ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

Εγκατάσταση, Παραμετροποίηση και Πιλοτική Λειτουργία ενός Honeypot Server

Μαρινέλλα Μπίλα



Αθήνα, 2019



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

Τριμελής Εξεταστική Επιτροπή

Καμαλάκης Θωμάς (Επιβλέπων)
Αναπληρωτής, Πληροφορικής και Τηλεματικής, Χαροκόπειο Πανεπιστήμιο

Λιμνιώτης Κωνσταντίνος
Συνεργάτης, Πληροφορικής και Τηλεματικής, Χαροκόπειο Πανεπιστήμιο

Ριζομυλιώτης Παναγιώτης
Επίκ. Καθηγητής, Πληροφορικής και Τηλεματικής, Χαροκόπειο Πανεπιστήμιο

Η Μαρινέλλα Μπίλα

δηλώνω υπεύθυνα ότι:

1) Είμαι ο κάτοχος των πνευματικών δικαιωμάτων της πρωτότυπης αυτής εργασίας και από όσο γνωρίζω η εργασία μου δε συκοφαντεί πρόσωπα, ούτε προσβάλει τα πνευματικά δικαιώματα τρίτων.

2) Αποδέχομαι ότι η ΒΚΠ μπορεί, χωρίς να αλλάξει το περιεχόμενο της εργασίας μου, να τη διαθέσει σε ηλεκτρονική μορφή μέσα από τη ψηφιακή Βιβλιοθήκη της, να την αντιγράψει σε οποιοδήποτε μέσο ή/και σε οποιοδήποτε μορφότυπο καθώς και να κρατά περισσότερα από ένα αντίγραφα για λόγους συντήρησης και ασφάλειας.

Ευχαριστίες

Η παρούσα πτυχιακή εργασία με θέμα «Εγκατάσταση, Παραμετροποίηση και πιλοτική Λειτουργία ενός Honeypot Server» πραγματοποιήθηκε στο πλαίσιο της πτυχιακής εργασίας του Τμήματος Πληροφορικής και Τηλεματικής του Χαροκοπείου Πανεπιστημίου το έτος 2019.

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κ.Θωμά Καμαλάκη, για την ευκαιρία που μου έδωσε να ασχοληθώ με ένα ιδιαίτερα ενδιαφέρον θέμα και για την καθοδήγηση που μου παρείχε. Τον ευχαριστώ θερμά για την βοήθειά του και τις ωφέλιμες παρατηρήσεις του κατά τη διάρκεια της συγγραφής του θέματος αυτού.

Ακόμη, θα ήθελα να ευχαριστήσω όλους καθηγητές που με δίδαξαν κατά τα τέσσερα χρόνια φοίτησής μου, και με εφοδίασαν με τις απαραίτητες γνώσεις για να ανταπεξέλθω ισάξια στις απαιτήσεις του αντικειμένου.

Επίσης, θα ήθελα να ευχαριστήσω τους φίλους και συμφοιτητές μου, που μου συμπαραστάθηκαν αυτά τα χρόνια και ήταν κοντά μου στις χαρές αλλά και σε όποιες δυσκολίες συνάντησα.

Σε αυτό το σημείο, θα ήθελα να ευχαριστήσω τον συνοδοιπόρο μου στη ζωή, Δημήτρη, που είναι πάντα δίπλα μου και πιστεύει σε μένα και στις δυνατότητες μου, που με παρότρυνε σε όλη τη διάρκεια της συγγραφής μου και που με βοήθησε με όποιον τρόπο μπορούσε για να φτάσω σε αυτό τον σταθμό της ζωής μου.

Τέλος, θα ήθελα να πω ένα μεγάλο «ευχαριστώ» στην οικογένεια μου, που με στηρίζει και με εμπιστεύεται, όλα αυτά τα χρόνια των σπουδών μου. Αλλά πιο πολύ τους ευχαριστώ, που μου έδωσαν τα εφόδια για να γίνω ένας σωστός και καλύτερος Άνθρωπος.

Πίνακας Περιεχομένων

Ευχαριστίες 3
Πίνακας Περιεχομένων 4
Περίληψη στα Ελληνικά 6
Abstract 7
Λίστα εικόνων 8
Λίστα πινάκων 9
Συντομογραφίες 10
Κεφάλαιο 1 11
Εισαγωγή 11
1.1 Κίνητρο 11
1.2 Στόχοι πτυχιακής εργασίας 11
Κεφάλαιο 2 12
Βασικές αρχές των Δικτύων 12
2.1. Πρωτόκολλα 12
2.1.1 Πακέτα 12
2.1.2 Επίπεδα Πρωτοκόλλων 13
2.2 Internet Protocol (IP) 14
2.3 Mac Addresses 14
2.4. ARP 14
2.5 Θύρες (Ports) 14
2.6 Τείχη Προστασίας (Firewalls) 15
Κεφάλαιο 3 16
Θεωρητικό υπόβαθρο 16
3.1 Honeypots 16
3.1.1 Honeyd 17
Σχεδιασμός του Honeyd 17
Αρχιτεκτονική του Honeyd 18
Μηχανή-Προσωπικότητα 19
Αρχεία Καταγραφής 19
3.1.2 Kippo 20
Kippo Directories 20
Προϋποθέσεις 21
3.1.3 Cowrie 21
Cowrie Directories 21
Προϋποθέσεις 22
3.2 Nmap 22
3.3 Επιθέσεις 22
3.3.1 Syn attack 22
3.3.2 Smurf attack 24
3.3.3. Xmas attack 24
3.3.4 Hydra 25
3.3.5 NCrack 25
3.3.6.Medusa 26
Κεφάλαιο 4 27
Υλοποίηση των Honeypots 27
4.1 Εγκατάσταση του Honeyd, διαμόρφωση, και επίθεση 27
4.1.1 Ρύθμιση της ψεύτικης εικονικής μηχανής Windows χρησιμοποιώντας το HoneyD 28

4.1.2 Σάρωση της λειτουργικότητας του HoneyD από εκτέλεση εκτύπωσης Nmap Foot31
4.1.3 Δοκιμάζοντας τη λειτουργικότητα του HoneyD μέσω SYN Attack 31
4.1.4 Αρχεία καταγραφής 32
4.2 Εγκατάσταση του Kippo, διαμόρφωση, επίθεση 35
4.2.1 Δοκιμάζοντας τη λειτουργικότητα του HoneyD μέσω Medusa Attack..... 38
4.3 Εγκατάσταση του Cowrie, διαμόρφωση, επίθεση 40
4.3.1 Σάρωση της λειτουργικότητας του Cowrie από εκτέλεση εκτύπωσης Nmap Foot. 42
4.3.2 Δοκιμάζοντας τη λειτουργικότητα του Cowrie μέσω Brute Forcing Passwords με Hydra, Ncrack και Medusa 43
4.3.2.1. Brute Force επίθεση με Hydra 43
Brute Force επίθεση με Ncrack..... 44
Brute Force επίθεση με Medusa 44
Κεφάλαιο 5 46
Wireshark..... 46
Κεφάλαιο 6 50
6.1 Συμπεράσματα 50
6.2 Μελλοντική έρευνα..... 50
Βιβλιογραφία 52
Σχετικά με τον συγγραφέα 53

Περίληψη στα Ελληνικά

Η τεχνολογία αναπτύσσεται με ταχύτατους ρυθμούς και το Internet χρησιμοποιείται από ολοένα και περισσότερους ανθρώπους παγκοσμίως. Οι δυνατότητες που παρέχει το διαδίκτυο -επικοινωνία, online αγορές, ανταλλαγή δεδομένων, υπηρεσίες, διασκέδαση- έχουν μετατρέψει τον κόσμο μας , σε έναν διαδικτυακό κόσμο, που επικοινωνεί και αλληλοεπιδρά πιο γρήγορα και εύκολα μέσω αυτού.

Ωστόσο, όσο αναπτύσσονται οι τεχνολογίες και το διαδίκτυο, τόσο περισσότερο αυξάνονται οι κίνδυνοι που εγκυμονούν. Ο διαδικτυακός κόσμος μπορεί να γίνει ευάλωτος και προσιτός σε επιθέσεις από κακόβουλους χρήστες, αν δεν ληφθεί υπόψη η ασφάλεια και η συνεχής βελτίωσή της.

Παράλληλα με ένα τείχος προστασίας και άλλες λύσεις ασφάλειας που βοηθούν στην προστασία ενός δικτύου από τους κακόβουλους χρήστες, μπορούν να χρησιμοποιηθούν και κάποιες πρόσθετες προστασίες ασφάλειας, όπως τα Honeypots.

Τα honeypots είναι συστήματα συνδεδεμένα με το δίκτυο που έχουν σχεδιαστεί για να προσελκύουν τους κακόβουλους χρήστες και να αποκτούν πληροφορίες σχετικά με τον τρόπο λειτουργίας των εγκληματιών στον κυβερνοχώρο.

Στη πτυχιακή αυτή, θα παρουσιαστεί ένα αληθινό σενάριο επίθεσης σε πληροφοριακό σύστημα όπως αυτά του Χαροκοπείου Πανεπιστημίου, χρησιμοποιώντας Honeypot Servers. Συγκεκριμένα, θα παρουσιαστεί η εγκατάσταση, η παραμετροποίηση και η πιλοτική λειτουργία των Honeypot Servers, με σκοπό την ανάλυση των κυβερνητικών επιθέσεων και την προστασία των κεντρικών υπηρεσιών. Θα χρησιμοποιηθούν εργαλεία ανοιχτού λογισμικού και θα απαιτηθεί η παραμετροποίηση του διαχειριστικού περιβάλλοντος, καθώς και η εξαγωγή στατιστικών στοιχείων σχετικά με τις απειλές. Η εγκατάσταση των servers θα πραγματοποιηθεί σε περιβάλλον Linux -Ubuntu 18.04.2, ενώ οι επιθέσεις θα πραγματοποιηθούν από Kali Linux 2019.1. Τέλος, και τα δύο λειτουργικά συστήματα, έχουν εγκατασταθεί σε Oracle VM VirtualBox.

Λέξεις κλειδιά: honeypot, honeyd, kippo, cowrie, επιθέσεις

Abstract

Technology is growing rapidly, and the Internet is being used by more and more people around the world. The capabilities of the Internet - communication, online shopping, data exchange, services, entertainment - have transformed our world into an online world that communicates and interacts more quickly and easily through it.

However, as technologies and the internet are growing, the more risks they pose. The internet world can become vulnerable and accessible to attacks by malicious users, if security and its continuous improvement are not taken into consideration.

Along with a firewall and other security solutions that help protecting a network from malicious users, some additional security protections, such as Honeypots, can also be used. Honeypots are network-connected systems designed to attract malicious users and obtain information about how cybercriminals operate.

In this dissertation, we will present a real scenario of an attack on an information system such as those of Harokopion University using Honeypot Servers. In particular, Honeypot Servers will be installed, configured and operated to analyze government attacks and protect central services. Open-source tools will be used and customization of the management environment will be required, as well as the extraction of statistics of threats. The installation of the servers will take place on Linux-Ubuntu 18.04.2 environment, while the attacks will be carried out by Kali Linux 2019.1. Finally, both operating systems are installed in Oracle VM VirtualBox.

Keywords: honeypot, honeyd, kippo, cowrie, attacks

Λίστα εικόνων

Εικόνα 1. Δομή πακέτου. 12
Εικόνα 2. Δομή επικεφαλίδας του IP πρωτοκόλλου. 13
Εικόνα 3. Πως χρησιμοποιούνται οι διευθύνσεις MAC. 14
Εικόνα 4. Εικονική απεικόνιση της σχεδίασης του Honeyd. 17
Εικόνα 5. Εικονική απεικόνιση της αρχιτεκτονικής του Honeyd 19
Εικόνα 6. Εκτέλεση nmap εντολής για την σάρωση της IP διεύθυνσης 192.168.56.122. 22
Εικόνα 7. Επικοινωνία client-server με TCP πρωτόκολλο. 23
Εικόνα 8. Επίθεση SYN. 23
Εικόνα 9. Επίθεση Smurff. 24
Εικόνα 10. Τα μηχανήματα που χρησιμοποιηθήκαν για την εγκατάσταση του Honeyd. 28
Εικόνα 11. Αρχείο διαμόρφωσης ψεύτικης προσωπικότητας – honeyd configuration file. 29
Εικόνα 12. Είσοδος στο φάκελο honeyd. 30
Εικόνα 13. Εκτέλεση του Honeyd. 30
Εικόνα 14. Σάρωσης διεύθυνσης 192.168.56.139 που δόθηκε στον honeypot διακομιστή. 31
Εικόνα 15. Επίθεση SYN από Kali Linux στον honeypot με διεύθυνση 192.168.56.139. 32
Εικόνα 16. Καταγραφή των πακέτων και των συνδέσεων στο αρχείο mylogs.log. 32
Εικόνα 17. Αποστολή πακέτων από μηχάνημα Linux 192.168.56.109 33
Εικόνα 18. Απεικόνιση του αρχείου καταγραφής του honeyd. Αναγράφονται οι κινήσεις από το μηχάνημα Kali με IP 192.168.56.109. 33
Εικόνα 19. Αναζήτηση στο αρχείο mylogs για ότι περιέχει tcp. 34
Εικόνα 20. Αναζήτηση στο αρχείο syslog για ότι περιέχει tcp. 34
Εικόνα 21. Αναζήτηση στο αρχείο mylogs για ότι περιέχει την IP 192.168.56.161. 35
Εικόνα 22. Αναζήτηση στο αρχείο syslog για ότι περιέχει την IP 192.168.56.161 και udp. 35
Εικόνα 23. Εγκατάσταση απαραίτητου λογισμικού για τον Kippo honeypot. 36
Εικόνα 24. Δημιουργία non-root χρήστη για την εκτέλεση του Kippo. 36
Εικόνα 25 37
Εικόνα 26. Αρχείο διαμόρφωσης kippo.cfg. 37
Εικόνα 27. Εκτέλεση του Kippo 38
Εικόνα 28. Εντολή netstat για την απεικόνιση των ενεργών συνδέσεων και υπηρεσιών. 38
Εικόνα 29. IP διευθύνσεις Ubuntu και Kali Linux. 39
Εικόνα 30. Σάρωση διεύθυνσης θύματος 192.168.56.122. 39
Εικόνα 31. Επίθεση Medusa από Kali Linux στον Kippo server. 39
Εικόνα 32. Δημιουργία non-root χρήστη. 40
Εικόνα 33. Μετάβαση στον φάκελο cowrie-master. 41
Εικόνα 34. Εκτέλεση του Cowrie. 42
Εικόνα 35. Σάρωση της 192.168.56.122 από Kali Linux. 43
Εικόνα 36. Επίθεση Hydra από Kali Linux. 44
Εικόνα 37. Επίθεση NCrack από Kali Linux. 44
Εικόνα 38. Επίθεση Medusa από Kali Linux. 45
Εικόνα 39. Εκτέλεση Honeyd server. 46
Εικόνα 40. Σάρωση της διεύθυνσης όπου τρέχει ο Honeyd. 47
Εικόνα 41. Επίθεση SYN από Kali Linux με IP 192.168.56.109. 47
Εικόνα 42. Κίνηση δικτύου, πρωτόκολλο TCP και πόρτα 80 ή IMCP. 48
Εικόνα 43. Απεικόνιση κίνησης δικτύου σε ότι έχει να κάνει με την IP 192.168.56.142. 48
Εικόνα 44. Κίνηση πακέτων που δεν χρησιμοποιούν ARP και η udp πόρτα δεν είναι 53. 49
Εικόνα 45. I/O Γράφημα για στατιστική ανάλυση των κινήσεων του δικτύου. 49

Λίστα πινάκων

Πίνακας 1. Σκοπός και χαρακτηριστικά πρωτοκόλλων 13
Πίνακας 2. Συνηθέστερες υπηρεσίες-πόρτες. 15
Πίνακας 3. Διαφορές χαμηλής-υψηλής αλληλεπίδρασης honeypots. 16
Πίνακας 4. Χαρακτηριστικά και δυνατότητες των Cowrie και Kippo honeypots..... 21

Συντομογραφίες

ARP	Address Resolution Protocol
DHCP	Dynamic Host Control Protocol
DNS	Domain Name Service
DoS	Denial of Service
FARPD	Fake ARP User Space Daemon
HTTP	Hyper Text Transfer Protocol
IP	Internet Protocol
ICMP	Internet Control Message Protocol
IT	Information Technology
LAN	Local Area Network
MAC	Media Access Control
NAT	Network Address Translation
NIC	Network Interface Card
SSL	Secure Socket Layer
SSH	Secure Shell
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
URL	Uniform Resource Locator

Κεφάλαιο 1

Εισαγωγή

Σε αυτή την ενότητα περιγράφονται τα κίνητρα και οι στόχοι της παρούσας προπτυχιακής εργασίας, προκειμένου να γίνει μια επισκόπησή της.

1.1 Κίνητρο

Η ασφάλεια δικτύων είναι απαραίτητη στις μέρες μας, καθώς οι ρυθμοί ανάπτυξης της τεχνολογίας είναι πολύ μεγάλοι. Οι περισσότερες ενέργειες σήμερα, διαδραματίζονται μέσω του διαδικτύου. Καθώς λοιπόν, η τεχνολογία αναπτύσσεται με ταχύτατους ρυθμούς, όλο και μεγαλύτερη είναι η ανάγκη να προστατέψουμε τα δεδομένα μας στο διαδίκτυο από τους κακόβουλους χρήστες.

Στο σημείο αυτό, όπου η γνώση περί ασφάλειας δικτύων παίζει καθοριστικό ρόλο, η μελέτη των Honeypots, είναι ένα θέμα που αξίζει να ανέβει στο προσκήνιο. Πέρα από την ανάγκη όμως να εξασφαλίσουμε την ασφάλειά των δεδομένων μας, ο τομέας της ασφάλειας είναι πολύ ενδιαφέρων, αφού υπάρχει μεγάλη και ολοένα αναπτυσσόμενη ποικιλία τεχνολογιών και εργαλείων, που αξίζει κάποιος να μάθει να χρησιμοποιεί.

Τα Honeypots είναι ένας νέος, σημαντικός εξοπλισμός με απίστευτες προοπτικές. Τα honeypots είναι σε θέση να κάνουν τα πάντα, από την ανίχνευση φρέσκων επιθέσεων που δεν έχουν εμφανιστεί ποτέ, μέχρι την παρακολούθηση των προγραμματισμένων απατών των πιστωτικών καρτών και την κλοπή της ατομικότητας. Με την πάροδο του χρόνου, είναι διαθέσιμα πλέον, πολλά πράγματα που σχετίζονται με τους εμπορικούς honeypots και με τις - ανοιχτού λογισμικού-λύσεις.

Συνεπώς, η μελέτη και η εμβάθυνση στον τομέα της Ασφάλειας των Δικτύων, είναι ένα ενδιαφέρον θέμα που, όχι μόνο θα επιφέρει χρήσιμα αποτελέσματα και προοπτικές, αλλά θα αλλάξει και την αντίληψη των ανθρώπων που ασχολούνται με την Πληροφορική, σε πολλά πράγματα και καταστάσεις που συναντούν καθημερινά στο Διαδίκτυο.

1.2 Στόχοι πτυχιακής εργασίας

Τους στόχους της εν λόγω πτυχιακής εργασίας αποτελούν η παρουσίαση της απαραίτητης θεωρίας δικτύων, η ανάλυση και η εγκατάσταση τριών διαφορετικών -Medium και Low Interaction- Honeypots σε μηχανήματα Ubuntu σε Oracle VM VirtualBox, η παρουσίαση της θεωρίας των επιθέσεων από άλλο μηχάνημα που βρίσκεται στο ίδιο δίκτυο και μετέπειτα η υλοποίησή τους, και τέλος, η παρουσίαση ενός εργαλείου καθοριστικής σημασίας για την ανίχνευση της κίνησης μέσα στο δίκτυο και εντέλει την ασφάλεια των δικτύων.

Κεφάλαιο 2

Βασικές αρχές των Δικτύων

Σε αυτό το κεφάλαιο παρουσιάζονται βασικές αρχές των δικτύων.

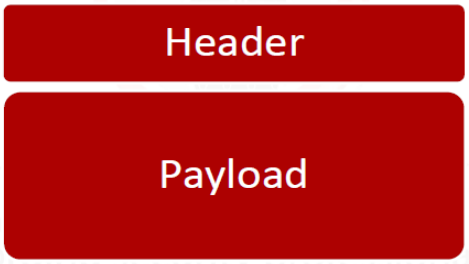
2.1. Πρωτόκολλα

Σε ένα δίκτυο υπολογιστών, οι μηχανές επικοινωνούν μεταξύ τους μέσω πρωτοκόλλων. Τα πρωτόκολλα αυτά εξασφαλίζουν ότι, διαφορετικοί υπολογιστές μπορούν να επικοινωνούν, χρησιμοποιώντας διαφορετικό υλικό και λογισμικό. Υπάρχει μια μεγάλη ποικιλία πρωτοκόλλων δικτύωσης στο Διαδίκτυο, καθένα από τα οποία έχει το δικό του σκοπό.

2.1.1 Πακέτα

Ο πρωταρχικός στόχος της δικτύωσης είναι η ανταλλαγή πληροφοριών μεταξύ δικτυωμένων υπολογιστών. Αυτές οι πληροφορίες μεταφέρονται με πακέτα. Τα πακέτα δεν είναι τίποτε άλλο παρά ροές μπιτ που λειτουργούν ως ηλεκτρικά σήματα στα φυσικά μέσα που χρησιμοποιούνται για τη μετάδοση δεδομένων. Το μέσο μπορεί να είναι ένα καλώδιο σε ένα δίκτυο LAN ή ο αέρας σε ένα δίκτυο WiFi. Αυτά τα ηλεκτρικά σήματα ερμηνεύονται στη συνέχεια ως δυαδικά ψηφία (μηδενικά και αυτά) που συνθέτουν τις πληροφορίες.

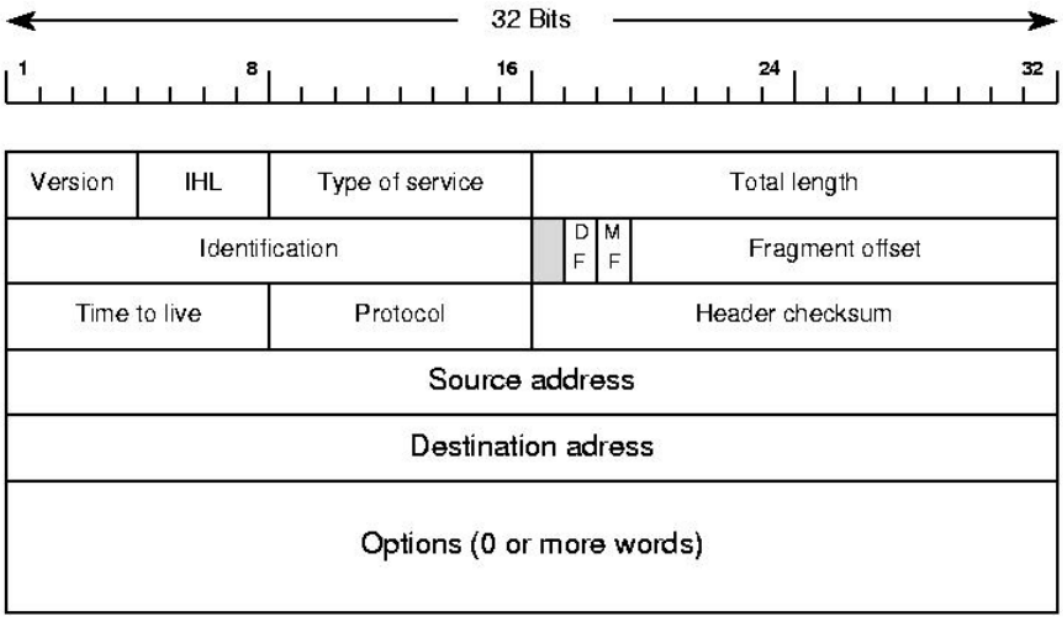
Κάθε πακέτο σε κάθε πρωτόκολλο έχει την μορφή που αναγράφεται στην Εικόνα 1:



Εικόνα 1. Δομή πακέτου.

Η κεφαλίδα(header) έχει δομή συγκεκριμένου πρωτοκόλλου: αυτό εξασφαλίζει ότι ο κεντρικός υπολογιστής λήψης μπορεί να ερμηνεύσει σωστά το payload και να χειριστεί τη συνολική επικοινωνία. Το payload είναι η πραγματική πληροφορία. Θα μπορούσε να είναι κάτι σαν μέρος ενός μηνύματος ηλεκτρονικού ταχυδρομείου ή το περιεχόμενο ενός αρχείου κατά τη λήψη.

Για παράδειγμα, η επιεφαλίδα του IP πρωτοκόλλου έχει μέγεθος τουλάχιστον 160 bits (20 bytes) και περιλαμβάνει πληροφορίες για να ερμηνεύσει το περιεχόμενο του IP πακέτου.



Εικόνα 2. Δομή επικεφαλίδας του IP πρωτοκόλλου.

Στην Εικόνα 2, φαίνεται η δομή της επικεφαλίδας του IP πρωτοκόλλου.Τα πρώτα τέσσερα bits αναγνωρίζουν την έκδοση IP. Σήμερα μπορούν να χρησιμοποιηθούν για να αντιπροσωπεύσουν την έκδοση IP 4 ή 6. Τα 32 bits που αρχίζουν στη θέση 128 αντιπροσωπεύουν τη διεύθυνση πηγή (source address). Τα ακόλουθα τέσσερα bytes αντιπροσωπεύουν τη διεύθυνση προορισμού (destination address). Χρησιμοποιώντας τις πληροφορίες στην κεφαλίδα, οι κόμβοι που συμμετέχουν στην επικοινωνία μπορούν να κατανοήσουν και να χρησιμοποιήσουν πακέτα IP.

2.1.2 Επίπεδα Πρωτοκόλλων

Στο προηγούμενο παράδειγμα, είδαμε την κεφαλίδα του IP, το πρωτόκολλο Internet. Υπάρχουν πολλά πρωτόκολλα, το καθένα για συγκεκριμένο σκοπό.
Τα επίπεδα (layers) των πρωτοκόλλων λειτουργούν το ένα πάνω στο άλλο και κάθε επίπεδο έχει το δικό του πρωτόκολλο. Στον Πίνακας 1 φαίνονται τα χαρακτηριστικά κάθε επιπέδου πρωτοκόλλου.

Πίνακας 1. Σκοπός και χαρακτηριστικά πρωτοκόλλων

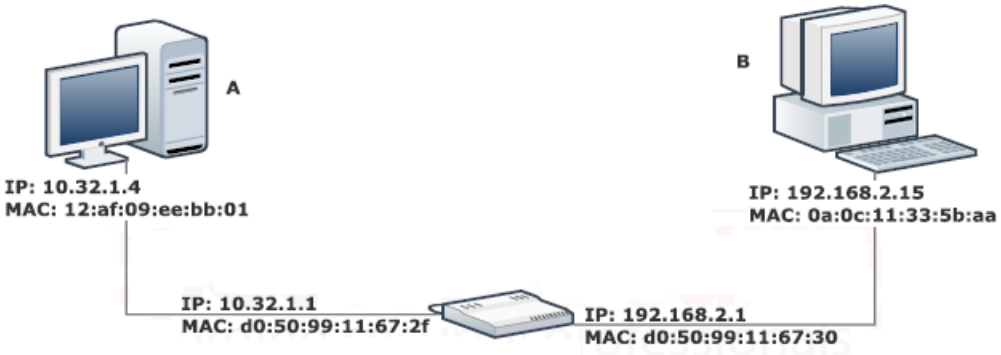
Σκοπός Πρωτοκόλλου	**Χαρακτηριστικά Πρωτοκόλλου**	**Επίπεδο Πρωτοκόλλου**
Ανταλλαγή ηλεκτρονικών μηνυμάτων, αρχείων ή πραγματοποόηση VoIP κλήσεων	Κάνει μία εφαρμογή να δουλεύει (mail client, FTP, browser,...)	Επίπεδο εφαρμογής (Application Layer)
Εγκαθίδρυση επικοινωνίας μεταξύ πελάτη-διακομιστή	Μεταφορά δεδομένων μεταξύ διεργασιών (client-server προγράμματα)	Επίπεδο μεταφοράς (Transport layer)
Αναγνώριση υπολογιστών σε ένα δίκτυο	Αναγνώριση κεντρικών υπολογιστών	Επίπεδο δικτύου (Network layer)
Μετάδοση πληροφοριών	Χρήση φυσικών μέσων για αποστολή πακέτων	Φυσικό επίπεδο (Physical layer)

2.2 Internet Protocol (IP)

Το Internet Protocol (IP) είναι το πρωτόκολλο που εκτελείται στο επίπεδο διαδικτύου της δομής του IP, γνωστό και ως TCP/IP. Το IP είναι υπεύθυνο για την παράδοση των datagrams (πακέτα IP που ονομάζονται datagrams) στους οικοδεσπότες που συμμετέχουν σε μια επικοινωνία. Το IP χρησιμοποιεί διευθύνσεις IP για τον εντοπισμό ενός κεντρικού υπολογιστή. Η πλειοψηφία των δικτύων τρέχει IP έκδοση 4 (IPv4). Μια διεύθυνση IPv4 αποτελείται από τέσσερις bytes ή οκτάδες (ένα byte είναι κατασκευασμένο από 8 bits).

2.3 Mac Addresses

Οι διευθύνσεις IP είναι το τρίτο επίπεδο (επίπεδο δικτύου) στο σχήμα διεύθυνσης που χρησιμοποιείται για τον εντοπισμό ενός κεντρικού υπολογιστή σε ένα δίκτυο. Οι διευθύνσεις MAC αντίθετα, εντοπίζουν μια κάρτα δικτύου (επίπεδο 2). Μια διεύθυνση MAC (Media Access Control) είναι επίσης γνωστή ως φυσική διεύθυνση. Οι διευθύνσεις MAC έχουν μήκος 48 bits (6 bytes) και εκφράζονται σε δεκαεξαδική μορφή (HEX), όπως για παράδειγμα 00: 11: AA: 22: EE: FF. Κάθε κεντρικός υπολογιστής σε ένα δίκτυο έχει MAC και IP διεύθυνση.



Εικόνα 3. Πως χρησιμοποιούνται οι διευθύνσεις MAC.

2.4. ARP

Όταν ένας κεντρικός υπολογιστής θέλει να στείλει ένα πακέτο σε έναν άλλο κεντρικό υπολογιστή, χρειάζεται να γνωρίζει την διεύθυνση IP και τη διεύθυνση MAC του προορισμού προκειμένου να δημιουργηθεί ένα κατάλληλο πακέτο. Οπότε, ο κεντρικός υπολογιστής πρέπει να γνωρίζει τις διευθύνσεις MAC των άλλων κόμβων δικτύου και μπορεί να τα μάθει χρησιμοποιώντας το πρωτόκολλο ARP (Address Resolution Protocol). Με το ARP, ένας κεντρικός υπολογιστής μπορεί να δημιουργήσει τη σωστή διεύθυνση IP – MAC διεύθυνση. Αυτό είναι ένα από τα πιο θεμελιώδη πρωτόκολλα κάθε μοντέρνου δικτύου.

2.5 Θύρες (Ports)

Οι θύρες (ports) χρησιμοποιούνται για τον εντοπισμό μιας ενιαίας διαδικασίας δικτύου σε ένα μηχάνημα. Για να εντοπιστεί σωστά μία διεργασία σε ένα δίκτυο, πρέπει να αναφέρεται το ζεύγος <IP>:<PORT>, όπως για παράδειγμα 192.168.5.3:80.

Οι θύρες 0-1023, οι πρώτες 1024, είναι γνωστές θύρες και χρησιμοποιούνται από τους διακομιστές για τις πιο συνηθισμένες υπηρεσίες. Για παράδειγμα, όταν ένα πρόγραμμα περιήγησης ιστού συνδέεται με ένα διακομιστή μέσω του HTTPS, ο χρήστης δεν χρειάζεται να καθορίσει την 443 ως θύρα προορισμού.

Οι πιο συνηθισμένες θύρες υπηρεσιών φαίνονται στον Πίνακας 2.

Πίνακας 2. Συνηθέστερες υπηρεσίες-πόρτες.

Υπηρεσία	*Θύρα*
SSH	22
HTTP	80
SFTP	115
FTP	21
Telnet	23
HTTPS	443
MySQL	3306

Τέλος, ένας δαίμονας είναι ένα πρόγραμμα που τρέχει μια υπηρεσία. Οι διαχειριστές του συστήματος μπορούν να αλλάξουν την διαμόρφωση του δαίμονα , αλλάζοντας τη θύρα που ακούει η υπηρεσία-σύνδεση. Αυτό γίνεται για να γίνει η αναγνώριση των υπηρεσιών πιο δύσκολη για τους κακόβουλους χρήστες (hackers).
Για παράδειγμα, θα μπορούσαμε να βρούμε έναν δαίμονα FTP που ακούει στη θύρα 4982 αντί για 21 ή SSH στη θύρα 8821.

2.6 Τείχη Προστασίας (Firewalls)

Τα τείχη προστασίας είναι εξειδικευμένα τμήματα λογισμικού που λειτουργούν σε έναν υπολογιστή ή σε ειδική συσκευή δικτύου. Χρησιμεύουν για να φιλτράρουν πακέτα που εισέρχονται και εξέρχονται από ένα δίκτυο. Τα τείχη προστασίας βοηθούν τους διαχειριστές συστημάτων και τους χρήστες της επιφάνειας εργασίας να ελέγχουν την πρόσβαση σε πόρους και υπηρεσίες δικτύου. Ένα τείχος προστασίας μπορεί να λειτουργήσει σε διαφορετικά επίπεδα του OSI μοντέλου, προκειμένου να παρέχει διαφορετικά χαρακτηριστικά και προστασίες.

Το πιο βασικό χαρακτηριστικό ενός τείχους προστασίας είναι το φιλτράρισμα πακέτων. Με το φιλτράρισμα πακέτων ένας διαχειριστής μπορεί να φιλτράρει τα πακέτα σύμφωνα με ορισμένα χαρακτηριστικά όπως Διεύθυνση IP προέλευσης (Source IP Address) , Διεύθυνση IP προορισμού (Destination IP Address), Πρωτόκολλο (Protocol) , Θύρα προέλευσης (Source Port) και Θύρα προορισμού(Destination Port).

Τα φίλτρα πακέτων επιθεωρούν την κεφαλίδα κάθε πακέτου για να επιλέξουν πώς να αντιμετωπίσουν το πακέτο. Οι πιο κοινές ενέργειες είναι:
• allow: επιτρέπουν στο πακέτο να περάσει.
• drop: ρίχνουν το πακέτο, χωρίς να έχει διαγνωστικό μήνυμα ο κεντρικός υπολογιστής πηγής πακέτων.
• deny: δεν αφήνουν πακέτο να περάσει, αλλά ενημερώνουν το μηχάνημα-πηγή.

Κεφάλαιο 3

Θεωρητικό υπόβαθρο

3.1 Honeypots

Ένα honeypot είναι ένα σύστημα που είναι σκοπίμως ανασφαλές και χρησιμοποιείται για την ανίχνευση και την προειδοποίηση για κακόβουλη δραστηριότητα των εισβολέων. Ο σκοπός ενός τέτοιου συστήματος είναι να αποκτήσει σημαντικές πληροφορίες για το πώς συμπεριφέρονται οι εισβολείς, προσελκύοντας τους να εισβάλουν στο σύστημά μας. Υπάρχουν τρία είδη honeypots - χαμηλής, μεσαίας και υψηλής αλληλεπίδρασης Honeypots-.

Στα χαμηλής αλληλεπίδρασης Honeypots, ένας μικρός αριθμός πρωτοκόλλων διαδικτύου και υπηρεσιών δικτύου είναι εξομοιωμένος, δίνοντας στον εισβολέα μια πολύ περιορισμένη πρόσβαση στο σύστημα, αρκετά για να τον εξαπατήσει. Δεν υπάρχει πραγματική αλληλεπίδραση μεταξύ του επιτιθέμενου και του συστήματος. Το πλεονέκτημα του honeypot χαμηλής αλληλεπίδρασης είναι η ευκολία και η απλή υποστήριξή του, καθώς απλώς αναπτύσσει το σύστημα και αρχίζει να συλλέγει τις πληροφορίες του εισβολέα.

Σε γενικές γραμμές, οι περισσότερες επιχειρήσεις προσομοιώνουν πρωτόκολλα όπως TCP και IP, τα οποία επιτρέπουν στον εισβολέα να νομίζει ότι συνδέεται με ένα πραγματικό σύστημα και όχι με ένα περιβάλλον honeypot. Ωστόσο, τέτοια honeypots δεν μπορούν να εγγυηθούν την καλύτερη απόδοση, δεδομένου ότι παρέχουν τη βασική προσομοίωση του συστήματος .

Από την άλλη πλευρά, τα Honeypots μεσαίας αλληλεπίδρασης προσφέρουν στους επιτιθέμενους την δυνατότητα να αλληλοεπιδρούν περισσότερο από τα Honeypots χαμηλής αλληλεπίδρασης, αλλά λιγότερο από τα Honeypots υψηλής αλληλεπίδρασης όπως θα δούμε παρακάτω στον Πίνακας 3. Ο επιτιθέμενος, σε αυτήν την περίπτωση, πιστεύει ότι βρίσκεται στο πραγματικό λειτουργικό σύστημα, μπορεί να περιηγηθεί, να αλλάξει ή να διαγράψει αρχεία ή και να αλλάξει τον κωδικό πρόσβασης του συστήματος.

Οι τελευταίοι και με τα περισσότερα πλεονεκτήματα Honeypots, Honeypots Υψηλής Αλληλεπίδρασης, δίνουν την ευκαιρία στους επιτιθέμενους να αλληλοεπιδρούν με το πλήρες σύστημα. Ο επιτιθέμενος εισβάλλει στο σύστημα, παίρνει τα αντίστοιχα αποτελέσματα από τις ενέργειές του και γενικά, μπορεί να θέσει σε κίνδυνο το μηχάνημα και να το ελέγξει. Με αυτήν την προσέγγιση, μπορούμε να λάβουμε πολλές πληροφορίες σχετικά με τις διαδικασίες και τη συμπεριφορά ενός εισβολέα. Ωστόσο, οι Honeypots Υψηλής Αλληλεπίδρασης έχουν κάποιο κίνδυνο, καθώς ο επιτιθέμενος μπορεί να καταχραστεί το honeypot και να επιτεθεί σε άλλα συστήματα στο Διαδίκτυο.

Πίνακας 3. Διαφορές χαμηλής-υψηλής αλληλεπίδρασης honeypots.

	Χαμηλής αλληλεπίδρασης	**Υψηλής αλληλεπίδρασης**
Βαθμός εμπλοκής	Χαμηλός	Υψηλός
Αληθινό λειτουργικό Σύστημα	Όχι	Ναι
Ρίσκο	Χαμηλό	Υψηλό
Συλλογή πληροφοριών	Συνδέσεις	Όλα

Γνώση για την εκτέλεση	Χαμηλή	Μεσσαία-Υψηλή
Χρόνος συντήρησης	Χαμηλός	Πολύ υψηλός

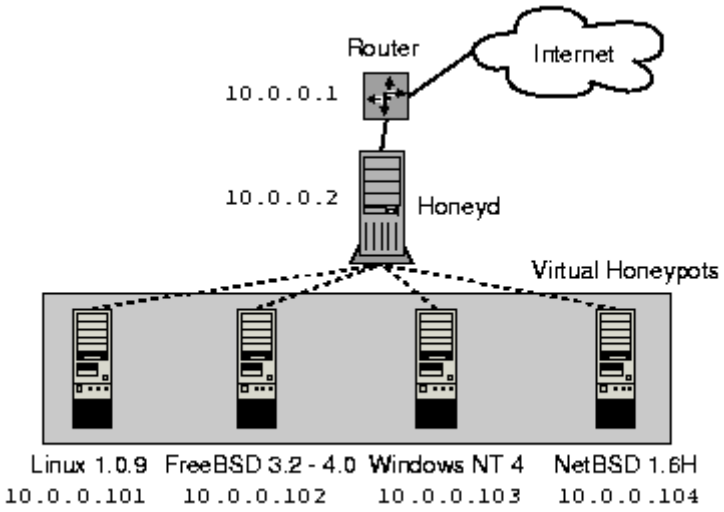
3.1.1 Honeyd

Το honeyd χρησιμοποιείται στα πλαίσια της ασφάλειας των πληροφοριακών συστημάτων, αφού είναι ένας μικρός «δαίμονας» που δημιουργήθηκε από τον Niels Provos, ο οποίος επιτρέπει στον χρήστη να εγκαταστήσει και να τρέξει πολλαπλούς εικονικούς υπολογιστές σε ένα δίκτυο υπολογιστών. Ο χρήστης μπορεί να διαμορφώσει αυτούς τους εικονικούς κεντρικούς υπολογιστές για να μιμηθεί διαφορετικούς τύπους εξυπηρετητών(servers). Το honeyd δίνει τη δυνατότητα σε έναν κεντρικό υπολογιστή να απαιτήσει πολλές διευθύνσεις σε ένα LAN για προσομοίωση δικτύου. Οι εικονικοί κεντρικοί υπολογιστές μπορούν να γίνουν pinged ή tracerouted και οποιοσδήποτε τύπος υπηρεσίας σε αυτούς μπορεί να προσομοιωθεί σύμφωνα με ένα configuration αρχείο, το οποίο μπορεί επίσης να κάνει proxy την υπηρεσία σε άλλο μηχάνημα.

Σχεδιασμός του Honeyd

Το Honeyd είναι ένα απλό λογισμικό που χρησιμοποιείται για να δημιουργήσει χιλιάδες εικονικούς υπολογιστές. Χιλιάδες διευθύνσεις IP με διαφορετικές υπηρεσίες δικτύου που εκτελούνται ξεχωριστά μπορούν να αναπτυχθούν από ένα μόνο μηχάνημα. Στην Εικόνα 4, φαίνεται ο σχεδιασμός του honeyd.

Στο σχεδιασμό του honeyd, ο σχεδιαστής αυτού του εργαλείου έχει κατά νου ότι το honeyd θα πρέπει να λειτουργεί στο επίπεδο του δικτύου που σημαίνει ότι ο εισβολέας που προσπαθεί να αλληλοεπιδράσει με αυτό το σύστημα περιορίζεται στο τρίτο επίπεδο του δικτύου. Με λίγα λόγια, η προσομοίωση του επιπέδου δικτύου δεν λαμβάνει υπόψη όλη την πτυχή του λειτουργικού συστήματος, και ο επιτιθέμενος δεν μπορεί να υπονομεύσει ολόκληρη την υπηρεσία. Παρόλα αυτά, το σύστημα μπορεί να παρακολουθεί τις ενεργές συνδέσεις και τις προσπάθειες που έχουν γίνει. Με αυτήν τη δυνατότητα, το TCP και το UDP, συμπεριλαμβανομένων των ICMP απαντήσεων, εξυπηρετούνται επίσης καλά.



Εικόνα 4. Εικονική απεικόνιση της σχεδίασης του Honeyd.

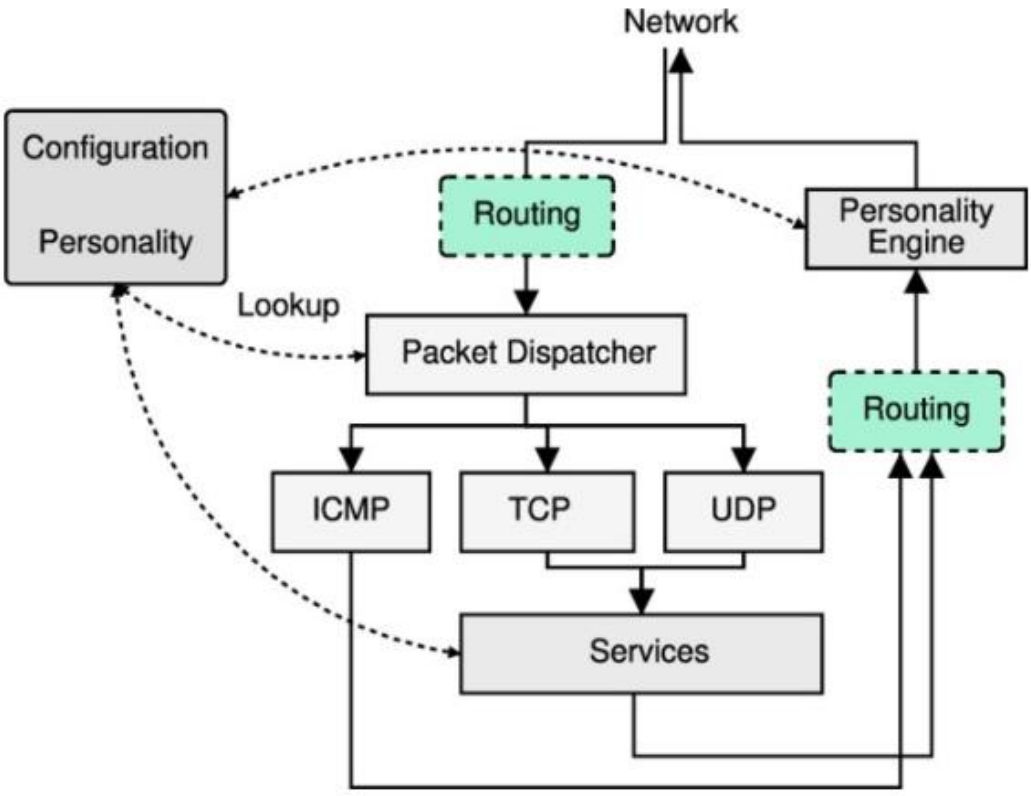
Αρχιτεκτονική του Honeyd

Η αρχιτεκτονική του Honeyd, όπως φαίνεται στην Εικόνα 5, περιέχει πολλά στοιχεία -έναν κεντρικό αποστολέα πακέτων, ένα στοιχείο δρομολόγησης (προαιρετικά), χειριστές πρωτοκόλλων, μία προσωπικότητα-μηχανή, και μια βάση δεδομένων για τη διαμόρφωση.

Ο κεντρικός αποστολέας πακέτων επεξεργάζεται τα εισερχόμενα πακέτα. Αρχικά, ελέγχει το μήκος ενός IP πακέτου και επιβεβαιώνεται ο έλεγχος του πακέτου αυτού. Η δομή της αρχιτεκτονικής του Honeyd αντιλαμβάνεται τα τρία σημαντικά πρωτόκολλα διαδικτύου TCP, UDP και ICMP. Τα πακέτα για άλλα πρωτόκολλα αποθηκεύονται και απλά απορρίπτονται. Ο αποστολέας πρέπει να ενημερωθεί για την βάση δεδομένων της διαμόρφωσης (configuration database) πριν επεξεργαστεί οποιοδήποτε πακέτο, για να μπορέσει να βρει μία διαμόρφωση honeypot (honeypot configuration) που να αντιστοιχεί στην IP διεύθυνση προορισμού (destination IP address). Αν δεν υπάρχει συγκεκριμένη διαμόρφωση, χρησιμοποιείται ένα προεπιλεγμένο πρότυπο. Λαμβάνοντας υπόψη τη διαμόρφωση, το πακέτο μεταβιβάζεται τον ειδικό χειριστή πρωτοκόλλου.

Τα περισσότερα ICMP αιτήματα (ICMP requests), υποστηρίζονται από τον ICMP διαχειριστή πρωτοκόλλου. Όλα τα honeypot configurations απαντούν σε echo requests και παράγουν απρόσιτα μηνύματα προορισμού (destination unreachable messages). Ο χειρισμός άλλων requests βασίζεται στα χαρακτηριστικά διαμόρφωσης (configuration characteristics).

Για το TCP και το UDP, η δομή της αρχιτεκτονικής, μπορεί να ξεκινήσει συνδέσεις με αυθαίρετες υπηρεσίες. Οι υπηρεσίες είναι εξωτερικές εφαρμογές, οι οποίες λαμβάνουν δεδομένα στο stdin και στο η έξοδος τους στέλνεται στο stdout. Η απόδοση μιας υπηρεσίας εξαρτάται πλήρως στην εξωτερική εφαρμογή. Το εργαλείο εξετάζει αν το πακέτο είναι μέρος μιας υπάρχουσας σύνδεσης όταν λαμβάνεται μια αίτηση σύνδεσης. Στην περίπτωση αυτή, κάθε νέα εγγραφή αποστέλλεται στην τρέχουσα εφαρμογή παροχής υπηρεσιών. Αν το πακέτο περιέχει μια αίτηση σύνδεσης, μια νέα διαδικασία έχει διαμορφωθεί για να τρέξει τη σωστή υπηρεσία. Η δομή υποστηρίζεται από υπό-συστήματα και εσωτερικές υπηρεσίες αντί για δημιουργία μίας νέας διαδικασίας για κάθε σύνδεση που έχει γίνει. Ένα υποσύστημα μπορεί να συνδεθεί σε θύρες, να αναγνωρίσει συνδέσεις και να ξεκινήσει την κυκλοφορία δικτύου. Μία εσωτερική υπηρεσία εκτελείται μέσα στο Honeyd, η οποία είναι ένα Python script, ενώ ένα υποσύστημα λειτουργεί ως μία εξωτερική διαδικασία. Οι εσωτερικές υπηρεσίες απαιτούν ακόμη λιγότερες κατοχές από αυτές των υποσυστήματα αλλά είναι σε θέση να δέχονται συνδέσεις και να μην τις ξεκινούν. Στην Εικόνα 5, φαίνεται η εικονική απεικόνιση της αρχιτεκτονικής αυτής.



Εικόνα 5. Εικονική απεικόνιση της αρχιτεκτονικής του Honeyd.

Μηχανή-Προσωπικότητα

Τα εργαλεία αποτύπωσης δακτυλικών αποτυπωμάτων όπως το Xprobe ή το Nmap εκτελούνται συνήθως από τους αντιπάλους για την κατάρτιση διαπιστευτηρίων σχετικά με ένα αντικειμενικό σύστημα. Για να τους κάνουμε να αναδυθούν σε έναν ανιχνευτή, το Honeyd προσομοιώνει τη συμπεριφορά δικτύου ενός συγκεκριμένου λειτουργικού συστήματος. Αυτό είναι ένα γνωστό χαρακτηριστικό ενός εικονικού honeypot. Διάφορα χαρακτηριστικά μπορούν να εκχωρηθούν σε διάφορα εικονικά honeypots. Η μηχανή-προσωπικότητα κάνει το επίπεδο δικτύου του honeypot να εκτελείται όπως καθορίζεται από τα χαρακτηριστικά εισάγοντας αλλαγές στα headers του πρωτοκόλλου κάθε εξερχόμενου πακέτου έτσι ώστε να ταιριάζει με τις προσωπικότητες του διαμορφωμένου λειτουργικού συστήματος.

Η δομή χρησιμοποιεί τη βάση δεδομένων δακτυλικών αποτυπωμάτων του Nmap ως αναφορά για μία TCP και UDP συμπεριφορά των προσωπικοτήτων, ενώ του Xprobe
ως αναφορά για τη συμπεριφορά ICMP της προσωπικότητας.

Αρχεία Καταγραφής

Η δομή της αρχιτεκτονικής του Honeyd, υποστηρίζει πολλούς τρόπους καταγραφής της δραστηριότητας μέσα στο δίκτυο. Μπορούν να δημιουργηθούν αρχεία καταγραφής σύνδεσης και ολοκληρωμένες συνδέσεις για κάθε πρωτόκολλο. Οι εφαρμογές υπηρεσιών είναι σε θέση να αναφέρουν πληροφορίες που θα καταγραφούν στο Honeyd χρησιμοποιώντας stderr.
Η δομή αυτή χρησιμοποιεί το syslog για να συσσωρεύσει τις πληροφορίες στο μηχάνημα.

3.1.2 Kippo

Μία πολλή συχνή μέθοδος για την απόκτηση πρόσβασης στο shell ενός λειτουργικού συστήματος απομακρυσμένα, είναι μέσω ενός πρωτόκολλου δικτύου -Secure Shell SSH-. Το πρωτόκολλο SSH παρέχει τη δυνατότητα απομακρυσμένης πρόσβασης σε μη ασφαλής περιοχή δικτύου. Για να επιτευχθεί αυτό, ο SSH διακομιστής είναι διαθέσιμος να παρέχει αυθεντική πρόσβαση στον χρήστη που χρησιμοποιεί το πρωτόκολλο για να αποκτήσει απομακρυσμένη πρόσβαση. Οι μέθοδοι που χρησιμοποιούν οι επιτιθέμενοι για να αποκτήσουν απομακρυσμένη πρόσβαση ενός SSH διακομιστή, μπορεί να είναι οποιαδήποτε διαθέσιμη τεχνική ή διαθέσιμο εργαλείο στην αγορά, ή ακόμα και ένα εργαλείο για brute force επιθέσεις ή ένα λεξικό-εργαλείο.

Για να μελετήσουμε τις ενέργειες των επιτιθέμενων αφού εισβάλλουν σε ένα σύστημα με SSH διακομιστή, μπορούμε να χρησιμοποιήσουμε τον Kippo honeypot ή τον Cowrie. Για τον δεύτερο, θα γίνει αναφορά παρακάτω. Ο Kippo SSH Honeypot είναι ένας Medium-interaction Honeypot, ο οποίος αποσκοπεί να καταγράψει πολλές από τις brute-force επιθέσεις, και ακόμα καλύτερα, να καταγράψει ολοκληρωτικά την αλληλεπίδραση που πραγματοποιείται στο τερματικό από τον επιτιθέμενο. Ο Kippo Honeypot προσομοιώνει ένα Ubuntu Linux σύστημα, το οποίο και βλέπει ο επιτιθέμενος αφού εισβάλλει και ενεργεί στο σύστημα. Ο επιτιθέμενος μπορεί να περιηγηθεί μέσα στο σύστημα αλλά δεν μπορεί να καταστρέψει τίποτα.

Τα Kippo Honeypots παρέχουν στους επιτιθέμενους την αίσθηση ότι βρίσκονται σε ένα πραγματικό SSH session στον διακομιστή. Αφού ο επιτιθέμενος εισβάλλει στο σύστημα παραβιάζοντας επιτυχώς τον κωδικό, μπορεί να περιηγηθεί στο «ψεύτικο» σύστημα. Κάθε μεμονωμένη εντολή του επιτιθέμενου αποθηκεύεται. Ακόμη, ο επιτιθέμενος μπορεί να κατεβάσει οποιοδήποτε αρχείο από το internet, το οποίο θα αποθηκευτεί στο «dl» directory του Kippo Honeypot. Γενικά, η κεντρική ιδέα της λειτουργίας του συγκεκριμένου honeypot είναι να δώσει στον επιτιθέμενο την εντύπωση ότι περιηγείται στο πραγματικό σύστημα.

Τέλος, ο Kippo Honeypot είναι αυτόνομος σε οποιοδήποτε λειτουργικό σύστημα, αφού έχει αναπτυχθεί σε python γλώσσα προγραμματισμού.

Kippo Directories

data: directory, όπου υπάρχουν διάφορα αρχεία, όπως η βάση για τους κωδικούς.
dl: directory, όπου αποθηκεύεται ό,τι έχει κατεβάσει ο επιτιθέμενος από το internet
honeyfs: directory, όπου μπορούν να σχεδιαστούν ψεύτικα directories, τα οποία περιέχουν όποια αρχεία που έχει πρόσβαση ο επιτιθέμενος
log: directory που παρέχει κάθε πληροφορία για τις επιθέσεις και τα sessions που δημιουργήθηκαν.
txtcmds: directory, όπου μπορούν να δημιουργηθούν εντολές που δεν είναι διαθέσιμος στον kippo διακομιστή.
utils: directory, που περιέχει το playlog.py, το οποίο χρησιμοποιείται για να ξανατρέξει το session.
start.sh: directory που παρέχει το start script του kippo
fs.pickle: εικονικό αρχείο συστήματος ανεπτυγμένο σε Python pickle.
kippo.cfg: το κυρίως αρχείο του διακομιστή που παίζει σημαντικό ρόλο , αφού όλες οι οδηγίες βρίσκονται εκεί.

Προϋποθέσεις

Το honeypot Kippo πρέπει να χρησιμοποιεί τα παρακάτω λογισμικά.

- Ένα λειτουργικό σύστημα (Εδώ χρησιμοποιούμε Ubuntu)
- Python 2.5+
- Twisted 8.0+
- PyCrypto
- Zope Interface

3.1.3 Cowrie

Ο Cowrie honeypot, είναι όπως και ο Kippo, ένας Medium-interaction SSH honeypot, ο οποίος καταγράφει τις brute force επιθέσεις και όλη τη δραστηριότητα του επιτιθέμενου στο τερματικό. Ο cowrie honeypot είναι βασισμένος στον kippo honeypot και έχει κάποιες επιπλέον λειτουργίες και δυνατότητες, οι οποίες φαίνεται στον Πίνακας 4.

Πίνακας 4. Χαρακτηριστικά και δυνατότητες των Cowrie και Kippo honeypots.

	Cowrie	**Kippo**
Ψεύτικο σύστημα αρχείων με ικανότητα προσθήκης ή διαγραφής αρχείων. Συμπεριλαμβάνεται ένα ολοκληρωμένο ψεύτικο σύστημα αρχείων που μοιάζει με Debian 5.0.	✓	✓
Δυνατότητα προσθήκης ψεύτικων περιεχομένων στα αρχεία, με σκοπό την παγίδευση του επιτιθέμενου όταν θα χρησιμοποιήσει την εντολή cat για να διαβάσει αρχεία όπως το /etc/passwd. Σε αυτά τα ψεύτικα αρχεία, περιλαμβάνονται ελάχιστα περιεχόμενα.	✓	✓
Υποστηρίζει επίσης τη μορφή αρχείου UML για τα αποθηκευμένα αρχεία καταγραφής (session logs), τα οποία μπορούν να αναλυθούν μετά με τους αληθινούς τους χρόνους.	✓	✓
Συσσωρεύει όλα τα αρχεία που έχουν ληφθεί σε ολόκληρη την SSH session με την wget για μεταγενέστερη εξέταση.	✓	✓
Προσομοιώνει τον τερματισμό του SSH session. Όταν ο επιτιθέμενος πληκτρολογεί «exit» για να βγει από το session, δεν βγαίνει ουσιαστικά, αλλά παρέχεται ένα άλλο shell σαν το terminal, μέσω του οποίου συλλέγονται και άλλες πληροφορίες για τις δραστηριότητες του επιτιθέμενου.		✓
SFTP και SCP υποστήριξη για το ανέβασμα των αρχείων	✓	✓
Υποστήριξη SSH εντολών	✓	
Καταγραφή των προσπαθειών για TCP σύνδεση (ssh proxying)	✓	
Καταγραφή σε JSON για εύκολη επεξεργασία των logs.	✓	
Πλήθος επιπρόσθετων εντολών	✓	

Cowrie Directories

log/cowrie.log: αρχείο που περιέχει τις καταγραφές.
log/cowrie.json: αρχείο που περιέχει τις αλληλεπιδράσεις σε json format.
utils/playlog.py: χρησιμοποιείται για να ξανατρέξει το session.
utlis/createfs.py: χρησιμοποιείται για να δημιουργήσει το fs.pickle.

data/fs.pickle: ψεύτικο αρχείο συστήματος
honeyfs: περιεχόμενα αρχείων για το ψεύτικο αρχείο συστήματος, τα οποία είναι προσβάσιμα από τον επιτιθέμενο.

Προϋποθέσεις

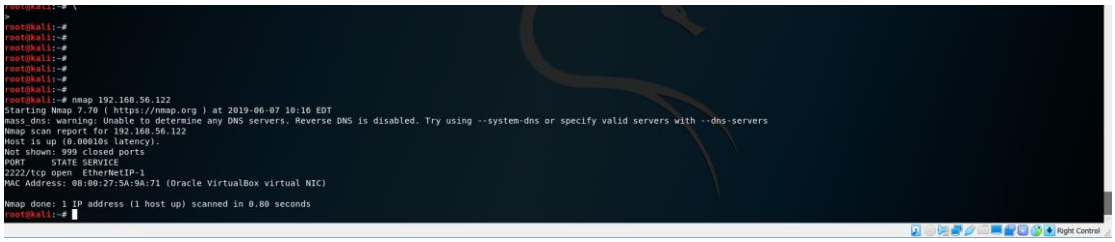
Το Cowrie Honeypot πρέπει να χρησιμοποιεί τα παρακάτω λογισμικά.

- Ένα λειτουργικό σύστημα (Εδώ χρησιμοποιούμε Ubuntu)
- Python 2.5+
- Twisted 8.0+
- PyCrypto
- pyasn1
- Zope Interface

3.2 Nmap

Το Nmap είναι ένα ανοιχτό εργαλείο εξερεύνησης του δικτύου, το οποίο χρησιμοποιείται για να σαρώνει γρήγορα μεγάλα δίκτυα αλλά και απλούς κεντρικούς υπολογιστές. Πολλοί κεντρικοί υπολογιστές συνδεδεμένοι στο διαδίκτυο μπορούν να σαρωθούν με την Nmap, και να βρεθεί ποιο λειτουργικό σύστημα χρησιμοποιούν και ποιες υπηρεσίες παρέχουν και ποιες πόρτες είναι ανοιχτές, για να συνδεθούν οι χρήστες. Στην Εικόνα 6, φαίνεται η σάρωση της IP 192.168.56.122 με την βοήθεια της nmap.
Η Nmap εκτελείται με δικαιώματα root χρήστη με την παρακάτω εντολή:

```
# nmap -sS -O -F <IP του υπολογιστή που θέλουμε να σαρώσουμε>
```



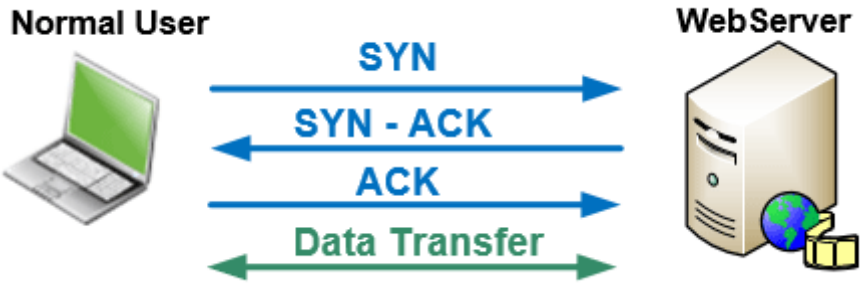
Εικόνα 6. Εκτέλεση nmap εντολής για την σάρωση της IP διεύθυνσης 192.168.56.122.

3.3 Επιθέσεις

3.3.1 Syn attack

Μία επίθεση SYN attack είναι μία επίθεση, στην οποία ο επιτιθέμενος χρησιμοποιεί το πρωτόκολλο TCP, για να στείλει πολλά πακέτα στο σύστημα-στόχο με σκοπό να «υπερχειλίσει» τις ουρές σύνδεσης και εντέλει να τις αναγκάσει να μην ανταποκρίνονται σε νόμιμα αιτήματα. Μία επίθεση SYN attack, ονομάζεται διαφορετικά TCP SYN attack ή SYN flood.

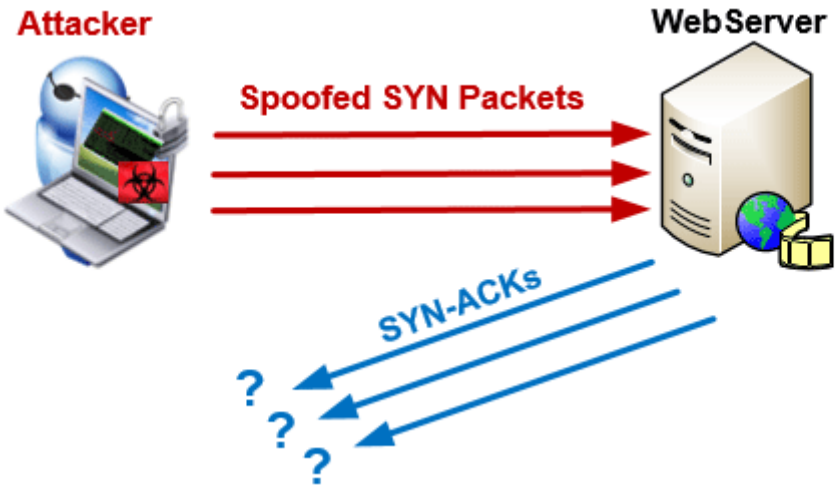
Πώς λειτουργεί η επίθεση SYN attack;
Όταν ένας υπολογιστής-πελάτης προσπαθεί να συνδεθεί στον διακομιστή χρησιμοποιώντας το TCP πρωτόκολλο, απαραίτητη προϋπόθεση πριν από την ανταλλαγή οποιασδήποτε πληροφορίας, είναι η πραγματοποίηση του three-way handshake. Από τη στιγμή που επιτυγχάνεται η χειραψία αυτή από τον υπολογιστή πελάτη, αποστέλλεται ένα πακέτο SYN στο διακομιστή, το οποίο έχει ένα SYN flag και έναν τυχαίο αριθμό ακολουθίας.



Εικόνα 7. Επικοινωνία client-server με TCP πρωτόκολλο.

Ύστερα, ο διακομιστής απαντάει, στέλνοντας το πακέτο SYN και ένα ACK flag με έναν άλλον τυχαίο αριθμό ακολουθίας. Ο αριθμός ACK είναι πάντα μια απλή αύξηση του SYN αριθμό που αποστέλλεται από τον υπολογιστή-πελάτη. Τελικά, ο υπολογιστής-πελάτης ολοκληρώνει το συγχρονισμό αποστέλλοντας ένα πακέτο ACK. Ο υπολογιστής-πελάτης συμπεριφέρεται ακριβώς όπως ο διακομιστής όταν στέλνει πακέτα ACK.

Κατά τη διάρκεια ενός SYN attack, ο επιτιθέμενος στέλνει πολλά πακέτα SYN στο διακομιστή χρησιμοποιώντας ψευδείς διευθύνσεις IP, αναγκάζοντας έτσι το διακομιστή να απαντάει(SYN-ACK), αφήνοντας μισο-ανοιχτές τις πόρτες, περιμένοντας απάντηση από έναν υπολογιστή, ο οποίος δεν υπάρχει.



Εικόνα 8. Επίθεση SYN.

Στην περίπτωση που ο επιτιθέμενος κάνει επιθέσεις χωρίς ψευδή διεύθυνση IP, θα διαμορφώσει τους κανόνες του τείχους προστασίας, έτσι ώστε να απορρίπτει τα πακέτα SYN-ACK πριν φτάσουν. Με αυτόν τον τρόπο, ο επιτιθέμενος, μπορεί εύκολα να κατακλύσει τους πόρους του στόχου, αφού στέλνει πολλά πακέτα SYN και δεν δέχεται απάντηση(ACK). Σαν αποτέλεσμα, ο υπολογιστής-στόχος, προσπαθεί να διαχειριστεί την κίνηση, η οποία προκαλεί αύξηση της χρήσης της CPU και την κατανάλωση μνήμης, οδηγώντας στην εξάντληση των πόρων της CPU και της RAM. Στο σημείο αυτό, ο διακομιστής δεν είναι σε θέση να εξυπηρετήσει νόμιμες αιτήσεις πελατών και τελικά οδηγείται σε άρνηση εξυπηρέτησης.

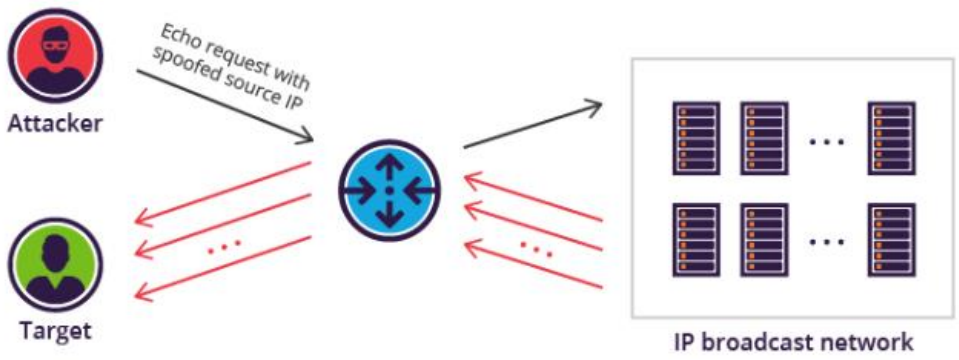
Πώς να επιτευχθεί μία επίθεση SYN attack με Kali Linux και hping3;

```
# sudo apt-get install hping3 (δεν χρειάζεται αν η επίθεση γίνεται με Kali Linux)
# hping3 -c 15000 -d 120 -S -w 64 -p 80 --flood --rand-source 192.168.1.159
```

3.3.2 Smurf attack

Η επίθεση Smurf, είναι κατά κάποιον τρόπο παρόμοια με τις πλημμύρες του ping, καθώς και οι δύο πραγματοποιούνται με την αποστολή πακέτων ICMP echo αιτημάτων. Παρόλα αυτά, ο Smurf είναι ένας φορέας επίθεσης που ενισχύει τις πιθανότητες βλάβης του συστήματος, αξιοποιώντας τα χαρακτηριστικά των δικτύων εκπομπής.

Πώς λειτουργεί η επίθεση Smurf attack;
Το κακόβουλο λογισμικό Smurf χρησιμοποιείται για τη δημιουργία ενός ψεύτικου αιτήματος Echo το οποίο περιέχει μια IP διεύθυνση πηγαίου κώδικα, η οποία είναι στην πραγματικότητα η διεύθυνση του διακομιστή προορισμού.Το αίτημα αποστέλλεται σε ένα ενδιάμεσο δίκτυο μετάδοσης IP και ύστερα μεταδίδεται σε όλους τους κεντρικούς υπολογιστές του δικτύου στο δίκτυο. Κάθε κεντρικός υπολογιστής αποστέλλει μια απάντηση ICMP στη διεύθυνση παραπλανητικής πηγής. Με την αποστολή αρκετών απαντήσεων ICMP, ο διακομιστής προορισμού «πέφτει».



Εικόνα 9. Επίθεση Smurff.

Πώς να επιτευχθεί μία επίθεση Smurf attack με Kali Linux και hping3;

```
# hping3-1 --flood -a 192.168.33.123 192.168.1.255
```

3.3.3. Xmas attack

Πώς λειτουργεί η επίθεση Xmas attack;
Τα πακέτα Xmas tree μπορούν να χρησιμοποιηθούν για να πραγματοποιήσουν μία επίθεση DoS, αξιοποιώντας το γεγονός ότι τα Xmas-tree πακέτα, απαιτούν πολλή περισσότερη επεξεργασία από τους δρομολογητές και τους τελικούς κεντρικούς υπολογιστές από ότι κάνουν τα συνηθισμένα πακέτα.

Τα Xmas tree πακέτα μπορούν εύκολα να εντοπιστούν από συστήματα ανίχνευσης εισβολών ή από πιο προηγμένα τείχη προστασίας. Από την άποψη της ασφάλειας του δικτύου, τα συγκεκριμένα πακέτα είναι πάντα ύποπτα και υποδεικνύουν μεγάλη πιθανότητα δραστηριοτήτων αναγνώρισης δικτύου.

Πώς να επιτευχθεί μία επίθεση Xmas attack με Kali Linux και hping3;

```
# sudo hping3 10.20.0.10 --flood --rand-source --destport 80 -c 25000 --syn --ack --fin
--rst --push --urg --xmas –ymas
```

3.3.4 Hydra

Η Ύδρα είναι ένα εργαλείο που σπάει τον κωδικό πρόσβασης. Στην ασφάλεια δικτύων (ασφάλεια IT), το σπάσιμο του κωδικού πρόσβασης είναι η μέθοδος κατά την οποία μαντεύετε κωδικούς πρόσβασης από τις βάσεις δεδομένων που είναι αποθηκευμένοι ή βρίσκονται υπό διαμετακόμιση από ένα σύστημα ή ένα δίκτυο υπολογιστών. Μία γενικότερη προσέγγιση και η προσέγγιση που χρησιμοποιεί η Ύδρα και πολλά άλλα παρόμοια εργαλεία ελέγχου αναφέρονται ως Brute Force.
Η βίαιη δύναμη απλά σημαίνει ότι το πρόγραμμα ξεκινά μια συνεχή ροή κωδικών πρόσβασης σε μια σύνδεση για να μαντέψουν τον κωδικό πρόσβασης. Όπως είναι γνωστό, η πλειονότητα των χρηστών έχουν σχετικά αδύναμους κωδικούς πρόσβασης και πολύ συχνά μπορούν να βρεθούν εύκολα. Λίγη κοινωνική μηχανική και η πιθανότητα να βρεθεί ο ακριβής κωδικός πρόσβασης για έναν χρήστη είναι μεγαλύτερη. Οι περισσότεροι άνθρωποι (ειδικά εκείνοι που δεν είναι IT επαγγελματίες), βασίζουν τους "μυστικούς" κωδικούς πρόσβασης σε πράγματα που δεν ξεχνάνε εύκολα. Αυτές οι λέξεις είναι συνήθως: τα αγαπημένα παιδιά, τα ονόματα των παιδιών, οι διευθύνσεις του δρόμου, ο τόπος γέννησης κλπ. Όλα αυτά είναι εύκολα διαθέσιμα μέσω των κοινωνικών μέσων ενημέρωσης. Έτσι ο επιτιθέμενος, μόλις συλλέξει αυτά τα δεδομένα, μπορεί τα μαζέψει μέσα σε μια "λίστα κωδικών".

Πώς να επιτευχθεί μία επίθεση Hydra;
Προσπάθεια πρόσβασης σαν root χρήστης χρησιμοποιώντας μία λίστα πιθανών κωδικών /usr/share/wordlists/metasploit/unix_passwords.txt στον ssh διακομιστή 192.168.1.123

```
root@kali:~# hydra -l root -P /usr/share/wordlists/metasploit/unix_passwords.txt -t
6 ssh://192.168.1.123
```

3.3.5 NCrack

Το NCrack είναι ένα γρήγορο εργαλείο παραβίασης για τον έλεγχο της ταυτότητας ενός δικτύου. Αναπτύχθηκε αρχικά, για να βοηθήσει τις εταιρείες να εξασφαλίσουν την ασφάλεια των δικτύων τους έναντι των αδύναμων κωδικών πρόσβασης, πραγματοποιώντας ελέγχους σε όλους τους υπολογιστές και στις συσκευές δικτύου.

Το NCrack σχεδιάστηκε χρησιμοποιώντας μία παρόμοια σύνταξη command line με αυτή του Nmap και μία δυναμική μηχανή που μπορεί να προσαρμόσει τη συμπεριφορά του βάσει απάντησης δικτύου. Επιτρέπει γρήγορο, αλλά αξιόπιστο έλεγχο πολλών κεντρικών υπολογιστών.

Ο χρήστης έχει πλήρη έλεγχο των λειτουργειών του δικτύου, και μπορεί να πραγματοποιήσει πολύ εξελιγμένες επιθέσεις bruteforcing, αλληλεπιδράσεις παρόμοιες με την Nmap και πολλές άλλες. Τα πρωτόκολλα που υποστηρίζονται περιλαμβάνουν SSH, RDP, FTP, Telnet, HTTP(S), POP3(S), IMAP, SMB, VNC, SIP, Redis, PostgreSQL, MySQL, MSSQL, MongoDB, Cassandra, WinRM και OWA.

Πώς να επιτευχθεί μία επίθεση Ncrack;
Προσπάθεια πρόσβασης με χρήστη root, στην πόρτα 22, χρησιμοποιώντας μία λίστα κωδικών 500-worst-passwords.txt, στον διακομιστή 10.10.10.10.

```
root@kali:# ncrack -p 22 --user root -P 500-worst-passwords.txt 10.10.10.10
```

3.3.6.Medusa

Στην ελληνική μυθολογία, η Μέδουσα ήταν ένα τέρας, γενικά περιγραφόμενο σαν φτερωτό ανθρώπινο θηλυκό με ζωντανά δηλητηριώδη φίδια στη θέση των μαλλιών.

Στα δίκτυα, η Μέδουσα, είναι ένα εργαλείο, όπως την Hydra και το NCrack, για την γρήγορη και παράλληλη σύνδεση σε έναν υπολογιστή. Στόχος της είναι να υποστηρίξει όσο το δυνατόν περισσότερες υπηρεσίες που επιτρέπουν την απομακρυσμένη πιστοποίηση ταυτότητας.

Με την Μέδουσα μπορούν να γίνουν παράλληλες δοκιμές με βάση τα threads που καθορίζονται, ενώ οι δοκιμές μπορούν να γίνουν με πολλούς κεντρικούς υπολογιστές, χρήστες ή κωδικούς ταυτόχρονα. Τα πρωτόκολλα που υποστηρίζονται περιλαμβάνουν SMB, HTTP, POP3, MS-SQL, SSHv2 και πολλά άλλα.

Πώς να επιτευχθεί μία επίθεση Medusa;
Προσπάθεια πρόσβασης με χρήστη root, χρησιμοποιώντας μία λίστα κωδικών 500-worst-passwords.txt, στον διακομιστή 10.10.10.10.

```
root@kali: # medusa -u root -P 500-worst-passwords.txt -h 10.10.10.10 -M ssh
```

Κεφάλαιο 4

Υλοποίηση των Honeypots

Στο κεφάλαιο αυτό, θα παρουσιαστεί η υλοποίηση τριών διαφορετικών honeypots -του Honeyd, του Kippo και του Cowrie-.

4.1 Εγκατάσταση του Honeyd, διαμόρφωση, και επίθεση

Ξεκινώντας, πρέπει να σημειωθεί ότι, ο αρχικός χρήστης δεν είναι root-χρήστης από προεπιλογή. Για αυτό και θα χρειαστεί να εκτελεστούν ,πολλές φορές, εντολές με το sudo μπροστά από κάθε εντολή.

Αρχικά, εκτελείται η ενημέρωση στο Ubuntu μηχάνημα με την παρακάτω εντολή:

```
#sudo apt-get update -y
```

Ύστερα, θα χρειαστεί να εγκατασταθεί η βιβλιοθήκη libdnet για ορισμένες δραστηριότητες δικτύου που εκτελεί το Honeyd:

```
# sudo apt-get install libdnet -y
```

Η συγκεκριμένη βιβλιοθήκη παρέχει μία βασική και κινητή διεπαφή σε πολλές ρουτίνες δικτύωσης χαμηλού επιπέδου που συμπεριλαμβάνουν:

- Τον χειρισμό των διευθύνσεων του δικτύου
- Φιλτράρισμα δικτύου
- Χειρισμό δικτύου και αναζήτηση διεπαφής
- Την σήραγγα του πρωτοκόλλου IP
- Την μετάδοση των πακέτων IP με πλαίσια Ethernet

Εκτός από την πιο πάνω βιβλιοθήκη, υπάρχουν και άλλες προϋποθέσεις, για την εγκατάσταση του Honeyd. Η εντολή για την εγκατάστασή τους φαίνεται παρακάτω:

```
root@raspberripi:/# sudo apt-get install libevent-dev libdumbnet-dev
libpcap-dev libpcre3-dev libedit-dev bison flex libtool automake
```

Συγκεκριμένα,

- libevent: Ορίζεται ως βιβλιοθήκη ειδοποιήσεων συμβάντων
- libdumbnet: Βιβλιοθήκη δικτύωσης
- libpcap: Παρέχει μια μεταφερόμενη δομή για δίκτυο χαμηλού επιπέδου
- παρακολούθησης. Το Libpcap μπορεί να προσφέρει στατιστικά στοιχεία δικτύου, παρακολούθηση και ανίχνευση σφαλμάτων
- libpcre3-dev: Αυτή είναι μια βιβλιοθήκη λειτουργιών που φέρει συνήθεις εκφράσεις των οποίων η γραμματική και η σημασιολογία στην οποία η γλώσσα Perl 5 είναι όμοια όσο είναι δυνατόν.
- libedit-dev: Βιβλιοθήκη BSD για την επεξεργασία γραμμών και ιστορίας

Το επόμενο βήμα της εγκατάστασης είναι η εγκατάσταση του honeyd:

```
# wget https://github.com/DataSoft/Honeyd/archive/master.zip
# sudo apt-get install unzip
```

```
# unzip master.zip
```

Και η παραμετροποίησή του:

```
# sudo ./autogen.sh
# sudo ./configure
```

Εάν η σύνταξη σταματήσει λόγω σφαλμάτων που σχετίζονται με την Python, πιθανότατα χρειάζεται να εκτελεστεί με τον παρακάτω τρόπο:

```
#./configure --without-python
```

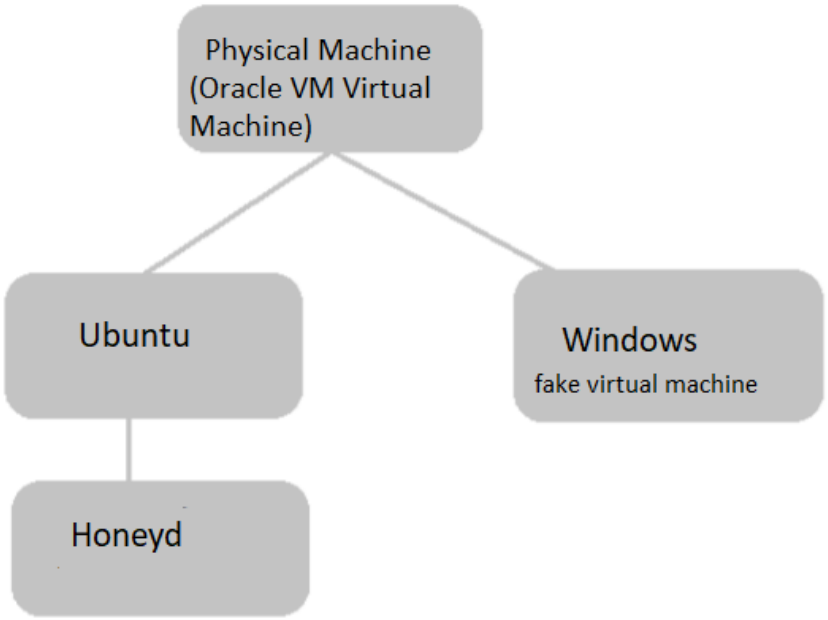
Και τελικά η εγκατάστασή του:

```
# sudo make
# sudo make install
```

4.1.1 Ρύθμιση της ψεύτικης εικονικής μηχανής Windows χρησιμοποιώντας το HoneyD

Για να την εγκατάσταση, τα απαραίτητα μηχανήματα είναι:

- ένα μηχάνημα Linux: Ubuntu, όπου σε αυτό θα τρέχει ο Honeyd, και
- ένα μηχάνημα Windows



Εικόνα 10. Τα μηχανήματα που χρησιμοποιηθήκαν για την εγκατάσταση του Honeyd.

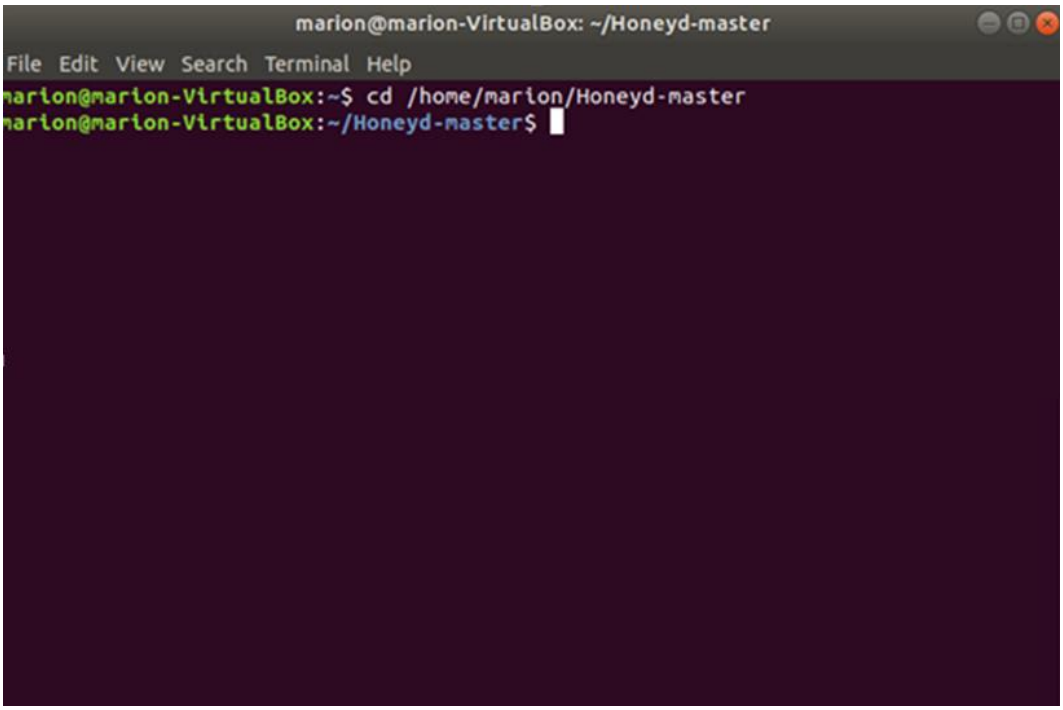
Μετά την επιτυχής εγκατάσταση του Honeyd στο Ubuntu μηχάνημα, το επόμενο βήμα είναι να δημιουργήσουμε το αρχείο διαμόρφωσης -configuration file. Το αρχείο αυτό είναι ουσιαστικά το κυρίως συστατικό ενός honeypot, αφού δείχνει στον honeyd ποιο λειτουργικό σύστημα μιμηθεί, ποιες πόρτες να ανοίξει και ποιες υπηρεσίες πρέπει να εκτελεστούν. Αυτό το αρχείο μπορεί να τροποποιηθεί για να μιμηθεί όλα τα είδη των εγκαταστάσεων, αλλά για την ώρα, θα επικεντρωθούμε στην εγκατάσταση ενός windows μηχανήματος.
Στην Εικόνα 11, φαίνεται το αρχείο διαμόρφωσης που χρησιμοποιήθηκε για την εγκατάσταση.



Εικόνα 11. Αρχείο διαμόρφωσης ψεύτικης προσωπικότητας – honeyd configuration file.

Η "create default" είναι μια δήλωση η οποία ενημερώνει το honeyd να εγκαταλείψει την κυκλοφορία εκτός αν και μέχρι να οριστεί στο αρχείο διαμόρφωσης. Αυτές οι γραμμές χρειάζονται όταν κάποιος αφήσει το honeypot να αποκτήσει μια διεύθυνση IP μέσω DHCP. Κάθε φορά που που έχουμε εμφάνιση "create" μέσα στο αρχείο διαμόρφωσης, σημαίνει δημιουργία ενός προτύπου για ένα honeypot, έτσι ώστε να μπορεί κανείς να δημιουργήσει τόσα honeypots όσο είναι απαραίτητο μέσα στο αρχείο honeyd.conf.
Για αυτό το πρότυπο Windows καθορίζουμε κάποια σημαντικά πράγματα. Πρώτα χρειάζεται να οριστεί η προσωπικότητα (personality). Η προσωπικότητα σημαίνει ότι όταν μία άλλη συσκευή στο δίκτυο συνδεθεί στο συγκεκριμένο honeypot, θα εμφανιστεί σαν να είναι ένα μηχάνημα Windows XP Professional SP1. Το πρότυπο windows περιέχει τρείς θύρες (135,25,80) που διατηρούνται ανοιχτές. Στο τέλος, η δήλωση "dhcp" δίνει εντολή στο πρότυπο windows να αποκτήσει μια διεύθυνση IP από το dhcp.
Αφού το αρχείο ρυθμίσεων honeyd.config είναι σωστά ρυθμισμένο, το επόμενο βήμα είναι να ξεκινήσει το honeyd.

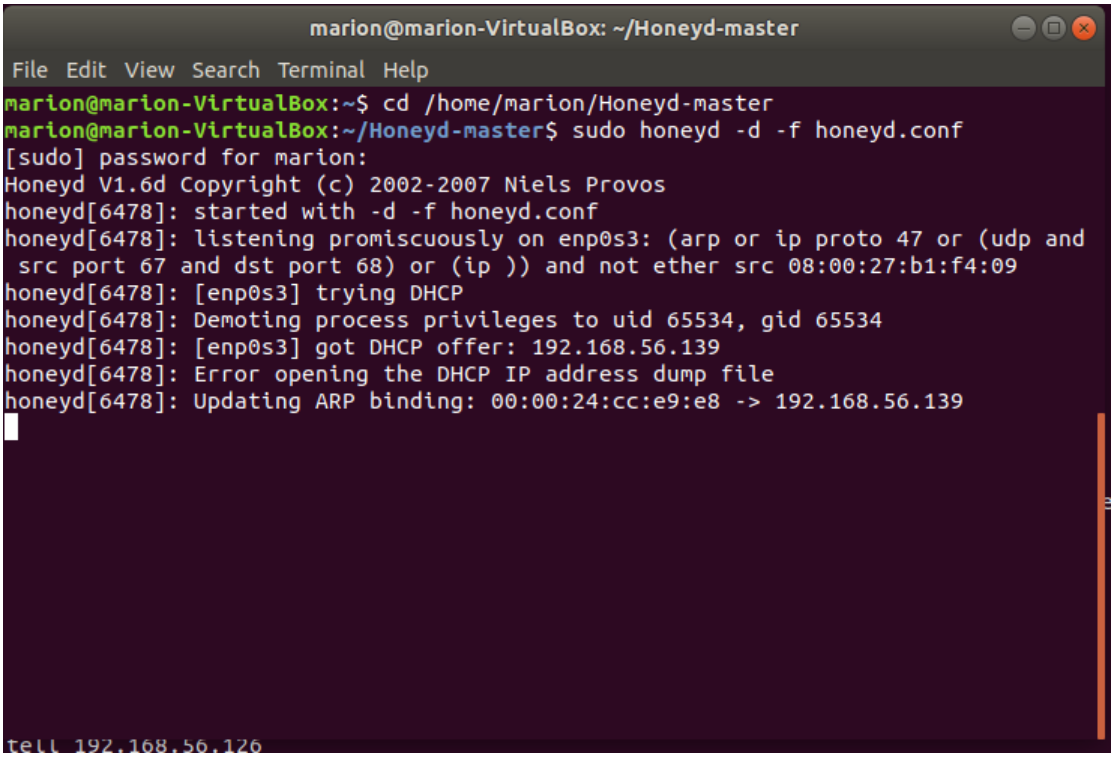
Αρχικά, μεταφερόμαστε στο μονοπάτι -directory, όπου βρίσκεται το αρχείο honeyd-master:



Εικόνα 12. Είσοδος στο φάκελο honeyd.

Και ύστερα τρέχουμε τον honeyd με την παρακάτω εντολή, όπως φαίνεται στην Εικόνα 13.

```
# sudo honeyd -d -f honeyd.conf
```



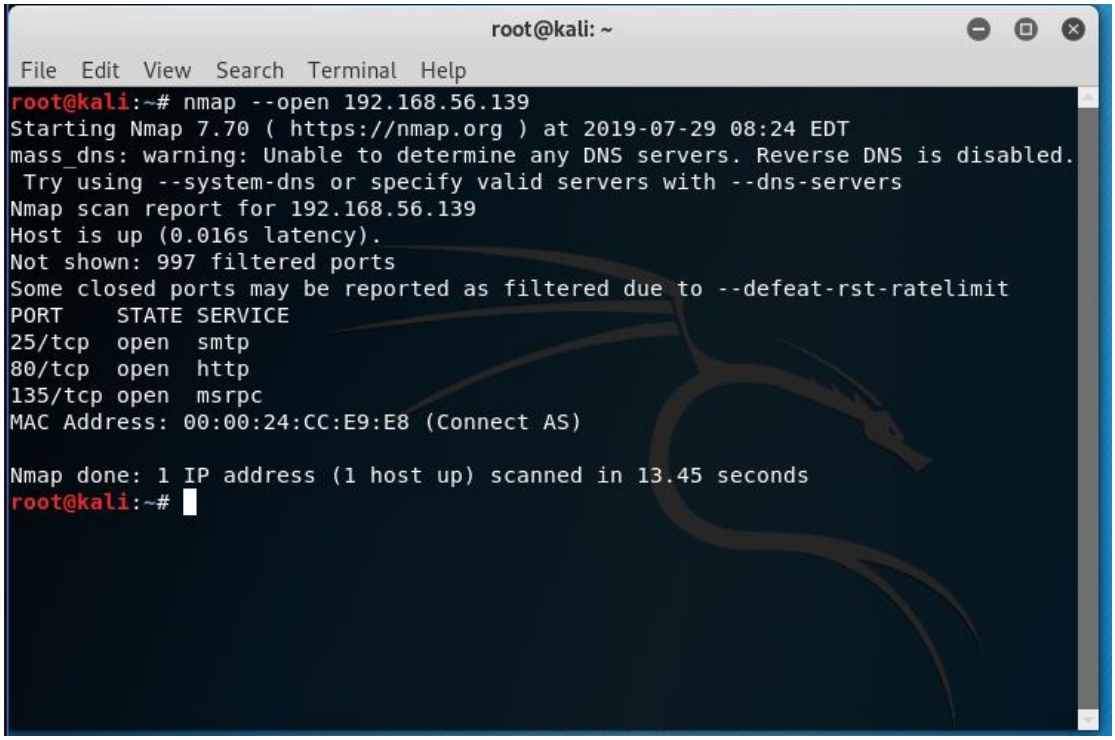
Εικόνα 13. Εκτέλεση του Honeyd.

Στην παραπάνω εντολή το -d χρησιμοποιείται έτσι ώστε να μην τρέχει στο background (ή δεν πρέπει να εκτελείται ως διαδικασία δαίμονα στην ορολογία του Linux). Αυτό θα επιτρέψει πιο λεπτομερή απόδοση, ώστε να μπορεί να αντιμετωπίσει τα προβλήματα

όταν χρειάζεται. Το τρέξιμο σε αυτή τη λειτουργία θα δείξει επίσης την IP που δόθηκε στο honeypot μέσω DCHP (εδώ 192.168.56.139).

4.1.2 Σάρωση της λειτουργικότητας του HoneyD από εκτέλεση εκτύπωσης Nmap Foot

Στο σημείο αυτό, ο honeyd έχει αναπτυχθεί και δοκιμαστεί με επιτυχία. Θα εκτελέσουμε παρακάτω μία σάρωση μέσω του NMAP εργαλείου που προαναφέρθηκε. Μέσω της σάρωσης αυτής θα δούμε τη διαθεσιμότητα του εικονικού συστήματος Windows που δημιουργείται από τον honeyd. Την σάρωση αυτή, την εκτελούμε από το μηχάνημα Kali Linux που βρίσκεται στο ίδιο δίκτυο.



Εικόνα 14. Σάρωσης διεύθυνσης 192.168.56.139 που δόθηκε στον honeypot διακομιστή.

Στην Εικόνα 14, φαίνονται οι ανοιχτές πόρτες(ports) και οι υπηρεσίες(services) του ψεύτικου εικονικού μηχανήματος Windows με διεύθυνση IP 192.168.56.139 που διαμορφώσαμε χρησιμοποιώντας τον honeyd.

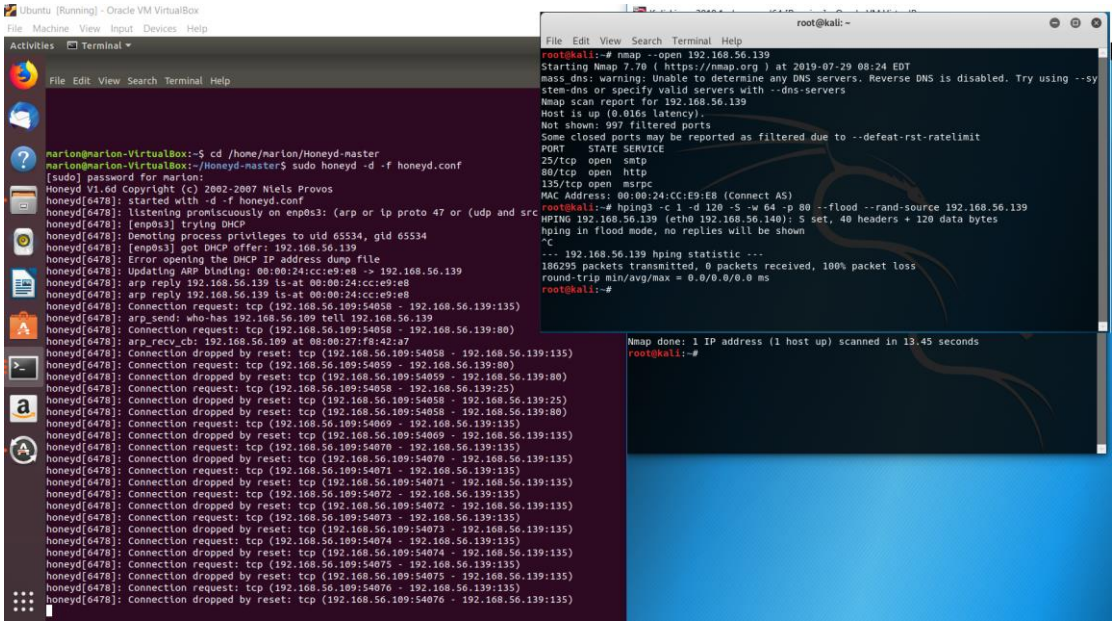
4.1.3 Δοκιμάζοντας τη λειτουργικότητα του HoneyD μέσω SYN Attack

Όπως αναφέραμε στην 3.3.1 Syn attack, η επίθεση SYN είναι η αποστολή πολλών πακέτων SYN στο σύστημα στόχος, εδώ στον honeypot, με σκοπό την υπερχείλιση των ουρών σύνδεσης και εντέλει την άρνηση του συστήματος. Στην Εικόνα 15, φαίνεται η επίθεση από το μηχάνημα Kali, χρησιμοποιώντας την παρακάτω εντολή:

```
# hping3 -c 15000 -d 120 -S -w 64 -p 80 --flood --rand-source 192.168.56.139
```

Η εντολή αυτή σημαίνει ότι στέλνουμε 15000 πακέτα (-c 15000) με μέγεθος 120 bytes (-d 120) το καθένα. Καθορίζουμε με το -S ότι το SYN flag πρέπει να είναι ενεργό με ένα παράθυρο TCP μεγέθους 64 (-w 64). Κατευθύνουμε την επίθεση στον διακομιστή δικτύου HTTP web server

του θύματος, ορίζοντας την πόρτα 80 (-p 80) και χρησιμοποιούμε την –flood για να στείλουμε πακέτα όσο το δυνατόν γρηγορότερα. Η –rand-source δημιουργεί εικονικές IP για να μην μπορεί να εντοπιστεί η πραγματική IP του επιτιθέμενου, αλλά και για να μην μπορεί το θύμα να στείλει πίσω SYN-ACK πακέτα.

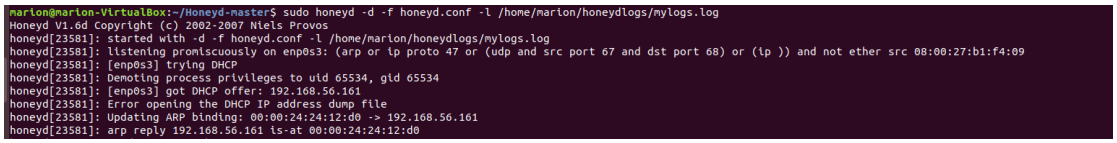


Εικόνα 15. Επίθεση SYN από Kali Linux στον honeypot με διεύθυνση 192.168.56.139.

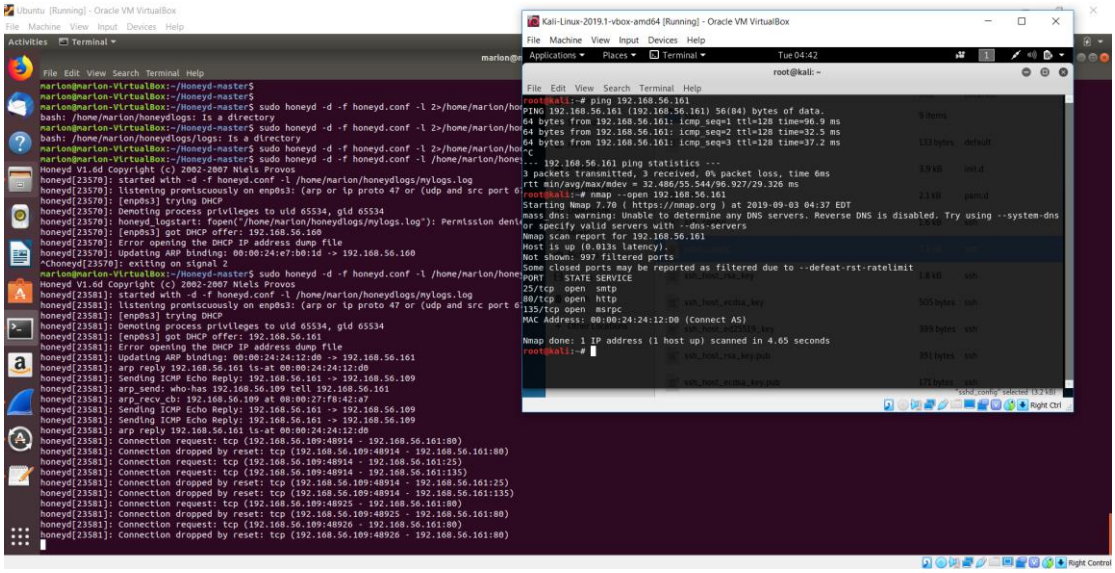
4.1.4 Αρχεία καταγραφής

Όπως είπαμε και στην ενότητα **Αρχεία Καταγραφής**, ο Honeyd υποστηρίζει πολλούς τρόπους καταγραφής της κίνησης του δικτύου.

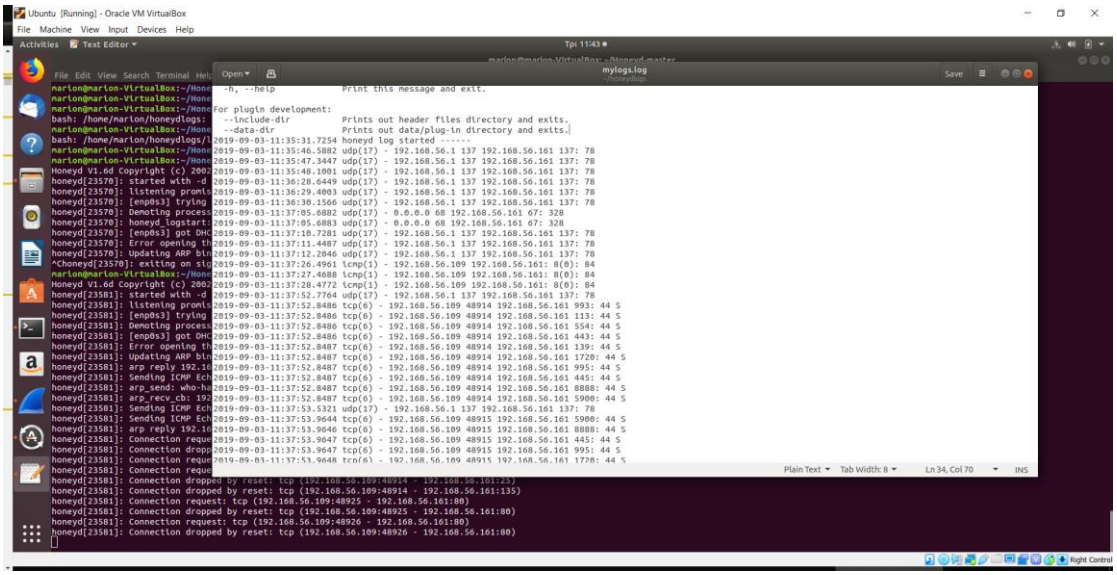
Μπορεί να δημιουργήσει αρχεία καταγραφής σύνδεσης, τις αναφορές που επιχειρήθηκαν και ολοκληρωμένες συνδέσεις για κάθε πρωτόκολλο. Οι εφαρμογές υπηρεσιών είναι σε θέση να αναφέρουν τις πληροφορίες που θα καταγραφούν στο Honeyd χρησιμοποιώντας το stderr. Η δομή του Honeyd χρησιμοποιεί το syslog για να συσσωρεύσει τις πληροφορίες στο μηχάνημα. Το -l είναι μια σημαία που χρησιμοποιείται για την αποθήκευση του των πακέτων και των συνδέσεων στο αρχείο καταγραφής, όπως φαίνεται στην Εικόνα 16.



Εικόνα 16. Καταγραφή των πακέτων και των συνδέσεων στο αρχείο mylogs.log.



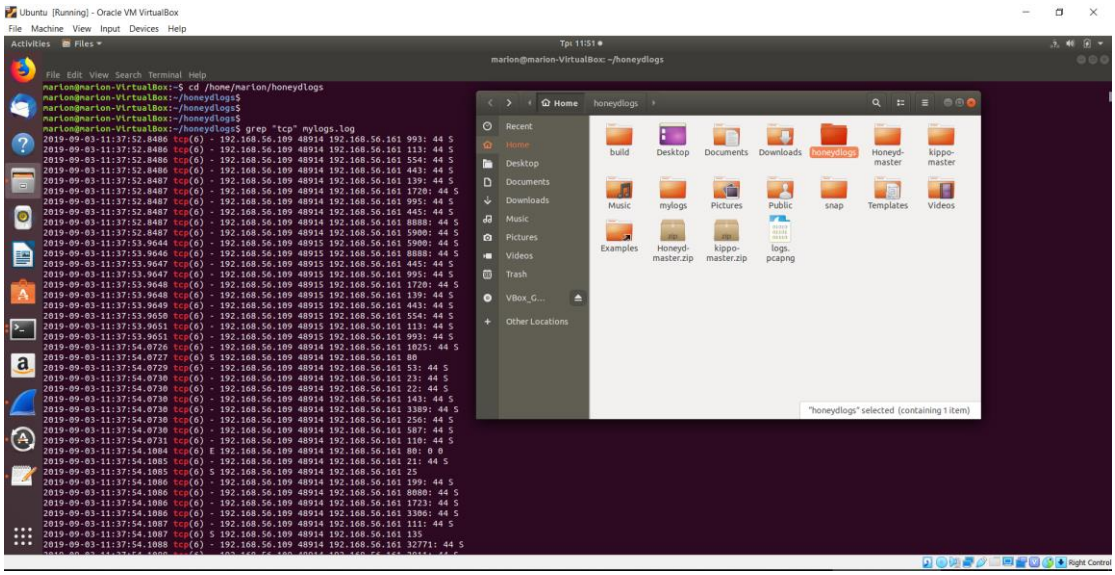
Εικόνα 17. Αποστολή πακέτων από μηχάνημα Linux 192.168.56.109.



Εικόνα 18. Απεικόνιση του αρχείου καταγραφής του honeyd. Αναγράφονται οι κινήσεις από το μηχάνημα Kali με IP 192.168.56.109.

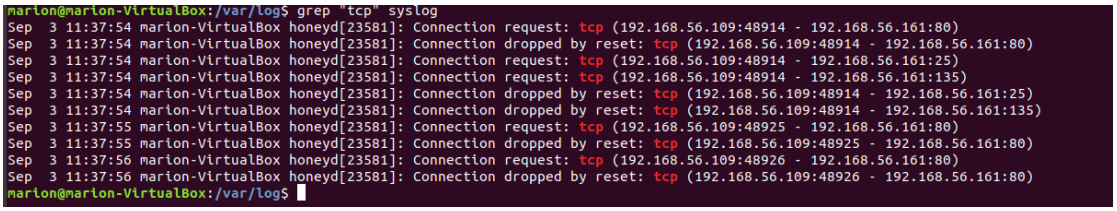
Ακόμα, μπορούμε να ψάξουμε στα αρχεία καταγραφής μέσω τερματικού, χρησιμοποιώντας την εντολή grep, όπως φαίνεται στις

```
#cd /home/marion/honeydlogs
#grep “tcp” mylogs.log
```



Εικόνα 19. Αναζήτηση στο αρχείο mylogs για ότι περιέχει tcp.

```
#cd /var/log
#grep “tcp” syslog
```



Εικόνα 20. Αναζήτηση στο αρχείο syslog για ότι περιέχει tcp.

```
#cd /home/marion/honeydlogs
#grep “192.168.56.161” mylogs.log
```

```
marion@marion-VirtualBox:~/honeydlogs$ grep "SYN" mylogs.log
marion@marion-VirtualBox:~/honeydlogs$ grep "UDP" mylogs.log
marion@marion-VirtualBox:~/honeydlogs$ grep "upd" mylogs.log
marion@marion-VirtualBox:~/honeydlogs$ grep "192.168.56.161" mylogs.log
2019-09-03-11:35:46.5882 udp(17) - 192.168.56.1 137 192.168.56.161 137: 78
2019-09-03-11:35:47.3447 udp(17) - 192.168.56.1 137 192.168.56.161 137: 78
2019-09-03-11:35:48.1001 udp(17) - 192.168.56.1 137 192.168.56.161 137: 78
2019-09-03-11:36:28.6449 udp(17) - 192.168.56.1 137 192.168.56.161 137: 78
2019-09-03-11:36:29.4003 udp(17) - 192.168.56.1 137 192.168.56.161 137: 78
2019-09-03-11:36:30.1566 udp(17) - 192.168.56.1 137 192.168.56.161 137: 78
2019-09-03-11:37:05.6882 udp(17) - 0.0.0.0 68 192.168.56.161 67: 328
2019-09-03-11:37:05.6883 udp(17) - 0.0.0.0 68 192.168.56.161 67: 328
2019-09-03-11:37:10.7281 udp(17) - 192.168.56.1 137 192.168.56.161 137: 78
2019-09-03-11:37:11.4487 udp(17) - 192.168.56.1 137 192.168.56.161 137: 78
2019-09-03-11:37:12.2046 udp(17) - 192.168.56.1 137 192.168.56.161 137: 78
2019-09-03-11:37:26.4961 icmp(1) - 192.168.56.109 192.168.56.161: 8(0): 84
2019-09-03-11:37:27.4688 icmp(1) - 192.168.56.109 192.168.56.161: 8(0): 84
2019-09-03-11:37:28.4772 icmp(1) - 192.168.56.109 192.168.56.161: 8(0): 84
2019-09-03-11:37:52.7764 udp(17) - 192.168.56.1 137 192.168.56.161 137: 78
2019-09-03-11:37:52.8486 tcp(6) - 192.168.56.109 48914 192.168.56.161 993: 44 S
2019-09-03-11:37:52.8486 tcp(6) - 192.168.56.109 48914 192.168.56.161 113: 44 S
2019-09-03-11:37:52.8486 tcp(6) - 192.168.56.109 48914 192.168.56.161 554: 44 S
2019-09-03-11:37:52.8486 tcp(6) - 192.168.56.109 48914 192.168.56.161 443: 44 S
2019-09-03-11:37:52.8487 tcp(6) - 192.168.56.109 48914 192.168.56.161 139: 44 S
2019-09-03-11:37:52.8487 tcp(6) - 192.168.56.109 48914 192.168.56.161 1720: 44 S
2019-09-03-11:37:52.8487 tcp(6) - 192.168.56.109 48914 192.168.56.161 995: 44 S
2019-09-03-11:37:52.8487 tcp(6) - 192.168.56.109 48914 192.168.56.161 445: 44 S
2019-09-03-11:37:52.8487 tcp(6) - 192.168.56.109 48914 192.168.56.161 8888: 44 S
2019-09-03-11:37:52.8487 tcp(6) - 192.168.56.109 48914 192.168.56.161 5900: 44 S
2019-09-03-11:37:53.5321 udp(17) - 192.168.56.1 137 192.168.56.161 137: 78
2019-09-03-11:37:53.9644 tcp(6) - 192.168.56.109 48915 192.168.56.161 5900: 44 S
2019-09-03-11:37:53.9646 tcp(6) - 192.168.56.109 48915 192.168.56.161 8888: 44 S
2019-09-03-11:37:53.9647 tcp(6) - 192.168.56.109 48915 192.168.56.161 445: 44 S
2019-09-03-11:37:53.9647 tcp(6) - 192.168.56.109 48915 192.168.56.161 995: 44 S
2019-09-03-11:37:53.9648 tcp(6) - 192.168.56.109 48915 192.168.56.161 1720: 44 S
2019-09-03-11:37:53.9648 tcp(6) - 192.168.56.109 48915 192.168.56.161 139: 44 S
2019-09-03-11:37:53.9649 tcp(6) - 192.168.56.109 48915 192.168.56.161 443: 44 S
```

Εικόνα 21. Αναζήτηση στο αρχείο mylogs για ότι περιέχει την IP 192.168.56.161.

```
#cd /var/log
#grep “192.168.56.161” “udp” syslog
```

```
marion@marion-VirtualBox:/var/log$ grep "192.168.56.161" "udp" syslog
grep: udp: No such file or directory
syslog:Sep  3 11:35:31 marion-VirtualBox honeyd[23581]: [enp0s3] got DHCP offer: 192.168.56.161
syslog:Sep  3 11:35:31 marion-VirtualBox honeyd[23581]: Updating ARP binding: 00:00:24:24:12:d0 -> 192.168.56.161
syslog:Sep  3 11:37:26 marion-VirtualBox honeyd[23581]: arp reply 192.168.56.161 is-at 00:00:24:24:12:d0
syslog:Sep  3 11:37:26 marion-VirtualBox honeyd[23581]: Sending ICMP Echo Reply: 192.168.56.161 -> 192.168.56.109
syslog:Sep  3 11:37:26 marion-VirtualBox honeyd[23581]: arp_send: who-has 192.168.56.109 tell 192.168.56.161
syslog:Sep  3 11:37:27 marion-VirtualBox honeyd[23581]: Sending ICMP Echo Reply: 192.168.56.161 -> 192.168.56.109
syslog:Sep  3 11:37:28 marion-VirtualBox honeyd[23581]: Sending ICMP Echo Reply: 192.168.56.161 -> 192.168.56.109
syslog:Sep  3 11:37:52 marion-VirtualBox honeyd[23581]: arp reply 192.168.56.161 is-at 00:00:24:24:12:d0
syslog:Sep  3 11:37:54 marion-VirtualBox honeyd[23581]: Connection request: tcp (192.168.56.109:48914 - 192.168.56.161:80)
syslog:Sep  3 11:37:54 marion-VirtualBox honeyd[23581]: Connection dropped by reset: tcp (192.168.56.109:48914 - 192.168.56.161:80)
syslog:Sep  3 11:37:54 marion-VirtualBox honeyd[23581]: Connection request: tcp (192.168.56.109:48914 - 192.168.56.161:25)
syslog:Sep  3 11:37:54 marion-VirtualBox honeyd[23581]: Connection request: tcp (192.168.56.109:48914 - 192.168.56.161:135)
syslog:Sep  3 11:37:54 marion-VirtualBox honeyd[23581]: Connection dropped by reset: tcp (192.168.56.109:48914 - 192.168.56.161:25)
syslog:Sep  3 11:37:54 marion-VirtualBox honeyd[23581]: Connection dropped by reset: tcp (192.168.56.109:48914 - 192.168.56.161:135)
syslog:Sep  3 11:37:55 marion-VirtualBox honeyd[23581]: Connection request: tcp (192.168.56.109:48925 - 192.168.56.161:80)
syslog:Sep  3 11:37:55 marion-VirtualBox honeyd[23581]: Connection dropped by reset: tcp (192.168.56.109:48925 - 192.168.56.161:80)
syslog:Sep  3 11:37:56 marion-VirtualBox honeyd[23581]: Connection request: tcp (192.168.56.109:48926 - 192.168.56.161:80)
syslog:Sep  3 11:37:56 marion-VirtualBox honeyd[23581]: Connection dropped by reset: tcp (192.168.56.109:48926 - 192.168.56.161:80)
syslog:Sep  3 11:45:31 marion-VirtualBox honeyd[23581]: arp reply 192.168.56.161 is-at 00:00:24:24:12:d0
syslog:Sep  3 11:45:44 marion-VirtualBox honeyd[23581]: arp reply 192.168.56.161 is-at 00:00:24:24:12:d0
syslog:Sep  3 12:05:31 marion-VirtualBox honeyd[23581]: arp reply 192.168.56.161 is-at 00:00:24:24:12:d0
marion@marion-VirtualBox:/var/log$
```

Εικόνα 22. Αναζήτηση στο αρχείο syslog για ότι περιέχει την IP 192.168.56.161 και udp.

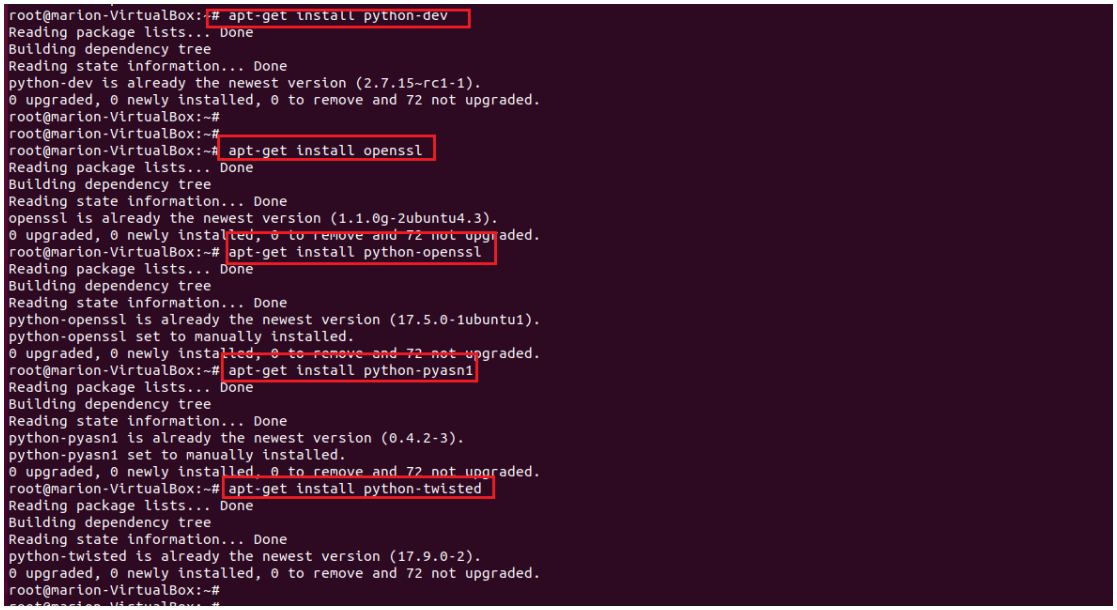
4.2 Εγκατάσταση του Kippo, διαμόρφωση, επίθεση

Για την εγκατάσταση του kippo, θα χρειαστεί να εγκατασταθούν τα παρακάτω λογισμικά πακέτα:

- python-dev: στατική βιβλιοθήκη Python.
- openssl: βιβλιοθήκη με σκοπό την κρυπτογράφηση που παράγει RSA κλειδιά και παρέχει την υλοποίηση των ανοιχτών SSL και TLS.

- python-openssl: Python για την ανοιχτή SSLβιβλιοθήκη.
- python-pyasn1: Υλοποίηση ASN.1 τύπων και προγραμματισμένο σε Python.
- python-twisted: Ειδική δομή για Python γλώσσα.

```
#sudo apt-get install python-dev openssl pythonopenssl python-pyasn1 python-twisted
```

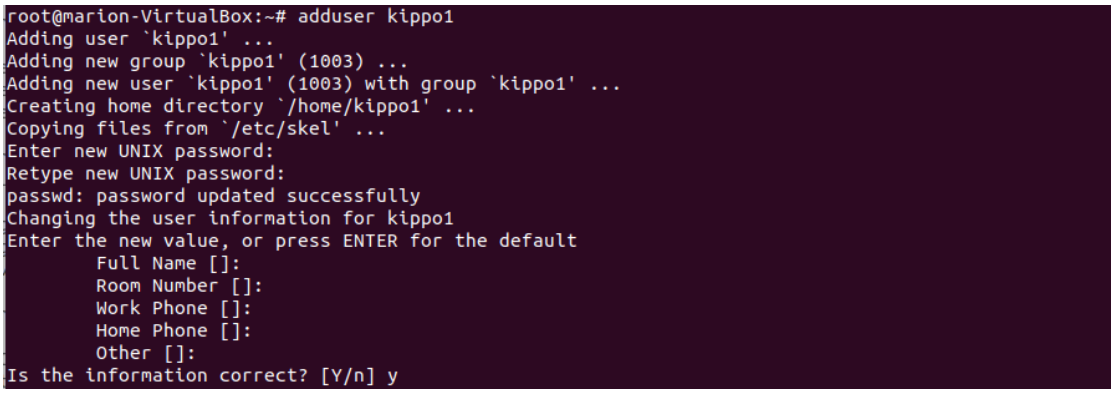


Εικόνα 23. Εγκατάσταση απαραίτητου λογισμικού για τον Kippo honeypot.

Ύστερα το git και το authbind:

```
#apt-get install git
#apt-get install authbind
```

Πρέπει να σημειωθεί ότι ο kippo δεν μπορεί να τρέξει με root user. Για αυτό θα πρέπει να δημιουργηθεί νέος user. Ονομάζουμε τον νέο χρήστη kippo1 μς κωδικό:123456 και συμπληρώνουμε τις απαιτούμενες πληροφορίες όπως φαίνεται παρακάτω. Ύστερα δίνουμε δικαιώματα sudo στον χρήστη kippo1 μέσω της εντολής
#sudo visudo και μετά μεταφερόμαστε στον χρήστη kippo1.



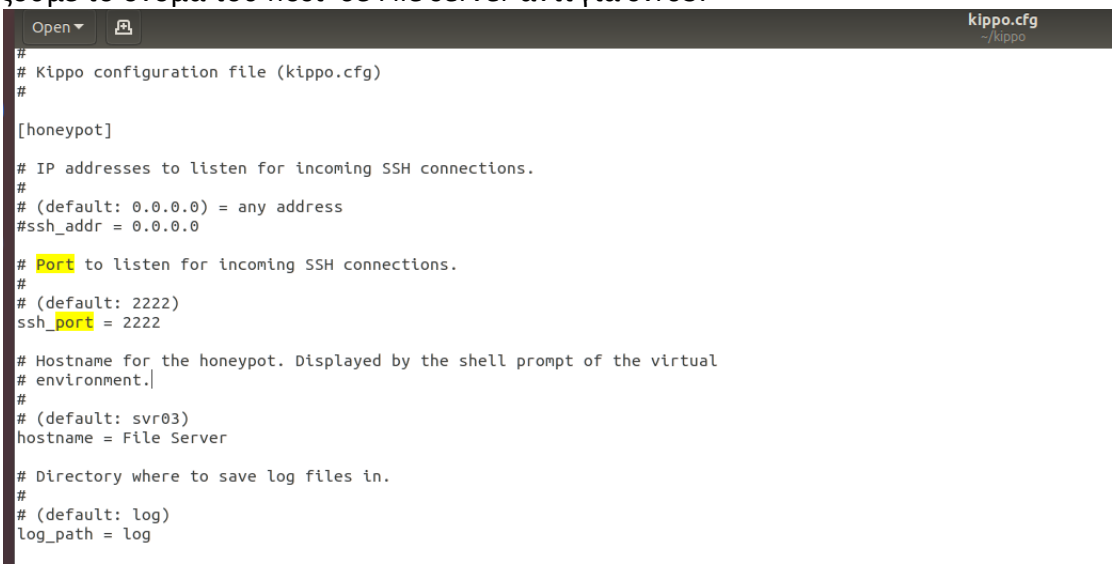
Εικόνα 24. Δημιουργία non-root χρήστη για την εκτέλεση του Kippo.

Κατεβάζουμε το αρχείο Kippo από το git, όπως φαίνεται στην Εικόνα 25.

```
kippo1@marion-VirtualBox:~$ sudo git clone https://github.com/desaster/kippo.git
[sudo] password for kippo1:
Cloning into 'kippo'...
remote: Enumerating objects: 1544, done.
remote: Total 1544 (delta 0), reused 0 (delta 0), pack-reused 1544
Receiving objects: 100% (1544/1544), 2.64 MiB | 2.64 MiB/s, done.
Resolving deltas: 100% (929/929), done.
kippo1@marion-VirtualBox:~$
kippo1@marion-VirtualBox:~$
```

Εικόνα 25.

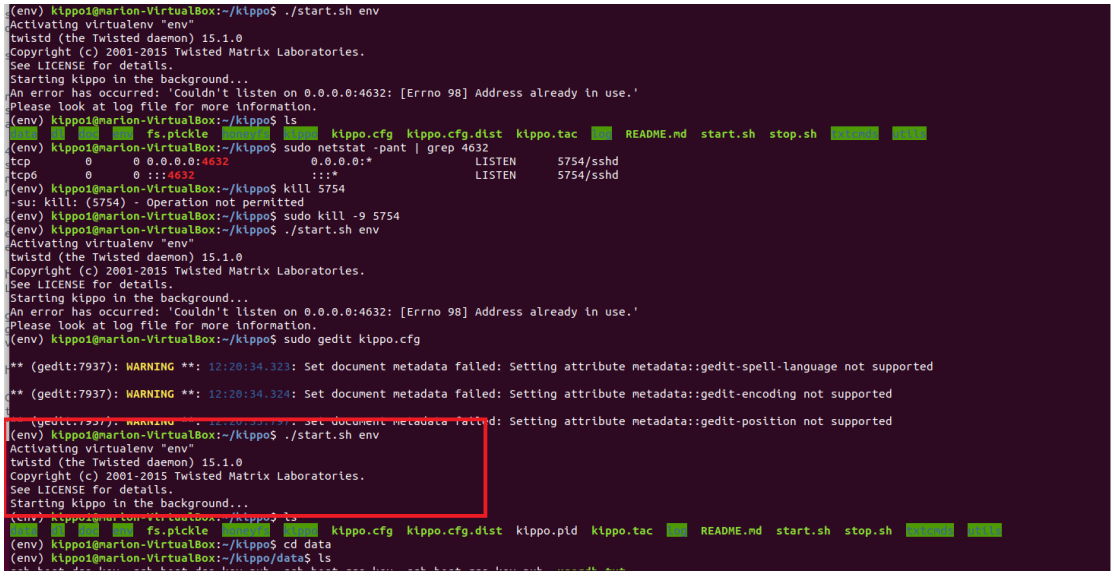
Ύστερα, αντιγράφουμε το αρχείο kippo.cfg.dist και να το ονομάζουμε kippo.cfg. Μπορούμε να αλλάξουμε το όνομα του host σε File server αντί για svr03.



Εικόνα 26. Αρχείο διαμόρφωσης kippo.cfg.

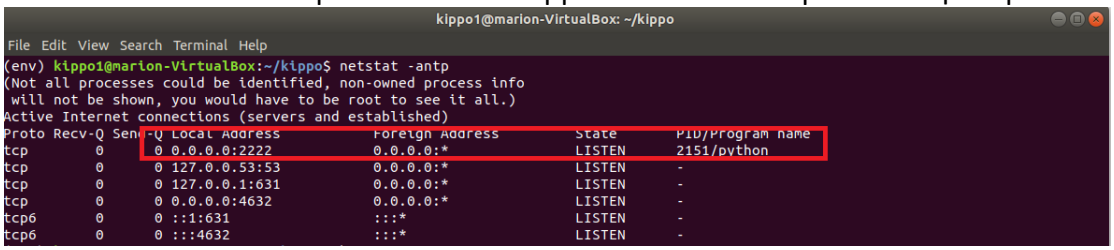
Ο Kippo χρειάζεται μερικά πακέτα python για να τρέξει. Ο καλύτερος τρόπος για την εγκατάστασή τους είναι η χρήση του pip. Για να αποφύγουμε να έχουμε προβλήματα συμβατότητας με παλαιότερες βιβλιοθήκες, θα πρέπει να εκτελεστούν οι παρακάτω εντολές.

```
# sudo apt-get install build-essential python-dev libmysqlclient-dev python-
virtualenv python-pip
# virtualenv env
#. ./env/bin/activate
(env)# pip install twisted==15.1.0
(env)# pip install pyasn1
(env)# pip install pycrypto
(env)# pip install MySQL-python
(env)# ./start.sh env
```



Εικόνα 27. Εκτέλεση του Kippo.

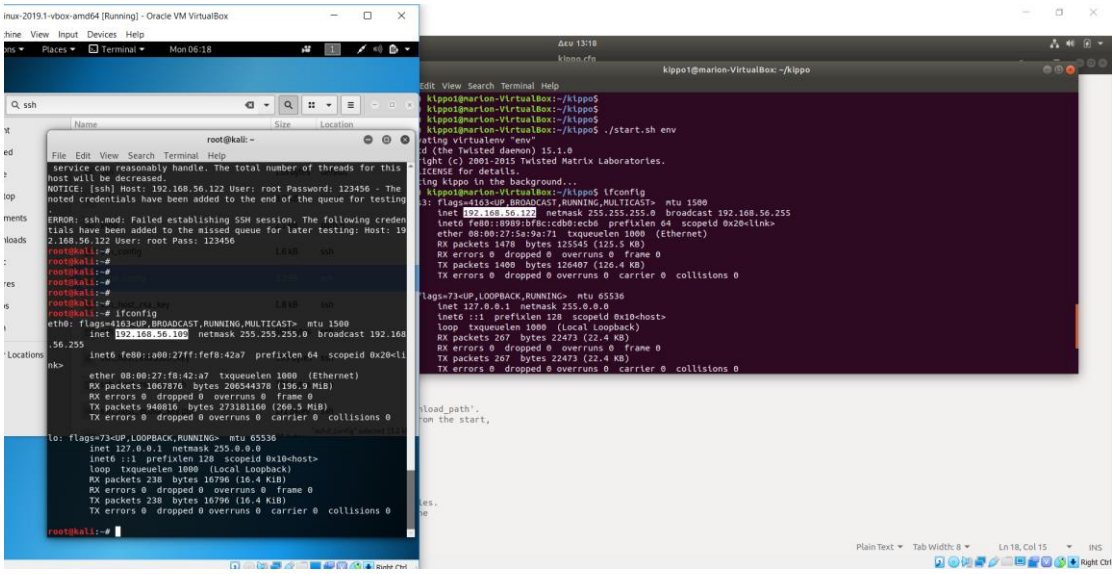
Αφού τρέξει ο Kippo, μπορούμε να ελέγξουμε την κατάσταση χρησιμοποιώντας την εντολή netstat -antp που φαίνεται στην Εικόνα 28, όπου φαίνεται ξεκάθαρα ότι το αρχείο Python το οποίο ουσιαστικά αντιπροσωπεύει τον Kippo server ανταποκρίνεται στην πόρτα 2222.



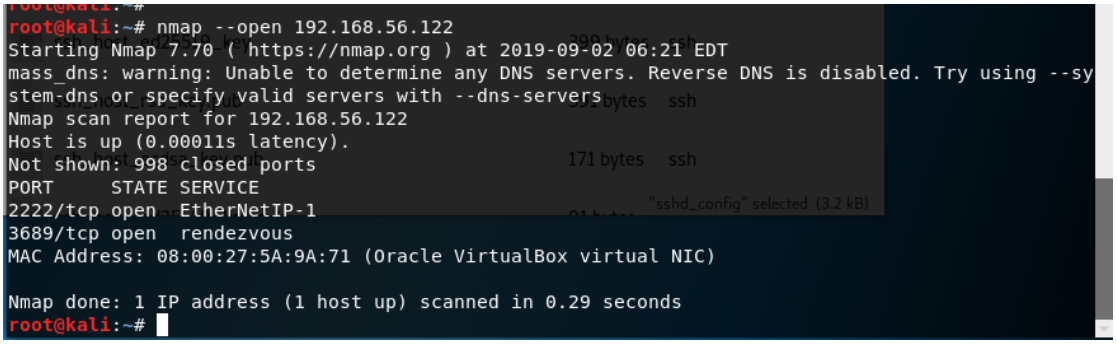
Εικόνα 28. Εντολή netstat για την απεικόνιση των ενεργών συνδέσεων και υπηρεσιών.

4.2.1 Δοκιμάζοντας τη λειτουργικότητα του Kippo μέσω Medusa Attack

Ο kippo server τρέχει στην IP 192.168.56.122. Η επίθεση θα γίνει από Kali Linux με IP 192.168.56.109, όπως φαίνεται στην Εικόνα 29. Θα πρέπει πρώτα να σαρώσουμε την IP του θύματος (192.168.56.122) για να δούμε ποιες πόρτες είναι ανοιχτές. Όπως φαίνεται στην Εικόνα 30, η πόρτα 2222, όπου τρέχει ο Kippo, είναι ανοιχτή.



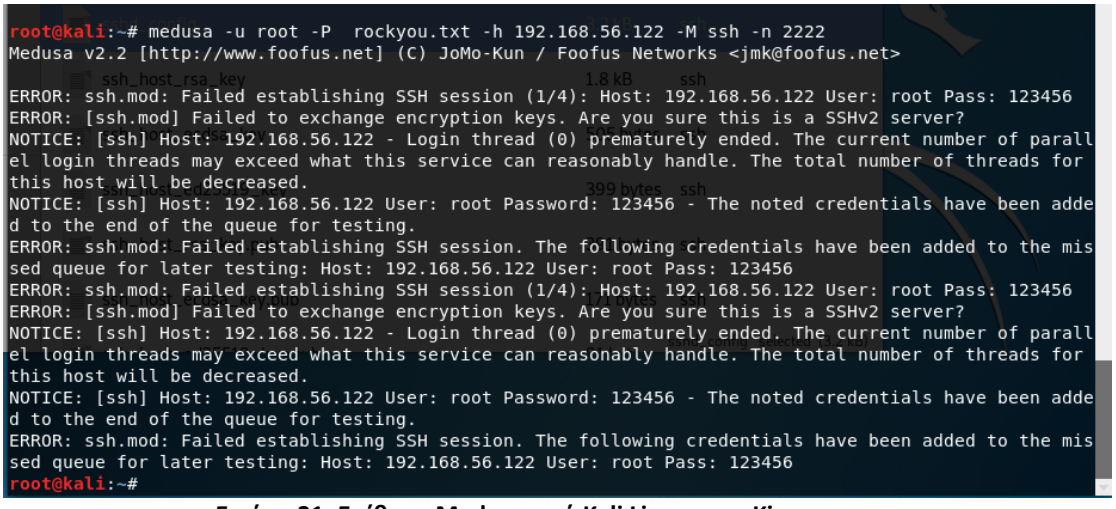
Εικόνα 29. IP διευθύνσεις Ubuntu και Kali Linux.



Εικόνα 30. Σάρωση διεύθυνσης θύματος 192.168.56.122.

Η επίθεση Μέδουσα θα πραγματοποιηθεί στην πόρτα 2222, εκτελώντας την παρακάτω εντολή, όπως φαίνεται στην Εικόνα 31.

```
# medusa -u root -P  rockyou.txt -h 192.168.56.122 -M ssh -n 2222
```



Εικόνα 31. Επίθεση Medusa από Kali Linux στον Kippo server.

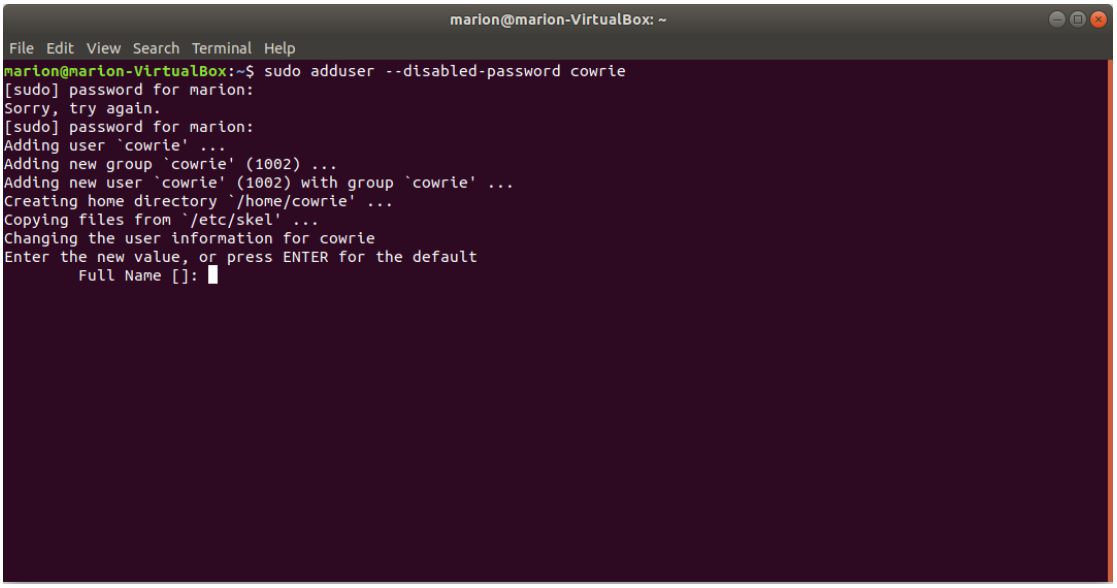
Ο Kippo εμπεριέχει ένα bug στην αποκρυπτογράφηση των κλειδιών με Diffie-Helman, με αποτέλεσμα να μην μπορεί να πραγματοποιηθεί η αποκρυπτογράφησή τους και να επιτευχθεί η επικοινωνία μεταξύ των μηχανημάτων. Παρόλα αυτά, αυτό θα μπορούσε να υποβληθεί για μελλοντική εργασία μαζί με την έρευνα άλλων αλγορίθμων αποκρυπτογράφησης.

4.3 Εγκατάσταση του Cowrie, διαμόρφωση, επίθεση

Αρχικά, πρέπει να εγκατασταθούν κάποιες εξαρτήσεις (dependencies), όπως υποστήριξη σε επίπεδο συστήματος για εικονικά περιβάλλοντα Python και άλλες εξαρτήσεις. Πραγματικά πακέτα Python εγκαθίστανται αργότερα.

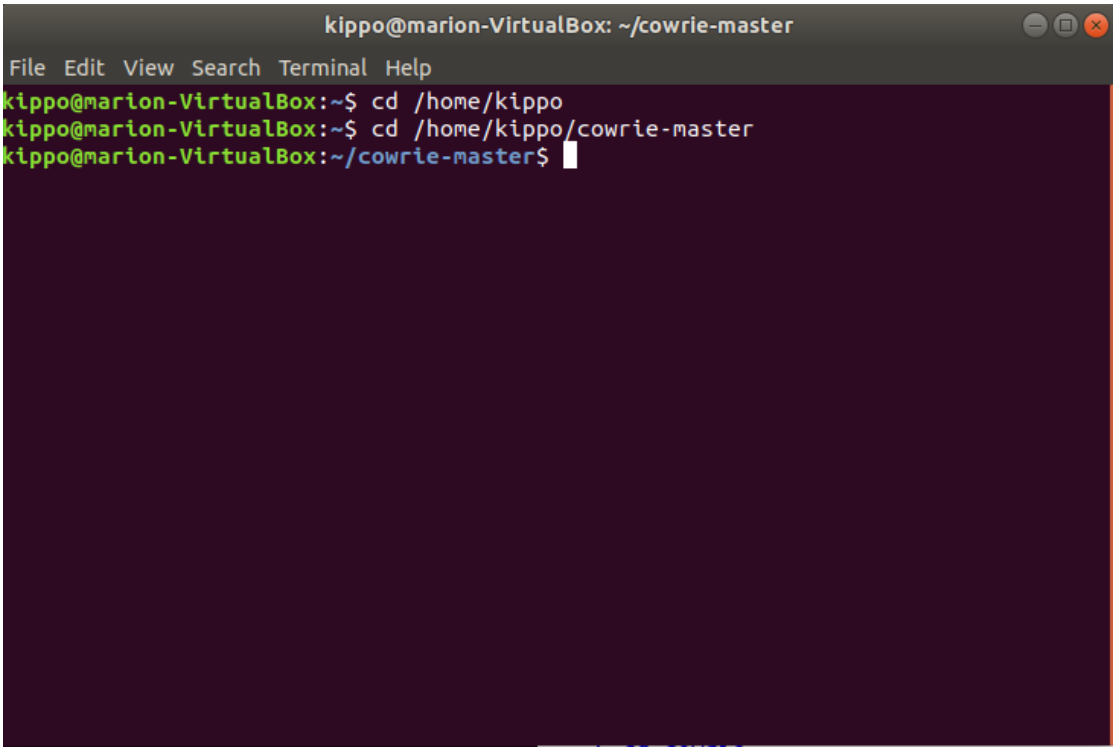
```
#sudo apt-get install git python-virtualenv libssl-dev libffi-dev build-essential
libpython-dev python2.7-minimal authbind
```

Στη συνέχεια, θα πρέπει να δημιουργηθεί ένας χρήστης, ο οποίος δεν είναι υπερ-χρήστης (non-root user). Τον χρήστη αυτόν θα τον ονομάσουμε kippo, Εικόνα 32.



Εικόνα 32. Δημιουργία non-root χρήστη.

Κατευθυνόμαστε ύστερα στον φάκελο cowrie-master αφού εισέλθουμε με τον νέο χρήστη στο τερματικό, όπως φαίνεται στην Εικόνα 33.



Εικόνα 33. Μετάβαση στον φάκελο cowrie-master.

Στη συνέχεια, χρειάζεται να δημιουργηθείο ένα Python2 περιβάλλον. Για αυτό, πληκτρολογούμε την παρακάτω εντολή.

```
# virtualenv --python=python2 cowrie-env
```

Ενεργοποιούμε το εικονικό περιβάλλον και εγκαθιστούμε τα παρακάτω πακέτα.

```
# source cowrie-env/bin/activate

(cowrie-env) # pip install --upgrade pip

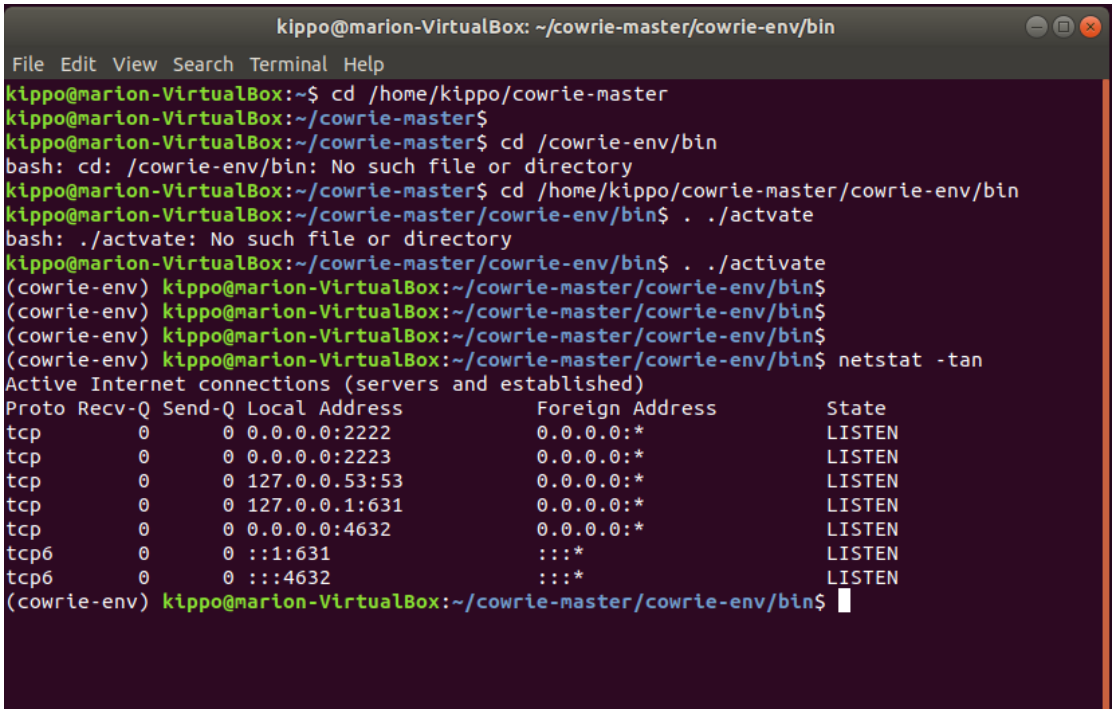
(cowrie-env) # pip install --upgrade -r requirements.txt
```

Στο επόμενο βήμα, εγκαθιστούμε το αρχείο διαμόρφωσης. Το αρχείο διαμόρφωσης είναι αποθηκευμένο στο cowrie.cfg.dist και cowrie.cfg. Και τα δύο αρχεία διαβάζονται κατά την εκκίνηση, όπου οι καταχωρήσεις από το cowrie.cfg έχουν προτεραιότητα. Το αρχείο .dist μπορεί να αντικατασταθεί από αναβαθμίσεις, το cowrie.cfg δε θα αγγιχτεί. Για να τρέξετε με μια τυπική διαμόρφωση, δεν χρειάζεται να αλλάξετε τίποτα. Για να ενεργοποιήσετε το telnet, για παράδειγμα, δημιουργήστε το cowrie.cfg και εισάγετε μόνο τα εξής:

```
[telnet]

enabled = true
```

Τέλος, ξεκινάμε τον cowrie, όπως φαίνεται στην Εικόνα 34.

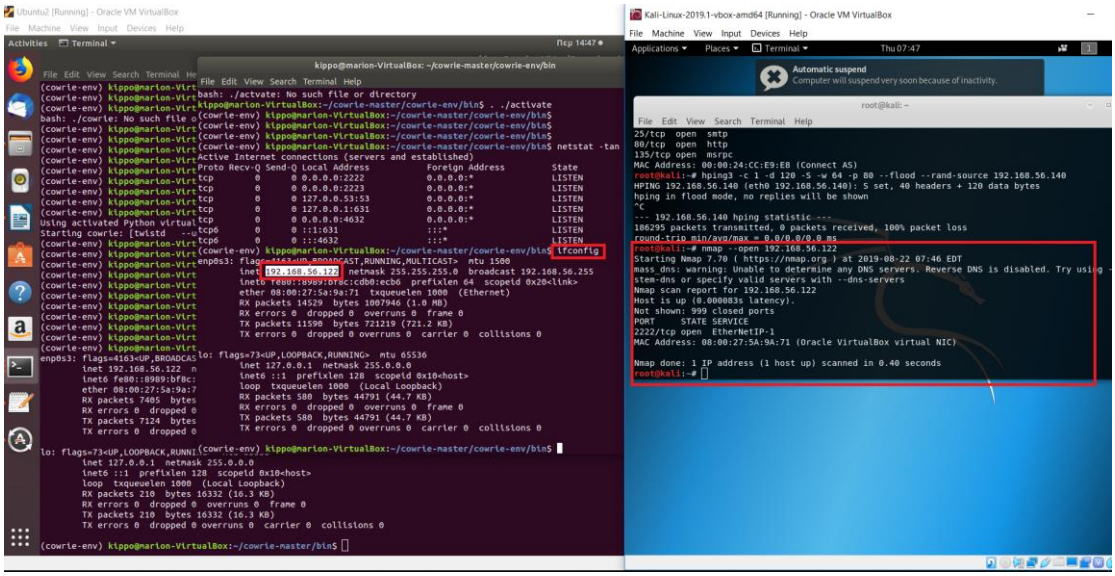


Εικόνα 34. Εκτέλεση του Cowrie.

Με την εντολή netstat, μπορούμε να δούμε λεπτομέρειες σχετικά με τις συνδέσεις δικτύου, γενικές στατιστικές και στατιστικές δικτύωσης για συγκεκριμένα πρωτόκολλα και πολλά άλλα τα οποία θα μπορούσαν να βοηθήσουν στην αντιμετώπιση συγκεκριμένων προβλημάτων δικτύωσης.

4.3.1 Σάρωση της λειτουργικότητας του Cowrie από εκτέλεση εκτύπωσης Nmap Foot

Στο σημείο αυτό, ο cowrie, έχει αναπτυχθεί και τρέχει επιτυχώς. Ας σαρώσουμε λοιπόν την διεύθυνση IP όπου τρέχει ο cowrie (192.168.56.122) για να δούμε ποιες πόρτες είναι ανοιχτές.



Εικόνα 35. Σάρωση της 192.168.56.122 από Kali Linux.

Όπως φαίνεται στην Εικόνα 35, η πόρτα 2222 είναι ανοιχτή. Συνεπώς, οι επιθέσεις που θα πραγματοποιηθούν θα πρέπει να έχουν ως στόχο την πόρτα 2222.

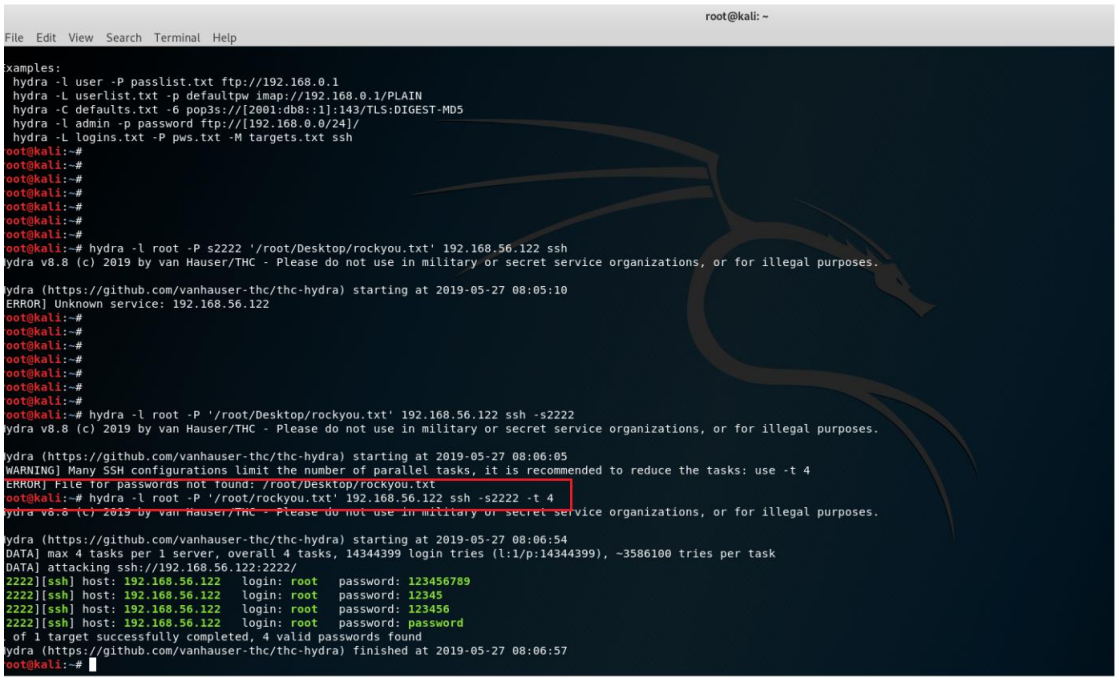
4.3.2 Δοκιμάζοντας τη λειτουργικότητα του Cowrie μέσω Brute Forcing Passwords με Hydra, Ncrack και Medusa

4.3.2.1. Brute Force επίθεση με Hydra

Όπως αναφέρθηκε και παραπάνω, η Ύδρα, είναι ένα εργαλείο που σπάει τον κωδικό πρόσβασης. Αυτό που χρειάζεται από τον επιτιθέμενο, είναι μία λίστα με πιθανούς κωδικούς. Στην περίπτωσή μας, αυτή η λίστα είναι το αρχείο rockyou.txt, το οποίο περιέχει μία μεγάλη λίστα από πιθανούς κωδικούς.

Στην Εικόνα 36, φαίνεται η επίθεση με Hydra χρησιμοποιώντας την λίστα κωδικών rockyou.txt (/root/rockyou.txt) με 4 threads στον συγκεκριμένο ssh server (192.168.56.122 και port:222 {-s2222}), προσπαθώντας να συνδεθούμε ως root χρήστης.

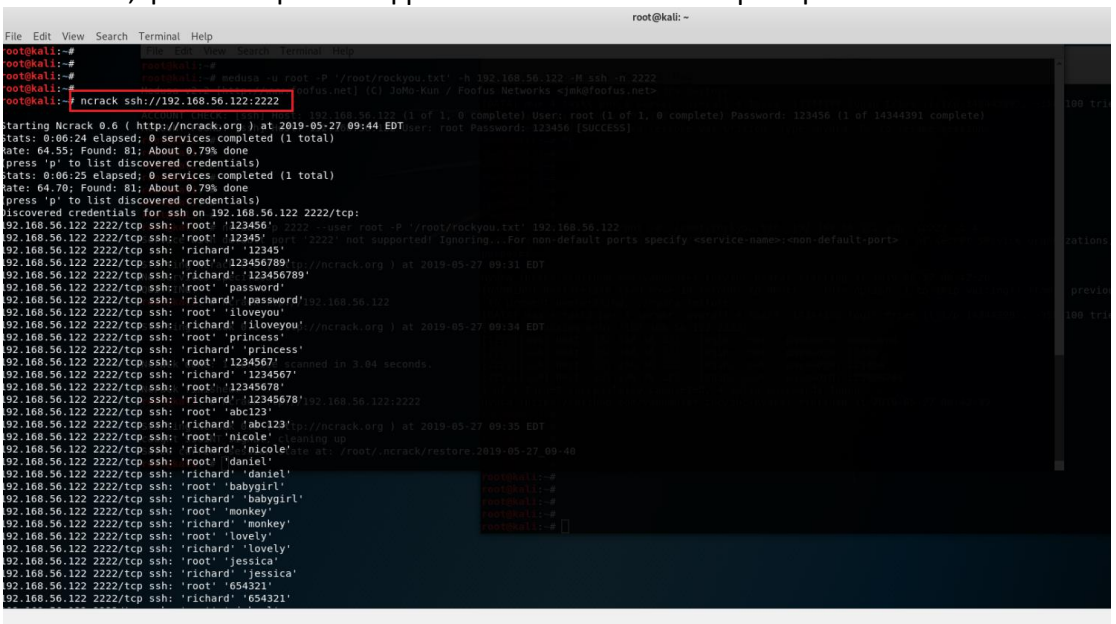
```
# hydra -l root -P ‘/root/rockyou.txt’ 192.168.56.122 -s2222 -t 4
```



Εικόνα 36. Επίθεση Hydra από Kali Linux.

Brute Force επίθεση με Ncrack

Στην Εικόνα 37, φαίνεται η επίθεση με NCrack στον ssh διακομιστή 192.168.56.122.



Εικόνα 37. Επίθεση NCrack από Kali Linux.

Brute Force επίθεση με Medusa

Στην Εικόνα 38, φαίνεται η επίθεση με Medusa χρησιμοποιώντας την λίστα κωδικών rockyou.txt (/root/rockyou.txt) στον συγκεκριμένο ssh server (192.168.56.122 και port:222 {-n 2222}), προσπαθώντας να συνδεθούμε ως root χρήστης.

```
# medusa -u root -P 'root/rockyou.txt' -h 192.168.56.122 -M ssh -n 2222
```



Εικόνα 38. Επίθεση Medusa από Kali Linux.

Κεφάλαιο 5

Wireshark

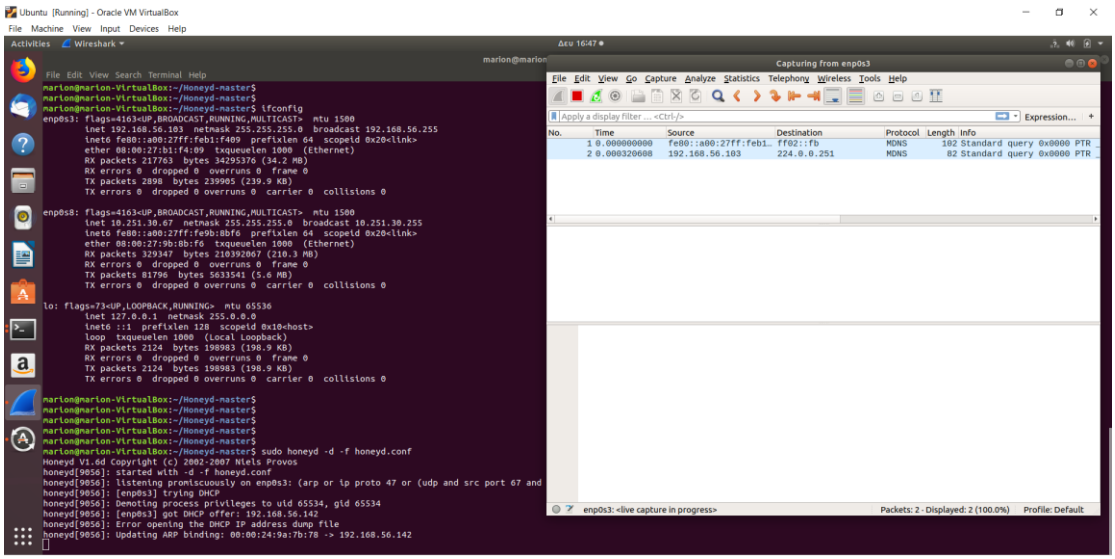
Το Wireshark είναι ένα ελεύθερο και ανοιχτού κώδικα λογισμικό ανάλυσης πρωτοκόλλων δικτύου υπολογιστών. Το wireshark χρησιμοποιείται κυρίως για ανάλυση και παρακολούθηση δικτύου. Παρόλα αυτά, μπορεί να χρησιμοποιηθεί επίσης για εντοπισμό και αντιμετώπιση προβλημάτων, καθώς και για εκπαιδευτικούς λόγους.
Το Wireshark είναι διαθέσιμο για UNIX και Windows. Ουσιαστικά, συλλαμβάνει «ζωντανά» τα πακέτα από μία διεπαφή δικτύου. Φιλτράρει, αναζητά, χρωματίζει τα πακέτα βάσει πολλών κριτηρίων και δημιουργεί διάφορα στατιστικά.

Στην περίπτωσή μας, χρησιμοποιούμε το Wireshark για να δούμε την κίνηση του δικτύου κατά τη διάρκεια μίας επίθεσης, να αναλύσουμε τα πακέτα και να δούμε ποια πρωτόκολλα χρησιμοποιούν, και τέλος, να εξάγουμε κάποια στατιστικά στοιχεία για τις επιθέσεις.

Για να εγκαταστήσουμε το Wireshark στο Ubuntu μηχάνημα, εκτελούμε την εντολή:

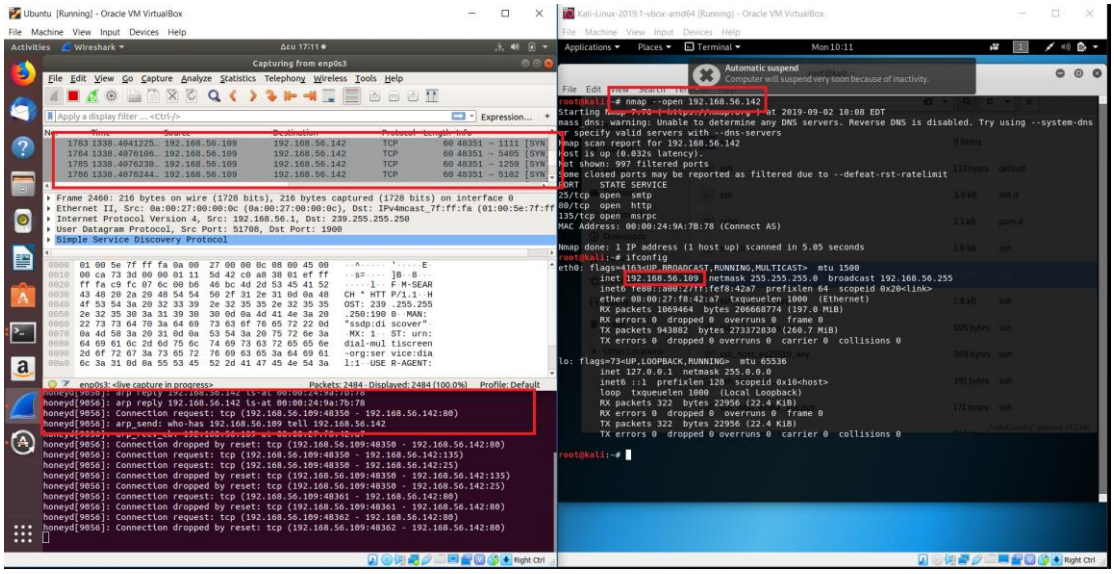
```
# sudo apt-get update
#sudo apt install wireshark
```

Για να δούμε πώς λειτουργεί το Wireshark, θα τρέξουμε τον Honeyd server όπως φαίνεται στην Εικόνα 39, και ύστερα θα κάνουμε μία επίθεση από Kali Linux για να δούμε την κίνηση των πακέτων στο δίκτυο.



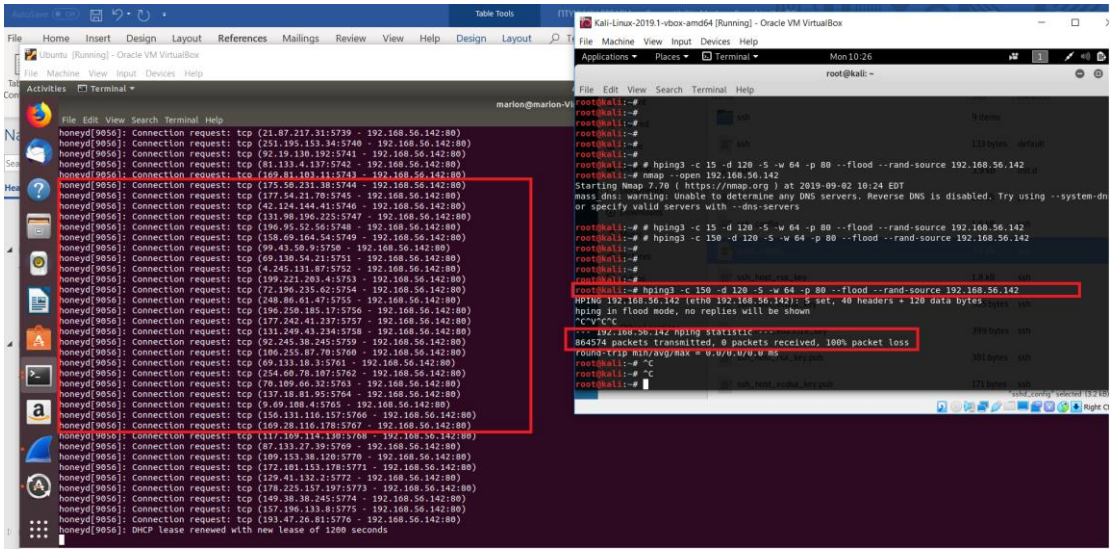
Εικόνα 39. Εκτέλεση Honeyd server.

Αρχικά, σαρώνουμε την διεύθυνση όπου τρέχει ο Honeyd server, ή οποία είναι η 192.168.56.142, όπως φαίνεται στην Εικόνα 40.



Εικόνα 40. Σάρωση της διεύθυνσης όπου τρέχει ο Honeyd.

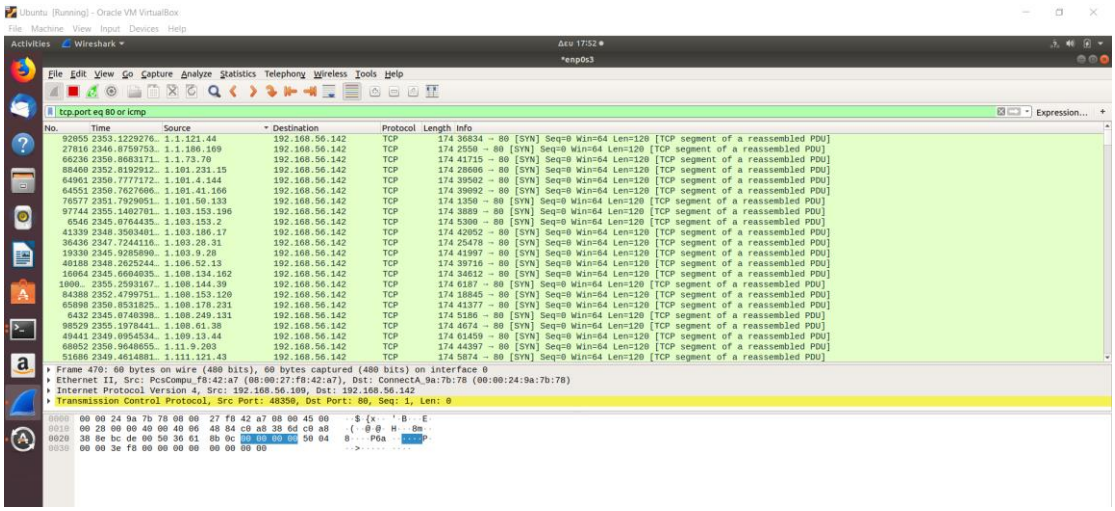
Κάνουμε επίθεση SYN στην πόρτα 80, όπως φαίνεται στην Εικόνα 42.



Εικόνα 41. Επίθεση SYN από Kali Linux με IP 192.168.56.109.

Ας δούμε τώρα τις δυνατότητες του Wireshark να αναζητήσει και να φιλτράρει τις κινήσεις του δικτύου, ανάλογα με τα κριτήρια που θα δώσουμε. Μερικά από αυτά φαίνονται παρακάτω.

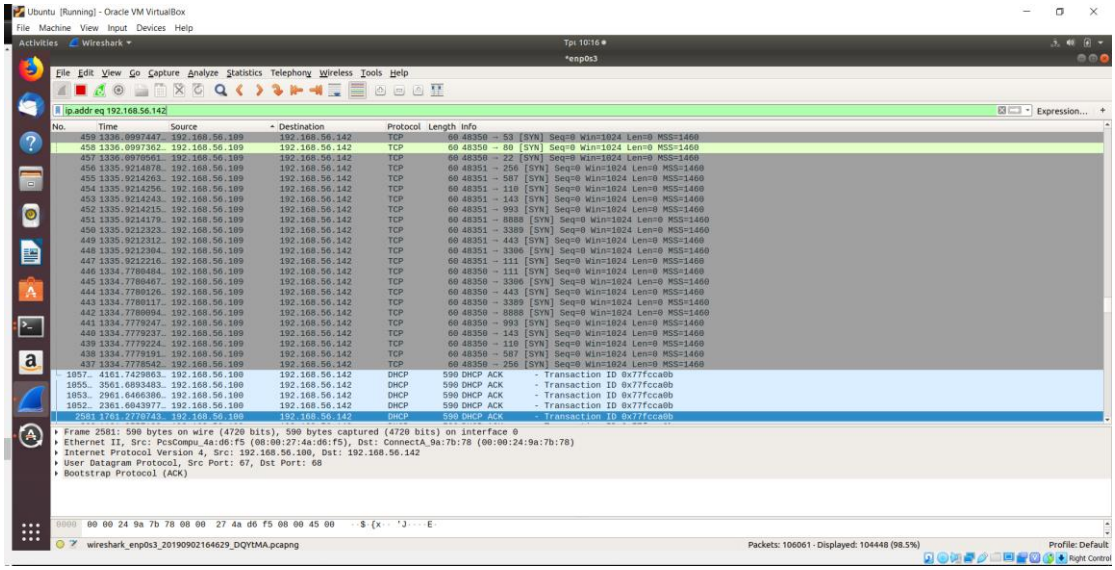
```
tcp.port eq 80 or icmp
ip.addr eq 192.168.56.142
not arp and !(udp.port == 53)
icmp
ip
http.request.method == GET
http.request.method == POST
```



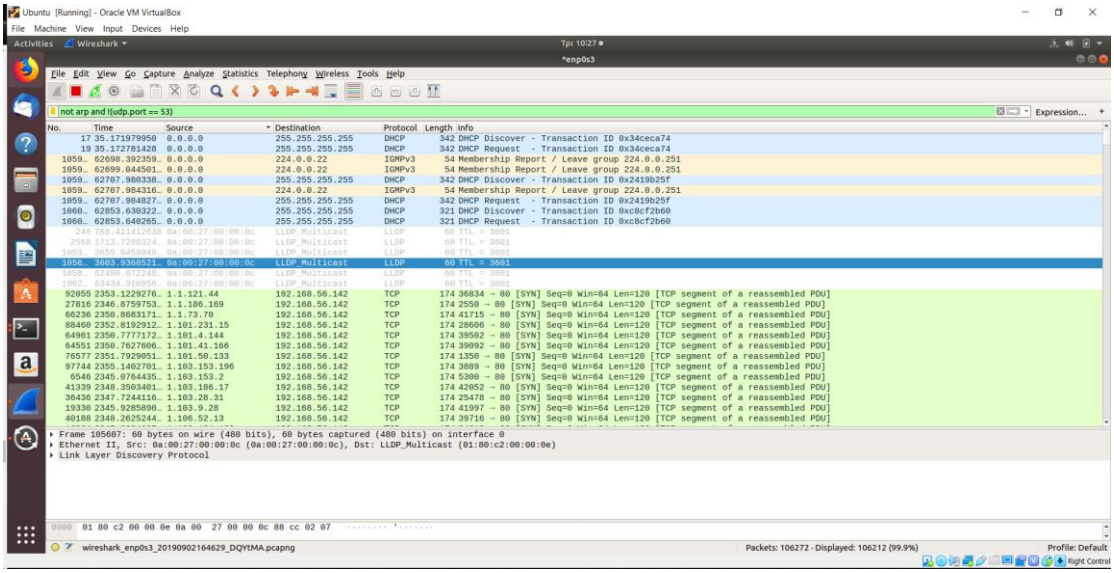
Εικόνα 42. Κίνηση δικτύου, πρωτόκολλο TCP και πόρτα 80 ή IMCP.

Στην Εικόνα 42, βλέπουμε ότι πακέτα SYN (Info) έχουν σταλεί με TCP (Protocol) στην διεύθυνση 192.168.56.142 (destination) , καθώς η διεύθυνση πηγή (source) φαίνεται να είναι άλλη κάθε φορά και ποτέ η αληθινή, του μηχανήματος δηλαδή, Kali που κάναμε την επίθεση (192.168.56.109).

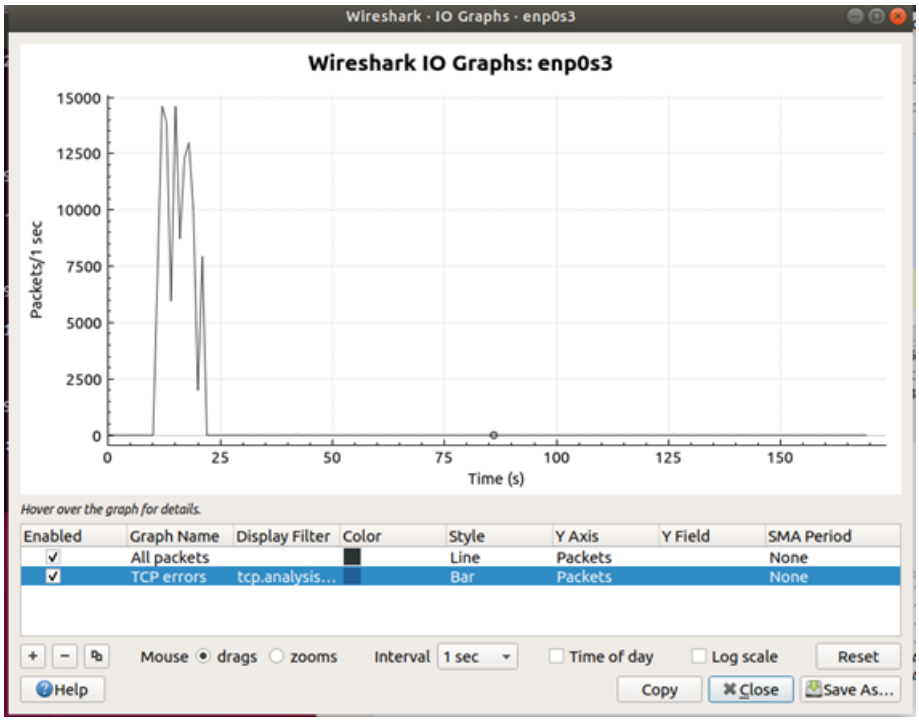
Στην Εικόνα 43, φαίνεται η κίνηση των πακέτων από και προς την IP 192.168.56.142.



Εικόνα 43. Απεικόνιση κίνησης δικτύου σε ότι έχει να κάνει με την IP 192.168.56.142.



Εικόνα 44. Κίνηση πακέτων που δεν χρησιμοποιούν ARP και η udp πόρτα δεν είναι 53.



Εικόνα 45. I/O Γράφημα για στατιστική ανάλυση των κινήσεων του δικτύου.

Κεφάλαιο 6

6.1 Συμπεράσματα

Η πτυχιακή αυτή, έχει στόχο τόσο την ανάλυση και υλοποίηση κάποιων χαμηλής και μεσαίας αλληλεπίδρασης Honeypots, όσο και την παρουσίαση των βασικών αρχών των δικτύων.

Η ανάλυση και η υλοποίηση των παραπάνω honeypots, προϋποθέτει την γνώση βασικών αρχών δικτύων. Στο Κεφάλαιο 2, αναγράφονται όλες τις βασικές αρχές που πρέπει κάποιος να γνωρίζει για να προχωρήσει στην υλοποίηση των Honeypots. Στις βασικές αρχές των δικτύων συμπεριλαμβάνεται θεωρία περί πρωτοκόλλων, πακέτων, διευθύνσεων, τειχών προστασίας κ.ά.

Στο Κεφάλαιο 3, αναφέρεται η θεωρία των honeypots που έχουν υλοποιηθεί, honeyd, kippo και cowrie καθώς και η θεωρία των επιθέσεων που έχουν πραγματοποιηθεί -syn, smurff , xmas, hydra, medusa, ncrack- επιθέσεις.

Τέλος, στο κεφάλαιο 4, αναγράφεται η υλοποίηση των honeypots και οι δοκιμές της λειτουργικότητας τους μέσω επιθέσεων από μηχάνημα Kali Linux, ενώ στο κεφάλαιο 5, αναφέρεται το εργαλείο Wireshark, η χρήση του, αλλά και το πόσο σημαντικό είναι για την εξασφάλιση της ασφάλειας των δικτύων.

Συμπερασματικά, η πτυχιακή αυτή εμπεριέχει την απαραίτητη θεωρία δικτύων, την υλοποίηση, και τις πλήρεις δοκιμές με τις πλήρεις λειτουργίες. Και οι τρεις honeypots, μπορούν να διαδραματίσουν έναν καλό ρόλο καθιστώντας την ασφάλεια της πληροφορικής πιο ασφαλή.

6.2 Μελλοντική έρευνα

Διαβάζοντας για τα δίκτυα και την ασφάλειά τους, καταλαβαίνει κανείς ότι δεν υπάρχει κάποια γραμμή για τη σήμανση του τέλους στον τομέα αυτόν. Το μόνο που μπορεί να υποσχεθεί κανείς, είναι η συνεχής μελέτη και εξάσκηση. Η παραπάνω ανάλυση, μου θύμισε πόσο ενδιαφέρων είναι ο τομέας της ασφάλειας των δικτύων και πόσο θα ήθελα να εμβαθύνω σε αυτόν στο μέλλον.

Διαβάζοντας την θεωρία των δικτύων και αναλύοντας τους συγκεκριμένους honeypots, τις υλοποιήσεις, τις διαμορφώσεις και τις επιθέσεις, αναδύθηκαν ενθουσιασμός και απορία. Ενθουσιασμός, γιατί απέκτησα πολλή γνώση και αντίληψη περί δικτύων, και απορία, γιατί δημιουργήθηκαν τόσα ερωτήματα και απορίες στο μυαλό μου.

Μερικές από τις σκέψεις που είχα κατά τη διάρκεια της μελέτης μου ήταν -πώς γίνεται η κρυπτογράφηση καθαυτή; Ποιοι είναι αυτοί οι αλγόριθμοι; Γιατί είναι τόσο περίπλοκοι; Πότε θα καταφέρω να υλοποιήσω έναν High Interaction Honeypot; Είναι τόσο δύσκολο; Πρέπει να έχεις μεγάλη εμπειρία;

Έχοντας όλες αυτές τις σκέψεις στο μυαλό μου, θα ήθελα στο μέλλον να ασχοληθώ περαιτέρω και να μελετήσω περισσότερο για την Ασφάλεια των Δικτύων. Συγκεκριμένα, οι ιδέες για μελλοντική έρευνα όσον αφορά την Ασφάλεια είναι:

- Έρευνα και ανάλυση αλγορίθμων κρυπτογράφησης.
- Ανάλυση και Υλοποίηση ενός High Interaction Honeypot.
- Έρευνα πάνω στο bug του Kippo κατά την αποκρυπτογράφηση των κλειδιών με Diffie-Helman.
- Συνεχής έρευνα στις νέες τεχνολογίες σχετικές με την Ασφάλεια Δικτύων.

Ευελπιστώ λοιπόν, ότι θα ασχοληθώ με την συγκεκριμένη μελλοντική έρευνα, είτε για ακαδημαϊκούς λόγους, είτε επαγγελματικά.

Βιβλιογραφία

1. Provos, N., & Holz, T. (2007). *Virtual honeypots: from botnet tracking to intrusion detection*. Pearson Education.
2. Provos, N. (2003, February). Honeyd-a virtual honeypot daemon. In *10th DFN-CERT Workshop, Hamburg, Germany*(Vol. 2, p. 4).
3. Provos, N. (2004, August). A Virtual Honeypot Framework. In *USENIX Security Symposium* (Vol. 173, No. 2004, pp. 1-14).
4. Lyon, G. F. (2009). *Nmap network scanning: The official Nmap project guide to network discovery and security scanning*. Insecure.
5. Sochor, T., & Zuzcak, M. (2014, June). Study of internet threats and attack methods using honeypots and honeynets. In *International Conference on Computer Networks* (pp. 118-127). Springer, Cham.
6. Orebaugh, A., Ramirez, G., & Beale, J. (2006). *Wireshark & Ethereal network protocol analyzer toolkit*. Elsevier.
7. Wang, S., Xu, D., & Yan, S. (2010, April). Analysis and application of Wireshark in TCP/IP protocol teaching. In *2010 International Conference on E-Health Networking Digital Ecosystems and Technologies (EDT)* (Vol. 2, pp. 269-272). IEEE.
8. Vigna, G., & Kemmerer, R. A. (1998, December). NetSTAT: A network-based intrusion detection approach. In *Proceedings 14th Annual Computer Security Applications Conference (Cat. No. 98EX217)* (pp. 25-34). IEEE.
9. Bogdanoski, M., Suminoski, T., & Risteski, A. (2013). Analysis of the SYN flood DoS attack. *International Journal of Computer Network and Information Security (IJCNIS)*, *5*(8), 1-11.
10. Baumann, R. (2003). Honeyd–A low involvement Honeypot in Action. *Originall published as part of the GCIA practical*, 14.
11. Najera-Gutierrez, G., & Ansari, J. A. (2018). *Web Penetration Testing with Kali Linux: Explore the methods and tools of ethical hacking with Kali Linux*. Packt Publishing Ltd.
12. Borrero, J. CRACKING YOUR PASSWORD.
13. Mokube, I., & Adams, M. (2007, March). Honeypots: concepts, approaches, and challenges. In *Proceedings of the 45th annual southeast regional conference* (pp. 321-326). ACM.
14. Wicherski, G. (2006). Medium interaction honeypots. *German Honeynet Project*.
15. Peter, E., & Schiller, T. (2011). A practical guide to honeypots. *Washington Univerity*.
16. Ayeni, O. A., Alese, B. K., & Omotosho, L. O. (2013). Design and implementation of a medium interaction honeypot. *International Journal of Computer Applications*, *975*, 8887.
17. Getz, K. (1997). Dynamic host configuration protocol.
18. Tingley, C. T., & Walsh, R. J. (2007). *U.S. Patent No. 7,260,648*. Washington, DC: U.S. Patent and Trademark Office.
19. Comer, D. E., & Droms, R. E. (2003). *Computer networks and internets*. Prentice-Hall, Inc..
20. Egevang, K., & Francis, P. (1994). *The IP network address translator (NAT)* (pp. 1-10). RFC 1631, may.
21. Harris, B., & Hunt, R. (1999). TCP/IP security threats and attack methods. *Computer communications*, *22*(10), 885-897.
22. Fettig, A., & Lefkowitz, G. (2005). *Twisted network programming essentials*. " O'Reilly Media, Inc.".
23. Diffie, W., Van Oorschot, P. C., & Wiener, M. J. (1992). Authentication and authenticated key exchanges. *Designs, Codes and cryptography*, *2*(2), 107-125.
24. Hickman, K., & Elgamal, T. (1995). The SSL protocol.
25. Fraunholz, D., Krohmer, D., Anton, S. D., & Schotten, H. D. (2017, June). Investigation of cyber crime conducted by abusing weak or default passwords with a medium interaction honeypot. In *2017 International Conference on Cyber Security And Protection Of Digital Services (Cyber Security)* (pp. 1-7). IEEE.

Σχετικά με τον συγγραφέα

Η Μαρινέλλα Μπίλα, γεννήθηκε στις 23 Μαρτίου του 1994 στην Αθήνα. Μεγάλωσε στην Αθήνα και άρχισε να φοιτά στο τμήμα της Πληροφορικής και Τηλεματικής στο Χαροκόπειο Πανεπιστήμιο το 2012.
Τον Φεβρουάριο του 2015 μετακόμισε στην Κρακοβία, στην Πολωνία. Σπούδασε εκεί για έξι μήνες ως φοιτήτρια Erasmus. Τον Σεπτέμβριο του 2017, μετακόμισε στο Μόναχο, στη Γερμανία. Εργάστηκε στην Infineon Technologies, μέχρι που της έγινε πρόταση για δουλειά στο CERN. Έτσι, τον Απρίλιο του 2018, μετακόμισε στη Γενεύη και εργάστηκε στο CERN. Ωστόσο, η επιθυμία της να είναι κοντά στην οικογένειά της και στα αγαπημένα της πρόσωμα, την έκανε να επιστρέψει στην Ελλάδα. Από τότε εργάζεται στην Intrasoft International, ως Σύμβουλος Πληροφορικής.