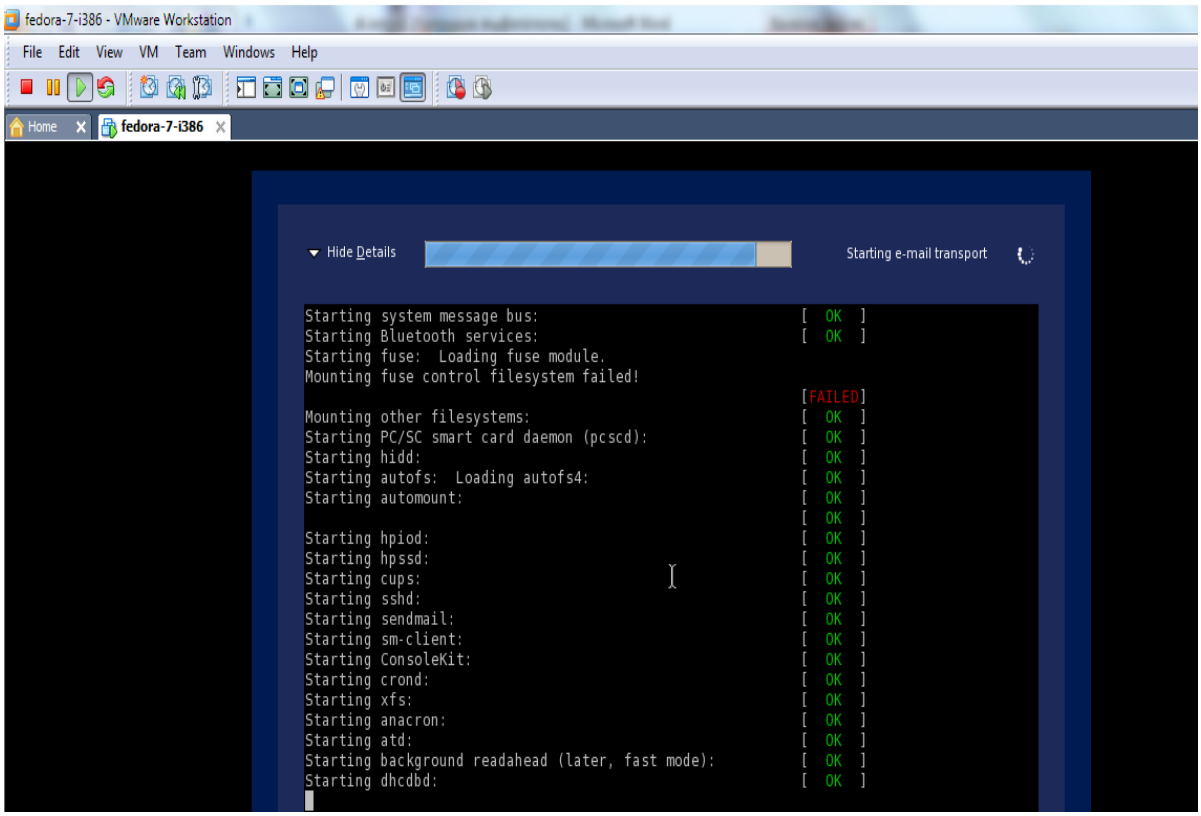
Εικόνα 39: Linux – fedora – core7



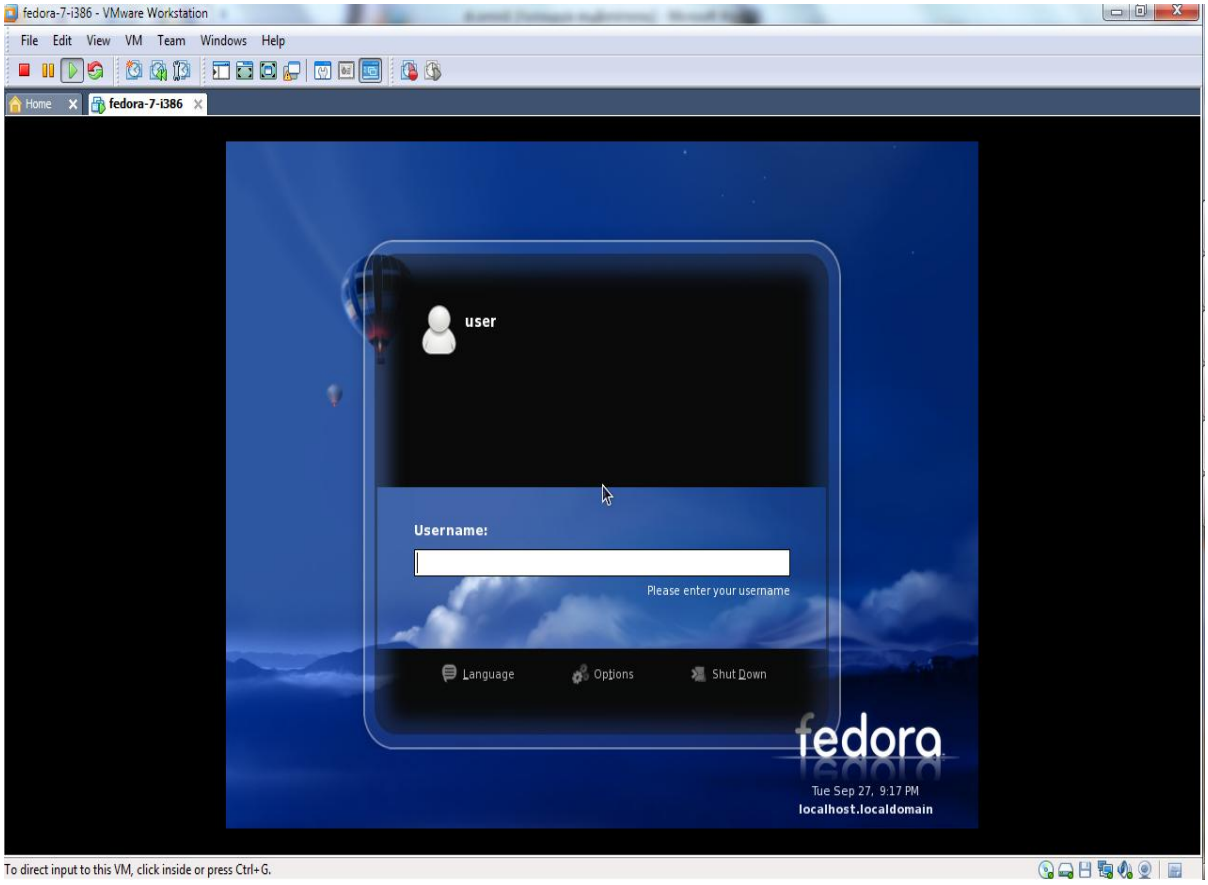
Εικόνα 40: VMware 7.1.3 build



Εικόνα 41: Φόρτωση Fedora

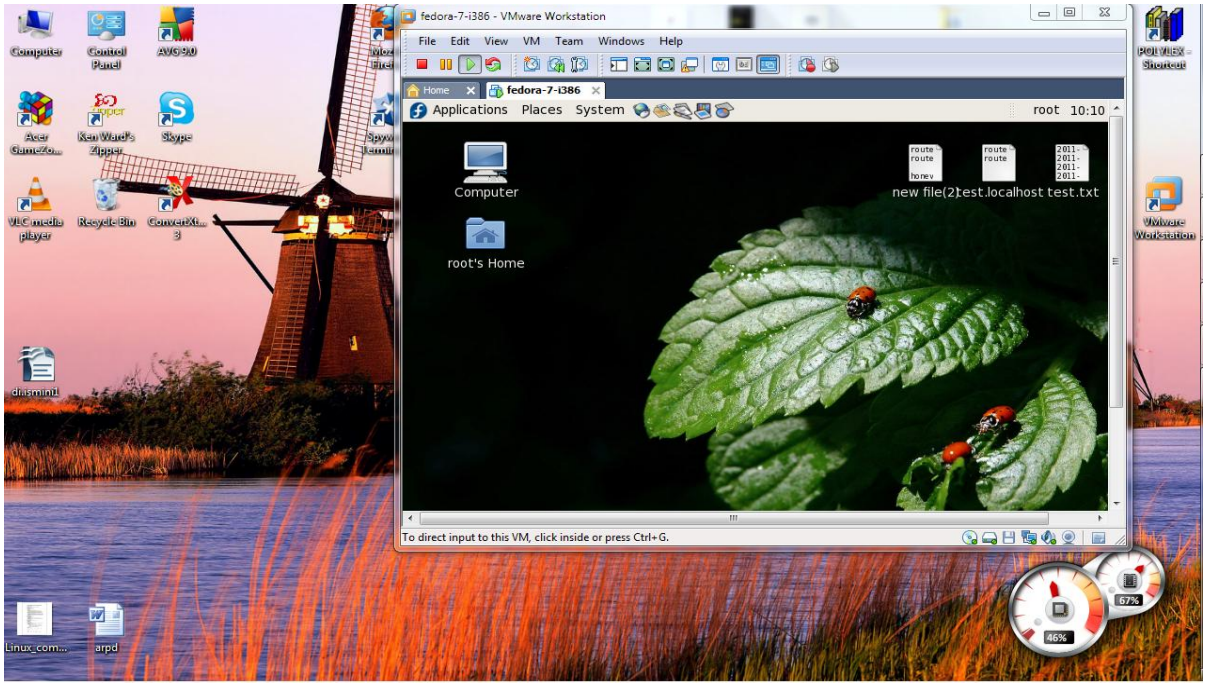


Εικόνα 42: Φόρτωση βιβλιοθηκών



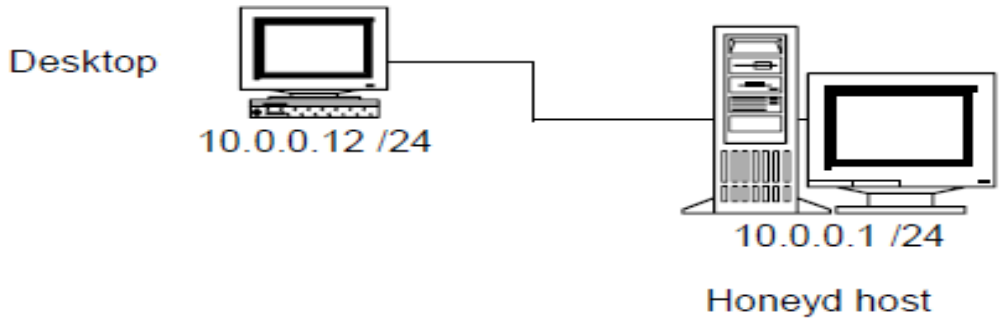
Εικόνα 43: Επιβεβαίωση Ταυτότητας Χρήστη

Η δημιουργία του εικονικού honeyd μέσω VMware μας δίνει την δυνατότητα χρήσης περισσότερων λειτουργικών συστημάτων, το καθένα από τα οποία θα είχε τη δική του ip και παράλληλα την ευκολότερη συντήρησή του με λιγότερες φυσικές απαιτήσεις.



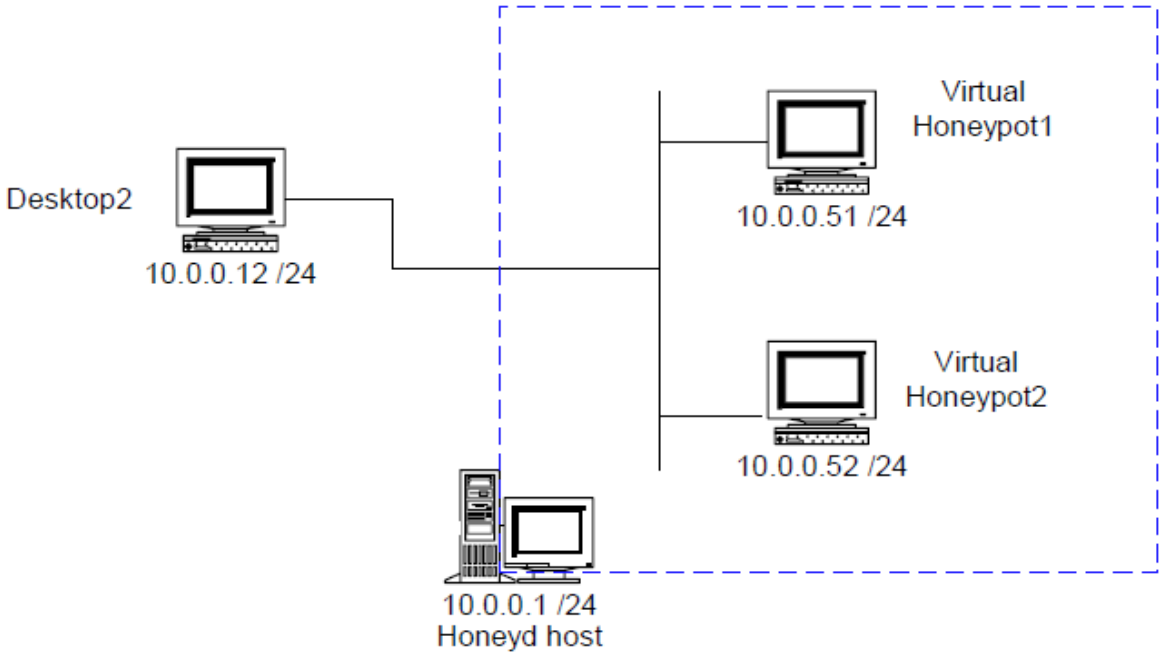
Εικόνα 44: Προσομοίωση Fedora μέσω VMware σε Windows 7

Έτσι λοιπόν το φυσικό μας δίκτυο είναι ένα εικονικό pc με λειτουργικό linux-fedora-core7, διαμορφωμένο να καλύπτει το εύρος των ip διευθύνσεων 10.0.0.0/24 και το Honeyd "οικοδεσπότης" δηλαδή το Honeyd Host έχει την διεύθυνση ip 10.0.0.1 όπως φαίνεται στην εικόνα 46.



Εικόνα 45: Τοπολογία δικτύου 1

Αρχικά δημιουργήσαμε 2 εικονικά Honeypot στον Honeyd host με διεύθυνση ip 10.0.0.51 και 10.0.0.52 αντίστοιχα (όπως βλέπουμε και στο σχήμα 47) για να δούμε με ποιον τρόπο μπορούν να εξομοιωθούν (simulate) από το Honeyd. Η μπλε γραμμή δείχνει τον Honeyd Host που μιμείται τα εικονικά honeypots.



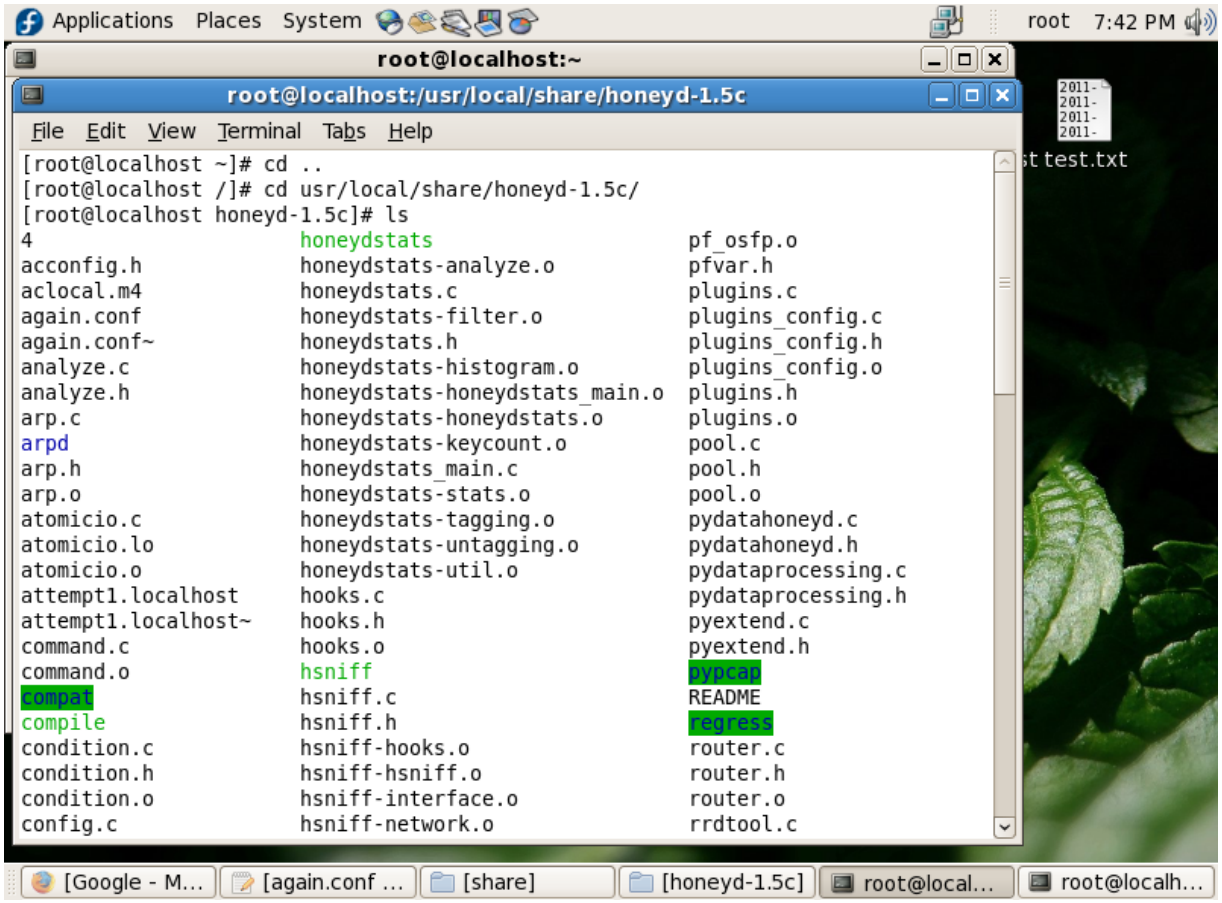
Εικόνα 46: Τοπολογία δικτύου 2

Στην συνέχεια τροποποιήθηκε το αρχείο του Honeyd (το honeyd.conf) για την δημιουργία των δύο Honeypot . Στο νέο αρχείο δόθηκε το όνομα again.conf και η σύνταξή του είναι η εξής:

```
### Windows computers
create windows
set windows personality "Windows NT 4.0 SP3"
add windows tcp port 80 "sh scripts/web.sh"
add windows tcp port 22 "sh scripts/test.sh"
add windows tcp port 139 open
add windows tcp port 137 open
add windows udp port 135 block
set windows default tcp action reset
bind 10.0.0.51 windows
bind 10.0.0.52 windows
```

Το template σύστημα θα παρουσιαστεί σαν “Windows NT 4.0 SP3”, όταν κάποιος θελήσει να μάθει για τί σύστημα πρόκειται μέσω NMap or XProbe, το οποίο έχει ανοιχτές τις πόρτες 80/tcp, 22/tcp, 137/tcp, 139/tcp ,η 135/tcp είναι block και τα 2 honeypot δεσμεύουν τις ip διευθύνσεις 10.0.0.51 και 10.0.0.52.

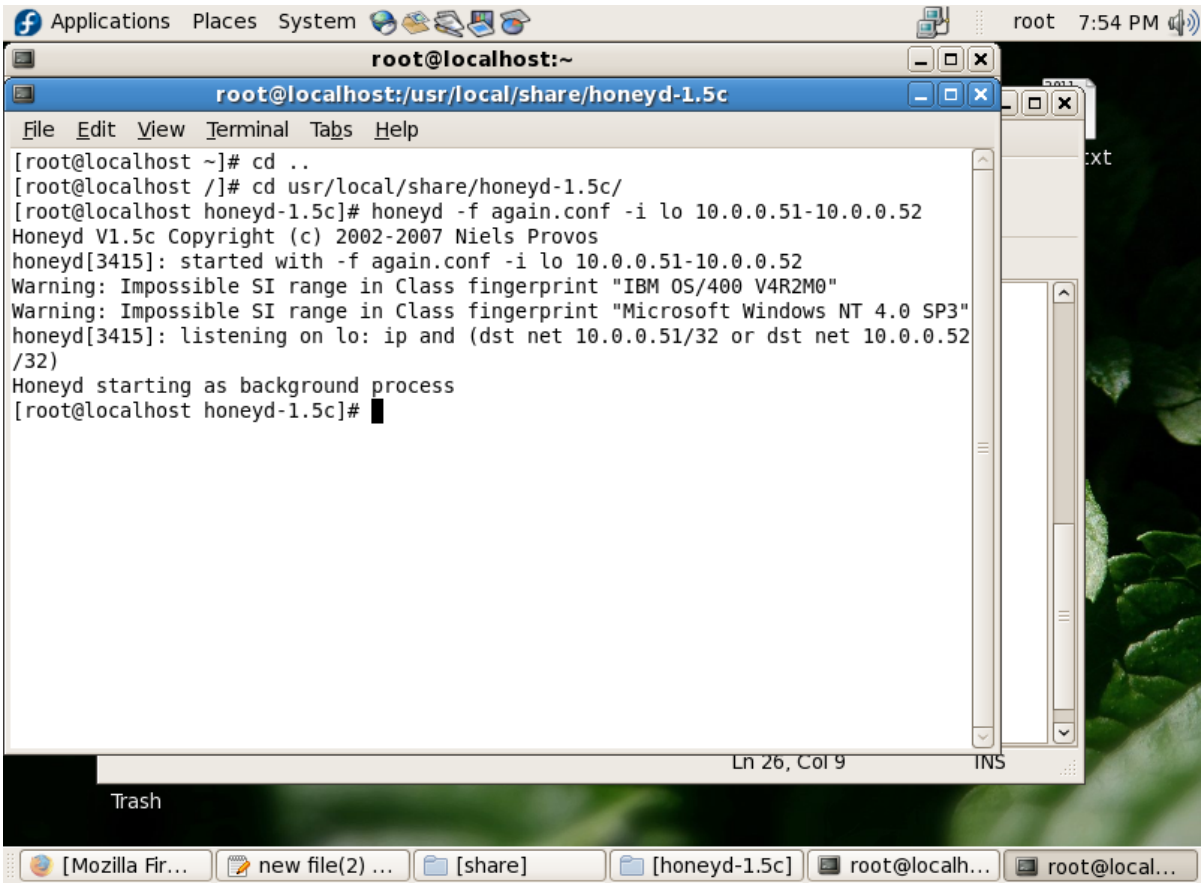
Για να εκκινήσουμε το Honeyd, θα πρέπει να μπούμε μέσα στο φάκελο όπου βρίσκεται το script το Honeyd το (again.conf)



Εικόνα 47: Τοποθεσία αρχείου

και μέσω terminal (τερματικού) να πληκτρολογήσουμε την εντολή

#./honeyd –f again.conf –i lo 10.0.0.51-10.0.0.52



Εικόνα 48: Εκκίνηση Honeyd

Οι παράμετροι που υπάρχουν στην παραπάνω εντολή εξηγούνται ως εξής:

-f Με αυτή την παράμετρο καθορίζουμε το αρχείο το οποίο θα «διαβάσει» το Honeyd όπου ορίζονται τα honeypots.

-I Με αυτή την παράμετρο καθορίζουμε τη διεπαφή την οποία θα χρησιμοποιήσει το Honeyd

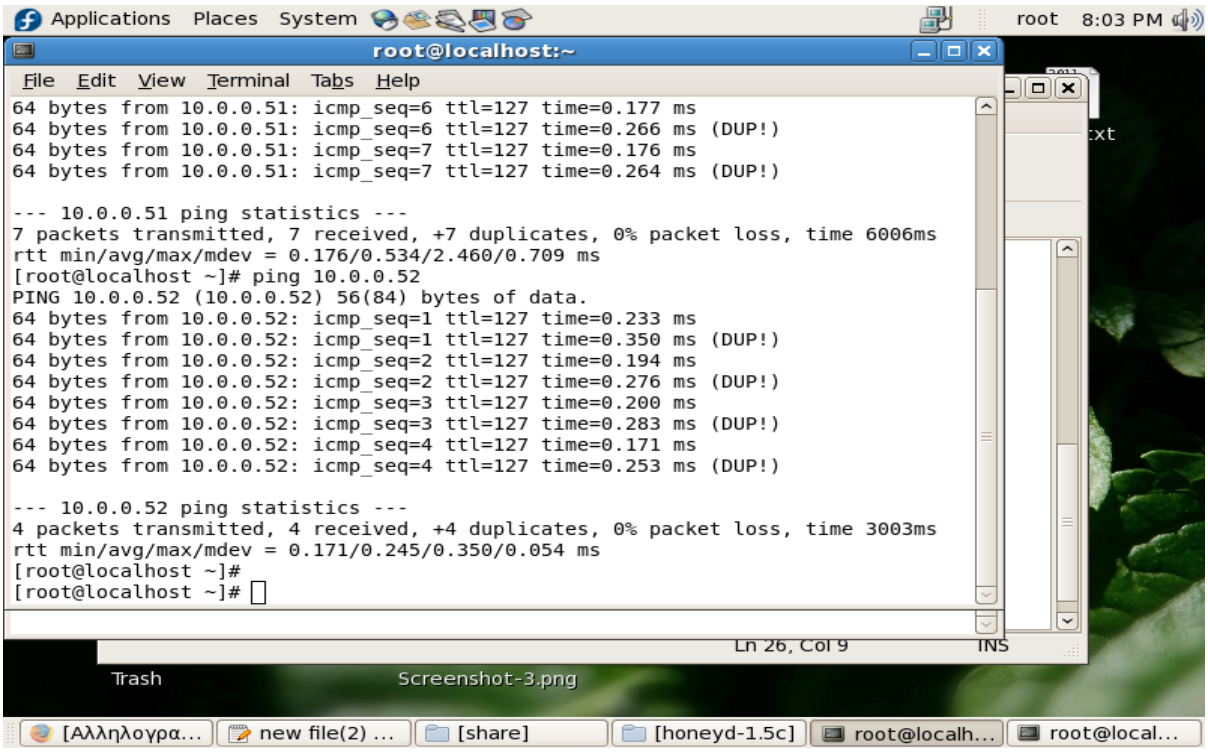
Σε ένα δεύτερο terminal μπορούμε να ελέγξουμε ότι πράγματι στις διευθύνσεις 10.0.0.51 και 10.0.0.52 ανταποκρίνονται τα 2 Honeypot αφού πρώτα καθορίσουμε το δίκτυο που των δύο Honeypot, με την εντολή

#./route –n add –net 10.0.0.0/24 lo

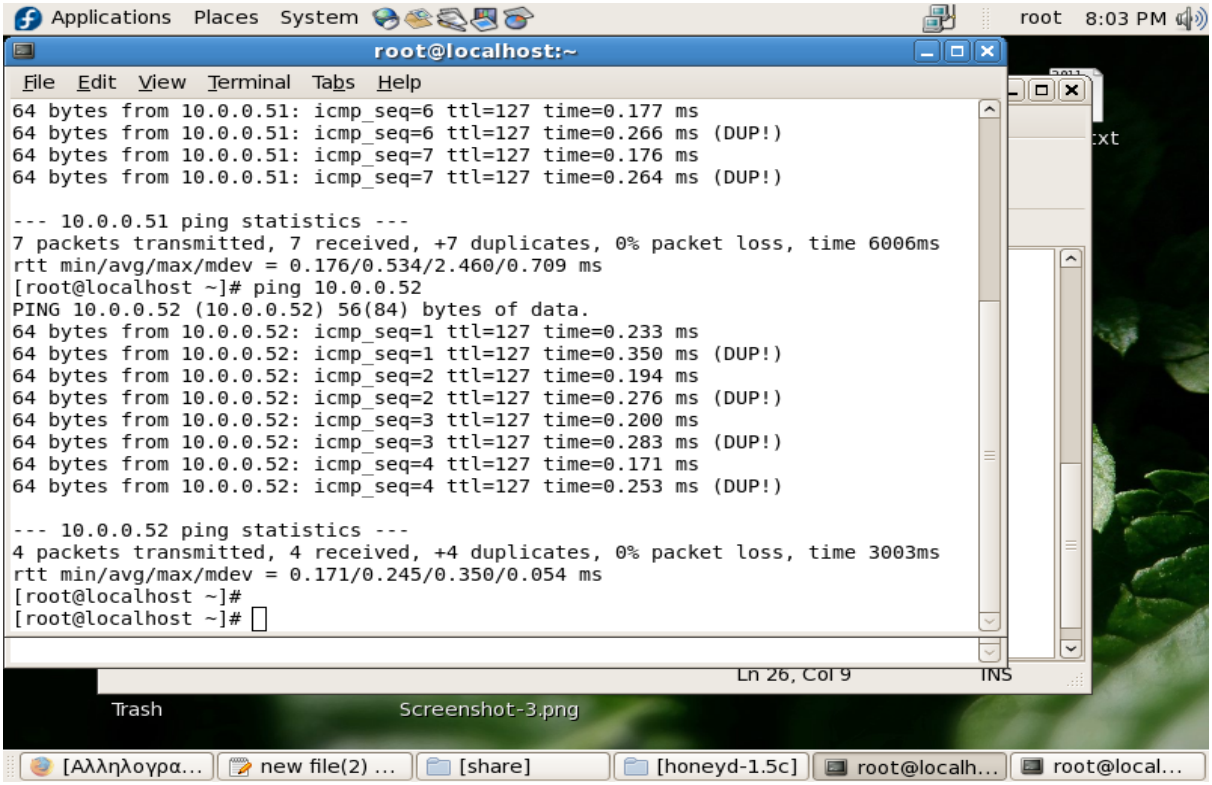
-n Με αυτή την παράμετρο καθορίζουμε αριθμητικά τη διεύθυνση του host honeyd

-net Με αυτή την εντολή καθορίζουμε το εύρος του δικτύου

Στο τέλος της εντολής καθορίζουμε το υποδίκτυο στο οποίο θα αντιστοιχηθεί το default template.

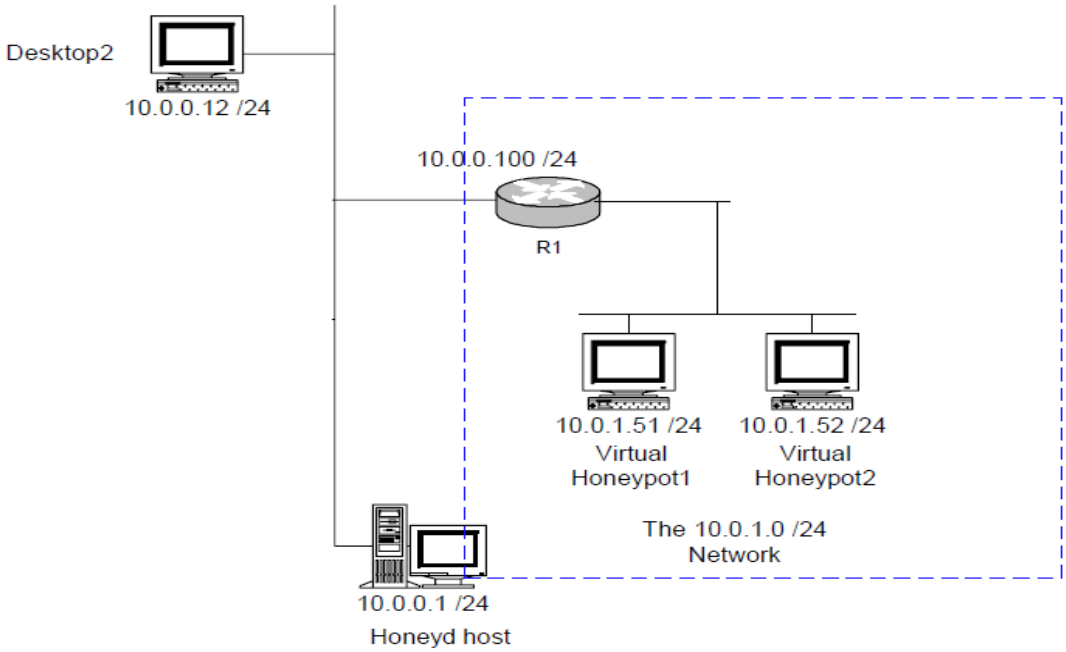


Εικόνα 49: ping 10.0.0.51



Εικόνα 50: ping 10.0.0.52

Έπειτα προχωρήσαμε στην δημιουργία ενός απλού δικτύου, για να δούμε αντίστοιχα πως εξομοιώνεται (simulate) από το Honeyd. Το εξομοιωμένο - simulated δίκτυο (όπως βλέπουμε και στην εικόνα 52) χρησιμοποιεί – δεσμεύει ένα εύρος διευθύνσεων ip 10.0.1.0 /24 και περιέχει δύο honeypots και έναν δρομολογήτή – router Cisco (R1).



Εικόνα 51: Τοπολογία δικτύου 3

Για τη ρύθμιση του router έγιναν οι απαραίτητες τροποποιήσεις στο honeyd.conf και επιπλέον στο router «δεσμεύτηκε» έγινε bind η ip 10.0.0.100.

```
### Cisco router

create router

set router personality "Cisco 7206 running IOS 11.1(24)"

set router default tcp action reset

add router tcp port 23 "scripts/router-telnet.pl"

add router tcp port 113 open

add router tcp port 114 reset

add router tcp port 115 block

bind 10.0.0.100 router
```

Ο router R1 είναι το σημείο εισόδου από το τοπικό δίκτυο στο εικονικό, ο οποίος έχει δεσμεύσει την ip 10.0.0.100 ενώ το εικονικό μας δίκτυο δεσμεύει το εύρος 10.0.0.0 /16

```
#./route entry 10.0.0.100 network 10.0.0.0/16
```

Επίσης ο router είναι άμεσα συνδεδεμένος με το εξομοιωμένο δίκτυο το οποίο όπως προαναφέραμε δεσμεύει το εύρος των διευθύνσεων 10.0.1.0/24.

```
#./route 10.0.0.100 link 10.0.1.0/24
```

Τα 2 honeypots δεσμεύουν αντίστοιχα τις διευθύνσεις

```
bind 10.0.1.51 windows
bind 10.0.1.52 windows
```

Θεωρητικά το δίκτυό μας είναι έτοιμο οπότε μας μένει να ελέγξουμε αν και πρακτικά το honeyd εξομοιώνει το δίκτυο.

Για την εκκίνηση του Honeyd πληκτρολογούμε την εντολή

```
#./honeyd -u 99 -g 99 -d -p nmap.prints –f again.conf -i lo 10.0.0.0/16
```

Οι παράμετροι που υπάρχουν στην παραπάνω εντολή εξηγούνται ως εξής:

-u Με αυτή την παράμετρο καθορίζεται το UID του χρήστη με τα δικαιώματα του οποίου θα «τρέχει» το Honeyd. Με το 99 καθορίζουμε πως το Honeyd θα «τρέχει» ως nobody.

-g Με αυτή την παράμετρο καθορίζεται το GID του group με τα δικαιώματα του οποίου θα «τρέχει» το Honeyd. Με το 99 καθορίζουμε πως το Honeyd θα «τρέχει» ως nobody.

-d Με αυτή την παράμετρο καθορίζουμε ότι το Honeyd θα τρέξει σε κατάσταση αποσφαλμάτωσης (debugging) που σημαίνει πως θα εμφανίζει όλα τα μηνύματα στην οθόνη του υπολογιστή και δεν θα τρέχει ως δαίμονας στο παρασκήνιο.

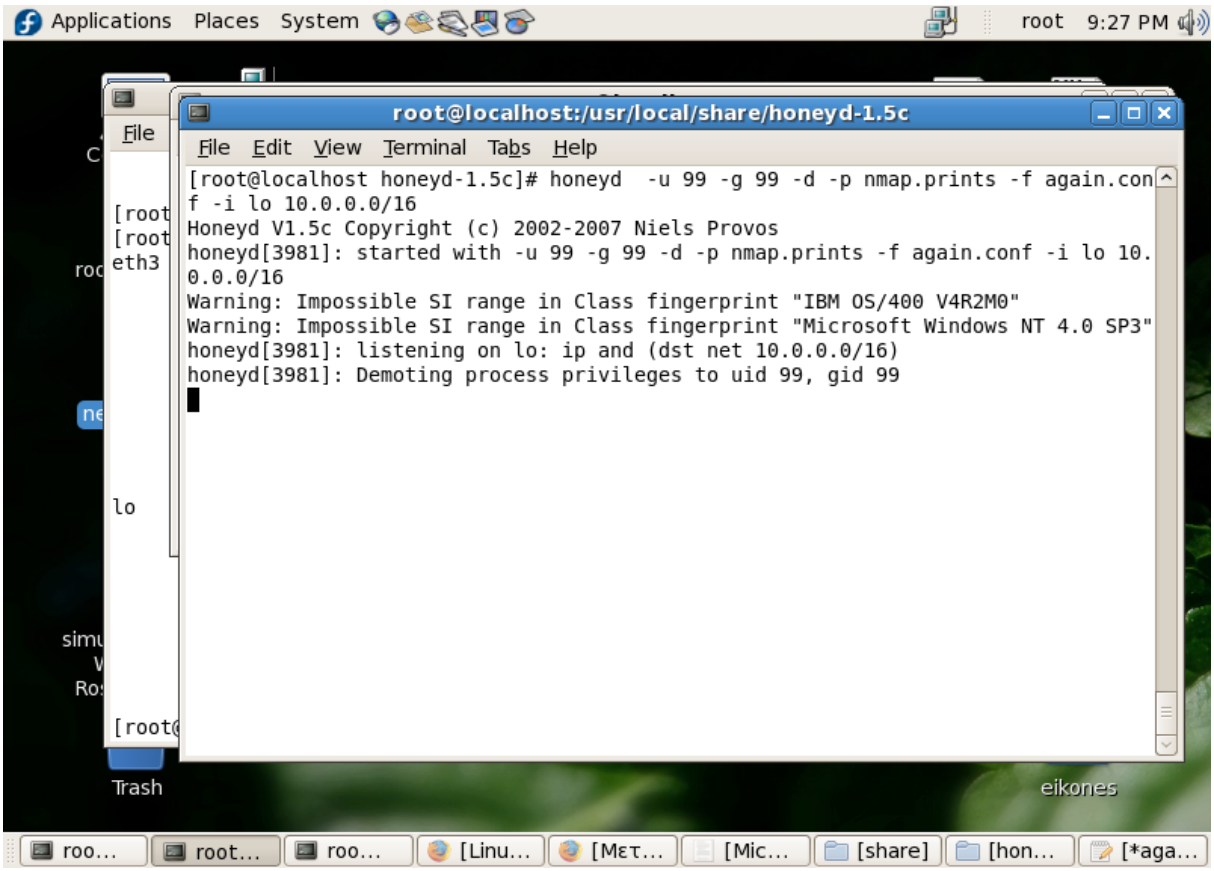
-l Με αυτή την παράμετρο καθορίζουμε την τοποθεσία και το αρχείο στο οποίο θα αποθηκεύονται τα μηνύματα που το Honeyd καταγράφει.

-p Με αυτή την παράμετρο καθορίζουμε το την τοποθεσία και το αρχείο με τα signatures τύπου Nmap τα οποίο θα χρησιμοποιήσει το Honeyd.

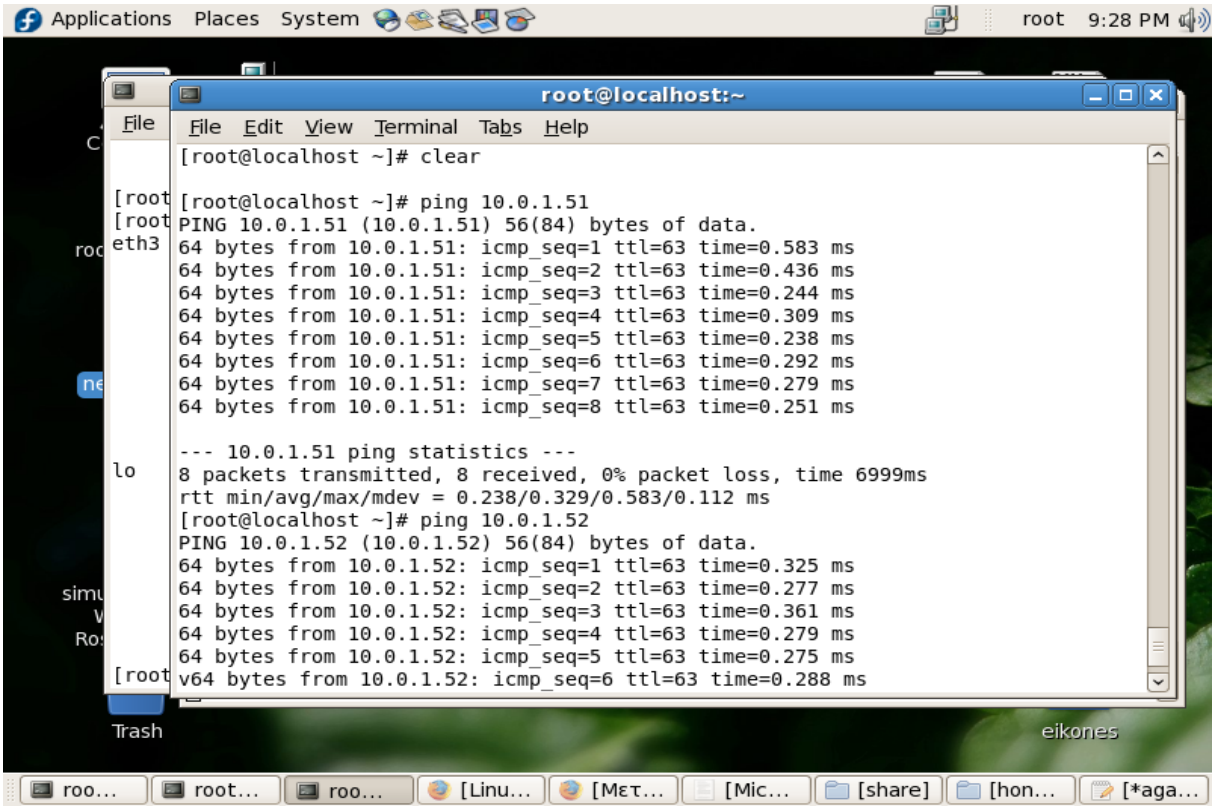
-f Με αυτή την παράμετρο καθορίζουμε το αρχείο το οποίο θα «διαβάσει» το Honeyd όπου ορίζονται τα honeypots.

-I Με αυτή την παράμετρο καθορίζουμε τη διεπαφή την οποία θα χρησιμοποιήσει το Honeyd

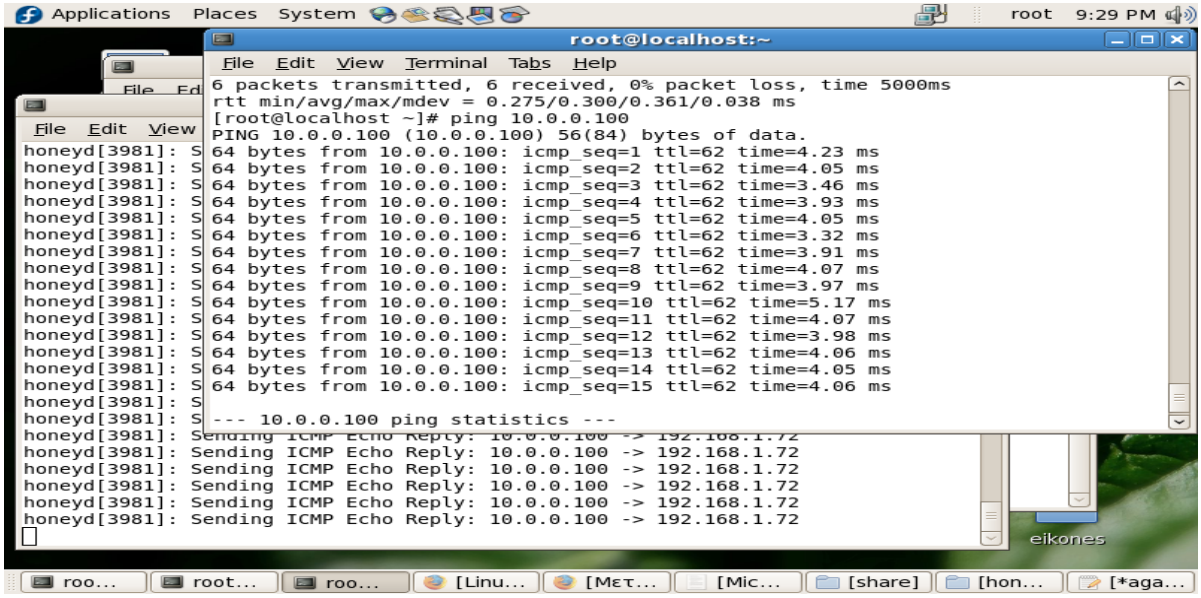
Στο τέλος της εντολής καθορίζουμε το υποδίκτυο στο οποίο θα αντιστοιχηθεί το default template.



Εικόνα 52: Εκκίνηση του Honeyd



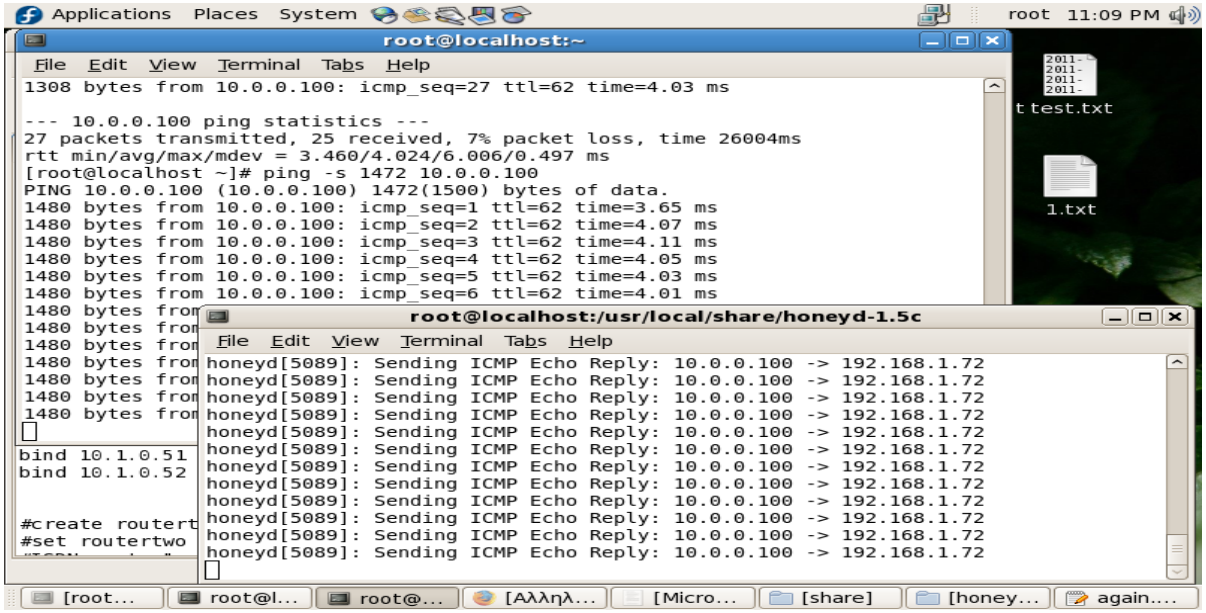
Εικόνα 53: Ping 10.0.0.51 & 10.0.0.52



Εικόνα 54: Ping 10.0.0.100

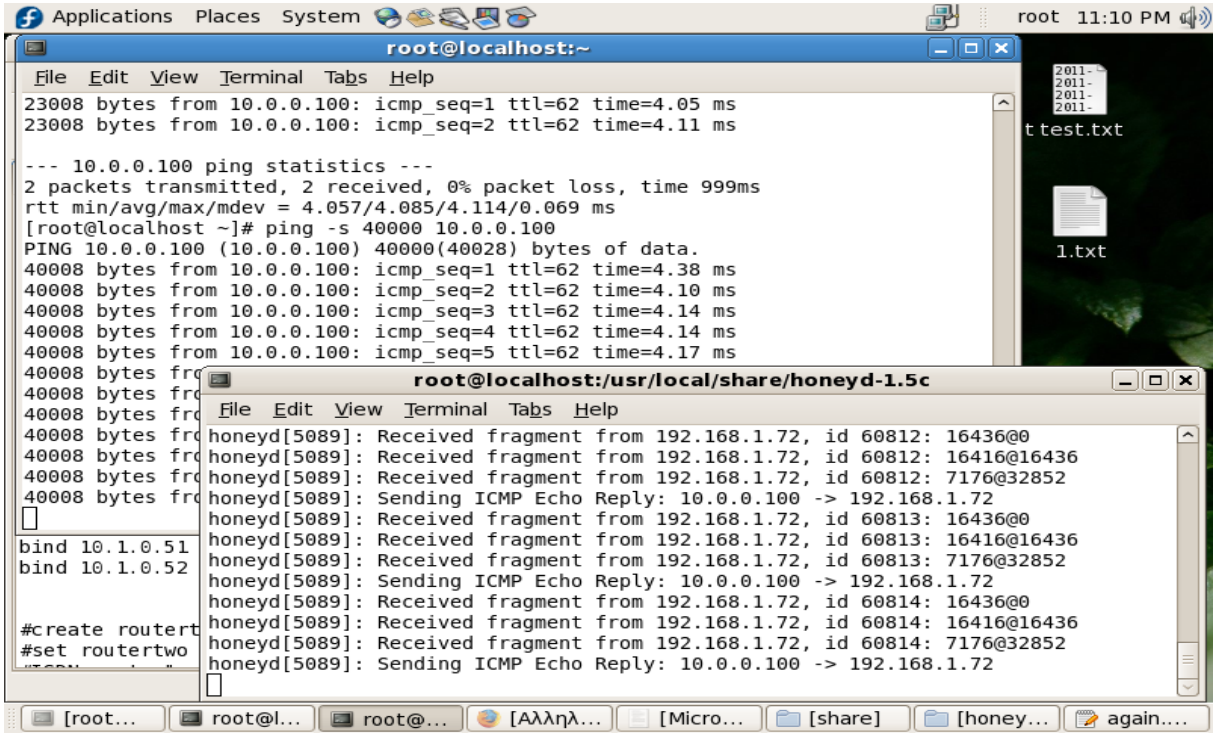
Όπως παρατηρούμε το honeyd μπορεί να εξομοιώσει το δίκτυο που περιγράψαμε παραπάνω. Οι επιθέσεις που μπορούν να γίνουν στο παραπάνω δίκτυο είναι όπως προαναφέραμε και στο πρώτο κεφάλαιο είναι η Άρνησης Υπηρεσίας (DOS), το Packet sniffing, η Malware επίθεση κ.α. . Στο παραπάνω δίκτυο πραγματοποιήθηκε μια Ping of Death επίθεση και μια telnet όπως θα δούμε και στις εικόνες που ακολουθούν.

Για την Ping of Death επίθεση αρχικά εστάλη ένας μικρός όγκος πακέτων και το honeyd απάντησε κανονικά. Συγκεκριμένα εστάλησαν 1472 πακέτα και η απάντηση του honeyd είναι η παρακάτω



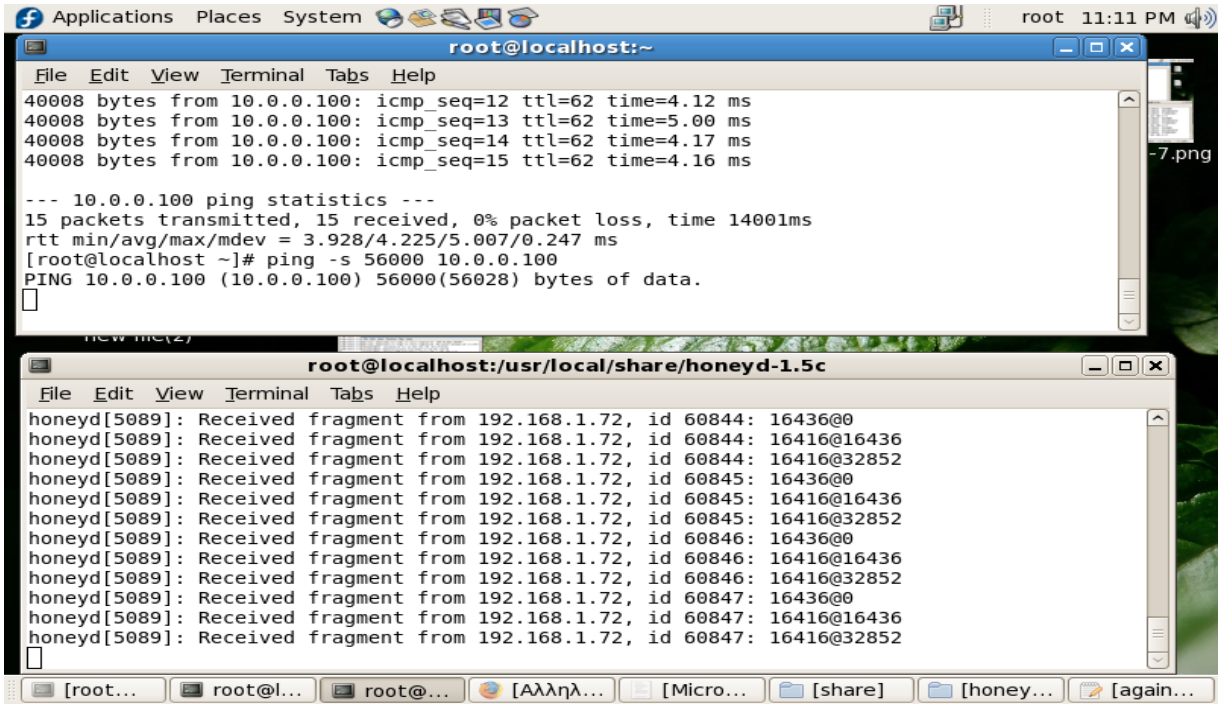
Εικόνα 55: Ping of death attack

Στη συνέχεια αυξήσαμε τον όγκο των πακέτων σε 40000 όπου πάλι το Honeyd απάντησε



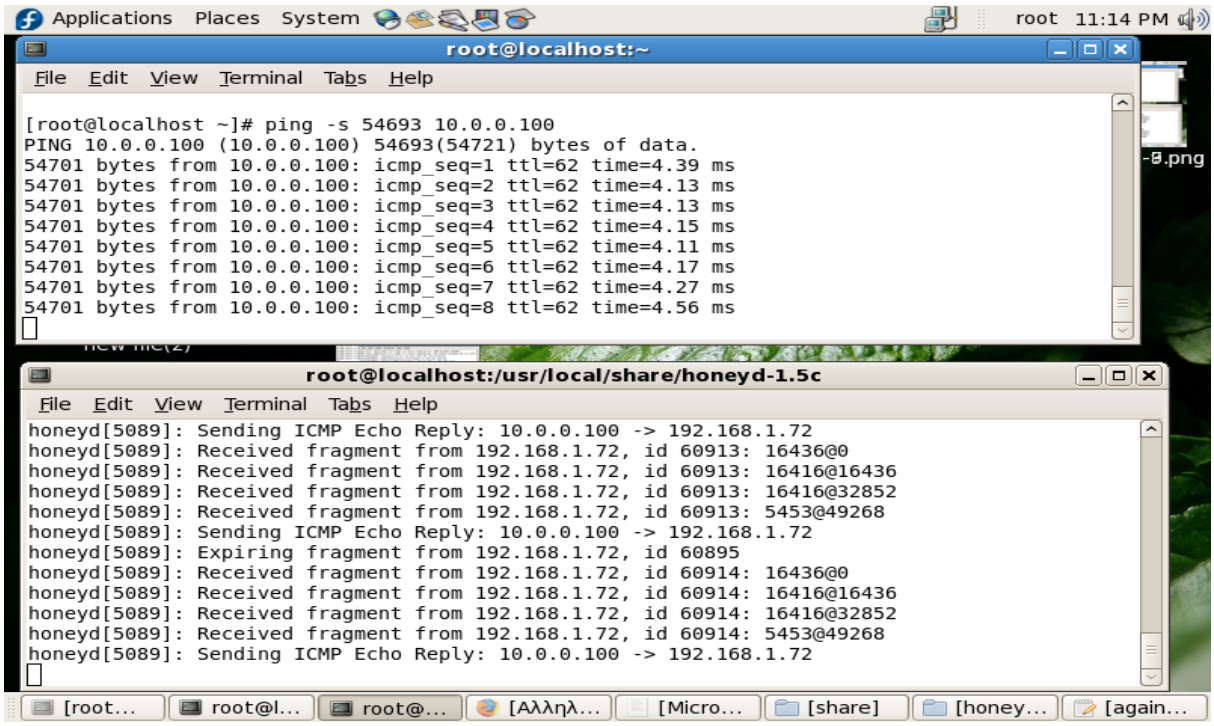
Εικόνα 56: Ping of death attack_2

Συνεχίσαμε να αυξάνουμε τα πακέτα σε 56000 και εδώ παρατηρήσαμε ότι το honeyd δεν απαντάει.



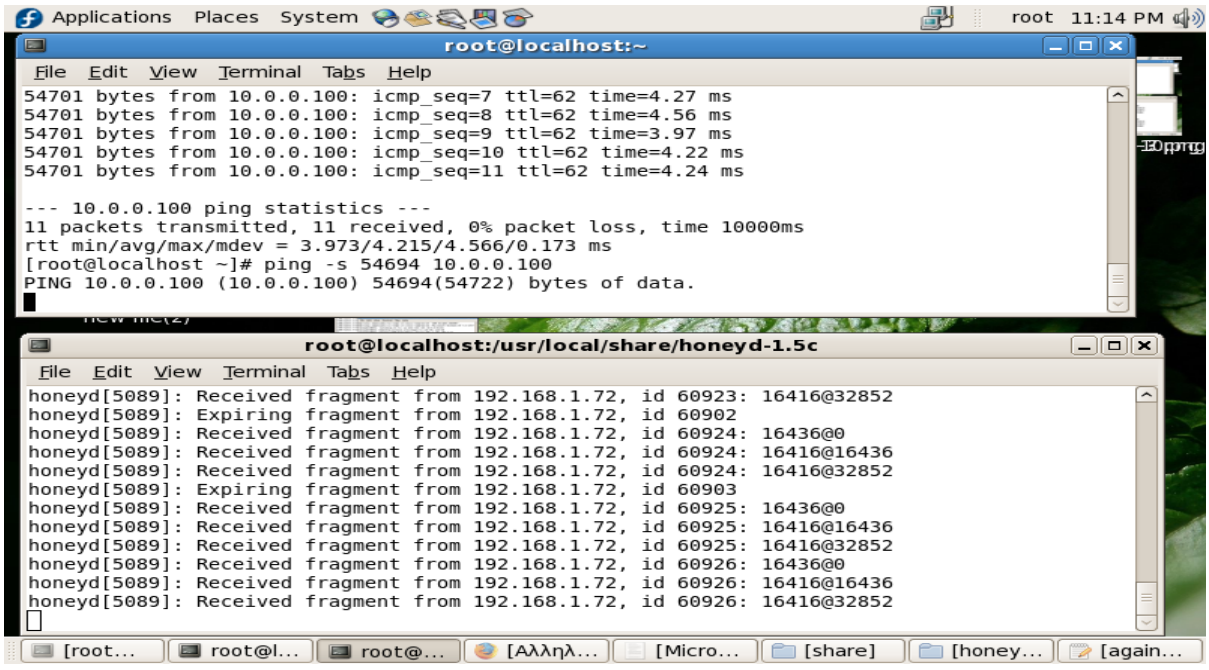
Εικόνα 57: Ping of death attack_3

Έτσι θελήσαμε να δούμε ποιος είναι ο μέγιστος αριθμός των πακέτων που μπορούσαμε να στείλουμε χωρίς να καταρρεύσει ο router μας. Μετά από αρκετές δοκιμές διαπιστώσαμε ότι μπορούσαμε να στείλουμε μέχρι και 54693 πακέτα



Εικόνα 58: Ping of death attack_4

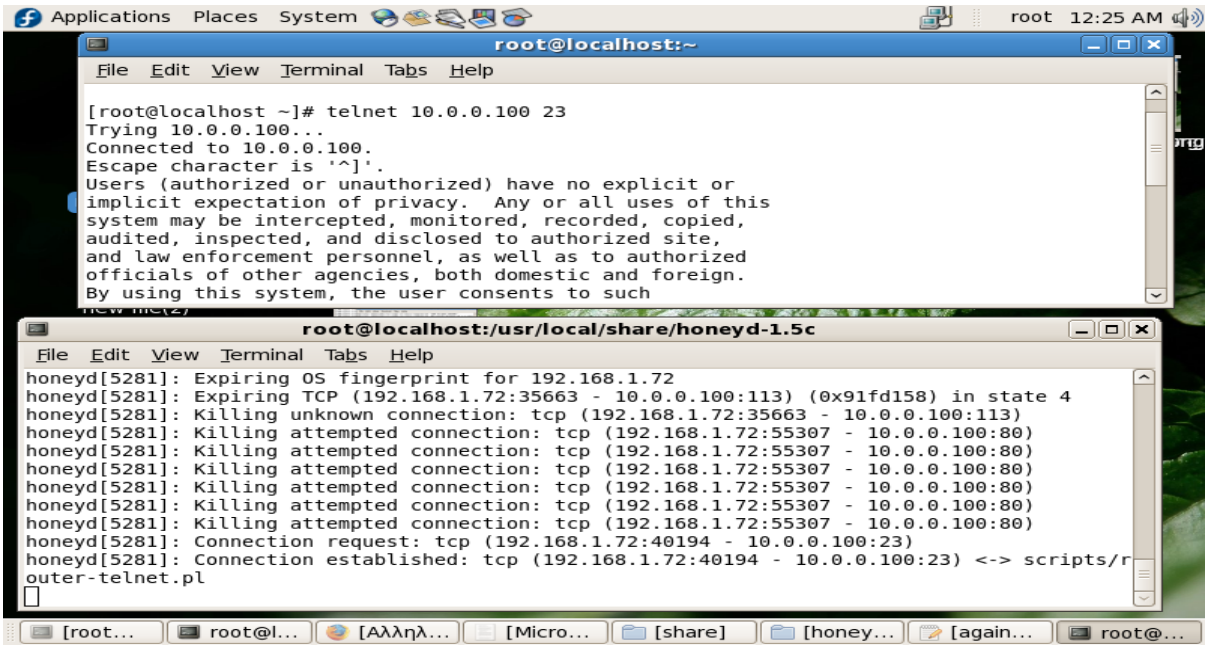
Αφού στα 54694 πακέτα έκανε reboot.



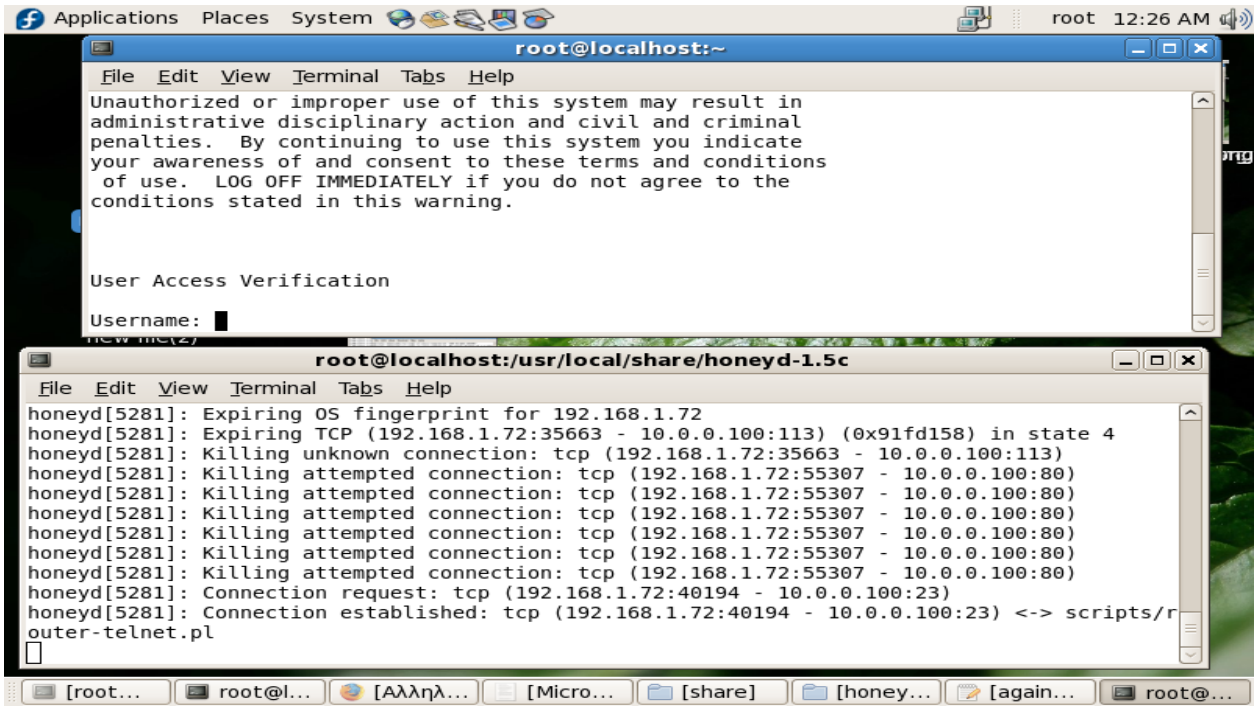
Εικόνα 59: Ping of death attack_5

Στην επίθεση μέσω telnet θελήσαμε να δούμε ποία θα είναι η απάντηση του Honeyd όταν θα προσπαθήσουμε να «παραβιάσουμε» τις πόρτες 23, 113, 150, 139 που ορίσαμε στο configuration file του honeyd , to again.conf

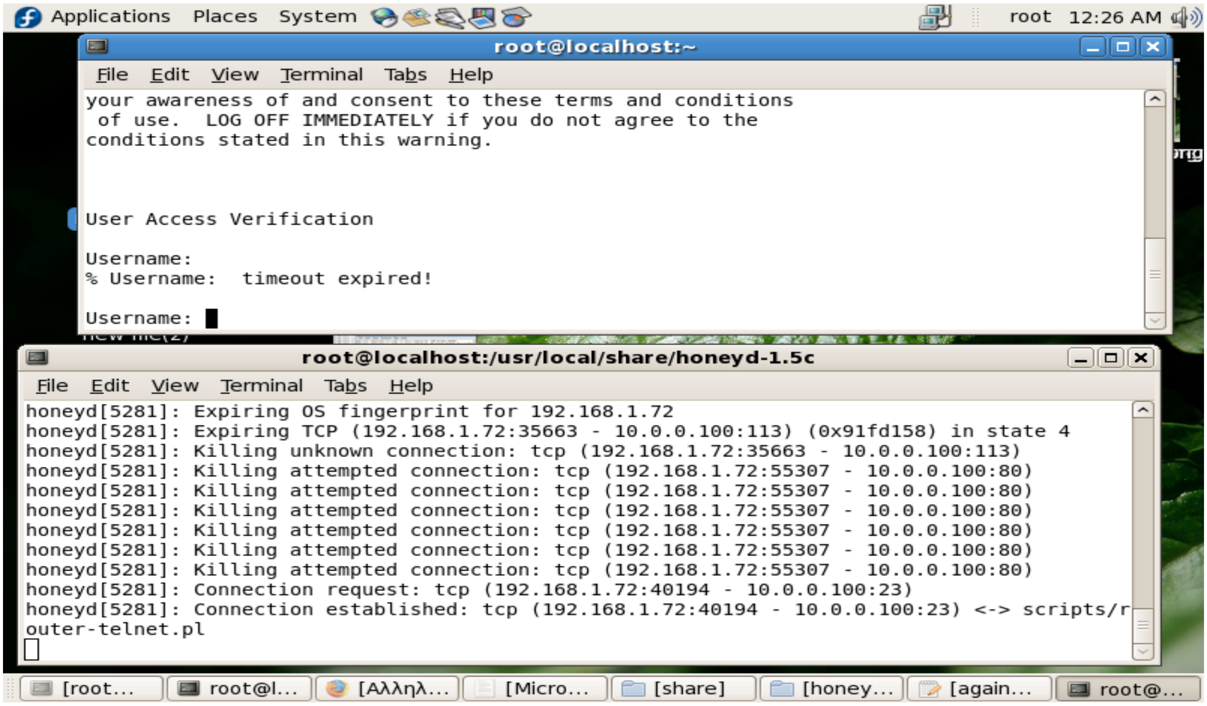
Τις πόρτες 23, 113 τις είχαμε ορίσει να είναι ανοιχτές. Έτσι εκτελώντας την εντολή telnet 10.0.0.100 23



Εικόνα 60: Telnet attack to port 23



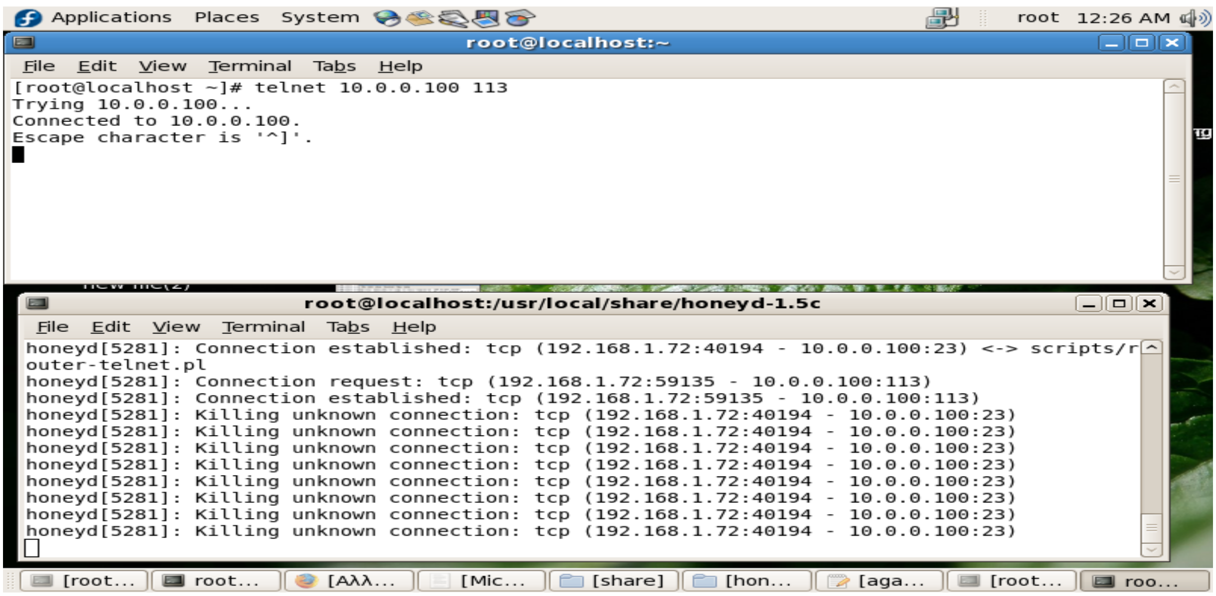
Εικόνα 61: Telnet attack to port 23- User Access Verification



Εικόνα 62: Telnet attack to port 23 - log in failed

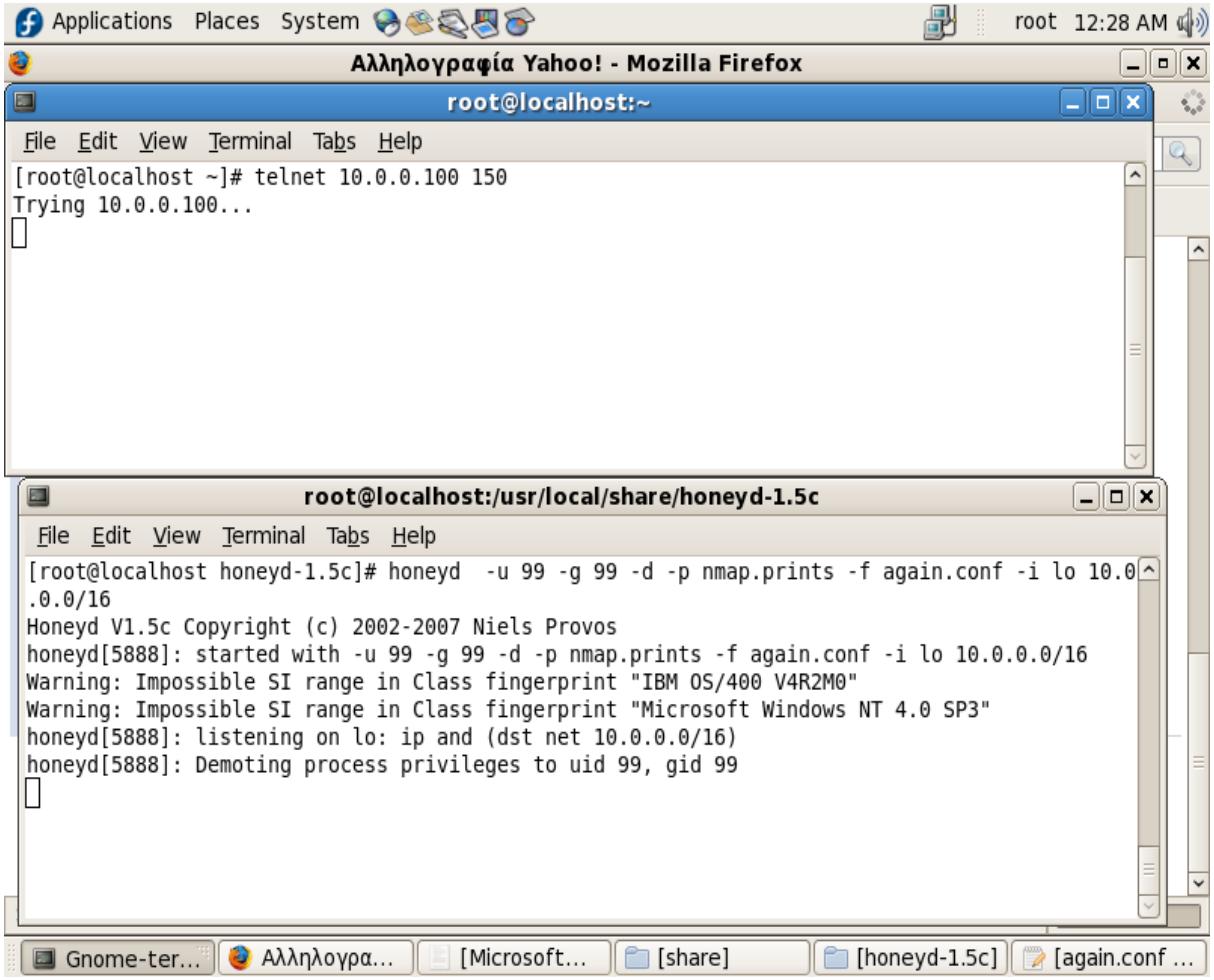
καταφέρνουμε να δούμε ότι η πόρτα είναι ανοιχτή και μας δίνεται η δυνατότητα να δώσουμε username και password αλλά με το που πάμε να πληκτρολογήσουμε το username τελειώνει και ο χρόνος πρόσβασης, οπότε πάλι πρέπει να πληκτρολογήσουμε το username , όπως βλέπουμε και στην (εικόνα 63).

Εκτελώντας την ίδια εντολή αλλά για την πόρτα 113 που είναι επίσης ανοιχτή αλλά δεν υποστηρίζει κάποια υπηρεσία, ναι μεν καταφέρνουμε να συνδεθούμε με το router αλλά δεν έχουμε κανένα άλλο δικαίωμα όπως βλέπουμε και στην (εικόνα 64)



Εικόνα 63: Telnet attack to port 113

Αντίστοιχα αν προσπαθήσουμε να παραβιάσουμε την πόρτα 150 (την οποία ορίσαμε σαν block), δεν παίρνουμε καμία απάντηση,



Εικόνα 64: Telnet attack to port 150

Μια πολύ σημαντική εφαρμογή του honeypot που θα μπορούσε να υλοποιηθεί ως μελλοντική εργασία, είναι η εφαρμογή Honeyd vs worms η οποία μπορεί να χρησιμοποιηθεί στην έρευνα για την καταπολέμηση των worms [28], να εντοπίσει καινούργια worms ή ακόμα και να λάβει ενεργά μέτρα ενάντια στα μολυσμένα μηχανήματα. Σ' αυτήν την εφαρμογή το δίκτυο με τα honeyd virtual hosts συγκεντρώνουν την κίνηση από τη δραστηριότητα των worms, η οποία αν μελετηθεί θα δώσει πληροφορίες για το ποια worms είναι ενεργά και για τη συμπεριφορά τους. Αν μάλιστα τρέχει και κάποιο ids ή network sniffer θα καταγράψει το payload του worm.

4 Κεφάλαιο 4: Συμπεράσματα

4.1 Συμπεράσματα

Ολοκληρώνοντας την παρούσα εργασία συμπεραίνουμε ότι .

- Τα honeypots μπορούν να αποτελέσουν ένα πολύ σημαντικό εργαλείο μάθησης. Η ενασχόληση μαζί τους παρέχει στην χρήστη τους ένα σημαντικό επίπεδο γνώσης για τις δικτυακές επιθέσεις, αλλά και τον τρόπο δράσης των κακόβουλων χρηστών.

- Το εύρος εφαρμογής των honeypots είναι αρκετά μεγάλο. Εκτός από μαθησιακούς σκοπούς, μπορούν να χρησιμοποιηθούν αποτελεσματικά ως εργαλεία παρακολούθησης του δικτύου ή ως εργαλεία αξιολόγησης των ήδη υπαρχόντων συστημάτων ασφαλείας. Η εγκατάσταση Honeypots με αυτό το σκοπό προϋποθέτει την διαχείριση τους από πιο έμπειρα άτομα που θα μπορέσουν να αξιολογήσουν τα αποτελέσματα που προκύπτουν.

- Κανένα δίκτυο δεν μπορεί να χαρακτηριστεί ασφαλές, όσο και αν φαίνεται, αν δεν έχουμε σαφή στοιχεία για το τι είδους δεδομένα λαμβάνει. Τα honeypots είναι ικανά να μας παρέχουν αυτού του είδους την πληροφόρηση.

- Ο όγκος των δεδομένων που παράγουν τα honeypots (τα υψηλής αλληλεπίδρασης) είναι αρκετά αυξημένος και απαιτεί πολύ χρόνο για ανάλυση. Γι’ αυτό το λόγο η συντήρηση δικτύων honeypots αποτελεί, συνήθως, αποκλειστική εργασία για ένα η περισσότερα άτομα. Θα πρέπει λοιπόν η ανάπτυξη εκτεταμένων δικτύων honeypots να γίνεται με γνώμονα την επάρκεια σε εργατικό δυναμικό.

Παράρτημα

Παρακάτω παρατίθενται ολόκληρο το αρχείο παραμετροποίησης του Honeyd, “honeyd.conf”:

```
#set up 2 pc
#route entry 10.0.0.1
#route 10.0.0.1 link 10.0.0.0/24

#set up a network with one router
#route entry 10.0.0.100
#route 10.0.0.100 add net 10.0.0.0/16 10.0.1.100
#route 10.0.0.100 link 10.0.1.0/24
#route 10.0.1.100 link 10.0.0.0/16

#bind 10.1.1.53 to eth3

#create router
#set router personality "Cisco 7206 running IOS 11.1(24)"
#set router default tcp action reset
#add router tcp port 23 "scripts/router-telnet.pl"
#add router tcp port 113 open
#add router tcp port 117 open
#add router tcp port 114 reset
#add router tcp port 115 block
#add router tcp port 150 block

#bind 10.0.0.100 router

#create windows
#set windows personality "Microsoft Windows NT 4.0 SP3"
#add windows tcp port 80 "sh scripts/web.sh"
#add windows tcp port 22 "sh scripts/test.sh"
#add windows tcp port 139 open
#add windows tcp port 137 open
#add windows tcp port 135 block
#set windows uid 32767
#set windows default tcp action reset

#bind 10.0.0.51 windows
#bind 10.0.0.52 windows
#bind 10.1.0.51 windows
#bind 10.1.0.52 windows

#bind 10.0.1.100 router
#bind 10.1.0.100 router
#bind 10.0.0.200 router
#bind 10.1.0.200 router
```

Ερμηνεία αγγλικών όρων

Cracker:

Αυτός που διεισδύει σε υπολογιστικά συστήματα, με κακόβουλη πρόθεση. Ο σωστός όρος για τον κακόβουλο χρήστη - επιτιθέμενο (βλέπε και hacker παρακάτω).

False negatives:

Τα false negatives είναι οι περιπτώσεις επιθέσεων τις οποίες το IDS δεν κατάφερε μετά από την εξέταση τους να τις επισημάνει. Τα false negatives συνήθως προκύπτουν από κακή ρύθμιση του IDS ή από την εμφάνιση μίας νέας επίθεσης για την οποία δεν υπάρχει προηγούμενη γνώση.

False positives:

Τα false positives είναι οι λανθασμένες επισημάνσεις που παράγει ένα IDS, όταν ανιχνεύσει κάποιο γεγονός σαν περίπτωση πιθανής επίθεσης, ενώ στην πραγματικότητα δεν είναι.

Hacker:

Ο όρος με την πρωταρχική του σημασία αναφέρεται κατά βάση σε προγραμματιστές με υψηλές γνώσεις που μπορούσαν να μεταβάλλουν προγράμματα, αλλάζοντας την φυσιολογική τους λειτουργία. Παρ' όλα αυτά, η έννοια του hacker έχει αποκτήσει πολύ κακή φήμη. Ο ρόλος των hackers έχει συνδεθεί με το ηλεκτρονικό έγκλημα. Η κλασσική έννοια του hacker όμως δεν έχει σχέση με την παρανομία. Η λέξη προέρχεται από την αγγλική ρίζα "hack" που σημαίνει το κόψιμο και την επεξεργασία ξύλου. Ο hacker έχει τη δυνατότητα να "πελεκεί" εφαρμογές, προγράμματα και δίκτυα με πολύ περίτεχνο και έξυπνο τρόπο.

IDS:

Συστήματα Ανίχνευσης Επιθέσεων. Έχουν ως σκοπό τους την ανίχνευση ενδεχόμενων παραβιάσεων στα συστήματα τα οποία επιβλέπουν (βλέπε κεφάλαιο 1).

Keylogger:

Keylogger είναι ένα πρόγραμμα καταγραφής (συνήθως software αλλά υπάρχουν και πολλές hardware υλοποιήσεις) όλων των keystrokes που γίνονται σε ένα σύστημα.

Keystrokes:

Το Keystrokes αντιπροσωπεύει μια κεντρική δράση στο πληκτρολόγιο, ή την ισοδύναμη συσκευή εισαγωγής και χρησιμοποιούνται για να καθορίσουν τα υψηλού επιπέδου γεγονότα δράσης

Nmap:

Το nmap (Network Mapper) είναι ένα από τα πιο γνωστά security scanners εργαλεία που χρησιμοποιούνται σήμερα, τόσο από την πλευρά των αναλυτών ασφαλείας (penetration testers) όσο και από την πλευρά των επιτιθέμενων. Δημιουργήθηκε το 1997 από τον Fyodor Vaskovich και είναι ανοιχτού λογισμικού (η τελευταία του έκδοση είναι η 5.5).

Passive fingerprinting:

Ως παθητικό fingerprinting ορίζεται η διαδικασία κατά την οποία γίνεται αναγνώριση ενός απομακρυσμένου συστήματος χωρίς ωστόσο την χρήση άμεσων συνδέσεων με αυτόν. Αυτό συνήθως επιτυγχάνεται με την λήψη και ανάλυση πακέτων που φεύγουν από το εν λόγω μηχάνημα. Όπως γίνεται κατανοητό η τεχνική αυτή υπερέχει ως προς τα ίχνη που αφήνει, αφού δεν εγκαθιδρύονται συνδέσεις με τον υπολογιστή στόχο.

Γενική Βιβλιογραφία - Σύνδεσμοι

- Roshen Chandran, Sangita Pakala, Simulating Networks with Honeyd
- Honeynet Project, Know Your Enemy: GenII Honeynets
- Θ. Κομνηνός, Π. Σπυράκης: «Ασφάλεια Δικτύων & Υπολογιστικών Συστημάτων», 2002 Ελληνικά Γράμματα
- Andrew S. Tanenbaum: Δίκτυα Υπολογιστών 4Η Έκδοση
- New Riders Publishing: Network Intrusion Detection 3rd Edition
- Ασφάλεια Πληροφοριακών Συστημάτων, Σωκράτης Κ. Κάτσικας, Δημήτρης Γκρίτζαλης, Στέφανος Γκρίτζαλης
- Ασφάλεια Δικτύων Υπολογιστών, Σωκράτης Κ. Κάτσικας, Δημήτρης Γκρίτζαλης, Στέφανος Γκρίτζαλης
- Hacking Exposed: Network Security Secrets and Solutions, Sixth Edition by
- Stuart McClure, Joel Scambray, and George Kurtz, 2009
- Know Your Enemy: Learning about Security Threats, the honeynet project
- The Art of Intrusion - The Real Stories behind the Exploits of Hackers, Intruders and Deceivers, Kevin Mitnick
- Underground Hacking Madness & Obsession on the Electronic Frontier, Suelette Dreyfus
- Fighting internet worms with honeypots, oudot, www.securityfocus.com

- http://www.vmware.com
- http://www.honeyathome.org
- http://www.philippinehoneynet.org
- http://www.honeynet.org
- http://old.honeynet.org
- http://en.wikipedia.org
- http://el.wikipedia.org/wiki/Μοντέλο_αναφοράς_OSI

Βιβλιογραφικές Αναφορές

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

[2] Stefan Axelsson : **Intrusion Detection Systems : A Survey and Taxonomy**

[3] The Honeynet Progect:

http://www.islab.demokritos.gr/gr/html/ptixiaki_galex/kefalaio_1.pdf

[4] Stoll, C. "The Cuckoo's Egg: Tracking a Spy through the Maze of Computer Espionage". Pocket. October 3, 2000.

[5] Cheswick, B. *"An Evening with Berferd in Which a Cracker Is Lured, Endured, and Studied"*. San Francisco, 1992. (paper available from

http://www.cheswick.com/ches/papers/berferd.pdf)

[6] http://en.wikipedia.org/wiki/Perl

[7] http://www.honeynet.org/

[8] The Honeynet Project. *"Know your Enemy: Learning about Security Threats"*. Addison-Wesley Professional; 2nd edition. May 27, 2004.

[9] http://en.wikipedia.org/wiki/Honeypot_%28computing%29

[10] http://students.kennesaw.edu/~amull2/formal_report.htm

[11] Cheswick, B. *"An Evening with Berferd in Which a Cracker Is Lured, Endured, and Studied"*. San Francisco, 1992. (paper available from)

[12] www.specter.com

[13] http://theartemisproject.com/

[14] http://www.vmware.com

[15] http://www.cert.org/advisories/CA-2002-01.html

[16] http://www.keyfocus.net/kfsensor/

[17] http://dl.acm.org/

[18] Honeypots: Catching the Insider Threat, Lance Spitzner Honeypots: Simple, Cost-Effective Detection, Lance Spitzner

[19] http://labrea.sourceforge.net/

[20] Honeytokens: The Other Honeypot, Lance Spitzner

[21] http://www.guardian.co.uk/technology/2009/nov/03/google

[22] http://en.wikipedia.org

[23] http://www.honeyd.org

[24] Simulating networks with honeyd (Chandran, Pakala)

[25] Πληροφορίες για το πρωτόκολλο GRE RFC 1701, October 1994, Generic Routing Encapsulation (GRE)

[26] Virtual Honeypots: From Botnet Tracking to Intrusion Detection, Niels Provos, Thorsten Holz

[27] www.insecure.org

[28] Douglas E. Comer, *Internetworking with TCP/IP, Volume 1: Principles, Protocols and Architecture*