**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ**

**Τμήμα Διδακτικής της Τεχνολογίας και Ψηφιακών Συστημάτων**

# Μελέτη Χαμηλής και Υψηλής Αλληλεπίδρασης Honeypots

**Πτυχιακή Εργασία**

Συγγραφέας: Τρούλης Σ. Ιωάννης

Υπεύθυνος Καθηγητής: Δρ. Κ. Λαμπρινουδάκης

Σε συνεργασία με το Εργαστήριο Islab, του Ινστιτούτου Πληροφορικής, του ΕΚΕΦΕ Δημόκριτος

Υπεύθυνος Ερευνητής: Δρ. Ι. Κοροβέσης

**Μάιος 2010**

# Ευχαριστίες

Η παρουσία μου στο ISLAB του ινστιτούτου Πληροφορικής&Τηλεπικοινωνιών του ΕΚΕΦΕ Δημόκριτος, πλάι σε καταξιωμένους επιστήμονες, αποτέλεσε για εμένα μια πολύ σημαντική εμπειρία. Εκτός από τις πολύ σημαντικές γνώσεις που αποκόμισα η παρουσία μου στο Εργαστήριο με έκανε να αναθεωρήσω αρκετά πράγματα και καταστάσεις.

Θα ήθελα να ευχαριστήσω θερμά τον υπεύθυνο ερευνητή και ιδρυτή του εργαστηριού Islab, κ.Ι.Κοροβέση που με δέχθηκε στα στους χώρους του εργαστηρίου, όπως και για την απρόσκοπτη καθοδήγηση του όλο αυτό το διάστημα.

Ιδιαιτέρως θα ήθελα να ευχαριστήσω τον κ. Κ.Μάγκο για την βοήθεια του στην επιλογή των θεμάτων της πτυχιακής εργασίας, την επίλυση των όποιων προβλημάτων προέκυπταν, όπως επίσης και για την ανεξάντλητη υπομονή του κατά την κοπιαστική εργασία διόρθωσης της. Χωρίς την καθοριστική συμβολή του η τελική μορφή της εργασίας θα ήταν αρκετά διαφορετική.

Τέλος θα ήθελα επίσης να ευχαριστήσω και τα υπόλοιπα μέλη του Εργαστηρίου, κ.Ν.Μαρούγκα και κ.Β.Νέσσυ για την επίσης πολύτιμη βοήθεια τους και για τις χρήσιμες συμβουλές τους.

# Πρόλογος

Το όνομα "Internet Systematics Lab" σηματοδοτεί την τρέχουσα φάση που βρίσκεται το Εργαστήριο μας μετά τις προηγούμενες φάσεις "Αριάδνη Τι?" και "Ερμής" που είχαν αντικείμενο την δημιουργία του Εθνικού Δικτύου Έρευνας και του σχετικού ανθρωπο-δικτύου στην Ελλάδα. Σχετικές περιγραφές βρίσκουμε στις ιστο-σελίδες:

www.ariadne-t.gr, www.ariadne-t.gr/epmhs.htm.

Η "συστηματική" αφορά την δημιουργία μαθησιακού περιεχομένου και την επικοινωνία της Γνώσης.
Το Εργαστήριο μας έχει σαφή προσανατολισμό τα φαινόμενα του Διαδικτύου και του Ελεύθερου/Ανοικτού Λογισμικού με πεδίο εφαρμογής την Ασφάλεια και την Αξιοπιστία της Διαδικτυακής Υποδομής καθώς λειτουργούμε και αναπτύσσουμε την υποδομή του ΕΚΕΦΕ 'Δ'.
Στο Data Center μας έχουμε τους κόμβους του ΕΔΕΤ και του Hellasgrid και συνεργαζόμαστε με τις κοινότητες τους.

Η ποσότητα Γνώσεων & Δεξιοτήτων που μεταδίδεται μέσω των "τεχνολογικών κοινοτήτων" αποκτά ολοένα και μεγαλύτερο στρατηγικό ρόλο. Από το 2000 επικεντρώσαμε στην περιοχή "Ασφάλεια Διαδικτύου" και αναπτύξαμε το πρόγραμμα του Εργαστηρίου βασισμένο σε εργαλεία Ελεύθερου Λογισμικού. Το Εργαστήριο έχει χαράξει ένα δρόμο που στηρίζεται στην προώθηση του "open source" σε ΑΕΙ, ΑΤΕΙ και το 2009 ιδρύσαμε μαζί τους την Εταιρεία ΕΛ/ΛΑΚ (www.ellak.gr).

Ο ρόλος που καλείται να παίξει ο σπουδαστής στο Εργαστήριο ακολουθώντας το μοντέλο "reflected learning path" τον αναδεικνύει σταδιακά σε πρωτοπόρο της γνώσης για την τοπικότητά μας. Έχουμε αναπτύξει ένα Content Management System που στηρίζει την καταγραφή και την επικοινωνία της Γνώσης από/προς τα μέλη του Εργαστηρίου. Το περιεχόμενο αφορά τον εμπλουτισμό του CMS με εμπειρίες μάθησης. Τα παραδείγματα των εργασιών που εξωτερικεύουν την δουλειά που έχει γίνει μέχρι σήμερα περιγράφονται στις ιστοσελίδες (κατηγορία ΠΤΥΧΙΑΚΕΣ): www.islab.demokritos.gr.

Η τρέχουσα εργασία του κ.Ι.Τρούλη με θέμα το "Μελέτη Χαμηλής και Υψηλής Αλληλεπίδρασης Honeypots" εφαρμόζει τη παραπάνω φιλοσοφία και αποτελεί ένα άριστο παράδειγμα.

Δρ.Ι.Κοροβέσης (ycor@iit.demokritos.gr)
Εργαστήριο Δικτύων
Ινστιτούτο Πληροφορικής & Τηλεπικοινωνιών
ΔΗΜΟΚΡΙΤΟΣ
Απρίλιος 2010

# ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

# Κατάλογος Σχημάτων

# Κατάλογος Πινάκων

# Ευρετήριο Όρων στην Ελληνική Γλώσσα

**Ανιχνευτές:** Είναι εφαρμογές οι οποίες επιτρέπουν στον χρήστη τους να συγκεντρώσει πολλές πληροφορίες για απομακρυσμένα μηχανήματα, χωρίς να έχει φυσική πρόσβαση σε αυτά.

**Ανιχνευτές Ενεργητικοί:** Οι ενεργητικοί ανιχνευτές συγκεντρώνουν τις απαραίτητες πληροφορίες αποστέλλοντας πακέτα στον απομακρυσμένο υπολογιστή και εξετάζοντας τις απαντήσεις. Συνήθως γίνονται εύκολα αντιληπτοί λόγω του μεγάλου όγκου που αρχείων που αποστέλλουν.

**Ανιχνευτές Παθητικοί:** Οι παθητικοί ανιχνευτές συγκεντρώνουν τις απαραίτητες πληροφορίες παρακολουθώντας τις συνομιλίες με τον απομακρυσμένο υπολογιστή και εξετάζοντας τα εισερχόμενα πακέτα. Δεν γίνονται αντιληπτοί από τον απομακρυσμένο υπολογιστή λόγω του ότι δεν παράγουν κανενός είδος δικτυακή κίνηση προς αυτόν.

**Αρχεία καταγραφής συμβάντων:** Αρχεία στα οποία καταγράφονται όλα τα συμβάντα (ο,τι συμβαίνει ή προκύπτει) στα πλαίσια μιας εφαρμογής ή ενός συστήματος. Κάθε εφαρμογή συνήθως περιλαμβάνει τα δικά της αποκλειστικά αρχεία καταγραφής συμβάντων.

**Διακομιστές:** Είναι απομακρυσμένα συστήματα που ως σκοπό έχουν την εξυπηρέτηση των αιτήσεων που τους υποβάλλονται.

**Ξενιστής:** Ξενιστές ονομάζονται τα συστήματα τα οποία φιλοξενούν εικονικά μηχανήματα.

**Σάρωση:** Η διαδικασία που εκτελείτε από τους ενεργητικούς ανιχνευτές προκειμένου να συγκεντρώσουν πληροφορίες για τον απομακρυσμένο υπολογιστή.

# Ευρετήριο Όρων στην Αγγλική Γλώσσα

**Cache poisoning:** Είδος επίθεσης στους διακομιστές DNS. Μέσω αυτής της επίθεσης επιτυγχάνεται η τροποποίηση μιας εγγραφής στην μνήμη cache ενός διακομιστή DNS έτσι ώστε αυτή να οδηγεί σε λανθασμένα αποτελέσματα.

**Drop:** Η παράμετρος "drop" χρησιμοποιείται για να καταδείξει πως όλα τα εισερχόμενα μηνύματα θα απορρίπτονται χωρίς να στέλνεται στον αποστολέα τους κάποιο προειδοποιητικό μήνυμα.

**Penetration test:** Αποτελεί μια μέθοδο αξιολόγησης του επιπέδου ασφαλείας ενός οργανισμού, μιας εταιρίας ή κάποιας υπηρεσίας. Σκοπός ενός "Penetration Test" είναι η ανακάλυψη όλων των κενών ασφαλείας, οποιασδήποτε μορφής (φυσική πρόσβαση, δικτυακή επίθεση κ.α), που θα μπορούσαν να οδηγήσουν στην επιτυχημένη εκτέλεση κάποιας επίθεσης..

**Script:** Ένα σύνολο εντολών που αποσκοπεί στην πραγματοποίηση μιας συγκεκριμένης εργασίας.

**Reject:** Η παράμετρος "reject" χρησιμοποιείται για να καταδείξει πως όλα τα εισερχόμενα μηνύματα θα απορρίπτονται και θα στέλνεται στον αποστολέα τους κάποιο προειδοποιητικό μήνυμα.

**IDS:** Συστήματα Ανίχνευσης Επιθέσεων. Έχουν ως σκοπό τους την ανίχνευση ενδεχόμενων παραβιάσεων στα συστήματα τα οποία επιβλέπουν.

**Sdrop:** Η παράμετρος "sdrop" (silence drop) χρησιμοποιείται για να καταδείξει πως όλα τα εισερχόμενα μηνύματα θα απορρίπτονται χωρίς να στέλνεται στον αποστολέα τους κάποιο προειδοποιητικό μήνυμα και χωρίς αυτό το περιστατικό να καταγράφεται στα αρχεία καταγραφής συμβάντων.

**Signatures:** Τα "signatures" αποτελούν δεδομένα που προκύπτουν κατά την εκτέλεση ιών, trojans ή διαφόρων ειδών επιθέσεων. Τα δεδομένα αυτά (που παράγονται από την κάθε επίθεση) τείνουν να είναι τόσο μοναδικά που λειτουργούν ως μέσο αναγνώρισης των επιθέσεων αυτών.

**Signature scanning:** Ο όρος “signature scanning” αναφέρεται στην αναγνώριση και ταυτοποίηση συγκεκριμένου τύπου επιθέσεων ή διαφόρων μορφών κακόβουλου λογισμικού (ιοί, trojans, malwares κ.α.) με βάση τα signatures που παράγουν όταν εκτελούνται.

# ΚΕΦΑΛΑΙΟ 1 - ΕΙΣΑΓΩΓΗ

## Ασφάλεια Πληροφοριακών Συστημάτων

Η ασφάλεια πληροφοριακών συστημάτων στοχεύει στην προστασία των πληροφοριών. Ως πληροφορία ορίζονται όλα αυτά τα δεδομένα τα οποία έχουν αξία για τον ιδιοκτήτη τους. Η αξία τους όμως είναι πολύ πιθανό να μειωθεί αν υποστούν ζημιά όπως κλοπή, τροποποίηση ή καταστροφή. Η προστασία των πληροφοριών είναι απαραίτητη εφόσον πάντα υπάρχουν κίνδυνοι που μπορούν να προκαλέσουν ζημιά. Για την προστασία των πληροφοριών ο ιδιοκτήτης τους χρησιμοποιεί κάποια μέσα προστασίας. Αυτά τα μέσα χρησιμοποιούνται είτε για να μειώσουν τον κίνδυνο να προκληθεί κάποια ζημιά είτε για να μειώσουν τις συνέπειες αυτής. Συνοψίζοντας η ασφάλεια πληροφοριακών συστημάτων έχει ως σκοπό της την διάγνωση των κινδύνων και την αποτελεσματική εφαρμογή μέσων προστασίας ώστε τα αγαθά να μην υποστούν κάποια ζημιά.

## Κύριες Έννοιες Ασφάλειας Πληροφοριών

Τρεις είναι οι βασικές έννοιες της ασφάλειας πληροφοριών. Οι έννοιες αυτές αποτελούν τον πυρήνα της ασφάλειας πληροφοριών για παραπάνω από είκοσι χρόνια και είναι οι εξής:

1. **Εμπιστευτικότητα:** Η εμπιστευτικότητα αναφέρεται στην προστασία των πληροφοριών από την αποκάλυψη τους σε μη εξουσιοδοτημένα άτομα.

2. **Ακεραιότητα:** Η ακεραιότητα αναφέρεται στην προστασία των πληροφοριών από την μη εξουσιοδοτημένη τροποποίηση τους.

3. **Διαθεσιμότητα:** Η διαθεσιμότητα αναφέρεται στην συνεχή και αδιάλειπτη παροχή των πληροφοριών στους εξουσιοδοτημένους χρήστες.

Όλες οι τεχνικές διασφάλισης και όλα τα μέσα προστασίας των πληροφοριών και των δεδομένων, θα πρέπει να έχουν ως στόχο τους την τήρηση των παραπάνω «αρχών».

## Τομείς Ασφάλειας

Για να μπορέσουμε να μελετήσουμε καλύτερα την ασφάλεια υπολογιστικών συστημάτων αλλά και να μπορέσουμε να κατηγοριοποιήσουμε καλύτερα τα διάφορα εργαλεία και μέσα που χρησιμοποιούνται θα μπορούσαμε να χωρίσουμε την ασφάλεια υπολογιστών σε τρεις τομείς.

- **Πρόληψη:** Η πρόληψη έχει τον ρόλο να κρατήσει τους κακόβουλους χρήστες μακριά από τα μηχανήματα παραγωγής. Αυτό επιτυγχάνεται μέσα από την χρήση firewalls, IPS (Intrusion Prevention Systems) καθώς και μέσα από το σωστό patching των συστημάτων.

- **Ανίχνευση:** Η ανίχνευση (detection) όπως φανερώνει και το όνομα της έχει το ρόλο της έγκαιρης προειδοποίησης για κάποια επίθεση. Την ανίχνευση δηλαδή κάποιας επίθεσης την ώρα που αυτή λαμβάνει χώρα. Κάτι τέτοιο επιτυγχάνεται από τα IDS (Intrusion Detection Systems)

- **Απόκριση:** Η απόκριση (response) περιλαμβάνει όλες τις κινήσεις που πρέπει να γίνουν και όλα τα μέτρα τα οποία θα πρέπει να ληφθούν αφού εντοπιστεί ή πραγματοποιηθεί κάποια επίθεση ώστε να περιοριστεί η ζημιά και να διασφαλιστεί ότι αυτή η επίθεση δεν θα συμβεί ξανά ή στην περίπτωση που ξανασυμβεί τα δεδομένα δεν θα υποστούν καμία ζημιά.

## Τύποι Επιθέσεων

Οι δικτυακές επιθέσεις μπορούν να κατηγοριοποιηθούν ανάλογα με την ζημιά που προκαλούν στους παρακάτω τύπους:

- **Άρνηση Υπηρεσίας:** Οι επιθέσεις άρνησης υπηρεσίας ή DoS (Denial of Service) όπως είναι ευρέως γνωστές, έχουν στόχο τους όπως φανερώνει και το όνομα τους, την διακοπή παροχής υπηρεσιών από την πλευρά των συστημάτων που δέχονται την επίθεση. Αυτό

επιτυγχάνεται συνήθως όταν ο επιτιθέμενος αποστέλλει μεγάλο αριθμό αιτήσεων εξυπηρέτησης στον διακομιστή-στόχο που αυτός αδυνατεί να τις διεκπεραιώσει στον προκαθορισμένο χρόνο. Το αποτέλεσμα είναι να γεμίζει η ουρά και να μην εξυπηρετούνται οι νόμιμες αιτήσεις. Τέτοιου τύπου επιθέσεις σε περίπτωση επιτυχής περάτωσης τους, αποφέρουν σημαντικά πλήγματα αξιοπιστίας αλλά και μείωση κερδοφορίας κυρίως σε επιχειρήσεις παροχής υπηρεσιών και πώλησης αγαθών (π.χ. αεροπορικές εταιρίες, ηλεκτρονικά καταστήματα, μηχανές αναζήτησης) οι οποίες και χρειάζεται να παρέχουν συνεχή και αδιάκοπη λειτουργία.

- **Επιθέσεις Πλαστοπροσωπίας (Spoofing):** Σε αυτές τις ηλεκτρονικές επιθέσεις ο επιτιθέμενος τροποποιεί κατάλληλα τις αιτήσεις έτσι ώστε να φαίνεται ότι προέρχονται από άλλη πηγή. Κατά αυτό τον τρόπο μπορεί να ξεγελάσει τα συστήματα-στόχους προσποιούμενος κάποιο έμπιστο μηχάνημα. Γνωστές επιθέσεις αυτού του είδους είναι συνήθως οι IP spoofing όπου κάποιος χρήστης τροποποιεί την διεύθυνση IP στα πακέτα που αποστέλλει.

- **Επιθέσεις Λαθρακρόασης (Eavesdropping):** Είναι μια παθητική επίθεση που πραγματοποιούνται από άτομα τα οποία παρεμβάλλονται στην επικοινωνία δύο υπολογιστών, αποκαλύπτουν το περιεχόμενο της και παραβιάζουν το απόρρητο της επικοινωνίας. Αυτό μπορεί να επιτευχθεί με διάφορους τρόπους. Ο πιο κοινός είναι με τη χρήση προγραμμάτων καταγραφής δικτυακής κίνησης (sniffers).

- **Επιθέσεις Επιπέδου Εφαρμογής:** Οι επιθέσεις αυτού του τύπου εκμεταλλεύονται αδυναμίες και κενά ασφαλείας που υπάρχουν σε εφαρμογές και προγράμματα τα οποία χρησιμοποιεί ο χρήστης ή οι υπηρεσίες που προσφέρει ένας διακομιστής. Παραδείγματα τέτοιου τύπου επιθέσεων αποτελούν τα trojans, οι ιοί, επιθέσεις στον διακομιστή Διαδικτύου (Web server), SQL Injection, Buffer overflow. Ανάλογα με τον τύπο της αδυναμίας ο επιτιθέμενος μπορεί από το να τροποποιήσει η να υποκλέψει δεδομένα μέχρι και να αποκτήσει πρόσβαση στη μηχανή.

- **Επιθέσεις Παράνομης Παραβίασης:** Οι επιθέσεις αυτές στοχεύουν στην παράνομη, εκ

μέρους κακόβουλων χρηστών, είσοδο σε κάποιο μηχάνημα δια μέσω του δικτύου. Σε αυτές περιλαμβάνονται επιθέσεις εναντίων των συστημάτων αυθεντικοποίησης, επιθέσεις κοινωνικής μηχανικής και βέβαια όπως αναφέραμε παραπάνω επιθέσεις εναντίον εφαρμογών.

## Honeypots

Τα honeypots αποτελούν παθητικά συστήματα ασφαλείας. Δεν συμμετέχουν δηλαδή ενεργά στην ασφάλεια υπολογιστών, αποτρέποντας δικτυακές επιθέσεις όπως άλλα συστήματα ασφαλείας (firewalls, IPS), αλλά παθητικά συλλέγοντας πληροφορίες. Τα honeypots αποτελούν παγίδες κατάλληλα παραμετροποιημένες, ώστε να προσομοιάζουν πραγματικά συστήματα με σκοπό την καταγραφή δεδομένων δικτυακών επιθέσεων. Η ανάλυση αυτών των δεδομένων, οδηγεί στην απόκτηση περισσότερης γνώσης. Η γνώση αυτή βοηθάει στην καλύτερη κατανόηση του προβλήματος και επομένως στην εφαρμογή ιδανικότερων μέτρων προστασίας από τα ήδη υπάρχοντα. Αυτό συνεπάγεται μείωση του κινδύνου πρόκλησης ζημίας στα ευαίσθητα δεδομένα. Η αξία των honeypots χαρακτηρίζεται από την ικανότητα τους να προκαλέσουν το ενδιαφέρον των κακόβουλων χρηστών ώστε να καταγράψουν τις περισσότερες δυνατές επιθέσεις. Σύμφωνα με τον ορισμό [4]:

*«Ένα honeypot είναι ένας πόρος πληροφοριακών συστημάτων του οποίου η αξία έγκειται στην μη εξουσιοδοτημένη ή παράνομη χρήση αυτού»*

Συνεπώς ο ρόλος των honeypots δεν είναι η προστασία από τις επιθέσεις, αλλά η παροχή περισσότερης γνώσης γύρω από αυτές, με σκοπό την καλύτερη αντιμετώπιση τους. Τα honeypots δεν εκτελούν καμία απολύτως παραγωγική εργασία. Επομένως η οποιαδήποτε δικτυακή κίνηση προς αυτά χαρακτηρίζεται ύποπτη και χρίζεται διερεύνησης.

Τα honeypots ανάλογα με σκοπό που εγκαθίστανται διαχωρίζονται σε δύο κατηγορίες:

1. Honeypots Παραγωγής

**2.** Honeypots Έρευνας

Τα honeypots παραγωγής εγκαθίστανται συνήθως σε εταιρίες ή οργανισμούς, παράλληλα με τα μηχανήματα παραγωγής και έχουν ως σκοπό να διερευνήσουν το είδος των δεδομένων που διακινούνται στο δίκτυο. Σε αυτές τις περιπτώσεις τα honeypots δρουν συμπληρωματικά με τα συστήματα προειδοποίησης επιθέσεων (IDS), επεκτείνοντας τις δυνατότητες τους. Τα honeypots παραγωγής δρουν επίσης και ως μέσα αξιολόγησης για τα ήδη υπάρχοντα μέτρα ασφαλείας (κυρίως firewalls), αξιολογώντας την αποτελεσματικότητα τους. Επιπλέον μπορούν να φανούν αρκετά αποτελεσματικά στην καταγραφή επιθέσεων που δεν μπόρεσαν να εντοπίσουν τα IDS, όπως για παράδειγμα μη διαδεδομένες επιθέσεις ή επιθέσεις που πραγματοποιούνται από υπαλλήλους του εκάστοτε οργανισμού ή εταιρίας, οι οποίοι και αποτελούν χρήστες του δικτύου με αυξημένα δικαιώματα.

Τα honeypots έρευνας από την άλλη πλευρά έχουν ως αποκλειστικό σκοπό τους την παροχή γνώσης γύρω από τις μεθόδους και τις τακτικές που χρησιμοποιούν οι κακόβουλοι χρήστες. Χρησιμοποιούνται ως εργαλεία εκμάθησης των μεθόδων εισβολής των κακόβουλων χρηστών στα συστήματα, αλλά και των κινήσεων που αυτοί πραγματοποιούν μετά από μια επιτυχημένη κατάληψη ενός μηχανήματος.

Τα honeypots δεν διαχωρίζονται μονάχα ως προς τον σκοπό τους αλλά και ως προς το είδος αλληλεπίδρασης το οποίο προσφέρουν. Τα honeypots, σύμφωνα με είδος αλληλεπίδρασης το οποίο προσφέρουν, διαχωρίζονται σε δυο κύριες κατηγορίες :

**1.** Τα honeypots χαμηλής αλληλεπίδρασης (low-interaction honeypots)

**2.** Τα honeypots υψηλής αλληλεπίδρασης (high-interaction honeypots)

Τα honeypots χαμηλής αλληλεπίδρασης, όπως δηλώνει και το όνομα τους, είναι συστήματα τα οποία δεν προσφέρουν υψηλό επίπεδο αλληλεπίδρασης στον κακόβουλο χρήστη. Αυτού του είδους τα honeypots έχουν περιορισμένες δυνατότητες και εξομοιώνουν μονάχα

βασικές λειτουργίες ενός πραγματικού μηχανήματος παραγωγής. Για παράδειγμα εξομοιώνουν την στοίβα του TCP/IP δικτύου ενός συστήματος και κάποιες άλλες απλές υπηρεσίες. Συνεπώς όντας εξομοιωτές μικρής κλίμακας και όχι πραγματικά συστήματα, ο επιτιθέμενος δεν έχει τη δυνατότητα να αναπτύξει μια ολοκληρωμένη επίθεση π.χ. παραβίαση μηχανής. Τα χαμηλής αλληλεπίδρασης honeypots έχουν τα πλεονεκτήματα ότι είναι εύκολα στην εγκατάσταση και τη συντήρηση και δεν απαιτούν πολλούς πόρους συστήματος προκειμένου να λειτουργήσουν. Επιπλέον η χρήση τους είναι ιδιαίτερα απλή, δεν διατρέχουν τον κίνδυνο να παραβιαστούν και παράγουν ένα μικρό σχετικά όγκο δεδομένων για ανάλυση. Αποτελούν συνήθως ιδανική λύση για κάποιο αρχάριο χρήστη ο οποίος θέλει να εισέλθει σε αυτό το χώρο.

Από την άλλη πλευρά τα honeypots χαμηλής αλληλεπίδρασης προσφέρουν ένα περιβάλλον με βασικές μονάχα λειτουργίες. Ένας έμπειρος επιτιθέμενος θα είναι σε θέση αρκετά σύντομα να καταλάβει ότι δεν έχει να αντιμετωπίσει ένα πραγματικό σύστημα. Άμεση συνέπεια αυτού του γεγονότος, θα είναι ο επιτιθέμενος να εγκαταλείψει την προσπάθεια αμέσως. Επίσης ο μικρός όγκος δεδομένων τον οποίο παράγουν τα χαμηλής αλληλεπίδρασης honeypots σημαίνει την καταγραφή λιγότερων πληροφοριών σχετικά με τις διενεργημένες επιθέσεις και συνεπώς έχουν μικρότερη εκπαιδευτική αξία. Παραδείγματα honeypots χαμηλής αλληλεπίδρασης είναι τα La Brera, το Nepenthess, το Honeyd κ.α.

Τα honeypots υψηλής αλληλεπίδρασης, προσφέρουν ένα περιβάλλον υψηλής αλληλεπίδρασης στον κακόβουλο χρήστη. Δεν αποτελούν εξομοιωτές, αλλά πρόκειται για πραγματικά συστήματα με πραγματικές υπηρεσίες που έχουν ως σκοπό να ξεγελάσουν πλήρως τον επιτιθέμενο χρήστη. Πολλές φορές αυτά τα συστήματα αποτελούν αντίγραφα πραγματικών παραγωγικών μηχανών, τα οποία έχουν ως σκοπό να εξετάσουν τα κενά ασφαλείας που υπάρχουν σε αυτά. Το ότι τα honeypots υψηλής αλληλεπίδρασης αποτελούν αντίγραφα πραγματικών συστημάτων συνεπάγεται ότι ο εντοπισμός τους από τον έμπειρο επιτιθέμενο απαιτεί περισσότερο χρόνο. Κατά αυτό τον τρόπο καταγραφούν ένα μεγάλο αριθμό επιθέσεων, παράγοντας ένα μεγάλο όγκο δεδομένων και συνεπώς ένα αυξημένο επίπεδο γνώσης για τους διαχειριστές τους.

Από την άλλη πλευρά τα honeypots υψηλής αλληλεπίδρασης είναι πιο δύσκολα στη

χρήση και ανάπτυξη τους, καθώς απαιτούν τη ρύθμιση πολλών παραμέτρων, ενώ χρειάζονται και περισσότερους πόρους συστήματος προκειμένου να λειτουργήσουν. Η ανάλυση του μεγάλου όγκου δεδομένων τον οποίο παράγουν είναι δύσκολη και απαιτεί πολύωρη εργασία. Επίσης επειδή δύναται να παραβιαστούν, ο επιτιθέμενος μπορεί να τα χρησιμοποιήσει ως πλατφόρμες επίθεσης σε άλλα συστήματα. Λόγω του ότι η χρήση honeypots έστω και για εκπαιδευτικούς σκοπούς, δεν μπορεί να αποτελέσει άλλοθι για κακουργηματικές πράξεις, θα πρέπει πάντα να υπάρχουν οι κατάλληλες δικλείδες ασφαλείας. Σε περίπτωση που αυτά τα συστήματα παραβιαστούν και χρησιμοποιηθούν για παράνομες δραστηριότητες, οι συνέπειες θα επιβαρύνουν τους διαχειριστές και όχι μόνο τους παραβάτες.

**Honeynets**

Τα honeynets αποτελούν ουσιαστικά ένα σύνολο από υψηλής αλληλεπίδρασης honeypots. Αυτά τα honeypots είναι δυνατό να υπάρχουν είτε ως φυσικά μηχανήματα είτε ως εικονικά. Συνήθως ένα honeynet προσπαθεί να μιμηθεί τη λειτουργία ενός πραγματικού δικτύου ή δικτυακού τόπου. Με αυτόν τον τρόπο προσφέρεται στον εκάστοτε επιτιθέμενο χρήστη ένα ολοκληρωμένο περιβάλλον της υψηλότερης αλληλεπίδρασης που θα μπορούσε να επιτευχθεί. Επίσης δημιουργούνται και πιο «ρεαλιστικοί» στόχοι, καθώς πλέον μεμονωμένοι υπολογιστές είναι πολύ δύσκολο να προσελκύσουν το ενδιαφέρον κάποιου κακόβουλου χρήστη. Έτσι λοιπόν είμαστε σε θέση να μελετήσουμε τους τρόπους με τους οποίους επιτίθενται οι κακόβουλοι χρήστες σε «πραγματικά» δίκτυα. Επιπροσθέτως είμαστε σε θέση να ανακαλύψουμε νέους τρόπους επιθέσεων, όπως και αδυναμίες των διάφορων υπηρεσιών και προγραμμάτων, που είναι εγκατεστημένα σε αυτούς τους υπολογιστές, και δεν τις γνωρίζαμε μέχρι τότε.

# ΚΕΦΑΛΑΙΟ 2 – ΠΡΟΓΡΑΜΜΑΤΑ ΚΑΙ ΕΡΓΑΛΕΙΑ

Σε αυτό το κεφάλαιο θα περιγραφούν και θα αναλυθούν τα διάφορα προγράμματα αλλά και τα εργαλεία που χρησιμοποιήθηκαν για την πραγματοποίηση του μετέπειτα πειράματος.

## HONEYD

### Με λίγα λόγια

Το Honeyd είναι ένα αρκετά ευέλικτο πρόγραμμα - εργαλείο για τη δημιουργία εικονικών χαμηλής αλληλεπίδρασης honeypots. Είναι σε θέση να χρησιμοποιήσει χιλιάδες μη δεσμευμένες διευθύνσεις IP, αντιστοιχίζοντας τες σε εικονικά honeypots, τα οποία ανταποκρίνονται στην εισερχόμενη κίνηση την οποία δέχονται. Μπορεί επίσης να εξομοιώσει ταυτόχρονα ένα μεγάλο αριθμό από λειτουργικά συστήματα, συσκευές και υπηρεσίες (όπως telnet, ftp, smtp κ.α.) μέσα από έτοιμες απαντήσεις και scripts[1]. Δημιουργήθηκε και συντηρείται από τον Niels Provos. Η πρώτη του έκδοση κυκλοφόρησε το 1992 ενώ τη στιγμή που γραφόταν αυτή εδώ η εργασία το Honeyd είχε φθάσει στην έκδοση 1.5c.

### Αναλυτικά

Με το Honeyd έχουμε τη δυνατότητα να δημιουργήσουμε πολλά εικονικά honeypots, τα οποία θα έχουν διαφορετική παραμετροποίηση και θα εξομοιώνουν διαφορετικά συστήματα. Τα εικονικά αυτά honeypots θα προσφέρουν στον επιτιθέμενο χρήστη ένα περιβάλλον με βασικές μονάχα υπηρεσίες, όπως και τα περισσότερα χαμηλής αλληλεπίδρασης honeypots. Το Honeyd μπορεί μονάχα να εξομοιώσει υπηρεσίες που βασίζονται στα πρωτόκολλα TCP και UDP, όμως, παράλληλα έχει τη δυνατότητα να αντιλαμβάνεται και να απαντάει σωστά και σε μηνύματα του πρωτοκόλλου ICMP. Για να εξομοιώσει τις διάφορες αυτές υπηρεσίες το Honeyd χρησιμοποιεί διάφορα scripts τα οποία ενεργοποιούνται όταν ανιχνευθεί κίνηση προς αντίστοιχες πόρτες που έχουμε ορίσει.

1 Βλέπε γλωσσάρι

Για να ρυθμίσουμε το Honeyd πραγματοποιούμε 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 πάντα με το συντακτικό και τις εντολές του honeyd όπως αυτές θα περιγραφούν παρακάτω. Έχουμε την 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.). Επίσης, στα honeypots που δημιουργούμε καθορίζουμε τις ανοικτές πόρτες που επιθυμούμε να μελετήσουμε και την εν γένη συμπεριφορά τους. Στην αρχή εκτέλεσης του Honeyd δηλώνουμε πάντα το αρχείο παραμετροποίησης που θα χρησιμοποιήσει.

Μία πολύ σημαντική δυνατότητα που έχει το Honeyd είναι να εξαπατά προγράμματα που βασίζονται στο signature scanning[1] όπως είναι το Nmap και το Xprobe, στέλνοντας κατάλληλες απαντήσεις. Το Honeyd διαβάζει ένα ειδικό αρχείο με signatures το οποίο καθορίζει την τροποποίηση της TCP/IP στοίβας ανά honeypot και συνεπώς της προσωπικότητας. Ο χρήστης έχει τη δυνατότητα να επιλέξει αρχείο signatures υπό την μορφή που ορίζει το NMAP, είτε υπό την μορφή που ορίζει το Xprobe, είτε τον συνδυασμό και των δύο ταυτοχρόνως. Η επιλογή γίνεται κατά την εκτέλεση της εντολής εκκίνησης του Honeyd. Όταν λοιπόν το Honeyd καταλάβει πως κάποιο σάρωση είναι σε εξέλιξη, θα αποκριθεί σύμφωνα με τα signatures που του ορίσαμε και ανάλογα με την προσωπικότητα που έχουμε προσδώσει στο κάθε honeypot. Στην άλλη άκρη, 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. Είναι σε θέση μάλιστα να διαχειριστεί ταυτόχρονα αρκετές χιλιάδες από αυτές. Σε δοκιμές που έγιναν πειραματικά το Honeyd κατάφερε να διαχειριστεί ταυτόχρονα πάνω από 60.000 διευθύνσεις IP στο φιλοξενούμενο σύστημα [5]. Είναι αντιληπτό πως μπορεί να αντεπεξέλθει επιτυχώς ακόμα και σε δίκτυα που δέχονται αρκετά μεγάλη κίνηση.

Πριν δεχθούν οποιαδήποτε δικτυακή κίνηση τα εικονικά honeypots που θα

1 Βλέπε γλωσσάρι

δημιουργήσουμε, θα χρειαστεί να «τρέξουμε» στον ξενιστή, παράλληλα με το Honeyd, το Fake ARP. Το Fake ARP είναι ένα πρόγραμμα το οποίο λειτουργεί συμπληρωματικά στον ARP δαίμονα του συστήματος.

Το ARP (Address Resolution Protocol) είναι ένα πρωτόκολλο το οποίο χρησιμοποιείται για τον καθορισμό της διεύθυνσης MAC (διεύθυνση υλισμικού) ενός υπολογιστή όταν είναι γνωστή μονάχα η IP διεύθυνση του. Για να επιτύχει την ανεύρεση της διεύθυνσης MAC το ARP πραγματοποιεί broadcast εκπομπές ρωτώντας ποια διεύθυνση MAC αντιστοιχεί στην υπό διερεύνηση IP. Εν συνεχεία, το ARP πρωτόκολλο της μηχανής στην οποίο αντιστοιχεί η διεύθυνση IP αποστέλλει την απάντηση πίσω στον ARP δαίμονα του αρχικού μηχανήματος. Ο συσχετισμός (διεύθυνσης IP και διεύθυνσης MAC) είναι απαραίτητος γιατί κατά τον σχηματισμό του Ethernet πλαισίου (frame) – διαδικασία κατά την οποία ενθυλακώνετε το IP πακέτο μέσα σε Ethernet header και footer- το λογισμικό της TCP/IP στοίβας δεν έχει άλλο τρόπο ώστε να εξάγει την διεύθυνση MAC. Το πρωτόκολλο ARP καλύπτει το συγκεκριμένο κενό επιτρέποντας την επικοινωνία μεταξύ των υπολογιστών ενός τοπικού δικτύου. Για περισσότερε πληροφορίες σχετικά με το ARP και τη λειτουργία του βλέπε [6].

Το Fake ARP εξετάζει όλες τις εισερχόμενες ερωτήσεις του ARP πρωτοκόλλου, για τις διευθύνσεις IP που βρίσκονται μέσα στα όρια του υποδικτύου που ορίσαμε. Αν ο ARP δαίμονας του συστήματος δεν απαντήσει σε κάποια από αυτές, που θα σημαίνει πως δεν υπάρχει κάποιο σύστημα που να χρησιμοποιεί αυτή τη διεύθυνση IP, αναλαμβάνει το Fake ARP να απαντήσει στη θέση του. Με λίγα λόγια αναλαμβάνει να συσχετίσει τις αδέσμευτες διευθύνσεις IP του δικτύου με τα εικονικά συστήματα του Honeyd στα οποία δεν έχει ρητά αποδοθεί άλλη IP.

## Βασικές εντολές στο Honeyd

### 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 προσδώσουμε κάποια προσωπικότητα. Σύμφωνα με το συντακτικό του

Honeyd μετά την εντολή create διακρίνουμε τρεις περιπτώσεις:

- Στην πρώτη περίπτωση δημιουργούμε απλώς ένα template προσθέτοντας μετά την εντολή create το όνομα της επιλογής μας π.χ create linux.

- Στην δεύτερη περίπτωση δημιουργούμε το default template. Το honeyd με αυτή την επιλογή αντιστοιχεί οποιαδήποτε ελεύθερη διεύθυνση IP (μη συσχετισμένη με κάποιο άλλο εικονικό honeypot) του υποδικτύου που του έχουμε ορίσει με το default template. Η λέξη default είναι δεσμευμένη από το honeyd και δεν μπορεί να χρησιμοποιηθεί για κανένα άλλο λόγο πλην αυτού που προαναφέρθηκε π.χ. create default.

- Στην τρίτη περίπτωση δημιουργούμε ένα δυναμικό template. Για παράδειγμα ένα template που θα είναι ορατό μόνο συγκεκριμένες ώρες τις ημέρας ή που θα είναι αόρατο σε κίνηση που προέρχεται από κάποιες διευθύνσεις IP. π.χ create dynamic linux.

**set**

Με την εντολή set μπορούμε να παραμετροποιήσουμε το template το οποίο δημιουργήσαμε με την εντολή create. Πιο συγκεκριμένα, συσχετίζουμε το template με κάποια προσωπικότητα από το αρχείο που περιέχει τα Nmap signatures. 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. Για να καθορίσουμε την συμπεριφορά τους προσθέτουμε μετά το set μία από τις παρακάτω επιλογές ως εξής:

- Προσθέτοντας “Open” καθορίζουμε πως όλες οι πόρτες θα είναι ανοικτές εξ ορισμού. Αυτή η εντολή επηρεάζει μόνο τα πρωτόκολλα TCP και UDP.

- Προσθέτοντας “Block” καθορίζουμε πως όλα τα πακέτα του αντίστοιχου πρωτοκόλλου θα γίνονται drop εξ ορισμού. Θα λαμβάνονται δηλαδή, αλλά δεν θα στέλνεται καμία

απάντηση πίσω στον αποστολέα. Αυτή η επιλογή θα μπορούσε να χρησιμοποιηθεί για να εξομοιώσει τη λειτουργία ενός firewall.

- Προσθέτοντας "Reset" καθορίζουμε 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.

Με την εντολή set μπορούμε επίσης να ορίσουμε και τον εικονικό χρόνο τον οποίο το εκάστοτε honeypot βρίσκεται σε λειτουργία.

**add**

Η εντολή add παραμετροποιεί τα εικονικά honeypots καθορίζοντας ποιες πόρτες θα μείνουν ανοικτές και προσθέτοντας υπηρεσίες μέσω κάποιον scripts. Με την εντολή add έχουμε τρεις δυνατότητες:

- 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

- Τέλος 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**

Η εντολή bind είναι η τελευταία κατά σειρά που χρησιμοποιείται για την ρύθμιση ενός εικονικού honeypot. Με την εντολή bind συσχετίζουμε τα εικονικά honeypot τα οποία δημιουργήσαμε με κάποια διεύθυνση IP. Αν θέλουμε μπορούμε να συσχετίσουμε ένα honeypot

με παραπάνω από μια διευθύνσεις.

**tarpit**

Η εντολή tarpit χρησιμοποιείται για να επιβραδύνει όλες τις συνδέσεις στην πόρτα ή στις πόρτες που έχουμε εμείς καθορίσει. Αυτό επιτυγχάνεται ορίζοντας στα TCP options των πακέτων μηδενικό TCP window αποτέλεσμα η επικοινωνία να προχωράει πολύ αργά. Αυτή η επιλογή έχει ως στόχο να περιορίσει την γρήγορη εξάπλωση των ιών, καθώς το Honeyd κρατάει απασχολημένο το «μολυσμένο» μηχάνημα με αποτέλεσμα να μην μπορεί να συνδεθεί με άλλα μηχανήματα του δικτύου και να εξαπλώσει τον ιό. Επίσης να αποθαρρύνονται πιθανοί κακόβουλοι χρήστες, οι οποίοι θα αναγκάζονται να παραμείνουν για πολύ ώρα σε ανοικτές συνδέσεις χωρίς ουσιαστικό αποτέλεσμα.

## HONEYWALL

Το Honeywall είναι μια διανομή GNU/Linux η οποία έχει προ εγκατεστημένα πληθώρα εργαλείων και έχει αναπτυχθεί από την Honeynet Research Alliance. Χρησιμοποιείται παράλληλα με τα honeypots που έχουμε εγκαταστήσει και σκοπός του είναι να μας βοηθήσει να ελέγξουμε, να παρακολουθήσουμε και να αναλύσουμε όλη την κίνηση από και προς αυτά. Το Honeywall συμπεριλαμβάνει μια πληθώρα εργαλείων που μας βοηθούν να έχουμε καλύτερη εποπτεία των honeypots. Μια σύντομη περιγραφή μερικών από των εργαλείων που περιλαμβάνονται στο Honeywall γίνεται παρακάτω.

**Iptables**

Το Iptables είναι το firewall σύμφωνα με το οποίο ενεργεί το Honeywall. Το Iptables είναι στην πραγματικότητα ένα εργαλείο διαχείρισης σε περιβάλλον χρήστη (user space) για το netfilter το οποίο ανήκει στην οικογένεια των firewalls λογισμικού που λειτουργούν σε επίπεδο IP (IP filter). Βρίσκεται ενσωματωμένο, υπό την μορφή διαφόρων modules, στον πυρήνα του Linux. Το netfilter ορίζει λίστες με κανόνες σύμφωνα με τις οποίες πραγματοποιεί έλεγχο της κίνησης. Το Iptables είναι μια εφαρμογή που εκτελείτε μέσω γραμμής εντολών μέσω της οποίας ρυθμίζουμε και τροποποιούμε τις λίστες αυτές. Με το Iptables τροποποιούμε βασικά τρεις

λίστες στις οποίες δημιουργούμε κανόνες ανάλογα με την πολιτική που έχουμε σχεδιάσει. Οι λίστες που τροποποιούμε είναι η INPUT, η λίστα που ελέγχει την είσοδο των πακέτων στο σύστημα, η FORWARD, η λίστα που ελέγχει τα πακέτα τα οποία δρομολογούνται μεταξύ διεπαφών και η OUTPUT, η λίστα η οποία ελέγχει τα πακέτα τα οποία εξέρχονται από το σύστημα. Με την βοήθεια του συνδυασμού Iptables-Netfilter μπορούμε να δημιουργήσουμε αρκετά δυνατά και ευέλικτα firewalls.

**Snort**

Το Snort είναι μια εφαρμογή που ανήκει στην οικογένεια των IDS (Intrusion Detection Systems – Συστήματα Ανίχνευσης Επιθέσεων). Είναι λογισμικό ανοικτού κώδικα και έχει τη δυνατότητα να λειτουργήσει σε αρκετά λειτουργικά συστήματα μέσα από τις διαφορετικές εκδόσεις του. Ο ρόλος του είναι να εξετάζει την εισερχόμενη κίνηση στο σύστημα που είναι εγκατεστημένο και να δημιουργεί προειδοποιήσεις (alerts) σε περίπτωση που εντοπίσει κάποια επίθεση. Για να εντοπίσει ενδεχόμενες επιθέσεις το Snort βασίζεται σε ένα αρχείο το οποίο περιέχει signatures από γνωστές επιθέσεις. Το Snort εξετάζει την εισερχόμενη κίνηση και συγκρίνει τα πακέτα που λαμβάνει με τα signatures αυτά και αν εντοπίσει κάποια συσχέτιση παράγει προειδοποιήσεις. Δυστυχώς υπάρχει πάντα η πιθανότητα το Snort να παράγει και λανθασμένες προειδοποιήσεις (false positives) λόγω ομοιότητας νόμιμης κίνησης με signatures.

**Snort Inline**

Το Snort inline λειτουργεί κατά διαφορετικό τρόπο από το Snort και ανήκει στην οικογένεια των IPS (Intrusion Prevention Systems). Η διαφορά του από το Snort είναι ότι το Snort inline εξετάζει την εξερχόμενη κίνηση (και όχι την εισερχόμενη όπως το πρώτο) με σκοπό να εντοπίσει και να σταματήσει ενδεχόμενες επιθέσεις που πραγματοποιούνται από τα δικά μας, πιθανώς «μολυσμένα» honeypots, προς άλλα συστήματα του Διαδικτύου. Για να το επιτύχει αυτό το Snort inline χρησιμοποιεί το ίδιο αρχείο με signatures από γνωστές επιθέσεις που χρησιμοποιεί και το Snort. Όταν λειτουργεί το Snort inline δέχεται πακέτα από το Iptables χρησιμοποιώντας την libipq βιβλιοθήκη (αντί για την libpcap που χρησιμοποιεί το snort) και εν συνεχεία εξετάζει αυτά τα πακέτα συγκρίνοντας τα με τα γνωστά signatures που έχει αποθηκευμένα. Αν εντοπίσει κάποια συσχέτιση παράγει προειδοποιήσεις και ειδοποιεί το

Iptables να κάνει drop, sdrop ή reject[1] αυτά τα πακέτα ανάλογα με τους κανόνες που του έχουμε ορίσει [11]. Κατά αυτόν τον τρόπο αποφεύγονται ενδεχόμενες επιθέσεις πριν αρχίσουν την κακόβουλη δράση τους. Φυσικά, όπως και στο snort, υπάρχει πάντα το ενδεχόμενο να λάβουμε λάθος προειδοποιήσεις.

**Sebek**

Το Sebek είναι ένα module εγκατεστημένο στον πυρήνα των υψηλής αλληλεπίδρασης honeypots. Σκοπός του είναι η συλλογή των πληκτρολογηθέντων χαρακτήρων (keystrokes) του συστήματος και η αποστολή τους και συλλογή τους σε απομακρυσμένο σύστημα. Σε μια ενδεχόμενη παραβίαση του συστήματος, θα μπορέσουμε να έχουμε όλες τις εντολές που πληκτρολόγησε ο κακόβουλος χρήστης, αφού εισήλθε σε αυτό. Επίσης, με αυτό τον τρόπο αντιμετωπίζουμε το πρόβλημα των κρυπτογραφημένων συνδέσεων. Οι περισσότεροι κακόβουλοι χρήστες πλέον, χρησιμοποιούν κρυπτογραφημένες συνδέσεις (SSH) για την επικοινωνία με το παραβιασμένο σύστημα. Ακόμα και αν καταγράψουμε ολόκληρη την κίνηση δεν θα είμαστε σε θέση μετά να την «διαβάσουμε». Με το Sebek αντίθετα καταγράφουμε ο,τι πληκτρολόγησε ο επιτιθέμενος βγάζοντας χρήσιμα συμπεράσματα. Ο λόγος που τα δεδομένα δεν αποθηκεύονται τοπικά είναι για να μην γίνουν αντιληπτά από τον κακόβουλο χρήστη και προσπαθήσει να τα σβήσει. Επίσης το Sebek προκειμένου να μην γίνει αντιληπτό φροντίζει να κρύβει τα πακέτα που στέλνει έτσι ώστε αυτά να μην εντοπίζονται ούτε από κάποιο πρόγραμμα καταγραφής δικτυακής κίνησης (sniffers).

**P0f**

Το P0f είναι ένα πρόγραμμα για παθητική ανίχνευση λειτουργικών συστημάτων. Ανήκει δηλαδή στην οικογένεια των παθητικών ανιχνευτών (passive scanners). Το P0f έχει τη δυνατότητα να εντοπίσει το λειτουργικό σύστημα που έχει εγκατεστημένο ο απομακρυσμένος υπολογιστής,, αν στην επικοινωνία παρεμβάλλεται κάποιο firewall, την αριθμό των ενδιάμεσων σταθμών (hops), το χρονικό διάστημα που αυτός είναι εν λειτουργία (uptime) καθώς και τον τρόπο της διασύνδεσης τους με το Διαδίκτυο. Για να το επιτύχει αυτό το P0f εξετάζει τα μηνύματα που προέρχονται από τον απομακρυσμένο υπολογιστή. Συγκεκριμένα το P0f πραγματοποιεί προβλέψεις σύμφωνα με τέσσερις τρόπους:

1 Βλέπε γλωσσάρι

1. Εξετάζοντας τα εισερχόμενα μηνύματα σύνδεσης (SYN flag ενεργό), που αποστέλλει ο απομακρυσμένος υπολογιστής.

2. Εξετάζοντας τα μηνύματα αποδοχής σύνδεσης (SYN+ACK flags ενεργά), που αποστέλλει ο απομακρυσμένος υπολογιστής.

3. Εξετάζοντας τα μηνύματα άρνησης σύνδεσης (RST flag ενεργό), που αποστέλλει ο απομακρυσμένος υπολογιστής.

4. Εξετάζοντας τα μηνύματα επιβεβαίωσης σύνδεσης (ACK flag ενεργό), που αποστέλλει ο απομακρυσμένος υπολογιστής.

Το P0f, όπως και οι υπόλοιποι παθητικοί ανιχνευτές, παραμένει αόρατο στον απομακρυσμένο υπολογιστή.

**Swatch**

Το Swatch σκοπός του είναι να ειδοποιεί εγκαίρως τους διαχειριστές του honeynet για ενδεχόμενες επιθέσεις ή παραβάσεις που πραγματοποιούν τα υπό παρακολούθηση honeypots. Το swatch παρακολουθεί σε φακέλους που του έχουμε ορίσει τα αρχεία καταγραφής συμβάντων των υπηρεσιών και αν εντοπίσει σε αυτά οποιαδήποτε μη φυσιολογική δραστηριότητα αποστέλλει ενημερωτικό ηλεκτρονικό μήνυμα. Το Swatch θα μας στείλει ένα e-mail για παράδειγμα, όταν θα εντοπίσει εξερχόμενη κίνηση από κάποιο από τα honeypots μας. Η χρησιμότητα του είναι πολύ μεγάλη, καθώς μας βοηθάει να έχουμε εποπτεία των honeypots χωρίς να χρειάζεται να τα παρακολουθούμε συνεχώς.

**Walleye**

Το Walleye είναι ένα δικτυακό γραφικό περιβάλλον (web interface) το οποίο προσφέρεται από το Honeywall ώστε να είναι ευκολότερη η εποπτεία του. Το Walleye βασίζεται στην διεργασία, hflow, η οποία συγκεντρώνει τις πληροφορίες από τα αρχεία καταγραφής συμβάντων των διάφορων υπηρεσιών και μας δίνει την δυνατότητα να τις προσπελάσουμε εύκολα μέσω του δικτυακού γραφικού περιβάλλοντος. Συγκεκριμένα το Walleye συγκεντρώνει τα μηνύματα

συμβάντων από όλα τα παραπάνω εργαλεία και μας δίνει τη δυνατότητα συσχετισμού μεταξύ τους. Κατά αυτόν τον τρόπο δεν χρειάζεται πλέον ο χρήστης να εκτελεί αυτήν τη διαδικασία χειροκίνητα, αλλά εύκολα μέσα από το γραφικό περιβάλλον. Επίσης δίνεται η δυνατότητα αλλαγής παραμετροποίησης της συσκευής διευκολύνοντας σημαντικά την απομακρυσμένη διαχείριση της..

## Διάφορα Προγράμματα και Εργαλεία

### Tcpdump

Το Tcpdump είναι ένα πρόγραμμα καταγραφής και ανάλυσης της δικτυακής κίνησης (packet sniffer/analyzer). Εκτελείται μέσα από γραμμή εντολών και δίνει τη δυνατότητα στο χρήστη να συλλάβει, να προβάλει και να αποθηκεύσει όλα τα πακέτα, ανεξαρτήτως πρωτοκόλλου, που λαμβάνονται και αποστέλλονται σε ένα δίκτυο στο οποίο ο υπολογιστής είναι συνδεδεμένος. Το Tcpdump δημιουργήθηκε από τους Van Jacobson, Craig Leres και Steven McCanne και κυκλοφόρησε το 1987.

### Wireshark

Το Wireshark είναι επίσης ένα πρόγραμμα καταγραφής και ανάλυσης της δικτυακής κίνησης. Περιλαμβάνει ένα γραφικό περιβάλλον το οποίο βασίζεται στο GTK+ και έχει τη δυνατότητα, όπως και το Tcpdump, να συλλάβει, να προβάλει και να αποθηκεύσει όλα τα πακέτα, ανεξαρτήτως πρωτοκόλλου, που λαμβάνονται και αποστέλλονται σε ένα δίκτυο στο οποίο ο υπολογιστής είναι συνδεδεμένος. Ανεξάρτητα από την δυνατότητα του να καταγράφει την δικτυακή κίνηση, το Wireshark χρησιμοποιείται κυρίως ως εργαλείο ανάλυσης πακέτων λόγω του εύχρηστου γραφικού περιβάλλοντος και των εργαλείων που αυτό περιλαμβάνει. Το Wireshark, ως πρόγραμμα, ξεκίνησε και έγινε γνωστό με την ονομασία Ethereal μέχρι το 2006 χρονολογία στην οποία μετονομάστηκε ως έχει σήμερα. Έχει γραφτεί από τον Gerald Combs και η πρώτη έκδοση του κυκλοφόρησε το 1998.

**Tshark**

Το Tshark είναι ένα ακόμα πρόγραμμα καταγραφής και ανάλυσης της δικτυακής κίνησης. Το Tshark αποτελεί στην ουσία την έκδοση του Wireshark που εκτελείται σε γραμμή εντολών. Γι'αυτό το λόγο το Tshark περιλαμβάνει μια πληθώρα εντολών για την ανάλυση της καταγεγραμμένης δικτυακής κίνησης. Είναι κατάλληλο για την ανάλυση αρχείων μεγάλου όγκου, τα οποία δεν μπορούν εύκολα να αναλυθούν με τη χρήση του Wireshark λόγω των αυξημένων απαιτήσεων σε μνήμη που έχει το γραφικό περιβάλλον. Το Tshark είναι ένα πρόγραμμα που εγκαθίσταται μαζί με το Wireshark.

# ΚΕΦΑΛΑΙΟ 3 - ΠΕΙΡΑΜΑ HONEYD

Στο Εργαστήριο Δικτύων του ΕΚΕΦΕ Δημόκριτος εκτελέσαμε πείραμα με το Honeyd προκειμένου να διαπιστώσουμε τις δυνατότητές του, αλλά κυρίως να προσπαθήσουμε να καταγράψουμε δικτυακές επιθέσεις. Για το σκοπό αυτό δημιουργήσαμε εικονικά honeypots μέσω του Honeyd τα οποία στη συνέχεια παραμετροποιήσαμε κατά τέτοιο τρόπο, ώστε να εξομοιώσουν όσο το δυνατό καλύτερα πραγματικά συστήματα. Για την πραγματοποίηση του πειράματος χρησιμοποιήθηκαν τρεις φυσικές μηχανές:

1. Η πρώτη ήταν η μηχανή στην οποία εγκαταστάθηκε το Honeyd, όπου με τη βοήθεια αυτού δημιουργήθηκαν τα εικονικά honeypots.

2. Το δεύτερο ήταν ένας DNS διακομιστής, που είχε το ρόλο ενός DNS honeypot.

3. Το τρίτο ήταν το σύστημα στο οποίο εγκαταστάθηκε το Honeywall και από το οποίο διερχόταν όλη κίνηση από και προς τα honeypots.

Προκειμένου να εγκαταστήσουμε το Honeyd επιλέξαμε πρώτα λειτουργικό σύστημα. Επιλέχθηκε προς εγκατάσταση το Ubuntu Server 8.04, το οποίο είναι ένα λειτουργικό σύστημα ειδικά σχεδιασμένο για χρήση σε διακομιστές, χωρίς γραφικό περιβάλλον και με την κύρια εγκατάσταση να είναι αρκετά «φτωχή» με ελάχιστες υπηρεσίες. Ο χρήστης μετά την εγκατάσταση έχει τη δυνατότητα να προσθέσει λογισμικό/υπηρεσίες ανάλογα με τις ανάγκες του. Αυτό γίνεται για λόγους ασφαλείας καθώς πρέπει οι διακομιστές να είναι αρκετά προστατευμένοι απέναντι σε επιθέσεις από κακόβουλους χρήστες. Στο σύστημα μας προστέθηκε μόνο η SSH υπηρεσία μετά την εγκατάσταση, με σκοπό την απομακρυσμένη διαχείριση. Εν συνεχεία εγκαταστήσαμε την έκδοση 1.5c του Honeyd μέσα από τις αποθήκες λογισμικού του Ubuntu. Είναι η πιο πρόσφατη έκδοση του Honeyd που κυκλοφορεί αυτή τη στιγμή με ημερομηνία έκδοσης 27-05-2007.

Στην επόμενη μηχανή εγκαταστήσαμε το Honeywall το οποίο βασίζεται στο λειτουργικό

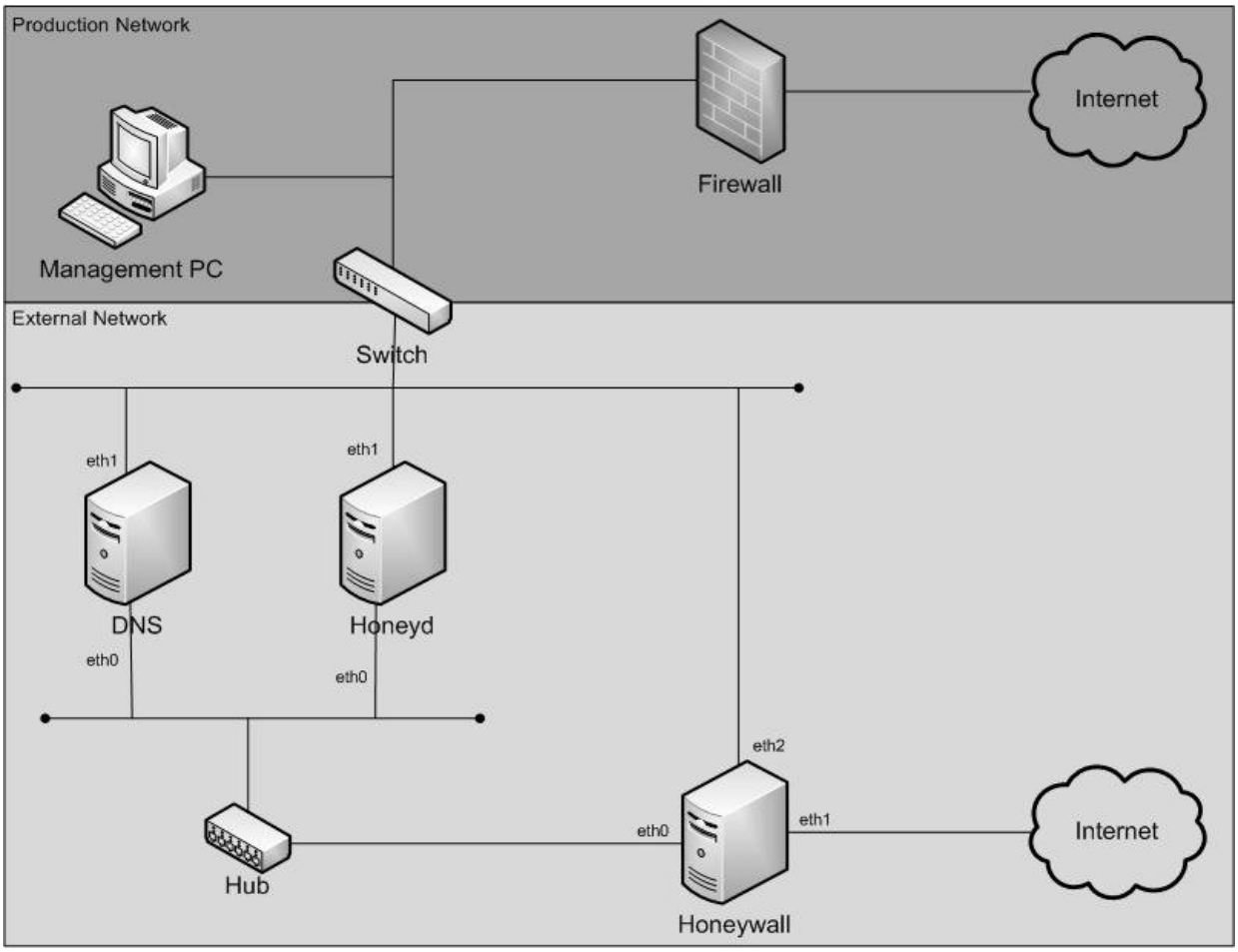
CentOS. Η πιο πρόσφατη έκδοση του Honeywall είναι η roo 1.4 με ημερομηνία έκδοσης 25-04-2009. Λόγω προβλημάτων, όμως, που υπήρχαν στη συγκέντρωση των δεδομένων και που δεν καταφέραμε να επιλύσουμε, αναγκαστήκαμε να εγκαταστήσουμε την beta έκδοση roo 1.3. Η διαδικασία είναι αρκετά απλή, αρκεί να φορτώσουμε το Live CD στον οδηγό και να ξεκινήσουμε την διαδικασία εγκατάστασης. Μετά την εγκατάστασή ακολουθούν κάποιες ερωτήσεις από το σύστημα με σκοπό την ρύθμιση του.

Ο DNS διακομιστής ο οποίος προϋπήρχε ήταν εγκατεστημένος στο λειτουργικό σύστημα Red Hat έκδοσης 9 (Shrike). Ο διακομιστής αυτός προϋπήρχε από την εποχή που το εργαστήριο συμμετείχε ως ενεργό μέλος του Honeynet Project.

Στα παραπάνω συστήματα δόθηκαν τα εξής ονόματα :

- Στο μηχάνημα που ήταν εγκατεστημένο το Honeyd δόθηκε το όνομα **Ikaria**
- Στο μηχάνημα που ήταν εγκατεστημένο το Honeywall δόθηκε το όνομα **Samos**
- Στο μηχάνημα που ήταν εγκατεστημένος ο DNS server υπήρχε το όνομα **Bilem**

Στο σχήμα 3.1 φαίνεται η τοπολογία του συστήματος.




Σχήμα 3.1: Τοπολογία πειράματος Honeyd

Όπως φαίνεται και από το σχήμα όλα τα φυσικά συστήματα έχουν δύο διεπαφές πλην του Samos το οποίο έχει τρεις. Η μία από τις δύο διεπαφές χρησιμοποιείτε για την διασύνδεση στο Διαδίκτυο, ενώ η δεύτερη ως διεπαφή διαχείρισης, δηλαδή για τη διαχείριση της μηχανής από απόσταση. Το Samos έχει τρεις διεπαφές. Οι δύο από αυτές χρησιμοποιούνται από το Honeywall ενώ η τρίτη έχει χρησιμοποιείται και εδώ για την διαχείριση της μηχανής από απόσταση. Πιο συγκεκριμένα η μια διεπαφή που χρησιμοποιεί το Honeywall ορίζεται ως διεπαφή εισόδου και η δεύτερη ως διεπαφή εξόδου. Η διεπαφή εισόδου συνδέεται με το Ikaria και η διεπαφή εξόδου με το διαδίκτυο. Οι δύο διεπαφές (διεπαφή εισόδου και διεπαφή εξόδου) γεφυρώνονται στο στρώμα 2 κατά OSI (bridging).

Η διασύνδεση με το Διαδίκτυο γίνεται απευθείας χωρίς την παρουσία κάποιου firewall ή κάποιων άλλων αντιμέτρων ασφαλείας. Αυτό γίνεται καθώς θέλουμε το honeynet να είναι

ανοικτό σε επιθέσεις και να μην τις περιορίζει, για να μπορέσουμε να καταγράψουμε όσο το δυνατόν περισσότερες πιθανές προσπάθειες. Αντίθετα, όλες οι διεπαφές διαχείρισης συνδέονται στο δίκτυο του Εργαστηρίου και προστατεύονται από το firewall αυτού. Το εργαστήριο πρέπει και αυτό με τη σειρά του να προστατεύεται από την εισερχόμενη κίνηση από τις διεπαφές για να αποφευχθεί η περίπτωση κάποιος, να αποκτήσει πρόσβαση στο εσωτερικό δίκτυο του Εργαστηρίου, αφού έχει επιτυχώς διεισδύσει σε ένα από τα Samos, Ikaria ή Bilem. Γι'αυτό το λόγω κρίθηκε απαραίτητη η ενεργοποίηση Iptables firewall που να επιτρέπει SSH συνδέσεις μονάχα από τις μηχανές διαχείρισης ενώ όλες οι υπόλοιπες αιτήσεις για σύνδεση απορρίπτονταν.

Για την ρύθμιση του Honeyd πραγματοποιούμε τις απαραίτητες αλλαγές στο αρχείο "honeyd.conf". Το Honeyd διαβάζει το αρχείο αυτό κατά την εκκίνηση του και δημιουργεί και ρυθμίζει κατάλληλα τα honeypots σύμφωνα με αυτά που έχουμε ορίσει. Παρακάτω στο σχήμα 3.2 βλέπουμε ένα απόσπασμα από το αρχείο παραμετροποίησης του Honeyd το "honeyd.conf".

honeyd.conf

```
create adsl_router
set adsl_router personality "DLink DI-604 ethernet router"
set adsl_router default tcp action reset
set adsl_router default udp action reset
set adsl_router default icmp action open
add adsl_router tcp port 22 open
add adsl_router tcp port 23 "/usr/share/honeyd/scripts/router-telnet.pl"
add adsl_router tcp port 80 open
set router uptime 3002545
```

Plain Text  Tab Width: 8  Ln 1, Col 1  INS

Σχήμα 3.2: Μέρος του αρχείου παραμετροποίησης του Honeyd, "honeyd.conf"

Ολόκληρο το αρχείο παραμετροποίησης του Honeyd θα είναι διαθέσιμο στο παράστημα A1.

Συνολικά δημιουργήθηκαν εφτά honeypots με σκοπό να προσομοιάσουν διαφορετικά συστήματα και υπηρεσίες. Παρακάτω στους πίνακες 3.1 έως 3.7 θα γίνει παρουσίαση των honeypots καθώς και των υπηρεσιών που υποστηρίζουν σε κάθε πόρτα.

Το πρώτο honeypot που ορίζεται είναι το default. Αυτό το honeypot συσχετίζεται με μια συγκεκριμένη IP άλλα επίσης αναλαμβάνει να «απαντάει» και σε όλες τις διευθύνσεις IP του υποδικτύου που του έχουμε ορίσει ως είσοδο και οι οποίες δεν είναι συσχετισμένες με κάποιο άλλο honeypot. Η ρύθμιση του φαίνεται στον πίνακα 3.1.

Πίνακας 3.1: Περιγραφή του default honeypot

| **Default Honeypot** | | | |
|---|---|---|---|
| **Προσωπικότητα:** Microsoft Windows XP SP1 | | | |
| **Διεύθυνση IP:** 143.XXX.XXX.94 | | | |
| **TCP action:** Reset | | **UDP action:** Reset | **ICMP action:** Open |
| **Πόρτα** | **Πρωτόκολλο** | **Ιδιότητα** | **Υπηρεσία** |
| 21 | TCP | - | Έχει ορισθεί το script msftp.sh το οποίο είναι ένας εξομοιωτής της Microsoft FTP υπηρεσίας. |
| 22 | TCP | open | - |
| 25 | TCP | - | Έχει ορισθεί το script exchange-smtp.sh, το οποίο εξομοιώνει την SMTP (simple mail transfer protocol) υπηρεσία. |
| 80 | TCP | - | Έχει ορισθεί το script iis.sh το οποίο εξομοιώνει τη λειτουργία του IIS Server της Microsoft. |
| 135 | TCP | open | - |
| 138 | TCP | open | - |
| 139 | TCP | open | - |
| 143 | TCP | - | Έχει ορισθεί το script exchange-imap.sh το οποίο εξομοιώνει τη λειτουργία ενός IMAP server. |
| 445 | TCP | open | - |
| 1080 | TCP | - | Έχει ορισθεί το script mydoom.pl το οποίο προσομοιώνει το backdoor που χρησιμοποιεί ο ιός mydoom ώστε να εξαπλωθεί. |
| 1433 | TCP | tarpit open | - |
| 1434 | UDP | tarpit open | - |
| 3117 | TCP | - | Έχει ορισθεί το script cmdexe.pl το οποίο εξομοιώνει το command |

| | | | |
|---|---|---|---|
| | | | prompt των Windows |
| 3127 | TCP | - | Έχει ορισθεί επίσης το script mydoom.pl |
| 3128 | TCP | - | Έχει ορισθεί επίσης το script mydoom.pl |
| 4444 | TCP | | Έχει ορισθεί το script 4444.sh το οποίο είναι ένα script γραμμένο για την αντιμετώπιση του worm msblast. |
| 5554 | TCP | | Έχει ορισθεί το script lsass4.sh το οποίο είναι γραμμένο για την αντιμετώπιση του worm sasser |
| 8967 | TCP | | Έχει ορισθεί επίσης το script cmdexe.pl |
| 9996 | TCP | | Έχει ορισθεί επίσης το script lsass4.sh |
| 10080 | TCP | | Έχει ορισθεί επίσης το script mydoom.pl |
| 20168 | TCP | | Έχει ορισθεί επίσης το script cmdexe.pl |

Πίνακας 3.2: Περιγραφή του router honeypot

| **Router Honeypot** | | | |
|---|---|---|---|
| **Προσωπικότητα:** Cisco 1601R router running IOS 12.1(5) | | | |
| **Διεύθυνση IP:** 143.XXX.XXX.93 | | | |
| **TCP action:** Reset | | **UDP action:** Reset | **ICMP action:** Open |
| **Πόρτα** | **Πρωτόκολλο** | **Ιδιότητα** | **Υπηρεσία** |
| 22 | TCP | - | Έχει ορισθεί το script test.sh το οποίο είναι script το οποίο |
| 23 | TCP | - | Έχει ορισθεί το script router-telnet.pl το οποίο εξομοιώνει τη λειτουργία της telnet υπηρεσίας. |

Πίνακας 3.3: Περιγραφή του adsl_router honeypot

| **Adsl_router Honeypot** | | | |
|---|---|---|---|
| **Προσωπικότητα:** DLink DI-604 ethernet router | | | |
| **Διεύθυνση IP:** 143.XXX.XXX.97 | | | |
| **TCP action:** Reset | | **UDP action:** Reset | **ICMP action:** Open |
| **Πόρτα** | **Πρωτόκολλο** | **Ιδιότητα** | **Υπηρεσία** |
| 22 | TCP | open | - |
| 23 | TCP | - | Έχει ορισθεί το script router-telnet.pl το οποίο εξομοιώνει τη |

| | | | |
|---|---|---|---|
| | | | λειτουργία της telnet υπηρεσίας. |
| 80 | TCP | open | - |

Πίνακας 3.4: Περιγραφή του linux honeypot

| **Linux Honeypot** | | | |
|---|---|---|---|
| **Προσωπικότητα:** Linux kernel 2.4.20 | | | |
| **Διεύθυνση IP:** 143.XXX.XXX.95 | | | |
| **TCP action:** Reset | | **UDP action:** Reset | **ICMP action:** Open |
| **Πόρτα** | **Πρωτόκολλο** | **Ιδιότητα** | **Υπηρεσία** |
| 21 | TCP | - | Έχει ορισθεί το script proftpd.sh το οποίο εξομοιώνει τη λειτουργία ενός ftp server. |
| 22 | TCP | - | Έχει ορισθεί το script ssh.sh το οποίο εξομοιώνει την λειτουργία της ssh υπηρεσίας. |
| 25 | TCP | - | Έχει ορισθεί το script sendmail.sh το οποίο εξομοιώνει τη λειτουργία του προγράμματος sendmail το οποίο χρησιμοποιείται από mail servers. |
| 80 | TCP | - | Έχει ορισθεί το script apache.sh το οποίο εξομοιώνει τη λειτουργία του Apache server. |
| 110 | TCP | - | Έχει ορισθεί το script qpop.sh το οποίο εξομοιώνει τη λειτουργία ενός pop3 mail server. |
| 143 | TCP | - | Έχει ορισθεί το script cyrus-imapd.sh το οποίο εξομοιώνει τη λειτουργία ενός IMAP mail server. |
| 514 | | - | Έχει ορισθεί το script syslogd.sh |
| 8080 | | - | Έχει ορισθεί το script squid.sh |

Πίνακας 3.5: Περιγραφή του solaris honeypot

| **Solaris Honeypot** | | |
|---|---|---|
| **Προσωπικότητα:** Sun Solaris 9 | | |
| **Διεύθυνση IP:** 143.XXX.XXX.96 | | |
| **TCP action:** Reset | **UDP action:** Reset | **ICMP action:** Open |

| Πόρτα | Πρωτόκολλο | Ιδιότητα | Υπηρεσία |
|---|---|---|---|
| 25 | TCP | - | Έχει ορισθεί το script smtp.pl το οποίο εξομοιώνει τη λειτουργία του SMTP(Simple Mail Transfer Protocol) πρωτοκόλλου. |
| 8080 | TCP | - | Έχει ορισθεί το script proxy.pl το οποίο εξομοιώνει τη λειτουργία ενός Internet proxy. |

Πίνακας 3.6: Περιγραφή του solaris honeypot

| **Linux2 Honeypot** | | | |
|---|---|---|---|
| **Προσωπικότητα:** DLink DI-604 ethernet router | | | |
| **Διεύθυνση IP:** 143.XXX.XXX.98 | | | |
| **TCP action:** Reset | | **UDP action:** Reset | **ICMP action:** Open |
| **Πόρτα** | **Πρωτόκολλο** | **Ιδιότητα** | **Υπηρεσία** |
| 21 | TCP | - | Έχει ορισθεί το script ftp.sh το οποίο προσομοιάζει την λειτουργία ενός FTP server. |
| 22 | TCP | - | Έχει ορισθεί το script ssh.sh το οποίο εξομοιώνει τη λειτουργία της ssh υπηρεσίας. |
| 25 | TCP | - | Έχει ορισθεί το script sendmail.sh το οποίο εξομοιώνει τη λειτουργία της εφαρμογής Sendmail. |
| 80 | TCP | open | - |
| 161 | TCP | - | Έχει ορισθεί το script fake-snmp.pl το οποίο εξομοιώνει τη λειτουργία του SNMP πρωτοκόλλου. |
| 3306 | TCP | tarpit open | - |
| 6881 | TCP | tarpit open | - |
| 6882 | TCP | tarpit open | - |
| 8080 | TCP | tarpit open | - |

Πίνακας 3.7: Περιγραφή του mail server honeypot

| **Mail_server Honeypot** | | |
|---|---|---|
| **Προσωπικότητα:** Linux 2.5.25 - 2.5.70 or Gentoo 1.2 Linux 2.4.19 rc1-rc7) | | |
| **Διεύθυνση IP:** 143.XXX.XXX.99 | | |
| **TCP action:** Reset | **UDP action:** Reset | **ICMP action:** Open |

| Πόρτα | Πρωτόκολλο | Ιδιότητα | Υπηρεσία |
|---|---|---|---|
| 23 | TCP | - | Έχει ορισθεί το script router-telnet.pl το οποίο εξομοιώνει τη λειτουργία της Telnet υπηρεσίας. |
| 25 | TCP | - | Έχει ορισθεί το script sendmail.sh το οποίο εξομοιώνει τη λειτουργία της εφαρμογής Sendmail. |
| 80 | TCP | open | - |
| 110 | TCP | - | Έχει ορισθεί το script qpop.sh το οποίο εξομοιώνει τη λειτουργία ενός pop3 mail server. |
| 143 | TCP | - | Έχει ορισθεί το script cyrus-imapd.sh το οποίο εξομοιώνει τη λειτουργία ενός IMAP mail server. |

Οι πόρτες που χρησιμοποιήθηκαν παραπάνω επεξηγούνται στον πίνακα Γ.1 στο παράρτημα Γ.

**Εκκίνηση του Honeyd**

Για να εκκινήσουμε το Honeyd πληκτρολογήσαμε την παρακάτω την εντολή:

```
honeyd -u 99 -g 99 -d -l /usr/share/honeyd/log/honeyd/messages -p /etc/honeypot/nmap.prints -f /etc/honeypot/honeyd.conf -i eth0 143.XXX.XXX.0/24
```

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.

Για να εκκινήσουμε το Fake ARP πληκτρολογούμε την παρακάτω εντολή:

```
farpd -d -i eth0 143.XXX.XXX.0/24
```

Οι παράμετροι που υπάρχουν στην παραπάνω εντολή εξηγούνται ως εξής:

-d Με αυτή την παράμετρο καθορίζουμε πως το Fake ARP θα εκκινήσει σε κατάσταση αποσφαλμάτωσης δηλαδή θα εμφανίζει όλα τα μηνύματα στην οθόνη του υπολογιστή και δεν θα τρέχει ως δαίμονας στο παρασκήνιο

-i Με αυτή την παράμετρο καθορίζουμε ποια διεπαφή θα παρακολουθεί το Fake ARP

Στο τέλος καθορίζουμε όπως και στο Honeyd το υποδίκτυο το οποίο θα παρακολουθεί το Fake

ARP.
Στο σύστημα γίνεται επίσης εκκίνηση και του Tcpdump με σκοπό την πλήρη καταγραφή της εισερχόμενης και της εξερχόμενης κίνησης στο σύστημα.

Η εκκίνηση του Tcpdump γίνεται με την παρακάτω εντολή:

```
tcpdump -w /usr/share/honeypot/log/tcpdump/file_1  -i eth0
```

Οι παράμετροι στην παραπάνω εντολή εξηγούνται παρακάτω

-w Με αυτή την παράμετρο καθορίζουμε το αρχείο στο οποίο θα αποθηκεύει το Tcpdump την κίνηση την οποία καταγράφει.

-i Με αυτή την παράμετρο καθορίζουμε ποια διεπαφή θα χρησιμοποιήσει το Tcpdump

## Ανάλυση Δεδομένων

Παρακάτω θα γίνει η ανάλυση των δεδομένων που προέκυψαν κατά το διάστημα που το πείραμα ήταν σε εξέλιξη. Συνολικά το πείραμα είχε χρονική διάρκεια περίπου ένα μήνα, από τις 10-06-2009 μέχρι και τις 13-07-2009. Τα δεδομένα χωρίζονται σε δύο μέρη, αυτά που προέκυψαν από το Tcpdump και αυτά που προέκυψαν από το Honeyd. Προτού πραγματοποιηθεί η ανάλυση των αποτελεσμάτων θα γίνει μια σύντομη αναφορά σε ζητήματα που επηρέασαν τα τελικά αποτελέσματα.

### Σημαντικά Ζητήματα

Παρακάτω θα γίνει αναφορά σε διάφορα σημαντικά ζητήματα τα οποία όπως φάνηκε επηρέασαν τα τελικά αποτελέσματα. Τα ζητήματα αυτά είναι τα εξής:

1. **Snort:** Το Snort δεν είχε τη δυνατότητα να παράγει προειδοποιήσεις για τις επιθέσεις που λάμβαναν χώρο στον χώρο διευθύνσεων που είχε αποδοθεί στο Default template.

Αντίθετα παρήγαγε προειδοποιήσεις μονάχα για τις διευθύνσεις IP που είχαν οριστεί ως honeypots στις ρυθμίσεις του Honeywall. Αυτό είναι ένα σημαντικό μειονέκτημα καθώς σημαίνει πως θα έπρεπε να γίνει προσεκτική ανάλυση καταγεγραμμένων δεδομένων εφόσον οι προειδοποιήσεις του Snort δεν περιείχαν όλη την απαραίτητη πληροφορία.

2. **Honeyd:** Στο μηχάνημα το οποίο φιλοξενούσε το Honeyd (Ikaria) αποδόθηκε μια δημόσια διεύθυνση IP, μέρος του Vlan που χρησιμοποιήθηκε για την πραγματοποίηση του πειράματος, προκειμένου να μπορέσει η διεπαφή να χρησιμοποιηθεί για την εκτέλεση του Honeyd. Αποτέλεσμα ήταν το Ikaria να μετατραπεί σε honeypot και η διεύθυνση IP του να βρίσκεται αρκετά ψηλά στη λίστα με τις διευθύνσεις που δέχθηκαν την περισσότερη δικτυακή κίνηση όπως θα δούμε και παρακάτω. Παρόλο που τίποτα από τα παραπάνω δεν ήταν επιθυμητό, δεν υπήρχε άλλη δυνατότητα επιλογής.

3. **DNS honeypot:** Το μηχάνημα το οποίο εκτελούσε το ρόλο του DNS honeypot (Bilem) ορίστηκε ως υπεύθυνος διακομιστής DNS για το Ikaria καθώς και για τα εικονικά honeypots μέσω του Honeywall. Αποτέλεσμα ήταν η διεύθυνση IP του Bilem να βρεθεί και αυτή ψηλά στη λίστα με τις διευθύνσεις IP που δέχθηκαν την περισσότερη δικτυακή κίνηση. Αυτό μας καταδεικνύει το πόσο πολύ χρησιμοποιούνται οι διακομιστές DNS ακόμα και για τις απλές επικοινωνίες.

4. **Μηχανή διαχείρισης:** Η συγκεκριμένη μηχανή χρησιμοποιήθηκε για τη διαχείριση των μηχανημάτων που συμμετείχαν στο πείραμα. Επιπλέον, από αυτό το μηχάνημα, πραγματοποιήθηκαν κάποιες ανιχνεύσεις (ενεργητικές) προς τα εικονικά μηχανήματα, κατά τη διάρκεια του πειράματος, για να πιστοποιηθεί η καλή λειτουργία τους. Λόγω του ότι οι ενεργητικές ανιχνεύσεις αποστέλλουν ένα μεγάλο αριθμό αρχείων η εξωτερική διεύθυνση IP (αυτή στην οποία αντιστοιχίζεται η διεύθυνση IP του εσωτερικού δικτύου σύμφωνα με το NAT) της μηχανής διαχείρισης βρέθηκε και αυτή στη λίστα με τις διευθύνσεις που πραγματοποίησαν την περισσότερη δικτυακή κίνηση. Και αυτή η ενέργεια κρίνεται εσφαλμένη καθώς η μηχανή διαχείρισης δεν θα έπρεπε σε καμία περίπτωση να επηρεάσει τα αποτελέσματα.

**Ανάλυση Δεδομένων Tcpdump**

Η ανάλυση ξεκίνησε από την ανάλυση των αρχείων του Tcpdump τα οποία ενώσαμε σε ένα ενιαίο με ονομασία Total.pcap. Αρχικά προσπαθήσαμε να αναλύσουμε την καταγεγραμμένη κίνηση με το πρόγραμμα Wireshark, αλλά λόγω του μεγέθους του αρχείου (854mb) στάθηκε αδύνατο. Το Tshark, έχοντας λιγότερες απαιτήσεις σε πόρους, μπορεί να αντεπεξέλθει τον φόρτο εργασίας της συγκεκριμένης εργασίας.

Οι εντολές που χρησιμοποιήσαμε στο Tshark φαίνονται στον πίνακα 3.8 συνοδευμένες με μια μικρή περιγραφή:

Πίνακας 3.8: Εντολές στο Tshark

| Εντολή | Περιγραφή |
|---|---|
| tshark -r (όνομα αρχείου) -z io,phs -q -o (όνομα αρχείου) | Με αυτήν την εντολή το Tshark μας παρουσιάζει στατιστικά για την ιεραρχία πρωτοκόλλων στο αρχείο. |
| tshark -r (όνομα αρχείου) -z conv,tcp -q -o (όνομα αρχείου) | Με αυτήν την εντολή το Tshark μας παρουσιάζει στατιστικά για τις συνομιλίες του πρωτοκόλλου TCP. |
| tshark -r (όνομα αρχείου) -z conv,udp -q -o (όνομα αρχείου) | Με αυτήν την εντολή το Tshark μας παρουσιάζει στατιστικά για τις συνομιλίες του πρωτοκόλλου UDP. |
| tshark -r (όνομα αρχείου) -z conv,ip -q -o (όνομα αρχείου) | Με αυτήν την εντολή το Tshark μας παρουσιάζει στατιστικά για τις συνομιλίες του πρωτοκόλλου IP. |
| tshark -r (όνομα αρχείου) -z conv,eth -q -o (όνομα αρχείου) | Με αυτήν την εντολή το Tshark μας παρουσιάζει στατιστικά για τις συνομιλίες του πρωτοκόλλου ETHERNET. |
| tshark -r (όνομα αρχείου) -z ip_hosts,tree -q -o (όνομα αρχείου) | Με αυτήν την εντολή το Tshark μας παρουσιάζει στατιστικά για τις IP διευθύνσεις που εμφανίζονται στο αρχείο. |

Οι παράμετροι που χρησιμοποιήθηκαν στις παραπάνω εντολές επεξηγούνται παρακάτω:

-r Με αυτή την παράμετρο καθορίζουμε στο Tshark το αρχείο με τα δεδομένα το οποίο θα διαβάσει.

-z Με αυτή την παράμετρο καθορίζουμε στο Tshark ότι θα παράγει στατιστικά για το αρχείο το οποίο του έχουμε ορίσει να διαβάσει.

-q Με αυτή την παράμετρο καθορίζουμε στο Tshark πως θέλουμε να μας τυπώσει μονάχα τα στατιστικά και όχι τις πληροφορίες που αφορούν το κάθε πακέτο.

-o Με αυτή την παράμετρο καθορίζουμε στο Tshark το αρχείο στο οποίο θα γράψει τα δεδομένα με την εκτέλεση των εντολών αντί να τα εμφανίσει απλώς στην οθόνη.

Αναλύοντας το Total.pcap με τη βοήθεια του Tshark έχουμε αρχικά τα εξής στοιχεία:

- 11.833.807 πακέτα ήταν η συνολική κίνηση η οποία καταγράφηκε από το Tcpdump στο σύστημα Ikaria ο οποίος φιλοξενούσε το Honeyd.
- 64.443 ήταν συνολικά οι διευθύνσεις IP οι οποίες καταγράφηκαν. Ο αριθμός αυτός αναφέρεται στις διευθύνσεις IP που εμφανίζονται έστω και μία φορά (είτε ως source, είτε ως destination IP) σε κάποιο από τα πακέτα που περιέχονται στο αρχείο Total.pcap.

Στον πίνακα 3.9 φαίνονται οι πέντε διευθύνσεις IP οι οποίες εμφανίζονται τις περισσότερες φορές στο αρχείο Total.pcap.

Πίνακας 3.9: Διευθύνσεις IP με τις περισσότερες εμφανίσεις

| **Διεύθυνση IP** | **Χώρα Προέλευσης** | **Αριθμός Εμφανίσεων** | **Ποσοστό επί της %** |
|---|---|---|---|
| 222.XXX.XXX.76 | Κίνα | 4.925.520 | 47,09 |
| 60.XXX.XXX.54 | Κίνα | 777.373 | 7,43 |
| 61.XXX.XXX.73 | Κίνα | 725.766 | 6,94 |
| 60.XXX.XXX.62 | Κίνα | 507.085 | 4,85 |
| 143.XXX.XXX.92 | Ελλάδα | 399.447 | 3,82 |

Όπως φαίνεται από τον παραπάνω πίνακα, συγκεντρωτικά, περίπου το 67% της κίνησης, η οποία καταγράφηκε από το Tcpdump, προέρχεται από υπολογιστές που βρίσκονται στην Κίνα. Η διεύθυνση IP που βρίσκεται στην πέμπτη θέση του πίνακα ανήκει στο Ikaria. Οι λόγοι που

συνέβη αυτό εξηγήθηκαν πιο πάνω.

Στον πίνακα 3.10 φαίνονται οι «συνομιλίες» για τα διάφορα πρωτόκολλα που κατεγράφησαν στο αρχείο Total.pcap. Σαν συνομιλία ορίζεται η ανταλλαγή πακέτων δεδομένων κατά τη διάρκεια μιας συγκεκριμένης σύνδεσης, από την εγκαθίδρυση της μέχρι και τον τερματισμό της. Αφορά σε όλα τα πρωτόκολλα IP, UDP, TCP ανεξάρτητα αν είναι συνδεμοστραφή ή όχι.

Πίνακας 3.10: Συνομιλίες που καταγράφηκαν για τα διάφορα πρωτόκολλα

| **Πρωτόκολλο** | **Αριθμός Συνομιλιών** |
|---|---|
| TCP | 346.717 |
| UDP | 72.087 |
| IP | 225.021 |
| Ethernet | 12 |

**Ανάλυση Δεδομένων Honeyd**

Πληροφορίες παίρνουμε επίσης από το αρχείο messages στο οποίο αποθηκεύει τα μηνύματα που παράγει το Honeyd.

Σύμφωνα λοιπόν με το αρχείο messages του Honeyd, έχουμε τα εξής δεδομένα:

- 784.026 ήταν συνολικά τα μηνύματα τα οποία κατέγραψε το Honeyd.
- Συνολικά 64.073 διαφορετικές διευθύνσεις IP καταγράφηκαν.

Στον πίνακα 3.11 φαίνονται οι πόρτες οι οποίες δέχθηκαν τον μεγαλύτερο αριθμό μηνυμάτων και προσπαθειών για σύνδεση. Όπως φαίνεται οι πόρτες 1433 και 1434 είναι αυτές οι οποίες δέχθηκαν τις περισσότερες επιθέσεις.

Πίνακας 3.11: Οι πέντε θύρες που δέχθηκαν τον μεγαλύτερο αριθμό μηνυμάτων

| **Θύρα** | **Αριθμός Μηνυμάτων** | **Ποσοστό επί της %** |
|---|---|---|
| 1433 | 196.296 | 25,03 |
| 1434 | 89.553 | 11,42 |
| 23 | 72.648 | 9,26 |
| 22 | 51.527 | 6,57 |
| 2967 | 44.240 | 5,64 |

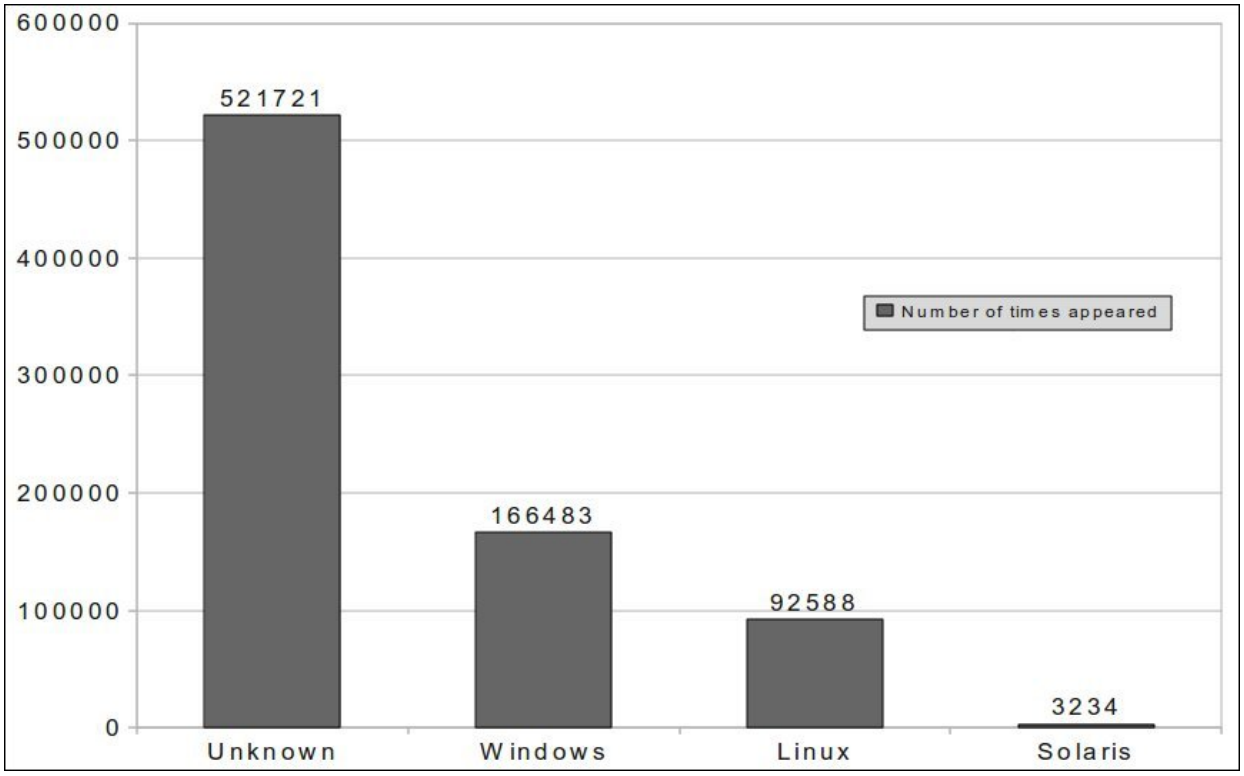
Παρακάτω στον πίνακα 3.12 φαίνονται οι πέντε διευθύνσεις IP που απέστειλαν τα περισσότερα μηνύματα προς τα εικονικά μας honeypots.

Πίνακας 3.12: Οι πέντε διευθύνσεις IP που απέστειλαν τα περισσότερα μηνύματα

| **Διεύθυνση IP** | **Χώρα Προέλευσης** | **Αριθμός Μηνυμάτων** | **Ποσοστό επί της %** |
|---|---|---|---|
| 222.XXX.XXX.76 | Κίνα | 74.116 | 9,45 |
| 143.XXX.XXX.143 | Ελλάδα | 33.835 | 4,31 |
| 61.XXX.XXX.73 | Κίνα | 25.528 | 3,25 |
| 60.XXX.XXX.54 | Κίνα | 20.793 | 2,65 |
| 143.XXX.XXX.68 | Ελλάδα | 14.454 | 1,84 |

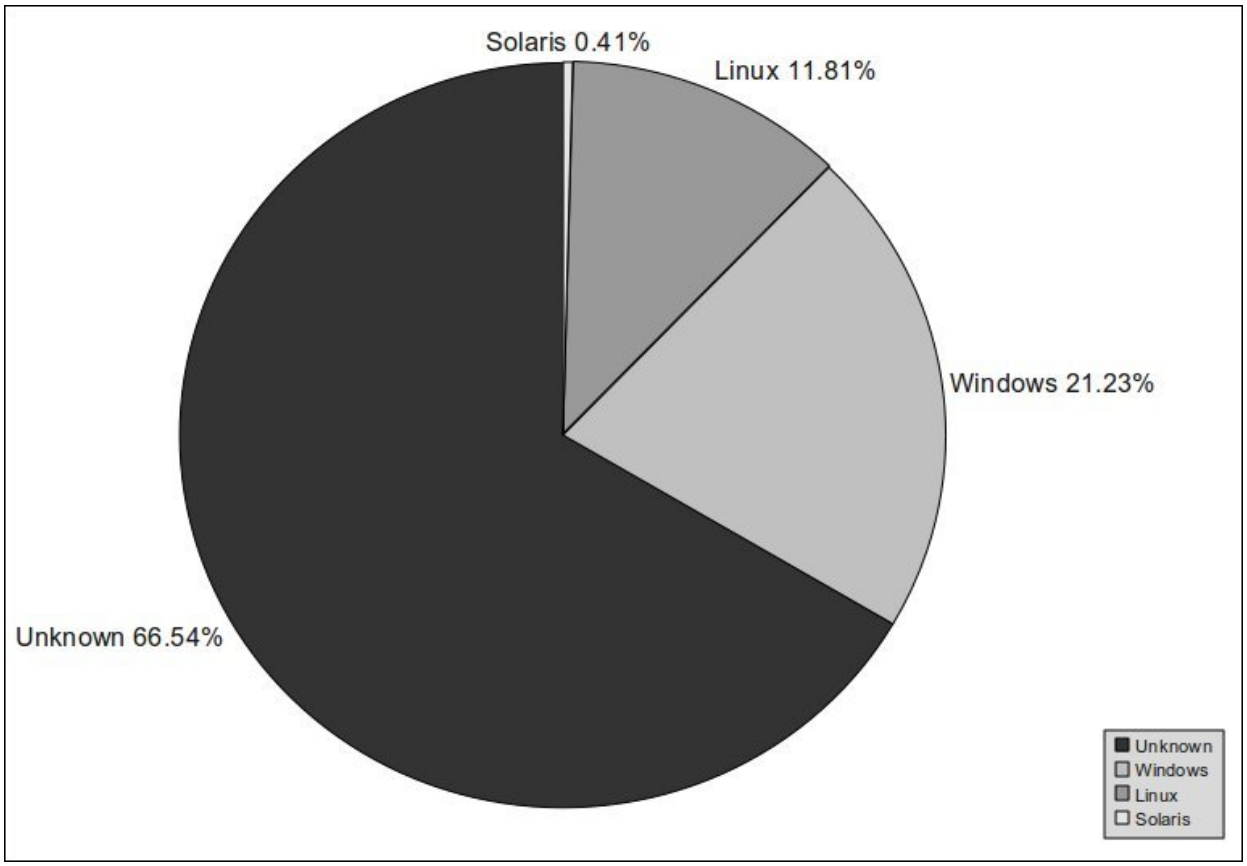
Όπως φαίνεται από τον πίνακα στις τρεις από τις πέντε θέσεις του πίνακα υπάρχουν διευθύνσεις που προέρχονται από υπολογιστές που βρίσκονται στην Κίνα. Οι άλλες δύο διευθύνσεις ανήκουν σε μηχανήματα του εργαστηρίου. Η πρώτη, που βρίσκεται στη δεύτερη θέση, ανήκει στο DNS honeypot και η δεύτερη, που βρίσκεται στην πέμπτη θέση, ανήκει στη μηχανή διαχείρισης. Ο λόγος που συνέβη αυτό εξηγήθηκαν πιο πάνω.

Παράλληλα με τα μηνύματα που καταγράφει, το Honeyd προσπαθεί να προβλέψει το λειτουργικό σύστημα του επιτιθέμενου. Στα σχήματα 3.3 και 3.4 φαίνονται τα είδη των λειτουργικών συστημάτων που εντόπισε το Honeyd.




Σχήμα 3.3: Αριθμός εμφανίσεων των λειτουργικών συστημάτων που εντόπισε το Honeyd

Παραπάνω στο σχήμα 3.3 τα λειτουργικά συστήματα ταξινομούνται με βάση τον αριθμό των εμφανίσεων τους  ενώ κάτω στο σχήμα 3.4 τα λειτουργικά συστήματα ταξινομούνται με βάση το ποσοστό επί της %.




Σχήμα 3.4: Ποσοστό επί της % των λειτουργικών συστημάτων που εντόπισε το Honeyd

## Σύγκριση Δεδομένων

Όπως είδαμε παραπάνω τα στοιχεία που προέκυψαν από την ανάλυση των δεδομένων του Honeyd και του Tcpdump δεν συμφωνούν απόλυτα μεταξύ τους. Συγκεκριμένα παρατηρώντας τους πίνακες 3.9 και 3.12 βλέπουμε, πως η ιεραρχία των διευθύνσεων IP που εμφανίζονται τις περισσότερες φορές, διαφέρει τόσο στην σειρά όσο και στις διευθύνσεις που καταγράφηκαν. Επίσης παρατηρήσαμε πως το αρχείο καταγραφής μηνυμάτων του Honeyd κατέγραψε ένα σημαντικά μικρότερο αριθμό όγκο δεδομένων (784.026 μηνύματα) σε σχέση με αυτό που κατέγραψε το Tcpdump (11.833.807 πακέτα). Αυτό συνέβη γιατί το Honeyd καταγράφει στο αρχείο μηνυμάτων του μονάχα τα εισερχόμενα πακέτα των πρωτοκόλλων TCP, UDP και ICMP. Αντίθετα το Tcpdump, επειδή λειτουργεί σε "promiscuous mode", καταγράφει οποιοδήποτε πακέτο ληφθεί ή αποσταλεί από την κάρτα δικτύου ανεξάρτητα από το ποιος είναι

ο τελικός του προορισμός. Κατά αυτό τον τρόπο τα δεδομένα του Tcpdump περιλαμβάνουν πακέτα διαφόρων πρωτοκόλλων, εκτός των TCP, UDP και ICMP (π.χ ARP, STP, TLD κ.α) και είναι ιδιαίτερα αυξημένα σε όγκο. Τα επιπλέον πρωτόκολλα που καταγράφονται συνήθως αποτελούν μέρος της τοπικής δικτυακής κίνησης (STP, ARP).

Αξιολογώντας την αξιοπιστία των καταγεγραμμένων δεδομένων θα μπορούσε να ειπωθεί, πως τα δεδομένα του Tcpdump είναι τα πληρέστερα, καθώς καταγράφεται σε αυτά η οποιαδήποτε δικτυακή κίνηση από και προς τον υπολογιστή στον οποίο λειτουργεί εν αντιθέσει με τα δεδομένα του Honeyd στα οποία καταγράφεται μονάχα η εισερχόμενη κίνηση συγκεκριμένων πρωτοκόλλων. Το ότι είναι πληρέστερα όμως δεν σημαίνει πως είναι και πιο αξιόπιστα. Δεδομένου ότι το Honeyd μπορεί να εξομοιώσει μοναχά υπηρεσίες που βασίζονται στα πρωτόκολλα TCP, UDP και ICMP καταγράφει ακριβώς την δικτυακή κίνηση την οποία χρειάζεται και όχι οτιδήποτε «πλεονασματικό» όπως συμβαίνει με το Tcpdump. Αυτό μπορούμε να το αντιληφθούμε εξετάζοντας τον συνολικό αριθμό μοναδικών διευθύνσεων IP που βρέθηκε στα δεδομένα του Tcpdump (64.443) και του Honeyd (64.073) αντίστοιχα. Όπως είναι φανερό οι αριθμοί διαφέρουν ελάχιστα. Δηλαδή το Honeyd δεν κατάγραψε λιγότερες επιθέσεις, απλώς λιγότερα δεδομένα.

Συνοψίζοντας θα μπορούσαμε να πούμε πως τα μηνύματα τα οποία καταγράφει το Honeyd μπορούν να αποτελέσουν βάση για μια σωστή ανάλυση. Ακόμα ορθότερα συμπεράσματα θα μπορούσε να εξάγει όμως κάποιος, εξετάζοντας τα δεδομένα του Tcpdump. Σκόπιμο θα ήταν να αποκλειστούν τα «πλεονάζοντα» πρωτόκολλα αφού ο αυξημένος όγκος δεδομένων απαιτεί και πολύωρη εργασία. Ιδανικότερη λύση είναι η εξέταση των δεδομένων του Honeyd σε πρώτο στάδιο και η αναζήτηση περισσότερων λεπτομερειών στα περιεχόμενα των Ethernet πακέτων που έχει καταγράψει το Tcpdump.

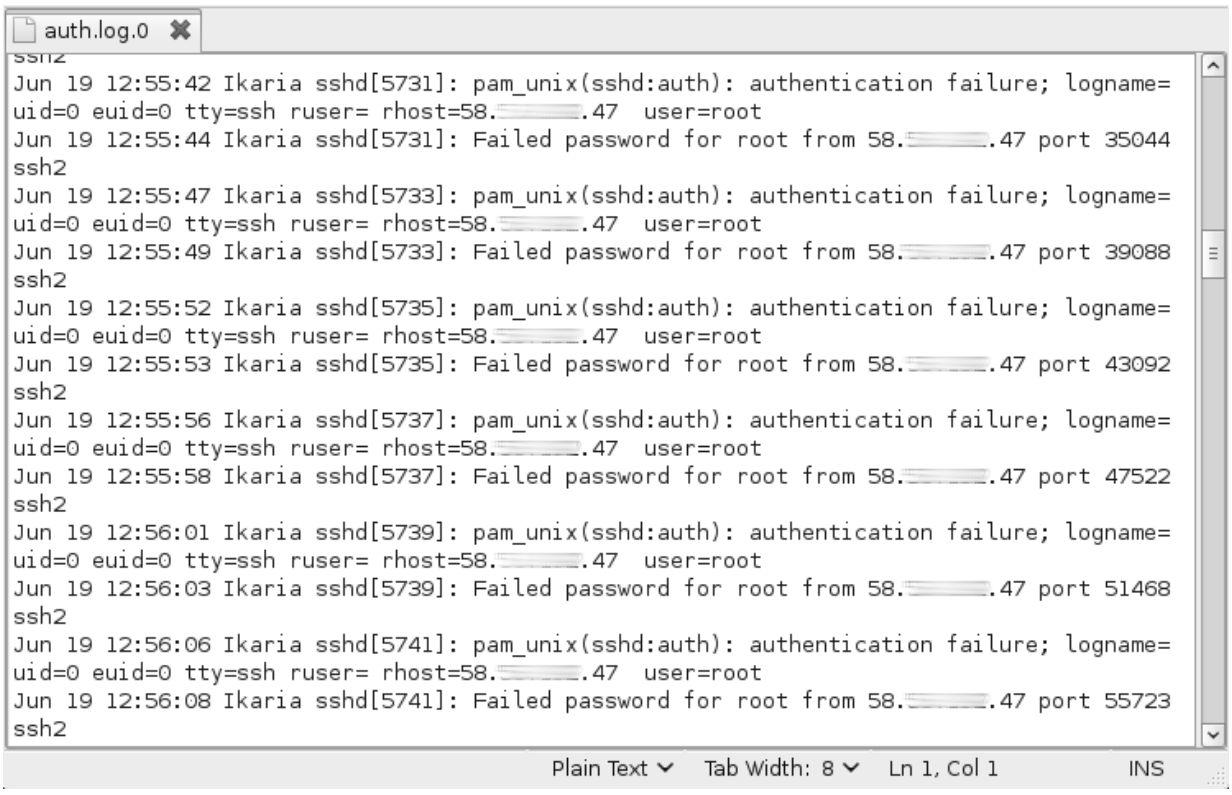
## Παραδείγματα Επιθέσεων

Παρακάτω θα παρατεθούν παραδείγματα επιθέσεων όπως αυτά καταγράφηκαν στα αρχεία καταγραφής συμβάντων των συστημάτων. Θα πρέπει βέβαια να αναφέρουμε πως αφού ο IP

χώρος που χρησιμοποιήθηκε για το πείραμα αποτελεί μέρος του αχρησιμοποίητου χώρου του ΕΚΕΦΕ Δημόκριτος, η οποιαδήποτε κίνηση προς αυτό θεωρείται επίθεση. Εξαίρεση μπορούν να αποτελέσουν μόνο οι broadcast μεταδόσεις καθώς και τα πακέτα που αποστέλλονται από την μηχανή απομακρυσμένης διαχείρισης, όταν αυτά συμβαδίζουν με τις ακριβείς ώρες πραγματοποίησης της διαχείρισης.

**SSH Επιθέσεις**

Στο σχήμα 3.5 φαίνεται μια από τις πολλές προσπάθειες για είσοδο στο σύστημα μέσω του πρωτοκόλλου SSH. Ο υπολογιστής με την διεύθυνση IP 58.XXX.XXX.47 κάνει μια επίθεση εξαντλητικής αναζήτησης για το όνομα χρήστη root. Δοκιμάζει συνεχώς δηλαδή πολλά διαφορετικά συνθηματικά για το όνομα χρήστη root έως ότου εντοπίσει το σωστό και εισέλθει στο σύστημα ως χρήστης με δικαιώματα διαχειριστή. Στα αρχεία καταγραφής συμβάντων του συστήματος καταγράφηκαν πολλές παρόμοιες επιθέσεις. Εκτός βέβαια του root καταγράφηκαν και πολλές επιθέσεις εξαντλητικής αναζήτησης που είχαν ως στόχο να μαντέψουν άλλους λογαριασμούς του συστήματος.




Σχήμα 3.5: Επίθεση εξαντλητικής αναζήτησης για την εύρεση του συνθηματικού

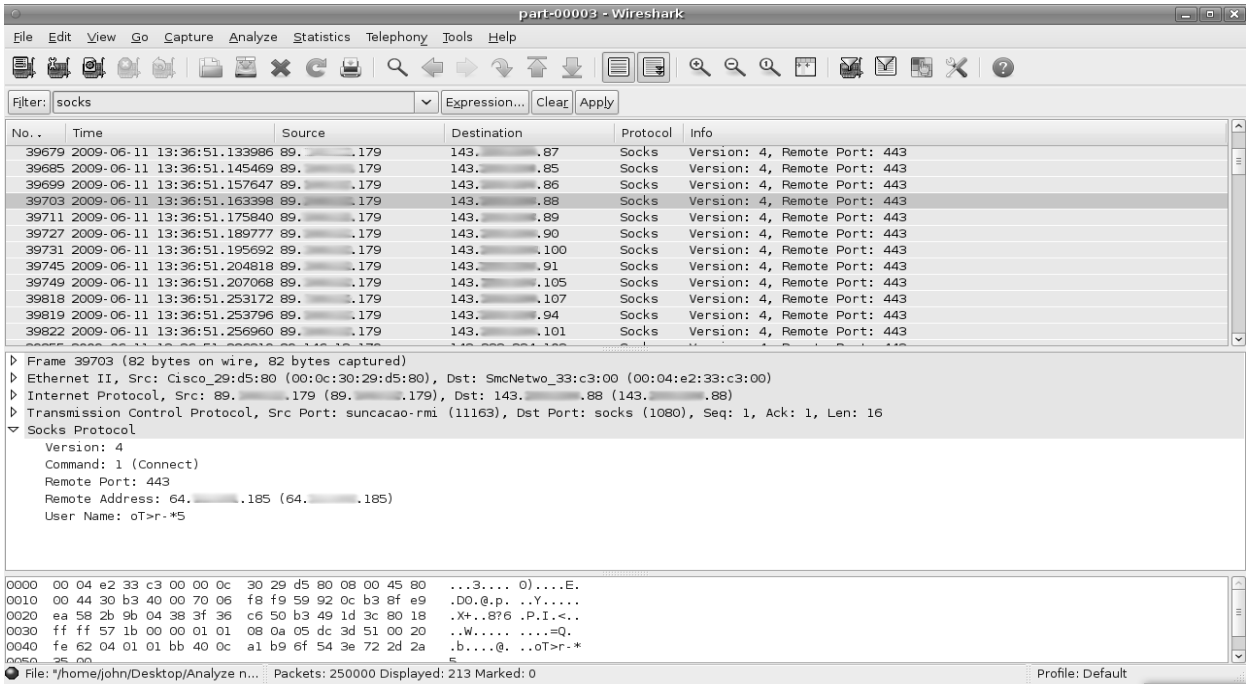
**Προσπάθεια για σύνδεση μέσω του πρωτοκόλλου SOCKS**

Παρακάτω θα αναλυθεί μια από τις προσπάθειες επίθεσης μέσω του πρωτοκόλλου SOCKS που καταγράφηκε από τα συστήματα μας. Το πρωτόκολλο SOCKS χρησιμοποιείται για την επικοινωνία μεταξύ δύο υπολογιστών (ενός πελάτη και ενός εξυπηρετητή) δια μέσω ενός διακομιστή proxy. Το SOCKS λειτουργεί στο στρώμα πέντε, το session layer του μοντέλου OSI. Ο τρόπος λειτουργίας του είναι σχετικά απλός. Όταν ένας πελάτης μιας εφαρμογής θέλει να συνδεθεί με τον διακομιστή της εφαρμογής αυτής επικοινωνεί αρχικά με τον SOCKS διακομιστή. Ο διακομιστής proxy, εν συνεχεία, συνδέεται με τον διακομιστή της εφαρμογής για λογαριασμό του πελάτη της εφαρμογής. Ότι δεδομένα στέλνει ο διακομιστής της εφαρμογής στον διακομιστή proxy εκείνος τα προωθεί στον πελάτη της εφαρμογής και αντίστροφα. Ουσιαστικά για τον διακομιστή της εφαρμογής, ο διακομιστής proxy είναι ο πελάτης της εφαρμογής. Η προκαθορισμένη πόρτα που χρησιμοποιεί το SOCKS για την ανταλλαγή των μηνυμάτων είναι η 1080. Το πρωτόκολλο SOCKS χρησιμοποιείται για την ανώνυμη επικοινωνία μεταξύ δύο υπολογιστών αλλά και για πολλές ακόμα εφαρμογές.

Η πόρτα 1080 είχε οριστεί να είναι ενεργή στο αρχείο παραμετροποίησης του Honeyd και ενδεχομένως πολλοί επιτιθέμενοι να θεώρησαν πως στο μηχάνημα αυτό υπάρχει κάποιος ενεργός SOCKS διακομιστή. Γι' αυτό το λόγο, όπως θα δούμε παρακάτω, υπάρχουν προσπάθειες για πραγματοποίηση επίθεσης μέσω του διακομιστή αυτού.

Στο σχήμα 3.6 βλέπουμε ένα πακέτο της έκδοσης τέσσερα του πρωτοκόλλου SOCKS. Όπως είναι φανερό από το σχήμα ο υπολογιστής με διεύθυνση IP 89.XXX.XXX.179 αποστέλλει πολλές αιτήσεις σύνδεσης προς τον απομακρυσμένο υπολογιστή με διεύθυνση IP 64.XXX.XXX.185 δια μέσω του πρωτοκόλλου SOCKS. Οι αιτήσεις αυτές για σύνδεση πραγματοποιούνται προς διάφορες διευθύνσεις IP, ενώ κάθε φορά δοκιμάζεται και διαφορετικό όνομα χρήστη. Ο χρήστης του υπολογιστή 89.XXX.XXX.179 βρίσκεται στην Ολλανδία. Η διεύθυνση αυτή ανήκει στο χώρο διευθύνσεων της εταιρίας RoutIT η οποία και είναι προφανώς ο Internet Provider του χρήστη που πραγματοποιεί την επίθεση. Από την άλλη η διεύθυνση 64.XXX.XXX.185 ανήκει στο χώρο διευθύνσεων της εταιρίας American Online η οποία είναι ευρέως γνωστή ως AOL. Η διεύθυνση αυτή ανήκει σε έναν από τους διακομιστές της εταιρίας που είναι υπεύθυνος για την αυθεντικοποίηση των χρηστών της υπηρεσίας AIM (AOL Instant

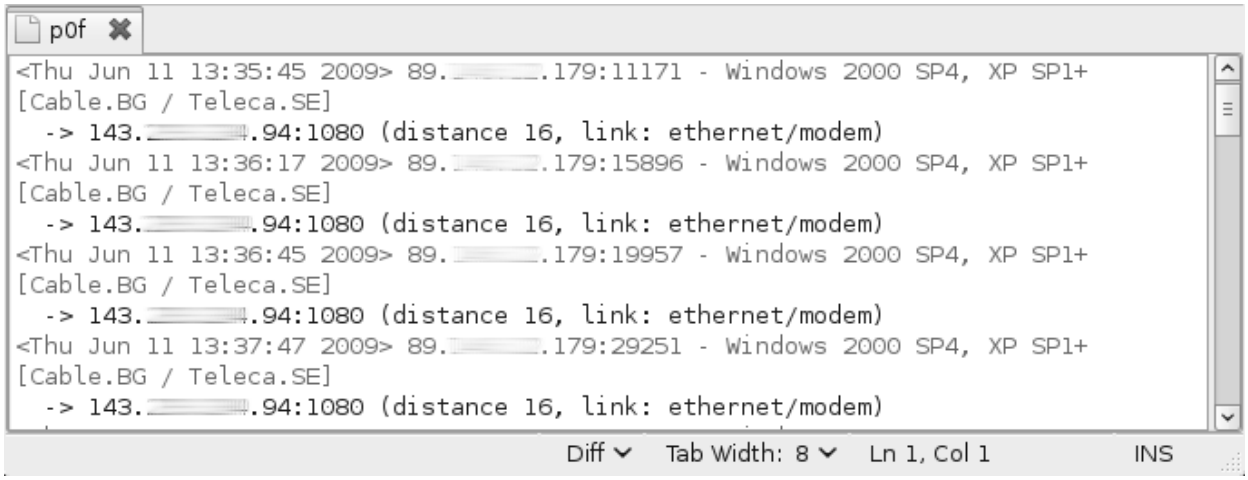
Messenger).




Σχήμα 3.6: Προσπάθεια εύρεσης κωδικού ενός proxy διακομιστή SOCKS

Ο απομακρυσμένος χρήστης λοιπόν προσπαθεί να συνδεθεί στον διακομιστή AIM στην πόρτα 443 (HTTPS) μέσω του υποτιθέμενου proxy διακομιστή SOCKS που υπάρχει στα honeypots. Το ότι αποστέλλονται τόσες πολλές αιτήσεις για σύνδεση από τον απομακρυσμένο υπολογιστή, προς διάφορα honeypots και με διαφορετικό όνομα χρήστη κάθε φορά μας προδιαθέτει ότι δεν πρόκειται για νόμιμη δικτυακή κίνηση. Στο παρελθόν έχουν γίνει αναφορές για επιθέσεις [16], τύπου DoS με τα ίδια χαρακτηριστικά στον Apache server. Σε αυτή την περίπτωση δεν μπορούμε να είμαστε σίγουρα για τα κίνητρα του χρήστη, καθώς δεν εντοπίσαμε ανάλογες αναφορές για το πρωτόκολλο SOCKS.

Περισσότερες πληροφορίες για το χρήστη μπορούμε να εξάγουμε από το Honeywall αναλύοντας τα αρχεία καταγραφής συμβάντων του P0f εκείνη τη δεδομένη χρονική στιγμή. Όπως φαίνεται στο σχήμα 3.7 το P0f μελετώντας τα πακέτα που ανταλλάσσονται προβλέπει πως ο απομακρυσμένος υπολογιστής χρησιμοποιεί το λειτουργικό σύστημα Windows και την έκδοση 2000 SP4 ή XP SP1+ αυτού. Επίσης μας πληροφορεί πως ο απομακρυσμένος υπολογιστής απέχει δεκαέξι ενδιάμεσους σταθμούς (hops).



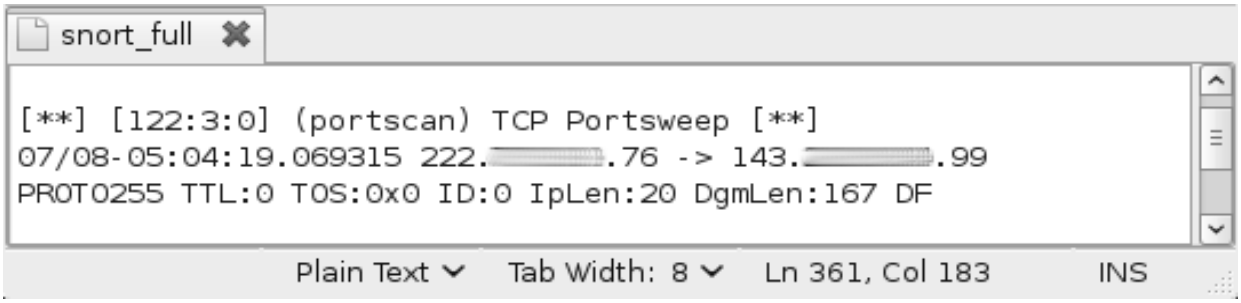

Σχήμα 3.7: Αρχεία καταγραφής συμβάντων του P0f

Μελετώντας όλα τα παραπάνω στοιχεία αλλά και τα καταγεγραμμένα δεδομένα εξάγουμε το συμπέρασμα πως η επίθεση έχει πραγματοποιηθεί από κάποιο αυτοματοποιημένο εργαλείο καθώς είναι αδύνατο ένας χρήστης να έχει κάνει τόσες προσπάθειες σε τόσο μικρό χρονικό διάστημα.

**Επίθεση στον Microsoft SQL Server**

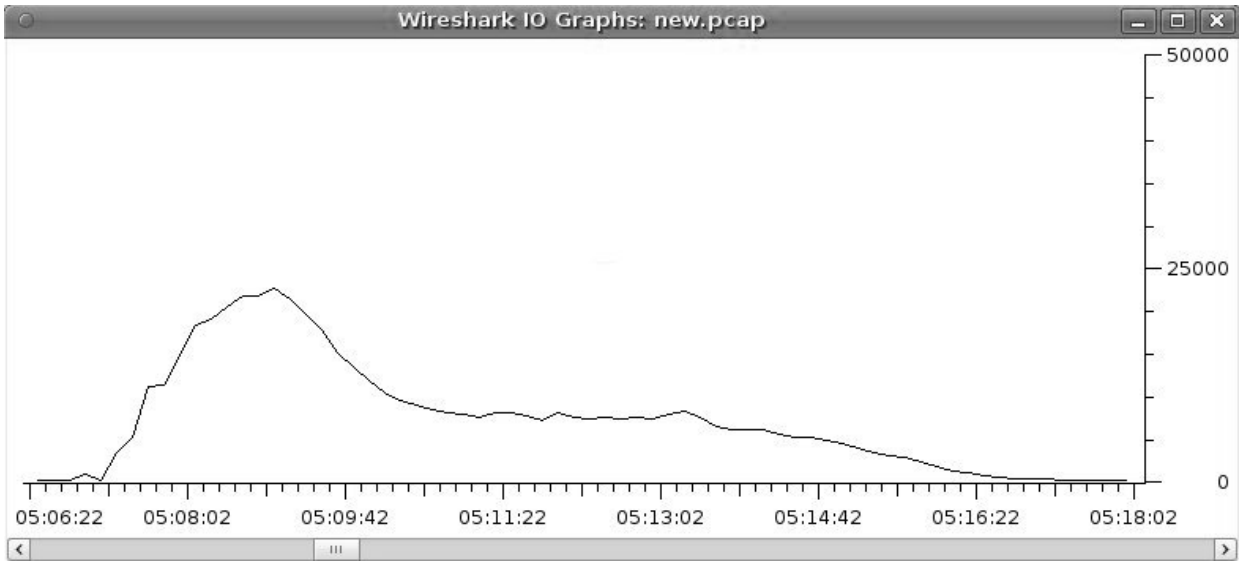
Παρακάτω θα γίνει μια περιγραφή κάποιας από τις πολλές επιθέσεις εις βάρος του εικονικού Microsoft SQL Server. Όπως φαίνεται στον πίνακα Γ.1 στο παράρτημα Γ οι πόρτες τις οποίες χρησιμοποιεί ο MS SQL Server είναι αυτές οι οποίες δέχθηκαν και τις περισσότερες επιθέσεις.

Αρχικά παρατηρώντας τα αρχεία καταγραφής συμβάντων του Honeywall και συγκεκριμένα τις προειδοποιήσεις που παράγει το Snort βλέπουμε πως στις 05:04:19 στις 08-07-2009 εκδίδει προειδοποίηση για την πραγματοποίηση κάποιας σάρωσης από τον υπολογιστή με διεύθυνση IP 222.XXX.XXX.76.




Σχήμα 3.8: Ειδοποίηση του Snort για την πραγματοποίηση κάποιας ανίχνευσης

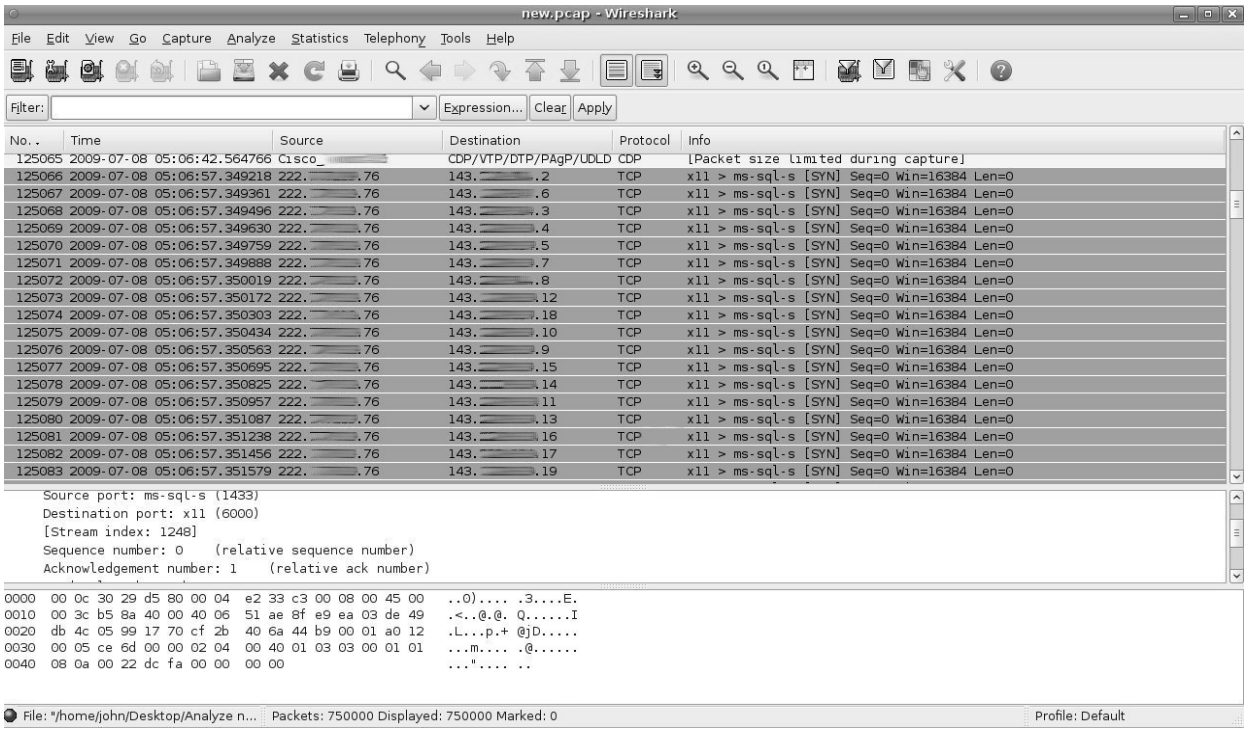
Θέλοντας να εξετάσουμε τι οδήγησε στην παραπάνω προειδοποίηση παρατηρούμε τα αρχεία καταγραφής. Στο σχήμα 3.9 φαίνεται πως στις 05:06:57 στις 8-07-2009 παρατηρείται μεγάλη αύξηση της παρατηρούμενης κίνησης στο δίκτυο. Σύμφωνα με την καμπύλη του σχήματος την δεδομένη στιγμή ο αριθμός των ανταλλασόμενων πακέτων ήταν πολύ κοντά στο μηδέν. Τα επόμενα δευτερόλεπτα παρατηρείται ραγδαία αύξηση ανταλλαγής πακέτων η οποία ολοκληρώνεται μέσα σε μερικά λεπτά και έπειτα η κίνηση ξαναεπιστρέφει στα φυσιολογικά επίπεδα. Είναι φανερό πως κάτι το ύποπτο συνέβη αυτή τη χρονική περίοδο.




Σχήμα 3.9: Ανταλλαγή αρχείων

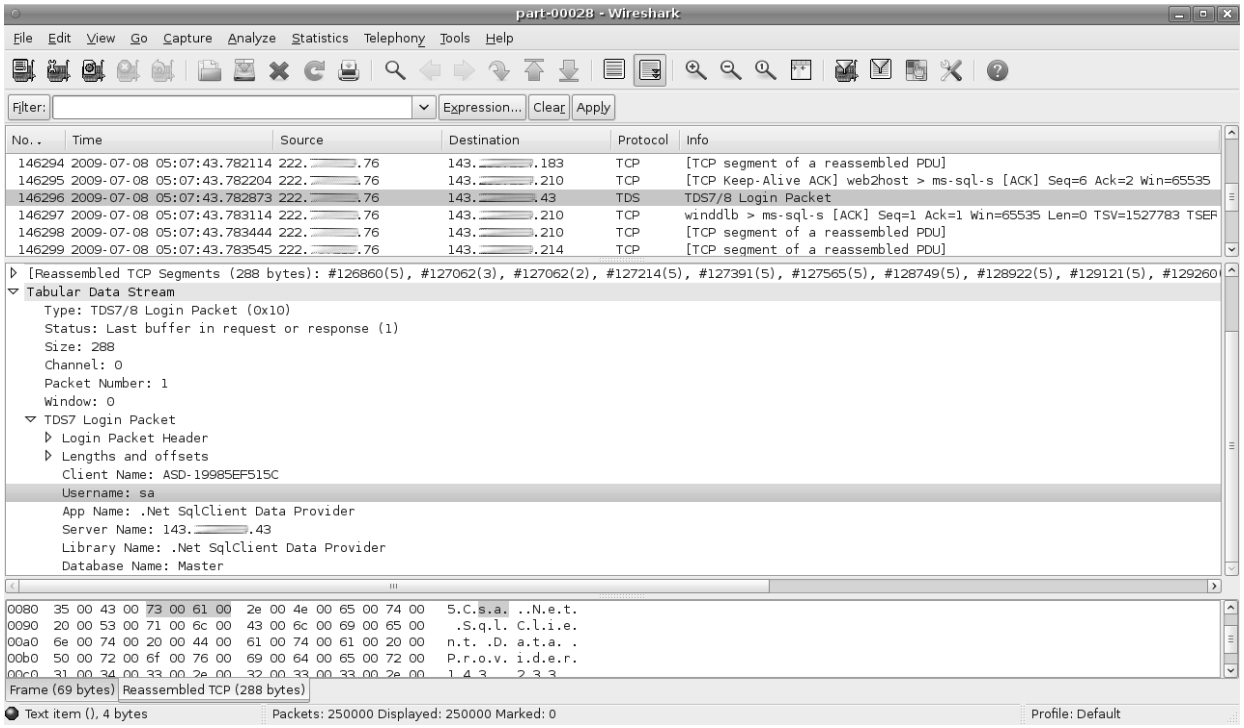
Εξετάζοντας προσεκτικά την κίνηση που προέκυψε στο δίκτυο εκείνη την χρονική στιγμή, διαπιστώνουμε πως προκλήθηκε από ένα και μόνο υπολογιστή. Όπως φαίνεται παρακάτω στο σχήμα 3.10 στο πρώτο πακέτο της ακολουθίας ο υπολογιστής στην διεύθυνση IP

222.XXX.XXX.76 αποστέλλει πακέτα για αίτηση νέας σύνδεσης (με SYN flag δηλαδή) προς διάφορες διευθύνσεις IP εντός του υποδικτύου μας αλλά με πόρτα προορισμού πάντα την 1433.



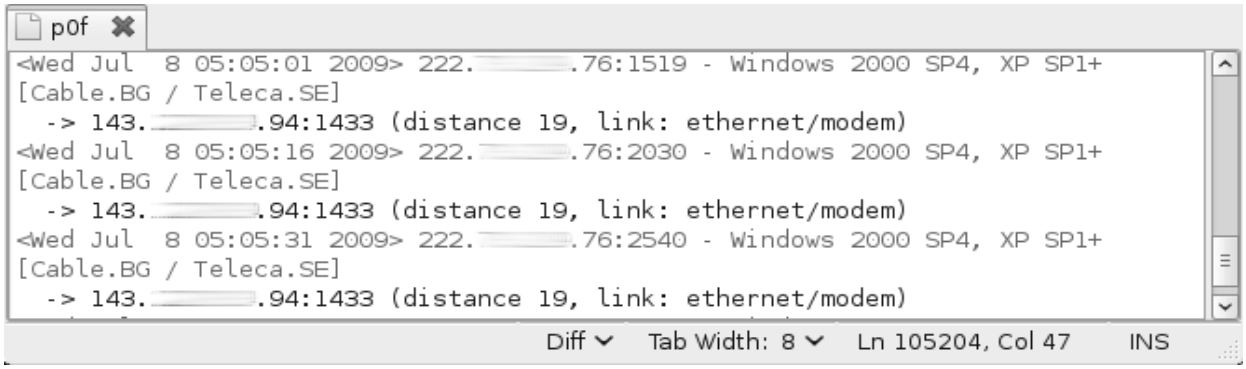

Σχήμα 3.10: Πακέτα σύνδεσης προς την πόρτα 1433

Η πόρτα 1433 όπως αναφέρθηκε και παραπάνω χρησιμοποιείται από τον Microsoft SQL Server με σκοπό την απομακρυσμένη σύνδεση στις βάσεις δεδομένων. Προχωρώντας παρακάτω στην ανάλυση των πακέτων που ελήφθησαν βλέπουμε στο σχήμα 3.11 πως ο επιτιθέμενος προσπαθεί να επιτύχει σύνδεση στον διακομιστή χρησιμοποιώντας ως όνομα εισόδου τη λέξη “sa” και αφήνοντας κενό το συνθηματικό (γι'αυτό το λόγω δεν υπάρχει στο πακέτο). Η λέξη “sa” είναι ουσιαστικά συντομογραφία και αναφέρεται στον “system administrator”. Δηλαδή ο επιτιθέμενος προσπαθεί να εκμεταλλευτεί μία αδυναμία του MS-SQL-Server και να συνδεθεί ως διαχειριστής στον διακομιστή. Η εν λόγω αδυναμία αναφέρεται στο γεγονός ότι όλες οι εκδόσεις του Microsoft SQL Server κυκλοφορούσαν μέχρι την έκδοση 2000 με κενό συνθηματικό για τον λογαριασμό “sa”.




Σχήμα 3.11: Προσπάθεια για σύνδεση στον Microsoft SQL Server

Κοιτάζοντας επίσης τα αρχεία καταγραφής συμβάντων του p0f και τα οποία κατέγραψε το Honeywall μπορούμε να εξάγουμε περισσότερες πληροφορίες για τον επιτιθέμενο χρήστη. Όπως βλέπουμε στο σχήμα 3.12 παρακάτω, το σύστημα μέσω του οποίου πραγματοποιείται η επίθεση έχει εγκατεστημένο το λειτουργικό σύστημα Windows 2000 SP4 ή Windows XP SP1+. Επίσης όπως διακρίνουμε ο χρήστης έχει distance 19, δηλαδή απέχει από το σύστημα μας 19 ενδιάμεσους σταθμούς (hops).




Σχήμα 3.12: Αρχεία καταγραφής συμβάντων του p0f

Συνεχίζοντας και μελετώντας και υπόλοιπα δεδομένα τα οποία καταγράφηκαν βλέπουμε πως

αφού αποτύχει η προσπάθεια εισόδου χωρίς συνθηματικό, υπάρχουν εν συνεχεία και άλλες προσπάθειες για σύνδεση στο ίδιο σύστημα με τη χρήση αυτή τη φορά συνθηματικών. Οι προσπάθειες αυτές γίνονται ως administrator με συνθηματικά όπως sa, querty, abc123, κ.α. Λόγω του ότι οι προσπάθειες αυτές για σύνδεση επαναλαμβάνονται αρκετές φορές σε σύντομο χρονικό διάστημα καταλήγουμε στο συμπέρασμα πως μάλλον πρόκειται για κάποια επίθεση που έχει γίνει από κάποιο worm ή με τη χρήση κάποιου αυτοματοποιημένου εργαλείου αφού είναι αδύνατο ένας χρήστης να πληκτρολογήσει τόσο γρήγορα.

# ΚΕΦΑΛΑΙΟ 4 – ΕΙΣΑΓΩΓΙΚΑ ΣΤΗΝ ΥΠΗΡΕΣΙΑ DNS

Στο προηγούμενο κεφάλαιο ασχοληθήκαμε με τα honeypots χαμηλής αλληλεπίδρασης, πραγματοποιώντας πείραμα με τη βοήθεια του Honeyd. Στα κεφάλαια που ακολουθούν θα δώσουμε κυρίως έμφαση στα honeypots υψηλής αλληλεπίδρασης. Τα honeypots υψηλής αλληλεπίδρασης όπως έχουμε αναφέρει και στο πρώτο κεφάλαιο, είναι πραγματικά συστήματα, με πραγματικές υπηρεσίες και κάποια επιπλέον προγράμματα για την παρακολούθηση της προκύπτουσας κίνησης. Συνεπώς πολλές φορές τόσο η εγκατάσταση όσο και η ρύθμιση τους ίσως αποδειχθεί μια αρκετά δύσκολη και πολύπλοκη διαδικασία. Επίσης, λόγω του μεγάλου όγκου των δεδομένων, η ανάλυση τους αποτελεί και αυτή μια ιδιαίτερα χρονοβόρα και κοπιαστική εργασία. Καταλαβαίνει κανείς λοιπόν πως η εγκατάσταση αλλά κυρίως η συντήρηση υψηλής αλληλεπίδρασης honeypots δεν είναι καθόλου εύκολη υπόθεση.

Από την άλλη πλευρά τα μειονεκτήματα που προαναφέρθηκαν, αντισταθμίζονται από το γεγονός ότι μπορούμε να μελετήσουμε πραγματικές επιθέσεις σε ένα ελεγχόμενο περιβάλλον, χωρίς κάποιο κόστος για τα δεδομένα. Έτσι κατά αυτό τον τρόπο, μέσα από την τριβή και τη μελέτη με τέτοιου είδους συστήματα αποκτάται πολύτιμη εμπειρία, η οποία συμβάλει στην καλύτερη κατανόηση του προβλήματος της ασφάλειας υπολογιστών.

Παρόλα αυτά, το σκηνικό του ηλεκτρονικού εγκλήματος έχει αλλάξει. Πλέον η μεγάλη πλειονότητα των κακόβουλων χρηστών στοχεύει κυρίως στο εύκολο κέρδος. Δηλαδή δεν έχουν πια σκοπό τους  να δοκιμάσουν διάφορες τεχνικές επιθέσεων, για να ικανοποιήσουν απλώς την περιέργεια τους. Αντιθέτως στοχεύουν κυρίως μικρές ή μεγάλες εταιρίες, με καλά οικονομικά στοιχεία με σκοπό να αποκομίσουν εύκολο κέρδος. Επειδή τα συστήματα ενός honeynet όσο υψηλής αλληλεπίδρασης και να είναι δεν είναι σε θέση, μια και δεν περιέχουν αξιόλογα δεδομένα, να προκαλέσουν το ενδιαφέρον των ηλεκτρονικών εγκληματιών, δύσκολα θα επιλεχθεί ένα honeynet για κάτι παραπάνω από μια απλή σάρωση. Πραγματικά αυτό που συνήθως καταγράφουν τα σημερινά honeynets είναι σαρώσεις παντός είδους και επιθέσεις από αυτοματοποιημένα εργαλεία.

Αποφασίσαμε, λοιπόν, να μην δημιουργήσουμε απλώς κάποιο honeypot υψηλής

αλληλεπίδρασης και αφού το συνδέσουμε στο διαδίκτυο να δούμε τι επιθέσεις θα καταγράψει. Κάτι τέτοιο θα μπορούσε να αποδειχθεί άσκοπο καθώς οι πιθανότητες να μην καταγραφεί καμία αξιόλογη επίθεση ήταν αρκετά μεγάλες. Αντί γι' αυτό αποφασίσαμε να εκτελέσουμε στο εργαστήριο μια επίθεση η οποία εκμεταλλεύεται γνωστές αδυναμίες των διακομιστών DNS προκειμένου να καταδείξουμε την δύναμη αλλά και την ικανότητα που έχουν τα honeypots υψηλής αλληλεπίδρασης να καταγράφουν τέτοιου είδους επιθέσεις.

Οι διακομιστές DNS αποτελούν βασικά συστήματα της δομής και λειτουργίας του Διαδικτύου. Οι συνήθεις επιθέσεις που απειλούν την DNS υπηρεσία ήταν οι λεγόμενες cache poisoning[1]. Αν και οι επιθέσεις αυτού του είδους θεωρούνταν δύσκολο να επιτευχθούν, αυτό ανατράπηκε τον Ιούλιο του 2008 όταν στο συνέδριο Blackhats ανακοινώθηκε από τον ερευνητή Dan Kaminsky, ένα σημαντικό λάθος σχεδιασμού του πρωτοκόλλου DNS. Με βάση αυτό το λάθος στον σχεδιασμό του πρωτοκόλλου, ο Dan Kaminsky, απέδειξε πως ακολουθώντας την κατάλληλη διαδικασία οι επιθέσεις cache poisoning εναντίων των recursive DNS servers ήταν τελικά ζήτημα μερικών λεπτών να πραγματοποιηθούν με επιτυχία. Ασφαλώς αυτή η ανακάλυψη προκάλεσε μεγάλη αναστάτωση καθώς αυτομάτως σήμαινε πως εκατοντάδες διακομιστές DNS παγκοσμίως ήταν ευάλωτοι. Αυτή την επίθεση πραγματοποιήσαμε σε ένα ελεγχόμενο περιβάλλον στο Εργαστήριο ώστε να εξετάσουμε τις δυνατότητες των υψηλής αλληλεπίδρασης honeypots.

## Εισαγωγικά στους DNS Servers

Οι DNS servers είναι από τα πλέον βασικά και απαραίτητα συστήματα σχεδόν από τις αρχές του Internet. Ο ρόλος και η σημασία τους είναι αδιαμφισβήτητα, ενώ έχουν συμβάλει καθοριστικά στη μορφή που έχει σήμερα το Internet αλλά και στην ευκολία της χρήσης του. Μπορεί η πλειονότητα των χρηστών να αγνοεί ακόμα και την ύπαρξη τους, ωστόσο είναι σίγουρο πως ο καθένας τους ξεχωριστά τους έχει χρησιμοποιήσει αρκετές φορές. Για την ακρίβεια οποιοσδήποτε έχει επισκεφθεί κάποια ιστοσελίδα πληκτρολογώντας το URL της στην μπάρα του browser, είτε έχει αποστείλει κάποιο e-mail έχει εμμέσως χρησιμοποιήσει και κάποιον DNS server. Περισσότερες πληροφορίες για τι ακριβώς είναι οι DNS servers, πως

1 Βλέπε γλωσσάρι

λειτουργούν και τι ακριβώς διαδικασίες εκτελούν παρατίθενται παρακάτω.

**Τι είναι οι DNS servers;**

Όλες οι εφαρμογές που παρέχουν επικοινωνία μεταξύ δύο υπολογιστών στο Διαδίκτυο λειτουργούν με βάση το πρωτόκολλο IP. Το πρωτόκολλο IP χρησιμοποιεί τις διευθύνσεις IP (αριθμητικές τιμές από 0 έως 255 διαχωρισμένες με τελείες) για τον προσδιορισμό και εντοπισμό των χρηστών στο Διαδίκτυο. Με δεδομένη τη δυσκολία του ανθρώπου στην απομνημόνευση αριθμητικών τιμών, η επικοινωνία στο Διαδίκτυο θα ήταν αδύνατο να επιτευχθεί με τη χρήση των διευθύνσεων IP.

Για τη λύση του παραπάνω προβλήματος αποφασίστηκε η αντιστοίχιση ονομάτων, τα ονόματα χώρου (domain names), σε κάθε διεύθυνση IP. Τα ονόματα χώρου χρησιμοποιούνται στην θέση των διευθύνσεων IP κατά τον ίδιο ακριβώς τρόπο, επιτρέποντας την εύκολη επικοινωνία μεταξύ δύο υπολογιστών. Για να μπορέσει όμως να τεθεί σε εφαρμογή αυτή η λύση, χρειαζόταν μηχανισμός υπεύθυνος για την μετάφραση των διευθύνσεων IP σε Ονόματα χώρου και το αντίστροφο. Έτσι περίπου στα μισά της δεκαετίας του 1970 γεννήθηκε η DNS υπηρεσία.

Το Domain Name System (DNS) είναι ένα ιεραρχικό σύστημα ονομάτων που εξυπηρετεί τους υπολογιστές, τις υπηρεσίες αλλά και οποιοδήποτε άλλο πόρο συνδεδεμένο με το Διαδίκτυο ή κάποιο άλλο ιδιωτικό δίκτυο. Κύρια εργασία του είναι να συσχετίζει διάφορες πληροφορίες με ονόματα χώρου που έχουν αποδοθεί στους χρήστες του Διαδικτύου (οργανισμοί, εταιρίες, ιδιώτες κλπ.). Συγκεκριμένα μεταφράζει τα κατανοητά στους ανθρώπους ονόματα χώρου, στις αριθμητικές τιμές (διευθύνσεις IP) που χρησιμοποιούνται από τις δικτυακές συσκευές, με σκοπό τον εντοπισμό αυτών των συσκευών στον κόσμο.

Το Domain Name System κατανέμει την ευθύνη απόδοσης ονομάτων χώρου, αλλά και τη συσχέτιση των ονομάτων με διευθύνσεις IP, καθορίζοντας authoritative διακομιστές ως υπεύθυνους για τη διαχείριση ενός συγκεκριμένου κομματιού του ιεραρχικού χώρου. Οι authoritative διακομιστές έχουν τη δυνατότητα μικρότερα κομμάτια του χώρου ευθύνης τους να τα μεταβιβάσουν σε άλλους authoritative διακομιστές. Αυτός ο μηχανισμός μετέτρεψε την

υπηρεσία DNS σε κατανεμημένη και ανθεκτική σε περιπτώσεις σφαλμάτων ενώ απέτρεψε την ανάγκη σχηματισμού ενός κεντρικού φορέα διαχείρισης των ονομάτων ο οποίος και θα χρειαζόταν διαρκή ενημέρωση.

Αν θέλαμε να προσομοιάσουμε τους DNS servers με κάτι κοινό και γνωστό σε όλους θα μπορούσαμε να πούμε πως είναι κάτι παρεμφερές με τους τηλεφωνικούς καταλόγους, Αυτό γιατί οι DNS servers περιέχουν και αυτοί εγγραφές με πληροφορίες που είναι ευρέως γνωστές και στις οποίες μπορούμε να αποταθούμε όταν τις χρειαστούμε. Οι τηλεφωνικοί κατάλογοι διατηρούν εγγραφές για τα τηλεφωνικά νούμερα και τα ονόματα των κατόχων τους ενώ οι DNS servers για τα ονόματα χώρου και την διευθύνσεις IP στις οποίες αυτά αντιστοιχίζονται.

## Χαρακτηριστικά

Η λειτουργία των DNS servers είναι πιο σύνθετη από ότι μπορεί κάποιος να φαντάζεται και απαιτεί τη συνεργασία πολλών συστημάτων και υπηρεσιών.

### Ονόματα Χώρου (Domain Names)

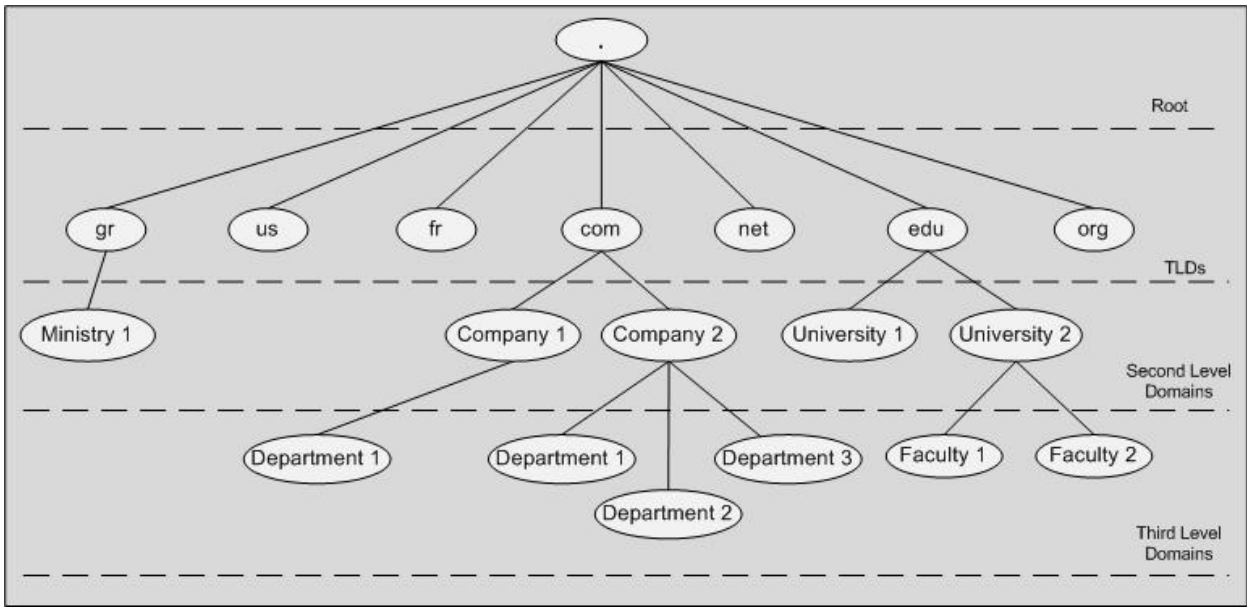
Ο χώρος ονομάτων του Διαδικτύου χωρίζεται σε τομείς ευθύνης που αντιστοιχούν στα “domains”. Είναι δυνατόν οι χώροι (domains) αυτοί να διαχωριστούν σε μικρότερους τομείς που καλούνται “subdomains”. Για παράδειγμα, στα επιμέρους τμήματα μιας εταιρείας που χρησιμοποιεί το domain “company.com”, θα μπορούσαν να αντιστοιχηθούν τα subdomains “department1.company.com” και “department2.company.com”. Σε κάθε υπολογιστή που ανήκει σε ένα από αυτά τμήματα θα δοθεί όνομα χώρου που θα περιέχει το subdomain του αντίστοιχου τμήματος, για παράδειγμα “host1.department2.example.com”.

Τα ονόματα χώρου αποτελούνται από ακολουθίες χαρακτήρων οι οποίοι διαχωρίζονται μεταξύ τους με τελείες. Κάθε τελεία αντιπροσωπεύει την αλλαγή ευθύνης στη διαχείριση του χώρου ονομάτων. Η επεξεργασία του ονόματος γίνεται από τα δεξιά προς τα αριστερά και το όνομα καταλήγει σε τελεία που αντιπροσωπεύει την ανώτερη αρχή διαχείρισης, το “root domain”, και συχνά παραλείπεται. Τα domains που ορίζονται στο root domain καλούνται Top

Level Domains (TLDs).

**Ιεραρχία**

Το DNS σύστημα ακολουθεί την ιεραρχία που περιγράφεται στο σχήμα 4.1.




Σχήμα 4.1: Ιεραρχία ονομάτων χώρου

Για κάθε domain ορίζεται συγκεκριμένο σύνολο διακομιστών DNS υπεύθυνο για την αντιστοίχιση ονομάτων χώρου σε IP διευθύνσεις. Οι διακομιστές υπεύθυνοι για το root domain ονομάζονται “Root (Name) Servers”, ενώ οι διακομιστές υπεύθυνοι για τα TLD ονομάζονται “TLD Servers”.

Όπως είπαμε παραπάνω ένα όνομα χώρου αναλύεται από τους διακομιστές DNS από τα δεξιά προς τα αριστερά. Η ανάλυση αυτή έχει σκοπό την επίλυση δηλαδή την αντιστοίχιση ονόματος σε IP διεύθυνση και γίνεται βήμα-βήμα ξεκινώντας από την πρώτη τελεία δηλαδή το επίπεδο “root” και συνεχίζοντας στα υπόλοιπα έως ότου επιλυθεί ολόκληρο. Η επεξεργασία, δηλαδή, γίνεται από το μεγαλύτερο προς το πιο συγκεκριμένο χώρο. Για παράδειγμα, έστω πως έχουμε το όνομα χώρου “www.ds.unipi.gr”. Το όνομα χώρου αυτό χωρίζεται σε πέντε μέρη. Τα μέρη αυτά είναι τα εξής:

1. “**.**”: Η τελεία όπως είπαμε παραπάνω προστίθενται στο τέλος κάθε ονόματος χώρου, αυτόματα, χωρίς ο χρήστης να χρειαστεί να την προσθέσει. Αντιπροσωπεύει το επίπεδο “root” την ανώτερη αρχή δηλαδή με βάση το πρωτόκολλο DNS .
2. “**gr**”: Το “gr” είναι ένα TLD, ανήκει στο πρώτο επίπεδο ονομάτων και αποτελεί συντομογραφία της αγγλικής λέξης Greece (Ελλάδα). Το συγκεκριμένο TLD προσδιορίζει δηλαδή πως η υπηρεσία την οποία θέλουμε να προσπελάσουμε βρίσκεται γεωγραφικά στην Ελλάδα.
3. **“unipi”:** Το “unipi” ανήκει στο δεύτερο επίπεδο ονομάτων και αποτελεί συντομογραφία του “University of Piraeus”. Καθορίζει δηλαδή πως το όνομα χώρου το οποίο εξετάζουμε ανήκει στο Πανεπιστήμιο Πειραιά το οποίο γεωγραφικά βρίσκεται στην Ελλάδα.
4. **“ds”:** Το “ds” ανήκει στο τρίτο επίπεδο ονομάτων και αποτελεί συντομογραφία του “Digital Systems”. Το ds καθορίζει δηλαδή πως το όνομα χώρου το οποίο εξετάζουμε ανήκει στο τμήμα Ψηφιακών Συστημάτων, του Πανεπιστημίου Πειραιώς, το οποίο γεωγραφικά βρίσκεται στην Ελλάδα.
5. **“www”:** Το “www” αποτελεί συντομογραφία του “World Wide Web” και καθορίζει πως η υπηρεσία που θέλουμε να προσπελάσουμε είναι προσβάσιμη μέσω ενός διακομιστή Διαδικτύου.

Όπως βλέπουμε παραπάνω η ανάλυση ενός ονόματος χώρου ξεκινάει από κάτι το γενικό, όπως είναι μια χώρα (Ελλάδα στη συγκεκριμένη περίπτωση) και καταλήγει σε κάτι το ειδικό που είναι ο διακομιστής Διαδικτύου μέσω του οποίου γίνεται προσβάσιμη η υπηρεσία.

**TLD (Top Level Domains)**

Τα TLD τα διαχειρίζονται οι διακομιστές TLD οι οποίοι βρίσκονται δεύτεροι στην ιεραρχία. Τα TLD χωρίζονται σε δύο κατηγορίες:

1. Τα Generic Top Level Domains (gTLD) τα οποία χρησιμοποιούνται για να προσδιορίσουν το αντικείμενο εργασιών του οργανισμού ή της εταιρείας στην οποία ανήκουν. Παραδείγματα gTLD είναι τα .com, .edu, .net, .org τα οποία απορρέουν από τα commercial, education, network και organization αντίστοιχα.

2. Τα Country Code Top Level Domains (ccTLD) τα οποία χρησιμοποιούνται για να

προσδιορίσουν τη χώρα στην οποία βρίσκονται οι οργανισμοί ή εταιρείες στις οποίες ανήκουν. Παραδείγματα ccTLD είναι τα .gr, .fr, .uk, .us τα οποία απορρέουν από τα Greece, France, United Kingdom, United States of America αντίστοιχα.

**Ερωτήματα**

Η λειτουργία των διακομιστών DNS βασίζεται στην εξυπηρέτηση των ερωτημάτων. Η βασική εργασία της DNS υπηρεσίας είναι η αποστολή ερωτημάτων από την πλευρά του χρήστη (πελάτη) και η επίλυση τους από την πλευρά των διακομιστών. Τα ερωτήματα αυτά αποστέλονται από την εκάστοτε εφαρμογή, η οποία επιθυμεί να μάθει την διεύθυνση IP που αντιστοιχίζεται σε κάποιο όνομα χώρου και στέλνονται προς επίλυση ακολουθώντας την υπάρχουσα ιεραρχία. Οι διακομιστές DNS εν συνεχεία αποστέλλουν απαντήσεις στα ερωτήματα που έλαβαν επιστρέφοντας την διεύθυνση IP που ζητήθηκε, αν είναι υπεύθυνοι για το όνομα χώρου, ή αν δεν είναι, ανακατευθύνουν την εφαρμογή στον υπεύθυνο διακομιστή. Τα ερωτήματα ακολουθούν κάποια συγκεκριμένη δομή που ορίζεται από το πρωτόκολλό DNS. Παρακάτω θα δούμε παραδείγματα DNS ερωτημάτων.

**Ζώνες**

Οι ζώνες, όπως και η σχέση που έχουν με τα ονόματα χώρου, είναι έννοιες που συχνά δυσκολεύονται να τις αντιληφθούν οι περισσότεροι χρήστες. Τα ονόματα χώρου όπως είδαμε και πιο πάνω ακολουθούν μια συγκεκριμένη δομή και χωρίζονται με τελείες σε μέρη, τα οποία εξυπηρετούνται από διαφορετικούς διακομιστές. Σύμφωνα με τον ορισμό λοιπόν:

*«Ως ζώνη καθορίζεται το κομμάτι εκείνο ενός ονόματος χώρου για το οποίο ένας διακομιστής ονομάτων έχει ακριβείς πληροφορίες και είναι υπεύθυνος (authoritative) για αυτό. »*

Οι πληροφορίες σχετικά με την αντιστοίχιση ονομάτων σε IP και γενικά οποιαδήποτε άλλη πληροφορία αφορά σε ονόματα χώρου (π.χ. λίστα διακομιστών ηλεκτρονικού ταχυδρομείου για ένα domain), εμφανίζεται μέσα σε μία ζώνη υπό τη μορφή εγγραφών (Resource Records – RR). Θα δούμε παρακάτω πιο αναλυτικά παραδείγματα.

Εκτός από τις τυπικές ζώνες στις οποίες καθορίζονται τα διάφορα ονόματα χώρου, υπάρχουν και ειδικές περιπτώσεις ζωνών οι οποίες έχουν αποκτήσει συγκεκριμένη ονομασία

λόγω της φύσης της εργασίας την οποία εκτελούν. Μια από αυτές τις περιπτώσεις είναι η ζώνη "hint" η οποία βρίσκεται σε κάθε διακομιστή και περιέχει λίστα με όλους τους Root Servers.

## Τύποι DNS servers

Οι DNS servers κατηγοριοποιούνται σε διάφορους τύπους, αναλόγως με τις εργασίες τις οποίες εκτελούν αλλά και τον τρόπο τον οποίο τις εκτελούν. Οι βασικοί τύποι των DNS servers είναι τέσσερις και είναι οι εξής:

1. Authoritative DNS servers
2. Caching DNS servers
3. Recursive DNS servers
4. Forwarding DNS servers

**Authoritative DNS servers**

Authoritative ονομάζονται οι διακομιστές οι οποίοι είναι υπεύθυνοι για κάποιες ζώνες. Οι διακομιστές αυτοί δέχονται ερωτήματα και απαντάνε σε αυτά κατάλληλα, ανάλογα με τις εγγραφές οι οποίες υπάρχουν στις ζώνες. Οι authoritative DNS servers χωρίζονται περαιτέρω στους εξής δύο τύπους:

1. **Master (Primary) DNS servers:** Master DNS servers ονομάζονται οι authoritative διακομιστές οι οποίοι έχουν αποθηκευμένα τοπικά τα αρχεία της ζώνης ή των ζωνών για την οποία ή τις οποίες είναι υπεύθυνοι. Οι master DNS servers έχουν την δυνατότητα να διαμοιράσουν αντίγραφα των ζωνών αυτών στους slave DNS servers μέσω της διαδικασίας "zone transfer". Σε περίπτωση που χρειαστεί να πραγματοποιηθούν αλλαγές στις ζώνες αυτές, τροποποιούνται τα αρχεία των master DNS servers οι οποίοι εν συνεχεία ενημερώνουν τους slave DNS servers. Οι master DNS servers θεωρούνται πιο αξιόπιστοι σε σχέση με τους slave DNS servers αφού πάντα περιέχουν την πιο πρόσφατη πληροφορία.

2. **Slave (Secondary) DNS servers:** Slave DNS servers' ονομάζονται οι authoritative

διακομιστές οι οποίοι λαμβάνουν μονάχα αντίγραφα των ζωνών των οποίων είναι υπεύθυνοι και δεν έχουν πρόσβαση στα πραγματικά αρχεία. Ουσιαστικά αποτελούν εφεδρικά μηχανήματα των master DNS servers από τους οποίους και λαμβάνουν τα αντίγραφα των ζωνών. Οι slave DNS servers ενημερώνονται αυτόματα από τους master DNS servers για τις οποιεσδήποτε αλλαγές πραγματοποιηθούν στις ζώνες, μέσω της διαδικασίας zone transfer, με την προϋπόθεση να αλλάξει το serial number που υπάρχει σε αυτές. Αν πραγματοποιηθούν αλλαγές στις ζώνες ενός Master DNS server αλλά το serial number παραμείνει το ίδιο τότε master διακομιστής δεν θα ενημερώσει τον Slave θεωρώντας πως δεν πραγματοποιήθηκαν αλλαγές.

Ειδικές κατηγορίες authoritative DNS servers αποτελούν επίσης και οι παρακάτω τύποι. Έχουν αποκτήσει ειδική ονομασία λόγω της συγκεκριμένης εργασίας που εκτελούν. Οι τύποι αυτοί είναι οι εξής:

- **Root (name) servers:** Η επίλυση όλων των ερωτημάτων ξεκινάει πάντα από αυτούς τους servers. Οι root DNS servers είναι authoritative για την ζώνη "." μέσα στην οποία υπάρχουν οι εγγραφές για τους διακομιστές οι οποίοι διαχειρίζονται τα TLDs (Top Level Domains). Η ζώνη αυτή είναι προκαθορισμένη, περιέχει εγγραφές για όλα τα TLDs και τροποποιείται λιγότερο συχνά σε σχέση με τις υπόλοιπες ζώνες. Το περιεχόμενο της root ζώνης καθορίζεται από την IANA (Internet Assigned Numbers Authority). Αυτή τη στιγμή υπάρχουν εκατοντάδες [19] root name servers ανά τον κόσμο, οι οποίοι όλοι τους φέρουν για ονόματα ένα γράμμα από το A έως το M.

- **TLD Servers:** Οι TLD servers έχουν εγγραφές για τους servers και διαχειρίζονται τα διάφορα ονόματα χώρου που έχουν κατάληξη το συγκεκριμένο TLD. Όταν λοιπόν οι TLD servers λάβουν κάποιο ερώτημα σε σχέση με τα domains που διαχειρίζονται, ανακατευθύνουν τον αποστολέα της ερώτησης στον κατάλληλο authoritative server.

**Caching DNS servers**

Οι caching DNS servers έχουν την ιδιότητα να κρατάνε σε μια προσωρινή μνήμη (την μνήμη

cache) μεμονωμένες εγγραφές, για ένα χρονικό διάστημα (TTL), το οποίο ορίζεται μέσα στις ζώνες που τις περιέχουν. Όταν δεχθούν κάποιο ερώτημα οι caching DNS servers θα ψάξουν στην μνήμη cache για να δουν αν κάποια εγγραφή περιέχει την κατάλληλη απάντηση. Αν δεν βρουν την κατάλληλη απάντηση θα επιστρέψουν ένα μήνυμα λάθους (nxdomain). Όταν το TTL «λήξει» οι caching DNS servers απομακρύνουν την συγκεκριμένη έγγραφή από την μνήμη cache.

**Recursive DNS servers**

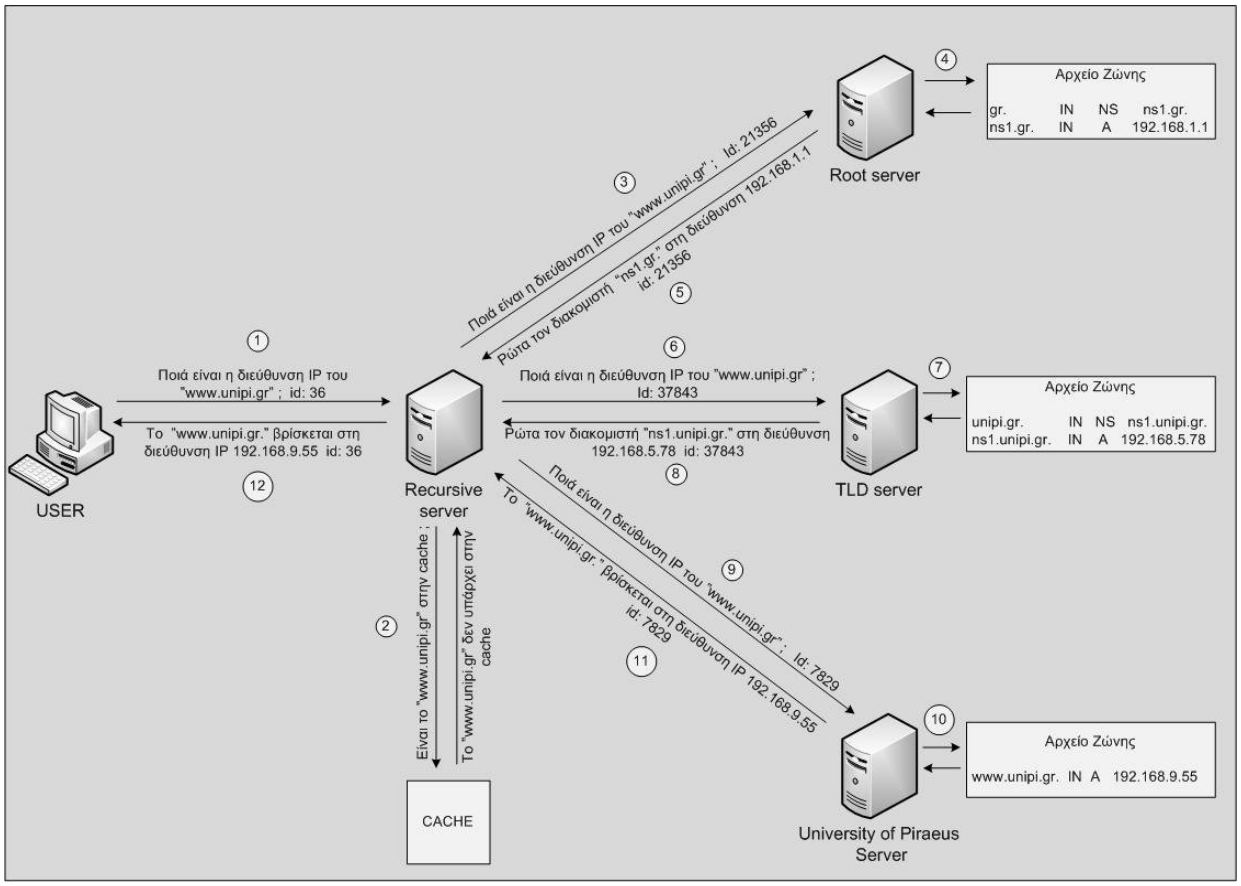
Οι recursive DNS servers είναι διακομιστές οι οποίοι αναλαμβάνουν να επιλύσουν το οποιοδήποτε ερώτημα τους ανατεθεί από κάποια εφαρμογή. Οι recursive DNS servers θα επιλύσουν τα ερωτήματα ακολουθώντας την υπάρχουσα ιεραρχία πραγματοποιώντας και αυτοί τα κατάλληλα ερωτήματα. Τέλος, επιστρέφουν στην εφαρμογή η οποία πραγματοποίησε το ερώτημα την τελική απάντηση την οποία έλαβαν. Οι recursive DNS servers λειτουργούν και ως caching ταυτοχρόνως. Είναι σύνηθες, επίσης, πολλοί name servers να λειτουργούν ως authoritative για κάποια ζώνη και ως recursive για όλα τα υπόλοιπα ερωτήματα. Η επίλυση όλων ερωτημάτων στο Διαδίκτυο γίνεται από τους recursive name servers.

**Forwarding (Proxy) DNS servers**

Οι Forwarding DNS servers είναι διακομιστές ονομάτων οι οποίοι προωθούν όλα τα ερωτήματα τα οποία λαμβάνουν σε κάποιους recursive διακομιστές ονομάτων και αποθηκεύουν εν συνεχεία τα αποτελέσματα στην προσωρινή μνήμη. Με αυτόν τον τρόπο βοηθούν εξυπηρετούν τα ερωτήματα χωρίς να προσθέτουν επιπλέον κίνηση στο τοπικό δίκτυο.

## Τρόπος Λειτουργίας

Παρακάτω θα αναλύσουμε τον τρόπο λειτουργίας των διακομιστών ονομάτων. Στο σχήμα 4.2 βλέπουμε μια αναλυτική περιγραφή του τρόπου λειτουργίας βήμα προς βήμα.




Σχήμα 4.2: Ιεραρχία ενεργειών κατά την επίλυση ενός ερωτήματος.

Τα βήματα που υπάρχουν στο παραπάνω σχήμα επεξηγούνται παρακάτω:

- **1:** Στο πρώτο βήμα πραγματοποιείται ένα ερώτημα από μία εφαρμογή που χρησιμοποιεί ο χρήστης για το όνομα χώρου του Πανεπιστημίου Πειραιώς, "www.unipi.gr", με αναγνωριστικό (id) τον αριθμό 36. Το ερώτημα αποστέλλεται σε έναν recursive διακομιστή ονομάτων.
- **2:** Ο recursive διακομιστής ονομάτων λαμβάνει το ερώτημα και εξετάζει πρώτα αν υπάρχει κάποια εγγραφή για το συγκεκριμένο όνομα χώρου στην μνήμη cache του. Αν δεν υπάρχει κάποια εγγραφή στην μνήμη cache συνεχίζει στο βήμα τρία. Σε αντίθετη περίπτωση περνάει στο βήμα δώδεκα.
- **3:** Στο βήμα τρία ο recursive διακομιστής, εφόσον δεν βρήκε την πληροφορία που αναζητούσε στην μνήμη cache, πραγματοποιεί ερώτημα, για το όνομα χώρου

“www.unipi.gr.” προς έναν root διακομιστή τοποθετώντας ως αναγνωριστικό τον τυχαίο αριθμό 21356.

- **4:** Εν συνεχεία στο βήμα τέσσερα ο root διακομιστής εξετάζει τις ζώνες για τις οποίες είναι authoritative με σκοπό να εντοπίσει τις κατάλληλες εγγραφές για το TLD του ονόματος χώρου για το οποίο δέχθηκε το ερώτημα. Σε περίπτωση που δεν βρεθεί κάποια εγγραφή για το συγκεκριμένο TLD ο root διακομιστής θα αποκριθεί με ένα μήνυμα μη εύρεσης (NXDOMAIN).
- **5:** Στο βήμα πέντε ο root διακομιστής αποκρίνεται στο ερώτημα που έλαβε, προτρέποντας τον recursive διακομιστή να απευθυνθεί στον κατάλληλο TLD server. Στην απάντηση τοποθετεί το ίδιο αναγνωριστικό το οποίο περιείχε και το ερώτημα (21356).
- **6:** Ο recursive διακομιστής αφού λάβει την κατάλληλη απάντηση θα πραγματοποιήσει, στο βήμα έξι, ερώτημα προς τον TLD server ρωτώντας ξανά την διεύθυνση IP του ονόματος χώρου “www.unipi.gr.” και τοποθετώντας ως αναγνωριστικό τον τυχαίο αριθμό 37843.
- **7:** Στο βήμα επτά ο TLD διακομιστής ελέγχει τα αρχεία των ζωνών για τις οποίες είναι authoritative με σκοπό να βρει την κατάλληλη εγγραφή. Αν δεν υπάρχει αντίστοιχη εγγραφή για το ερώτημα που πραγματοποιήθηκε, ο TLD διακομιστής θα αποκριθεί με ένα μήνυμα μη εύρεσης (NXDOMAIN).
- **8:** Στο βήμα οκτώ ο TLD διακομιστής επιστρέφει στον recursive διακομιστή την απάντηση στο ερώτημα που του έθεσε, προτρέποντας τον να απευθυνθεί στον διακομιστή ονομάτων του Πανεπιστημίου, τοποθετώντας το ίδιο αναγνωριστικό με αυτό που έλαβε στο ερώτημα (37843).
- **9:** Στο βήμα εννέα ο recursive διακομιστής πραγματοποιεί ερώτημα στον διακομιστή ονομάτων του Πανεπιστημίου ζητώντας του την επίλυση του ονόματος χώρου “www.unipi..gr.”.
- **10:** Ο διακομιστής ονομάτων του Πανεπιστημίου, στο βήμα δέκα, πραγματοποιεί αναζήτηση στις ζώνες στις οποίες είναι authoritative με σκοπό να βρει την κατάλληλη εγγραφή. Σε περίπτωση που δεν υπάρχει αντίστοιχη εγγραφή για το ερώτημα που πραγματοποιήθηκε ο διακομιστής θα αποκριθεί με ένα μήνυμα μη εύρεσης

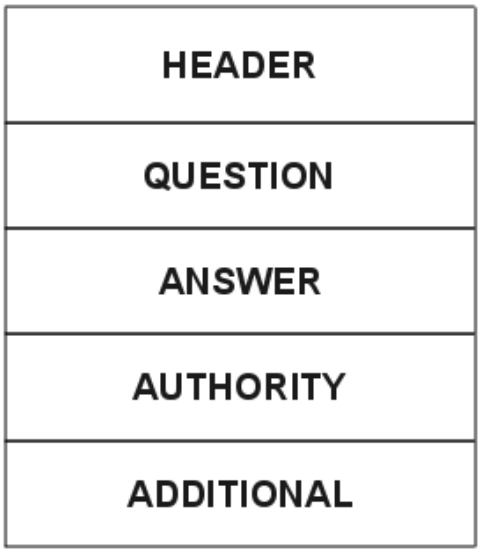
(NXDOMAIN).

- **11:** Στο βήμα έντεκα ο διακομιστής ονομάτων του Πανεπιστημίου επιστρέφει στον recursive διακομιστή την διεύθυνση IP στην οποία μπορεί να προσπελάσει την ιστοσελίδα του Πανεπιστήμιου.
- **12:** Τέλος, στο βήμα δώδεκα, ο recursive διακομιστής επιστρέφει στην εφαρμογή του χρήστη την απάντηση στο ερώτημα που έθεσε τοποθετώντας το ίδιο αναγνωριστικό με αυτό που έλαβε (36).

## Δομή χαρακτηριστικών υπηρεσίας

### Μορφή Μηνυμάτων

Τα μηνύματα που ανταλλάσσονται μεταξύ των DNS διακομιστών αλλά και μεταξύ των DNS διακομιστών και των χρηστών ακολουθούν μία συγκεκριμένη μορφολογία. Τα μηνύματα αυτά ανταλλάσσονται διαμέσου του πρωτοκόλλου UDP. Αυτό συμβαίνει καθώς το πρωτόκολλο DNS δεν έχει δημιουργηθεί έτσι ώστε να κάνει χρήση των υπηρεσιών του πρωτοκόλλου TCP. Επιπλέον οι συγκεκριμένες υπηρεσίες του TCP καθώς και αυξημένος όγκος της κεφαλίδας του θα πρόσθεταν αρκετή καθυστέρηση στην επικοινωνία κάτι που φυσικά δεν είναι επιθυμητό. Αντίθετα το TCP είναι απαραίτητο για την πραγματοποίηση της διαδικασίας zone transfer μεταξύ των master και authoritative διακομιστών με σκοπό την αξιόπιστη και ακέραια μεταφορά των δεδομένων εφόσον δεν μας προβληματίζει ο χρόνος διεκπεραίωσης της διαδικασίας. Στο σχήμα 4.3 βλέπουμε τα κύρια μέρη από τα οποία αποτελείται ένα DNS ερώτημα.



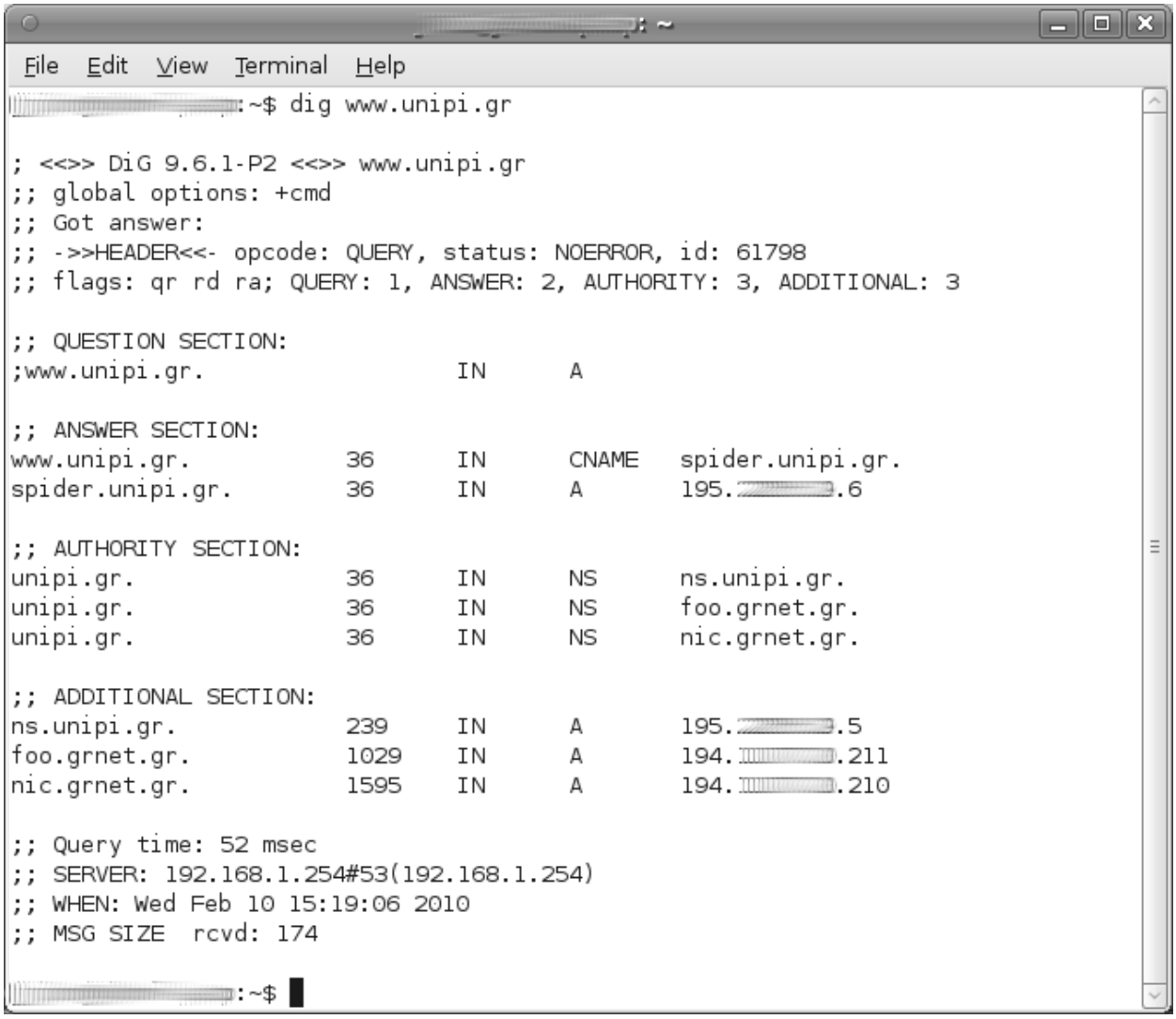

Σχήμα 4.3: Δομή ενός DNS ερωτήματος

Τα παραπάνω μέρη επεξηγούνται παρακάτω:

- **Header (Κεφαλίδα):** Η κεφαλίδα ενός DNS ερωτήματος έχει συνολικό μέγεθος δώδεκα bytes. Τα πρώτα δύο byte περιέχουν ένα μοναδικό αναγνωριστικό (query id), το οποίο χρησιμοποιείται για να συσχετισθούν οι απαντήσεις με τα ερωτήματα. Με βάση δηλαδή αυτό το μοναδικό αναγνωριστικό, καθορίζει ποια απάντηση προορίζεται για ποιο ερώτημα αφού και τα δύο φέρουν το ίδιο. Τα υπόλοιπα bytes της κεφαλίδας χρησιμοποιούνται για τον καθορισμό άλλων μεταβλητών όπως μηνύματα λάθους, τύπος απαντήσεων (authoritative ή μη) κ.α..
- **Question (Ερώτημα):** Σε αυτό το κομμάτι προστίθεται το ερώτημα που τίθεται προς απάντηση από τους αρμόδιους διακομιστές ονομάτων.
- **Answer (Απάντηση):** Σε αυτό το κομμάτι προστίθεται η απάντηση στο ερώτημα που έθεσε κάποιος πελάτης. Για να γίνει δεκτό το μήνυμα της απάντησης από τον πελάτη θα πρέπει, όπως ειπώθηκε πιο πάνω, να περιέχει το ίδιο μοναδικό αναγνωριστικό με το ερώτημα.
- **Authority (Υπεύθυνος):** Σε αυτό το κομμάτι προστίθενται οι υπεύθυνοι (authoritative) διακομιστές ονομάτων για το όνομα χώρου για το οποίο πραγματοποιήθηκε το ερώτημα.
- **Additional (Επιπλέον):** Σε αυτό το κομμάτι προστίθενται όλες οι απαραίτητες πληροφορίες που θα χρειαστεί ο πελάτης προκειμένου να επικοινωνήσει με τους

διακομιστές ονομάτων που περιγράφονται στο authority κομμάτι. Προστίθενται δηλαδή οι διευθύνσεις IP των συγκεκριμένων διακομιστών ονομάτων. Επίσης σε αυτό το κομμάτι είναι δυνατό να προστεθεί και επιπλέον πληροφορία η οποία ίσως να μην είναι απαραίτητη αλλά ο υπεύθυνος διακομιστής ονομάτων θεωρεί ότι ο πελάτης ίσως χρειαστεί.

Παρακάτω στο σχήμα 4.4 βλέπουμε ένα DNS μήνυμα χρησιμοποιώντας την εφαρμογή dig.




Σχήμα 4.4: Ένα μήνυμα DNS όπως φαίνεται χρησιμοποιώντας την εφαρμογή dig

**Μορφή ζωνών**

Οι ζώνες ακολουθούν και αυτές μια συγκεκριμένη δομή και κάποιο συγκεκριμένο συντακτικό. Σε μία ζώνη καλούμαστε να καθορίσουμε αρκετές μεταβλητές, οι οποίες θα περιγραφούν με ακρίβεια παρακάτω. Παρακάτω βλέπουμε το παράδειγμα μίας ζώνης.

```
01  ;Αυτό είναι ένα σχόλιο
02  ;
03  ;
04  fakezone.gr.  IN   SOA    ns1.fakezone.gr. hostmaster.fakezone.gr(
05                             2009121801 ; Serial number
06                             28800      ; Refresh
07                             3600       ; Retry
09                             604800     ; Expire
08                             10800      ; Ncache TLL
11                             )
10
12  fakezone.gr.           IN     NS    ns1.fakezone.gr.
13  fakezone.gr.           IN     MX    mail.fakezone.gr.
14
15  ;Servers
16  ;-------
17  ns1.fakezone.gr.       IN     A         86400     192.168.1.25
18  ns1.fakezone.gr.       IN    AAAA       86400     2001:db8:0:1::1
19  mail.fakezone.gr.      IN     MX        3600      192.168.1.30
20
21  ;CNAME
22  ;-----
23  ns1                    IN   CNAME   ns1.fakezone.gr.
```

Στο παραπάνω κομμάτι βλέπουμε τον ορισμό της ζώνης «fakezone.gr». Σε κάθε γραμμή υπάρχει αρίθμηση ώστε να γίνεται καλύτερη κατανόηση των όσων επεξηγούνται.

Όπως φαίνεται από το παραπάνω παράδειγμα μια ζώνη αποτελείται από διάφορα μέρη. Πιο

σημαντική είναι η εγγραφή SOA η οποία είναι απαραίτητη σε κάθε ζώνη και καθορίζει τους authoritative διακομιστές. Επίσης πολύ σημαντικές είναι οι εγγραφές "NS" και "MX" οι οποίες αντιστοιχίζουν ονόματα χώρου με διακομιστές ονομάτων και διακομιστές ηλεκτρονικού ταχυδρομείου αντίστοιχα, όπως επίσης και οι εγγραφές A και AAAAA οι οποίες αντιστοιχίζουν ονόματα χώρου με διευθύνσεις IP εκδόσεως τέσσερα και έξι αντίστοιχα. Το παραπάνω παράδειγμα της ζώνης επεξηγείτε αναλυτικά παρακάτω:

- **Σχόλια:** Στην πρώτη γραμμή έχουμε ένα σχόλιο. Ότι ακολουθεί μετά από ένα ελληνικό ερωτηματικό (**;**) σε ένα αρχείο μιας ζώνης αποτελεί αυτομάτως σχόλιο. Δηλαδή ότι υπάρχει μετά το ελληνικό ερωτηματικό δεν θα διαβάζεται από το λογισμικό του DNS server και δεν θα λαμβάνεται υπόψιν.
- **TTL:** Το TTL είναι η χρονική μεταβλητή που προσδιορίζει το χρόνο που θα μείνει στην μνήμη cache ενός recursive διακομιστή η συγκεκριμένη εγγραφή (Resource Records) δίπλα στην οποία δηλώθηκε. Ο ορισμός του TTL σε κάθε εγγραφή είναι υποχρεωτικός. Παραδείγματα TTL μπορούμε να δούμε στις γραμμές 17, 18 και 19. Στις συγκεκριμένες γραμμές βλέπουμε πως έχουν οριστεί ως TTL οι χρονικές μεταβλητές 86400.

**Τύποι Εγγραφών (RR)**

Κάθε ζώνη περιέχει πολλά είδη εγγραφών κάθε ένα από τα οποία έχει και διαφορετικό ρόλο. Τα είδη των εγγραφών είναι τα εξής:

- **SOA:** Στην SOA (Start Of Authority) καθορίζουμε τα διάφορα χαρακτηριστικά της ζώνης. Σε αυτήν την εγγραφή ορίζεται ο master διακομιστή της ζώνης (ns1 στο παραπάνω παράδειγμα στη γραμμή 04) και υποχρεωτικά το e-mail του διαχειριστή της. Μέσα στην παρένθεση ορίζουμε τα εξής στοιχεία:
    - **Serial Number:** Στη γραμμή 6 ορίζουμε το serial number της ζώνης fakezone.gr. Κάθε ζώνη έχει το δικό της serial number, ώστε να είναι κατανοητό πότε γίνονται αλλαγές σε αυτή. Είναι καλή πρακτική το serial number να αποτελείται από την ημερομηνία της τροποποίησης της ζώνης και έναν διψήφιο αριθμό που υποδηλώνει πόσες φορές έχει υποστεί αλλαγές η ζώνη εκείνη την ημέρα. Στο παράδειγμα μας, στη γραμμή έξι το serial number 2009121801 υποδηλώνει πως η

ζώνη τροποποιήθηκε μια φορά στις 18-12-2009. Αν την τροποποιήσουμε και πάλι, μέσα στην ίδια ημέρα, θα έχουμε το 2009121802. Αυτό συμβαίνει για να αντιληφθεί ο master DNS server ότι πραγματοποιήθηκαν αλλαγές στη ζώνη και να ενημερώσει τους slave DNS servers μέσω του πρωτοκόλλου NOTIFY ώστε αυτοί να προχωρήσουν στη διαδικασία zone transfer. Ακόμα και αν το NOTIFY είναι απενεργοποιημένο ο έλεγχος για αλλαγές στη ζώνη γίνεται περιοδικά όπως ορίζεται στην μεταβλητή Refresh (βλέπε παρακάτω). Σε περίπτωση που το serial number παραμείνει το ίδιο, ο master DNS server δεν θα ενημερώσει τους slave με αποτέλεσμα αυτοί να συνεχίσουν να εξυπηρετούν τα ερωτήματα με βάση τις παλιές ρυθμίσεις των ζωνών.

- **Refresh:** Στην χρονική μεταβλητή refresh καθορίζουμε το πόσο συχνά θα ελέγχουν οι secondary DNS servers για αλλαγές στα δεδομένα της ζώνης. Αν κατά τη διάρκεια του ελέγχου ανακαλύψουν κάποιο serial number μεγαλύτερο από αυτό που υπάρχει στην δικιά τους ζώνη, τότε προχωράνε σε zone transfer. Στην γραμμή εφτά φαίνεται ένα παράδειγμα της χρονικής μεταβλητής refresh.
- **Retry:** Αν ο slave DNS server δεν μπορέσει να επικοινωνήσει με τον master server μετά το πέρας της χρονικής μεταβλητής refresh, τότε θα προσπαθεί να επικοινωνεί περιοδικά με τον master server σύμφωνα με τη χρονική μεταβλητή retry. Στην γραμμή οκτώ φαίνεται ένα παράδειγμα για τη χρονική μεταβλητή retry.
- **Expire:** Στην χρονική μεταβλητή expire, καθορίζεται ο χρόνος μετά τη λήξη του οποίου ο slave DNS server θα πάψει να παρέχει πληροφορίες για τη συγκεκριμένη ζώνη αν δεν έχει καταφέρει σε αυτό το διάστημα να επικοινωνήσει με τον master server. Είναι προφανές ότι για τη σωστή λειτουργία της υπηρεσίας θα πρέπει ο χρόνος που ορίζεται ως expire να είναι μεγαλύτερος από το χρόνο που ορίζεται ως refresh. Στην σειρά εννέα φαίνεται ένα παράδειγμα της χρονικής μεταβλητής expire.
- **Ncache TLL (Negative caching):** Στην μνήμη cache καταγράφονται τόσο οι εγγραφές που προέκυψαν από ερωτήσεις σε άλλους διακομιστές όσο και τα αποτελέσματα από ανύπαρκτες εγγραφές, δηλαδή ερωτήσεις για ονόματα χώρου

που πήραν αρνητική απάντηση - NXDOMAIN. Σε αυτή τη χρονική μεταβλητή καθορίζεται το χρονικό διάστημα το οποίο θα παραμείνει στην μνήμη cache μια NXDOMAIN απάντηση. Άμεσο αποτέλεσμα θα είναι ο συγκεκριμένος διακομιστής να απαντάει με μηνύματα μη εύρεσης, σε όλα τα ερωτήματα που λαμβάνει και αφορούν το συγκεκριμένο όνομα χώρου, έως το πέρας του Ncache TTL όπου θα πραγματοποιήσει και πάλι αναζήτηση.

- **NS:** Με την εγγραφή "NS" (Name Server) καθορίζονται οι authoritative διακομιστές για τη ζώνη αυτή. Για λόγους διαθεσιμότητας θα πρέπει να ορίζονται τουλάχιστον δύο DNS servers για κάθε ζώνη. Ο master server και τουλάχιστον ένας slave server.
- **MX:** Με την εγγραφή MX (Mail Exchange) καθορίζονται οι διακομιστές e-mail οι οποίοι εξυπηρετούν την ηλεκτρονική αλληλογραφία για το συγκεκριμένο domain.
- **CNAME:** Με την εγγραφή CNAME (Canonical Name) καθορίζουμε ένα ψευδώνυμο για κάποιο όνομα χώρου. Κατά αυτό τον τρόπο δημιουργούμε ευκολότερα ονόματα προς χρήση αντικαθιστώντας άλλα σύνθετα και αρκετά δύσκολα.
- **A:** Οι εγγραφές A πραγματοποιούν την αντιστοίχιση ονομάτων χώρου σε διευθύνσεις IP έκδοσης 4 (IPv4) .
- **AAAA:** Οι εγγραφές A πραγματοποιούν την αντιστοίχιση ονομάτων χώρου σε διευθύνσεις IP έκδοσης 6 (IPv6) .

## Ανάστροφες Ζώνες (Reverse Zones)

Οι ανάστροφες ζώνες χρησιμοποιούνται για τον καθορισμό ενός ονόματος χώρου όταν είναι γνωστή η διεύθυνση IP του. Εκτελούν δηλαδή ακριβώς τον αντίθετο ρόλο από ότι οι κανονικές ζώνες. Παρακάτω βλέπουμε το παράδειγμα μιας ανάστροφης ζώνης:

```
01  ;Αυτό είναι ένα σχόλιο
02  ;
03  ;
04  1.168.192.in-addr.arpa. IN  SOA ns1.fakezone.gr. root.fakezone.gr(
05                               2009121801 ; Serial number
06                               28800      ; Refresh
```

```
07                              3600       ; Retry
09                              604800     ; Expire
08                              10800      ; Ncache TLL
11                              )
10
12  1.168.192.in-addr.arpa.     IN     NS    ns1.fakezone.gr.
13
14  ;Pointers
15  ;-------
16  25                      IN      PTR       86400     fakeschool.gr.
17  30                      IN      PTR       86400  mail.fakeschool.gr.
```

**Τύποι Εγγραφών (RR) Reverse Ζωνών**

Οι κύριοι τύποι εγγραφών περιγράφηκαν πιο πάνω και δεν έχουν κάποια διαφορά για τις reverse ζώνες. Επιπλέον στις ανάστροφες, όχι όμως αποκλειστικά[1], εμφανίζεται η εγγραφή PTR η οποία επεξηγείται παρακάτω:

- **PTR:** Με τη χρήση αυτής της εγγραφής αντιστοιχίζεται ο τελικός χρήστης ή υπηρεσία με ένα όνομα χώρου σύμφωνα **ανάστροφη αντιστοίχιση** στο πρωτόκολλο DNS.

## BIND

Το BIND (Berkeley Internet Name Protocol) είναι ένα πρόγραμμα ανοικτού κώδικα για την υλοποίηση του DNS πρωτοκόλλου, το οποίο δημιουργήθηκε αρχικά στο Πανεπιστήμιο του Berkeley της Καλιφόρνιας από που και πήρε του όνομα του. Πλέον συντηρείται από τον μη κερδοσκοπικό οργανισμό ISC (Internet Systems Consortium). Το BIND χάρις την υψηλή

1 Στα πρότυπα που περιγράφουν το DNS δεν ορίζεται ποιες εγγραφές εμφανίζονται στις κανονικές και ανάστροφες ζώνες. Είναι δυνατόν PTR εγγραφές να εμφανίζονται σε κανονικές ζώνες και A εγγραφές να εμφανίζονται σε ανάστροφες ζώνες.

ποιότητα του, την σταθερότητα του, αλλά και το μηδαμινό του κόστος είναι το πλέον διαδεδομένο λογισμικό DNS παγκοσμίως.

Η εγκατάσταση του BIND είναι μια αρκετά απλή διαδικασία. Σε πολλές διανομές του Linux, όπως για παράδειγμα το Ubuntu, το BIND είναι εύκολα διαθέσιμο μέσω της αποθήκης λογισμικού (Package Repository) που αυτό διαθέτει. Σε περίπτωση που το BIND δεν είναι διαθέσιμο μέσα από την αποθήκη λογισμικού θα πρέπει να γίνει η εγκατάσταση του χρησιμοποιώντας τον πηγαίο κώδικα. Το BIND διατίθεται επίσης και σε έκδοση για τα λειτουργικά συστήματα Microsoft Windows παρόλο που ξεκίνησε ως μια εφαρμογή μονάχα για λειτουργικά συστήματα τύπου UNIX. Η πιο πρόσφατη σταθερή έκδοση του BIND, την στιγμή που γραφόταν αυτή η εργασία, ήταν η 9.7.0 με ημερομηνία κυκλοφορίας την 16 Φεβρουαρίου 2010.

Η ρύθμιση του BIND γίνεται μέσω ενός συγκεκριμένου αρχείου παραμετροποίησης. Το αρχείο παραμετροποίησης του BIND είναι το "named.conf". Σε αυτό το αρχείο καθορίζονται οι ζώνες, όπως και οι authoritative και slave διακομιστές για τις ζώνες αυτές. Επίσης σε αυτό το αρχείο καθορίζονται και οι ρυθμίσεις του BIND όπως για παράδειγμα για το αν θα εξυπηρετεί τα recursive ερωτήματα. Στις νεότερες εκδόσεις του BIND το αρχείο παραμετροποίησης διαιρείται σε περισσότερα από ένα με σκοπό την ομαδοποίηση όμοιων χαρακτηριστικών.

**Ζώνες στο BIND**

Παρακάτω θα εξηγήσουμε τις διαφορές που υπάρχουν ανάμεσα στις ζώνες που ορίζονται σύμφωνα με το πρότυπο και σε αυτές που ορίζονται στο Bind. Παρακάτω βλέπουμε την ίδια ζώνη που ορίστηκε πιο πάνω και πως αυτή αντίστοιχα ορίζεται στο Bind.

```
01  ;Αυτό είναι ένα σχόλιο
02  $ORIGIN fakezone.gr.
03  $TTL   86400
04  ;
05  @   IN       SOA    ns1.fakezone.gr. hostmaster.fakezone.gr(
06               2009121801 ; Serial number
```

```
07              28800      ; Refresh
08              3600       ; Retry
09              604800     ; Expire
10              10800      ; Ncache TLL
11              )
12
13  @      IN      NS    ns1.fakezone.gr.
14  @      IN      MX    mail.fakezone.gr.
15
16 ;Servers
17   ;-------
18  ns1          IN      A        192.168.1.25
19  ns1          IN     AAAA      2001:db8:0:1::1
20  mail         IN      A        192.168.1.30
21
22  ;CNAME
23  ;-----
24  ns1          IN    CNAME   ns1.fakezone.gr.
```

Όπως βλέπουμε και στο παράδειγμα παραπάνω ο ορισμός μιας ζώνης στο Bind έχει μικρές διαφορές από το πρότυπο. Οι διαφορές αυτές αναλύονται παρακάτω:

- **$ORIGIN:** Σε αυτήν την μεταβλητή καθορίζεται το όνομα χώρου της ζώνης. Αν έχει οριστεί η μεταβλητή "$ORIGIN", τα σημεία στα οποία χρειάζεται να συμπληρωθεί το όνομα χώρου μπορούν να αντικατασταθούν από το χαρακτήρα @. Η χρήση αυτής της μεταβλητής είναι προαιρετική και γίνεται κυρίως για λόγους ευκολίας.
- **$TTL:** Σε αυτή τη μεταβλητή τοποθετείτε το TTL. Ο χρόνος δηλαδή τον οποίο θα παραμείνουν στην μνήμη cache των recursive διακομιστών. Αυτό ισχύει για όλες τις εγγραφές μέσα σε μία ζώνη του Bind αν δεν έχει οριστεί κάτι διαφορετικό (αν δεν έχει οριστεί άλλη χρονική μεταβλητή δίπλα από τις ζώνες).

Οι παραπάνω τύποι ισχύουν και για τις "reverse" ζώνες.

## Metasploit Framework

Το Metasploit framework είναι μια πλατφόρμα ανοικτού κώδικα η οποία περιέχει πολλά εργαλεία για την πραγματοποίηση δικτυακών επιθέσεων διαφόρων τύπων. Το Metasploit framework χρησιμοποιείται ως ένα εργαλείο για την πραγματοποίηση penetration testing[1], κυρίως από επαγγελματίες της ασφάλειας υπολογιστών με σκοπό την εύρεση των αδυναμιών σε συστήματα. Αν και ο σκοπός του είναι κυρίως εκπαιδευτικός , η πλατφόρμα χρησιμοποιείται ενίοτε και από κακόβουλους χρήστες για την πραγματοποίηση επιθέσεων. Συγγραφέας του Metasploit framework είναι ο H. D. Moore και η πρώτη έκδοση του κυκλοφόρησε τον Ιούλιο του 2003. Η πιο πρόσφατη έκδοση του Metasploit framework, μέχρι τη στιγμή που γραφόταν αυτή εδώ η εργασία, είναι η 3.3.3 με ημερομηνία κυκλοφορίας στις 23 Δεκεμβρίου 2009.

1 Βλέπε γλωσσάρι

# ΚΕΦΑΛΑΙΟ 5 - ΠΕΙΡΑΜΑ ΜΕ ΥΨΗΛΗΣ ΑΛΛΗΛΕΠΙΔΡΑΣΗΣ DNS HONEYPOTS

Όπως προαναφέρθηκε και πιο πάνω εκτελέσαμε ένα πείραμα στο Εργαστήριο, εκτελώντας επίθεση που εκμεταλλεύεται κενά ασφαλείας στους διακομιστές ονομάτων με σκοπό την μελέτη των υψηλής αλληλεπίδρασης honeypots. Παρακάτω παρατίθενται πληροφορίες για την επίθεση καθώς και για τα συστήματα που χρησιμοποιήθηκαν στο Εργαστήριο προκειμένου αυτή να επιτευχθεί.

**Τι είναι το "DNS Insufficient Socket Entropy Vulnerability" (Kaminsky bug);**

Το Kaminsky bug αποτελεί μια ακόμα διαδικασία για την πραγματοποίηση επιθέσεων τύπου DNS cache poisoning. Αποτέλεσε, μέχρι τη στιγμή που γραφόταν αυτή εδώ η εργασία, την επίθεση με το μεγαλύτερο αντίκτυπο στους DNS διακομιστές αποκαλύπτοντας παράλληλα ένα πολύ σημαντικό κενό ασφαλείας σε αυτούς, που υπήρχε για χρόνια. Τα προβλήματα από αυτήν την επίθεση θα ήταν αρκετά σημαντικά αν ο ερευνητής Dan Kaminsky, ο οποίος ανακάλυψε την συγκεκριμένη αδυναμία, δεν είχε φροντίσει να ενημερώσει τους υπευθύνους αρκετό καιρό πριν προβεί σε λεπτομερείς ανακοινώσεις. Την επίθεση αυτή εκτελέσαμε στο Εργαστήριο πραγματοποιώντας πείραμα για τη δοκιμή των υψηλής αλληλεπίδρασης honeypots. Η ακριβής διαδικασία που ακολουθείται για την επίτευξη της επίθεσης θα περιγραφεί αναλυτικά παρακάτω κατά την ανάλυση των δεδομένων που προκύπτουν από αυτή.

**Σκοπός του Πειράματος**

Το πείραμα που εκτελέσαμε στο εργαστήριο αποτελούταν από ηλεκτρονική επίθεση εναντίον ενός ευάλωτου recursive διακομιστή στην παραπάνω αδυναμία και έπειτα η ανάλυση των δεδομένων με την βοήθεια των εργαλείων του Honeywall (roo 1.3). Πιο συγκεκριμένα πραγματοποιήσαμε επίθεση χρησιμοποιώντας το Metasploit Framework, από υπολογιστή που ορίσαμε ως επιτιθέμενο, σε έναν recursive διακομιστή στοχεύοντας στην τοποθέτηση πλαστής εγγραφής στην cache μνήμη του. Η πλαστή εγγραφή αφορούσε στην αντιστοίχηση ενός

ιστοχώρου σε διεύθυνση IP διαφορετική από την πραγματική. Ο απώτερος σκοπός του πειράματος είναι η καταγραφή όσο τον δυνατόν περισσότερων πληροφοριών σχετικά με την εξέλιξη της επίθεσης και η ανάλυση αυτών.

## Περιγραφή Μηχανημάτων

Για τις ανάγκες αυτού του πειράματος χρειάστηκε να δημιουργήσουμε μια πολύπλοκη τοπολογία αποτελούμενη από αρκετά διαφορετικά συστήματα. Χρειάστηκε να χρησιμοποιηθούν συνολικά τέσσερα φυσικά μηχανήματα. Τα δύο εξ αυτών χρησιμοποιήθηκαν ως ξενιστές[1] (hosts) και φιλοξένησαν συνολικά επτά εικονικά μηχανήματα.

Οι τρεις φυσικές μηχανές οι οποίες χρησιμοποιήθηκαν για το πείραμα ήταν οι εξής:

1. Ξενιστής Anafi
2. Ξενιστής Sikinos
3. Ξενιστής Schinousa
4. Ξενιστής Samos

Στους πίνακες 5.1 έως 5.4 γίνεται μια σύντομη περιγραφή των χαρακτηριστικών τους.

Πίνακας 5.1: Περιγραφή του Ξενιστή Anafi.

| **Ξενιστής Anafi** | |
|---|---|
| **Λειτουργικό Σύστημα** | Ubuntu 9.10 Server Edition x386 |
| **Γραφικό Περιβάλλον** | Fluxbox |
| **Interfaces** | eth0: Διεπαφή διαχείρισης (δίκτυο παραγωγής)<br>eth1: Διεπαφή πειράματος (ιδιωτικό VLan) |
| **Υπηρεσίες** | Open ssh |
| **Προγράμματα** | Sun VirtualBox |
| **Σύντομη Περιγραφή** | Στον ξενιστή Anafi δημιουργήθηκε το εικονικό DNS honeypot στο οποίο πραγματοποιήθηκε η επίθεση. |

1 Βλέπε γλωσσάρι

Πίνακας 5.2: Περιγραφή του Ξενιστή Sikinos

| **Ξενιστής Sikinos** | |
|---|---|
| **Λειτουργικό Σύστημα** | Ubuntu 9.10 Server Edition x386 |
| **Γραφικό Περιβάλλον** | Fluxbox |
| **Interfaces** | eth0: Διεπαφή διαχείρισης (δίκτυο παραγωγής)<br>eth1: Διεπαφή πειράματος (ιδιωτικό VLan) |
| **Υπηρεσίες** | Open ssh |
| **Προγράμματα** | Sun VirtualBox |
| **Σύντομη Περιγραφή** | Στον ξενιστή Sikinos δημιουργήθηκαν τα εικονικά μηχανήματα τα οποία ήταν απαραίτητα για την πραγματοποίηση της επίθεσης. |

Πίνακας 5.3: Περιγραφή του Ξενιστή Schinousa.

| **Ξενιστής Schinousa** | |
|---|---|
| **Λειτουργικό Σύστημα** | Ubuntu 9.10 Server Edition x386 |
| **Γραφικό Περιβάλλον** | Fluxbox |
| **Interfaces** | eth0: Διεπαφή διαχείρισης (δίκτυο παραγωγής)<br>eth1: Διεπαφή πειράματος (ιδιωτικό VLan) |
| **Υπηρεσίες** | Open ssh |
| **Προγράμματα** | Sun VirtualBox |
| **Σύντομη Περιγραφή** | Στον ξενιστή Schinousa δημιουργήθηκαν τα εικονικά μηχανήματα που χρησιμοποιήθηκαν από τον επιτιθέμενο χρήστη. |

Πίνακας 5.4: Περιγραφή του Ξενιστή Samos

| **Ξενιστής Samos** | |
|---|---|
| **Λειτουργικό Σύστημα** | CentOS (αποτελεί μέρος της προκαθορισμένης εγκατάστασης του Honeywall) |
| **Γραφικό Περιβάλλον** | Δεν υπάρχει |
| **Interfaces** | eth0: bridged interface in (είσοδος δεδομένων)<br>eth1: bridged interface out (έξοδος δεδομένων) |

| | |
|---|---|
| | eth2: Διεπαφή διαχείρισης (δίκτυο παραγωγής) |
| **Υπηρεσίες** | - |
| **Προγράμματα** | - |
| **Σύντομη Περιγραφή** | Στον ξενιστή Samos εγκαταστάθηκε το Honeywall. |

Όπως φαίνεται από τους πίνακες 5.1, 5.2 και 5.3 στα μηχανήματα Anafi, Sikinos και Schinousa εγκαταστάθηκε και γραφικό περιβάλλον, απαραίτητο για τη λειτουργία του λογισμικού δημιουργίας και διαχείρισης εικονικών μηχανημάτων Sun VirtualBox.

Το VirtualBox, είναι ένα εύχρηστο πρόγραμμα ανοικτού κώδικα που χρησιμοποιείται για την δημιουργία εικονικών μηχανημάτων. Δημιουργήθηκε από την εταιρία Inotek, η οποία εξαγοράσθηκε στη συνέχεια από την εταιρία Sun Microsystems και η πρώτη κυκλοφορία του ήταν στις 15 Ιανουαρίου 2007. Η εφαρμογή αναπτύσσεται πλέον από την εταιρία Oracle, η οποία εξαγόρασε την Sun Microsystems. Η πιο πρόσφατη έκδοση του VirtualBox, τη στιγμή που γραφόταν αυτή εδώ η εργασία, είναι η 3.1.4 με ημερομηνία έκδοσης την 12 Φεβρουαρίου 2010.

Για τη σωστή λειτουργία του VirtualBox απαιτείται η ύπαρξη γραφικού περιβάλλοντος. Γι' αυτό το λόγω επιλέχθηκε να εγκατασταθεί το Fluxbox, το οποίο είναι ένας λιτός διαχειριστής παραθύρων (window manager), ο οποίος καταναλώνει ελάχιστους πόρους συστήματος. Στη θέση του VirtualBox θα μπορούσε να είχε χρησιμοποιηθεί πρόγραμμα που εκτελείτε μέσω γραμμής εντολών (ώστε να μην είναι απαραίτητη η χρήση γραφικού περιβάλλοντος), όπως για παράδειγμα το KVM. Η επιλογή του VirtualBox υπερίσχυσε λόγω της ευκολίας χρήσης του.

Στους παραπάνω ξενιστές αναπτύχθηκαν τα εικονικά συστήματα που περιγράφονται στη συνέχεια:

### Ξενιστής Anafi

Ο ξενιστής Anafi περιείχε το honeypot τα παρακάτω εικονικά μηχανήματα:

1. Target DNS Server

Το παραπάνω εικονικά μηχανήματα αναλύεται στον πίνακα 5.5.

Πίνακας 5.5: Περιγραφή του εικονικού μηχανήματος Target DNS Server

| **Target DNS Server** | |
|---|---|
| **Διεύθυνση IP:** | 192.168.0.130 |
| **Λειτουργικό Σύστημα:** | Ubuntu Server 9.10 x386 |
| **Προγράμματα – Υπηρεσίες:** | Bind 9.2 |
| **Σύντομη Περιγραφή:** | Το εικονικό μηχάνημα “Target DNS Server” παραμετροποίηθηκε ως ένας recursive DNS server και αποτέλεσε τον αποδέκτη της επίθεσης που πραγματοποιήθηκε στο εργαστήριο. Προκειμένου να επιτευχθεί η επίθεση χρειάστηκε να εγκατασταθεί μια παλιότερη και ευάλωτη έκδοση του Bind η “9.2”. Η έκδοση αυτή του Bind χρειάστηκε να γίνει να εγκατασταθεί μέσω του πηγαίου κώδικα εφόσον δεν υπάρχει πλέον στις αποθήκες λογισμικού. |

### Ξενιστής Sikinos

Στον ξενιστή Sikinos δημιουργήθηκαν τα παρακάτω εικονικά μηχανήματα:

1. Root Server
2. TLD DNS Server
3. Victim DNS Server
4. Victim Web Site

Η παραμετροποίηση των παραπάνω εικονικών μηχανημάτων περιγράφεται στους πίνακες 5.6 έως 5.9.

Πίνακας 5.6: Περιγραφή του εικονικού μηχανήματος Root DNS Server

| **Root DNS Server** | |
|---|---|
| **Διεύθυνση IP:** | 192.168.0.140 |
| **Λειτουργικό Σύστημα:** | Ubuntu Server 9.10 x386 |
| **Προγράμματα – Υπηρεσίες:** | Bind 9.6 |
| **Σύντομη Περιγραφή:** | Το εικονικό μηχάνημα “Root DNS Server” παραμετροποιήθηκε ως ένας «ψεύτικος» Root DNS server για τις ανάγκες του πειράματος. Τοποθετήθηκε μια εγγραφή για το TLD “gr”. |

Πίνακας 5.7: Περιγραφή του εικονικού μηχανήματος TLD DNS Server

| **TLD DNS Server** | |
|---|---|
| **Διεύθυνση IP:** | 192.168.0.150 |
| **Λειτουργικό Σύστημα:** | Ubuntu Server 9.10 x386 |
| **Προγράμματα – Υπηρεσίες:** | Bind 9.6 |
| **Σύντομη Περιγραφή:** | Το εικονικό μηχάνημα “TLD DNS Server” παραμετροποιήθηκε ως ένας «ψεύτικος» TLD server. Ορίστηκε ως υπεύθυνος για την ζώνη “gr.” στην οποία τοποθετήθηκε μια εγγραφή για το όνομα χώρου “fakechool.gr”. |

Πίνακας 5.8: Περιγραφή του εικονικού μηχανήματος Victim DNS Server

| **Victim DNS Server** | |
|---|---|
| **Διεύθυνση IP:** | 192.168.0.160 |
| **Λειτουργικό Σύστημα:** | Ubuntu Server 9.10 x386 |
| **Προγράμματα – Υπηρεσίες:** | Bind 9.6 |
| **Σύντομη Περιγραφή:** | Το εικονικό μηχάνημα “Victim DNS Server” ορίστηκε ως ο υπεύθυνος διακομιστής ονομάτων για την ζώνη “fakeschool.gr”. Σε αυτή τη ζώνη τοποθετήθηκε μια εγγραφή για το όνομα χώρου “school.fakeschool.gr.”. |

Πίνακας 5.9: Περιγραφή του εικονικού μηχανήματος Victim Web Site

| **Victim Web Site** | |
|---|---|
| **Διεύθυνση IP:** | 192.168.0.170 |
| **Λειτουργικό Σύστημα:** | Ubuntu Server 9.10 x386 |
| **Προγράμματα – Υπηρεσίες:** | Bind 9.6 |
| **Σύντομη Περιγραφή:** | Το εικονικό μηχάνημα “Victim Web Site” περιείχε την πραγματική ιστοσελίδα η οποία αντιστοιχίζεται στο όνομα χώρου “school.fakeschool.gr.”. Για να είναι προσπελάσιμη η ιστοσελίδα χρειάστηκε να εγκατασταθεί ο Apache Server. |

### Ξενιστής Schinousa

Στον ξενιστή Schinousa δημιουργήθηκαν τα παρακάτω εικονικά μηχανήματα:

1. Attacker
2. Malicious WEB Site

Η παραμετροποίηση των παραπάνω εικονικών μηχανημάτων περιγράφεται στους πίνακες 5.10 και 5.11.

Πίνακας 5.10: Περιγραφή του εικονικού μηχανήματος Attacker

| **Attacker** | |
|---|---|
| **Διεύθυνση IP:** | 192.168.0.60 |
| **Λειτουργικό Σύστημα:** | Ubuntu Server 9.10 x386 |
| **Προγράμματα – Υπηρεσίες:** | Metasploit Framework |
| **Σύντομη Περιγραφή:** | Το εικονικό μηχάνημα “Attacker” είναι το μηχάνημα από το οποίο πραγματοποιείται η επίθεση προς τα DNS Honeypots. |

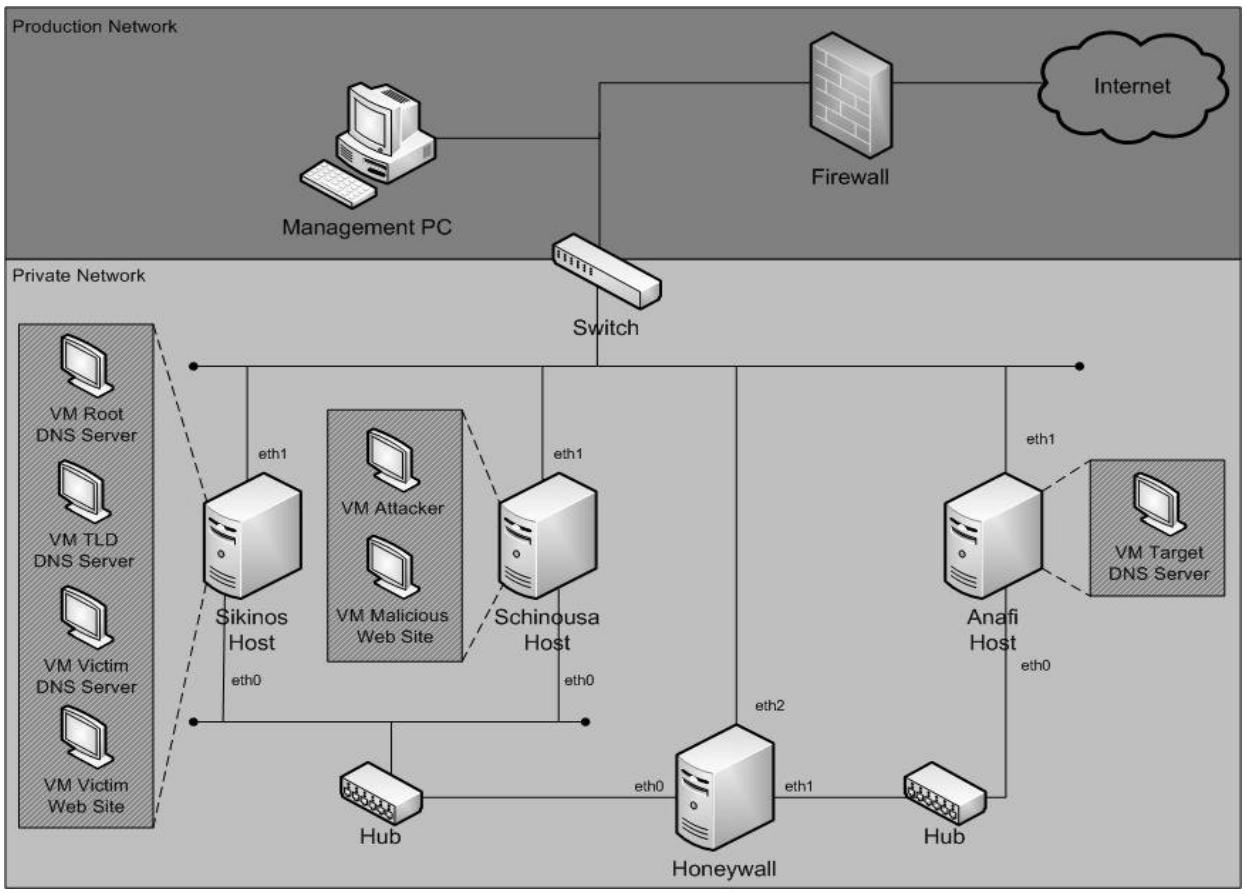
Πίνακας 5.11: Περιγραφή του εικονικού μηχανήματος Malicious Web Site

| **Malicious Web Site** | |
|---|---|
| **Διεύθυνση IP:** | 192.168.0.70 |
| **Λειτουργικό Σύστημα:** | Ubuntu Server 9.10 x386 |
| **Προγράμματα – Υπηρεσίες:** | Apache Server 2.2 |
| **Σύντομη Περιγραφή:** | Το εικονικό μηχάνημα “Malicious Web Site” περιείχε την ιστοσελίδα στην οποία ανακατευθύνθηκε η κίνηση μετά την επίτευξη της επίθεσης. Για να είναι προσπελάσιμη η ιστοσελίδα χρειάστηκε να εγκατασταθεί ο Apache Server. |

### Ξενιστής Samos

Στον ξενιστή Samos εγκαταστάθηκε το Honeywall. Το Honeywall καθώς και τα εργαλεία τα οποία περιλαμβάνει περιγράφονται αναλυτικά στο κεφάλαιο 2.

Η τοπολογία που δημιουργήθηκε περιγράφεται σχηματικά παρακάτω στο σχήμα 5.1




Σχήμα 5.1: Τοπολογία που σχηματίστηκε για την εκτέλεση του πειράματος.

Όπως φαίνεται και από το παραπάνω σχήμα οι δύο ξενιστές που φιλοξενούσαν τα εικονικά μηχανήματα τοποθετήθηκαν αντιδιαμετρικά, με το Honeywall να βρίσκεται ανάμεσα τους και να καταγράφει όλες τις συνδέσεις και όλα τα πακέτα που ανταλλάσσονται. Επίσης σε όλα τα μηχανήματα ορίστηκαν διεπαφές διαχείρισης οι οποίες ήταν συνδεδεμένες με το εσωτερικό δίκτυο του Εργαστηρίου. Η διαχείριση των μηχανών πραγματοποιόταν μέσω του υπολογιστή διαχείρισης (Management PC στο παραπάνω σχήμα), ο οποίος άνηκε στο εσωτερικό δίκτυο του Εργαστηρίου, δια μέσω των διεπαφών διαχείρισης.

**Παραμετροποίηση Συστημάτων**

Στο παράρτημα Β υπάρχει λεπτομερής οδηγός για την παραμετροποίηση των συστημάτων.

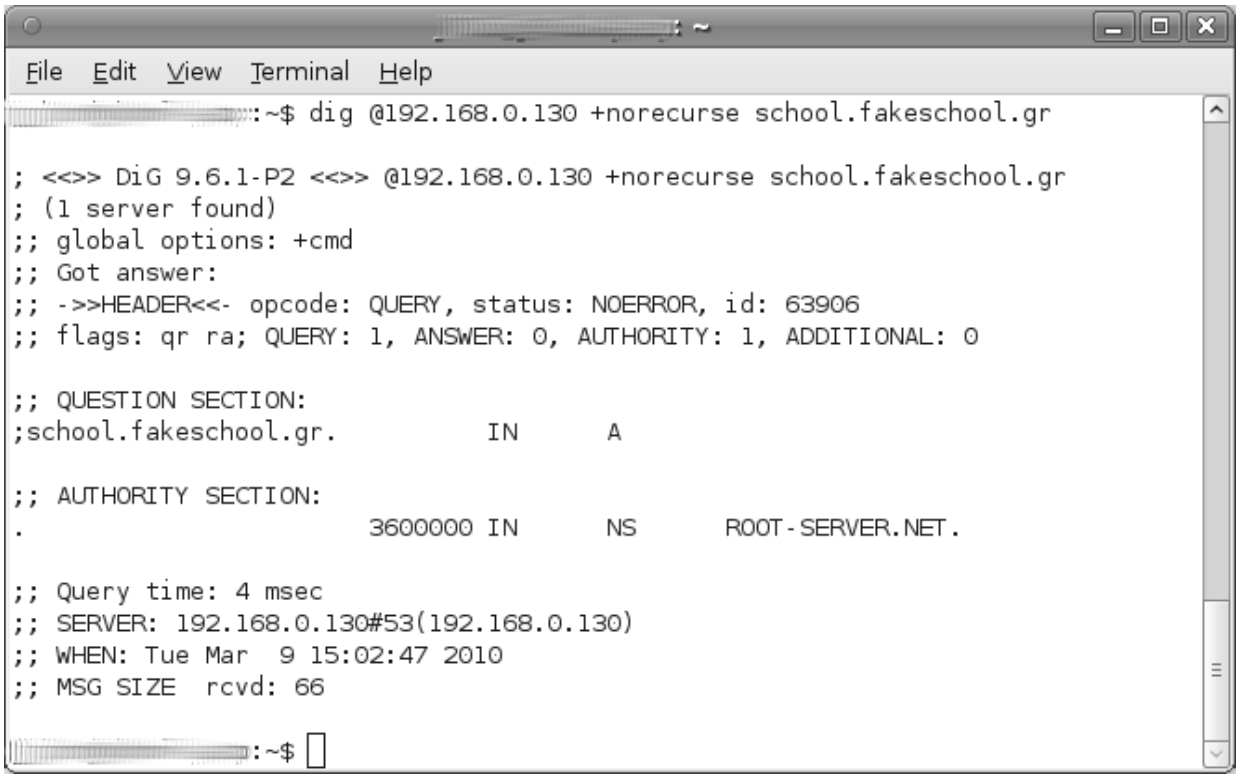
**Επιβεβαίωση σωστής λειτουργίας**

Πρώτο μας βήμα αρχικά είναι να επιβεβαιώσουμε τη σωστή λειτουργία των μηχανημάτων αλλά και τα σωστά αποτελέσματα των υπηρεσιών πριν από την πραγματοποίηση της επίθεσης.

Αρχικά θα εξετάσουμε την απάντηση που θα μας δώσει ο Victim DNS server, για το όνομα χώρου "school.fakeschool.gr", σε ένα μη recursive ερώτημα όταν η μνήμη cache του διακομιστή είναι κενή. Για να το πραγματοποιήσουμε το ερώτημα χρησιμοποιήσαμε την εφαρμογή "dig" και συγκεκριμένα εκτελέσαμε την παρακάτω εντολή:

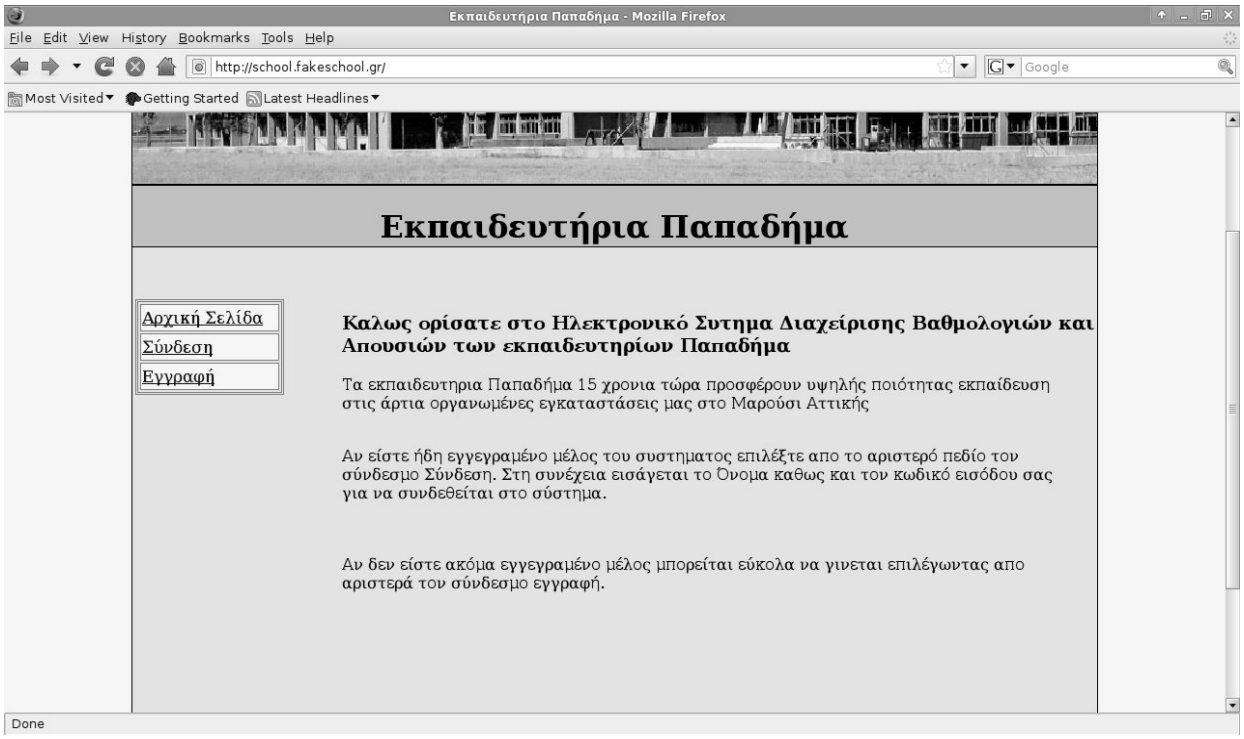
```
dig @192.168.0.130 +norecurse school.fakeschool.gr
```

Η παράμετρος "+norecurse" ειδοποιεί τον διακομιστή να μην εκτελέσει το ερώτημα ως recursive. Δηλαδή να μην ρωτήσει κανένα άλλο διακομιστή για την επίλυση του ερωτήματος. Όπως βλέπουμε παρακάτω στο σχήμα 5.2 ο Target DNS server μας αποστέλλει μια απάντηση με την μοναδική πληροφορία την οποία έχει, δηλαδή την πληροφορία για τη ζώνη "." (root).




Σχήμα 5.2: Απάντηση του Victim DNS server με κενή την μνήμη cache.

Εν συνεχεία συνδέοντας κάποιο υπολογιστή στο δίκτυο ως χρήστη δοκιμάζουμε να προσπελάσουμε την ιστοσελίδα, για το όνομα χώρου της οποίας, θα πραγματοποιηθεί η επίθεση. Ο συγκεκριμένος υπολογιστής-χρήστης είχε εγκατεστημένο λειτουργικό σύστημα τύπου Linux. Τοποθετώντας λοιπόν το αρχείο "/etc/resolv.conf"[1] την διεύθυνση IP του "Target DNS server" και αφαιρώντας τις υπόλοιπες εγγραφές καθορίσαμε ως υπεύθυνο διακομιστή του υπολογιστή-χρήστη τον recursive διακομιστή. Στο σχήμα 5.3 βλέπουμε την πραγματική ιστοσελίδα που εμφανίζεται πριν την πραγματοποίηση της επίθεσης. Η σελίδα προβλήθηκε μέσω της χρήσης του "Target DNS server".

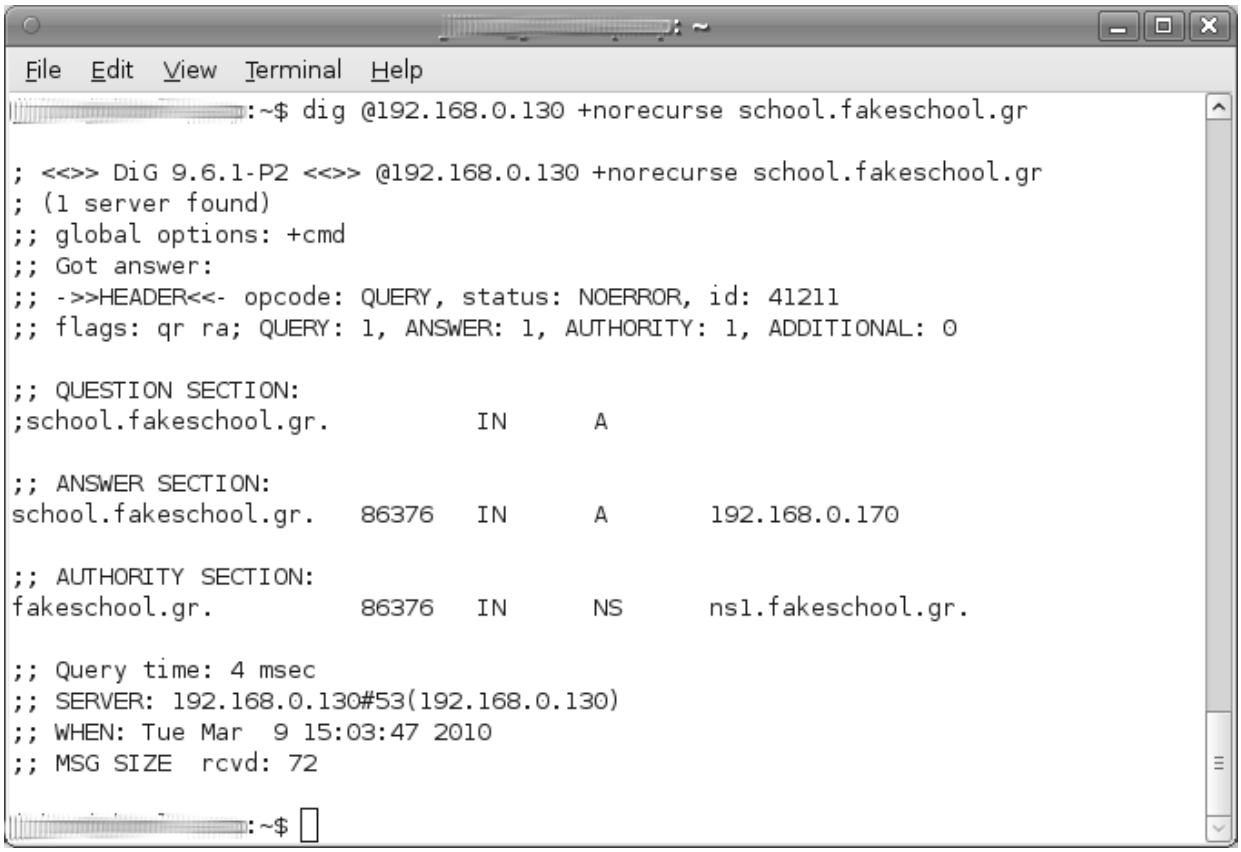



Σχήμα 5.3: Η πραγματική ιστοσελίδα πριν από την πραγματοποίηση της επίθεσης

Μετά τη προβολή της ιστοσελίδας εξετάζουμε ξανά την μνήμη cache πραγματοποιώντας και πάλι ένα μη recursive ερώτημα όπως και παραπάνω. Αυτή τη φορά όπως βλέπουμε και στο σχήμα 5.4 ο διακομιστής μας επιστρέφει την σωστή απάντηση με την διεύθυνση IP στην οποία βρίσκεται η ιστοσελίδα. Δηλαδή, μετά από την προσπέλαση της ιστοσελίδας από τον υπολογιστή-χρήστη η μνήμη cache περιέχει την εγγραφή που επιθυμούμε λόγω του της

1 Είναι το αρχείο στο οποίο καθορίζονται στα τα λειτουργικά συστήματα τύπου UNIX, οι recursive διακομιστές ονομάτων που θα χρησιμοποιεί το μηχάνημα.

“recursive” ερώτησης που πραγματοποιήθηκε. Η εγγραφή αυτή έχει συγκεκριμένο TTL (86376), όπως φαίνεται και στο σχήμα, μετά το πέρας του οποίου θα απομακρυνθεί από την προσωρινή μνήμη.




Σχήμα 5.4: Προβολή της μνήμης cache μετά την προσπέλαση της ιστοσελίδας

Στη συνέχεια «αδειάζουμε» την μνήμη cache του Target DNS έτσι ώστε αυτός να είναι έτοιμος για την εκκίνηση της επίθεσης. Για να «αδειάσουμε» την μνήμη cache χρησιμοποιούμε την εντολή:

```
rndc flush
```

**Εκτέλεση επίθεσης**

Για την εκτέλεση της επίθεσης χρησιμοποιήθηκε η πλατφόρμα ανοικτού κώδικα Metasploit Framework. Το Metasploit Framework περιέχει δύο Modules για την πραγματοποίηση της

επίθεσης. Τα modules αυτά είναι τα εξής:

1. **Bailiwicked_host:** Αυτό ήταν το module που χρησιμοποιήθηκε για την πραγματοποίηση της επίθεσης στο εργαστήριο. Έχει τη δυνατότητα να τοποθετεί μεμονωμένες εγγραφές, με πλαστά στοιχεία, στην μνήμη cache ενός recursive διακομιστή ονομάτων με σκοπό την παραπλάνηση των χρηστών που θα χρησιμοποιήσουν τη συγκεκριμένη υπηρεσία..

2. **Bailiwicked_server:** Το module αυτό εκτελεί μια επίθεση με μεγαλύτερο αντίκτυπο εν συγκρίσει με αυτή . Συγκεκριμένα επιτρέπει την τοποθέτηση πλαστών εγγραφών, στην μνήμη cache ενός recursive server με σκοπό την εξυπηρέτηση ενός ονόματος χώρου από κάποιο ελεγχόμενο από τον επιτιθέμενο χρήστη διακομιστή Διαδικτύου. Περισσότερες πληροφορίες γι'αυτό το module παρατίθενται στο παράρτημα Δ.

Λόγω του ότι υπήρξαν προβλήματα στην εκτέλεση της επίθεσης με την έκδοση 3.3.3 του Metasploit Framework επιλέχθηκε τελικά η έκδοση 3.2 αυτού. Για την εκτέλεση της επίθεσης χρησιμοποιήθηκε το "msf_console" το οποίο βρίσκεται στο φάκελο εγκατάστασης του Metasploit Framework και εκτελείται μέσω γραμμής εντολών. Για να εκτελέσουμε την επίθεση ακολουθούμε τα παρακάτω βήματα:

1. Επιλέγουμε το επιθυμητό module, χρησιμοποιώντας την δεσμευμένη λέξη "use", εκτελώντας στο "msf_console" την παρακάτω εντολή:

   ```
   use spoof/dns/bailiwicked_host
   ```

2. Εν συνεχεία προβάλουμε τα πεδία που περιλαμβάνει προς συμπλήρωση το συγκεκριμένο module χρησιμοποιώντας την παρακάτω εντολή:

   ```
   show options
   ```

3. Έπειτα καθορίζουμε τις τιμές των πεδίων, χρησιμοποιώντας την δεσμευμένη λέξη "set", σύμφωνα με την παρακάτω εντολή:

```
set   <Όνομα Πεδίου προς συμπλήρωση>   <Επιθυμητή Τιμή>
```

4. Τέλος εκκινούμε την επίθεση εκτελώντας της παρακάτω εντολές:

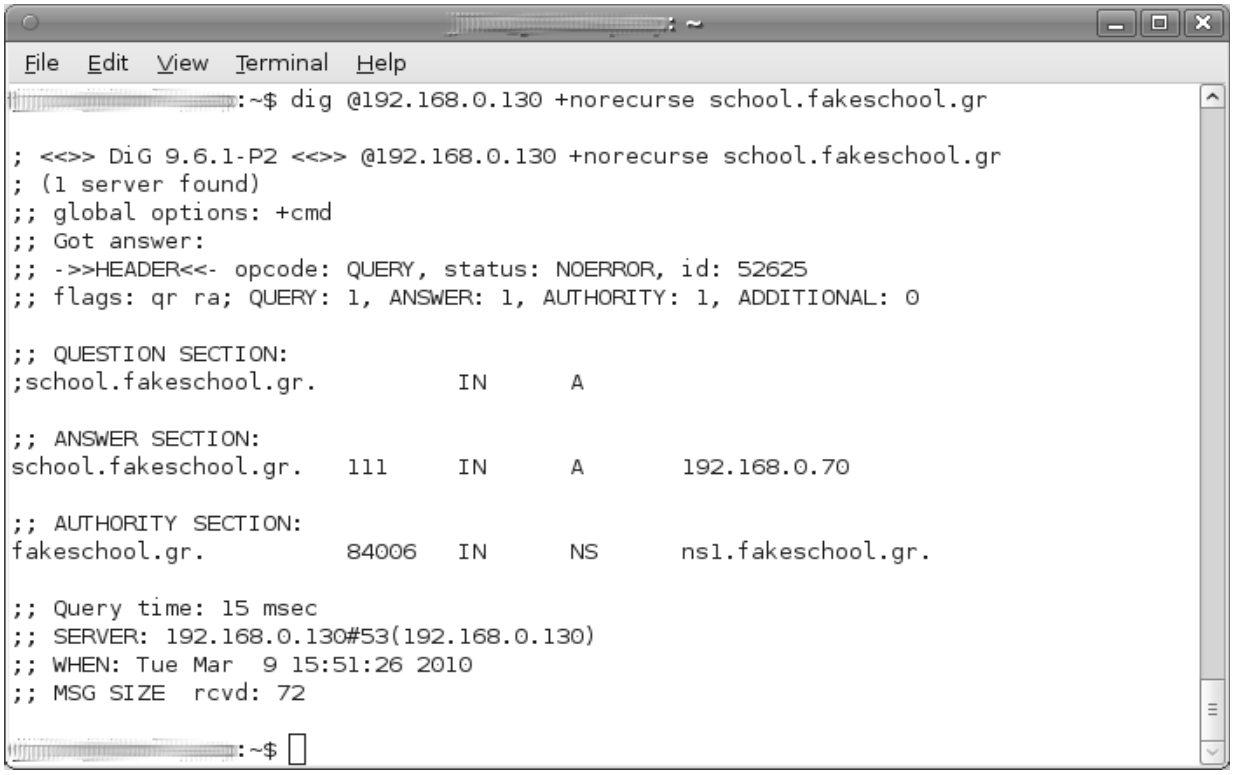
```
run ή exploit
```

Τα πεδία που περιλαμβάνει το module Bailiwicked_host και χρειάστηκε να συμπληρωθούν για την πραγματοποίηση της επίθεσης είναι τα παρακάτω:

- **HOSTNAME:** Σε αυτό το πεδίο συμπληρώνεται το όνομα χώρου για το οποίο επιθυμούμε να πραγματοποιήσουμε την επίθεση.
- **NEWADDR:** Σε αυτό το πεδίο συμπληρώνεται η διεύθυνση IP, με την οποία παράνομα θέλουμε να συσχετιστεί το όνομα χώρου που συμπληρώθηκε στο πεδίο HOSTNAME.
- **RECONS:** Σε αυτό το πεδίο συμπληρώνεται η διεύθυνση IP του διακομιστή ονομάτων
- **RHOST:** Σε αυτό το πεδίο συμπληρώνεται η διεύθυνση IP του recursive διακομιστή ονομάτων στον οποίο θα πραγματοποιηθεί η επίθεση.
- **SRCADDR:** Σε αυτό το πεδίο ορίζουμε αν επιθυμούμε στα πακέτα που αποστέλλονται από το σύστημα μας να περιέχουν την πραγματική διεύθυνση IP ή κάποιες τυχαίες.
- **SRCPORT:** Σε αυτό το πεδίο συμπληρώνεται η σταθερή "source" πόρτα που χρησιμοποιεί ο recursive διακομιστής, στον οποίο γίνεται η επίθεση, για την αποστολή των μηνυμάτων DNS. Με την επιλογή "0" επιλέγεται αυτομάτως η σωστή πόρτα, κάτι που στο εργαστήριο
- **TTL:** Σε αυτό το πεδίο συμπληρώνεται ο χρόνος που επιθυμούμε να παραμείνει στην μνήμη cache του recursive διακομιστή η πλαστή εγγραφή, μετά την επιτυχημένη πραγματοποίηση της επίθεσης.
- **XIDS:** Σε αυτό το πεδίο συμπληρώνεται ο αριθμός των πλαστών απαντήσεων που θα αποστέλλονται για κάθε ερώτημα (θα εξηγηθεί παρακάτω επακριβώς). Με την επιλογή "0" υπάρχει η δυνατότητα να γίνεται αυτόματα η επιλογή των ιδανικότερων τιμών.

**Αποτελέσματα Επίθεσης**

Αμέσως μετά την ολοκλήρωση της επίθεσης, εξετάζουμε την μνήμη cache του recursive διακομιστή η οποία όπως διαπιστώνουμε περιέχει μια «μολυσμένη» εγγραφή. Όπως φαίνεται και στο σχήμα 5.5, πραγματοποιώντας ένα μη recursive ερώτημα παρατηρούμε πως σύμφωνα με την εγγραφή που υπάρχει στην μνήμη cache η ιστοσελίδα "school.fakeschool.gr" βρίσκεται στην διεύθυνση IP "192.168.0.70" αντί της διεύθυνσης IP "192.168.0.170" η οποία είναι η πραγματική. Η επίθεση λοιπόν έχει στεφθεί με επιτυχία και έχει επιτύχει το σκοπό της.




Σχήμα 5.5: Η μνήμη cache του Target DNS server μετά την πραγματοποίηση της επίθεσης

Δοκιμάζοντας να προσπελάσουμε και πάλι την ιστοσελίδα "school.fakeschool.gr" μέσω του υπολογιστή που συνδέσαμε ως χρήστη στο δίκτυο, κατευθυνόμαστε σε στην ιστοσελίδα που υπάρχει στον "Malicious Web Site" όπως φαίνεται και στο σχήμα 5.6. Η ιστοσελίδα αυτή δημιουργήθηκε για τις ανάγκες του πειράματος και περιέχει απλώς πληροφορίες για την επίθεση. Σε περιπτώσεις κανονικών επιθέσεων είναι πολύ πιθανό η ιστοσελίδα να παραπλανήσει τον ανυποψίαστο χρήστη με σκοπό να τον πείσει να δώσει προσωπικά του δεδομένα.


Σχήμα 5.6: Η σελίδα στην οποία κατευθύνεται ο χρήστης μετά την πραγματοποίηση της επίθεσης

## Ανάλυση Επίθεσης

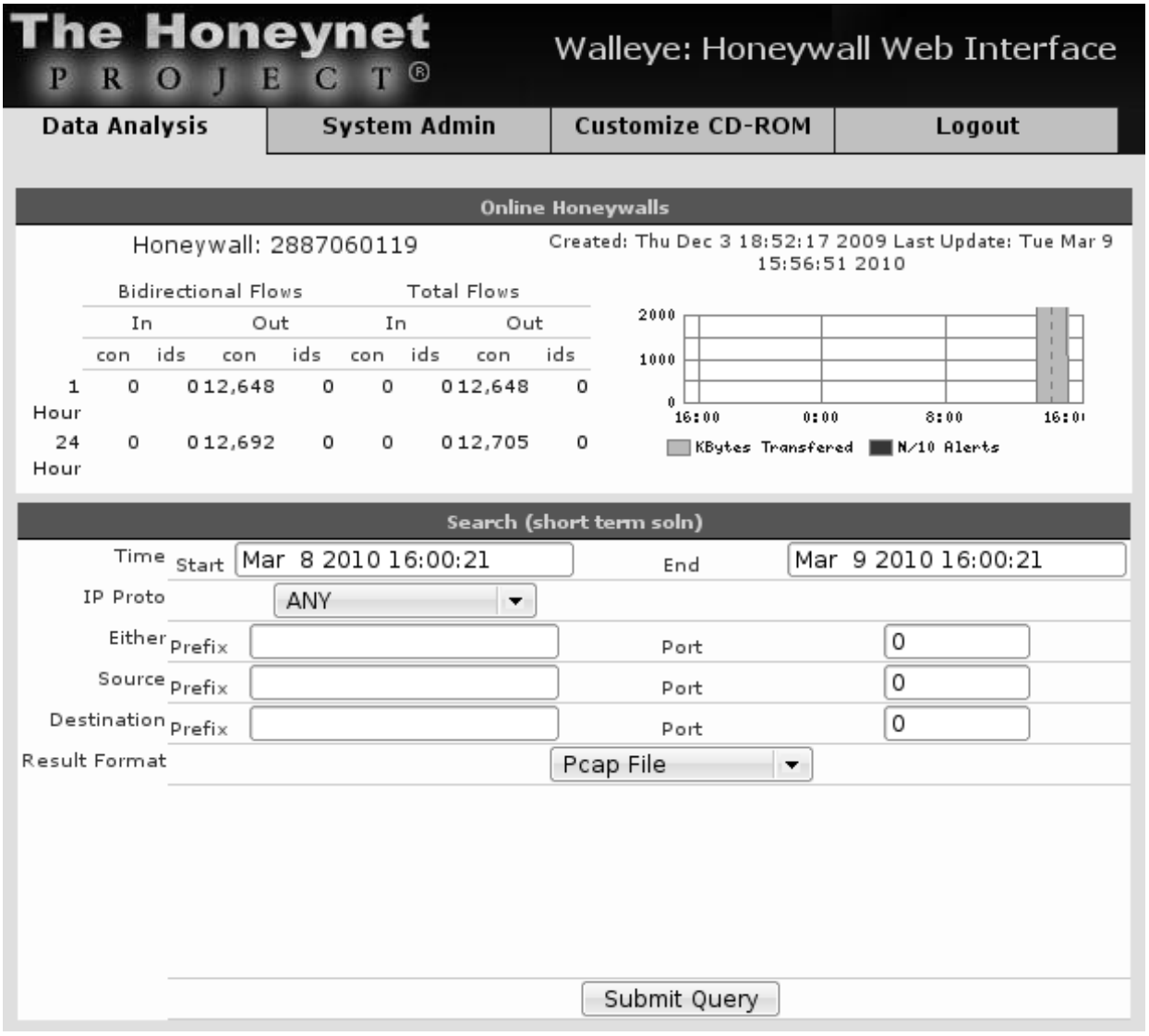
Παρακάτω θα αναλύσουμε την επίθεση όπως πραγματοποιήθηκε στο εργαστήριο αναλύοντας και επεξηγώντας σταδιακά τα αρχεία καταγραφής συμβάντων.

### Ανάλυση των αρχείων καταγραφής συμβάντων

Η ανάλυση της επίθεσης θα γίνει με τέτοιο τρόπο ώστε να εξομοιωθεί η καθημερινότητα ενός honeynet διαχειριστή. Με αυτό το σκεπτικό ένας honeynet διαχειριστής, ο οποίος εξετάζει την καταγραφή συμβάντων του Honeywall για ενδείξεις επιθέσεων, θα αντιμετωπίσει την κατάσταση όπως περιγράφεται στη συνέχεια.

Ενδείξεις επίθεσης

Στις 9 Μαρτίου παρατηρούμε στο δικτυακό περιβάλλον διαχείρισης του Honeywall, το Walleye, μια σημαντική αύξηση στη διερχόμενη κίνηση, όπως φαίνεται στο σχήμα 5.7. Όπως είναι φανερό από το σχήμα πραγματοποιήθηκαν συνολικά 12.648 συνδέσεις από το εγκατεστημένο honeypot μέσα σε μικρό χρονικό διάστημα. Εφόσον ο αριθμός των συνδέσεων και το μέγεθος της κίνησης είναι αρκετά αυξημένα για τα δεδομένα του honeynet είναι φανερό πως κάτι το μη φυσιολογικό συνέβη.




Σχήμα 5.7: Καταγραφή του Walleye μετά την πραγματοποίηση της επίθεσης

Αρχεία καταγραφής συμβάντων Iptables

Εν συνεχεία εξετάζουμε τα αρχεία καταγραφής συμβάντων του Iptables. Σύμφωνα με αυτά όπως φαίνεται και παρακάτω, παρατηρούμε ένα πλήθος εισερχόμενων αλλά και εξερχόμενων συνδέσεων οι οποίες στο μεγαλύτερο σύνολο τους αφορούν το πρωτόκολλο UDP. οι εισερχόμενες συνδέσεις που καταγράφονται από το Iptables εκλαμβάνονται ως ύποπτη δραστηριότητα γι'αυτό και επισημαίνονται (INBOUND UDP)

```
Mar  9 15:39:49 samos kernel: INBOUND UDP: IN=br0 OUT=br0 PHYSIN=eth0 PHYSOUT=eth1
SRC=192.168.0.60 DST=192.168.0.130 LEN=73 TOS=0x00 PREC=0x00 TTL=64 ID=16810 PROTO=UDP
SPT=24116 DPT=53 LEN=53
```

Αντιθέτως τα εξερχόμενα μηνύματα (παρακάτω) χαρακτηρίζονται ως νόμιμα (Legal DNS) πιθανώς γιατί πρόκειται για εξερχόμενες απαντήσεις σε DNS ερωτήματα.

```
Mar  9 15:39:49 samos kernel: Legal DNS: IN=br0 OUT=br0 PHYSIN=eth1 PHYSOUT=eth0
SRC=192.168.0.130 DST=192.168.0.160 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=0 DF PROTO=UDP
SPT=40667 DPT=53 LEN=64
```

Swatch

Εξετάζοντας προσεκτικά όλα τα εισερχόμενα μηνύματα ηλεκτρονικού ταχυδρομείου αλλά και τα κατάλληλα αρχεία καταγραφής παρατηρήσαμε πως το swatch δεν απέστειλε κανένα μήνυμα για την παραπάνω καταγεγραμμένη κίνηση. Αυτό οφείλεται πιθανώς στο Iptables, τα αρχεία καταγραφής του οποίου εξετάζει το Swatch προκειμένου να αποστείλει προειδοποιήσεις. Όπως προείπαμε το Iptables χαρακτηρίζει τα εξερχόμενα UDP μηνύματα ως νόμιμη κίνηση του πρωτοκόλλου DNS. Το Swatch λαμβάνοντας υπόψιν τον συγκεκριμένο χαρακτηρισμό, από την πλευρά του Iptables, να μην εκκινεί τη διαδικασία αποστολής προειδοποιήσεων.

Snort

Συνεχίζοντας την έρευνα, επόμενη μας κίνηση είναι να εξετάσουμε τα αρχεία καταγραφής συμβάντων του Snort. Παρατηρήσαμε πως το Snort δεν εξέδωσε καμία προειδοποίηση κατά το

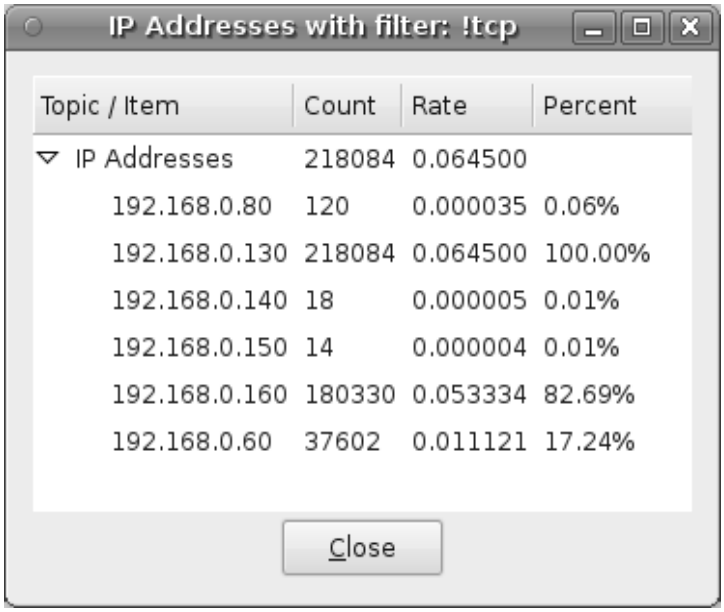
χρονικό διάστημα το οποίο εξετάζουμε. Λόγω του ότι η έκδοση του Snort, η οποία χρησιμοποιήθηκε (και η οποία ήταν κομμάτι της εγκατάστασης του Honeywall ), ήταν αρκετά παλιά, υπήρχε πάντα η πιθανότητα να μην έχει τα κατάλληλα signatures ώστε να αναγνωρίσει την συγκεκριμένη επίθεση. Επομένως δεν καταφέραμε να εξάγουμε σαφή συμπεράσματα από την ανάλυση των αρχείων καταγραφής συμβάντων του Snort.

Pcap αρχεία καταγραφής του Honeywall

Εν συνεχεία ο διαχειριστής του συστήματος θα εξετάσει την καταγεγραμμένη κίνηση από το Honeywall. Σύμφωνα με αυτή:

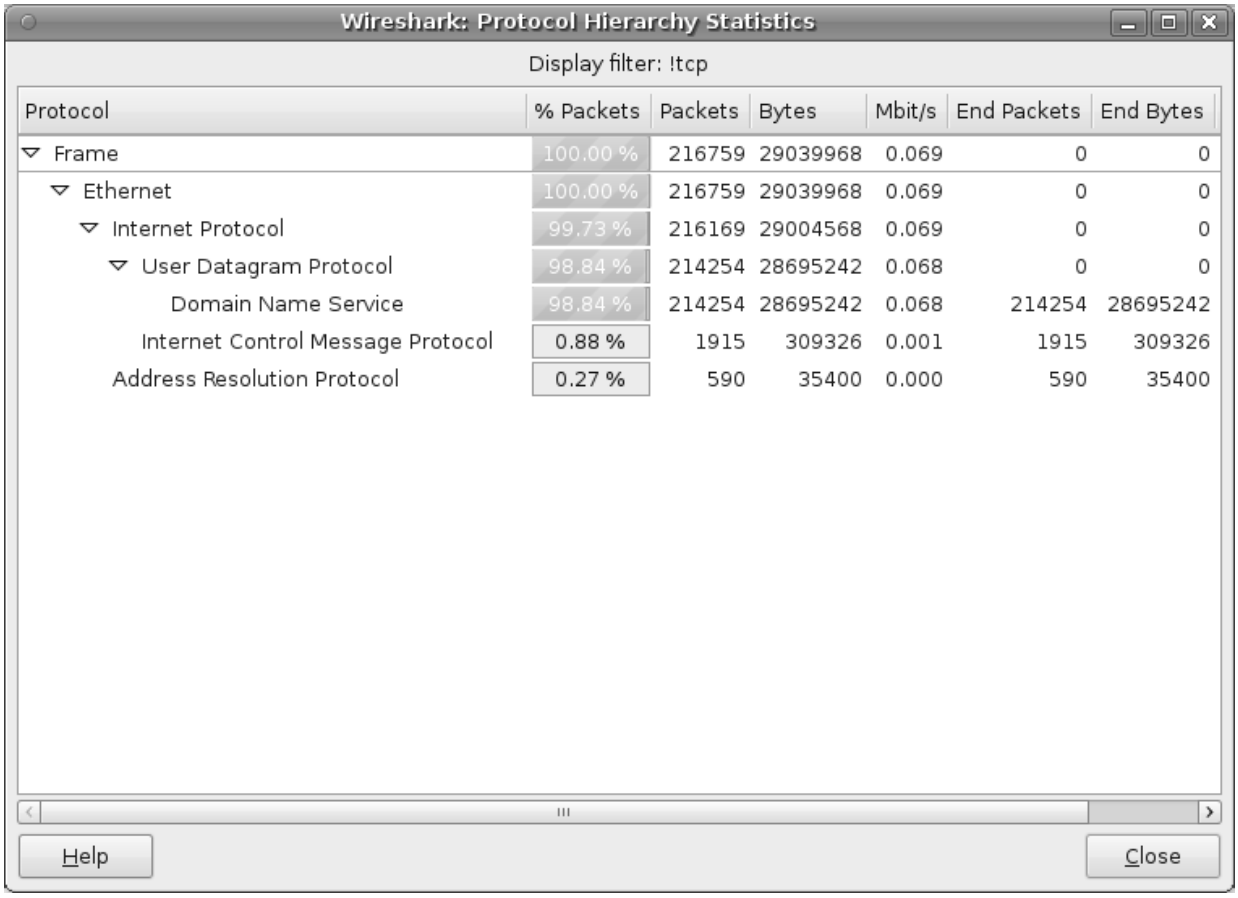
- 216.759 πακέτα ήταν η συνολική κίνηση η οποία καταγράφηκε.
- 6 ήταν συνολικά οι διευθύνσεις IP οι οποίες εντοπίστηκαν στα παραπάνω πακέτα

Όπως βλέπουμε παραπάνω ολόκληρη η δικτυακή κίνηση προέκυψε μονάχα από έξι διευθύνσεις IP γεγονός που εξαρχής γεννά υποψίες. Συνεχίζοντας την ανάλυση βλέπουμε, στο σχήμα 5.8, πως στον κύριο όγκο της κίνησης συμμετέχουν δύο διευθύνσεις IP. Πιο συγκεκριμένα οι διευθύνσεις IP “192.168.0.130” και “192.168.0.160” υπάρχουν στο 100% και 82,69% των πακέτων που ανταλλάχθηκαν αντίστοιχα.




Σχήμα 5.8: Διευθύνσεις IP που καταγράφηκαν

Όπως φαίνεται από το σχήμα 5.9 το 98,84% των μηνυμάτων, ανταλλάχθηκαν δια μέσω του πρωτοκόλλου UDP και συγκεκριμένα αφορούσαν όλα την υπηρεσία DNS. Είναι σαφές επομένως πως κάτι ύποπτο έχει συμβεί στα DNS honeypots καθώς μια τόσο μεγάλη ανταλλαγή αρχείων δεν συμβαίνει υπό φυσιολογικές συνθήκες λειτουργίας του honeynet.

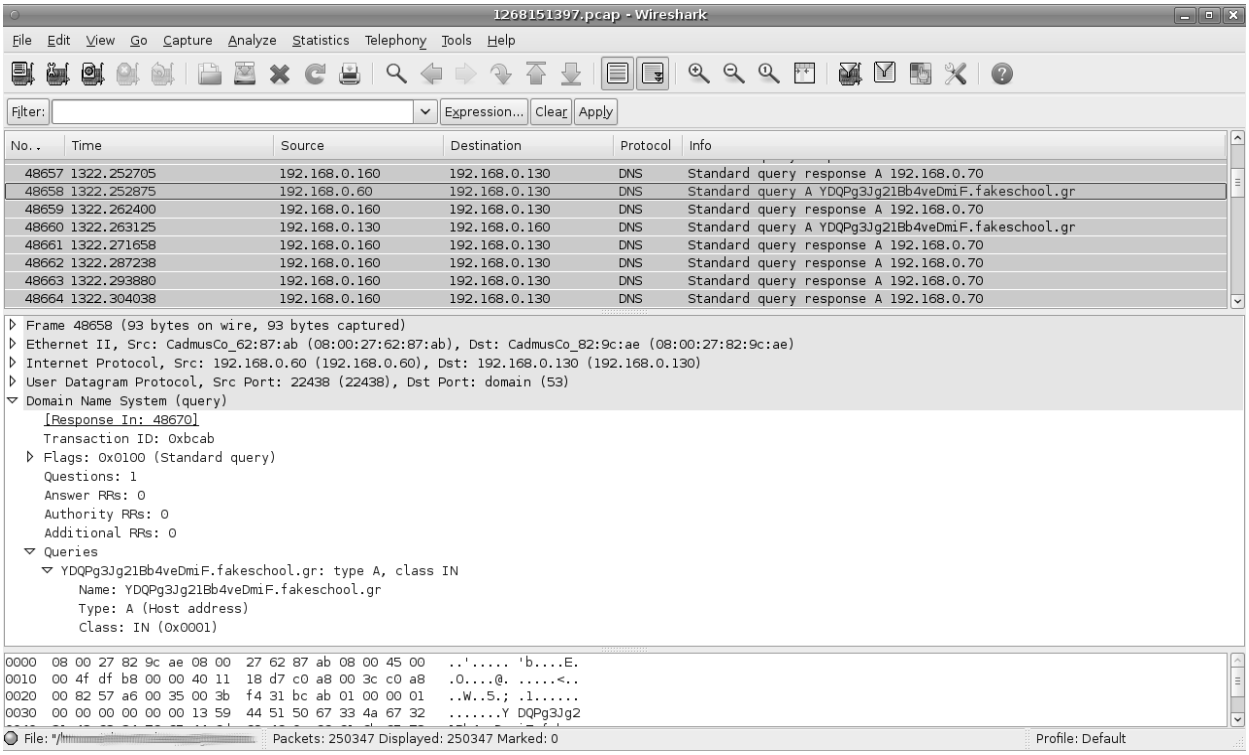



Σχήμα 5.9: Ιεραρχία πρωτοκόλλων των καταγεγραμμένων δεδομένων

Ανάλυση της επίθεσης

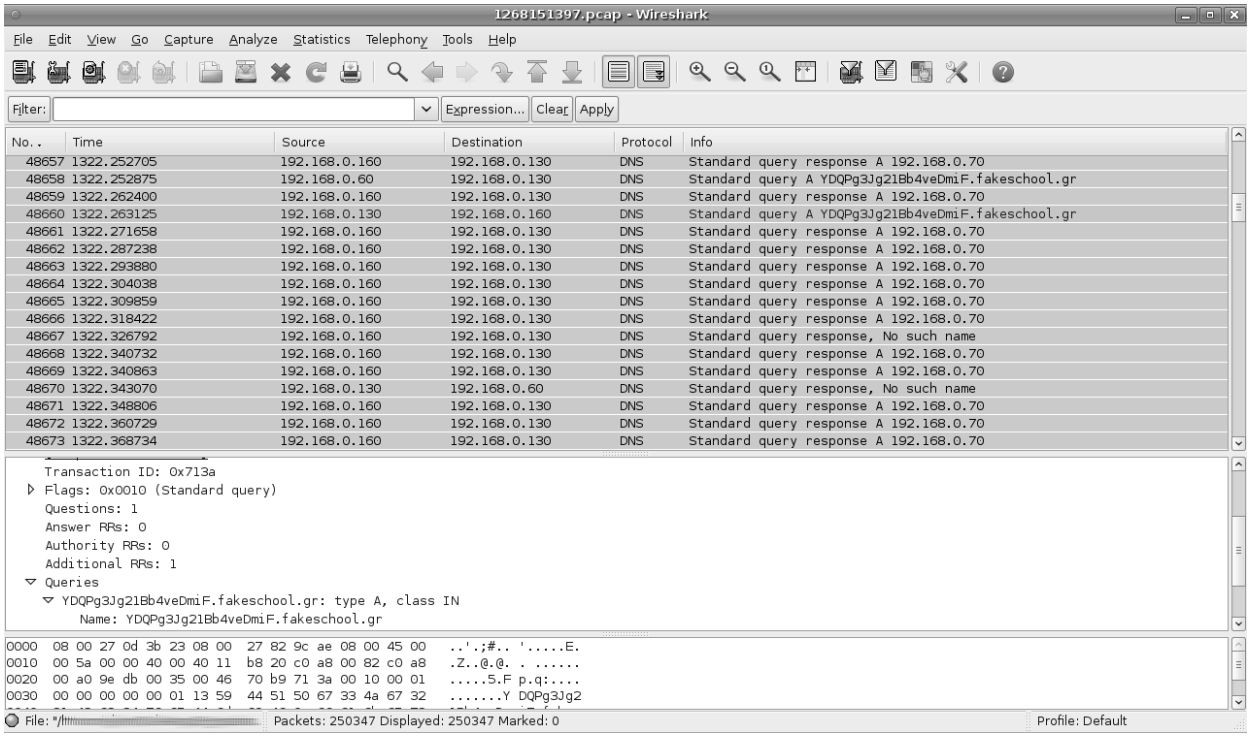
Αναλύοντας προσεκτικότερα τα πακέτα παρακάτω βλέπουμε πως ο υπολογιστής που έχει την διεύθυνση IP 192.168.0.60 αποστέλλει προς επίλυση ερωτήματα που αφορούν παράξενα και μάλλον ανύπαρκτα ονόματα χώρου. Στο σχήμα 5.10 φαίνεται ένα από τα ερωτήματα. Στο συγκεκριμένο ερώτημα ο απομακρυσμένος υπολογιστής πραγματοποιεί ένα ερώτημα για το όνομα χώρου “YDQPg3Jg21Bb4veDmiF.fakeschool.gr.”. Το συγκεκριμένο όνομα χώρου είναι υπερβολικά παράξενο για να είναι πραγματικό. Υπήρξαν πολλά παρόμοια ερωτήματα από τον

συγκεκριμένο υπολογιστή.



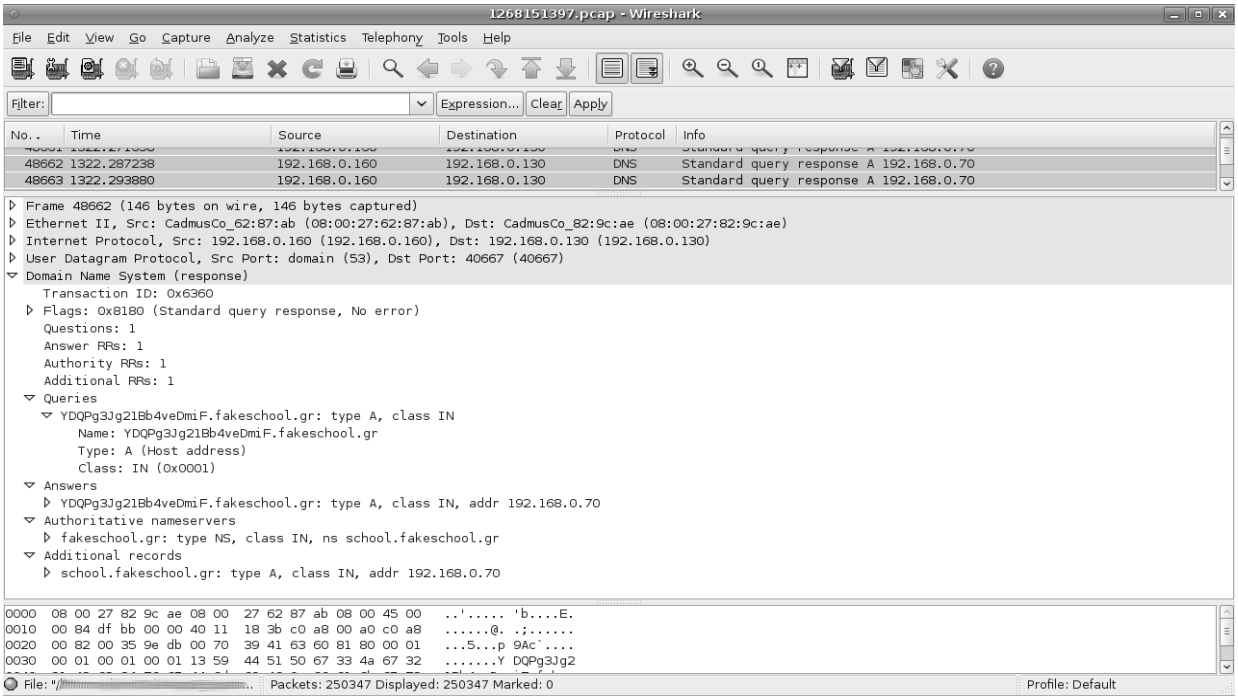

Σχήμα 5.10: Ερώτημα από τον υπολογιστή με διεύθυνση IP “192.168.0.60”

Εν συνεχεία, βλέπουμε στο σχήμα 5.11 , πως αφού ο απομακρυσμένος υπολογιστής αποστείλει το ερώτημα υπάρχουν πολλές απαντήσεις (για το συγκεκριμένο ερώτημα) από τον Victim DNS server (ο οποίος είναι authoritative για την ζώνη “fakeschool.gr.”). Αυτό βέβαια είναι πολύ ύποπτο καθώς κανένας διακομιστής δεν αποστέλλει παραπάνω από μια απάντηση για το ίδιο ερώτημα.

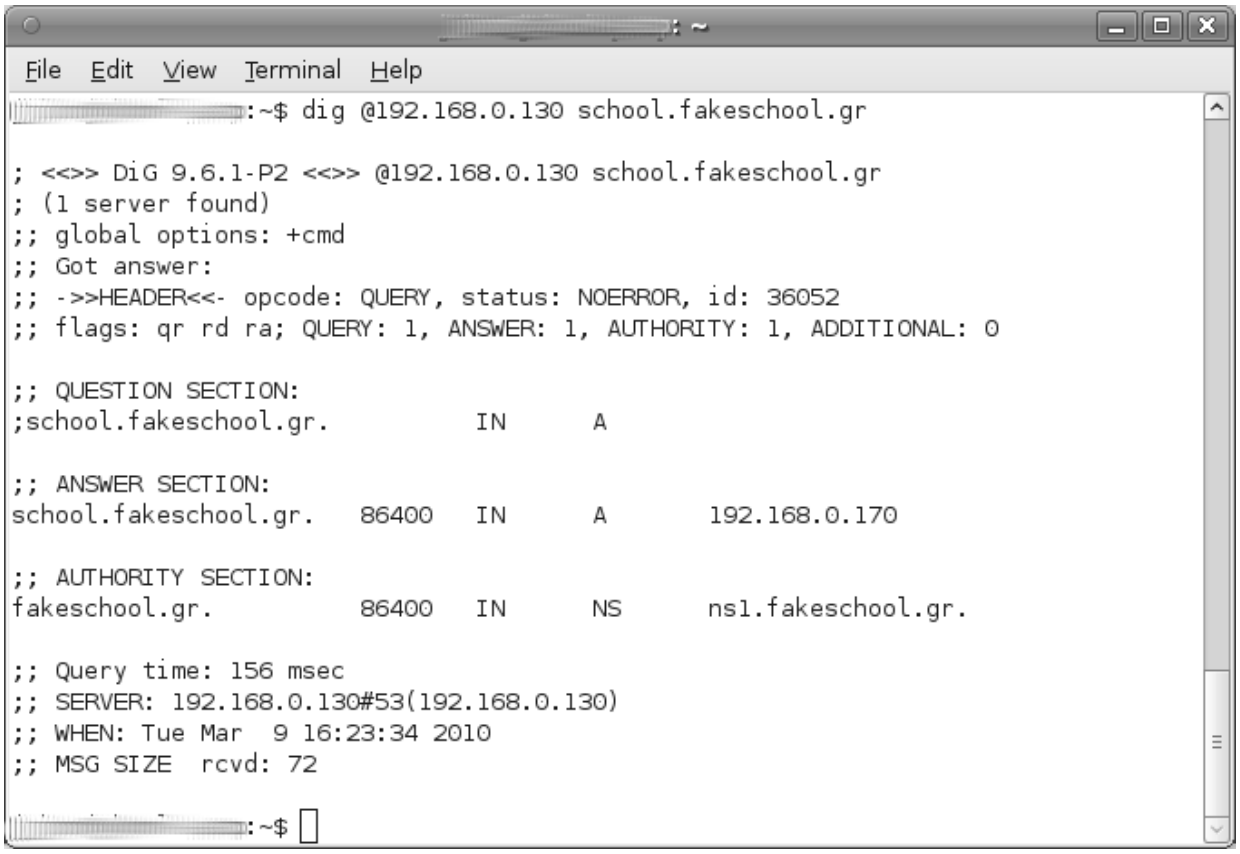



Σχήμα 5.11: Απαντήσεις απο τον authoritative διακομιστή

Εξετάζοντας προσεκτικότερα τις απαντήσεις που εστάλησαν παρατηρούμε, στο σχήμα 5.12, πως ο Victim DNS server επιστρέφει ως απάντηση στο ερώτημα για το όνομα χώρου “YDQPg3Jg21Bb4veDmiF.fakeschool.gr.” τη διεύθυνση IP “192.168.0.70”. Επίσης προσθέτει στο “Additional” κομμάτι την πληροφορία πως το όνομα χώρου “school.fakeschool.gr.” αντιστοιχίζεται στην διεύθυνση IP “192.168.0.70”. Εκτελώντας ένα ερώτημα για το όνομα χώρου “school.fakeschool.gr” στον Target DNS Server χρησιμοποιώντας την εφαρμογή dig, όπως φαίνεται και στο σχήμα 5.13, βλέπουμε πως το συγκεκριμένο όνομα χώρου αντιστοιχίζεται με την διεύθυνση IP “192.168.0.170” και όχι με την “192.168.0.70”. Αυτό αποτελεί και την τελική επιβεβαίωση πως πρόκειται για επίθεση και συγκεκριμένα για επίθεση τύπου spoof.



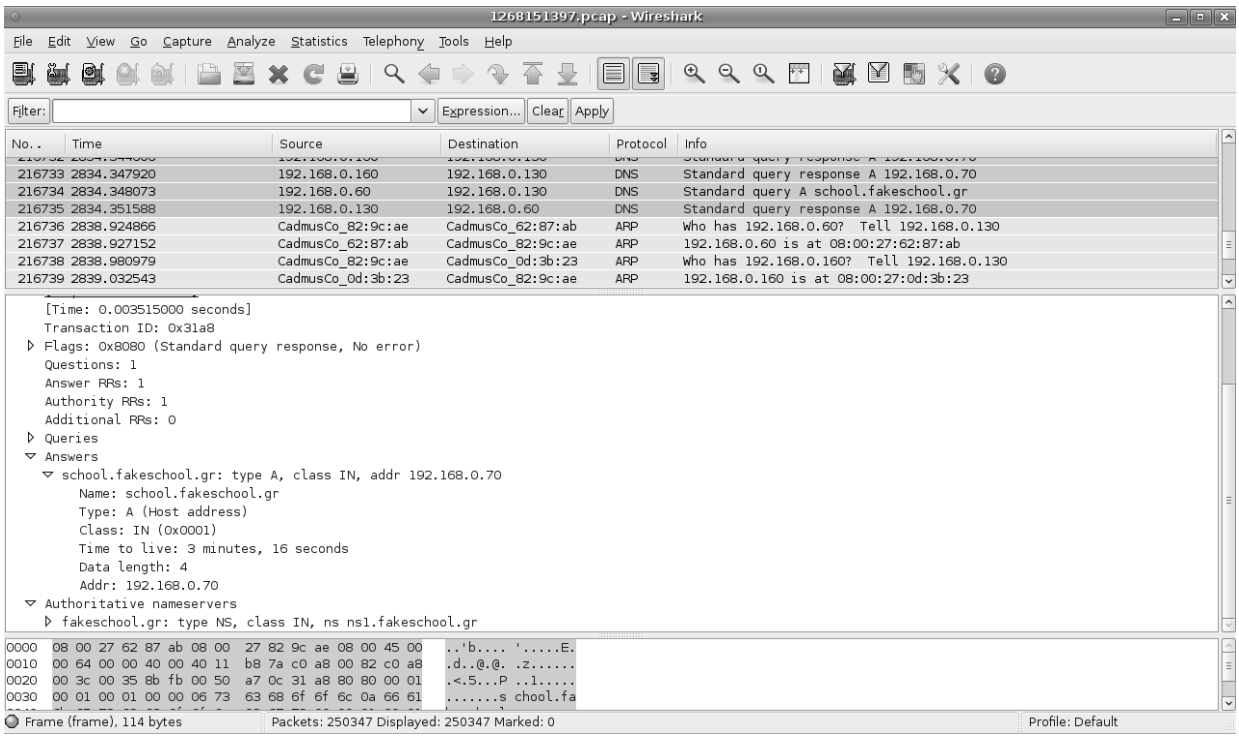

Σχήμα 5.12: Απάντηση για το όνομα χώρου “YDQPg3Jg21Bb4veDmiF.fakeschool.gr.”




Σχήμα 5.13: Ερώτημα για το όνομα χώρου “school.fakeschool.gr”

Θα πρέπει να επισημάνουμε πως όλες οι απαντήσεις που φαίνονται στο σχήμα 5.11 είναι παρόμοιες και η μόνη διαφορά είναι ότι φέρουν διαφορετικά αναγνωριστικά. Μάλιστα, όπως φαίνεται αυτά τα αναγνωριστικά αυξάνουν προοδευτικά κατά ένα ανά απάντηση, αλλά βρίσκονται όλα εντός ενός συγκεκριμένου διαστήματος.

Τέλος προχωρώντας και εξετάζοντας όλη την κίνηση, βλέπουμε πως η επίθεση στέφεται τελικά με επιτυχία. Στο σχήμα 5.14 βλέπουμε την απάντηση που αποστέλλει ο recursive διακομιστής, σύμφωνα με την οποία το όνομα χώρου "school.fakeschool.gr" συσχετίζεται με την διεύθυνση IP "192.168.0.70". Αυτό σημαίνει πως ο recursive διακομιστής είχε εκείνη την δεδομένη στιγμή μια πλαστή εγγραφή στην μνήμη cache. Η συγκεκριμένη εγγραφή παρέμεινε στην μνήμη cache για χρονικό διάστημα διακοσίων δευτερολέπτων (τόσο είχε οριστεί το TTL από τον επιτιθέμενο χρήστη όπως φαίνεται στην εξέταση των πακέτων). Για το συγκεκριμένο χρονικό διάστημα ο recursive διακομιστής απέστελνε λανθασμένη πληροφορία στα ερωτήματα που δεχόταν για το όνομα χώρου "school.fakeschool.gr".




Σχήμα 5.14: Επιβεβαίωση επίθεσης από τον επιτιθέμενο χρήστη.

**Μεθοδολογία Επίθεσης**

Αυτό που επιχειρεί, λοιπόν, παραπάνω ο επιτιθέμενος χρήστης είναι μια επίθεση τύπου cache poisoning στον Target DNS server με σκοπό να παραλλάξει την εγγραφή "school.fakeschool.gr". Για να πραγματοποιήσει τη συγκεκριμένη επίθεση ο κακόβουλος χρήστης αποστέλλει όπως προείπαμε στον recursive διακομιστή ερωτήματα για ανύπαρκτα ονόματα χώρου. Ο recursive διακομιστής προσπαθώντας να επιλύσει τα συγκεκριμένα ερωτήματα, αποστέλλει εν συνεχεία ερωτήματα στον διακομιστή που είναι authoritative για τη ζώνη αυτή. Στο μεσοδιάστημα που ακολουθεί μεταξύ της αποστολής του ερωτήματος από τον recursive διακομιστή έως την έλευση της απάντησης από τον authoritative διακομιστή, ο επιτιθέμενος αποστέλλει όσο το δυνατό περισσότερες "spoofed" απαντήσεις (τοποθετώντας δηλαδή την διεύθυνση IP του authoritative διακομιστή στα πακέτα που αποστέλλει) υποδυόμενος τον authoritative διακομιστή (ή τους authoritative αν είναι πολλοί), προσπαθώντας να μαντέψει το αναγνωριστικό που χρησιμοποιήθηκε στην επικοινωνία (του recursive διακομιστή με τον authoritative διακομιστή). Όσο το αναγνωριστικό δεν συμπίπτει με αυτό που χρησιμοποιήθηκε, ο recursive διακομιστής απορρίπτει τα μηνύματα εκλαμβάνοντας τα ως λανθασμένα. Όταν το αναγνωριστικό που θα αποστείλει ο επιτιθέμενος συμπέσει με αυτό που χρησιμοποιήθηκε στην επικοινωνία recursive-authoritative και η απάντηση φτάσει πριν από εκείνη του υπεύθυνου διακομιστή, η επίθεση στέφεται με επιτυχία.

**Κενό ασφαλείας**

Το κενό ασφαλείας που εκμεταλλεύεται η συγκεκριμένη επίθεση αφορά λάθος σχεδιασμού του DNS πρωτοκόλλου. Συγκεκριμένα, το μοναδικό αναγνωριστικό που προστίθεται σε κάθε επικοινωνία μέσω του DNS πρωτοκόλλου είναι μεγέθους 16 bit. Αυτό σημαίνει πως το αναγνωριστικό μπορεί να λάβει μια τιμή μεταξύ του "0" και του "65535". Μπορεί η επιλογή αναγνωριστικού, για κάθε επικοινωνία, να γίνεται τυχαία από το συγκεκριμένο σύνολο, το εύρος όμως των τιμών που το αναγνωριστικό αυτό μπορεί να λάβει δεν αποτρέπει ενδεχόμενες επιθέσεις σωστής του πρόβλεψης.

Συγκεκριμένα η διαδικασία που περιγράφηκε πιο πάνω και την οποία ανακάλυψε ο ερευνητής

Dan Kaminsky δίνει το απαιτούμενο χρονικό περιθώριο στον επιτιθέμενο χρήστη να αποστείλει ένα μεγάλο αριθμό μηνυμάτων και να μαντέψει επιτυχώς το αναγνωριστικό.

**Λύση του Προβλήματος**

Η λύση που προτάθηκε στο πρόβλημα είναι η τυχαία επιλογή UDP "source" πόρτας κατά την αποστολή μηνυμάτων. Με αυτό τον τρόπο δεν εξαλείφεται η περίπτωση κάποια επίθεση να στεφθεί με επιτυχία (μέσω σωστής πρόβλεψης αναγνωριστικού), απλώς μειώνεται σημαντικά η πιθανότητα σωστής απάντησης από την πλευρά του επιτιθέμενου (από 1/ 65.536 για κάθε απάντηση γίνεται πλέον 1/(65.536 * 65.536) ). Χρήση αυτής της λύσης κάνουν εξ ορισμού πλέον όλες οι νεότερες εκδόσεις του DNS λογισμικού Bind. Η πιο αποτελεσματική μέθοδος ασφάλισης εναντίων αυτής της επίθεσης αλλά και παρόμοιων είναι η χρήση του DNSSEC (DNS Security).

# Συμπεράσματα

Ολοκληρώνοντας την παρούσα εργασία και έχοντας μελετήσει διεξοδικά τα ζητήματα που πραγματεύτηκε, μπορούμε να εξάγουμε κάποια συμπεράσματα.

1. 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.

2. 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.

3. 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.

4. Ο συνδυασμός των εγκατεστημένων με αλλά συστήματα ασφαλείας και εν προκειμένω με τα IDS αποδείχθηκε μια καλή επιλογή. Ο συνδυασμός αυτός βοηθάει το διαχειριστή να ξεχωρίσει την ύποπτη και κακόβουλη δικτυακή κίνηση από την φυσιολογική. Θα πρέπει όμως το IDS είναι προσφάτως ενημερωμένο καθώς υπάρχει πάντα η πιθανότητα ορισμένες επιθέσεις να περάσουν απαρατήρητες.

5. 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.

# Επίλογος

Ο τομέας της ασφάλειας πληροφοριακών συστημάτων είναι ένας συνεχώς μεταβαλλόμενος χώρος γεμάτος προκλήσεις. Η παρούσα εργασία κάλυψε ένα μικρό μόνο κομμάτι, του τομέα αυτού, με την μελέτη, τόσο σε θεωρητικό όσο και σε πρακτικό επίπεδο, της θεωρίας των Honeypots. Ελπίζω η παρούσα εργασία να μεταδώσει στους αναγνώστες της πληροφορίες και γνώσεις, που θα τους φανούν χρήσιμα στο πλαίσιο κατανόησης των ζητημάτων που πραγματεύτηκε. Για εμένα η συγγραφή της αποτέλεσε μια σπουδαία εμπειρία.

Τρούλης Ιωάννης

# ΠΑΡΑΡΤΗΜΑ Α

## Παράρτημα Α1

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

```
create default
set default personality "Microsoft Windows XP Professional SP1"
set default uptime 1006630
add default tcp port 80 "/usr/share/honeyd/scripts/win32/win2k/iis.sh"
add default tcp port 22 open
add default tcp port 135 open
add default tcp port 21
"/usr/share/honeyd/scripts/win32/win2k/msftp.sh $ipsrc $sport $ipdst
$dport"

add default tcp port 143
"/usr/share/honeyd/scripts/win32/win2k/exchange-imap.sh $ipsrc $sport
$ipdst $dport"

add default tcp port 25
"/usr/share/honeyd/scripts/win32/win2k/exchange-smtp.sh $ipsrc $sport
$ipdst $dport"

add default tcp port 137 open
add default tcp port 138 open
add default tcp port 139 open
add default tcp port 445 tarpit open
add default tcp port 1433 tarpit open
add default udp port 1434 tarpit open
add default tcp port 1080 "/usr/share/honeyd/scripts/mydoom.pl"
```

```
add default tcp port 3127 "/usr/share/honeyd/scripts/mydoom.pl"
add default tcp port 3128 "/usr/share/honeyd/scripts/mydoom.pl"
add default tcp port 10080 "/usr/share/honeyd/scripts/mydoom.pl"
add default tcp port 4444 "/usr/share/honeyd/scripts/4444.sh $ipsrc
$ipdst"
add default tcp port 5554 "/usr/share/honeyd/scripts/lsass4.sh $ipsrc
$ipdst"
add default tcp port 9996 "/usr/share/honeyd/scripts/lsass4.sh $ipsrc
$ipdst"
add default tcp port 8967 "/usr/share/honeyd/scripts/cmdexe.pl -p
winxp -l /usr/share/honeyd/log/cmd/cmdexe"

add default tcp port 20168 "/usr/share/honeyd/scripts/cmdexe.pl -p
winxp -l /usr/share/honeyd/log/cmd/cmdexe "

add default tcp port 3117 "/usr/share/honeyd/scripts/cmdexe.pl -p
winxp -l /usr/share/honeyd/log/cmd/cmdexe "

set default default tcp action  reset
set default default udp action  reset
set default default icmp action  open


create router
set router personality "Cisco 1601R router running IOS 12.1(5)"
set router default tcp action reset
set router default udp action  reset
add router tcp port 22 "/usr/share/honeyd/scripts/test.sh"
add router tcp port 23 "/usr/share/honeyd/scripts/router-telnet.pl"
set router uptime 5504689


create adsl_router
set adsl_router personality "DLink DI-604 ethernet router"
```

```
set adsl_router default tcp action reset
set adsl_router default udp action reset
set adsl_router default icmp action open
add adsl_router tcp port 22 open
add adsl_router tcp port 23 "/usr/share/honeyd/scripts/router-
telnet.pl"
add adsl_router tcp port 80 open
set router uptime 3002545


create linux
set linux personality "Linux kernel 2.4.20"
add linux tcp port 80
"/usr/share/honeyd/scripts/unix/linux/suse8.0/apache.sh $ipsrc $sport
$ipdst $dport"

add linux tcp port 21
"/usr/share/honeyd/scripts/unix/linux/suse8.0/proftpd.sh $ipsrc $sport
$ipdst $dport"
add linux tcp port 25
"/usr/share/honeyd/scripts/unix/linux/suse8.0/sendmail.sh $ipsrc
$sport $ipdst $dport"

add linux tcp port 22
"/usr/share/honeyd/scripts/unix/linux/suse8.0/ssh.sh $ipsrc $sport
$ipdst $dport"

add linux tcp port 110
"/usr/share/honeyd/scripts/unix/linux/suse8.0/qpop.sh $ipsrc $sport
$ipdst $dport"

add linux tcp port 143
"/usr/share/honeyd/scripts/unix/linux/suse8.0/cyrus-imapd.sh $ipsrc
$sport $ipdst $dport"
```

```
add linux tcp port 8080
"/usr/share/honeyd/scripts/unix/linux/suse8.0/squid.sh $ipsrc $sport
$ipdst $dport"

add linux udp port 514
"/usr/share/honeyd/scripts/unix/linux/suse8.0/syslogd.sh $ipsrc $sport
$ipdst $dport"

set linux default tcp action reset
set linux default udp action reset
set linux default icmp action open
set linux uptime 8740375


create Solaris
set Solaris personality "Sun Solaris 9"
set Solaris default tcp action reset
set Solaris default udp action reset
set Solaris default icmp action open
add Solaris tcp port 25 "/usr/share/honeyd/scripts/smtp.pl -q"
add Solaris tcp port 8080 "/usr/share/honeyd/scripts/proxy.pl"
set Solaris uptime 47583955


create linux2
set linux2 personality "Linux 2.5.25 - 2.5.70 or Gentoo 1.2 Linux
2.4.19 rc1-rc7)"
set linux2 default tcp action reset
set linux2 default udp action reset
set linux2 default icmp action open
add linux2 tcp port 21 "/usr/share/honeyd/scripts/unix/linux/ftp.sh"
add linux2 tcp port 22
"/usr/share/honeyd/scripts/unix/linux/suse8.0/ssh.sh"
```

```
add linux2 tcp port 25 "/usr/share/honeyd/scripts/suse8.0/sendmail.sh
$ipsrc $sport $ipdst $dport"

add linux2 tcp port 80 open

add linux2 udp port 161 "/usr/share/honeyd/scripts/snmp/fake-snmp.pl
$ipsrc $sport $ipdst $dport"

add linux2 tcp port 3306 tarpit open
add linux2 tcp port 6881 tarpit open
add linux2 tcp port 6882 tarpit open
add linux2 tcp port 8080 tarpit open
set linux2 uptime 2800567


create mail_server
set mail_server personality "Linux 2.5.25 - 2.5.70 or Gentoo 1.2 Linux
2.4.19 rc1-rc7)"
set mail_server default tcp action reset
set mail_server default udp action reset
set mail_server default icmp action open
add mail_server tcp port 23 "/usr/share/honeyd/scripts/router-
telnet.pl"
add mail_server tcp port 25 "/usr/share/honeyd/scripts/sendmail.sh
$ipsrc $sport $ipdst $dport"
add mail_server tcp port 80 open

add mail_server tcp port 110
"/usr/share/honeyd/scripts/unix/general/pop/pop3.sh $ipsrc $sport
$ipdst $dport"

add mail_server tcp port 143
"/usr/share/honeyd/scripts/unix/linux/suse8.0/cyrus-imapd.sh $ipsrc
$sport $ipdst $dport"
```

```
set mail_server uptime 3753605
bind XXX.XXX.234.93 router
bind XXX.XXX.234.94 default
bind XXX.XXX.234.95 linux
bind XXX.XXX.234.96 Solaris
bind XXX.XXX.234.97 adsl_router
bind XXX.XXX.234.98 linux2
bind XXX.XXX.234.99 mail_server
```

## Παράρτημα Α2

### Εκκίνηση των μηχανημάτων

Αρχικά ενεργοποιούμε όλα τα συστήματα σε όλους τους ξενιστές. Όλες οι υπηρεσίες είναι ρυθμισμένες και εκκινούν αυτόματα κατά το χρόνο εκκίνησης των λειτουργικών συστημάτων.

Η μόνη υπηρεσία την οποία και θα χρειάστηκε να εκκινήσουμε χειροκίνητα είναι η DNS υπηρεσία του Victim Web Server καθώς εκεί χρησιμοποιούμε μια πιο παλιά και ευάλωτη έκδοση του Bind. Για να εκκινήσουμε λοιπόν την DNS υπηρεσία στον Victim Web Server πληκτρολογούμε την παρακάτω εντολή:

```
named –d 2 –c /etc/bind/named.conf –g
```

Τα flags που υπάρχουν στην παρακάτω εντολή επεξηγούνται στον παρακάτω πίνακα:

-d Με αυτή την παράμετρο καθορίζεται πως το Bind θα ξεκινήσει σε κατάσταση αποσφαλμάτωσης. Θα παράγει δηλαδή αναλυτικότερες πληροφορίες για ο,τι προκύπτει από ότι αν εκκινούσε στην προκαθορισμένη κατάσταση. Ο αριθμός δίπλα δηλώνει το επίπεδο του debug mode. Όσο μεγαλύτερος ο αριθμός τόσο μεγαλύτερο το επίπεδο αποσφαλμάτωσης δηλαδή τόσο αναλυτικότερες οι πληροφορίες που

τυπώνονται.

-c Με αυτή την παράμετρο καθορίζουμε την διαδρομή στην οποία βρίσκεται το αρχείο παραμετροποίησης ( συνήθως named.conf ) το οποίο χρησιμοποιεί το Bind για την ρύθμιση του.

-g Με αυτή την παράμετρο καθορίζεται πως το Bind θα εκκινήσει σε verbose κατάσταση. Δηλαδή δεν θα τρέχει ως δαίμονας αλλά θα εκτυπώνει στην οθόνη όλα τα μηνύματα τα οποία παράγει.

Για την πραγματοποίηση της επίθεσης χρησιμοποιήσαμε το Bailiwicked_host module από το Metasploit framework 3.2. Για να επιλέξουμε το συγκεκριμένου module πληκτρολογούμε την παρακάτω εντολή στο metasploit console:

```
use spoof/dns/bailiwicked_host
```

Εν συνεχεία με τις εντολές `show options` και `set` βλέπουμε τα πεδία τα οποία υπάρχουν προς συμπλήρωση και τα συμπληρώνουμε με τις κατάλληλες τιμές αντίστοιχα.

Λόγω όμως του ότι όλα τα εικονικά μηχανήματα που είναι ευάλωτα στην επίθεση βρίσκονταν στο ίδιο φυσικό μηχάνημα ο χρόνος που μεσολαβούσε για τις μεταξύ τους επικοινωνίες ήταν πολύ μικρός με αποτέλεσμα η επίθεση να μην μπορούσε να επιτευχθεί. Γι'αυτό το λόγο προσθέσαμε μια καθυστέρηση που πλησιάζει αυτή πραγματικών συνθηκών στον victim DNS server. Με την παρακάτω εντολή θα προσθέσουμε καθυστέρηση 50ms στην εξυπηρέτηση όλων των πακέτων που καταφθάνουν στο eth1 interface:

```
tc qdisc add dev eth1 root netem delay 50ms
```

Η χρονική καθυστέρηση των 50ms θα δώσει το απαραίτητο χρονικό περιθώριο στον επιτιθέμενο χρήστη να επιτύχει το σκοπό του και να προσθέσει την λανθασμένη εγγραφή τον Victim DNS server.

# ΠΑΡΑΡΤΗΜΑ Β

## Ρυθμίσεις για τα συστήματα

Αρχικά θα περιγράψουμε τις ρυθμίσεις που έγιναν στα μηχανήματα του Host Anafi και μετέπειτα του Host Schinousa. Έχουμε λοιπόν τα εξής:

**Target DNS Server**

Ο target DNS server όπως ειπώθηκε και πιο πάνω ήταν ο recursive server ο οποίος χρησιμοποιήθηκε στην τοπολογία μας και στον οποίο έγινε η επίθεση. Για να είναι πετυχημένη η επίθεση θα έπρεπε να χρησιμοποιηθεί μια παλαιότερη έκδοση του BIND, καθώς οι νεότερες εκδόσεις έχουν κάποιο patch ώστε να μην είναι πλέον ευάλωτες στο Kaminsky bug. Γι' αυτό το λόγο κατεβάσαμε, κάναμε compile και εγκαταστήσαμε την έκδοση 9.2.0 του BIND. Η έκδοση αυτή του BIND είναι αρκετά παλιά και χρονολογείται από το 2004. Προκειμένου να ρυθμίσουμε τον server εκτελέσαμε τα παρακάτω βήματα:

1. Μετατρέπουμε τον server σε recursive προσθέτοντας στο αρχείο name.conf.options που βρίσκεται στο φάκελο /etc/bind/ την επιλογή «recursion yes;».

2. Τροποποιούμε το αρχείο /etc/bind/db.root έτσι ώστε όλοι οι root servers που είναι συγκεντρωμένοι σε αυτό το αρχείο να «δείχνουν» προς τον Root DNS Server που εμείς θέλουμε να χρησιμοποιήσουμε. Αλλάζουμε δηλαδή τις διευθύνσεις IP όλων των root servers με την IP διεύθυνση του Root DNS Server που θα εγκαταστήσουμε και θα χρησιμοποιήσουμε για την εκτέλεση του πειράματος.

3. Στις ρυθμίσεις του VirtualBox για τον Target DNS Server, στις επιλογές του δικτύου προσθέτουμε μια δεύτερη κάρτα δικτύου την οποία και επιλέγουμε να λειτουργεί ως bridged adapter. Εν συνεχεία τροποποιούμε το αρχείο /etc/network/interfaces και τοποθετούμε την στατική IP 192.168.0.130 του υποδικτύου 192.168.0.0/24.

4. Τοποθετούμε στο αρχείο /etc/resolv.conf τον nameserver που μόλις δημιουργήσαμε.

Τοποθετούμε δηλαδή την εγγραφή «nameserver 127.0.0.1».

**Root Server**

Ο Root Server που εγκαταστήσαμε στο εργαστήριο εξομοιώνει τη λειτουργία ενός πραγματικού root DNS server. Σε αυτήν την περίπτωση ο Root Server θα έχει μόνο μια εγγραφή για το ccTLD «.gr». Εφόσον θέλουμε να μετατρέψουμε τον παραπάνω σε root server θα πρέπει να του ορίσουμε να διαχειρίζεται την ζώνη «.». Να είναι υπεύθυνος για την επίλυση της τελείας που υπάρχει στο τέλος του domain name. Ύστερα αφού κοιτάξει τις εγγραφές μέσα στην ζώνη «.» θα ανακατευθύνει την κίνηση στον κατάλληλο TLD server. Για να ρυθμίσουμε τον Root Server ακολουθήσαμε τα παρακάτω βήματα:

1. Πρώτα δημιουργούμε το αρχείο "/etc/bind/named.conf.zones" στο οποίο θα αποθηκεύσουμε τις ζώνες τις οποίες θα «δηλώσουμε». Θα πρέπει να ορίσουμε στο Bind την ύπαρξη του συγκεκριμένου αρχείου τοποθετώντας στο αρχείο named.conf την πρόταση include "named.conf.zones".

2. Πρώτα «δηλώνουμε» στο αρχείο /etc/bind/named.conf.zones την ζώνη «.» εισάγοντας το παρακάτω τμήμα:

```
01    zone "." {
02    type master;
03    file "/etc/bind/zones/dot.db";
04    }
05
06    zone "0.168.192.in-addr.arpa" {
07          type slave;
08          file "0.168.192.in-addr.arpa";
09          masters { 192.168.0.160; };
10    }
```

Όπως φαίνεται από τη γραμμή 2 παραπάνω, ο root server θα είναι master για αυτή τη ζώνη. Στην τρίτη γραμμή δηλώνουμε το που θα βρίσκεται αποθηκευμένη η ζώνη.

**3.** Εν συνεχεία δημιουργούμε το φάκελο /etc/bind/zones και μέσα στο φάκελο αυτό δημιουργούμε το αρχείο dot.db. Σε αυτό το αρχείο αποθηκεύουμε την ζώνη «.» η οποία και έχει ως εξής:

```
01  ;This is the file for the ".” zone
02  $ORIGIN .
03  $TTL 86400
04  ;SOA
05  @ IN    SOA   root-server.net. admin.root-server.net. (
06                          2009121401 ; Serial Number
07                          28800 ; Refresh
08                          3600       ; Retry
09                          604800     ; Expire
10                          10800      ; Ncache TTL
11                          )
12
13  ;. zone
14  --------
15  .                       IN    NS    root-server.net.
16  root-server.net.        IN    A     192.168.0.140
17
18  ;TLDs
19  --------
20  gr.                     IN    NS    tld.gr.
21  tld.gr            IN    A     192.168.0.150
```

**4.** Στις ρυθμίσεις του VirtualBox για τον Target DNS Server, στις επιλογές του δικτύου προσθέτουμε μια δεύτερη κάρτα δικτύου, την οποία και επιλέγουμε να λειτουργεί ως bridged adapter. Εν συνεχεία τροποποιούμε το αρχείο /etc/network/interfaces και τοποθετούμε την στατική IP 192.168.0.140 του υποδικτύου 192.168.0.0/24.

**5.** Τοποθετούμε στο αρχείο “/etc/resolv.conf” τον nameserver που μόλις δημιουργήσαμε. Τοποθετούμε δηλαδή την εγγραφή «nameserver 127.0.0.1» που πρόκειται για την

διεύθυνση του localhost.

**TLD Server**

Ο επόμενος server που έπρεπε να ρυθμίσουμε ήταν ο TLD server, ο οποίος θα είναι authoritative για τη ζώνη «gr.». Μέσα στη ζώνη αυτή θα υπάρχει μία εγγραφή για το domain name fakezone.gr. Για να ρυθμίσουμε τον TLD server ακολουθήσαμε τα παρακάτω βήματα:

1. Πρώτα δημιουργήσαμε όπως και στην περίπτωση του root server το αρχείο /etc/bind/named.conf.zones, στο οποίο «δηλώνουμε» τις ζώνες τις οποίες θα δημιουργήσουμε. Θα πρέπει να ορίσουμε στο Bind την ύπαρξη του συγκεκριμένου αρχείου τοποθετώντας στο αρχείο named.conf την πρόταση include “named.conf.zones”.

2. Ακολούθως «δηλώνουμε» στο αρχείο “/etc/bind/named.conf.zones” την ζώνη «gr.» εισάγοντας το παρακάτω τμήμα:

```
01   zone “gr.” {
02    type master;
03         file “/etc/bind/zones/gr.db”;
04         };
05
06   zone “0.168.192.in-addr.arpa” {
07         type slave;
08         file “0.168.192.in-addr.arpa”;
09         masters { 192.168.0.160; };
10    }
```

3. Εν συνεχεία δημιουργούμε τον φάκελο /etc/bind/zones και μέσα σε αυτόν το αρχείο gr.db. Το αρχείο gr.db είναι το αρχείο της ζώνης «.gr» μέσα στο οποίο αποθηκεύουμε την ζώνη η οποία έχει την εξής μορφή:

```
01  ;This is the file for the gr. zone
```

```
02  $ORIGIN gr.
03  $TTL 86400
04  ;SOA
05  @      IN    SOA   tld.gr. root.tld.gr. (
06                          2009121401 ; Serial Number
07                          28800 ; Refresh
08                          3600       ; Retry
09                          604800     ; Expire
10                          10800      ; Ncache TTL
11                          )
12
13  ;gr zone
14  --------------------
15                               IN    NS    tld.gr.
16  tld.gr.                      IN    A     192.168.0.150
17
18  ;fakeschool.gr zone
19  --------
20  fakeschool.gr.               IN    NS    ns1.fakeschool.gr.
21  ns1.fakeschool.gr.           IN    A     192.168.0.160
```

4. Στις ρυθμίσεις του VirtualBox για τον TLD Server, στις επιλογές του δικτύου προσθέτουμε μια δεύτερη κάρτα δικτύου την οποία και επιλέγουμε να λειτουργεί ως bridged adapter. Εν συνεχεία τροποποιούμε το αρχείο "/etc/network/interfaces" και τοποθετούμε την στατική IP 192.168.0.150 του υποδικτύου 192.168.0.0/24

5. Τοποθετούμε στο αρχείο "/etc/resolv.conf" τον nameserver που μόλις δημιουργήσαμε. Τοποθετούμε δηλαδή την εγγραφή «nameserver 127.0.0.1» που πρόκειται για την διεύθυνση του localhost.

**Victim DNS Server**

Ο επόμενος και τελευταίος DNS server που έπρεπε να ρυθμίσουμε ήταν ο Victim DNS Server ο

οποίος θα είναι authoritative για τη ζώνη “fakeschool.gr.”. Αυτή η ζώνη θα έχει κάποια εγγραφή για το subdomain ”school.fakeschool.gr” στο οποίο και θα πραγματοποιηθεί η επίθεση. Για να ρυθμίσουμε τον TLD server ακολουθήσαμε τα παρακάτω βήματα:

1. Πρώτα δημιουργήσαμε όπως και στις παραπάνω περιπτώσεις το αρχείο “/etc/bind/named.conf.zones”, στο οποίο θα «δηλώσουμε» τις ζώνες τις οποίες πρόκειται να δημιουργήσουμε. Θα πρέπει να ορίσουμε στο Bind την ύπαρξη του συγκεκριμένου αρχείου τοποθετώντας στο αρχείο named.conf την πρόταση include “named.conf.zones”.

2. Ακολούθως «δηλώνουμε» στο αρχείο “/etc/bind/named.conf.zones” την ζώνη ”gr.” εισάγοντας το παρακάτω τμήμα:

```
01   zone “fakeschool.gr.” {
02    type master;
03         file “/etc/bind/zones/fakeschool.gr”;
04         }
05
06   zone “0.168.192 .in-addr.arpa” {
07         type master;
08         file “/etc/bind/zones/0.168.192.in-addr.arpa”;
09         }
```

3. Εν συνεχεία αφού δημιουργούμε τον φάκελο “/etc/bind/zones” όπως και στις προηγούμενες περιπτώσεις δημιουργούμε μέσα σε αυτόν τα αρχεία fakeschool.gr και 0.168.192.in-addr.arpa. Το αρχείο fakeschool.gr είναι το αρχείο της ζώνης «fakeschool.gr» μέσα στο οποίο αποθηκεύουμε την ζώνη η οποία θα έχει την εξής μορφή:

```
01  ;This is the file for the gr. zone
02  $ORIGIN gr.
03  $TTL 86400
04  ;SOA
```

```
05  @      IN    SOA   ns1.fakeschool.gr. root (
06                         2009121501 ; Serial Number
07                         28800      ; Refresh
08                         3600       ; Retry
09                         604800     ; Expire
10                         10800      ; Ncache TTL
11                         )
12
13  ;fakeschool.gr zone
14  -------------------
15  fakeschool.gr.         IN    NS    ns1.fakeschool.gr.
16  ns1.fakeschool.gr.     IN    A     192.168.0.160
17
18 ;Site
19  -----
20  www.school             IN    A     192.168.0.170
21  school                 IN    A     192.168.0.170
```

Ακολούθως και αντίστοιχα αποθηκεύουμε στο αρχείο 0.168.192.in-addr.arpa το οποίο θα έχει την εξής μορφή:

```
01  ;This is the file for the reverse zone
02  $ORIGIN 0.168.192.in-addr.arpa.
03  $TTL 86400
04  ;SOA
05  @      IN    SOA   ns1.fakeschool.gr. root (
06                         2009121501 ; Serial Number
07                         28800 ; Refresh
08                         7200       ; Retry
09                         604800     ; Expire
10                         86400 ; Ncache TTL
11                         )
12
13  ;Reverse zone
```

```
14  ;------------
15  @           IN    NS    ns1.fakeschool.gr.
16
17  ;Pointers
18  ;--------
19  160         IN    PTR   ns1.fakeschool.gr.
20  170         IN    PTR   school.fakeschool.gr.
21  170         IN    PTR   www.school.fakeschool.gr.
```

4. Στις ρυθμίσεις του VirtualBox για το μηχάνημα Victim Server, στις επιλογές του δικτύου προσθέτουμε μια δεύτερη κάρτα δικτύου την οποία και επιλέγουμε να λειτουργεί ως bridged adapter. Εν συνεχεία τροποποιούμε το αρχείο "/etc/network/interfaces" και τοποθετούμε την στατική IP 192.168.0 .160 του υποδικτύου 192.168.0.0/24

5. Τοποθετούμε στο αρχείο "/etc/resolv.conf" τον nameserver που μόλις δημιουργήσαμε. Τοποθετούμε δηλαδή την εγγραφή «nameserver 127.0.0.1» που πρόκειται για την διεύθυνση του localhost.

**Victim Web Site**

Το Victim Web Site είναι το μηχάνημα στο οποίο εγκαταστάθηκε ο Apache web server, προκειμένου να εμφανίζεται η πραγματική ιστοσελίδα, της οποίας η κίνηση θα ανακατευθυνθεί σε κάποια άλλη διεύθυνση IP λόγω της αδυναμίας που υπάρχει και την οποία εκμεταλλεύεται το Kaminsky bug. Προκειμένου να ρυθμίσουμε τον Apache server ακολουθούμε τα παρακάτω βήματα:

1. Κάνουμε copy το αρχείο /etc/apache2/sites-available/default και το μετονομάζουμε σε ότι εμείς θέλουμε. Το αρχείο default είναι default από την εγκατάσταση του Apache server και περιέχει τις ρυθμίσεις για το default site που εμφανίζει ο Apache server.

2. Αφού μετονομάσουμε το αρχείο το παραλλάσσουμε κατάλληλα πραγματοποιώντας τις κατάλληλες ρυθμίσεις:
    - Αλλάζουμε την διαδρομή μπροστά από το DocumentRoot τοποθετώντας την

κατάλληλη στην οποία βρίσκεται ο φάκελος στον οποίο έχουμε αποθηκευμένη την ιστοσελίδα μας. Στην προκειμένη περίπτωση αλλάζουμε τη διαδρομή ως εξής:

```
DocumentRoot      /home/mysite
```

- Αλλάζουμε επίσης την διαδρομή μπροστά από το Directory τοποθετώντας και πάλι την κατάλληλη στην οποία βρίσκεται αποθηκευμένη η ιστοσελίδα μας. Στην προκειμένη περίπτωση αλλάζουμε ως τη διαδρομή ως εξής:

```
<Directory         /home/mysite/>
```

2. Εν συνεχεία απενεργοποιούμε το default site και ενεργοποιούμε το το victim site με τις παρακάτω εντολές:
    - Με την εντολή **a2dissite default** απενεργοποιούμε το default site.
    - Με την εντολή **a2ensite mysite** ενεργοποιούμε το victim site mysite.

Οι παραπάνω εντολές θα πρέπει να εκτελεστούν με δικαιώματα root.

**Attacker**

Το μηχάνημα attacker είναι αυτό από το οποίο έγινε η επίθεση για την εκμετάλλευση του Kaminsky bug. Γι' αυτό το λόγο εγκαταστάθηκε σε αυτό το μηχάνημα το Metasploit 3.2 το οποίο περιέχει το module που πραγματοποιεί την επίθεση. Η πιο νέα έκδοση του Metasploit είναι η 3.3, αλλά λόγω προβλημάτων στην εκτέλεση της επίθεσης επιλέχθηκε η έκδοση 3.2. Το Metasploit δεν χρειάζεται καμία ιδιαίτερη ρύθμιση. Αρκεί να αποσυμπιέσει κάποιος το αρχείο της έκδοσης που έκανε download και εν συνεχεία αφού εισέλθει στο φάκελο που το αποσυμπίεσε να το χρησιμοποιήσει.

**Malicious Web Site**

Το Malicious Web Site είναι το μηχάνημα το οποίο θα φιλοξενεί το παράνομο web site του επιτιθέμενου, στο οποίο θα ανακατευθυνθεί η κίνηση αφού πραγματοποιηθεί η επίθεση. Όπως και στην περίπτωση του Victim Web Site παραπάνω χρειάστηκε να ρυθμίσουμε τον Apache server κατάλληλα ώστε να είναι προσβάσιμο το web site. Προκειμένου να ρυθμίσουμε τον Apache server ακολουθούμε τα παρακάτω βήματα:

1. Κάνουμε copy το αρχείο "/etc/apache2/sites-available/default" και το μετονομάζουμε σε ότι εμείς θέλουμε. Το αρχείο default είναι προκαθορισμένο από την εγκατάσταση του Apache server και περιέχει τις ρυθμίσεις για το default site που εμφανίζει ο Apache server.
2. Αφού μετονομάσουμε το αρχείο το παραλλάσσουμε κατάλληλα πραγματοποιώντας τις κατάλληλες ρυθμίσεις:
    - Αλλάζουμε την διαδρομή μπροστά από το DocumentRoot τοποθετώντας την κατάλληλη στην οποία βρίσκεται ο φάκελος στον οποίο έχουμε αποθηκευμένη την ιστοσελίδα μας. Στην προκειμένη περίπτωση αλλάζουμε ως τη διαδρομή ως εξής:

      ```
      DocumentRoot      /home/MaliciousSite
      ```
    - Αλλάζουμε επίσης την διαδρομή μπροστά από το Directory τοποθετώντας και πάλι την κατάλληλη στην οποία βρίσκεται αποθηκευμένη η ιστοσελίδα μας. Στην προκειμένη περίπτωση αλλάζουμε ως τη διαδρομή ως εξής:

      ```
      Directory         /home/MaliciousSite/
      ```
3. Εν συνεχεία απενεργοποιούμε το default site και ενεργοποιούμε το victim site με τις παρακάτω εντολές:
    - Με την εντολή **a2dissite default** απενεργοποιούμε το default site.
    - Με την εντολή **a2ensite mysite** ενεργοποιούμε το victim site MaliciousSite

Οι παραπάνω εντολές θα πρέπει να εκτελεστούν με δικαιώματα root.

# ΠΑΡΑΡΤΗΜΑ Γ

Παρακάτω στον πίνακα Γ.1 παρατίθενται πληροφορίες για όλες τις πόρτες που χρησιμοποιήθηκαν συνολικά από όλα τα honeypots.

Πίνακας Γ.1: Δεσμευμένες θύρες που χρησιμοποιήθηκαν και υπηρεσίες

| **Πόρτα** | **Πρωτό-κολλο** | **Υπηρεσία** |
|---|---|---|
| 21 | TCP | Είναι η προκαθορισμένη πόρτα που χρησιμοποιείται για τις FTP συνδέσεις. |
| 22 | TCP | Είναι η προκαθορισμένη πόρτα που χρησιμοποιείται για τις SSH συνδέσεις. |
| 23 | TCP | Είναι η προκαθορισμένη πόρτα που χρησιμοποιείται για τις Telnet συνδέσεις. |
| 25 | TCP | Είναι η προκαθορισμένη πόρτα που χρησιμοποιεί το SMTP (Simple Mail Transfer Protocol). |
| 80 | TCP | Είναι η πόρτα που χρησιμοποιείται για τις συνδέσεις HTTP, δηλαδή ότι έχει σχέση με λήψη και αποστολή δεδομένων από και προς το διαδίκτυο. |
| 110 | TCP | Είναι η πόρτα που χρησιμοποιείται για αποστολή και λήψη ηλεκτρονικού ταχυδρομείου μέσω του πρωτοκόλλου POP3. |
| 135 | TCP | Είναι η πόρτα που χρησιμοποιείται ως RPC (Microsoft Remote Procedure Call). |
| 137 | TCP | Είναι η πόρτα που χρησιμοποιείται από το name service του NetBIOS. |
| 138 | TCP | Είναι η πόρτα που χρησιμοποιείται από το Datagram Service του NetBIOS. |
| 139 | TCP | Είναι η πόρτα που χρησιμοποιείται από το Session Service του NetBIOS. |
| 143 | TCP | Είναι η πόρτα που χρησιμοποιείται για αποστολή και λήψη ηλεκτρονικού ταχυδρομείου μέσω του πρωτοκόλλου IMAP. |
| 161 | UDP | Είναι η πόρτα που χρησιμοποιείται από το SNMP (Simple Network |

| | | |
|---|---|---|
| | | Management Protocol). |
| 445 | TCP | Είναι η πόρτα που χρησιμοποιείται από το πρωτόκολλο SMB (Server Message Block). Το SMB χρησιμοποιείται για κοινή χρήση εκτυπωτών, συσκευών, αρχείων κ.α. και είναι ιδιαίτερα ευάλωτη σε επιθέσεις. |
| 514 | UDP | Είναι η πόρτα που χρησιμοποιείται για να ανοίξει κάποιο shell στο απομακρυσμένο σύστημα για την εκτέλεση κάποιων εντολών. |
| 1080 | TCP | Είναι η πόρτα που χρησιμοποιείτε από το πρωτόκολλο SOCKS |
| 1433 | TCP | Είναι η πόρτα που χρησιμοποιείται από τον Microsoft SQL Server για απομακρυσμένες συνδέσεις στη βάση δεδομένων. |
| 1434 | UDP | Είναι η πόρτα που χρησιμοποιείται για συνδέσεις προς τον Microsoft SQL Server ώστε να παρακολουθήσει ο χρήστης τις βάσεις δεδομένων. Είναι επίσης η πόρτα που χρησιμοποιείται από το worm Slammer. |
| 3117 | TCP | Είναι η πόρτα που χρησιμοποιείτε από την υπηρεσία mctet-jserv. |
| 3127 | TCP | Είναι η πόρτα που χρησιμοποιείτε από τον γνωστό ιό mydoom. Εκτός από τον mydoom βέβαια η πόρτα αυτή αποτελεί στόχο αρκετών ακόμα malware προγραμμάτων. |
| 3128 | TCP | Είναι η πόρτα που χρησιμοποιείτε από το trojan RingZero. |
| 3306 | TCP | Είναι η πόρτα που χρησιμοποιείτε από την MySQL. Χρησιμοποιείται επίσης και από πολλά MySQL Bots. |
| 4444 | TCP | Είναι μία από τις πόρτες τις οποίες χρησιμοποιεί το worm Mblaster και μέσω της οποίας επιτρέπει απομακρυσμένες συνδέσεις στα μολυσμένα συστήματα. |
| 5554 | TCP | Η πόρτα αυτή μαζί με την 9996 χρησιμοποιούνται από το worm Sasser. Συγκεκριμένα αφού το Sasser καταφέρει και μολύνει κάποιο σύστημα εν συνεχεία «ανοίγει» στις πόρτες 5554 και 9996 κάποια command prompt για την εκτέλεση κάποιων εντολών. |
| 6881 | TCP | Είναι η πόρτα που χρησιμοποιείτε από peer-to-peer προγράμματα |
| 6882 | TCP | Είναι η πόρτα που χρησιμοποιείτε από peer-to-peer προγράμματα |

| | | |
|---|---|---|
| 8080 | TCP | Είναι η πόρτα που χρησιμοποιείτε ως εναλλακτική της θύρας 80. |
| 8967 | TCP | Είναι η πόρτα που χρησιμοποιείτε από το worm Dapper ως backdoor. |
| 9996 | TCP | Είναι όπως προαναφέρθηκε μία από τις πόρτες που χρησιμοποιεί το worm Sasser. Συγκεκριμένα αφού το Sasser καταφέρει και μολύνει κάποιο σύστημα εν συνεχεία «ανοίγει» στις πόρτες 5554 και 9996 κάποια command prompt για την εκτέλεση εντολών. |
| 10080 | TCP | Είναι μία από τις πόρτες που χρησιμοποιεί το worm Mydoom ως backdoor. |
| 20168 | TCP | Η πόρτα αυτή χρησιμοποιείται από κάποιο worm για την πραγματοποίηση TFTP συνδέσεων. |

# ΠΑΡΑΡΤΗΜΑ Δ

**Επεξήγηση του "Bailiwicked_server Module"**

Το Metasploit Framework, όπως προαναφέρθηκε και στο κεφάλαιο πέντε της εργασίας, περιλαμβάνει δύο modules για την εκτέλεση της επίθεσης. Κατά το πείραμα στο Εργαστήριο χρησιμοποιήθηκε το module "Bailiwicked_host" και έγινε πλήρης ανάλυση των αποτελεσμάτων που εκτέλεση του επιφέρει. Εκτελώντας το συγκεκριμένο module ένας χρήστης είναι σε θέση να τοποθετήσει στη μνήμη cache ενός recursive διακομιστή μεμονωμένες πλαστές εγγραφές για κάποιο όνομα χώρου. Αντίθετα με τη χρήση του module Bailiwicked_server προστίθενται στην μνήμη cache εγγραφές που καθορίζουν τον authoritative διακομιστή για κάποιο όνομα χώρου. Μια τέτοιου είδους επίθεση έχει πολύ μεγαλύτερο αντίκτυπο από ότι αυτή που πραγματοποιήσαμε στο εργαστήριο.

Η διαφοροποίηση του module Bailiwicked_server από το Bailiwicked_host ως προς την εκτέλεση της επίθεσης εντοπίζεται στο Additional κομμάτι. Συγκεκριμένα, όπως διαπιστώσαμε από την εκτέλεση της επίθεσης στο Εργαστήριο, το Bailiwicked_host τοποθετεί στο Additional κομμάτι μια πλαστή εγγραφή με την οποία συσχετίζει το όνομα χώρου το οποίου επιθυμούμε να πάρουμε υπό έλεγχο με μια νέα διεύθυνση IP. Το module Bailiwicked_server από την άλλη τοποθετεί στο Additional κομμάτι δύο εγγραφές με σκοπό να καθορίσει ως authoritative διακομιστή για κάποιο όνομα χώρου (συγκεκριμένα για κάποια ζώνη) έναν διακομιστή ελεγχόμενο από τον επιτιθέμενο χρήστη.

Πραγματοποιώντας μια επιτυχημένη επίθεση με τη χρήση του module Bailiwicked_server, ένας χρήστης έχει τη δυνατότητα να εκμεταλλευτεί όλα τα sub-domains που βρίσκονται, ιεραρχικά, κάτω από το συγκεκριμένο όνομα χώρου παραπλανώντας ανυποψίαστους χρήστες. Επίσης έχει τη δυνατότητα να εκμεταλλευτεί και τις υπηρεσίες που εξυπηρετούνται με βάση το συγκεκριμένα domains και subdomains. Για παράδειγμα έστω πως ένας χρήστης πραγματοποιώντας τη συγκεκριμένη επίθεση επιτυγχάνει να τοποθετήσει στην μνήμη cache ενός recursive διακομιστή μια πλαστή εγγραφή για το όνομα χώρου "unipi.gr". Η

εγγραφή αυτή καθορίζει πως authoritative διακομιστής για τη ζώνη “unipi.gr” είναι κάποιος ο οποίος βρίσκεται υπό τον έλεγχο του επιτιθέμενου χρήστη. Μετά την επίτευξη της επίθεσης ο recursive διακομιστής θα ανακατευθύνει τα όποια ερωτήματα λαμβάνει προς επίλυση, στον παράνομο διακομιστή ονομάτων. Ο κακόβουλος χρήστης πλέον έχει τον έλεγχο ολοκλήρου του domain “unipi.gr” . Συγκεκριμένα:

1. Έχει τη δυνατότητα να συσχετίσει όλα τα subdomains του domain “unipi.gr” με νέες διευθύνσεις IP, μέσω της τοποθέτησης κατάλληλων εγγραφών. Για παράδειγμα, ο κακόβουλος χρήστης, μπορεί να ανακατευθύνει την κίνηση προς τους ιστοσελίδες “www.ds.unipi.gr” και “dtps.unipi.gr”, όπως και πολλών αντίστοιχων ακόμα, σε νέες σελίδες παραπλανητικές.

2. Έχει τη δυνατότητα να εκμεταλλευτεί την υπηρεσία ηλεκτρονικού ταχυδρομείου, λαμβάνοντας όλη την ηλεκτρονική αλληλογραφία που προορίζεται για χρήστες του συγκεκριμένου domain ή των subdomains όπως επίσης να αποστείλει ηλεκτρονική αλληλογραφία εκ μέρους τους.

# ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ


[1] Σωκράτης Κ. Κατσικας, Δημήτρης Γκρίτζαλης, Στέφανος Γκρίτζαλης, *Ασφάλεια Πληροφοριακών Συστημάτων*, Αθήνα 2004

[2] http://en.wikipedia.org/wiki/Information_security

[3] Chistos Douligeris, Dimitrios. N.Serpanos, *Network Security*, IEEE Press, 2007

[4] Αλεξανδράτος Γεώργιος, *Ανάλυση Δεδομένων από την Λειτουργία των Honeynets στο Εργαστήριο Islab του ΕΚΕΦΕ Δημόκριτος Μέθοδος, Εργαλεία και Βασικές Εννοιες*, Κεφάλαιο Εισαγωγή, σελίδα 4

[5] Camilo Viecco «Improving Honeynet Data Analysis», Proceedings of the 2002 IEEE Workshop on Information Assurance and Security T1B2 1555 United States Military Academy, West Point, NY, 17–19 June 2002

[6] Douglas E. Comer, *Internetworking with TCP/IP, Volume 1: Principles, Protocols and Architecture*, Κεφάλαιο 5, σελίδα 77

[7] Niels Provos, Thorsten Holtz, *Virtual Honeypots: From Botnet Tracking to Intrition Detection*, Addison-Wesley Professional, July 26, 2007

[8] Γώγουλος Μάρκος, *Ανάλυση Δικτυακών Επιθέσεων με το Honeyd*,Αθήνα, 2005

[9] http://en.wikipedia.org/wiki/Address_Resolution_Protocol

[10] The Honeynet Project, *Know Your Enemy*, Addison Wesley, 2004

[11] http://snort-inline.sourceforge.net/

[12] http://en.wikipedia.org/wiki/Tcpdump

[13] http://en.wikipedia.org/wiki/Wireshark

[14] http://www.wireshark.org/docs/man-pages/tshark.html

[15] http://en.wikipedia.org/wiki/SOCKS

[16] http://www.checkpoint.com/defense/advisories/public/2009/cpai-20-Jan.html#details

[17] Libor Dostalek, Alena Kabelova, *DNS in Action*, PACT Publishing, March, 2006

[18] Ronald G. F. Aitchison, *Pro DNS And Bind*, Apress, 2005

[19] http://en.wikipedia.org/wiki/Root_nameserver