Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	**Μελέτη Honeypot συστημάτων με πειραματική εφαρμογή.** **Honeypot Systems' study with experimental results.**
Ονοματεπώνυμο Φοιτητή	**Μαρίκα Πατρώνη**
Πατρώνυμο	**Αριστείδης**
Αριθμός Μητρώου	**ΜΠΣΠ/ 10015**
Επιβλέπων	**Χρήστος Δουληγέρης, Καθηγητής**

Ημερομηνία Παράδοσης **Οκτώβριος 2013**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)	(υπογραφή)	(υπογραφή)
Όνομα Επώνυμο Βαθμίδα	Όνομα Επώνυμο Βαθμίδα	Όνομα Επώνυμο Βαθμίδα

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή της εργασίας μου, Καθηγητή κύριο Χρήστο Δουληγέρη, για την εμπιστοσύνη που μου έδειξε αναθέτοντάς μου την παρούσα μεταπτυχιακή διατριβή, για την επίβλεψη και καθοδήγηση καθ' όλη τη διάρκεια εκπόνησής της. Ομοίως, να ευχαριστήσω ιδιαίτερα τον υποψήφιο Διδάκτορα κύριο Εμμανουήλ Γεωργακάκη για τη συνεργασία και συνεχή βοήθεια. Οι εύστοχες επισημάνσεις του, η συνεχής στήριξή του σε όλα τα επίπεδα και η αμεσότητά του συνέβαλαν καθοριστικά στην επίλυση των προβλημάτων που ανέκυπταν, στην εξέλιξη γενικότερα και ολοκλήρωση της παρούσας εργασίας.

Τέλος, ένα μεγάλο ευχαριστώ στο σύντροφό μου και την οικογένειά μου για τη στήριξη που μου παρείχαν όλα αυτά τα χρόνια των σπουδών μου.

Περίληψη

Η παρούσα μεταπτυχιακή διατριβή έχει ως αντικείμενο μελέτης τα Honeypot συστήματα και τη συμβολή τους στην Ασφάλεια Πληροφοριακών Συστημάτων. Ως Honeypots ορίζονται τα συστήματα που φαινομενικά μοιάζουν με πλήρως λειτουργικά υπολογιστικά συστήματα παραγωγής αλλά στην πραγματικότητα είναι προσομοιωτές ή πραγματικά συστήματα που παρουσιάζουν πληθώρα ευπαθειών ασφαλείας ώστε να προσελκύσουν εισβολείς. Στην ουσία λειτουργούν σα δολώματα προς τους εισβολείς για τη διενέργεια επιθέσεων.

Στα πλαίσια αυτά, εγκαταστάθηκαν, παραμετροποιήθηκαν και λειτούργησαν τρία Honeypot συστήματα. Τα αποτελέσματα των επιθέσεων καταγράφηκαν, μελετήθηκαν διεξοδικά και παρουσιάστηκαν με λεπτομέρεια σε γραφήματα και πίνακες.

Σκοπός της εφαρμογής ήταν η καταγραφή της συνολικής κίνησης αλληλεπίδρασης με τα Honeypot συστήματα ώστε να μελετηθεί η συμπεριφορά των επιτιθέμενων και η μεθοδολογία τους.

Συγκεκριμένα, υλοποιήθηκαν δύο προσομοιωτές διακομιστών ssh και http πρωτοκόλλου και ένας προσομοιωτής πληθώρας πρωτοκόλλων που δύναται να συλλέγει αρχεία κακόβουλου περιεχομένου. Και οι τρεις προσομοιωτές ως Honeypots εμφάνιζαν ευπάθειες ως προς την ασφάλειά τους και για αυτό το λόγο έγιναν αντικείμενο επίθεσης από εισβολείς. Τα δεδομένα που καταγράφηκαν από τις επιθέσεις αποθηκεύτηκαν σε βάσεις δεδομένων και στη συνέχεια με χρήση τεχνολογιών σχεσιακών βάσεων δεδομένων έγιναν αντικείμενο ανάλυσης και επεξεργασίας.

Στα πλαίσια ανάπτυξης αυτής της εφαρμογής μελετήθηκαν βασικές έννοιες Ασφάλειας Πληροφοριακών Συστημάτων, δικτυακές ευπάθειες καθώς και τύποι λογισμικού κακόβουλου περιεχομένου και σε συνάρτηση με αυτές τις έννοιες μελετήθηκαν, παρουσιάστηκαν και κατηγοριοποιήθηκαν τα σημαντικότερα και πιο πρόσφατα αναπτυγμένα Honeypot συστήματα.

Λέξεις – Κλειδιά: Honeypot, Honeynet, Ασφάλεια Πληροφοριακών Συστημάτων, Ασφάλεια Δικτύων, κακόβουλο λογισμικό, διαδικτυακές επιθέσεις

Abstract

The objective of this MSc thesis is to study Honeypot systems' functionality and how they contribute to Information Systems Security. Honeypots are defined as systems that resemble fully functional production computing systems but are actually emulators or real systems with several security vulnerabilities in order to attract attackers. Essentially they act as baits for the attackers to carry out attacks.

In this context, three Honeypot Systems were installed, configured and deployed, the results of which were studied and presented in detail in tables and charts.

The Honeypot Systems' application aimed at recording the whole interaction between Honeypots and attackers in order to study the attackers' behavior and methodology.

Specifically, an SSH emulator, an HTTP emulator and one that emulates a significant number of protocols and has the ability to collect malware, were deployed. All three emulators as Honeypots presented security vulnerabilities and for this reason they became subject to attacks. All related data gathered were stored in databases and were then analysed and processed by using relational database technologies.

Moreover, the basic concepts of Information Systems Security, network vulnerabilities and types of malware were studied and the most significant and recently developed Honeypot systems were presented and classified.

Keywords: Honeypot, Honeynet, Information Systems Security, Network Security, malware, internet attacks

Πίνακας περιεχομένων

Πίνακας εικόνων

Κατάλογος πινάκων

1. ΕΙΣΑΓΩΓΗ – ΣΥΝΤΟΜΗ ΠΕΡΙΓΡΑΦΗ ΠΡΟΒΛΗΜΑΤΟΣ

Η ραγδαία εξέλιξη των τεχνολογιών Πληροφορικής και Τηλεπικοινωνιών, η αναπόφευκτη ένταξή τους σχεδόν σε όλες τις εκφάνσεις της καθημερινότητας του ανθρώπου και η ανάγκη για ακριβή, αποτελεσματική και αξιόπιστη διαχείριση και μετάδοση πληροφορίας κατέστησαν την Ασφάλεια Πληροφοριακών Συστημάτων αναπόδραστο κομμάτι της χρήσης Πληροφοριακών Συστημάτων ώστε να προστατεύονται τα στοιχεία του Συστήματος και το Σύστημα στο σύνολό του από κάθε σκόπιμη ή τυχαία απειλή.

Ειδικότερα στον αχανή χώρο του Διαδικτύου που την τελευταία δεκαετία έχει εξαπλωθεί με φρενήρεις ρυθμούς και όπου η διακινούμενη πληροφορία δύναται να μην ελέγχεται για την ακριβή προέλευση και αξιοπιστία της, η Ασφάλεια και Προστασία της πληροφορίας απαιτείται για την εύρυθμη χρήση και λειτουργία του εκάστοτε συνδεόμενου Πληροφοριακού Συστήματος.

Για το λόγο αυτό, έχει αναπτυχθεί πληθώρα αντιμέτρων, πρακτικών και μοντέλων ασφαλείας και προγραμμάτων λογισμικού που σκοπό έχουν την προστασία της διαχείρισης και μετάδοσης πληροφορίας. Σε αυτά τα πλαίσια και με τον ίδιο σκοπό έχουν αναπτυχθεί και τα Honeypots, τα οποία μελετούνται στην παρούσα μεταπτυχιακή διατριβή.

Τα Honeypots λειτουργούν σα δολώματα που επιδιώκουν να προσελκύσουν εισβολείς. Είναι ουσιαστικά προγράμματα λογισμικού που συγκεντρώνουν πληθώρα ευπαθειών αλλά λειτουργούν απομονωμένα, επομένως κατορθώνουν την καταγραφή δραστηριότητας των επιτιθέμενων για μετέπειτα έρευνα και παρακολούθηση. Στον εισβολέα παρουσιάζονται σαν πραγματικά συστήματα παραγωγής που έχουν όμως ευπάθειες τις οποίες εκμεταλλεύονται οι εισβολείς.

Με δεδομένη την ολοένα αυξανόμενη διενέργεια διαδικτυακών επιθέσεων για υπολογιστικούς, οικονομικούς και πολιτικούς ενδεχομένως σκοπούς και με απαιτούμενη την ασφάλεια δικτύων και συστημάτων, κρίθηκε ενδιαφέρουσα και ουσιαστική η μελέτη και λειτουργία των Honeypots τόσο σε θεωρητικό όσο και πρακτικό επίπεδο.

Στην παρούσα μεταπτυχιακή διατριβή εγκαταστάθηκαν και μελετήθηκαν τρία Honeypots προκειμένου να μελετηθεί η λειτουργικότητά τους, να καταγραφούν οι επιθέσεις εναντίον τους, οι απόπειρες μη εξουσιοδοτημένης πρόσβασης και χρήσης εναντίον τους και να εξαχθούν ενδεχομένως σημαντικά συμπεράσματα για την Προστασία Πληροφοριακών Συστημάτων εν γένει. Απώτερος σκοπός της εγκατάστασής τους ήταν η απόκτηση γνώσης της μεθοδολογίας των εισβολέων τους.

1.1 Πλάνο εργασίας

Στο πρώτο κεφάλαιο της παρούσας μεταπτυχιακής διατριβής δίδεται μια εισαγωγή των ζητημάτων που πραγματεύεται η εργασία καθώς και η δομή της εργασίας.

Στο δεύτερο κεφάλαιο περιγράφονται βασικές έννοιες της Ασφάλειας Πληροφοριακών Συστημάτων, κύριες ευπάθειες δικτύων καθώς και κύριοι τύποι λογισμικού κακόβουλου περιεχομένου.

Στο τρίτο και το τέταρτο κεφάλαιο ορίζεται η έννοια του Honeypot συστήματος και δίδονται οι βασικές κατηγοριοποιήσεις των Honeypots.

Εν συνεχεία παρατίθενται τα πλεονεκτήματα και μειονεκτήματα της χρήσης των Honeypots ώστε να γίνει κατανοητή η συμβολή τους στην Ασφάλεια Πληροφοριακών Συστημάτων.

Στο έκτο κεφάλαιο περιγράφεται η λειτουργικότητα των σημαντικότερων και πιο πρόσφατα αναπτυγμένων Honeypots ανά κατηγορία και εν συνεχεία παρουσιάζονται τα Honeypots που χαρακτηρίζονται ως προηγμένα καθώς συνδυάζουν στοιχεία διαφορετικών κατηγοριών και διαθέτουν πολλαπλή λειτουργικότητα.

Στο όγδοο κεφάλαιο εξετάζονται ζητήματα αρχιτεκτονικής των Honeypots και στο ένατο κεφάλαιο εξετάζονται νομικά ζητήματα της χρήσης των Honeypots.

Στο δέκατο κεφάλαιο περιγράφεται η εφαρμογή που εκπονήθηκε στα πλαίσια της παρούσας μεταπτυχιακής διατριβής και η οποία περιλαμβάνει την εγκατάσταση, παραμετροποίηση και εκτέλεση τριών Honeypots. Μελετήθηκε η λειτουργικότητά τους και

καταγράφηκαν οι επιθέσεις που συγκέντρωσαν κατά την τρίμηνη λειτουργία τους. Τα αποτελέσματα που προέκυψαν παρουσιάζονται σε πίνακες και σε γραφήματα.

Στο ενδέκατο κεφάλαιο αναφέρονται τα συμπεράσματα που προέκυψαν από την εφαρμογή και δίδεται συνοπτικά μία περίληψη της συνολικής μελέτης που εκπονήθηκε στα πλαίσια της παρούσας μεταπτυχιακής διατριβής.

Τέλος, παρατίθενται ο κώδικας που χρησιμοποιήθηκε κατά την εφαρμογή, οδηγίες εγκατάστασης και παραμετροποίησης των συστημάτων που χρησιμοποιήθηκαν καθώς και το αποτέλεσμα ειδικών εργαλείων που χρησιμοποιήθηκαν για την ανάλυση και επεξεργασία των δεδομένων.

2. ΑΣΦΑΛΕΙΑ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

2.1 Ορισμός

Στόχος της Ασφάλειας Πληροφοριακών Συστημάτων είναι η προστασία των πολύτιμων πόρων ενός οργανισμού, όπως πληροφορίες, υλικό και λογισμικό. Μέσα από την επιλογή και εφαρμογή κατάλληλων μηχανισμών και διασφαλίσεων η Ασφάλεια Πληροφοριακών Συστημάτων συμβάλλει στην εκπλήρωση του στόχου του οργανισμού προστατεύοντας φυσικούς και οικονομικούς πόρους, τη φήμη, τη νόμιμη θέση, τους εργαζομένους και άλλα υλικά και άυλα περιουσιακά στοιχεία.

Συχνά, κακώς θεωρείται ότι η Ασφάλεια Πληροφοριακών Συστημάτων παρεμποδίζει τη λειτουργία της επιχείρησης, επιβάλλοντας αυστηρούς, ενοχλητικούς κανόνες και διαδικασίες σε χρήστες, διαχειριστές και συστήματα. Αυτό οφείλεται στη μη κατάλληλη για την επιχείρηση επιλογή κανόνων ασφαλείας. Οι κανόνες ασφαλείας τίθενται προς όφελος της επιχείρησης και κατά συνέπεια υποστηρίζουν την εύρυθμη λειτουργία της και συμβάλλουν στην κερδοφορία της.

2.2 Βασικές έννοιες

Οι θεμελιώδεις τρεις αρχές της Ασφάλειας Πληροφοριακών Συστημάτων είναι η εμπιστευτικότητα (Confidentiality), η ακεραιότητα (Integrity) και η διαθεσιμότητα (Availability). Όλοι οι έλεγχοι που υλοποιούνται στα πλαίσια της Ασφάλειας Πληροφοριών, καθώς και όλες οι απειλές, τα τρωτά σημεία και εν γένει οι διαδικασίες που σχετίζονται, υπόκεινται στα τρία παραπάνω κριτήρια. Παρακάτω αναλύονται οι έννοιες αυτές.

- Εμπιστευτικότητα: Η έννοια αυτή αφορά στην αποτροπή της από πρόθεση ή χωρίς μη εξουσιοδοτημένης αποκάλυψης του περιεχόμενου ενός μηνύματος. Απώλεια εμπιστευτικότητας μπορεί να συμβεί με πολλούς τρόπους, όπως για παράδειγμα κατά την εσκεμμένη διαρροή πληροφοριών ιδιωτικής εταιρείας ή κατά την εσφαλμένη ανάθεση δικαιωμάτων πρόσβασης σε χρήστες ενός δικτύου.

  Στα δίκτυα τηλεπικοινωνιών, η εμπιστευτικότητα εξασφαλίζεται με χρήση πρωτοκόλλων ασφαλείας, με υπηρεσίες ελέγχου και πιστοποίησης ταυτότητας καθώς και με υπηρεσίες κρυπτογράφησης δεδομένων.

- Ακεραιότητα: Η έννοια αυτή εξασφαλίζει τη μη τροποποίηση των δεδομένων από μη εξουσιοδοτημένους χρήστες και διαδικασίες αλλά και την επιτρεπόμενη τροποποίηση των δεδομένων από εξουσιοδοτημένους χρήστες και διαδικασίες. Εξασφαλίζει παράλληλα και το δέσιμο εσωτερικών και εξωτερικών δεδομένων, ότι δηλαδή τα εσωτερικά δεδομένα είναι συνεπή με το περιεχόμενο των υποενοτήτων του συστήματος και τα εξωτερικά δεδομένα είναι συνεπή με το περιεχόμενο του εξωτερικού περιβάλλοντος του συστήματος.

  Για παράδειγμα, έστω ότι μία εσωτερική βάση δεδομένων κρατάει τον αριθμό των μονάδων ενός συγκεκριμένου αντικειμένου ενός τμήματος του οργανισμού. Συνέπεια στα εσωτερικά δεδομένα σημαίνει ότι το άθροισμα του πλήθους των μονάδων σε κάθε τμήμα θα πρέπει να ισούται με το συνολικό αριθμό μονάδων που η βάση δεδομένων έχει καταγράψει εσωτερικά για όλο τον οργανισμό. Αντίστοιχα, στο ίδιο παράδειγμα, συνέπεια στα εξωτερικά δεδομένα σημαίνει ότι το πλήθος των μονάδων που έχει καταγραφεί στη βάση δεδομένων για κάθε τμήμα είναι ίδιο με το πλήθος των μονάδων που υπάρχουν πραγματικά σε κάθε τμήμα στον οργανισμό.

  Η έννοια της ακεραιότητας εμπεριέχει και την έννοια της μη αποποίησης ευθυνών, (nonrepudiation), το να μην μπορεί δηλαδή κάποιος να αρνηθεί μια ενέργειά του. Η μη αποποίηση ευθύνης εξασφαλίζει ότι ένα έγγραφο ή μήνυμα που έχει υπογραφεί ψηφιακά από κάποιον, έχει όντως υπογραφεί από αυτόν και ότι σε μελλοντικό χρόνο ο συγγραφέας αυτός δεν μπορεί να αρνηθεί την υπογραφή του. Για τη μη αποποίηση ευθύνης χρησιμοποιούνται ψηφιακές υπογραφές και πιστοποιητικά καθώς και αλγόριθμοι κρυπτογράφησης.

Στα δίκτυα τηλεπικοινωνιών, η ακεραιότητα εξασφαλίζεται με τη χρήση τειχών προστασίας, την οργάνωση και διαχείριση πολιτικών ασφάλειας επικοινωνιών καθώς και με υπηρεσίες ανίχνευσης παρείσφρησης .

- Διαθεσιμότητα: Η έννοια αυτή εξασφαλίζει την αξιόπιστη και έγκαιρη πρόσβαση σε δεδομένα και υπολογιστικούς πόρους από τους κατάλληλους χρήστες. Εξασφαλίζεται δηλαδή έτσι η απρόσκοπτη λειτουργία του συστήματος όταν απαιτείται. Επιπλέον, η έννοια αυτή εξασφαλίζει ότι οι απαιτούμενες υπηρεσίες ασφάλειας πληροφοριών βρίσκονται ομοίως σε κατάσταση λειτουργίας.

  Στα δίκτυα τηλεπικοινωνιών, η διαθεσιμότητα εξασφαλίζεται με περιορισμένη ανοχή σφαλμάτων, με χρήση αντιγράφων ασφαλείας, με ελέγχους πρόσβασης χρηστών στο σύστημα καθώς και με χρήση αξιόπιστων και διαλειτουργικών διαδικασιών ασφαλείας και μηχανισμών ασφάλειας δικτύων.

Το αντίθετο των τριών αυτών εννοιών είναι η γνωστοποίηση (Disclosure), η αλλοίωση (Alteration) και η καταστροφή (Disaster).

Άλλες εξίσου σημαντικές έννοιες που σχετίζονται με την ασφάλεια πληροφοριακών συστημάτων είναι οι παρακάτω.

- Αναγνώριση (Identification): Η έννοια αυτή αφορά στις διαδικασίες ελέγχου πρόσβασης και σχετίζεται με τις διαδικασίες πιστοποίησης και εξουσιοδότησης χρηστών.
- Πιστοποίηση (Authentication): Διασταυρώνει τα στοιχεία της ταυτότητας ενός χρήστη, πιστοποιεί την ταυτότητα του, εξασφαλίζει ότι ο χρήστης είναι αυτός που διατείνεται ότι είναι.
- Ευθύνη – Λογοδοσία (Accountability): Η ικανότητα ενός συστήματος να καθορίσει τις ενέργειες και τη συμπεριφορά ενός ατόμου στο σύστημα με χρήση μονοπατιών ελέγχου και αρχείων καταγραφής.
- Εξουσιοδότηση (Authorization): Η έννοια αυτή αφορά στα δικαιώματα που έχουν εκχωρηθεί σε έναν χρήστη και επιτρέπουν τη χρήση ενός υπολογιστικού πόρου. Μόλις η ταυτότητα ενός χρήστη πιστοποιηθεί, τα επίπεδα εξουσιοδότησης καθορίζουν την έκταση των δικαιωμάτων που μπορεί να έχει ο χρήστης.
- Ιδιωτικότητα (Privacy): Η έννοια αυτή καθορίζει το επίπεδο εμπιστευτικότητας που διαθέτει ένας χρήστης στο σύστημα, καθώς και το επίπεδο προστασίας απορρήτου που διατίθεται στο χρήστη από το σύστημα. [53]

2.3 Κοινές απειλές

Τα Υπολογιστικά Συστήματα θεωρούνται ευπαθή σε διαφόρων τύπων απειλές που με τη σειρά τους μπορούν να προκαλέσουν διαφόρων τύπων βλάβες και να οδηγήσουν σε σημαντικές απώλειες. Οι βλάβες μπορούν να ποικίλουν από λάθη που επηρεάζουν την ακεραιότητα της βάσης δεδομένων μέχρι πυρκαγιές που καταστρέφουν ολόκληρα υπολογιστικά κέντρα. Απώλειες μπορεί να προκύψουν, για παράδειγμα, από τις ενέργειες δήθεν έμπιστων εργαζομένων που εξαπατούν ένα σύστημα, από hackers, από απρόσεκτους υπαλλήλους που εισάγουν λανθασμένα δεδομένα στο σύστημα. Ακρίβεια στην εκτίμηση των απωλειών δεν είναι εφικτή καθώς πολλές απώλειες μπορεί να μη γίνουν ποτέ αντιληπτές ενώ άλλες αποκρύπτονται εσκεμμένα, ώστε να αποφευχθεί η ενδεχόμενη αρνητική δημοσιότητα. Οι απειλές έχουν ως στόχο να βλάψουν κάποια ή και τις τρεις θεμελιώδεις αρχές της Ασφάλειας Πληροφοριακών Συστημάτων, δηλαδή την εμπιστευτικότητα, την ακεραιότητα και τη διαθεσιμότητα.

Παρακάτω, παρατίθενται και αναλύονται οι πιο κοινές απειλές κατά της Ασφάλειας Πληροφοριακών Συστημάτων. Οι απειλές αυτές επιλέχθηκαν βάσει της επικράτησης και της σημασίας τους στα σύγχρονα υπολογιστικά συστήματα. Κάποιες απειλές ενδέχεται να συνδυάζουν στοιχεία από παραπάνω από μία περιοχές.

- Σφάλματα και παραλείψεις: Τα σφάλματα και οι παραλείψεις που πραγματοποιούνται από χρήστες, υπαλλήλους που εισάγουν δεδομένα, διαχειριστές συστημάτων και προγραμματιστές αφορούν άμεσα ή έμμεσα στην ασφάλεια δεδομένων. Ειδικά τα προγραμματιστικά λάθη, που αναφέρονται ως «bugs», κυμαίνονται σε σοβαρότητα από

αδιάφορα έως καταστροφικά για το σύστημα. Μπορούν να λάβουν χώρα καθ'όλη τη διάρκεια ζωής του συστήματος. Στην ίδια κατηγορία ανήκουν και τα σφάλματα εγκατάστασης και συντήρησης συστημάτων.

- Εξαπάτηση: Τα υπολογιστικά συστήματα μπορούν να αξιοποιηθούν κακόβουλα για εξαπάτηση χρηστών, συχνά με χρήση αυτοματοποιημένων μεθόδων. Υπεξαίρεση χρημάτων είναι ο πιο συχνός σκοπός, ωστόσο τα χρηματοπιστωτικά συστήματα δεν είναι τα μόνα που αποτελούν στόχο. Συστήματα ελέγχου πρόσβασης σε οποιοδήποτε πόρο αποτελούν στόχο, όπως συστήματα παρακολούθησης χρόνου, συστήματα απογραφής, σχολικά συστήματα και συστήματα διαχείρισης υπεραστικών τηλεφωνημάτων.

  Η εξαπάτηση χρηστών μπορεί να πραγματοποιηθεί και από γνώστες του συστήματος που εργάζονται εντός της επιχείρησης και από μη εξουσιοδοτημένους εξωτερικούς απομακρυσμένους χρήστες. Δεδομένου ότι οι γνώστες του συστήματος έχουν πρόσβαση και οικειότητα με το υπολογιστικό σύστημα, μπορούν εύκολα να το χρησιμοποιήσουν για πράξεις κακόβουλου τύπου. Οι γνώστες αυτοί μπορεί να είναι γενικοί χρήστες, όπως υπάλληλοι, ή τεχνικό προσωπικό. Πρώην εργαζόμενοι της επιχείρησης μπορεί επίσης να αποτελούν απειλή, ιδιαίτερα εάν η πρόσβασή τους στο σύστημα δεν έχει τερματιστεί πλήρως.

  Επιπρόσθετα, υλικό και λογισμικό συχνά μπορεί να αφήνουν τρύπες ασφαλείας που γίνονται στη συνέχεια αντικείμενο εκμετάλλευσης για ενέργειες κακόβουλου τύπου.

- Δολιοφθορά – Σαμποτάζ: Τέτοιες ενέργειες μπορεί να είναι η καταστροφή υλικού και τεχνικών εγκαταστάσεων, η διαγραφή ή τροποποίηση δεδομένων, η εισαγωγή με λάθος τρόπο ή λανθασμένων δεδομένων στο σύστημα, η εισαγωγή ιογενούς λογισμικού στον κώδικα. Πραγματοποιούνται από εργαζομένους που νιώθουν ότι ο κόπος για την εργασία τους δεν ανταμείβεται ή βαριούνται ή παρενοχλούνται στο περιβάλλον εργασίας του και επιθυμούν να προκαλέσουν ζημιά στην επιχείρηση και ενδεχομένως να πάρουν εκδίκηση.
- Απώλεια φυσικών υποδομών: Στην κατηγορία αυτή ανήκουν εκτός από τις φυσικές καταστροφές και διακοπές ρεύματος και επικοινωνιών, διαρροές νερού, προβλήματα αποχέτευσης, προβλήματα στις μεταφορές και μετακινήσεις, απεργίες.
- Λογισμικό κακόβουλου τύπου: Η κατηγορία αυτή αναλύεται διεξοδικά σε επόμενο κεφάλαιο.
- Βιομηχανική κατασκοπεία: Η συλλογή απόρρητων στοιχείων λειτουργίας μιας επιχείρησης από άλλες ή από την κυβέρνηση με σκοπό οι μεν εταιρείες να βελτιωθούν έναντι άλλων επιχειρήσεων και η δε κυβέρνηση να βελτιώσει τους δείκτες ανταγωνιστικότητας εγχώριων επιχειρήσεων. Η Ασφάλεια Πληροφοριακών Συστημάτων μπορεί να καταστήσει την εκμαίευση τέτοιων πληροφοριών αδύνατη, ωστόσο η απειλή εξουσιοδοτημένοι υπάλληλοι να πωλούν τέτοιες ιδιωτικές πληροφορίες παραμένει. [54]

2.4 Δικτυακές επιθέσεις

Παρακάτω παρατίθενται οι πιο κοινές δικτυακές επιθέσεις. Κάποιες από αυτές είναι παθητικές, σκοπό έχουν δηλαδή την καταγραφή δεδομένων ενώ άλλες είναι ενεργητικές, σκοπό έχουν δηλαδή την τροποποίηση ή και καταστροφή δικτυακών δεδομένων.

- Eavesdropping (υποκλοπή): Αυτός ο τύπος επίθεσης σκοπό έχει τη μη εξουσιοδοτημένη παρακολούθηση δικτυακής κίνησης. Ορισμένες μέθοδοι μετάδοσης δικτυακής κίνησης όπως μέσω δορυφόρων, κινητών, PDAs, ασύρματα κ.α. θεωρούνται ευπαθείς όσον αφορά στις υποκλοπές. Χωρίς χρήση κρυπτογράφησης, τα δεδομένα είναι ευάλωτα στην υποκλοπή καθώς διασχίζουν ένα δίκτυο.

  Οι eavesdropping επιθέσεις διακρίνονται σε ενεργητικές και παθητικές. Ως παθητικές χαρακτηρίζονται αυτές κατά τις οποίες παρακολουθούνται και καταγράφονται μεταδόσεις χωρίς άδεια από τον αποστολέα ή τον παραλήπτη. Ως ενεργητικές χαρακτηρίζονται οι προσπάθειες δημιουργίας καναλιού σηματοδότησης εντός του καναλιού μετάδοσης επικοινωνίας ώστε να υποκλέπτονται οι μεταδόσεις.

Μία παραλλαγή μίας ενεργητικής eavesdropping επίθεσης ονομάζεται Covert Channel και χαρακτηρίζει τη χρήση μίας κρυφής μη εξουσιοδοτημένης δικτυακής σύνδεσης για τη μετάδοση απόρρητων πληροφοριών. Στην ουσία μεταδίδονται δεδομένα επικοινωνίας μεταξύ διαδικασιών, οι οποίες βάσει πολιτικής ασφαλείας δεν επιτρέπεται να επικοινωνούν μεταξύ τους. Αντίστοιχα, στο Covert Timing Channel – μία άλλη παραλλαγή – η κάθε διαδικασία μεταδίδει σήμα δεδομένων διαμορφώνοντας το χρόνο επεξεργασίας, ώστε να διαμορφωθεί στη συνέχεια αντίστοιχα ο χρόνος απόκρισης της δεύτερης διαδικασίας. [53]

- Επίθεση Άρνησης Εξυπηρέτησης – Denial Of Service, (DOS): Αυτός ο τύπος επιθέσεων χαρακτηρίζει τις επιθέσεις που σκοπό έχουν να καταστήσουν ένα σύστημα ανίκανο να δεχτεί άλλες συνδέσεις και έτσι να μην μπορεί να εξυπηρετήσει άλλους πιθανούς πελάτες. Στόχο αποτελούν οι δικτυακές συσκευές, διακομιστές, υποδομές διαμοιρασμού εύρους ζώνης, εν γένει οτιδήποτε περικλείει μεγάλο όγκο επισκεψιμότητας ώστε να μπορεί να υποβαθμιστεί σταδιακά.

  Μία επίθεση άρνησης υπηρεσίας χρησιμοποιεί τις παρακάτω μεθόδους προκειμένου το σύστημα να διαθέσει όλους του τους πόρους και να μην μπορεί στη συνέχεια να δεχτεί άλλες συνδέσεις:

  - Με χρήση τεράστιων συνημμένων ηλεκτρονικών μηνυμάτων και αρχείων, να γεμίσει ο σκληρός δίσκος.
  - Με αποστολές μηνυμάτων να προκληθεί αλλαγή της μάσκας υποδικτύου του υπολογιστή προορισμού και διαταραχή στη δρομολόγηση των δεδομένων στο δίκτυο.
  - Χρήση του συνόλου των πόρων του υπολογιστή προορισμού που χρησιμοποιούνται για την αποδοχή συνδέσεων δικτύου, ώστε πρόσθετες συνδέσεις να μην επιτρέπονται.

  Τύποι άρνησης επιθέσεων παρατίθενται και αναλύονται παρακάτω.

  - Υπερχείλιση του buffer: Η επίθεση αυτή συμβαίνει όταν μία διαδικασία λαμβάνει πολλά περισσότερα δεδομένα από όσα μπορεί να διαχειριστεί. Εάν η διαδικασία δε διαθέτει κάποια προγραμματισμένη ρουτίνα που να ενεργοποιείται σε περιπτώσεις υπερχείλισης του buffer, τότε λειτουργεί με μη αναμενόμενο τρόπο, τον οποίο ένας εισβολέας μπορεί να εκμεταλλευθεί για τους σκοπούς του.

    Μία κοινή τέτοια επίθεση ονομάζεται “Ping of death”, κατά την οποία ο εισβολέας στέλνει μία ping αίτηση που αποτελείται από ένα ειδικά διαμορφωμένο πολύ μεγάλο IP δεδομενόγραμμα, υπερχειλίζοντας έτσι τους buffers του συστήματος και προκαλώντας την κατάρρευση του συστήματος. Μία ping αίτηση κανονικά αποτελείται από 56 bytes (ή 84 bytes συμπεριλαμβανομένης και της IP κεφαλίδας. Όταν το μέγεθος αυτό ξεπεράσει τα 65535 bytes, τότε πραγματοποιείται η “Ping of death” επίθεση.
  - SYN επίθεση: Η επίθεση αυτή πραγματοποιείται όταν ένας εισβολέας εκμεταλλεύεται το χώρο μνήμης του buffer κατά την έναρξη μίας TCP, (Transmission Control Protocol), χειραψίας. Ο εισβολέας πλημμυρίζει το στόχο με αιτήσεις σύνδεσης, γεμίζοντας την ουρά, αλλά δεν απαντάει σε καμία από τις αιτήσεις απόκρισης που στέλνει πίσω ο στόχος. Έτσι εξαντλείται ο χρόνος αναμονής κατά τον οποίο ο στόχος αναμένει συνδέσεις, οπότε το σύστημα καταρρέει ή καθίσταται αδρανές.
  - Teardrop επίθεση: Η επίθεση αυτή πραγματοποιείται τροποποιώντας το μέγεθος και τα πεδία διευθυνσιοδότησης κατακερματισμού σε διαδοχικά πακέτα IP. Το σύστημα που αποτελεί το στόχο παθαίνει σύγχυση και τελικά καταρρέει αφού λαμβάνει αντιφατικές οδηγίες για το πώς τα κατακερματισμένα πακέτα πρέπει να ανασυντεθούν στα αρχικά πακέτα.
  - Smurf επίθεση: Η επίθεση αυτή χρησιμοποιεί έναν συνδυασμό ICMP, (Internet Control Message Protocol), πακέτων και IP παραπλάνησης (IP spoofing), ώστε η κίνηση σε ένα δίκτυο να αυξηθεί κατακόρυφα. Πιο συγκεκριμένα, αποστέλλονται ICMP πακέτα με διαφορετική από την πραγματική IP διεύθυνση πηγής σε όλο το δίκτυο με χρήση της διεύθυνσης εκπομπής του δικτύου. Ως απόκριση στα

μηνύματα αυτά, θα αποσταλούν μηνύματα προς τη μη πραγματική IP διεύθυνση πηγής προκαλώντας υπερχείλιση.

Το όνομα Smurf προκύπτει από το αρχείο smurf.c, τον πηγαίο κώδικα του προγράμματος, που γράφτηκε το 1997 από τον TFreak. [53]

- Δικτυακές εισβολές, (Network Intrusion): Αυτός ο τύπος επίθεσης αναφέρεται στη μη εξουσιοδοτημένη πρόσβαση σε ένα δίκτυο κυρίως όμως από εξωτερικές πηγές. Οι εισβολείς δεν είναι γνωστοί στην εταιρεία ή τον εν λόγω οργανισμό που δέχεται την επίθεση και εκμεταλλεύονται ευπάθειες της πολιτικής ασφαλείας.

  Παρακάτω παρατίθενται και αναλύονται οι πιο κοινοί τύποι δικτυακών εισβολών.

  - Παραπλάνηση, (Spoofing): Πραγματοποιείται όταν ένας εισβολέας σκόπιμα προκαλεί έναν χρήστη ή ένα αντικείμενο ώστε να εκτελέσει μία εσφαλμένη ενέργεια δίνοντας του ανακριβείς πληροφορίες.

    Ειδικότερα IP Spoofing είναι μία επίθεση κατά την οποία ένα σύστημα θεωρεί ότι επικοινωνεί με μία γνωστή οντότητα που με τη σειρά της δίνει πρόσβαση σε έναν εισβολέα. Πιο συγκεκριμένα, τα πακέτα σε επίπεδο TCP τροποποιούνται ώστε συστήματα διασυνδεδεμένα στο Διαδίκτυο που χρησιμοποιούν TCP/IP υπηρεσίες, να μπορούν να γίνουν αντικείμενο επίθεσης. Ο εισβολέας στέλνει πακέτα που περιέχουν διαφορετική IP διεύθυνση πηγής από την πραγματική, προκαλώντας σύγχυση και τελικά κατάρρευση του συστήματος.
  - Piggy-backing: Πραγματοποιείται όταν ένας εισβολέας αποκτά μη εξουσιοδοτημένη πρόσβαση σε ένα μηχάνημα χρησιμοποιώντας τα στοιχεία σύνδεσης ενός εξουσιοδοτημένου χρήστη. Αυτό μπορεί να γίνει για παράδειγμα όταν ένας χρήστης αφήσει ανοιχτή μία συνεδρία ή αποσυνδεθεί εσφαλμένα.
  - Back-door επιθέσεις: Αναφέρεται σε εισβολές μέσω συνδέσεων τηλεφώνου ή σε ασύγχρονες εκτός δικτύου συνδέσεις. [53]
- IP επιθέσεις κατακερματισμού, (IP fragmentation attacks): Αυτού του τύπου οι επιθέσεις χρησιμοποιούν κατακερματισμό IP πακέτων ώστε να «μεταμφιέσουν», να διαφοροποιήσουν δηλαδή τα TCP πακέτα τους από αυτά που αναμένει ο στόχος.

  Παρακάτω παρατίθενται και αναλύονται παραδείγματα τέτοιων επιθέσεων:

  - Επίθεση μικρού κατακερματισμού, (tiny fragment attack): Η επίθεση αυτή πραγματοποιείται όταν ο εισβολέας στέλνει ένα πολύ μικρό τεμάχιο που αναγκάζει κάποια από τα TCP πεδία κεφαλίδας να κατακερματισθούν σε δεύτερο τεμάχιο. Εάν ο στόχος δε χρησιμοποιεί το ίδιο ελάχιστο μέγεθος στην ανασύνθεση των πακέτων, τότε το δεύτερο τεμάχιο θα μπορέσει να διασχίσει όλο το δίκτυο.
  - Επίθεση επικάλυψης κατακερματισμού, (overlapping fragment attack): Η επίθεση αυτή αποτελεί παραλλαγή της teardrop επίθεσης. Διαδοχικά πακέτα τροποποιούν την διεύθυνση προορισμού του αρχικού πακέτου, οπότε το δεύτερο πακέτο φτάνει στον προορισμό όπου αν δεν υποστηρίζεται διευθυνσιοδότηση κατακερματισμού στο ελάχιστο μέγεθος για μη-μηδενικές διευθύνσεις, το δεύτερο πακέτο θα μπορέσει να διασχίσει όλο το δίκτυο. [53]

Παρακάτω παρατίθενται και αναλύονται κάποιες συνδέσεις που επιτρέπουν τη μη εξουσιοδοτημένη πρόσβαση και θεωρούνται μη θεμιτές, ωστόσο δεν αποτελούν επιθέσεις. Φέρουν όμως το ίδιο αποτέλεσμα με μία επίθεση, δηλαδή τη μη εξουσιοδοτημένη πρόσβαση σε συστήματα και υπηρεσίες και για αυτό το λόγο παρατίθενται και στο παρόν κεφάλαιο.

- Μη εξουσιοδοτημένη πρόσβαση σε περιορισμένες δικτυακές υπηρεσίες: Αυτός ο τύπος σύνδεσης συχνά χαρακτηρίζεται και ως κατάχρηση σύνδεσης. Αναφέρεται σε εξουσιοδοτημένους χρήστες που κατορθώνουν και αποκτούν πρόσβαση σε υπηρεσίες όπου κανονικά θα είχαν περιορισμένη ή και μηδαμινή πρόσβαση. Εστιάζει σε εσωτερικούς χρήστες του συστήματος ή χρήστες που βρίσκονται σε χαμηλό επίπεδο ασφάλειας και παριστάνουν ότι είναι άλλοι χρήστες. Για αυτούς χρησιμοποιείται ο όρος “masquerading”, (μεταμφίεση). Ένα τέτοιο παράδειγμα είναι ένας εισβολέας που έχει εκμαιεύσει από τρίτους και χρησιμοποιεί κωδικούς Διαδικτύου για να συνδεθεί στο Διαδίκτυο.

- Μη εξουσιοδοτημένη χρήση δικτύου για μη επαγγελματικούς σκοπούς: Αυτός ο τύπος κατάχρησης δικτυακής σύνδεσης αναφέρεται σε μη επαγγελματική ή προσωπική χρήση από εξουσιοδοτημένους χρήστες. Για παράδειγμα, περιήγηση σε ιστοσελίδες πορνογραφικού περιεχομένου. Η χρήση δικτυακών υπηρεσιών για μη επαγγελματικούς σκοπούς σε εταιρείες και επιχειρήσεις θεωρείται μη επιτρεπτή. [53]

2.5 Λογισμικό κακόβουλου τύπου (malware)

2.5.1 Ορισμός

Ο όρος malware αποτελεί σύντμηση των λέξεων malicious software και αναφέρεται σε κάθε τύπο λογισμικού που έχει σχεδιαστεί με σκοπό να προκαλέσει βλάβη σε έναν μεμονωμένο υπολογιστή, σε ένα διακομιστή, σε ένα δίκτυο και εν γένει σε ένα οποιοδήποτε υπολογιστικό σύστημα. Μπορεί να έχει τη μορφή κώδικα, προγραμμάτων σεναρίου (scripts), ενεργού περιεχομένου και εν γένει οποιουδήποτε λογισμικού προγράμματος. Η βλάβη μπορεί να είναι η διατάραξη της υπολογιστικής λειτουργίας, η συλλογή ευαίσθητων πληροφοριών ή και η απόκτηση πλήρους πρόσβασης σε ιδιωτικά, απόρρητα, προσωπικά συστήματα πληροφοριών.

Το λογισμικό κακόβουλου τύπου διακρίνεται σε δυο κατηγορίες, σε αυτό που χρειάζεται ένα πρόγραμμα – φορέα για να εκτελεστεί, δηλαδή δεν μπορεί να εκτελεστεί μόνο του και στο αυθύπαρκτο που μπορεί να εκτελεστεί μόνο του και ανεξάρτητα κάτω από τον έλεγχο του λειτουργικού συστήματος, όπως και κάθε άλλο πρόγραμμα.

Επακόλουθα, μία άλλη κατηγοριοποίηση του λογισμικού κακόβουλου τύπου μπορεί να γίνει σε αναπαραγόμενα και μη αναπαραγόμενα προγράμματα. Τα αναπαραγόμενα είναι προγράμματα που εκτελούνται όταν καλείται το πρόγραμμα – φορέας προς εκτέλεση. Τα μη αναπαραγόμενα είναι τμήματα προγράμματος αλλά και αυθύπαρκτα προγράμματα, τα οποία θα εκτελεστούν αργότερα στο ίδιο ή σε κάποιο άλλο υποπρόγραμμα του συστήματος. [43]

Παρακάτω αναλύονται τα κυριότερα είδη κακόβουλου λογισμικού.

2.5.2 Ιός (virus)

Ως ιός ορίζεται ένα πρόγραμμα που μπορεί να εξαπλώνεται σε άλλα προγράμματα, να «μολύνει» δηλαδή άλλα προγράμματα τροποποιώντας τα και με σκοπό να βλάψει χρήσιμα αρχεία. Η τροποποίηση πραγματοποιείται και με την εισαγωγή στο πρόγραμμα που μολύνεται ενός αντιγράφου του ιού, που με τη σειρά του κλωνοποιεί τον εαυτό του κ.ο.κ. για να μολύνει και άλλα προγράμματα.

Ο Fred Cohen το Νοέμβριο του 1983 ήταν ο πρώτος που εισήγαγε την έννοια του ιού όπως είναι γνωστή σήμερα. Συγκεκριμένα, παρουσίασε σε ένα σεμινάριο το πρόγραμμα που είχε γράψει για λειτουργικό UNIX, το οποίο όταν εκτελούνταν, δημιουργούσε κλώνους του εαυτού του και εξαπλωνόταν από έναν υπολογιστή σε άλλον σε ένα δίκτυο.

Ένα χρόνο αργότερα, ο Fred Cohen δημοσίευσε το πρώτο του άρθρο περί συμπεριφοράς των ιών εν γένει, το οποίο συμπεριλήφθηκε στη διδακτορική του διατριβή «Computer Viruses – Theory and Experiments». Αυτό που ουσιαστικά απέδειξε ο Cohen ήταν ότι η μόλυνση είναι δυνατόν να υπάρξει όποτε υπάρχει διαμοιράσιμη πληροφορία ή μη ελεγχόμενη ροή πληροφορίας. Επακόλουθα, ο μόνος σίγουρος τρόπος για να εμποδιστεί η μόλυνση που οφείλεται σε ιό, είναι η απαγόρευση της ύπαρξης διαμοιράσιμων πόρων και της ροής πληροφορίας στο σύστημα. Τότε όμως το σύστημα ενδέχεται να καταλήξει να μη λειτουργεί. [45]

Όπως ακριβώς και οι βιολογικοί ιοί εξαπατούν τα κύτταρα ενός οργανισμού ενσωματώνοντας το γενετικό τους υλικό στο γονιδίωμα αυτών και στη συνέχεια χρησιμοποιούν τους μηχανισμούς αντιγραφής του κυττάρου για να αναπαραχθούν, έτσι και οι ιοί του υπολογιστή εμπεριέχουν στον κώδικά τους τη δυνατότητα δημιουργίας τέλειων αντιγράφων του εαυτού τους.

Αφού πρώτα εγκατασταθεί σε ένα υπολογιστικό σύστημα, ο ιός αποκτά, προσωρινά, τον έλεγχο του λειτουργικού συστήματος. Εν συνεχεία, οποτεδήποτε ο μολυσμένος

υπολογιστής έρθει σε επαφή με μη μολυσμένο πρόγραμμα, το πρόγραμμα αυτό μολύνεται με την εισαγωγή στον κώδικα του ενός αντιγράφου του ιού. Έτσι, η μόλυνση αυτή μπορεί να διαδοθεί από υπολογιστή σε υπολογιστή μέσω ανυποψίαστων χρηστών που ανταλλάσσουν λογισμικό περιεχόμενο δικτυακά. Τα περιβάλλοντα δικτύου, μάλιστα, με τη δυνατότητα που παρέχουν στους χρήστες για προσπέλαση εφαρμογών και υπηρεσιών συστήματος που βρίσκονται σε απομακρυσμένους υπολογιστές, αποτελούν ιδανικό περιβάλλον για τη διάδοση των ιών. Ένας ιός μπορεί να κάνει οτιδήποτε και σε οποιοδήποτε άλλο πρόγραμμα αφού πρώτα προσαρτηθεί στο πρόγραμμα αυτό και εκτελεστεί το δεύτερο.

Κατά τη διάρκεια της ζωής του, ένας ιός περνά από τις εξής φάσεις.

1. Φάση ύπνωσης: Κατά τη φάση αυτή ο ιός είναι ανενεργός και αναμένει την ενεργοποίησή του από κάποιο γεγονός, όπως την έλευση μιας ημερομηνίας, την παρουσία ενός άλλου προγράμματος ή αρχείου ή την υπέρβαση κάποιου αποθηκευτικού ορίου στο δίσκο. Η φάση αυτή δεν υπάρχει σε όλους τους ιούς.
2. Φάση διάδοσης: Κατά τη φάση αυτή ο ιός δημιουργεί ένα ακριβές αντίγραφο του εαυτού του, το οποίο τοποθετείται προς εκτέλεση σε άλλα προγράμματα ή σε συγκεκριμένες περιοχές του δίσκου. Το αντίγραφο θα εκτελεστεί και θα δημιουργήσει εκ νέου ένα αντίγραφο, κ.ο.κ. Κάθε μολυσμένο πρόγραμμα θα περιέχει τώρα έναν κλώνο του ιού, ο οποίος με τη σειρά του θα μπει σε φάση διάδοσης.
3. Φάση ενεργοποίησης: Ο ιός ενεργοποιείται για να επιτελέσει τη λειτουργία για την οποία έχει σχεδιαστεί. Όπως και με τη φάση διάδοσης, η φάση ενεργοποίησης μπορεί να πυροδοτηθεί από την εμφάνιση κάποιου γεγονότος σχετικού με το σύστημα. Το πλέον σύνηθες γεγονός που προκαλεί την πυροδότηση αυτή είναι η δημιουργία συγκεκριμένου αριθμού αντιγράφων του ιού ή η έλευση μιας συγκεκριμένης ημερομηνίας.
4. Φάση εκτέλεσης: Κατά τη φάση αυτή, ο κώδικας του ιού εκτελείται. Η λειτουργία μπορεί να είναι αβλαβής, όπως η απλή εμφάνιση ενός μηνύματος στην οθόνη, ή επιβλαβής, όπως η καταστροφή ή διαγραφή προγραμμάτων και αρχείων δεδομένων ή και η αποστολή μηνυμάτων ηλεκτρονικού ταχυδρομείου.

Οι περισσότεροι ιοί εκτελούν τη δουλειά τους με τρόπο ειδικά σχεδιασμένο για το συγκεκριμένο λειτουργικό σύστημα, και σε μερικές περιπτώσεις, ειδικά σχεδιασμένο για συγκεκριμένη πλατφόρμα υλικού. Έτσι, είναι σχεδιασμένοι να εκμεταλλεύονται τις λεπτομέρειες και αδυναμίες συγκεκριμένων μηχανισμών. [43]

2.5.3 Δούρειος ίππος (trojan horse)

Ο όρος «Δούρειος ίππος» είναι δανεισμένος από το γνωστό Δούρειο ίππο του τρωικού πολέμου, που προσφέρθηκε ως δώρο από τον Οδυσσέα προς τους Τρώες, με σκοπό να τους εξαπατήσει και να καταφέρουν οι κρυμμένοι στο εσωτερικό του Δούρειου ίππου στρατιώτες να εισέλθουν στην πόλη της Τροίας. Με παρόμοια λειτουργία, οι Δούρειοι ίπποι στον τομέα των υπολογιστών εξαπατούν τους χρήστες, καθώς φαίνονται να είναι κανονικά και χρήσιμα προγράμματα, τα οποία όμως περιέχουν βλαβερό κώδικα, που ενεργοποιείται όταν εκτελεστεί και το πρόγραμμα. Δεν μπορούν να δράσουν αυτόνομα επομένως αλλά η εκτέλεσή τους εξαρτάται από τις ενέργειες του υποψήφιου θύματος.

Ο όρος «Δούρειος ίππος» χρησιμοποιήθηκε για πρώτη φορά από τον Ken Thompson το 1983 κατά τη διάλεξή του στην τελετή απονομής βραβείων Turing. Ο Thompson ονόμασε «Δούρειο ίππο» τη δυνατότητα προσθήκης κώδικα κακόβουλου τύπου στην εντολή login του Unix με στόχο την υποκλοπή κωδικών πρόσβασης. Επιπρόσθετα, διαπίστωσε ότι οποιοσδήποτε μεταγλωττιστής C μπορεί να μετατραπεί κατάλληλα ώστε να προσθέτει αυτόματα κώδικα κακόβουλου τύπου στα προγράμματα που δημιουργεί, γεγονός που καθιστά δύσκολο τον εντοπισμό του κακόβουλου κώδικα. [46]

Οι συνηθέστερες λειτουργίες των Δούρειων ίππων είναι είτε η καταστροφή ή τροποποίηση δεδομένων κατά την εκκίνησή τους είτε η προσπάθεια υποκλοπής προσωπικών δεδομένων, όπως κωδικών πρόσβασης και αριθμών πιστωτικών καρτών. Χρησιμοποιούνται εν γένει για να πραγματοποιήσουν έμμεσα λειτουργίες που ο μη εξουσιοδοτημένος χρήστης δεν μπορεί άμεσα να εκτελέσει.

Σε αντίθεση με τους ιούς, οι δούρειοι ίπποι δε μεταδίδονται μολύνοντας αρχεία, δεν αναπαράγονται δηλαδή. [43]

2.5.4 Σκουλήκι (worm)

Ως σκουλήκι ορίζεται ένα πρόγραμμα που διαδίδεται αντιγράφοντας τον εαυτό του από υπολογιστή σε υπολογιστή μέσω τοπικών δικτύων ή μέσω Διαδικτύου. Η λειτουργία του είναι πολύπλευρη, μπορεί δηλαδή κατά την ενεργοποίησή του να συμπεριφερθεί ως ιός ή να εισαγάγει Δούρειους ίππους ή και να εκτελέσει οποιαδήποτε καταστροφική ενέργεια.

Πιο συγκεκριμένα, ένα σκουλήκι αποστέλλει αντίγραφα του εαυτού του σε άλλα συστήματα είτε μέσω ηλεκτρονικού ταχυδρομείου είτε μέσω απομακρυσμένης σύνδεσης. Τα νέα αντίγραφα δημιουργούν εκ νέου αντίγραφα κ.ο.κ.

Τα σκουλήκια έχουν τις ίδιες φάσεις με τους ιούς: φάση ύπνωσης, φάση ενεργοποίησης, φάση διάδοσης και φάση εκτέλεσης.

Κατά τη φάση ύπνωσης, το σκουλήκι είναι ανενεργό και περιμένει να ενεργοποιηθεί από την έλευση κάποιου γεγονότος, όπως την εκτέλεση μιας συγκεκριμένης εντολής από το χρήστη, την έλευση μιας ημερομηνίας, την παρουσία ενός άλλου προγράμματος ή αρχείου ή την υπέρβαση κάποιου αποθηκευτικού ορίου στο δίσκο.

Κατά τη φάση ενεργοποίησης, το σκουλήκι ενεργοποιείται και εκκινεί τη λειτουργία για την οποία έχει σχεδιαστεί. Παράλληλα, ενεργοποιείται και η φάση διάδοσης και το σκουλήκι αρχίζει την αναζήτηση άλλων συστημάτων προς μόλυνση, με εξέταση των πινάκων που περιέχουν διευθύνσεις απομακρυσμένων συστημάτων, εν συνεχεία με εγκαθίδρυση απομακρυσμένης σύνδεσης σε μηχάνημα και τέλος με την αντιγραφή του εαυτού του στο απομακρυσμένο σύστημα και την εκτέλεσή του. [43]

Ο όρος «σκουλήκι» χρησιμοποιήθηκε για πρώτη φορά το 1975 στο μυθιστόρημα του John Brunner “The Shockwave Rider”, με την έννοια που γνωρίζουμε σήμερα. Δηλαδή ως «σκουλήκι» χαρακτηρίστηκε ο τύπος λογισμικού που σχεδίασε ένας εκ των πρωταγωνιστών του μυθιστορήματος για να συλλέξει στοιχεία που θα λειτουργούσαν ως αντίμετρο στη μαζική συμμόρφωση που προκαλούσε ένα εικονικό ψηφιακό εργαλείο που δρούσε υπέρ της κυβέρνησης. [47]

Αργότερα, το 1988, ο Robert Tappan Morris, ένας μεταπτυχιακός φοιτητής Πληροφορικής στο πανεπιστήμιο του Cornwell, δημιούργησε και εξαπέλυσε το πρώτο σκουλήκι, που έμεινε γνωστό με την ονομασία Morris, το οποίο μόλυνε ένα μεγάλο αριθμό υπολογιστών συνδεδεμένων στο Διαδίκτυο, που αργότερα υπολογίστηκε ότι αποτελούσαν το ένα δέκατο του συνόλου των υπολογιστών που διασυνδέονταν στο Διαδίκτυο. Το Morris σκουλήκι φαινόταν ότι είχε απελευθερωθεί από το MIT, ωστόσο είχε απελευθερωθεί στην πραγματικότητα από το Cornwell πανεπιστήμιο. Ο δημιουργός του δεν ήθελε να μολύνει άλλους υπολογιστές αλλά να εκτιμήσει το μέγεθος του Διαδικτύου. Αυτό που μετέτρεψε το σκουλήκι από μία ακίνδυνη εργασία σε μία επίθεση Άρνησης Υπηρεσίας, (Denial Of Service attacks, - DoS attacks), ήταν ο μηχανισμός εξάπλωσης. Θα μπορούσε ο χρήστης να καθοδηγεί το σκουλήκι να αναπαράγεται μόνο στην περίπτωση που δεν εκτελείται ήδη ένα άλλο αντίγραφο στον υπολογιστή. Ωστόσο, επειδή αυτό θα καθιστούσε ιδιαίτερα εύκολη την εξόντωση του σκουληκιού, ο Morris το σχεδίασε έτσι ώστε βάσει πιθανοτήτων το σκουλήκι να αντιγράφει τον εαυτό του 1 στις 7 φορές. Αναπόφευκτα, ο Morris ήταν και ο πρώτος που καταδικάστηκε για καταπάτηση του άρθρου 18 του αμερικανικού συντάγματος περί ηλεκτρονικής απάτης. [48]

2.5.5 Κερκόπορτα (trapdoor)

Ως κερκόπορτα ορίζεται μία μυστική είσοδος σε ένα πρόγραμμα, εφαρμογή, λειτουργικό σύστημα ή διαδικτυακή υπηρεσία που επιτρέπει σε όποιον τη γνωρίζει, να αποκτήσει πρόσβαση στο σύστημα παρακάμπτοντας τις συνήθεις διαδικασίες ελέγχου προσπέλασης. Οι κερκόπορτες δημιουργούνται συνήθως από τους ίδιους προγραμματιστές – δημιουργούς κατά τη διάρκεια εκσφαλμάτωσης ή δοκιμής του συστήματος και εν συνεχεία κλείνονται. Αν αφεθούν ανοιχτές, αποτελούν τρύπες ασφαλείας. Επιπρόσθετα, είναι σχεδόν αδύνατο να απομακρυνθούν κι έτσι απαιτείται σχεδόν πάντα επαναδιαμόρφωση, (format), του σκληρού δίσκου. [43]

Πιο συγκεκριμένα, συνήθως κατά τη διαδικασία αυθεντικοποίησης, ο προγραμματιστής προκειμένου να έχει ειδικά προνόμια και για να αποφύγει την απαιτούμενη διαδικασία εγκατάστασης και αυθεντικοποίησης, εισάγει στον κώδικα του κερκόπορτες. Μια κερκόπορτα μπορεί να είναι μια ειδική ακολουθία εισόδου ή κώδικας που ενεργοποιείται από συγκεκριμένο χρήστη ή ακόμα και μια συγκεκριμένη ακολουθία γεγονότων.

Αποτελεί πάγια τακτική εισβολέων αφού έχουν αποκτήσει πρόσβαση σε ένα σύστημα, να δημιουργούν και να αφήνουν ανοιχτές κερκόπορτες ώστε να μπορούν σε μελλοντικό χρόνο εύκολα και γρήγορα να επανασυνδεθούν. [49]

Η καλύτερη μέθοδος για προστασία από κερκόπορτες είναι η πρόληψη κατά τις διαδικασίες ανάπτυξης και συντήρησης λογισμικού.

Οι κερκόπορτες αναφέρονται συχνά και ως πίσω πόρτες, (backdoors).

2.5.6 Λογικές βόμβες (logic bombs)

Οι λογικές βόμβες είναι κομμάτια κώδικα ενσωματωμένα σε νόμιμα προγράμματα εφαρμογών, η εκτέλεση των οποίων ρυθμίζεται να πυροδοτείται από συγκεκριμένες συνθήκες – γεγονότα, όπως η έλευση μιας ημερομηνίας, η εκτέλεση της εφαρμογής από έναν συγκεκριμένο χρήστη, η απομάκρυνση από το σύστημα ενός κρίσιμου – απαραίτητου αρχείου, η μνήμη του σκληρού δίσκου του συστήματος να ξεπεράσει ένα συγκεκριμένο ποσοστό κ.α. [43]

Από τη στιγμή που ενεργοποιηθεί μία λογική βόμβα, μπορεί να διαγράψει δεδομένα ή και αρχεία, να σταματήσει την εκτέλεση του προγράμματος ή να κάνει οποιαδήποτε άλλη ζημιά.

Συχνά, οι λογικές βόμβες αναπτύσσονται από τους προγραμματιστές και για την πρόληψη και αποφυγή επιθέσεων στο σύστημα. Μπορεί να λειτουργήσουν ως honeytokens, δηλαδή να αποτελούν μια ψηφιακή οντότητα που θα είναι ευπαθής σε επιθέσεις ώστε να λειτουργήσει σα συναγερμός στην περίπτωση μη εξουσιοδοτημένης χρήσης. [49]

2.5.7 Bots – Botnets

Ως bots χαρακτηρίζονται τα αυτοματοποιημένα προγράμματα που εκτελούν συγκεκριμένες διεργασίες μέσω του Διαδικτύου. Οι διεργασίες που εκτελούν είναι απλές στη δομή και επαναλαμβανόμενες, σε ρυθμό όμως αρκετά μεγαλύτερο από αυτόν που θα είχε ένας χρήστης που θα εκτελούσε τις ίδιες εργασίες. Τα bots λειτουργούν κατά κάποιο τρόπο ως ανιχνευτές και λαμβάνουν, αναλύουν και καταγράφουν πληροφορίες από διαδικτυακούς διακομιστές με ταχύτητα πολύ μεγαλύτερη από φυσικούς χρήστες. Κάθε διακομιστής διαθέτει ένα αρχείο robots.txt που περιέχει τα ονόματα των bots που τον έχουν ήδη επισκεφτεί, ώστε να μπορεί ο διαχειριστής του ιστοτόπου να χειριστεί ανάλογα τεχνικές βελτιστοποίησης εύρεσης του ιστοτόπου από τις μηχανές αναζήτησης. Υπό αυτή την έννοια, τα bots δε χαρακτηρίζονται ως λογισμικό κακόβουλου περιεχομένου. [51]

Ως λογισμικό κακόβουλου περιεχομένου θεωρούνται τα botnets, που αποτελούν σύνολα από υπολογιστές που έχουν μολυνθεί από κάποιο πρόγραμμα κακόβουλου περιεχομένου και δρουν ως bots, δηλαδή μπορούν να γίνουν αντικείμενο χειρισμού απομακρυσμένα και να μολύνουν και άλλους υπολογιστές ή να τελέσουν πράξεις κακόβουλου τύπου, όπως να αποστείλουν μαζικά μηνύματα spam, να φιλοξενήσουν ιστοσελίδες εξαπάτησης (phishing), να προωθούν διαφημίσεις, να διενεργούν κατανεμημένες επιθέσεις άρνησης υπηρεσίας (DDoS attacks), κ.α. [50]

Η λέξη bot προέρχεται από τη λέξη robot. Τα bots αναφέρονται συχνά και ως Internet bots ή web robots ή www robots.

2.5.8 Rootkits

Ως rootkit χαρακτηρίζεται ο τύπος λογισμικού που χρησιμοποιείται από τους εισβολείς για να καλύψει οποιοδήποτε ίχνος τους που σχετίζεται με την επίθεσή τους στο σύστημα. Συνήθως περιλαμβάνει τη διαγραφή των εγγραφών που αφορούν στην επίθεση από τα αρχεία καταγραφής του συστήματος. Συνακόλουθα, η λειτουργία του rootkit δε γίνεται αντιληπτή από το σύστημα, καθώς ενσωματώνεται σε κάποιο από τα βασικά αρχεία του λειτουργικού συστήματος

και σταδιακά αποκτά τον πλήρη έλεγχο του συστήματος, συμπεριλαμβανομένου και του αντιικού λογισμικού.

Η εγκατάσταση του rootkit μπορεί να είναι αυτοματοποιημένη ή να υλοποιείται από τον εισβολέα αμέσως μετά την επίθεση. Επιπλέον, συνήθως οι επιτιθέμενοι εγκαθιστούν και μία trapdoor στο σύστημα ώστε να μπορούν οποτεδήποτε επιθυμούν σε μελλοντικό χρόνο να συνδεθούν στο σύστημα και μάλιστα με δικαιώματα υπερχρήστη, (superuser).

Το rootkit δεν αποτελεί καθεαυτό λογισμικό κακόβουλου περιεχομένου αλλά εκτελείται αφού ένας υπολογιστής έχει «μολυνθεί» από λογισμικό κακόβουλου περιεχομένου και για να καλύψει ενέργειες των επιτιθέμενων. [52]

3. Εισαγωγη στα Honeypots

3.1 Ορισμός

Τα Honeypot συστήματα είναι παγίδες που έχουν στηθεί για να εντοπίσουν, να εκτρέψουν ή να εξουδετερώσουν τις απόπειρες μη εξουσιοδοτημένης πρόσβασης και χρήσης που πραγματοποιούνται με στόχο πληροφοριακά συστήματα. Μπορεί να αποτελούνται από οποιοδήποτε συστατικό πληροφοριακού συστήματος, συμπεριλαμβανομένων διαδικτυακών σελίδων, μεμονωμένων υπολογιστών ή και δικτύων υπολογιστών. Παραπλανούν τον επιτιθέμενο δημιουργώντας την ψευδαίσθηση ύπαρξης μεγάλου δικτύου υπολογιστών ενώ στην πραγματικότητα είναι μεμονωμένα και παρακολουθούνται. Αποτελούν επομένως παθητικά συστήματα ασφαλείας. Δε συμμετέχουν ενεργά στην ασφάλεια πληροφοριακών συστημάτων, αποτρέποντας δικτυακές επιθέσεις, όπως άλλα συστήματα ασφαλείας, για παράδειγμα τείχη προστασίας ή συστήματα Ανίχνευσης Παρείσφρησης, (firewalls, Intrusion Detection Systems – IDS), αλλά παθητικά συλλέγοντας πληροφορίες. Δεν αντικαθιστούν σε καμία περίπτωση άλλα παραδοσιακά συστήματα ασφάλειας Διαδικτύου αλλά παρέχουν ένα πρόσθετο επίπεδο προστασίας. [4]

Ένα Honeypot είναι ένας πόρος πληροφοριακών συστημάτων του οποίου η αξία έγκειται στη μη εξουσιοδοτημένη ή παράνομη χρήση του [1]. Ουσιαστικά είναι ένας εξυπηρετητής που έχει ρυθμιστεί έτσι ώστε να ανιχνεύει έναν εισβολέα προσομοιώνοντας ένα πραγματικό σύστημα παραγωγής. Φαίνεται να λειτουργεί σαν ένας συνηθισμένος εξυπηρετητής αλλά στην πραγματικότητα όλα τα δεδομένα και οι συναλλαγές που πραγματοποιεί είναι ψεύτικες [2]. Τοποθετημένο είτε εντός είτε εκτός του τείχους προστασίας, χρησιμοποιείται για να καταγράφει και εν τέλει να ενημερώνεται για τις τεχνικές του εισβολέα καθώς και να διαπιστώνει ευπάθειες στο πραγματικό σύστημα.

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

Τα Honeypots μπορούν να εγκατασταθούν οπουδήποτε. Συνηθέστερα, τοποθετούνται εντός τείχους προστασίας για καλύτερο έλεγχο. Κατά μία έννοια, αποτελούν παραλλαγές των κλασσικών συστημάτων ανίχνευσης εισβολών, (Intrusion Detection Systems – IDS), αλλά εστιάζουν περισσότερο στη συγκέντρωση πληροφοριών και την εξαπάτηση.

Εγκαθίστανται έτσι ώστε να αποτελούν εύκολη λεία για τους εισβολείς σε σχέση με τα πραγματικά συστήματα παραγωγής αλλά με μικρές τροποποιήσεις έτσι ώστε η δραστηριότητά τους να μπορεί να ανιχνευτεί και να καταγραφεί. Η λειτουργία τους βασίζεται στην πεποίθηση ότι άπαξ και ένας εισβολέας επιτεθεί σε ένα σύστημα, θα επαναλάβει την επίθεσή του αρκετές φορές. Κατά τη διάρκεια των ακόλουθων αυτών επιθέσεων πρόσθετες πληροφορίες θα συγκεντρωθούν και επιπλέον προσπάθειες πρόσβασης στο σύστημα αρχείων και ασφαλείας θα παρακολουθηθούν και θα αποθηκευτούν.

Οι πιο δημοφιλείς λόγοι για την εγκατάσταση ενός Honeypot είναι οι εξής:

1. Εκμάθηση του πώς οι εισβολείς διερευνούν και προσπαθούν να αποκτήσουν πρόσβαση σε ένα σύστημα. Από τη στιγμή που πραγματοποιείται καταγραφή της δραστηριότητας του εισβολέα, ο χρήστης μπορεί να αποκτήσει γνώση για τις μεθοδολογίες εισβολών για την καλύτερη προστασία των πραγματικών συστημάτων παραγωγής.
2. Συγκέντρωση όλων των σχετιζόμενων πληροφοριών που απαιτούνται για την κατανόηση των κινήσεων και των κινήτρων των εισβολέων. [3]

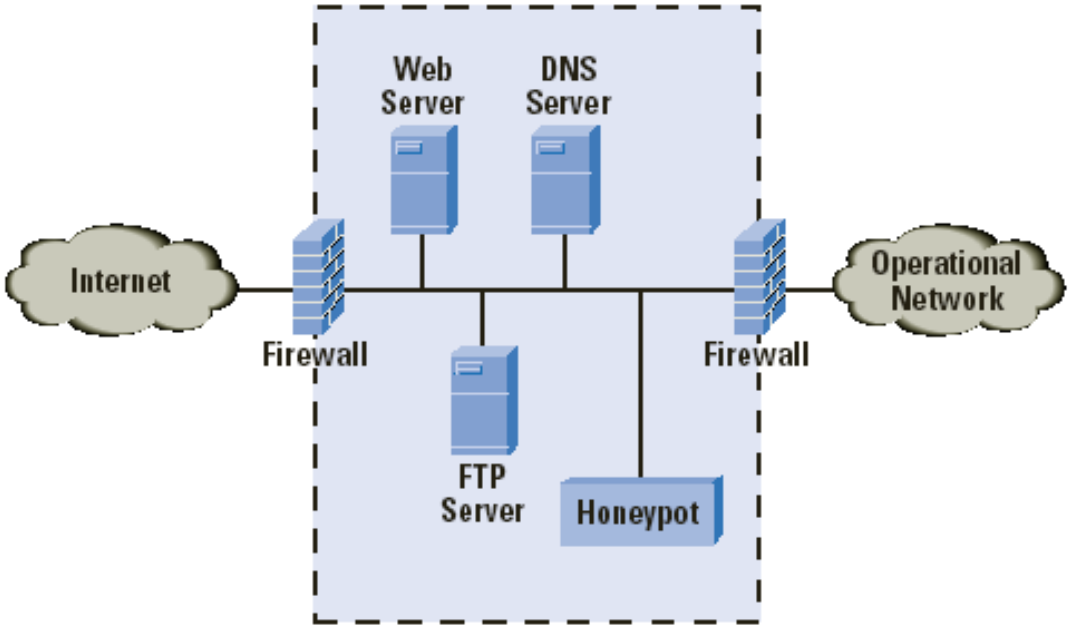
Στην εγκατάσταση και χρήση των Honeypots θεωρείται αποδεκτή η χρήση μεθόδων εξαπάτησης και παραπλάνησης όσον αφορά στους εισβολείς, γεγονός που με τη σειρά του καθορίζει ορισμένες προδιαγραφές που πρέπει να πληρούνται.

1. Το Honeypot πρέπει να φαίνεται όσο πιο γενικό γίνεται. Αν για παράδειγμα, χρησιμοποιείται ένα Microsoft NT σύστημα, θα πρέπει να φαίνεται στους πιθανούς εισβολείς ότι το σύστημα δεν έχει τροποποιηθεί ή ότι μια αποσύνδεση θα λάβει χώρα προτού οι εισβολείς κατορθώσουν να συγκεντρώσουν τον επιθυμητό όγκο δεδομένων.

2. Χρειάζεται προσοχή στην κίνηση που επιτρέπεται σε έναν εισβολέα να στείλει πίσω στο Διαδίκτυο ώστε ο υπολογιστής να μην αποτελέσει ένα σημείο εκκίνησης για επιθέσεις εναντίον άλλων οντοτήτων στο Διαδίκτυο. Το γεγονός αυτό αποτελεί και έναν από τους λόγους γιατί το Honeypot εγκαθίσταται εντός του τείχους προστασίας.
3. Χρειάζεται προσοχή στα δεδομένα που θα περιέχει το Honeypot ώστε να φαίνεται νόμιμο και οι εισβολείς να θεωρούν ότι εκπλήρωσαν το στόχο τους. [3]

Παράλληλα, κάποια ζητήματα που πρέπει να ληφθούν σοβαρά υπόψη κατά το σχεδιασμό και τη χρήση των Honeypots είναι ότι τα δεδομένα που έχουν συγκεντρωθεί από τα Honeypots δεν μπορούν να χρησιμοποιηθούν ως αποδεικτικά τεκμήρια για την άσκηση ποινικής δίωξης αλλά και το ότι hacking οργανώσεις συχνά βάλλουν κατά της δραστηριότητας των Honeypots, που στήνουν παγίδες εναντίον τους αλλά και τους μετατρέπουν ενδεχομένως σε στόχο για άλλους εισβολείς. [3]

Στην εικόνα 3-1 διαφαίνεται ένα τυπικό μοντέλο συστήματος Honeypot, στο οποίο το Honeypot έχει τοποθετηθεί εντός τείχους ασφαλείας.



Εικόνα 3-1: Τυπικό μοντέλο Honeypot εντός τείχους προστασίας. [63]

4. Κατηγοριοποιηση των Honeypots

Τα Honeypots μπορούν να κατηγοριοποιηθούν ανάλογα με το σκοπό που έχουν, με το βαθμό αλληλεπίδρασής τους με λογισμικό κακόβουλου τύπου και με την αρχιτεκτονική υλοποίησής τους.

4.1 Κατηγοριοποίηση με βάση την αλληλεπίδραση

Τα Honeypots ανάλογα με το βαθμό αλληλεπίδρασής τους με λογισμικό κακόβουλου τύπου, κατηγοριοποιούνται σε Honeypots:

- χαμηλής αλληλεπίδρασης,
- μεσαίας αλληλεπίδρασης και
- υψηλής αλληλεπίδρασης.

4.1.1 Honeypots χαμηλής αλληλεπίδρασης

Τα χαμηλής αλληλεπίδρασης Honeypots ονομάζονται έτσι ακριβώς λόγω της περιορισμένης αλληλεπίδρασης που μπορεί να έχει ένας επιτιθέμενος ή γενικά κακόβουλο λογισμικό με το σύστημα. Προσομοιώνουν μόνο τμήματα ενός λειτουργικού συστήματος, δηλαδή υπηρεσίες και αδυναμίες. Για αυτό το λόγο ακριβώς, ο επιτιθέμενος δεν μπορεί να αποκτήσει πρόσβαση σε όλο το λειτουργικό σύστημα και να εκμεταλλευτεί το λογισμικό για περαιτέρω δικτυακές επιθέσεις [6]. Για παράδειγμα, η HTTP υπηρεσία σε ένα χαμηλής αλληλεπίδρασης Honeypot, θα υποστήριζε μόνο τις εντολές που είναι απαραίτητες για την αναγνώριση μίας επίθεσης [4].

Επομένως τα χαμηλής αλληλεπίδρασης Honeypots έχουν περιορισμένες δυνατότητες αλλά είναι χρήσιμα για τη συλλογή πληροφοριών για υψηλότερο επίπεδο, όπως για δραστηριότητες των worms ή για προγράμματα παρακολούθησης δικτύου, (network sniffers). Μπορούν επίσης να χρησιμοποιηθούν για την ανάλυση δραστηριότητας αποστολέων ανεπιθύμητης αλληλογραφίας, (spammers), ή για τη λήψη αντιμέτρων εναντίον ιών τύπου worms. [6]

Είναι γραμμένα συνήθως σε γλώσσες σεναρίου, (script-based), και τις χρησιμοποιούν και για την αλληλεπίδραση με τον επιτιθέμενο. Είναι εύκολα στην εγκατάσταση και τη συντήρησή τους και δεν απαιτούν πολλούς υπολογιστικούς πόρους για τη λειτουργία τους. Επιπρόσθετα, η χρήση τους είναι απλή, δε διατρέχουν τον κίνδυνο να παραβιαστούν και παράγουν ένα σχετικά μικρό όγκο δεδομένων για ανάλυση. Αποτελούν ιδανική λύση για έναν αρχάριο χρήστη που θέλει να γνωρίσει τη λειτουργικότητα των Honeypots. [5]

Τα πλεονεκτήματα των χαμηλής αλληλεπίδρασης Honeypots αποτελούν ταυτόχρονα όμως και μειονεκτήματα γιατί οι περιορισμένες δυνατότητες τους μπορούν να γεμίσουν υποψίες έναν εισβολέα για τον εάν αντιμετωπίζει ένα πραγματικό σύστημα ενώ ο παραγόμενος μικρός όγκος δεδομένων σημαίνει και μικρότερη καταγραφή πληροφοριών σχετικά με τις επιθέσεις και επομένως και μικρότερη εκπαιδευτική αξία.

Για παράδειγμα, ένα Honeypot χαμηλής αλληλεπίδρασης μπορεί να προσομοιώσει έναν τυπικό εξυπηρετητή Unix που εκτελεί διάφορες υπηρεσίες όπως Telnet και FTP. Ένας εισβολέας θα μπορούσε να εκτελέσει το Telnet προς το Honeypot, να αποκτήσει ένα banner που να δηλώνει την κατάσταση του λειτουργικού συστήματος καθώς και μία προτροπή για είσοδο, (login prompt), και να προσπαθήσει να εισέλθει στο σύστημα. Το Honeypot θα συγκεντρώσει αυτές τις προσπάθειες αλλά δεν υπάρχει πραγματικό λειτουργικό σύστημα για να εισέλθει ο εισβολέας. Επομένως η αλληλεπίδραση του εισβολέα περιορίζεται σε προσπάθειες σύνδεσης. [1]

Ένα άλλο παράδειγμα είναι ένας προσομοιωθείς εξυπηρετητής FTP όπου ίσως ο εισβολέας μπορεί μέσω ανώνυμου λογαριασμού να αποκτήσει σύνδεση στο Honeypot και να κάνει λήψη ενός αντιγράφου του αρχείου κωδικών πρόσβασης του συστήματος, μια τακτική που χρησιμοποιείται γενικά από πολλούς εισβολείς. Ωστόσο, ο ανώνυμος λογαριασμός θα ήταν η μόνος με πρόσβαση στο σύστημα ενώ το αρχείο κωδικού πρόσβασης δε θα ισχύει γιατί θα είναι ένα πλαστό αρχείο δημιουργημένο από το Honeypot που θα χρησιμοποιείται για την εξαπάτηση

του εισβολέα. Η αλληλεπίδραση επομένως περιορίζεται στις προσπάθειες σύνδεσης, την ανώνυμη πρόσβαση και τη δυνατότητα λήψης πλαστού αρχείου κωδικών πρόσβασης. [1]

4.1.2 Honeypots υψηλής αλληλεπίδρασης

Τα Honeypots υψηλής αλληλεπίδρασης προσομοιώνουν όλες τις πτυχές ενός λειτουργικού συστήματος, μιμούνται όλες τις δραστηριότητες ενός πραγματικού συστήματος που φιλοξενεί μια ποικιλία από υπηρεσίες και επομένως μπορούν να παραβιαστούν, επιτρέποντας σε έναν εισβολέα να αποκτήσει πλήρη πρόσβαση στο σύστημα και να το χρησιμοποιήσει για περαιτέρω δικτυακές επιθέσεις. Επειδή αποτελούν πλήρη λειτουργικά συστήματα, ο εντοπισμός τους από εισβολείς απαιτεί περισσότερο χρόνο αλλά έτσι τα Honeypots υψηλής αλληλεπίδρασης καταγράφουν ένα μεγάλο αριθμό επιθέσεων, παράγοντας ένα μεγάλο όγκο δεδομένων και ένα αυξημένο επίπεδο γνώσης για τους διαχειριστές τους. [6]

Έχουν σχεδιαστεί έτσι ώστε να μπορούν να καταγράψουν το μέγιστο της πληροφορίας για τις μεθόδους του επιτιθέμενου. Κάθε εντολή ή εφαρμογή που θα περίμενε να λάβει ένας τελικός χρήστης του συστήματος είναι διαθέσιμη από το Honeypot και γενικά υπάρχει ελάχιστος ή και μηδαμινός περιορισμός στις δυνατότητες του εισβολέα αφού επιτύχει την επίθεσή του. [1]

Σύμφωνα με τις τελευταίες έρευνες στην τεχνολογία των υψηλής αλληλεπίδρασης Honeypots, με τη χρήση εικονικών μηχανημάτων πολλά Honeypots μπορούν να εγκατασταθούν και να λειτουργήσουν σε έναν μόνο υπολογιστή. Έτσι, ακόμα κι αν το Honeypot παραβιαστεί, μπορεί να επανακτηθεί πολύ γρήγορα.

Γενικά, τα υψηλής αλληλεπίδρασης Honeypots παρέχουν περισσότερη ασφάλεια αφού είναι δύσκολο να ανιχνευθούν αλλά είναι ιδιαίτερα δαπανηρά να συντηρηθούν. Εάν δε χρησιμοποιηθούν εικονικά μηχανήματα, ένα Honeypot διατηρείται σε έναν υπολογιστή, που είναι ομοίως εξαιρετικά ακριβό.

Σε σχέση με τα χαμηλής αλληλεπίδρασης Honeypots, είναι πιο δύσκολα στη χρήση και ανάπτυξή τους, καθώς απαιτούν τη ρύθμιση πολλών παραμέτρων ενώ απαιτούν και περισσότερους υπολογιστικούς πόρους για τη λειτουργία τους. Η ανάλυση του μεγάλου όγκου δεδομένων τον οποίο παράγουν είναι δύσκολη και απαιτεί πολύωρη εργασία, ενώ όπως ήδη αναφέρθηκε ο επιτιθέμενος μπορεί να τα χρησιμοποιήσει ως πλατφόρμες για επίθεση σε άλλα συστήματα. [5]

4.1.3 Honeypots μεσαίας αλληλεπίδρασης

Τα Honeypots μεσαίας αλληλεπίδρασης προσφέρουν λειτουργικότητα ενδιάμεση των Honeypots υψηλής αλληλεπίδρασης και των Honeypots χαμηλής αλληλεπίδρασης. Προσφέρουν στους εισβολείς μεγαλύτερη δυνατότητα αλληλεπίδρασης σε σχέση με τα Honeypots χαμηλής αλληλεπίδρασης αλλά μικρότερη σε σχέση με τα Honeypots υψηλής αλληλεπίδρασης. Αναμένουν συγκεκριμένο τύπο δραστηριότητας και έχουν σχεδιαστεί έτσι ώστε να δίνουν συγκεκριμένες απαντήσεις πέρα από αυτές που θα έδινε ένα Honeypot χαμηλής αλληλεπίδρασης. [1]

Το βασικό χαρακτηριστικό των Honeypots μεσαίας αλληλεπίδρασης είναι η εικονικοποίηση σε επίπεδο εφαρμογής. Αυτά τα Honeypots δεν αποσκοπούν στην πλήρη προσομοίωση ενός πλήρως λειτουργικού περιβάλλοντος ούτε και υλοποιούν όλες τις λεπτομέρειες του πρωτοκόλλου της εφαρμογής. Το είδος αυτό των Honeypots παρέχει επαρκείς απαντήσεις τις οποίες περιμένουν οι εισβολείς σε συγκεκριμένες θύρες οι οποίες ξεγελούν τον εισβολέα ώστε να τους στείλει χρήσιμη πληροφορία.

Όταν ολοκληρωθεί η διαδικασία αυτή ο κώδικας κελύφους, (shell code), εξάγεται και αναλύεται. Το μεσαίας αλληλεπίδρασης τότε Honeypot μιμείται τις δράσεις που θα έκανε ο κώδικας κελύφους για να κατεβάσει το κακόβουλο λογισμικό. Επομένως το Honeypot πρέπει να διαθέσει κάποιο εικονικό σύστημα αρχείων, καθώς και εικονικά προγράμματα του λειτουργικού συστήματος προς λήψη. Το Honeypot μπορεί τότε να κατεβάσει το κακόβουλο λογισμικό και να το αποθηκεύσει τοπικά ή να το υποβάλλει κάπου αλλού για ανάλυση. [7]

Για παράδειγμα, έστω ότι υπάρχει ανίχνευση για ιούς τύπου worm όταν πρόκειται για συγκεκριμένες ευπάθειες IIS, (Internet Information Services). Θα μπορούσε να σχεδιαστεί ένα

Honeypot που να μιμείται τον Microsoft IIS web server και να περιλαμβάνει και την πρόσθετη λειτουργικότητα που συνήθως συνοδεύει την εφαρμογή. Ο IIS web server που έχει προσομοιωθεί θα μπορούσε στη συνέχεια να προσαρμοστεί ώστε να παρουσιάζει τη λειτουργικότητα ή συμπεριφορά την οποία θα αναζητούσε ο συγκεκριμένος ιός τύπου worm. Οποτεδήποτε μία HTTP σύνδεση γινόταν προς το Honeypot, το Honeypot θα απαντούσε ως IIS web server, δίνοντας έτσι στον εισβολέα τη δυνατότητα να αλληλεπιδράσει με τον πραγματικό IIS web server. Αυτό το επίπεδο αλληλεπίδρασης είναι σαφώς καλύτερο από αυτό του Honeypot χαμηλής αλληλεπίδρασης που πιθανότατα σε αυτή την περίπτωση θα παρουσίαζε απλά ένα HTTP banner. Όσον αφορά στο worm στόχος ήταν να επιτεθεί στο honeypot ώστε να καταγραφεί το payload για μελλοντική ανάλυση. Ωστόσο στο worm δεν είχε δοθεί ένα πλήρες λειτουργικό σύστημα για να αλληλεπιδράσει, περιορίζοντας έτσι τον επιπλέον κίνδυνο εκμετάλλευσης. [1]

4.2 Κατηγοριοποίηση με βάση το σκοπό

Τα Honeypots ανάλογα με το σκοπό για τον οποίο εγκαθίστανται, κατηγοριοποιούνται σε Honeypots:

- έρευνας και
- παραγωγής.

4.2.1 Honeypots έρευνας

Τα Honeypots έρευνας έχουν σχεδιαστεί έτσι ώστε να αποκτούν πληροφορίες που αφορούν στην hacking κοινότητα. Χρησιμοποιούνται για να συλλέξουν πληροφορίες σχετικά με τις γενικές απειλές ασφαλείας που αντιμετωπίζουν οι οργανισμοί, επιτρέποντας στους οργανισμούς να προστατευθούν καλύτερα ενάντια σε αυτές τις απειλές. Η πρωταρχική λειτουργία τους είναι να μελετήσουν τους τρόπους με τους οποίους οι επιτιθέμενοι εξελίσσονται ώστε να διαπιστώσουν τις τακτικές που ακολουθούν για τις επιθέσεις τους. Είναι πολύπλοκα στην εγκατάσταση και την παραμετροποίηση και χρονοβόρα και καταγράφουν μεγάλο όγκο δεδομένων.

Πολύ λίγα συνεισφέρει ένα Honeypot έρευνας στην ασφάλεια ενός οργανισμού, αν και τα διδάγματα που αντλούνται χρησιμοποιούνται για τη βελτίωση της πρόληψης των επιθέσεων, της ανίχνευσης και της απόκρισης. Χρησιμοποιούνται συνήθως από πανεπιστήμια, κυβερνήσεις και στρατιωτικούς οργανισμούς που ενδιαφέρονται να μάθουν περισσότερα για τις απειλές σε ερευνητικό επίπεδο.

Τα Honeypots έρευνας παρέχουν μία πλατφόρμα για τη μελέτη κυβερνο-επιθέσεων. Οι εισβολείς παρακολουθούνται σε δράση και καταγράφονται οι κινήσεις τους βήμα-βήμα καθώς επιτίθενται και καταλαμβάνουν το σύστημα. Η συλλογή αυτών των δεδομένων αποτελεί ένα από τα σημαντικότερα χαρακτηριστικά των Honeypots έρευνας. Βοηθούν επίσης στην ανάλυση και την πρόβλεψη των επιθέσεων καθώς και στην ανακάλυψη νέων ιών τύπου worms. [8]

4.2.2 Honeypots παραγωγής

Τα Honeypots παραγωγής χρησιμοποιούνται από εταιρείες ή οργανισμούς για την ασφάλεια των πληροφοριακών συστημάτων τους μέσω του ελέγχου δικτυακής κίνησης. Απαιτούν μικρότερη λειτουργικότητα σε σχέση με τα Honeypots έρευνας και είναι ευκολότερη η εγκατάσταση και η λειτουργία τους. Αναγνωρίζουν τις μεθοδολογίες των επιθέσεων αλλά δίνουν λιγότερη πληροφορία για τους εισβολείς σε σχέση με τα Honeypots έρευνας. Για παράδειγμα, με χρήση των Honeypots παραγωγής μπορεί ένας χρήστης να μάθει την προέλευση των εισβολέων και τις μεθόδους που χρησιμοποιούν αλλά όχι και την ακριβή ταυτότητά τους.

Τα Honeypots παραγωγής τείνουν να αντικατοπτρίζουν το δίκτυο παραγωγής της εταιρείας ή συγκεκριμένες υπηρεσίες, προσκαλώντας έτσι τους εισβολείς να αλληλεπιδράσουν μαζί τους ώστε να αποκαλυφθούν οι τρέχουσες ευπάθειες του δικτύου. Η αποκάλυψη αυτών των ευπαθειών και η προειδοποίηση των διαχειριστών για πιθανές επιθέσεις μειώνει τον κίνδυνο των εισβολών. Για αυτό ακριβώς το λόγο, τα Honeypots παραγωγής ως μηχανισμός πρόληψης έχουν ελάχιστη αξία. Οι βέλτιστες πρακτικές που ενδείκνυνται για πρόληψη επιθέσεων αφορούν σε κλασσικά συστήματα ασφαλείας, όπως για παράδειγμα τείχη

προστασίας και συστήματα Ανίχνευσης Παρείσφρησης. Τα δεδομένα που παράγονται από το Honeypot συλλέγονται και μελετούνται για το σχεδιασμό αντιμέτρων ενάντια σε μελλοντικές απειλές.

Η ιδιαίτερη αξία των Honeypots έγκειται στην ανίχνευση επιθέσεων. Επειδή είναι πιο απλά στη λειτουργία σε σχέση με τα IDS, μπορούν να αντιμετωπίσουν τις προκλήσεις που αντιμετωπίζουν τα IDS σαφώς καλύτερα, αφού βρίσκουν λιγότερα ψευδώς θετικά και ψευδώς αρνητικά αποτελέσματα. Υπάρχουν πολλές περιπτώσεις στις οποίες ένα IDS δεν μπορεί να εκδώσει μια προειδοποίηση, όπως η επίθεση να είναι πολύ πρόσφατη ή ο κανόνας που να ταιριάζει με αυτή την επίθεση να προκαλεί πολλά ψευδώς θετικά αποτελέσματα ή να βλέπει πολλή κίνηση και να απορρίπτει τα πακέτα του. Ψευδώς θετικά αποτελέσματα προκύπτουν όταν ένα IDS προκαλεί πολλές παραπάνω από όσες θα έπρεπε προειδοποιήσεις σε δίκτυο με θεωρούμενη κανονική κίνηση. Οι προειδοποιήσεις αυτές σύντομα αγνοούνται ή οι κανόνες που ενεργοποιούνται με αυτές τις προειδοποιήσεις τροποποιούνται αλλά έτσι στη συνέχεια πραγματικές επιθέσεις ενδέχεται να μη γίνουν καν αντιληπτές. Επιπρόσθετα, τα IDS συστήματα αντιμετωπίζουν πρόβλημα όταν υπάρχει μεγάλος όγκος δεδομένων για ανάλυση σε ένα μεγάλο σύστημα με αυξημένη δικτυακή κίνηση. Τα Honeypots μπορούν να αντιμετωπίσουν αυτές τις προκλήσεις, αφού δεν έχουν καθόλου παραγωγική δραστηριότητα, οπότε όλη η κίνηση που στέλνεται σε ένα Honeypot είναι σχεδόν σίγουρο ότι θα είναι μη εξουσιοδοτημένη, πράγμα που σημαίνει καθόλου ψευδώς θετικά αποτελέσματα, καθόλου ψευδώς αρνητικά και όχι μεγάλα σύνολα δεδομένων για ανάλυση.

Παράλληλα, όταν εντοπιστεί μία επίθεση, το μηχάνημα μπορεί να τεθεί εκτός δικτύου ώστε να πραγματοποιηθεί ενδελεχής έρευνα και ανάλυση. Ωστόσο, αυτή η διαδικασία είναι αν όχι ανέφικτη, αρκετά δύσκολη σε ένα σύστημα παραγωγής.

Γενικά, οι εμπορικές οργανώσεις αποκομίζουν τα πιο άμεσα οφέλη από τα Honeypots παραγωγής. [8]

Οι ανωτέρω κατηγοριοποιήσεις των Honeypots αποτελούν απλά έναν οδηγό για την αναγνώριση του σκοπού τους, οι διακρίσεις δεν είναι απόλυτες. Πολύ συχνά το ίδιο Honeypot μπορεί να είναι και παραγωγής και έρευνας. Δεν είναι τόσο το πώς έχει κατασκευαστεί ένα Honeypot αυτό που καθορίζει την κατηγορία του, αλλά το πώς χρησιμοποιείται.

4.3 Κατηγοριοποίηση με βάση την υλοποίηση

Τα Honeypots ανάλογα με την υλοποίησή τους, χωρίζονται σε δύο κατηγορίες:

- φυσικά και
- εικονικά.

4.3.1 Φυσικά Honeypots

Τα φυσικά Honeypots αποτελούν εγκαταστάσεις λειτουργικών συστημάτων σε υπολογιστές οι οποίοι παρακολουθούνται. Εκτελούνται σε πραγματικά συστήματα και το καθένα έχει τη δική του διεύθυνση IP. Μπορούν να τρέχουν σε οποιοδήποτε λειτουργικό σύστημα – Linux, Unix, Windows, Mac Os κ.α.

Είναι εύκολα στην εγκατάσταση και μπορεί να γίνει εγκατάσταση οποιουδήποτε λειτουργικού συστήματος, δεν υπάρχουν περιορισμοί. Ωστόσο, τα συστήματα αυτά έχουν και ορισμένα μειονεκτήματα με το σοβαρότερο όλων να είναι ότι μόνο ένα λειτουργικό σύστημα μπορεί να εκτελεστεί σε έναν υπολογιστή με άμεσο αποτέλεσμα την ανάγκη ανακατανομής των χρησιμοποιούμενων υπολογιστικών πόρων. Παράλληλα, η επανεγκατάσταση ενός συστήματος που έχει δεχθεί επίθεση μπορεί να προκαλέσει περαιτέρω προβλήματα αλλά και να είναι χρονοβόρα, ακόμα και αν έχουν διατηρηθεί αντίγραφα ασφαλείας. Μη αυτόματη πρόσβαση απαιτείται σε πολλές περιπτώσεις ώστε να μην υπάρχει περίπτωση ο επιτιθέμενος να εμποδίσει τη λειτουργία μιας εγκατάστασης. Η παρακολούθηση τέτοιων συστημάτων μπορεί να επιτευχθεί μόνο από εξωτερικούς μηχανισμούς, δεδομένου ότι όλες οι αλλαγές στο λειτουργικό σύστημα του Honeypot μπορούν να εντοπιστούν από έμπειρους εισβολείς που προσπαθούν να αποκτήσουν πλήρη πρόσβαση στο μηχάνημα.

4.3.2 Εικονικά Honeypots

Τα εικονικά Honeypots είναι συστήματα που εγκαθίστανται και λειτουργούν εικονικά πάνω από ένα άλλο υπάρχον κεντρικό λειτουργικό σύστημα.

Η χρήση εικονικών Honeypots προσφέρει πολλά πλεονεκτήματα. Ανάλογα με το κεντρικό λειτουργικό σύστημα και την τεχνική εικονικής διαμόρφωσης που θα ακολουθηθεί, το φιλοξενούμενο λειτουργικό σύστημα μπορεί να είναι τελείως διαφορετικό. Τα εικονικά λειτουργικά συστήματα μπορούν να εγκατασταθούν παράλληλα σε έναν υπολογιστή και σε ορισμένες περιπτώσεις πολλά και διαφορετικά λειτουργικά συστήματα μπορούν να συνυπάρχουν. Τα εικονικά Honeypots μπορούν να γίνουν στόχος επίθεσης χωρίς να κινδυνεύει το λειτουργικό σύστημα στο οποίο φιλοξενούνται. Ο διαχειριστής του συστήματος μπορεί να καταγράφει την κατάσταση του εικονικού Honeypot οποιαδήποτε στιγμή, οπότε και να καταγράφει τις κινήσεις του εισβολέα που έχει επιτεθεί στο Honeypot. Για την επανεγκατάσταση των εικονικών Honeypots απαιτείται από το διαχειριστή του συστήματος μόνο η ρύθμιση των εικονικών κόμβων κατά τον επιθυμητό τρόπο.

Φυσικά, το λειτουργικό σύστημα του κεντρικού υπολογιστή δεν πρέπει να είναι ευάλωτο και πρέπει να παραμένει κρυφό από τον επιτιθέμενο. Αυτό μπορεί να επιτευχθεί με την προσάρτηση του κεντρικού λειτουργικού συστήματος σε ένα ξεχωριστό επίπεδο διασύνδεσης του δικτύου ή σε ένα σειριακό καλώδιο που είναι προσβάσιμο μόνο από το διαχειριστή του Honeypot.

Το μεγάλο μειονέκτημα των εικονικών μηχανημάτων είναι συχνά μια έλλειψη ακρίβειας του επιπέδου εικονικοποίησης που οδηγεί σε καταστάσεις κατά τις οποίες ο εισβολέας μπορεί να αναγνωρίσει αν επιτέθηκε σε ένα πραγματικό σύστημα. Αυτό μπορεί να θεωρηθεί και πλεονέκτημα όμως καθώς πλέον οι υπηρεσίες παροχής Διαδικτύου, (Internet Service Providers – ISPs), χρησιμοποιούν τεχνικές εικονικοποίησης για να βελτιώσουν την αποδοτικότητα των διακομιστών τους και επομένως οι εισβολείς να μην μπορούν τελικά να αναγνωρίσουν αν ένα εικονικό λειτουργικό σύστημα αποτελεί Honeypot. [9]

5. Συμβολη των Honeypots στην Ασφαλεια

Τα Honeypots συμβάλουν ιδιαίτερα στην Ασφάλεια Υπολογιστικών Συστημάτων καλύπτοντας τα κενά ασφαλείας που αφήνουν τα τείχη προστασίας αλλά και τα Συστήματα Ανίχνευσης Παρείσφρησης. Στο παρόν κεφάλαιο θα αναλυθούν τα πλεονεκτήματα και μειονεκτήματα της χρήσης των Honeypot καθώς και οι ιδιαίτερες δυνατότητές τους που συμβάλουν στην αύξηση της ασφάλειας ενός δικτύου, ενός πληροφοριακού συστήματος εν γένει.

5.1 Πλεονεκτήματα της χρήσης των Honeypots

1) Τα Honeypots παράγουν ελάχιστες λανθασμένες ειδοποιήσεις κι αυτό γιατί ο μόνος λόγος ύπαρξής τους είναι να γίνουν αντικείμενα επίθεσης. Επομένως όλη η κίνηση που πραγματοποιείται από και προς αυτά καθώς και όλη η δραστηριότητα που καταγράφεται σε αυτά θεωρείται κακόβουλου περιεχομένου και θα πρέπει να εξετάζεται και να ερευνάται ως τέτοια. Απόρροια του γεγονότος αυτού είναι η άμεση αντίληψη των επιθέσεων από τους διαχειριστές του δικτύου και η λήψη κατάλληλων πρακτικών ασφαλείας αλλά και η ανακάλυψη νέων απειλών. Πρωτοεμφανιζόμενες επιθέσεις μπορούν να διαπιστωθούν άμεσα και να καταγραφούν.
2) Τα Honeypots προσφέρουν τη δυνατότητα πλήρους καταγραφής των κινήσεων και της δραστηριότητας ενός εισβολέα. Έτσι, οι διαχειριστές στη συνέχεια μπορούν να επεξεργαστούν το μεγάλο αυτό όγκο δεδομένων για να κατανοήσουν πλήρως και με ακρίβεια τη φύση και το σκοπό του εισβολέα ώστε να λάβουν τα κατάλληλα αντίμετρα.
3) Οι προδιαγραφές απαιτήσεων σε υλικό για την εγκατάσταση των Honeypots είναι προκαθορισμένες και συνήθως δεν περιλαμβάνουν χρήση τελευταίων τεχνολογιών. Τα πιο απλά Honeypots μπορούν να στηθούν σε οποιοδήποτε προσωπικό υπολογιστή μέτριων δυνατοτήτων. Παρά το γεγονός όμως ότι απαιτούν χαμηλές προδιαγραφές σε υλικό, μπορούν να λειτουργήσουν παράλληλα με τελευταίες τεχνολογίες, όπως για παράδειγμα την IPv6.
4) Τα Honeypots επειδή ακριβώς λειτουργούν σαν παγίδες για τους επιτιθέμενους, δρουν αποπροσανατολιστικά για τους εισβολείς απομακρύνοντας τους από το κυρίως δίκτυο και τις κρίσιμες λειτουργίες του και ωθώντας τους προς το ίδιο το Honeypot.
5) Ο όγκος δεδομένων που αφορά στην αλληλεπίδραση με τους εισβολείς και καταγράφεται από το Honeypot δεν περιέχει καμία λανθασμένη ειδοποίηση και επομένως μπορεί ευκολότερα να γίνει αντικείμενο επεξεργασίας και να αναλυθεί.
6) Τα Honeypots προσομοιώνοντας πραγματικά συστήματα παραγωγής, συλλέγουν επιθέσεις που στόχο θα είχαν τα πραγματικά συστήματα παραγωγής. Ομοίως προσομοιώνοντας υπηρεσίες, συλλέγουν επιθέσεις που θα στόχευαν αυτές τις υπηρεσίες ή θα χρησιμοποιούσαν αδυναμίες των υπηρεσιών αυτών για να επιτεθούν σε πραγματικά συστήματα παραγωγής. Τα Honeypots επομένως προλαμβάνουν επιθέσεις. [55]

5.2 Μειονεκτήματα της χρήσης των Honeypots

1) Όπως προαναφέρθηκε ο μόνος λόγος ύπαρξης των Honeypots είναι να δεχτούν επιθέσεις. Οπότε αν κανείς δεν επιτεθεί σε αυτά ή αν κάποιος εισβολέας αντιληφθεί την παρουσία τους, τα παρακάμψει και εξαπολύσει την επίθεση προς το υπόλοιπο δίκτυο, προς πραγματικά συστήματα παραγωγής, τότε τα Honeypots καθίστανται άχρηστα.
2) Το Honeypot ενδέχεται να αποτελέσει σημείο εκκίνησης για επιθέσεις εναντίον άλλων οντοτήτων στο Διαδίκτυο. Αυτό μπορεί να συμβεί συχνότερα σε Honeypots υψηλής αλληλεπίδρασης και συνιστά και έναν από τους λόγους γιατί το Honeypot εγκαθίσταται εντός του τείχους προστασίας.
3) Η χρήση των Honeypots μπορεί να αναγνωριστεί από εξειδικευμένους στην Ασφάλεια Πληροφοριακών Συστημάτων προγραμματιστές ή μέσω ειδικών προγραμμάτων.

4) Ο λόγος ύπαρξής των Honeypots είναι η λήψη επιθέσεων. Επομένως έτσι αυξάνονται οι επιτιθέμενοι σε ένα δίκτυο και μειώνεται η ασφάλεια στο δίκτυο. Συνακόλουθα, τα Honeypots έτσι επιβαρύνουν και το φόρτο του δικτύου και μειώνουν και τη ρυθμαπόδοση, (throughput), του δικτύου. [55]

6. HONEYPOTS – ΑΝΑΛΥΤΙΚΑ (ΒΑΣΕΙ ΑΛΛΗΛΕΠΙΔΡΑΣΗΣ)

6.1 Honeypots χαμηλής αλληλεπίδρασης

6.1.1 Dionaea

Το Dionaea είναι ένα χαμηλής αλληλεπίδρασης Honeypot που σκοπό έχει να αποκτά αντίγραφα λογισμικού κακόβουλου τύπου εξομοιώνοντας διάφορα πρωτόκολλα. Θεωρείται ο απόγονος του χαμηλής αλληλεπίδρασης Honeypot Nepenthes. Περιέχει ενσωματωμένη την Python ως γλώσσα σεναρίου για ευκολία και περισσότερη ευελιξία, χρησιμοποιεί τη βιβλιοθήκη libemu για τον εντοπισμό και χειρισμό κώδικα κελύφους και υποστηρίζει την τεχνολογία IPv6 καθώς και την τεχνολογία Transport Layer Security, (TLS), πρωτόκολλο ασφάλειας επικοινωνίας που λειτουργεί σε επίπεδο ανώτερο του Διαδικτύου και απαιτείται από ορισμένα άλλα πρωτόκολλα.

Υποστηριζόμενα πρωτόκολλα

Το Dionaea προσαρμόζει τα πρωτόκολλα που χρησιμοποιούνται από λογισμικό κακόβουλου τύπου με χρήση της Python γλώσσας. Παρακάτω αναλύονται όλα τα υποστηριζόμενα από το Dionaea πρωτόκολλα.

SMB, (Server Message Protocol): Είναι το βασικό πρωτόκολλο που προσομοιώνει το Dionaea. Ένα δικτυακό πρωτόκολλο που λειτουργεί σε επίπεδο εφαρμογής καθορίζοντας την πρόσβαση σε αρχεία, εκτυπωτές, θύρες σε ένα δίκτυο. Είναι ευρύτερα γνωστό με την ονομασία CIFS, (Common Internet File System), και χρησιμοποιείται κυρίως σε υπολογιστές με λειτουργικό Microsoft Windows. Το SMB αποτελεί έναν ιδιαίτερα ελκυστικό στόχο για worms, για αυτό και προσομοιώνεται από το Dionaea. Το Dionaea χρησιμοποιεί τη θύρα 445 για να δέχεται συνδέσεις προς το SMB πρωτόκολλο.

HTTP, HTTPS: Το Dionaea χρησιμοποιεί τη θύρα 80 για να δέχεται συνδέσεις προς το HTTP πρωτόκολλο και τη θύρα 443 για το πρωτόκολλο HTTPS, ωστόσο τα δεδομένα που συλλέγονται δε χρησιμοποιούνται και δεν αναλύονται περαιτέρω από το Dionaea.

FTP, (File Transfer Protocol): Το Dionaea προσφέρει έναν βασικό FTP διακομιστή στη θύρα 21 ο οποίος μπορεί να χρησιμοποιηθεί για δημιουργία καταλόγων, για αποστολή και λήψη αρχείων.

TFTP, (Trivial File Transfer Protocol): Το Dionaea προσφέρει έναν TFTP διακομιστή στη θύρα 69, ο οποίος μπορεί να χρησιμοποιηθεί για το διαμοιρασμό αρχείων. Αρχικά σχεδιάστηκε ώστε να εξεταστεί ο κώδικας που αφορούσε στις UDP, (User Datagram Protocol), συνδέσεις.

MSSQL, (Microsoft SQL Server): Το Dionaea υποστηρίζει το πρωτόκολλο TDS, (Tabular Data Stream), που χρησιμοποιείται από τον Microsoft SQL Server. Χρησιμοποιείται η θύρα 1433 και το Dionaea επιτρέπει τη σύνδεση στον SQL διακομιστή και την εκτέλεση επερωτήσεων προς τη βάση δεδομένων αλλά δεν υπάρχει βάση δεδομένων, οπότε οι αιτήσεις προς τη βάση δε λαμβάνουν απόκριση και τα δεδομένα δεν υφίστανται καμία επεξεργασία.

MySQL: Το Dionaea υποστηρίζει το πρωτόκολλο MySQL στη θύρα 3306. Όλες οι επερωτήσεις προς τη MySQL βάση δεδομένων προωθούνται σε μία τοπική SQLite βάση δεδομένων.

SIP: Το Dionaea υποστηρίζει το πρωτόκολλο SIP για χρήση τηλεφωνίας πάνω από IP, (Voice Over IP – VoIP). Σε αντίθεση με άλλα VoIP Honeypots, το Dionaea δε συνδέεται με κάποιον εξωτερικό διακομιστή VoIP. Αναμένει για εισερχόμενα SIP μηνύματα, όπως OPTIONS, INVITE, ACK, CANCEL, BYE, καταγράφει όλα τα δεδομένα που αφορούν στο SIP, όπως για παράδειγμα την RTP κίνηση αλλά και όλα τα σχετικά γεγονότα που λαμβάνουν χώρα και ενεργεί αναλόγως, για παράδειγμα εγκαθιδρύοντας μια SIP σύνδεση που συμπεριλαμβάνει ένα RTP κανάλι. Μπορεί να υποστηρίξει την ύπαρξη πολλαπλών συνεδριών SIP και καναλιών RTP, προσφέρει τη δυνατότητα επιλογής στο χρήστη ονόματος χρήστη SIP και κωδικού πρόσβασης και καταγράφει όλα τα δεδομένα σε SQLite βάση δεδομένων. [59]

Περιγραφή λειτουργίας

Το Dionaea ανοίγει όλες τις παραπάνω δικτυακές θύρες και αναμένει για καινούριες συνδέσεις. Οποιοδήποτε αίτημα θεωρείται κακόβουλου περιεχομένου. Αφού εγκαθιδρυθεί μία σύνδεση, ο επιτιθέμενος στέλνει τελικά ένα φορτίο, (payload), το οποίο εξετάζεται από το Dionaea για να διαπιστωθεί εάν είναι κακόβουλου περιεχομένου και να ληφθεί αντίγραφο αυτού του φορτίου. Για την εξέταση του φορτίου, το Dionaea χρησιμοποιεί τη βιβλιοθήκη libemu. Η βιβλιοθήκη libemu ανιχνεύει τον κώδικα κελύφους, (shell code), που περιέχεται στο φορτίο ώστε να εξετάσει αν είναι κακόβουλου περιεχόμενου και εάν απαιτείται να το εκτελέσει. Η ανίχνευση του κώδικα κελύφους γίνεται μέσω ευριστικών μεθόδων GetPC. Πιο συγκεκριμένα, ο κώδικας κελύφους εκτελείται στο εικονικό μηχάνημα της βιβλιοθήκης libemu και καταγράφονται οι παράμετροι που απαιτούνται κατά την εκτέλεση καθώς και οι κλήσεις προς το API, (Application Programming Interface), απαραίτητα στοιχεία για να διαπιστωθούν οι προθέσεις των επιτιθέμενων και να ληφθούν οι απαραίτητες ενέργειες. Στην περίπτωση πολυεπίπεδης χρήσης κώδικα κελύφους, δηλαδή όταν ένας κώδικας κελύφους ενεργοποιεί έναν άλλον κ.ο.κ. απαιτείται η εκτέλεση και του δεύτερου κώδικα κελύφους.

Παρακάτω περιγράφεται η συμπεριφορά διάφορων τύπων φορτίων.

Shells – bind/connectback: Ο τύπος αυτός φορτίου προσφέρει ένα κέλυφος στον επιτιθέμενο, είτε ανοίγοντας μια δικτυακή θύρα και αναμένοντας τη σύνδεση του εισβολέα είτε εγκαθιδρύοντας μια σύνδεση από το Honeypot προς τον εισβολέα. Και στις δύο περιπτώσεις, το Dionaea προσφέρει στον εισβολέα ένα προσομοιωμένο αρχείο cmd.exe, του οποίου επεξεργάζεται την είσοδο. Συνήθως η είσοδος αυτή περιλαμβάνει εντολές λήψης αρχείων μέσω ftp ή tftp.

URLDownloadToFile: Ο τύπος αυτός φορτίου χρησιμοποιεί την ομώνυμη κλήση για να λάβει ένα αρχείο μέσω http και στη συνέχεια να το εκτελέσει.

Exec: Ο τύπος αυτός φορτίου χρησιμοποιεί την εντολή WinExec, η κλήση της οποίας και τα αποτελέσματά της εκτέλεσής της γίνονται αντικείμενα επεξεργασίας από το Dionaea, όπως ακριβώς και στην περίπτωση φορτίων τύπου bind/connectback.

Πολυεπίπεδος κώδικας κελύφους – Multi Stage Shellcodes: Όπως ήδη αναφέρθηκε, στην περίπτωση πολυεπίπεδου κώδικα κελύφους απαιτείται η εκτέλεση του δεύτερου επιπέδου στην εικονική μηχανή της βιβλιοθήκης libemu ώστε να διαπιστωθεί ο σκοπός του επιτιθέμενου.

Μετά την ανάλυση του κώδικα κελύφους, το Dionaea αποκτά πρόσβαση στην τοποθεσία από την οποία ο κώδικας κελύφους επιδιώκει να λάβει αρχείο κακόβουλου περιεχομένου και προσπαθεί να λάβει το εν λόγω αρχείο. Το πρωτόκολλο που χρησιμοποιείται για τη λήψη αρχείων μέσω ftp ή tftp έχει γραφτεί σε Python, ενώ για τη λήψη αρχείων μέσω http χρησιμοποιείται η βιβλιοθήκη libcurl.

Αφού ληφθεί ένα αρχείο κακόβουλου περιεχομένου, αποθηκεύεται τοπικά και μπορεί να αποσταλεί διαδικτυακά σε υπηρεσίες που πραγματοποιούν ανάλυση κακόβουλου λογισμικού, όπως VirusTotal, Anubis, Norman SandBox και CWSandbox.

Όλες οι επιθέσεις που επιχειρούνται προς το Honeypot καταγράφονται σε αρχείο κειμένου, ωστόσο η ανάγνωση και επεξεργασία αυτού του αρχείου κειμένου είναι ιδιαίτερα δύσκολη. Για το λόγο αυτό το Dionaea χρησιμοποιεί ένα εσωτερικό σύστημα επικοινωνίας που αποτελείται από περιστατικά, (incidents). Ένα περιστατικό έχει μία προέλευση που είναι μία συμβολοσειρά, ένα μονοπάτι και ιδιότητες οι οποίες μπορεί να είναι ακέραιοι, συμβολοσειρές ή και δείκτες προς μία σύνδεση. Τα περιστατικά γίνονται αντικείμενα επεξεργασίας από χειριστές περιστατικών, (ihandlers), οι οποίοι μπορούν να μετατρέψουν τα περιστατικά σε οποιαδήποτε μορφή επιθυμεί ο χρήστης ώστε να μπορεί να τα διαβάσει και να τα επεξεργαστεί. Ο logsql για παράδειγμα χειριστής περιστατικών καταγράφει όλα τα περιστατικά του Honeypot σε SQLite βάση δεδομένων. [59]

6.1.2 BackOfficer friendly

Το BackOfficer friendly, (BOF), αναπτύχθηκε από τον Marcus Ranum και τον Andrew Lambeth, μέλη της ομάδας που δημιούργησε το Network Flight Recorder, (NFR), ένα εμπορικά διαθέσιμο IDS.

Το BOF είναι ένα πρόγραμμα που εκτελείται στα περισσότερα λειτουργικά συστήματα βασισμένα σε Windows. Είναι χαμηλής αλληλεπίδρασης Honeypot και έχει παρόμοια λειτουργία με το Honeypot Specter, ωστόσο το BOF είναι πολύ πιο απλό.

Προσομοιώνει κάποιες βασικές υπηρεσίες, τις HTTP, FTP, TELNET, SMTP, POP3, IMAP, BackOrifice. Η βασική του λειτουργία έγκειται στην καταγραφή οποιασδήποτε προσπάθειας σύνδεσης στις ανοιχτές TCP αντίστοιχες θύρες. Έχει επίσης την επιλογή «faking replies», η οποία απαντά στις συνδέσεις του εισβολέα με χρήση συμβολοσειρών. Η επιλογή αυτή δεν είναι παραμετροποιήσιμη από τη μεριά του χρήστη. (Baumann and Plattner, 2002)

Με αυτό τον τρόπο μπορεί κανείς να καταγράψει HTTP, FTP, TELNET, SMTP, POP3, IMAP και BackOrifice επιθέσεις, όπως φαίνεται στην εικόνα 6-1.

Ουσιαστικά η αξία του BOF μπορεί να παρομοιαστεί με αυτή ενός οικιακού συναγερμού καθώς το BOF μπορεί να παρακολουθήσει μόνο ένα περιορισμένο αριθμό θυρών. Ωστόσο, οι θύρες αυτές εν γένει γίνονται αντικείμενο επίθεσης πολύ συχνά. [1]

Το πρόγραμμα ενδείκνυται περισσότερο για οικιακή χρήση και όχι δικτυακή. Διανέμεται ελεύθερα για προσωπική χρήση. Η πρώτη έκδοση πραγματοποιήθηκε το 1998 αλλά σήμερα έχει σταματήσει η διανομή του από το επίσημο site της NFR. Ο Lance Spitzner με την άδεια του Marcus Ranum διατηρεί αντίγραφο του λογισμικού στην προσωπική του ιστοσελίδα, από όπου και η ελεύθερη λήψη του λογισμικού μπορεί να πραγματοποιηθεί. [9]

Εικόνα 6-1: Στιγμιότυπο λειτουργίας BOF. [9]

6.1.3 Specter

Το Specter είναι ένα εμπορικά διαθέσιμο Honeypot που αναπτύχθηκε και διατίθεται στην αγορά από την NeoWorx. Η λειτουργία του είναι παρόμοια με του BOF, ωστόσο το Specter έχει αρκετά μεγαλύτερη λειτουργικότητα και μπορεί να προσομοιώσει περισσότερες υπηρεσίες.

Το Specter προσομοιώνει ένα ολόκληρο μηχάνημα, παρέχοντας έτσι έναν ενδιαφέροντα στόχο για επίδοξους εισβολείς. Παρέχει κοινές διαδικτυακές υπηρεσίες όπως SMTP και FTP, που φαίνονται απολύτως φυσιολογικές στους επιτιθέμενους, ωστόσο στην πραγματικότητα είναι παγίδες που καταγράφουν οποιαδήποτε κίνηση σύνδεσης και παράγουν ειδοποιήσεις. Το Specter έχει τη δυνατότητα να ελέγχει τους επιτιθέμενους καθώς αυτοί προσπαθούν να αποκτήσουν πρόσβαση στο μηχάνημα, χρησιμοποιώντας WHOIS εγγραφές και ανακτώντας τις προκαθορισμένες από τον ίδιο πληροφορίες χρήστη, τον αριθμό των θυρών από τις οποίες έχει διέλθει αλλά και το πλήρες μονοπάτι των διαδικτυακών κόμβων που ακολούθησε για να πραγματοποιήσει την επίθεσή του. Παράλληλα παράγει μαζικά δολώματα που μπορεί να είναι φωτογραφίες, MP3 αρχεία, ηλεκτρονικά μηνύματα, αρχεία κωδικών, αρχεία κειμένου. Τα μαζικά αυτά δολώματα είναι στην ουσία προγράμματα που αφήνουν κρυμμένα σημάδια στον υπολογιστή του επιτιθέμενου. Συγκεκριμένα, μπορεί να παράξει μέχρι και 250 διαφορετικά εκτελέσιμα προγράμματα, τα οποία μπορούν να αφήσουν μέχρι και 32 κρυφά σημάδια στον υπολογιστή του εισβολέα.

Αυτοματοποιημένες διαδικτυακές αναβαθμίσεις επιτρέπουν στο Honeypot να αναβαθμίζεται ομοίως συνεχώς χωρίς τη βοήθεια του χρήστη. Η τελευταία έκδοση του Specter, (Specter 8.0), χρησιμοποιεί μια καινούρια λειτουργική μονάδα ονόματι WATCHER, η οποία επιτρέπει στο Specter να καταγράψει κάθε ICMP πακέτο, TCP σύνδεση και UDP δεδομενόγραμμα σε κάθε κόμβο. [10]

Το Specter περιέχει εκατοντάδες προεπιλεγμένα ονόματα χρήστη και συνδέσεις ειδικά διαμορφωμένες για αγγλόφωνα περιβάλλοντα. Ωστόσο, δίνει τη δυνατότητα και για προσαρμογή των ονομάτων χρήστη και των συνδέσεων σε οποιαδήποτε γλώσσα. Παράλληλα, δίνει τη δυνατότητα και για προσαρμογή των δολωμάτων στο λογισμικό του χρήστη. Για παράδειγμα, αν έχει προσομοιωθεί ένας HTTP εξυπηρετητής, μπορούν να παραχθούν για δολώματα διαδικτυακά αρχεία όμοια με αυτά του εξυπηρετητή.

Ένα Honeypot γενικά δεν αντιμετωπίζει βαριά κίνηση, ωστόσο μπορεί να είναι στόχος DOS, DDOS επιθέσεων και επιθέσεων πλημμύρας. Το Specter χρησιμοποιεί διάφορους μηχανισμούς για να επιλύσει τέτοιες επιθέσεις, που είναι αυτοματοποιημένοι αλλά μπορούν και να ενεργοποιηθούν, απενεργοποιηθούν, παραμετροποιηθούν από το χρήστη.

Το Specter εκτελείται σε Windows XP. Σχεδιασμένο να εκτελείται και σε παλαιότερα μηχανήματα – ένα Pentium 90 με 32 Mb μνήμη RAM είναι η ελάχιστη απαίτηση – το Specter θέτει την κάρτα δικτύου του υπολογιστή που θα δεχτεί τις επιθέσεις σε μία διακριτή κατάσταση λειτουργίας και καταγράφει όλα τα πακέτα που θα διαπεράσουν τη σύνδεση. Αντί να συλλέγει ροές πακέτων και να τις θέτει προς σύγκριση σε μία βάση υπογραφών, το Specter προσομοιώνει έναν αριθμό κοινών λειτουργικών συστημάτων και υπηρεσιών, οδηγώντας έτσι έναν επίδοξο εισβολέα στην πεποίθηση ότι επιτίθεται σε ένα πραγματικό σύστημα. Καταγράφει λοιπόν όλες τις προσπάθειες σύνδεσης και μπορεί ακόμα να ξεκινήσει και την καταγραφή των κόμβων από τους οποίους έχει περάσει ο εισβολέας από τη διαδικτυακή διεύθυνση από την οποία ξεκίνησε μέχρι τον τελικό του προορισμό. [9]

Το Specter αποτελείται από δύο μέρη, τη μηχανή και τον έλεγχο. Η μηχανή διαχειρίζεται τις δικτυακές συνδέσεις και το packet sniffing, (παρακολούθηση των πακέτων ενός δικτύου), ενώ ο έλεγχος παρέχει ένα απλό γραφικό περιβάλλον που εκτελείται στο προσκήνιο για παραμετροποίηση. Η ολική παραμετροποίηση υλοποιείται σε μία μόνο οθόνη. Το Specter μπορεί να προσομοιώσει οποιοδήποτε από τα λειτουργικά συστήματα Windows NT, 95, 98, 2003, XP, MacOS, MacOS X, Linux, SunOS/ Solaris, DigitalUNIX, NeXTStep, Irix και Unisys UNIX, FreeBSD, Tru64 και παρέχει ψευδή αρχεία κωδικών στον εισβολέα σε format αντίστοιχο του λειτουργικού στο οποίο εκτελείται. Τα αρχεία κωδικών που παράγονται μπορούν να είναι Unix passwd, συμπιεσμένο δυαδικό αρχείο των Windows και μη συμπιεσμένο δυαδικό αρχείο των Windows. Το προσομοιωθέν λειτουργικό μπορεί να λάβει τους εξής χαρακτηρισμούς: open (κακώς παραμετροποιημένο), secure, failing (μηχάνημα με ευπάθειες σε υλικό και λογισμικό), strange (απρόβλεπτο) και aggressive (πρώτα συλλέγει πληροφορίες από τον εισβολέα και κατόπιν αποκαλύπτεται και διακόπτει τη σύνδεση με τον εισβολέα). Το Specter μπορεί να προσομοιώσει τις εξής δικτυακές υπηρεσίες: SMTP, FTP, TELNET, FINGER, POP3, IMAP4, HTTP, SSH, DNS, SUN-RPC, NETBUS, SUB7, BO2K, GENERIC TRAP. Οι πληροφορίες που συλλέγονται κατά την καταγραφή των συνδέσεων είναι το όνομα του τηλεπικοινωνιακού παρόχου του εισβολέα, η ακριβή ώρα της επίθεσης, ο τύπος της υπηρεσίας και η κατάσταση της μηχανής κατά τη στιγμή της σύνδεσης. Με τη γενική προσδιορισμένη από το χρήστη παγίδα, (GENERIC TRAP), μπορεί ο διαχειριστής να καθορίσει τη θύρα παρακολούθησης. Ένα μικρό παράθυρο στην οθόνη του γραφικού περιβάλλοντος παρέχει σε πραγματικό χρόνο συνοπτικές λεπτομέρειες των ειδοποιήσεων που παράγονται και πιο λεπτομερείς πληροφορίες μπορεί να σταλθούν ηλεκτρονικά στο διαχειριστή ή να καταγραφούν στο δίσκο. Όταν το Specter επομένως εντοπίσει μία ύποπτη σύνδεση, μπορεί να παραμετροποιηθεί έτσι ώστε να ελέγξει τον εισβολέα αλλά και να καταγράψει την αλληλουχία των κόμβων από τους οποίους πέρασε ο εισβολέας για να φτάσει στον προορισμό του. [10]

Προσομοιώνοντας μόνο υπηρεσίες και λειτουργικά συστήματα, το Specter παραμένει ένα χαμηλής αλληλεπίδρασης Honeypot. Υπάρχει μηδενικός κίνδυνος ο εισβολέας να καταλάβει το μηχάνημα και να το χρησιμοποιήσει σαν πλατφόρμα για καινούριες επιθέσεις.

Η αξία του Specter έγκειται στην ανίχνευση. Το Specter μπορεί πολύ εύκολα να εντοπίσει ποιος κάνει τι. Ως Honeypot μειώνει τα ποσοστά των ψευδώς αρνητικών και ψευδώς θετικών απλοποιώντας τη διαδικασία εντοπισμού. [9]

6.1.4 Honeyd

Το Honeyd δημιουργήθηκε από τον Niels Provos το 2003 και είναι ένα ιδιαίτερα δυνατό ανοιχτού κώδικα Honeypot που διανέμεται ελεύθερα. Η τελευταία του έκδοση ανακοινώθηκε το Μάιο του 2007.

Λειτουργεί σαν ένα daemon πρόγραμμα, ένα πρόγραμμα που εκτελείται συνεχώς στο παρασκήνιο δηλαδή, που δημιουργεί εικονικούς υπολογιστικούς κόμβους σε ένα δίκτυο. Οι κόμβοι αυτοί παραμετροποιούνται για την εκτέλεση αυθαίρετων εργασιών, ωστόσο η συμπεριφορά τους προσαρμόζεται ώστε να φαίνονται ότι προσομοιώνουν συγκεκριμένα λειτουργικά συστήματα. Το Honeyd επιτρέπει σε έναν κόμβο να καταλάβει πολλαπλές διευθύνσεις, μέχρι και 65536 στο πλήθος, σε ένα τοπικό δίκτυο για διαδικτυακή προσομοίωση και βελτιώνει τη διαδικτυακή ασφάλεια παρέχοντας μηχανισμούς ανίχνευσης και αξιολόγησης απειλών. Έτσι αποθαρρύνει τους εχθρούς κρύβοντας τα πραγματικά συστήματα πίσω από τα εικονικά. [11]

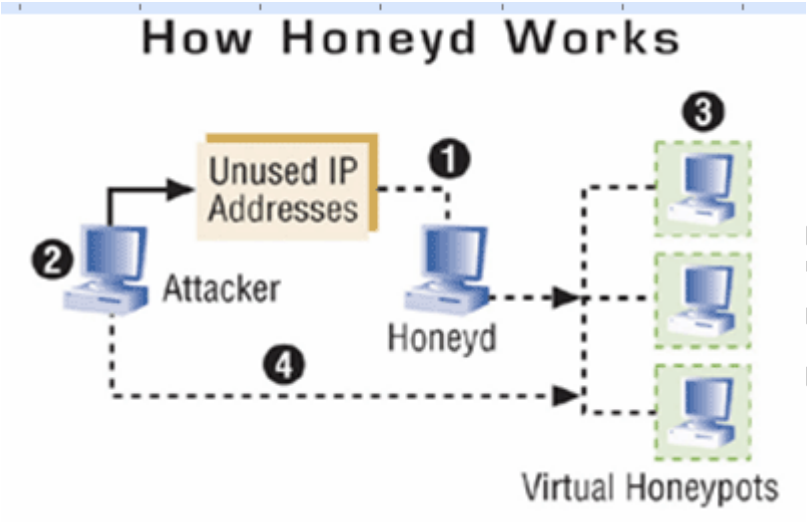
Το Honeyd δίνει τη δυνατότητα διαχείρισης των εικονικών μηχανημάτων στο χρήστη. Ο χρήστης μπορεί να εκτελέσει την εντολή ping στα εικονικά μηχανήματα ώστε να δει τη διεύθυνση ip τους. Ακόμα, οποιοσδήποτε τύπος υπηρεσίας μπορεί να προσομοιωθεί στα εικονικά μηχανήματα μόνο με τη χρήση ενός αρχείου παραμετροποίησης ενώ είναι δυνατή και η ανακατεύθυνση μιας υπηρεσίας από ένα μηχάνημα σε ένα άλλο. Με τη χρήση του nmap διαπιστώνονται οι ανοιχτές TCP θύρες ώστε εν συνεχεία να εκτιμηθούν οι θύρες που θα χρησιμοποιηθούν για FIN-scans ή εν γένει για αποστολή και αναμετάδοση TCP πακέτων.

Το Honeyd χρησιμοποιείται για να δημιουργήσει ένα εικονικό δίκτυο honeynet ή για γενική παρακολούθηση ενός δικτύου. Υποστηρίζει τη δημιουργία μίας εικονικής τοπολογίας δικτύου που συμπεριλαμβάνει αποκλειστικές διαδρομές, (dedicated routes), και δρομολογητές. Οι διαδρομές μπορεί να είναι επιφορτισμένες με καθυστέρηση και απώλεια πακέτων ώστε να κάνουν την τοπολογία να φαίνεται πιο ρεαλιστική. Με τη χρήση GRE tunneling επιτρέπεται η δημιουργία κατανεμημένων υποδομών στο Honeyd ώστε να δημιουργηθούν ευρύτερα μεγάλες υποδομές δικτύου. Παράλληλα επιτρέπεται και στα εικονικά μηχανήματα να μπορούν να εξαπλωθούν σε ξεχωριστές διευθύνσεις χώρου, καθώς το GRE tunneling βασίζεται σε διευθύνσεις προέλευσης.

Λόγω της αλληλεπίδρασης του honeyd με δυνητικά κακόβουλους εχθρούς, ενδείκνυται η χρήση ενός sandbox, και ειδικότερα του Systrace, μιας και το Systrace αποτρέπει έναν εχθρό από το να εκμεταλλευτεί bugs των Honeyd scripts. [1]

Το Honeyd μπορεί να δημιουργήσει εικονικές υπηρεσίας εκτελώντας Unix εφαρμογές σαν υποσυστήματα που εκτελούνται στον εικονικό χώρο διευθύνσεων ενός παραμετροποιημένου Honeypot. Αυτό επιτρέπει σε οποιαδήποτε δικτυακή εφαρμογή να συνδέσει δυναμικά θύρες, να δημιουργήσει TCP και UDP συνδέσεις χρησιμοποιώντας μία εικονική IP διεύθυνση. Τα υποσυστήματα διαμορφώνονται εικονικά με την υποκλοπή δικτυακών αιτήσεων και την ανακατεύθυνσή τους στο Honeyd. Κάθε πρότυπο ρύθμισης παραμέτρων μπορεί να περιλαμβάνει υποσυστήματα που εκκινούν σαν διακριτές διαδικασίες όταν το πρότυπο είναι δεμένο σε μία εικονική IP διεύθυνση. Αυτό οδηγεί τα Honeypots στη δημιουργία σποραδικής κίνησης που εκτελείται στο παρασκήνιο, όπως αίτηση ιστοσελίδων, ανάγνωση email κλπ.

Η λειτουργία του Honeyd παρουσιάζεται στην εικόνα 6-2. Παρακολουθούνται αχρησιμοποίητες IP διευθύνσεις (1). Όταν ένας εισβολέας (2) ανιχνεύσει μία αχρησιμοποίητη IP διεύθυνση, το Honeyd εντοπίζει τον ανιχνευτή, ανακαταλαμβάνει αυτή την IP μέσω ARP spoofing, και δημιουργεί εικονικά Honeypots (3) για τον εισβολέα ώστε να αλληλεπιδράσει (το Honeyd μπορεί να δημιουργήσει πολλαπλά εικονικά Honeypots για να ξεγελάσουν τον εισβολέα για όλες τις αχρησιμοποίητες διευθύνσεις). Τελικά, ο εισβολέας νομίζει ότι αλληλεπιδρά με ένα πραγματικό σύστημα στο οποίο κατέφερε και απέκτησε πρόσβαση (4). Επιπλέον, το Honeyd ενημερώνει αυτόματα τον κατάλογο των αχρησιμοποίητων IPs ως συστήματα τα οποία προστίθενται ή αφαιρούνται από το δίκτυο. [11]



Εικόνα 6-2: Λειτουργία Honeyd. [1]

Συμπερασματικά, το Honeyd χρησιμοποιείται κυρίως για την ανίχνευση επιθέσεων. Λειτουργεί παρακολουθώντας αχρησιμοποίητες διευθύνσεις IP. Κάθε φορά που ένας εισβολέας προσπαθεί να εξετάσει ή να επιτεθεί σε ένα ανύπαρκτο σύστημα, το Honeyd, μέσω spoofing ARP, καταλαμβάνει τη διεύθυνση IP του θύματος και στη συνέχεια αλληλεπιδρά με τον εισβολέα μέσω μίμησης. Οι υπηρεσίες μίμησης είναι scripts που αντιδρούν σε προκαθορισμένες ενέργειες. Για παράδειγμα, ένα σενάριο μπορεί να αναπτυχθεί για να συμπεριφέρεται σαν μία Telnet υπηρεσία για ένα router Cisco, στη Cisco IOS διεπαφή. Οι υπηρεσίες μίμησης του Honeyd είναι επίσης ανοιχτού λογισμικού. [1]

Τα scripts μπορούν να γραφτούν σε σχεδόν οποιαδήποτε γλώσσα, όπως shell ή Perl. Μόλις γίνει η σύνδεση, ο εισβολέας πιστεύει ότι αλληλεπιδρά με ένα πραγματικό σύστημα. Το Honeyd όχι μόνο μπορεί δυναμικά να αλληλεπιδρά με εισβολείς, αλλά μπορεί να ανιχνεύσει δραστηριότητα σε οποιαδήποτε θύρα. Τα περισσότερα Honeypots χαμηλής αλληλεπίδρασης περιορίζονται στην ανίχνευση επιθέσεων μόνο για τις θύρες στις οποίες έχουν υπηρεσίες μίμησης. Το Honeyd είναι διαφορετικό, καθώς εντοπίζει και καταγράφει συνδέσεις για κάθε port, ανεξάρτητα από το αν υπάρχει εκεί υπηρεσία. Η υπόθεση ότι υπάρχουν συστήματα που στην πραγματικότητα δεν υπάρχουν και η δυνατότητα ανίχνευσης δραστηριότητας σε οποιαδήποτε θύρα, δίνει στο Honeyd μεγάλη αξία ως εργαλείο που μπορεί να ανιχνεύσει και να εξακριβώσει μη εξουσιοδοτημένη πρόσβαση. [11]

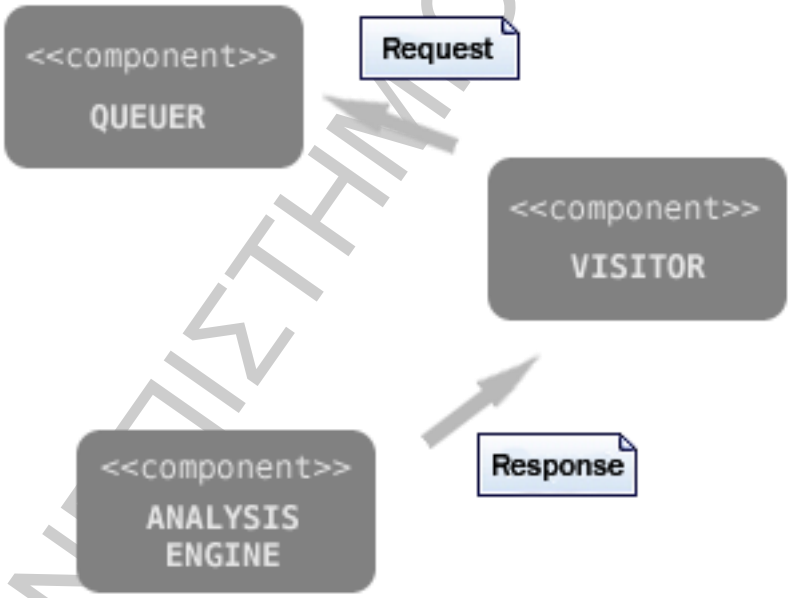
6.1.5 HoneyC

Το HoneyC είναι ένα χαμηλής αλληλεπίδρασης Client Honeypot, που αναπτύχθηκε στο Victoria University of Wellington από τον Christian Seifert το 2006 και επιτρέπει την αναγνώριση κακόβουλων εξυπηρετητών στο Διαδίκτυο. Αντί να χρησιμοποιήσει ένα πλήρες λειτουργικό σύστημα και έναν πλήρως λειτουργικό client για αυτή τη λειτουργία, (υψηλής αλληλεπίδρασης Honeypots υλοποιούν αυτή τη λειτουργία, όπως HoneyMonkey και HoneyClient), το HoneyC χρησιμοποιεί προσομοιωμένους clients που μπορούν να αναζητήσουν την απάντηση που απαιτείται από έναν εξυπηρετητή για ανάλυση κακόβουλου περιεχομένου. Το HoneyC έχει δυνατότητες επέκτασης με πολλούς τρόπους καθώς μπορεί να χρησιμοποιήσει διάφορους clients που λειτουργούν ως επισκέπτες και χρησιμοποιεί και συστήματα αναζήτησης και ανάλυσης αλγορίθμων.

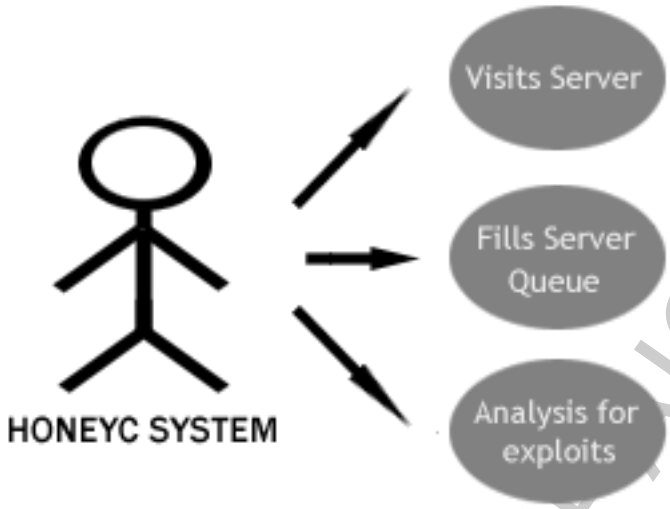
Η αρχική έκδοση HoneyC επικεντρωνόταν στη βασισμένη σε snort υπογραφές αναζήτηση για κακόβουλους διαδικτυακούς διακομιστές. Δεν περιείχε καμία υπογραφή κακόβουλου λογισμικού αλλά η επόμενη έκδοση αναμένεται να έχει. Η τελευταία έκδοση ανακοινώθηκε το 2008, (έκδοση 1.3.0). [12]

Ως Client Honeypot ανιχνεύει το δίκτυο ώστε να εντοπίσει μέσω της απόκρισης που στέλνει την ύπαρξη εξυπηρετητών που εκμεταλλεύονται τον client. Το HoneyC είναι ένα χαμηλής αλληλεπίδρασης Honeypot, καθώς χρησιμοποιεί clients που έχουν προσομοιωθεί, (π.χ. χρησιμοποιείται η εντολή wget για να προσομοιώσει τον φυλλομετρητή), αλλά και έναν μηχανισμό ανάλυσης που εκτός από την παρακολούθηση της κατάστασης του λειτουργικού συστήματος επιπρόσθετα χρησιμοποιεί και ειδικούς αλγόριθμους.

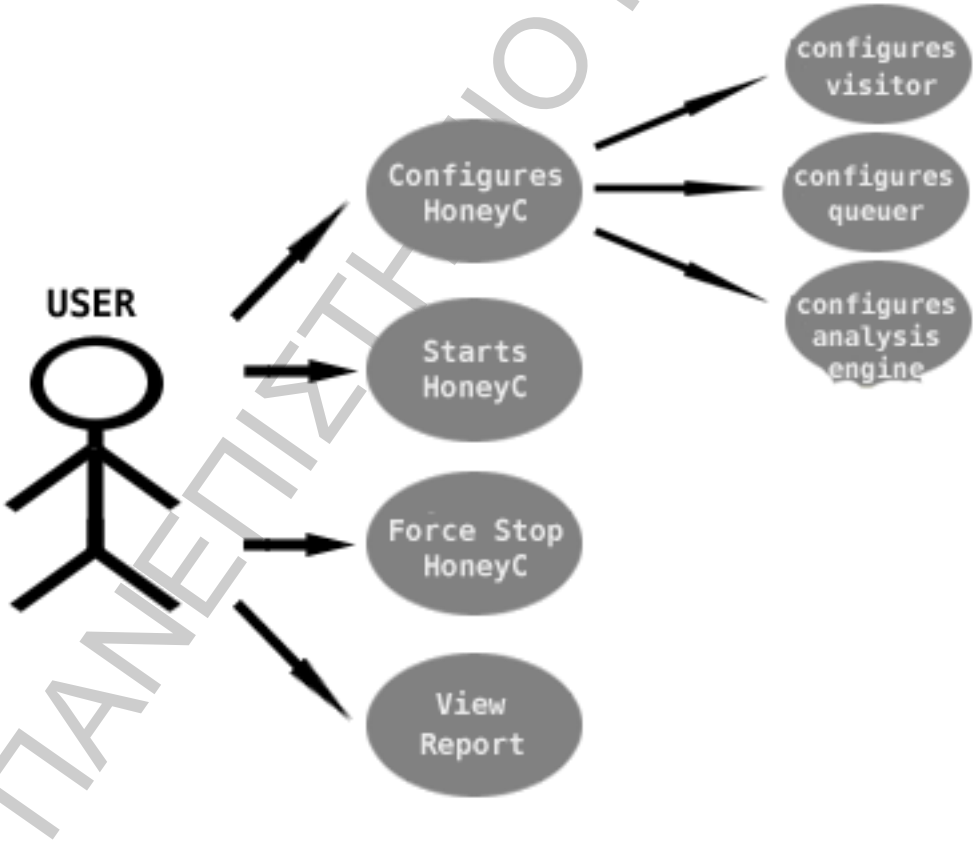
Το HoneyC αποτελείται από τρία συστατικά, τον Visitor, τον Queuer και την Analysis Engine. Ο Visitor είναι το συστατικό που είναι υπεύθυνο για την επικοινωνία με τον εξυπηρετητή. Κάνει ένα αίτημα προς τον εξυπηρετητή και μετά λαμβάνει και επεξεργάζεται την απόκρισή του. Ο Queuer αναλαμβάνει τη δημιουργία μίας ουράς από εξυπηρετητές για τον Visitor ώστε να έχει εξυπηρετητές να αλληλεπιδράσει. Ο Queuer χρησιμοποιεί διάφορους αλγόριθμους για να δημιουργήσει αυτή την ουρά, όπως ανίχνευση στο Διαδίκτυο, ενσωμάτωση μέσω μηχανισμού αναζήτησης κ.α. Η Analysis Engine αναλαμβάνει να εκτιμήσει αν κάποιος μηχανισμός ασφάλειας παραβιάστηκε κατά την αλληλεπίδραση του Visitor με τον εξυπηρετητή. Καθένα από αυτά τα στοιχεία επιτρέπει τη χρήση πρόσθετων λειτουργικών μονάδων για την κάλυψη αναγκών που ενδεχομένως προκύψουν. Οι μονάδες αυτές συνδέονται μέσω εντολών ανακατεύθυνσης, (pipes), με τους μηχανισμούς αίτησης κι απόκρισης ώστε η λειτουργία τους να είναι ανεξάρτητη. Για παράδειγμα, θα μπορούσε να δημιουργήσει κάποιος έναν Queuer που παράγει αντικείμενα αίτησης μέσω ενσωμάτωσης σε ένα συγκεκριμένο περιβάλλον διεπαφής μηχανής αναζήτησης, που έχει γραφτεί σε Ruby ή ομοίως θα μπορούσε κάποιος να γράψει σε C έναν Queuer που ανιχνεύει ένα δίκτυο. Η εικόνα 6-3 παρουσιάζει τα τρία κύρια συστατικά του HoneyC και η εικόνα 6-4 παρουσιάζει τις βασικές λειτουργίες που επιτελεί το HoneyC.



Εικόνα 6-3: Διάγραμμα συστατικών HoneyC. [12]



Εικόνα 6-4: Βασική λειτουργία HoneyC. [12]



Εικόνα 6-5: Αλληλεπίδραση χρήστη με το HoneyC. [12]

Η εικόνα 6-5 δείχνει την αλληλεπίδραση ενός τελικού χρήστη με το HoneyC. Από την αρχή μέχρι την παύση της λειτουργίας του HoneyC, η πλήρης αλληλεπίδραση διαφαίνεται στην εικόνα. Το HoneyC διακόπτει από μόνο του τη λειτουργία του όταν η ουρά είναι γεμάτη.

Ο χρήστης μπορεί να αλλάξει και να προσαρμόσει τα συστατικά στοιχεία του HoneyC ώστε να πραγματοποιηθεί η ανίχνευση όπως ακριβώς αυτός επιθυμεί. Αφού έχει ολοκληρωθεί η ανίχνευση, ο χρήστης μπορεί να δει τα αποτελέσματα της ανίχνευσης, δηλαδή ποιους εξυπηρετητές επισκέφτηκε το HoneyC και ποιοι εξυπηρετητές απομόνωσαν μία κακόβουλη απάντηση. [12]

6.1.6 Monkey – Spider

Το Monkey – Spider είναι ένα χαμηλής αλληλεπίδρασης Client Honeypot που ανιχνεύει ιστοσελίδες για να υποδείξει ενδεχόμενες απειλές σε διαδικτυακούς clients. Ξεκίνησε ως μεταπτυχιακή διατριβή από τον Ali Ikinci στο πανεπιστήμιο του Mannheim το Μάιο του 2007 και έκτοτε διανέμεται ελεύθερα υπό την GPLv3 άδεια. Η τελευταία έκδοση 2.0 εκδόθηκε το Μάρτιο του 2009.

Το Monkey – Spider ανιχνεύει οποιοδήποτε είδος κακόβουλου λογισμικού, όπως virus, trojan, worm, spyware, adware, phishing και hoax τα οποία γίνονται αντιληπτά μέσω προκαθορισμένων υπογραφών κακόβουλου λογισμικού, καθώς και μέσω εμπορικών anti – virus και anti – spyware ανιχνευτών. Ωστόσο μπορεί να ανιχνεύσει και άγνωστες απειλές μέσω αυτοματοποιημένων τεχνικών προσομοίωσης ανάλυσης κακόβουλου λογισμικού.

Ενδείκνυται περισσότερο για εταιρείες, οργανισμούς και ερευνητές ασφαλείας που επιθυμούν αυτοματοποιημένα να ανακαλύπτουν απειλές στις εταιρικές ιστοσελίδες τους. Ομοίως ενδείκνυται και για κοινοτικές ιστοσελίδες ή forums όπου ο χρήστης μπορεί να ανεβάσει περιεχόμενο.

Δε διαθέτει διεπαφή χρήστη, παρά μόνο διασύνδεση μέσω γραμμής εντολών. Αρχικά υπήρχε μία διεπαφή χρήστη αλλά δεν ήταν λειτουργική οπότε εν συνεχεία εγκαταλήφθηκε αλλά αναμένεται σε επόμενες εκδόσεις. Δεν υποστηρίζει αναζήτηση μέσω Google. Αρχικά υπήρχε μία διασύνδεση με το προγραμματιστικό περιβάλλον SOAP αναζήτησης της Google, (Google SOAP Search API), αλλά εξαιτίας της μη έκδοσης καινούριων API κλειδιών για το SOAP Search API, στη συνέχεια εγκαταλήφθηκε. Δεν υποστηρίζει ανάλυση επισυναπτόμενων αρχείων. Για την αυθεντικοποίηση των αιτήσεων απαιτεί έγκυρα αναγνωριστικά που είναι μοναδικά για κάθε χρήστη Yahoo ή Microsoft Live και ανακτώνται από τα επίσημα sites της Yahoo και της Microsoft.

Για την εγκατάσταση του Monkey – Spider απαιτείται λειτουργικό σύστημα Linux, τα εργαλεία sed, awk, wget, grep, unzip και Python 2.5 τουλάχιστον καθώς και τα παρακάτω πακέτα:

- διαδικτυακός ανιχνευτής Heritrix
- PostgreSQL βάση δεδομένων που απαιτεί αυθεντικοποίηση με κωδικό για διασύνδεση πάνω από το δίκτυο
- ClamAV anti – virus ανιχνευτή
- την Python διεπαφή PyGreSql για την PostgreSQL βάση δεδομένων
- SOAPpy για τη σύνδεση στη μηχανή αναζήτησης Microsoft Live και
- pYsearch για τη σύνδεση στη μηχανή αναζήτησης Yahoo.

Σε υλικό, το ελάχιστο που απαιτείται είναι ένας υπολογιστής Pentium με 128 MB.

Το Monkey – Spider αποτελείται από διάφορα scripts, τα οποία μπορούν να εκτελεστούν ανεξάρτητα ανάλογα με το ερευνητικό ενδιαφέρον του χρήστη. Η λειτουργία του αποτελείται από τα εξής βήματα:

- Seed generation, όπου καθορίζεται το μέρος του Διαδικτύου που θα ανιχνευτεί,
- Crawl setup και crawling, όπου καθορίζεται ο τρόπος ανίχνευσης και
- Scanning, όπου καθορίζεται ο μηχανισμός ελέγχου του ενδεχόμενου κακόβουλου λογισμικού από τον ClamAV anti – virus ανιχνευτή που έχει ήδη ανιχνευτεί κατά τα δύο προηγούμενα βήματα.

Αναλυτικότερα, οι λειτουργίες που υλοποιούνται κατά το seeding είναι οι εξής:

Ο Heritrix ανιχνευτής ξεκινά την ανίχνευση έχοντας ένα άδειο αρχείο κειμένου, ονόματι seeds.txt. Υπάρχουν τέσσερις διαφορετικές μέθοδοι για την παραγωγή εναρκτήριων seeds για τον ανιχνευτή.

- Χειροκίνητη προσθήκη URL διευθύνσεων. Ενδείκνυται σε περιπτώσεις ανάλυσης προκαθορισμένου σετ διευθύνσεων.
- Διαδικτυακή αναζήτηση. Το Monkey – Spider παρέχει δύο scripts, το ms-seeder-websearch-yahoo και το ms-seeder-websearch-livesearch, τα οποία κάνουν χρήση web services της Yahoo και της Windows Live Search για τη συγκέντρωση URL λιστών σε πραγματικό χρόνο, σε περίπτωση που ο χρήστης παρέχει ένα έγκυρο αναγνωριστικό για τα web services.
- Blacklist seeding. Το Monkey – Spider παρέχει ένα script ονόματι ms-seeder-blacklist, το οποίο αυτόματα συγκεντρώνει μία λίστα από γνωστές διευθύνσεις URL που θεωρούνται ύποπτες για περιεχόμενο κακόβουλου λογισμικού. Οι διευθύνσεις αυτές αποτελούν URLs που σχετίζονται με διαφημίσεις.
- Mail seeding. Το Monkey – Spider παρέχει και ένα script που ονομάζεται ms-seeder-mail-pop3, το οποίο εξετάζει αν ένας email λογαριασμός περιέχει διευθύνσεις, ώστε να εξεταστούν οι διευθύνσεις αυτές.

Αναλυτικά, κατά το crawling ο Heritrix αφού παραμετροποιηθεί, παράγει ARC αρχεία, τα οποία ακολουθούν ένα συγκεκριμένο πρότυπο και περιέχουν το λογισμικό που θεωρείται κακόβουλο.

Αναλυτικά, κατά το scanning, εκτελείται το script ms-processfolder, το οποίο εξετάζει το περιεχόμενο όλων των ARC αρχείων, κάνοντας χρήση του ClamAV anti – virus. Τα τελικά αποτελέσματα αποθηκεύονται στη βάση δεδομένων. [32]

6.1.7 PhoneyC

Το PhoneyC είναι ένα εικονικό Client Honeypot, δηλαδή δεν είναι πραγματική εφαρμογή που μπορεί να παραβιαστεί από εισβολείς και στη συνέχεια να γίνει αντικείμενο παρακολούθησης για ανάλυση από τους εισβολείς, αλλά client που έχει προσομοιωθεί. Αναπτύχθηκε από τον Jose Nazario το 2009. Έχει γραφτεί σε Python. Στόχος του είναι να διακρίνει ιστοσελίδες με κακόβουλο περιεχόμενο και κάνοντας χρήση δυναμικής ανάλυσης να απομακρύνει τα επικίνδυνα στοιχεία. Διατρέχει ιστοσελίδες ψάχνοντας για εκείνες που αντιδρούν με το πρόγραμμα περιήγησης. Επιπλέον μπορεί να προσομοιώσει διάφορες ευπαθείς υπηρεσίες ώστε να εντοπίσει το φορέα της επίθεσης. Αποτελεί ουσιαστικά ένα σπονδυλωτό πλαίσιο που επιτρέπει τη μελέτη ιστοσελίδων κακόβουλου περιεχομένου και αντιλαμβάνεται τις σύγχρονες αδυναμίες και τεχνικές του εισβολέα.

Κάποια βασικά χαρακτηριστικά του PhoneyC είναι τα εξής:

- Αντιλαμβάνεται τη χρήση HTML ετικετών για απομακρυσμένες συνδέσεις.
- Αντιλαμβάνεται scripting γλώσσες, όπως javascript και visual basic.
- Υποστηρίζει ActiveX λειτουργίες για ανίχνευση ευπαθειών.
- Ανιχνεύει και αναλύει κώδικα κελύφους.
- Ανιχνεύει σωρούς.
- Υποστηρίζει Anti – virus ανίχνευση μέσω ClamAV.

Το PhoneyC αποτελεί λογισμικό ανοιχτού κώδικα που διανέμεται ελεύθερα υπό την άδεια της GNU. [13]

6.1.8 SpyBye

Το SpyBye είναι ένα Honeypot χαμηλής αλληλεπίδρασης που αναπτύχθηκε από τον Niels Provos. Είναι ένα εργαλείο που βοηθά τους διαχειριστές ιστοσελίδων να αποφασίσουν αν οι ιστοσελίδες τους φιλοξενούν προγράμματα ή εν γένει λειτουργίες που θα μπορούσαν να γίνουν αντικείμενο εκμετάλλευσης και με τη σειρά τους να μολύνουν με κακόβουλο λογισμικό τους

επισκέπτες αυτών των σελίδων. Λειτουργεί σαν έναν proxy HTTP διακομιστή μεσολάβησης και παρεμβαίνει σε όλα τα αιτήματα του προγράμματος περιήγησης. Χρησιμοποιεί μια σειρά από απλούς κανόνες για να διαπιστώσει αν ενσωματωμένες συνδέσεις σε μία ιστοσελίδα είναι ακίνδυνες ή όχι. Η ολοένα και αυξανόμενη τάση παραβίασης web τοποθεσιών οδήγησε στη δημιουργία του SpyBye.

Η λειτουργία του SpyBye συνοψίζεται ως εξής: Ως διακομιστής μεσολάβησης μπορεί να δει όλες τις λήψεις λογισμικού, (fetches), που κάνει το πρόγραμμα περιήγησης. Εφαρμόζει απλούς κανόνες σε κάθε διεύθυνση που λαμβάνει και φορτώνει τα απαραίτητα στοιχεία κάθε φορά και την αντίστοιχη σελίδα. Οι κανόνες αυτοί επιτρέπουν την κατηγοριοποίηση της διεύθυνσης σε τρεις κατηγορίες, ακίνδυνη, άγνωστη και επικίνδυνη. Αν και υπάρχει μεγάλο περιθώριο λάθους, οι κατηγορίες επιτρέπουν σε ένα διαχειριστή ιστοσελίδων να εξετάσει τις URL διευθύνσεις και να αποφανθεί αν ανήκουν όντως στην κατηγορία στην οποία αρχικά εντάχθηκαν από το SpyBye. Συνήθως η ένταξη μίας σελίδας σε μη αναμενόμενη κατηγορία σημαίνει την παραβίαση της σελίδας.

Το SpyBye δεν προστατεύει από το να γίνει μία σελίδα αντικείμενο εκμετάλλευσης. Προσπαθεί να πάρει έγκαιρες προφυλάξεις για να αποφύγει τη μόλυνση κατά τη διάρκεια χρήσης της. Στην ιδανική περίπτωση εκτελείται το πρόγραμμα περιήγησης σε εικονικό μηχάνημα και όταν διαπιστωθεί επικίνδυνο πρόγραμμα, γίνεται επαναφορά του συστήματος σε καθαρό στιγμιότυπο. [14]

Το SpyBye αποτελεί λογισμικό ανοιχτού κώδικα που διανέμεται ελεύθερα. Το SpyBye 0.1 διανέμεται ελεύθερα υπό την άδεια της BSD και οι δύο επόμενες εκδόσεις διανέμονται ελεύθερα υπό την άδεια της GPL. Και οι τρεις εκδόσεις ανακοινώθηκαν το 2007.

6.1.9 LaBrea

Το LaBrea ανακοινώθηκε τον Οκτώβριο του 2003. Καταλαμβάνει αχρησιμοποίητες IP διευθύνσεις και δημιουργεί εικονικούς εξυπηρετητές που προσελκύουν worms, hackers και λοιπές απειλές και συνδέονται σε αυτά. Το πρόγραμμα απαντάει στις προσπάθειες σύνδεσης με τέτοιο τρόπο ώστε το μηχάνημα στο άλλο άκρο της σύνδεσης να «κολλάει» και ορισμένες μάλιστα φορές για μεγάλο χρονικό διάστημα. Για το λόγο αυτό το LaBrea έχει ονομαστεί και sticky Honeypot.

Αναλυτικότερα, το LaBrea λειτουργεί παρακολουθώντας ARP αιτήσεις και απαντήσεις. Όταν διαπιστωθούν διαδοχικές ARP αιτήσεις που απέχουν μεταξύ τους μερικά δευτερόλεπτα, χωρίς καμία ARP απάντηση, τότε το LaBrea θεωρεί την εν λόγω διεύθυνση IP μη απασχολημένη. Τότε δημιουργεί μία ARP απάντηση με μία ψεύτικη MAC διεύθυνση, και ενεργοποιεί έτσι πάλι τον αποστολέα. Το LaBrea παρακολουθεί επίσης την TCP κίνηση για τη διεύθυνση MAC. Όταν λάβει ένα εισερχόμενο πακέτο TCP SYN, απαντάει μέσω του εικονικού εξυπηρετητή που έχει δημιουργηθεί με ένα TCP/ACK πακέτο που πιστοποιεί την προσπάθεια σύνδεσης.

Το LaBrea έχει δοκιμαστεί σε FreeBSD, Linux, Solaris και Windows 98 και χρησιμοποιεί autoconf/automake. [33]

6.1.10 Nepenthes

Το Nepenthes είναι ένα Honeypot χαμηλής αλληλεπίδρασης το οποίο προσομοιώνει ευπαθή σημεία πολλών υπηρεσιών με σκοπό να καταγράψει τις δραστηριότητες του κακόβουλου λογισμικού τη στιγμή που εκμεταλλεύεται τις ευπάθειες του συστήματος. Το Nepenthes προσομοιώνει γνωστές αδυναμίες και αποθηκεύει τον κώδικα κελύφους και το δυαδικό αρχείο που προέρχεται από το κακόβουλο λογισμικό.

Το Nepenthes έχει την ικανότητα να αυτοματοποιεί τη διαδικασία ανάλυσης υποβάλλοντας κάθε ένα από τα δυαδικά αρχεία που συλλέγει σε διάφορα sandboxes. Έτσι οι αναλυτές μπορούν άμεσα να διαπιστώσουν τι είδους αρχεία έχουν στην κατοχή τους ενώ παράλληλα αποφεύγουν να αναλύσουν περισσότερες από μία φορές το ίδιο αρχείο. Ωστόσο, η διαδικασία υποβολής δε λειτουργεί πάντα όπως αναμένεται.

Το Nepenthes κυκλοφόρησε τον Ιούνιο του 2005 από τον Paul Bacher και τον Markus Kotter, οι οποίοι σε συνεργασία και με τον Georg Wicherski υλοποίησαν και το σχεδιασμό του mwcollect 3.0, το οποίο είχε ήδη δημοσιευτεί. Από το Φεβρουάριο του 2006 τα δύο Honeypots συγχωνεύτηκαν και έκτοτε κυκλοφορούν ως ένα με την ονομασία Nepenthes. Το Nepenthes αποτελεί πλέον το πρότυπο των Honeypots ελεύθερου λογισμικού μεσαίας αλληλεπίδρασης και δημοσιεύτηκε από τους Paul Bacher, Markus Kotter και Georg Wicherski του mwcollect.org. Βάσει του Nepenthes, αναπτύχθηκε και το Dionaea, ένα ομοίως χαμηλής αλληλεπίδρασης honeypot. [15]

6.1.11 Thug

Το Thug είναι ένα client Honeypot χαμηλής αλληλεπίδρασης που αναπτύχθηκε από τον Angelo Dell'Aera. Προσομοιώνει τη συμπεριφορά ενός διαδικτυακού φυλλομετρητή και εστιάζει στην ανίχνευση και εντοπισμό κακόβουλων ιστοσελίδων. Χρησιμοποιεί τη μηχανή Google V8 JavaScript και διαθέτει το δικό του Document Object Model, (DOM).

Τα πιο σημαντικά χαρακτηριστικά του είναι τα στοιχεία ελέγχου ActiveX που χρησιμοποιούνται για την αντιμετώπιση ευπαθειών και οι δυνατότητες στατικής και δυναμικής ανάλυσης μέσω της χρήσης του Abstract syntax tree και της βιβλιοθήκης ανάλυσης κελύφους Libemu. Έχει γραφτεί σε Python και διανέμεται ελεύθερα υπό την άδεια της GNU. [16]

6.1.12 Tiny

Το Tiny honeypot αναπτύχθηκε από τον Γεώργιο Μπάκο το 2002. Δεν έχει σχεδιαστεί με στόχο να γίνει αντικείμενο επίθεσης από εισβολείς ως κλασσικό honeypot, αλλά να καταγράψει τη δικτυακή κίνηση που υπό κανονικές συνθήκες δε θα έπρεπε να υφίσταται, επιτρέποντας να γραφούν και να διαμορφωθούν ανάλογα κανόνες που ανιχνεύουν την ανεπιθύμητη δικτυακή κυκλοφορία, (iptables).

Η λειτουργία του Tiny honeypot έγκειται στην καταγραφή οποιασδήποτε προσπάθειας σύνδεσης στις θύρες 22, (TCP) και 80, (HTTP). Η θύρα 53, (UDP-DNS), είναι η μόνη θύρα στην οποία επιτρέπεται η εξερχόμενη κίνηση από το honeypot. [34]

6.1.13 Amun

Το Amun αναπτύχθηκε από τον Jan Gerrit Göbel το 2008. Έχει παρόμοια λειτουργία με το Nepenthes, δηλαδή συλλέγει αυτοματοποιημένα λογισμικό που θεωρείται κακόβουλου τύπου. Έχει γραφτεί σε python και επιτρέπει την εύκολη ενσωμάτωση νέων χαρακτηριστικών. Αποτελεί λογισμικό ανοιχτού κώδικα και διανέμεται υπό της άδεια της GNU. Η τελευταία προσθήκη – αναβάθμιση εκδόθηκε το 2008.

Η αρχιτεκτονική του αποτελείται από τα εξής τμήματα:

- Amun Kernel: Αποτελεί τον πυρήνα του προγράμματος, που είναι υπεύθυνος για τις ρουτίνες εκκίνησης και παραμετροποίησης, καθώς και εν γένει τις βασικές ρουτίνες του προγράμματος. Εκτελείται σε ένα μόνο νήμα, (thread), και χρησιμοποιεί τον τελεστή select για να λειτουργεί πάνω από φορείς υποδοχής, (sockets). Εκτός από τις λειτουργίες στους φορείς υποδοχής διαχειρίζεται τις λήψεις, τις επαναφορτώσεις λογισμικού μετά από αλλαγές σε παραμέτρους, την επεκτασιμότητα του κελύφους και την καταγραφή γεγονότων στο κύριο βρόγχο.

  Κατά τη φάση εκκίνησης, ο πυρήνας αρχικοποιεί τις κανονικές εκφράσεις που χρησιμοποιούνται στον κώδικα, διαβάζει το αρχείο παραμέτρων και δημιουργεί τα vulnerability modules που προσομοιώνουν μεμονωμένες ευπάθειες, τα logging modules που καταγράφουν οποιαδήποτε προσπάθεια σύνδεσης και τα submission modules που για παράδειγμα γράφουν στον σκληρό δίσκο τα δυαδικά αρχεία που έχουν ληφθεί. Στη συνέχεια, ανοίγει τις αντίστοιχες δικτυακές θύρες για καθένα vulnerability module και εισέρχεται στη βασική δομή επανάληψης του προγράμματος.
- Request Handler: Είναι υπεύθυνος για όλη την εισερχόμενη και εξερχόμενη κίνηση του honeypot. Όταν επιχειρείται μία σύνδεση, ο Request Handler τη χειρίζεται και τη στέλνει

προς τον Amun Kernel. Διατηρεί μία λίστα των vulnerability modules που έχουν φορτωθεί και αναθέτει την εισερχόμενη κίνηση στα modules που αντιστοιχούν στην εκάστοτε θύρα.

- Vulnerability Modules: Προσομοιώνουν δικτυακές υπηρεσίες όχι πλήρως αλλά στο βαθμό που απαιτείται για να προσελκύσουν επιθέσεις. Υλοποιούνται ως πεπερασμένα αυτόματα, που σημαίνει ότι μόνο όταν μία επίθεση καταλήξει στο τελικό βήμα του αυτόματου γίνεται δεκτή, διαφορετικά απλά καταγράφεται από τον Request Handler.
- Shellcode Analyzer: Στην περίπτωση που ένα Vulnerability Module επιτυχώς προσομοιώσει μία υπηρεσία ώστε ο εισβολέας να αποστείλει κώδικα, όλη η εισερχόμενη κίνηση καταγράφεται και τελικά μεταφέρεται στον Shellcode Analyzer. Εκεί επιχειρείται η αποκωδικοποίηση του κώδικα με χρήση κανονικών εκφράσεων και εν τέλει η αναγνώριση του είδους της επίθεσης.
- Download Modules: Δημιουργούνται για τη λήψη αρχείων που χαρακτηρίζονται κακόβουλου τύπου βάσει πρωτοκόλλων HTTP, FTP και TFTP. Ακόμα υποστηρίζεται και η απευθείας λήψη αρχείων, όπου το honeypot συνδέεται στη διεύθυνση και τη θύρα που υποδεικνύει ο εισβολέας και λαμβάνει το εν λόγω αρχείο.
- Logging Modules: Καταγράφουν κάθε προσπάθεια σύνδεσης και εν γένει οτιδήποτε σχετίζεται με επιθέσεις στο honeypot. Το Amun διαθέτει τέσσερις τύπους modules: log-syslog (ενημέρωση του δαίμονα syslog), log-mail (αποστολή στοιχείων καταγραφής μέσω ηλεκτρονικού ταχυδρομείου), log-mysql (αποθήκευση στοιχείων καταγραφής σε βάση δεδομένων MySql), log-surfnet (SURFids) και log-blastomat (Blast-o-Mat). [35]

6.1.14 Glastopf

Το Glastopf είναι ένα χαμηλής αλληλεπίδρασης Honeypot που εξομοιώνει έναν διαδικτυακό διακομιστή με ευπάθειες προκειμένου να συγκεντρώσει επιθέσεις που στόχο έχουν διαδικτυακές εφαρμογές. Έχει γραφτεί σε Python και η πρακτική που ακολουθεί είναι να στέλνει στον εισβολέα την αναμενόμενη απόκριση από ένα στόχο στον οποίο επιτέθηκε επιτυχώς.

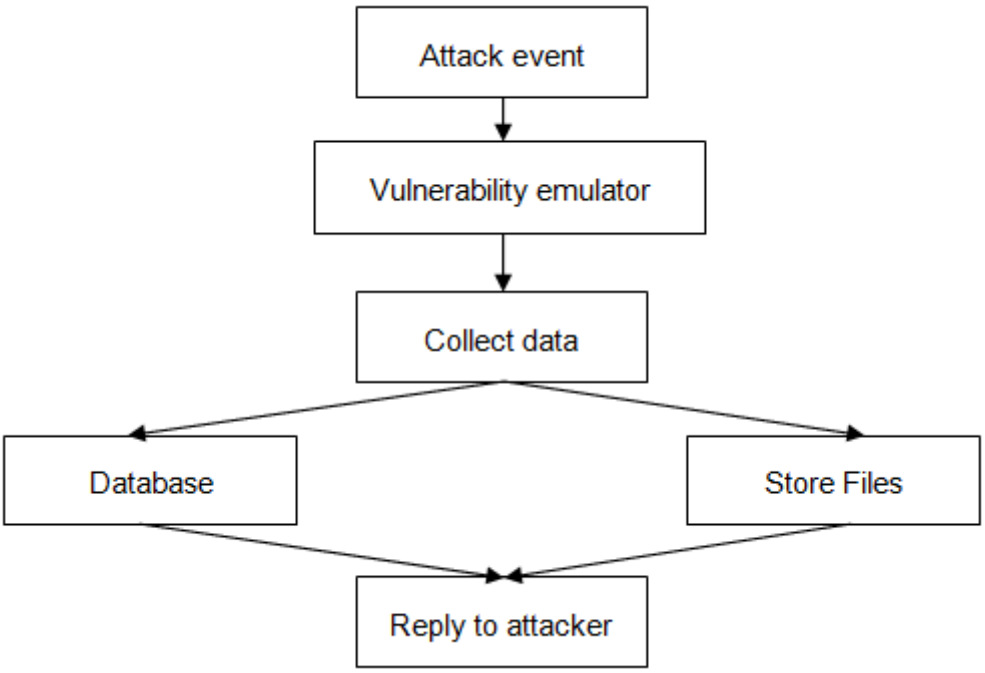
Το Glastopf σκοπό έχει να συγκεντρώσει πληροφορίες για επιθέσεις που αφορούν σε διαδικτυακές εφαρμογές, όπως εκτέλεση κώδικα σε απομακρυσμένη ιστοσελίδα μέσω κελύφους που φιλοξενείται στη σελίδα του εισβολέα, (Remote File Inclusion – RFI), χρήση επερωτήσεων για εκμαίευση ευαίσθητων πληροφοριών, (SQL injections), πρόσβαση σε ευαίσθητους καταλόγους, (Local File Inclusion – LFI).

Η διαφορά του Glastopf από άλλα διαδικτυακά Honeypots είναι ότι μπορεί να προσελκύσει πολυεπίπεδες επιθέσεις με χρήση ενός εξομοιωτή ευπαθειών και μίας λίστας αιτήσεων κακόβουλου τύπου που θα περιγραφούν παρακάτω αντί να χρησιμοποιεί τροποποιημένα πρότυπα διαδικτυακών εφαρμογών που χρησιμοποιούνται από μηχανές αναζήτησης για να προσελκύουν εισβολείς.

Πιο συγκεκριμένα, το Glastopf ελέγχει τις εισερχόμενες αιτήσεις ψάχνοντας για περιεχόμενες συμβολοσειρές όπως για παράδειγμα “=http://” ή “cast(0x”. Αν βρει κάποια, επιχειρεί να λάβει το αρχείο αυτό και να απαντήσει στον εισβολέα ιδανικά όπως αυτός αναμένει, οπότε ο εισβολέας στη συνέχεια ενδεχομένως στέλνει λογισμικό κακόβουλου τύπου. Όλες οι εισερχόμενες στο Glastopf συνδέσεις και όλες οι πληροφορίες και τα στοιχεία που αφορούν στις συνδέσεις καταγράφονται σε βάση δεδομένων.

Στην εικόνα 6-6 παρουσιάζεται η γενική αρχιτεκτονική του Glastopf.

Το Glastopf αναπτύχθηκε το 2009 από τον Lukas Rist. Έκτοτε έχουν ανακοινωθεί δύο εκδόσεις, η Glastopf v1 και η Glastopf NG v2 και επί του παρόντος αναπτύσσεται η έκδοση Glaspot v3. [58]



Εικόνα 6-6: Γενική αρχιτεκτονική Glastopf. [58]

Αρχιτεκτονική – Λειτουργικότητα

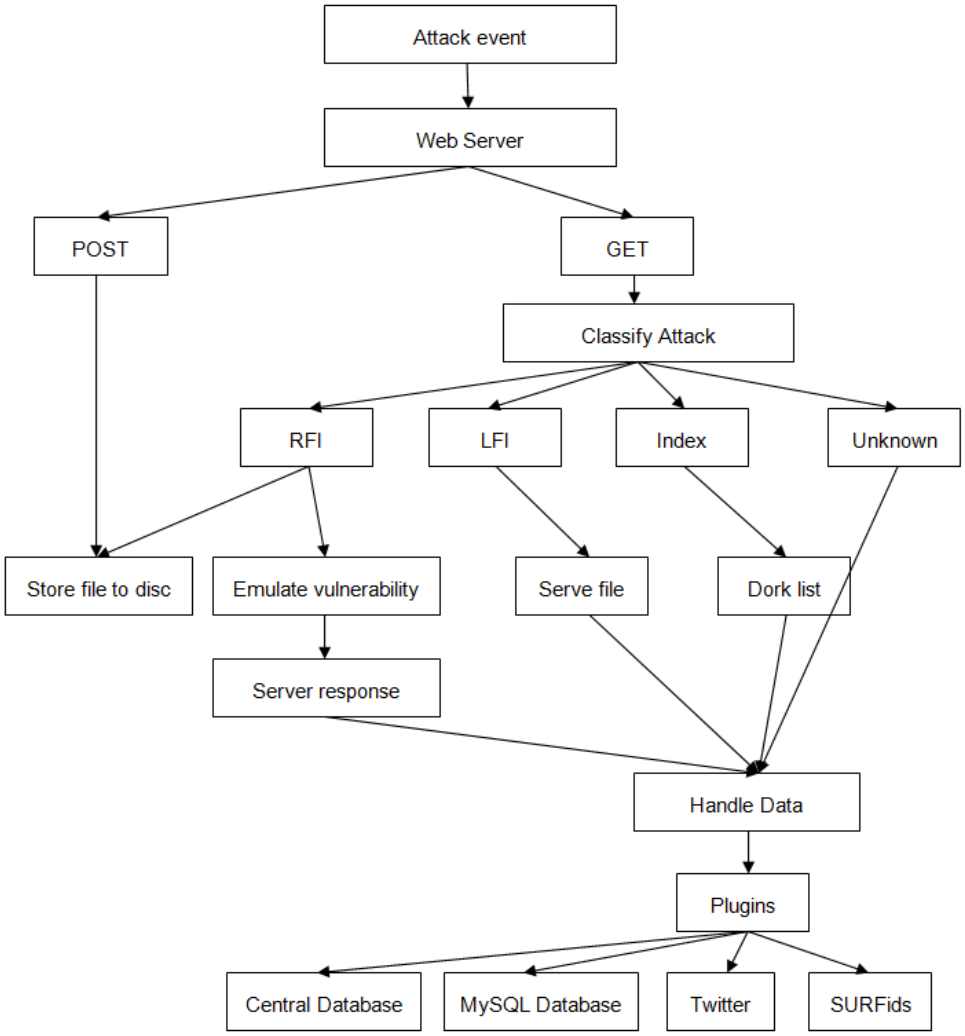
Το Glastopf είναι ουσιαστικά ένας διαδικτυακός διακομιστής που σκοπό έχει να δεχτεί αιτήσεις κακόβουλου τύπου, να τις επεξεργαστεί και να στείλει πίσω την αναμενόμενη απόκριση, ώστε να θεωρηθεί από τον εισβολέα ευπαθής.

Μία πλήρης αίτηση HTTP αποτελείται από τρία τμήματα, από τη μέθοδο που χρησιμοποιείται, (HEAD, GET ή POST), από την καθεαυτή αίτηση και από την έκδοση πρωτοκόλλου HTTP που χρησιμοποιείται. Το Honeypot απαντάει στις HEAD αιτήσεις με μία γενική τυπική επικεφαλίδα διαδικτυακού διακομιστή ενώ στις POST αιτήσεις αποθηκεύει το πλήρες περιεχόμενο της αίτησης. Οι GET αιτήσεις είναι αυτές που επιδέχονται ανάλυση βάσει της οποίας προκύπτουν οι αποκρίσεις του Glastopf.

Για την ανάλυση των GET αιτήσεων το Glastopf χρησιμοποιεί ένα σύνολο προτύπων, (patterns), προκειμένου να τις ταξινομήσει. Έτσι, οι αιτήσεις μπορούν για παράδειγμα να ταξινομηθούν σε RFI επιθέσεις ή επιθέσεις που στοχεύουν στην phpmyadmin πλατφόρμα ή στον διαχειριστή tomcat manager του apache tomcat διακομιστή.

Στη συνέχεια, το Glastopf βάσει της ταξινόμησης της επίθεσης διαμορφώνει την αναμενόμενη για τον εισβολέα απόκριση. [58]

Στην εικόνα 6-7 παρουσιάζεται ο τρόπος λειτουργίας του Glastopf.



Εικόνα 6-7: Λειτουργικότητα Glastopf. [58]

Κατά την RFI επίθεση ο εισβολέας συμπεριλαμβάνει το αρχείο κακόβουλου τύπου στην αίτησή του ώστε να εκτελεστεί στον διακομιστή προς τον οποίο έγινε η αίτηση. Το Glastopf σε αυτή την περίπτωση αποκρίνεται στον εισβολέα με την επικεφαλίδα HEAD, λαμβάνει το αρχείο κακόβουλου τύπου, ψάχνει στον κώδικα του εισβολέα για echo εντολές – στην περίπτωση php κελύφους – ή εν γένει για μεταβλητές που αναμένονται να συμπληρωθούν από τον διακομιστή και αποκρίνεται στον εισβολέα εκτελώντας τις εντολές ή συμπληρώνοντας τις μεταβλητές. Ο εισβολέας κατ'αυτόν τον τρόπο λαμβάνει την απάντηση που περίμενε.

Κατά την LFI επίθεση ο εισβολέας προσπαθεί να αποκτήσει αυστηρά σημαντικές πληροφορίες συστήματος ή να εκτελέσει κώδικα δικό του απομακρυσμένα. Εάν επιχειρήσει να αποκτήσει αρχεία συστήματος όπως passwd, τότε το Glastopf απαντάει με δυναμικά παραγόμενα αρχεία που φέρουν όμως ψεύτικα δεδομένα, ώστε να προκαλέσει και να ενθαρρύνει επιπλέον επιθέσεις.

Ως Dork List καλείται η λίστα που δημιουργείται κατά τις επιθέσεις προς το Honeypot και περιέχει όλες τις συμβολοσειρές που αφορούν σε μονοπάτια που οδηγούν σε αρχεία κακόβουλου τύπου. Είναι μια λίστα δηλαδή ευπαθών αιτήσεων – διευθύνσεων, τις οποίες οι εισβολείς μέσω μηχανών αναζήτησης ακολουθούν όταν ψάχνουν για νέα θύματα. Το Honeypot οποτεδήποτε δέχεται μία αίτηση κακόβουλου τύπου που περιέχει διεύθυνση, καταχωρεί τη διεύθυνση αυτή στη βάση δεδομένων. Με τη σειρά τους οι μηχανές αναζήτησης όταν ανιχνεύσουν το Honeypot, θα καταχωρήσουν την ευπαθή αυτή διεύθυνση ως διεύθυνση του

Honeypot. Έτσι, οι εισβολείς ψάχνοντας για στόχους που θα δεχτούν ευπαθείς διευθύνσεις θα οδηγηθούν στο Glastopf. Η μέθοδος αυτή αυξάνει τη δημοφιλία του Glastopf στους εισβολείς καθώς όλο και περισσότερες επερωτήσεις θα εμφανίζουν όλο και ψηλότερα το Glastopf στα αποτελέσματα των μηχανών αναζήτησης.

Η Central Database του Glastopf είναι μία MySQL βάση δεδομένων που καταγράφει δεδομένα που προέρχονται από Glastopf διακομιστές. Λειτουργεί βάσει script που έχει γραφτεί σε Python. Για την εκκίνηση της λειτουργίας της απαιτείται η ανάθεση IP διεύθυνσης στο μηχάνημα του Glastopf διακομιστή, η ανάθεση θύρας στο μηχάνημα για να δέχεται συνδέσεις και η εγκαθίδρυση MySQL σύνδεσης. Προς το παρόν, η δυνατότητα αυτή δεν είναι διαθέσιμη.

SURFids είναι μία διαδικτυακή πλατφόρμα που μπορεί να δεχτεί συνδέσεις από Glastopf διακομιστές, να καταχωρήσει τα δεδομένα τους σε βάση δεδομένων και να προβάλλει στατιστικά στοιχεία. Στην τρέχουσα έκδοση η δυνατότητα αυτή πλέον δεν παρέχεται.

Τέλος, στην πρώτη έκδοση του Glastopf παρεχόταν η δυνατότητα σύνδεσης με το Twitter. Ο χρήστης μπορούσε να λάβει στο λογαριασμό του στο Twitter ποσοτικά δεδομένα σχετικά με τα αποτελέσματα του Glastopf. Η δυνατότητα αυτή πλέον δεν παρέχεται. [58]

6.1.15 Σύνοψη

Honeypot	Χρόνος	Client	Server	Λειτουργικότητα
Dionaea	2009		✓	Συλλέγει αντίγραφα λογισμικού κακόβουλου περιεχομένου εξομοιώνοντας μια σειρά από πρωτόκολλα, (SMB, HTTP, FTP, MSSQL, MYSQL, SIP κ.α.)
BackOfficer friendly	2002		✓	Προσομοιώνει τα HTTP, FTP, TELNET, SMTP, POP3, IMAP και BackOrifice πρωτόκολλα
Specter	2002		✓	Προσομοιώνει 13 λειτουργικά συστήματα και 14 πρωτόκολλα, καταγράφει την πλήρη δραστηριότητα του εισβολέα
Honeyd	2003			Daemon πρόγραμμα – δημιουργεί εικονικούς κόμβους σε ένα δίκτυο καταλαμβάνοντας αχρησιμοποίητες IP διευθύνσεις
HoneyC	2008	✓		Ανιχνεύει για κακόβουλους εξυπηρετητές στο Διαδίκτυο εξετάζοντας την απόκρισή τους
Monkey – Spider	2007	✓		Ανιχνεύει ιστοσελίδες για ύπαρξη

				κακόβουλου περιεχομένου
PhoneyC	2009	✓		Εικονικό Honeypot, ανιχνεύει ιστοσελίδες για ύπαρξη κακόβουλου περιεχομένου
SpyBye	2007		✓	Proxy server – Κατηγοριοποιεί τις λήψεις λογισμικού που εκτελεί ο browser σε ακίνδυνες ή μη
LaBrea	2003			Καταλαμβάνει αχρησιμοποίητες IP διευθύνσεις και δημιουργεί εικονικούς servers που προσελκύουν worms, hackers και λοιπές απειλές και συνδέονται σε αυτά. → αργές αποκρίσεις – sticky Honeypot.
Nepenthes	2005	✓		Προσομοιώνει πολλές υπηρεσίες, αποθηκεύει shell code και δυαδικό αρχείο κακόβουλου περιεχομένου – πρόγονος του Dionaea.
Thug	2011	✓		Προσομοιώνει browser, ανιχνεύει για κακόβουλες ιστοσελίδες
Tiny	2002			Καταγράφει δικτυακή κίνηση κάνοντας χρήση iptables.
Amun	2008		✓	Συλλέγει λογισμικό που θεωρείται κακόβουλου περιεχομένου.
Glastopf	2010		✓	Web server – Ανιχνεύει για web επιθέσεις, (SQL injections, RFI, LFI).

Πίνακας 6-1: Συνοπτική παρουσίαση Honeypots χαμηλής αλληλεπίδρασης.

6.2 Honeypots μεσαίας αλληλεπίδρασης

6.2.1 Kippo

Το Kippo είναι ένα Honeypot μεσαίας αλληλεπίδρασης το οποίο έχει σχεδιαστεί για να καταγράφει επιθέσεις αλλά και την πλήρη σύνδεση και διάδραση του εισβολέα με το κέλυφος.

Αντικείμενο έμπνευσης για το Kippo στάθηκε το Kojoney. Έχει γραφτεί σε Python και είναι ένα project ανοιχτού κώδικα το οποίο γίνεται host στο Google Code από τον Upi Tamminen.

Ουσιαστικά προσομοιώνει έναν ssh server.

Μερικά ενδιαφέροντα χαρακτηριστικά του kippo είναι τα ακόλουθα:

- Κάνει χρήση ενός πλασματικού συστήματος αρχείων με τη δυνατότητα για πρόσθεση/ αφαίρεση αρχείων. Στην τελευταία έκδοση περιλαμβάνεται ένα πλήρες πλασματικό σύστημα αρχείων το οποίο προσομοιώνει την εγκατάσταση του λειτουργικού Debian 5.
- Παρέχει τη δυνατότητα προσθήκης πλασματικού περιεχομένου αρχείων ώστε ο εισβολέας να μπορεί να διαβάσει, να έχει πρόσβαση σε αρχεία ασφαλείας, όπως /etc/passwd. Στην τελευταία έκδοση περιέχονται τέτοια αρχεία περιορισμένων δυνατοτήτων.
- Δυνατότητα καταγραφής και αποθήκευσης δεδομένων σε format συμβατό με τη γλώσσα UML ώστε τα δεδομένα να μπορούν να αναπαραχθούν εύκολα και στην πραγματική χρονική διάρκειά τους.
- Αποθήκευση αρχείων που έχουν «κατέβει» με χρήση του wget με στόχο τη μελλοντική αξιοποίησή τους.
- Χρήση διάφορων τεχνασμάτων, όπως δημιουργία ψευδών συνδέσεων με το ssh και μη αληθινή εκτέλεση εντολών, π.χ. η εντολή exit να μην τερματίζει πραγματικά.

Οι απαιτήσεις σε λογισμικό για τη λειτουργία του kippo είναι οι εξής: λειτουργικό σύστημα – έχει δοκιμαστεί σε Debian, CentOS, FreeBSD και Windows 7, Python 2.5+, Twisted 8.0+, PyCrypto και Zope Interface. [17]

Η βασική λειτουργία του kippo είναι ότι «ακούει» για ssh συνδέσεις στη θύρα 2222 και οποιαδήποτε σύνδεση δεχτεί στη θύρα αυτή, την καταγράφει σε βάση δεδομένων.

Το Kippo εκδόθηκε το 2009 και η τελευταία του αναβάθμιση ανακοινώθηκε το Νοέμβρη του 2010.

6.2.2 Deception Toolkit

Το Deception Toolkit (DTK) είναι ένα από τα πρώτα Honeypots. Δημιουργήθηκε από τον Fred Cohen το 1998 και η τελευταία έκδοσή του έγινε το 1999.

Το DTK είναι ένα εργαλείο που προσομοιώνει έναν μεγάλο αριθμό ευρέως γνωστών ευπαθειών. Η προσομοίωση αυτή είναι ελεγχόμενη και συνήθως περιορίζεται στην παραγωγή αποτελέσματος που θα συνάδει με την εισβολή – είσοδο δεδομένων του εισβολέα στο σύστημα, ώστε η προσομοίωση να μην μπορεί να γίνει αντιληπτή από τον εισβολέα. Ωστόσο, υπάρχουν και κάποια μειονεκτήματα.

- Ο φόρτος εργασίας του εισβολέα αυξάνεται ραγδαία γιατί ο εισβολέας δεν μπορεί να αναγνωρίσει ποιες από τις επιθέσεις του ήταν επιτυχείς και ποιες όχι. Για παράδειγμα, αν μία εισβολή παράξει ένα Unix αρχείο κωδικών, ο εισβολέας εκτελεί ένα αρχείο crack για να ανακτήσει τους κωδικούς και να αποκτήσει πλήρη πρόσβαση στο σύστημα. Αν το αρχείο όμως αυτό είναι ψεύτικο, ο εισβολέας θα έχει καταναλώσει χρόνο και πόρους χωρίς αντίκρυσμα και ενδεχομένως θα αντιληφθεί την «παγίδα» που στήθηκε.
- Η δημοφιλία του DTK το κατέστησε σταδιακά μη ιδιαίτερα αποδοτικό, αφού από τη στιγμή που όλο και περισσότερες επιθέσεις μπορούσαν να γίνουν αντιληπτές και εν συνεχεία να αντιμετωπισθούν, οι εισβολείς με τη σειρά τους άρχισαν να αντιλαμβάνονται τη λειτουργία του εργαλείου και να αναζητούν καινούρια συστήματα

για την υλοποίηση των επιθέσεών τους. Αντίστοιχα όμως δημιουργήθηκε το κίνητρο για περαιτέρω ανάπτυξη του DTK ώστε να μπορεί να αντιμετωπίσει και πιο «δύσκολες» επιθέσεις.

Το DTK έχει μεγαλύτερες δυνατότητες από το χαμηλής αλληλεπίδρασης Honeypot Specter και μπορεί να μας δώσει περισσότερες πληροφορίες, αλλά χρειάζεται περισσότερη δουλειά για να εγκατασταθεί και έχει πρόσθετους κινδύνους. Ωστόσο, δεν είναι ακόμα ένα Honeypot υψηλής αλληλεπίδρασης, καθώς δεν προσομοιώνει πραγματικό λειτουργικό σύστημα για να αλληλεπιδράσει με τον εισβολέα, παρά μόνο υπηρεσίες.

Το DTK είναι μια συλλογή από Perl scripts και C προγράμματα σχεδιασμένα για Unix συστήματα που μιμούνται μια ποικιλία από γνωστά τρωτά σημεία. Για παράδειγμα, ορισμένα scripts μιμούνται ευάλωτους διακομιστές SMTP.

Ένα εξίσου μεγάλο πλεονέκτημα του DTK είναι ότι αποτελεί λογισμικό ανοικτού κώδικα. Αυτό αποτελεί όμως ταυτόχρονα και μειονέκτημα καθώς ο κώδικας μπορεί δυνητικά να αξιοποιηθεί από τον εισβολέα. [18]

6.2.3 Mwcollectd

Το mwcollectd ήταν το πρώτο Open Source Honeypot της κατηγορίας αυτής. Αναπτύχθηκε από τον Georg Wicherski και πρωτοδημοσιεύτηκε το Μάρτιο του 2005. Τον Σεπτέμβριο του 2005 βγήκε η έκδοση 3.0 στην οποία το mwcollectd ήταν αισθητά βελτιωμένο σε σχέση με το παλιό. Η τελευταία έκδοσή του, v4, κυκλοφόρησε το 2009. [19]

Έχει γραφτεί σε C++ και επιτρέπει την ενσωμάτωση Python ενοτήτων, (modules), που καθιστούν το Honeypot ευέλικτο και επεκτάσιμο με νέα πρωτόκολλα και χαρακτηριστικά.

Το Φεβρουάριο του 2006 το mwcollectd συγχωνεύθηκε με το Nepenthes για ευρύτερη ανάπτυξη και μεγαλύτερη αποδοτικότητα. Δεδομένου ότι και τα δύο λειτουργούσαν ήδη υπό την άδεια GNU, δεν υπήρχαν ζητήματα αδειοδότησης. Ο κώδικας Nepenthes χρησιμοποιήθηκε ως ο νέος κώδικας βάσης, δεδομένου ότι περιείχε περισσότερα modules. Από τότε το νέο Honeypot ονομάζεται Nepenthes και διαμορφώθηκε ως Honeypot χαμηλής αλληλεπίδρασης. [7]

6.2.4 Multipot

Το Multipot είναι ένα Honeypot μεσαίας αλληλεπίδρασης για Windows που κάνει χρήση γραφικού περιβάλλοντος και έχει περιορισμένη επεκτασιμότητα για διανομή και ανάπτυξη. Προσομοιώνει ευπαθείς υπηρεσίες ώστε να συγκεντρώνει με ασφαλή τρόπο κακόβουλο λογισμικό. Διαθέτει αρκετά κοινά στοιχεία με το mwcollectd.

Αρχικά κυκλοφόρησε τον Ιούλιο του 2005 από την iDefense και έκτοτε δεν έχει κυκλοφορήσει κάποια άλλη επίσημη έκδοση. [7]

6.2.5 HoneySpider

Το δίκτυο HoneySpider αναπτύχθηκε ως κοινοπραξία μεταξύ Nask / CERT Polska, GOVCERT.NL και Surfnet.

Στόχος της ανωτέρω κοινοπραξίας ήταν η ανάπτυξη ενός Client Honeypot συστήματος με χρήση των πλέον προηγμένων Client Honeypots, το οποίο έχοντας τη δυνατότητα επεξεργασίας διευθύνσεων URL θα ανιχνεύει και θα διακρίνει διευθύνσεις κακόβουλου περιεχομένου.

Ουσιαστικά το σύστημα αφορά σε επιθέσεις που γίνονται ενάντια σε ή εν γένει αφορούν σε web browsers. Αυτές οι επιθέσεις περιλαμβάνουν ανιχνεύσεις για ελεγχόμενες διαδικτυακές λήψεις, δυαδικά αρχεία κακόβουλου περιεχομένου καθώς και απόπειρες ηλεκτρονικού ψαρέματος. Παράλληλα, το σύστημα αυτόματα λαμβάνει και αναλύει το κακόβουλο λογισμικό και τελικά αναπαράγει μία ψηφιακή υπογραφή για αυτό.

Το βασικότερο κίνητρο για την ανάπτυξη του δικτύου ήταν η ραγδαία αύξηση των επιθέσεων σε browsers μέσω διεπαφών χρήστη. Οι επιθέσεις αυτές βρίσκονται εκτός της εμβέλειας των υφιστάμενων συστημάτων παρακολούθησης που χρησιμοποιούνται εντός

τοπικών δικτύων. Οπότε απαιτήθηκε η επέκταση των υφιστάμενων δυνατοτήτων παρακολούθησης και η πρόληψη ενάντια σε απειλές από κακόβουλο λογισμικό.

Σταδιακά το σύστημα αναμένεται να βελτιώσει την επίγνωση από τη μεριά του χρήστη της εκάστοτε κατάστασης του συστήματος στο Διαδίκτυο αλλά και τις προσφερόμενες υπηρεσίες ασφάλειας. [20]

6.2.6 Trigona

Το Trigona Honeypot ανακοινώθηκε το Δεκέμβριο του 2010 από την Αυστραλιανή Ομάδα Ανάπτυξης Honeypots, (Australian Honeynet Project). Σχεδιάστηκε ώστε να μπορεί να εξακριβώσει εάν μία διεύθυνση URL είναι κακόβουλη. Ως κακόβουλη χαρακτηρίζεται μία διεύθυνση, η οποία εφόσον προσπελαστεί από κάποιο browser, εγκαθιστά ένα εκτελέσιμο αρχείο στον υπολογιστή του χρήστη.

Χρησιμοποιεί την εφαρμογή Sandboxie, η οποία λειτουργεί σαν ένα sandbox πρόγραμμα, επιτρέπει δηλαδή τη μεμονωμένη εκτέλεση ενός προγράμματος στον υπολογιστή, ώστε το πρόγραμμα που εκτελείται να μην επηρεάζει οποιοδήποτε άλλο πρόγραμμα αλλά και να μη δεσμεύει οποιοδήποτε άλλο υπολογιστικό πόρο. Με τη χρήση του Sandboxie μπορούν να παραμένουν ταυτόχρονα στο ίδιο στιγμιότυπο του Sandboxie ανοιχτοί πολλαπλοί Internet Explorer browsers και να προσπελάζουν κακόβουλες URL διευθύνσεις. Κατόπιν, το Sandboxie μπορεί να κλείσει και το σύστημα αρχείων να εξεταστεί για οποιαδήποτε ύποπτη αλλαγή. Εάν οι αλλαγές αυτές αποτελούν καινούρια δυαδικά αρχεία, τότε τα περιεχόμενα του συστήματος αρχείων συμπιέζονται και η εκάστοτε διεύθυνση URL χαρακτηρίζεται στη βάση δεδομένων ως ύποπτη.

Στην ουσία, το Trigona Honeypot είναι ένα VirtualBox το οποίο σχεδιάστηκε για υψηλή ρυθμαπόδοση και χρησιμοποιεί τα καλύτερα υψηλής και χαμηλής αλληλεπίδρασης Client Honeypots σε συνδυασμό με Perl scripts. Η χρήση υψηλής αλληλεπίδρασης Client Honeypots ενδείκνυται γιατί μπορεί να καταγράψει οτιδήποτε δεν μπορεί να καταγράψει ένα χαμηλής αλληλεπίδρασης Client Honeypot και αντίστροφα. Ωστόσο ένα υψηλής αλληλεπίδρασης Client Honeypot είναι ελαφρώς βραδύτερο από ένα χαμηλής αλληλεπίδρασης καθώς απαιτεί ένα εικονικό μηχάνημα για κάθε διεύθυνση URL που αναλύεται σε αντίθεση με ένα εργαλείο γραμμής εντολών που μπορεί σε μικρό χρονικό διάστημα να ελέγξει πολλαπλές συνδέσεις. [21]

6.2.7 Σύνοψη

Honeypot	Χρόνος	Client	Server	Λειτουργικότητα
Kippo	2009		✓	SSH server – παρέχει πλήρες πλασματικό σύστημα αρχείων
Deception Toolkit	1998		✓	Προσομοιώνει μεγάλο αριθμό ευπαθειών, παράγει αποκρίσεις αναμενόμενες για τον εισβολέα.
Multipot	2005		✓	Προσομοιώνει ευπαθείς υπηρεσίες, συλλέγει κακόβουλο λογισμικό.
HoneySpider		✓		Ανίχνευση web επιθέσεων μέσω browsers.

Trigona	2010	✓		VirtualBox – Εξετάζει ιστοσελίδες αν είναι κακόβουλου περιεχομένου, χρησιμοποιεί υψηλής και χαμηλής αλληλεπίδρασης Honeypots.

Πίνακας 6-2: Συνοπτική παρουσίαση Honeypots μεσαίας αλληλεπίδρασης.

6.3 Honeypots υψηλής αλληλεπίδρασης

6.3.1 ManTrap

Το ManTrap είναι ένα εμπορικό Honeypot υψηλής αλληλεπίδρασης που αναπτύχθηκε, σχεδιάστηκε και πωλείται από την Recourse Technologies. Μετατρέπει το Honeypot σε έναν ψευδή κόμβο υποδοχής. Αποτελεί ουσιαστικά ένα πλήρες λειτουργικό σύστημα που λειτουργεί σα δόλωμα για τον επιτιθέμενο. Αυτοματοποιεί διάφορες εργασίες, όπως το «γέμισμα» του κόμβου υποδοχής με δεδομένα και την αναβάθμιση εν συνεχεία αυτών των δεδομένων προκειμένου να πειστεί ο επιτιθέμενος ότι ο στόχος του είναι ένας πλήρως λειτουργικός και παραγωγικός κόμβος. Προσπαθεί να αναγνωρίσει παράλληλα και την ιδιότητα και τους συγκεκριμένους σκοπούς του επιτιθέμενου ενώ τα εργαλεία ανάλυσης που χρησιμοποιεί επιταχύνουν τον έλεγχο δραστηριότητας του επιτιθέμενου. [1]

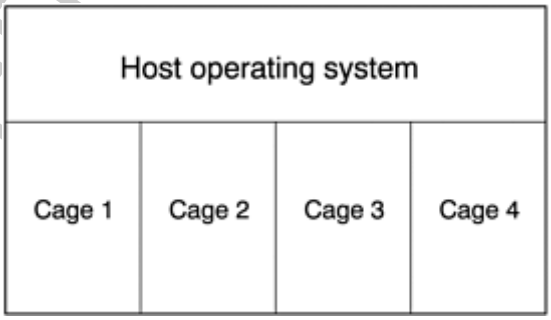
Γενική περιγραφή – λειτουργία

Το λειτουργικό σύστημα που δημιουργεί το ManTrap περιέχει εικονικά υποσυστήματα, που αποκαλούνται φυλακές, (cages). Οι φυλακές είναι πλήρως ελεγχόμενα και απομονωμένα περιβάλλοντα από τα οποία ο επιτιθέμενος δεν μπορεί να εξέλθει και να επιτεθεί σε όλο το σύστημα. Ωστόσο, αντί οι φυλακές αυτές να είναι άδειες και περιορισμένης λειτουργικότητας, είναι ακριβή αντίγραφα του λειτουργικού συστήματος στο οποίο ανήκουν. Διαθέτουν το δικό τους σύστημα αρχείων, τα δικά τους δυαδικά αρχεία, τις δικές τους βιβλιοθήκες και τη δική τους κάρτα δικτύου. Επομένως καθένα από αυτά τα υποσυστήματα είναι ένα πλήρες λειτουργικό σύστημα με αναμενόμενες δυνατότητες μίας εγκατάστασης παραγωγής. Οι εισβολείς άπαξ και εισέλθουν εγκλωβίζονται σε αυτά τα υποσυστήματα και κάθε κίνησή τους καταγράφεται. Καμία φυλακή δε γνωρίζει για την ύπαρξη των άλλων ή και για το κεντρικό λειτουργικό σύστημα.

Στην εικόνα 6-8 διαφαίνεται ένας ManTrap κόμβος με τέσσερις φυλακές.

Ωστόσο, η πεποίθηση των εισβολέων ότι έχουν επιτεθεί σε ένα πραγματικό λειτουργικό σύστημα περιορίζεται κυρίως σε εισβολείς, οι οποίοι επικεντρώνονται σε συγκεκριμένους στόχους, εν γένει ευκαιριακούς όπως σε εύκολη καταστροφή διαδικασιών και εφαρμογών. Οι εισβολείς αυτοί χρησιμοποιούν αυτοματοποιημένα εργαλεία καθολικής αναζήτησης και επίθεσης σε ευπαθή συστήματα. Η παραπλάνηση και η αποτροπή δε λειτουργούν εναντίον αυτοματοποιημένων επιθέσεων. Επομένως, το ManTrap μπορεί να χρησιμοποιηθεί για την παραπλάνηση ή την αποτροπή επιθέσεων πραγματικών ανθρώπων. [1]

Επί του παρόντος, το ManTrap εκτελείται σε Solaris και σε βασισμένα σε Unix περιβάλλοντα.



Εικόνα 6-8: Ένας ManTrap κόμβος με τέσσερις φυλακές. [1]

6.3.2 Capture – HPC

Το Capture – HPC αναπτύχθηκε στο Victoria University of Wellington από τους Ramon Steenson και Christian Seifert το 2008 και είναι ένα υψηλής αλληλεπίδρασης Client Honeypot. Ως Client Honeypot ψάχνει για κακόβουλους εξυπηρετητές σε ένα δίκτυο, τους οποίους αναγνωρίζει μέσω αλληλεπίδρασης χρησιμοποιώντας ένα εικονικό μηχάνημα και

παρακολουθώντας τις αλλαγές καταστάσεων στο σύστημά του. Εάν οποιαδήποτε αλλαγή παρατηρηθεί στην κατάσταση του συστήματος του εικονικού μηχανήματος, ο εξυπηρετητής με τον οποίο αλληλεπίδρασε το εικονικό μηχάνημα θεωρείται κακόβουλου περιεχομένου.

Έχει ορισμένα πλεονεκτήματα σε σχέση με άλλα Client Honeypots. Είναι γρηγορότερο καθώς οι αλλαγές κατάστασης ανιχνεύονται χρησιμοποιώντας ένα μοντέλο βασισμένο σε γεγονότα, (event-driven), που επιτρέπει την αντίδραση στις αλλαγές της κατάστασης, όπως αυτές συμβαίνουν. Είναι επεκτάσιμο αφού ένας κεντρικός εξυπηρετητής μπορεί να ελέγξει πολυάριθμους clients σε ένα δίκτυο και επιτρέπει τη χρήση πολλαπλών clients. Η αρχική έκδοση του Capture – HPC υποστήριζε τον Internet Explorer, αλλά η τρέχουσα έκδοση υποστηρίζει και άλλα δημοφιλή προγράμματα περιήγησης (Firefox, Opera, Safari), καθώς και άλλες εφαρμογές, όπως εφαρμογές γραφείου και εφαρμογές αναπαραγωγής πολυμέσων. [22]

Λειτουργία

Η λειτουργία του Capture – HPC χωρίζεται σε δύο τομείς, τον Capture server και τον Capture client. Ο βασικός σκοπός του Capture server είναι να ελέγχει πολυάριθμους Capture clients ώστε να αλληλεπιδρούν με web servers. Μπορεί να εκκινεί και να παύει clients, να οδηγεί τους clients να επικοινωνούν με web servers ζητώντας μία συγκεκριμένη διεύθυνση ή συγκεντρώνοντας τις αποκρίσεις των clients σχετικά με τους web servers με τους οποίους έχουν αλληλεπιδράσει. Ο server παρέχει αυτή τη λειτουργικότητα μέσω scripts και οι clients υλοποιούν αυτή τη λειτουργία υπακούοντας στις εντολές του server. Καθώς ένας client επικοινωνεί με έναν web server, παρακολουθεί την κατάστασή του για τυχόν αλλαγές στο σύστημα αρχείων, στο μητρώο και στις διαδικασίες που εκτελούνται. Δεδομένου ότι ορισμένα γεγονότα συμβαίνουν κατά τη διάρκεια κανονικής λειτουργίας, (π.χ. γράψιμο αρχείων στην προσωρινή μνήμη του προγράμματος περιήγησης στο web), λίστες εξαιρέσεων δεν επιτρέπουν συγκεκριμένο τύπο γεγονότων. Αν ανιχνευθούν αλλαγές που δεν ανήκουν στη λίστα εξαιρέσεων, ο client ταξινομεί ως κακόβουλο τον web server και στέλνει τις πληροφορίες αυτές στον Capture server. Οι λίστες εξαιρέσεων υλοποιούνται από την Boost:: regex βιβλιοθήκη. Από τη στιγμή που αλλάζει η κατάσταση ενός client, ο client επαναφέρει την κατάστασή του στην αρχική κατάσταση προτού ανακτήσει καινούριες οδηγίες από τον Capture server. Σε περίπτωση που δεν αλλάξει η κατάσταση του client, ο client περιμένει καινούριες οδηγίες από τον Capture server χωρίς να επαναφέρει την κατάστασή του σε αρχική κατάσταση. Όταν ένας κακόβουλος server ανιχνευθεί, ο Capture server αυτόματα συλλέγει χωματερές από το δίκτυο και ληφθέντα αρχεία. [22]

Τεχνική περιγραφή

Ο Capture server είναι ένας TCP/IP server που ελέγχει διάφορους clients καθώς και τους VMware servers που φιλοξενούν το εικονικό μηχάνημα πάνω στο οποίο εκτελούνται οι Capture clients. Κατανέμει κάθε διεύθυνση που λαμβάνει στους διαθέσιμους clients σύμφωνα με τον αλγόριθμο χρονοπρογραμματισμού Round – Robin. Ακούει για clients που συνδέονται στον server σε συγκεκριμένη TCP θύρα. Έχει γραφτεί σε Java και ελέγχει τους VMware servers κάνοντας χρήση του VMware C API. Το πρωτόκολλο επικοινωνίας που χρησιμοποιείται μεταξύ του Capture server και του Capture client είναι βασισμένο σε XML και ονομάζεται Capture Communication Protocol.

Ο Capture client αποτελείται από δύο συστατικά στοιχεία, ένα σύνολο προγραμμάτων οδήγησης πυρήνα, (kernel drivers), και μια διαδικασία για το χώρο μνήμης του χρήστη, (user space process). Τα προγράμματα οδήγησης πυρήνα λειτουργούν σε χώρο μνήμης πυρήνα και χρησιμοποιούν μηχανισμούς ανίχνευσης που βασίζονται στα γεγονότα για την παρακολούθηση των αλλαγών κατάστασης του συστήματος. Η διαδικασία για το χώρο μνήμης του χρήστη δέχεται αιτήσεις επισκεψιμότητας από τον Capture server, οδηγεί τον client να αλληλεπιδρά με τον server και μεταφέρει τις αλλαγές κατάστασης συστήματος πίσω στον server μέσω μίας απλής σύνδεσης TCP/IP. Ακόμα, καταγράφει τις αλλαγές κατάστασης από τα προγράμματα οδήγησης του πυρήνα και φιλτράρει τα γεγονότα βάσει λιστών εξαιρέσεων. Και τα δύο συστατικά στοιχεία είναι γραμμένα σε C.

Ο Capture client χρησιμοποιεί τα προγράμματα οδήγησης πυρήνα για να παρακολουθεί το σύστημα κάνοντας χρήση του υπάρχοντος μηχανισμού ανάκλησης του πυρήνα που ειδοποιεί

τα εγγεγραμμένα προγράμματα οδήγησης όταν ένα συγκεκριμένο γεγονός λαμβάνει χώρα. Ο μηχανισμός ανάκλησης καλεί συναρτήσεις μέσα στο πρόγραμμα οδήγησης πυρήνα που περνούν τις πληροφορίες του γεγονότος στον πυρήνα ώστε ο πυρήνας να τροποποιηθεί ή να παρακολουθηθεί. Οι συναρτήσεις που καλούνται είναι οι εξής:

- CmRegistryCallback
- PsSetCreateProcessNotifyRoutine
- FilterLoad, FltRegisterFilter

Όταν γεγονότα φτάνουν στα προγράμματα οδήγησης πυρήνα, σχηματίζουν ουρά περιμένοντας να σταλθούν στο χώρο μνήμης του αντίστοιχου στοιχείου που προκάλεσε το γεγονός. Αυτό επιτυγχάνεται με το πέρασμα ενός buffer γεμάτου με δεδομένα του χρήστη από το χώρο μνήμης του χρήστη στο χώρο μνήμης του πυρήνα ώστε τα προγράμματα οδήγησης να μπορούν να γράψουν στον buffer και τελικά η εφαρμογή να επεξεργαστεί τον buffer στο χώρο μνήμης του χρήστη. Το πέρασμα αυτό πραγματοποιείται ουσιαστικά μεταξύ του Win32 API και του I/O Manager. [22]

Η τελευταία έκδοση του Capture – HPC είναι η 2.5.1-389 και ανακοινώθηκε το 2008.

6.3.3 HoneyMonkey

Το HoneyMonkey, (Strider HoneyMonkey Exploit Detection System), είναι ένα εμπορικό Honeypot που αναπτύχθηκε από τη Microsoft. Η εφαρμογή χρησιμοποιεί ένα δίκτυο υπολογιστών για να ανιχνεύσει το Διαδίκτυο ψάχνοντας για τοποθεσίες που χρησιμοποιούν ευπάθειες διακομιστών προκειμένου να εγκαταστήσουν κακόβουλο λογισμικό στον HoneyMonkey υπολογιστή. Ένα στιγμιότυπο της μνήμης, των εκτελέσιμων αρχείων και του μητρώου του Honeypot υπολογιστή καταγράφονται πριν από την ανίχνευση μιας τοποθεσίας. Μετά την επίσκεψη στην τοποθεσία, η κατάσταση της μνήμης, τα εκτελέσιμα και το μητρώο καταγράφονται και συγκρίνονται με το προηγούμενο στιγμιότυπο. Οι οποιεσδήποτε αλλαγές αναλύονται για να καθοριστεί αν η εν λόγω τοποθεσία εγκατέστησε κακόβουλο λογισμικό στον client Honeypot υπολογιστή.

Το HoneyMonkey ψάχνει για τοποθεσίες που προσπαθούν να το εκμεταλλευτούν και επιτρέπει την εύρεση τρυπών ασφαλείας που δεν είναι ήδη γνωστές αλλά χρησιμοποιούνται από εισβολείς. Ο όρος HoneyMonkey επινοήθηκε από τη Microsoft Research ομάδα το 2005. [23]

Γενική περιγραφή – λειτουργία

Το HoneyMonkey είναι ένα αυτοματοποιημένο πρόγραμμα που προσπαθεί να μιμηθεί τις ενέργειες ενός χρήστη που περιηγείται στο Διαδίκτυο. Εκτελείται συνήθως σε εικονικά μηχανήματα που εκτελούνται σε Windows XP με διάφορα επίπεδα ευπάθειας. Μπορεί ένα HoneyMonkey να είναι πλήρως ευπαθές ή πλήρως άτρωτο ή και κάτι ενδιάμεσο. Χρησιμοποιεί τον Internet Explorer για να επισκεφτεί έναν ιστότοπο. Καταγράφει κάθε ενέργεια διαβάσματος και γραψίματος στο σύστημα αρχείων και το μητρώο, διατηρώντας αρχείο των δεδομένων που συλλέχθηκαν από τον ιστότοπο και του λογισμικού που είναι ήδη εγκατεστημένο. Δεν επιτρέπει pop-up παράθυρα και δεν επιτρέπει να γίνει εγκατάσταση οποιουδήποτε λογισμικού προγράμματος. Όταν η εφαρμογή εγκαταλείψει τον ιστότοπο, τα δεδομένα που έχουν καταγραφεί αναλύονται ώστε να διαπιστωθεί αν φορτώθηκε κακόβουλο λογισμικό. Στην ουσία οποιαδήποτε λειτουργία διαβάσματος ή γραψίματος πραγματοποιηθεί εκτός του προσωρινού φακέλου του Internet Explorer θεωρείται κακόβουλη και καταγράφεται ως τέτοια. Τα δεδομένα αυτά τότε αποστέλλονται σε ένα εξωτερικό πρόγραμμα που παίζει το ρόλο του ελεγκτή που φορτώνει τα δεδομένα αυτά και επανεκκινεί το εικονικό μηχάνημα ώστε αυτό να ανιχνεύσει και άλλες τοποθεσίες.

Μία λίστα καθορίζει ποιες τοποθεσίες θα ανιχνεύσει αρχικά το HoneyMonkey. Οι τοποθεσίες αυτές είναι γνωστές για τη χρήση ευπαθειών διακομιστών. Το HoneyMonkey εν συνεχεία ακολουθώντας συνδέσεις από αυτές τις τοποθεσίες ανιχνεύει και άλλες τοποθεσίες, καθώς θεωρείται ότι ευπαθείς τοποθεσίες οδηγούν σε ευπαθείς τοποθεσίες. Καταγράφει επίσης και τον αριθμό των συνδέσεων που οδηγούν σε μία ευπαθή τοποθεσία για στατιστικούς λόγους.

Για να αναγνωρίζει τις ευπαθείς τοποθεσίες χρησιμοποιεί μαύρο κουτί και όχι για παράδειγμα υπογραφές από τις ευπάθειες των διακομιστών. [23]

6.3.4 SHELIA

Το SHELIA είναι ένα υψηλής αλληλεπίδρασης Honeypot που αναπτύχθηκε από τον Robert Joan Rocaspana στο Vrije Universiteit στο Amsterdam. Αποτελεί ένα σύστημα Ανίχνευσης Παρείσφρησης, (Intrusion Detection System – IDS), για τον client. Περιέχει έναν προσομοιωτή email client που σαρώνει την ηλεκτρονική αλληλογραφία μέσω ενός φακέλου αλληλογραφίας που καθορίζεται στη γραμμή εντολών. Συνήθως ο φάκελος αυτός μπορεί να είναι ο φάκελος ανεπιθύμητης αλληλογραφίας. Σε αυτόν το φάκελο ο προσομοιωτής email client μπορεί να ακολουθεί κάθε διεύθυνση URL και να ανοίγει κάθε επισυναπτόμενο αρχείο.

Η πρώτη έκδοση του SHELIA για Windows απαιτούσε την ύπαρξη του Outlook Express για την ανάγνωση των emails. Οι επόμενες εκδόσεις περιείχαν αλλαγές και επεκτάσεις από τους Γεώργιος Πορτοκαλίδης, Philip Homburg και Herbert Bos και το Outlook Express δεν ήταν πλέον απαραίτητο καθώς το SHELIA μπορούσε να χειριστεί το IMAP πρωτόκολλο, ύποπτες διευθύνσεις και επισυνάψεις και εν γένει οτιδήποτε πέρναγε ως είσοδος από το σύστημα βάσης δεδομένων του SHELIA. [25]

Η τελευταία έκδοση του SHELIA ανακοινώθηκε το 2009.

Γενικές έννοιες

Η βασική ιδέα που ακολουθεί το SHELIA είναι η προσομοίωση ενός αφελούς χρήστη, κάποιου δηλαδή που ακολουθεί όλες τις συνδέσεις και ανοίγει όλες τις επισυνάψεις ανεπιθύμητης αλληλογραφίας που λαμβάνονται ακόμα και με άλλους τρόπους, όπως instant messaging. Κάθε φορά που το SHELIA αναγνωρίζει την ύπαρξη κακόβουλου λογισμικού, παράγει μία προειδοποίηση.

Το SHELIA διαφέρει από τα λοιπά Honeypot συστήματα στο ότι δεν παράγει ψευδώς θετικά αποτελέσματα, (false positives), αλλά δύναται να παράξει ψευδώς αρνητικά, (false negatives). Εκ κατασκευής θεωρεί τα ψευδώς θετικά αποτελέσματα πολύ πιο σημαντικά από τα ψευδώς αρνητικά.

Χαρακτηριστικά – Πρόσφατες προσθήκες

Οι τελευταίες προσθήκες του SHELIA είναι οι εξής:

- Βελτιωμένο χειρισμό εισόδου δεδομένων. Η παλιά έκδοση του SHELIA απαιτούσε το Outlook Express για να λάβει μηνύματα ανεπιθύμητης αλληλογραφίας από έναν POP διακομιστή. Στην τελευταία έκδοση, τα δεδομένα εισόδου ανακτώνται από μία βάση δεδομένων. Οποιαδήποτε τεχνική που μπορεί να γεμίσει τη βάση του SHELIA, είναι συμβατή με το SHELIA. Για παράδειγμα, ένας IMAP mail client φιλτράρει διευθύνσεις URL και επισυνάψεις και τις ταξινομεί και τις εισάγει σε μία βάση δεδομένων. Στη βάση τα δεδομένα αυτά εξετάζονται με προτεραιότητα. Αν δεν έχει οριστεί προτεραιότητα, χρησιμοποιείται μία στατική μέθοδος απονομής προτεραιοτήτων που τίθεται από το σύστημα, επιτρέποντας έτσι την εισαγωγή αντικειμένων που χρήζουν επείγουσας αντιμετώπισης στο SHELIA χωρίς να χρειάζεται αναμονή για την καταγραφή τους.
- Καλύτερος έλεγχος και ασφάλεια. Το SHELIA εκτελείται σε ένα εικονικό μηχάνημα και επανεκκινείται μετά από ν ελέγχους ώστε να αποφευχθούν οι επιθέσεις που δεν μπορούν να ανιχνευθούν από αυτό. Επιπλέον κάθε έλεγχος ματαιώνεται μετά από ν δευτερόλεπτα.
- Δύσκολη πρόσβαση. Η αλληλεπίδραση μεταξύ εικονικού και πραγματικού μηχανήματος έχει σχεδιαστεί και διαμορφωθεί με τέτοιο τρόπο ώστε εισβολείς να μην μπορούν να επιτύχουν την κατάρρευση του συστήματος.
- Χρήση συναγερμού στη βάση δεδομένων και web διεπαφής χρήστη. Όποτε το SHELIA διαπιστώσει μία ευπάθεια, πραγματοποιεί εκτεταμένη ανάλυση (ποιες API κλήσεις κάνει ο κώδικας, ποιος είναι ο φόρτος κάθε επίθεσης κ.α.) και αποθηκεύει αυτή την ανάλυση με μία συγκεκριμένη δομή στη βάση δεδομένων. Στην προηγούμενη έκδοση δεν

υπήρχε συγκεκριμένη δομή, απλά καταγραφόταν η ανάλυση αυτή στη βάση. Επιπλέον, ο κάθε χρήστης με τις σωστές πιστοποιήσεις μπορεί να έχει πρόσβαση στη βάση δεδομένων μέσω της web διεπαφής χρήστη. [25]

6.3.5 UW Spycrawler

Το Spycrawler που αναπτύχθηκε στο Πανεπιστήμιο της Ουάσιγκτον είναι ένα ακόμη Honeypot υψηλής αλληλεπίδρασης που βασίζεται στο Mozilla πρόγραμμα περιήγησης και αναπτύχθηκε από τον Moshchuk και την ερευνητική του ομάδα το 2005. Δεν είναι διαθέσιμο για download. Εντοπίζει επιθέσεις παρακολουθώντας τα αρχεία, τις διαδικασίες, τις εγγραφές, και τις διακοπές του προγράμματος περιήγησης. Ο μηχανισμός ανίχνευσης βασίζεται στα γεγονότα. Επιπλέον, αυξάνει το χρόνο εκτέλεσης της εικονικής μηχανής που διαθέτει το Spycrawler ώστε να ξεπεράσει ή να μειώσει τις επιπτώσεις των ωρολογιακών βομβών στο σύστημα. [26]

6.3.6 Web Exploit Finder

Το Web Exploit Finder (WEF) είναι ένα υψηλής αλληλεπίδρασης Client Honeypot που σχεδιάστηκε και αναπτύχθηκε από τους Thomas Müller, Benjamin Mack και Mehmet Arziman στο Stuttgard Media University (HdM) το 2006. Ιδιαίτερο χαρακτηριστικό του είναι ότι έχει υλοποιηθεί έτσι ώστε να ανιχνεύει επιθέσεις αξιολογώντας τις μεταβολές του λειτουργικού συστήματος. Ο μηχανισμός ελέγχου συνίσταται στην παρακολούθηση και την εξέταση των κλήσεων του συστήματος. Το WEF μπορεί να αποτελέσει ουσιαστικά ένα Honeynet στην αρχιτεκτονική του οποίου συμπεριλαμβάνεται ένα εικονικό επίπεδο, όπου οι επιτιθέμενοι ενεργούν χωρίς να παραβιαστεί πραγματικά το client Honeypot. [27]

Αρχιτεκτονική

Η αρχιτεκτονική του συστήματος αποτελείται από τα εξής μέρη:

- Ένα επίπεδο εικονικοποίησης, (virtualization layer), που χρησιμοποιεί τον VMware server για την προστασία του συστήματος και την προσπέλαση πολλαπλών σελίδων ταυτόχρονα.
- Ένα εξειδικευμένο επίπεδο που λειτουργεί σαν κονσόλα εργαλείων, (rootkit), και τροποποιεί το λειτουργικό σύστημα και ανιχνεύει τις σελίδες κακόβουλου λογισμικού.
- Έναν ελεγκτή για το πρόγραμμα περιήγησης, (browser control – BC), που ελέγχει την κονσόλα εργαλείων και τον Internet Explorer και επικοινωνεί με την κονσόλα επικοινωνίας.
- Μία κονσόλα ελέγχου, (management console – MC), που παραμετροποιεί και ελέγχει όλο το σύστημα.

Στην εικόνα 6-9 διακρίνεται η αρχιτεκτονική του Web Exploit Finder.

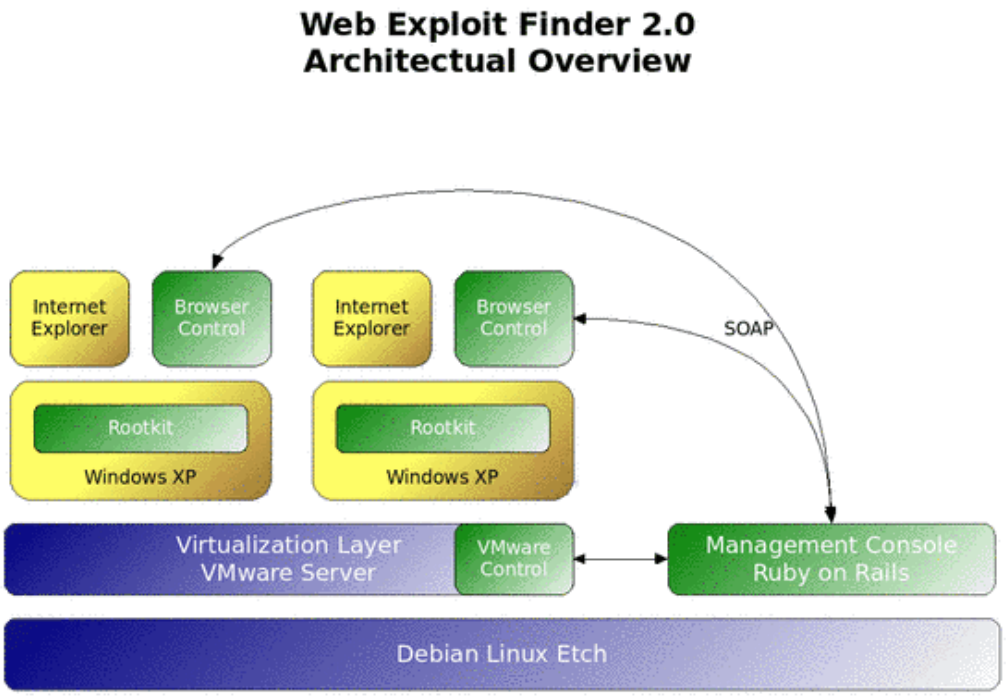
Λειτουργία

Οι χρήστες μπορούν να ελέγχουν και να παρακολουθούν το σύστημα μέσω της web διεπαφής της κονσόλας ελέγχου. Για να ξεκινήσει το WEF τη λειτουργία του ο χρήστης πρέπει να εισάγει μία λίστα από διευθύνσεις URL. Σε επόμενες εκδόσεις, αυτό δεν είναι απαραίτητο γιατί χρησιμοποιείται ένας web ανιχνευτής που εξάγει διευθύνσεις URL που είναι ήδη συνδεδεμένες σε ιστοσελίδες που χαρακτηρίζονται ως κακόβουλες.

Προκειμένου να αποφευχθεί να τεθεί σε κίνδυνο η ασφάλεια του συστήματος ή να διακόπτεται η απρόσκοπτη λειτουργία του συστήματος με συνεχείς επανεγκαταστάσεις, το σύστημα επισκέπτεται ιστοσελίδες μέσα σε ένα sandbox. Το sandbox αποτελεί εικόνα του λειτουργικού συστήματος που φιλοξενείται στο εικονικό μηχάνημα του VMware server. Έχει κλωνοποιηθεί και παραμετροποιηθεί με χρήση εσωτερικών scripts και του VMware C-API. Η διαδικασία αυτή μπορεί να επαναληφθεί ν φορές, ανάλογα με το πόσα στιγμιότυπα Windows XP χρησιμοποιούνται για να ελέγχουν διευθύνσεις URL ταυτόχρονα. Αυτό εξαρτάται επίσης και από την απόδοση του VMware συστήματος. Για την προστασία της διαδικασίας κλωνοποίησης,

τα scripts ειδοποιούν την κονσόλα ελέγχου για την κατάσταση του sandbox, γεγονός που φαίνεται και στη web διεπαφή.

Τα sandbox στιγμιότυπα εγγράφονται και εκκινούν τη λειτουργία τους μέσω του VMware server. Μόλις ολοκληρωθεί η διαδικασία εκκίνησης, ένα επιπλέον script δημιουργεί ένα στιγμιότυπο της τρέχουσας κατάστασης, αντιγράφει την πιο πρόσφατη έκδοση του browser control καθώς και του rootkit στο sandbox και εκτελεί τον browser control. Με χρήση αυτής της τεχνικής στιγμιότυπου, το σύστημα μπορεί να κάνει επαναφορά σε προηγούμενο στιγμιότυπο οποτεδήποτε το σύστημα προσβληθεί από κακόβουλο λογισμικό. Η διαδικασία αυτή διαρκεί μόλις λίγα δευτερόλεπτα σε αντίθεση με τη διαγραφή του sandbox και τη δημιουργία ενός καινούριου. [27]



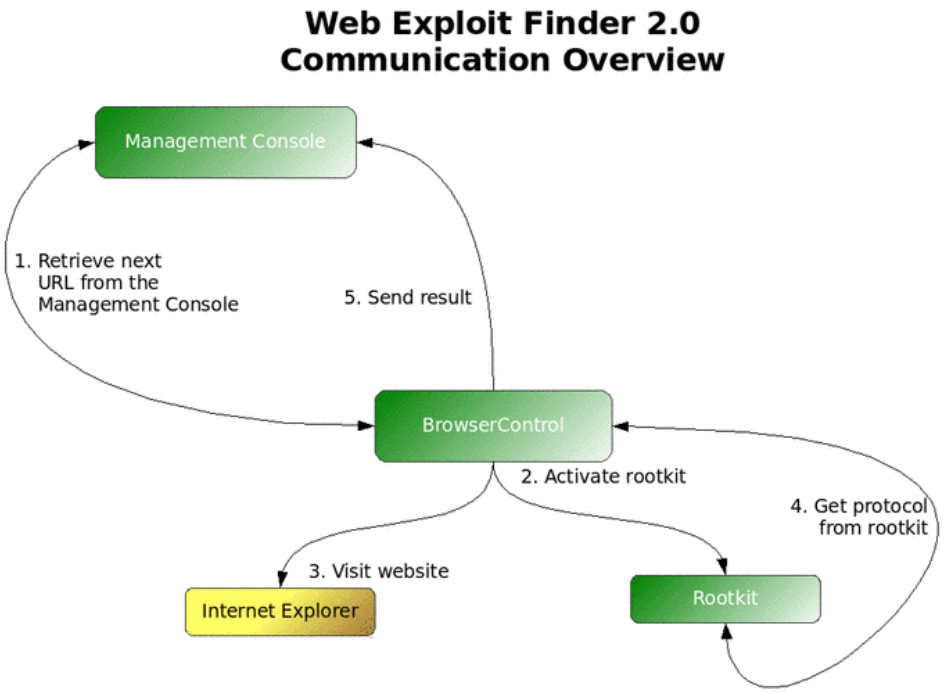
Εικόνα 6-9: Αρχιτεκτονική Web Exploit Finder. [27]

Ο browser control φορτώνει το rootkit σαν πρόγραμμα οδήγησης των Windows στα Windows XP. Στη συνέχεια καταχωρεί τον εαυτό του στο sandbox και εμφανίζει την IP διεύθυνση στην κονσόλα ελέγχου. Στη συνέχεια τα συστήματα διακλαδώνονται σύμφωνα με τα παρακάτω βήματα.

1. BC ζητάει από τον MC την επόμενη διεύθυνση URL.
2. BC στέλνει στο rootkit ένα μήνυμα για να ενεργοποιηθεί η κονσόλα του συστήματος. Όλες οι περαιτέρω σχετικές κλήσεις συστήματος, (CreateFile, DeleteFile, Execute κ.α.), ανακατευθύνονται και παρακολουθούνται.
3. BC ξεκινά τον Internet Explorer και του προσφέρει τη URL διεύθυνση που λαμβάνει.
4. Άπαξ και φορτωθεί ολόκληρη η σελίδα ή αν ληφθεί μήνυμα για την εξάντληση του χρονικού ορίου, ο Internet Explorer παύει τη λειτουργία του και η παρακολούθηση σταματά.
5. Τώρα ο BC ζητά από το rootkit μία λίστα αποτελεσμάτων. Αν η σελίδα ήταν «καθαρή», η λίστα θα είναι άδεια, αλλιώς η λίστα θα περιέχει όλες τις ύποπτες κλήσεις συστήματος με πληροφορίες ημέρας και ώρας, καθώς και ένα αναγνωριστικό διαδικασίας, (PID), της αντίστοιχης εφαρμογής.
6. Αν η λίστα είναι άδεια, ο BC ενημερώνει τον MC, ο οποίος χαρακτηρίζει ως «καθαρή» την εν λόγω διεύθυνση στη βάση δεδομένων ενώ αν η λίστα περιέχει διευθύνσεις, οι

διευθύνσεις αυτές μεταφέρονται στον MC. Αν το sandbox χαρακτηριστεί ότι περιέχει κακόβουλο λογισμικό, καλείται ένα script που επαναφέρει το σύστημα στο αρχικό του στιγμιότυπο. Μετά την αντιγραφή του BC και του rootkit και την επανακαταχώρησή του με τον MC, το σύστημα ξεκινά πάλι από το βήμα 1.

Η ανωτέρω διαδικασία που περιγράφεται αναλυτικά, παρουσιάζεται στην εικόνα 6-10.



Εικόνα 6-10: Λειτουργία Web Exploit Finder. [27]

6.3.7 High Interaction Honeypot Analysis Toolkit (HiHAT)

Το High Interaction Honeypot Analysis Toolkit (HiHAT) επιτρέπει την μετατροπή PHP εφαρμογών σε web based Honeypots υψηλής αλληλεπίδρασης. Επιπλέον μέσω μίας γραφικής διεπαφής χρήστη που παρέχει, υποστηρίζει τις διαδικασίες παρακολούθησης του Honeypot και της ανάλυσης των δεδομένων που αποκτήθηκαν.

Μια τυπική χρήση του HiHAT αποτελεί η μετατροπή του PHPNuke, του PHPMyAdmin και του OSCommerce σε πλήρως λειτουργικό Honeypot, που προσφέρει την πλήρη λειτουργικότητα της εφαρμογής στους χρήστες, αλλά ταυτόχρονα στο παρασκήνιο εκτελεί πλήρη καταγραφή και έλεγχο.

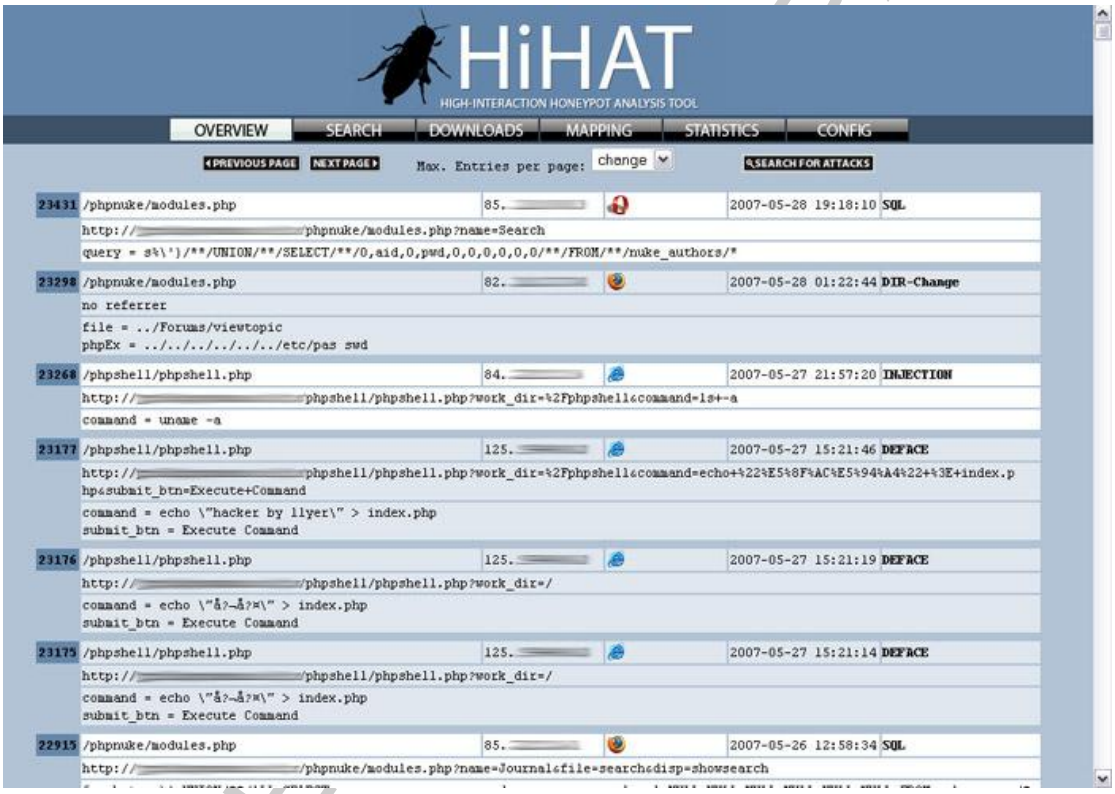
Τα κυριότερα χαρακτηριστικά του HiHAT είναι τα εξής:

- Σαρώνει αυτόματα για γνωστές επιθέσεις.
- Ανιχνεύει SQL-Injections, (απομακρυσμένη) συμπερίληψη αρχείων, cross-site scripting (XSS), προσπάθειες για λήψη αρχείων με περιεχόμενο κακόβουλου τύπου π.χ. μέσω χρήσης εντολής wget ή curl κ.α.
- Παρέχει μία λειτουργία επισκόπησης που επιτρέπει την αναζήτηση και το σάρωμα για νέα περιστατικά γρήγορα. (ημιαυτόματη λειτουργία)
- Υποστηρίζει λεπτομερή πληροφόρηση σχετικά με όλα τα δεδομένα που σχετίζονται με οτιδήποτε έχει πρόσβαση στο Honeypot. Αυτό περιλαμβάνει αλλά δεν περιορίζεται στα δεδομένα των HTTP-GET, HTTP-POST και COOKIE.

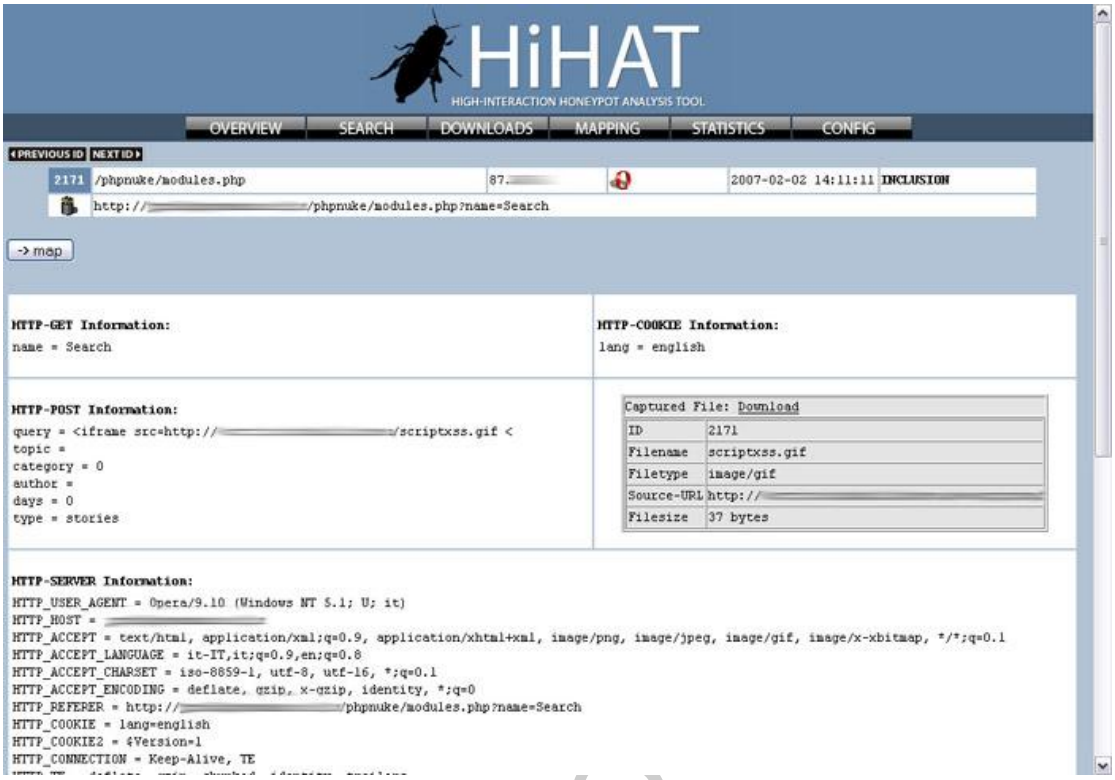
- Αποθηκεύει αντίγραφα των κακόβουλων εργαλείων σε ένα ασφαλές απομονωμένο μέρος για μελλοντική ανάλυση.
- Παρέχει μία γεωγραφική χαρτογράφηση βάσει των IP διευθύνσεων για τις πηγές της επίθεσης. Ο παραγόμενος χάρτης δείχνει την προέλευση των επιθέσεων και προσφέρει επιπλέον λεπτομέρειες για κάθε θέση.
- Δημιουργεί στατιστικά στοιχεία για όλη την κυκλοφορία που αναγνωρίζεται στο σύστημα.

Η πρώτη έκδοση του εργαλείου αυτού δημοσιεύτηκε τον Ιούλιο του 2007 ενώ το Σεπτέμβριο του ίδιου έτους έγινε η τελευταία προσθήκη. [36]

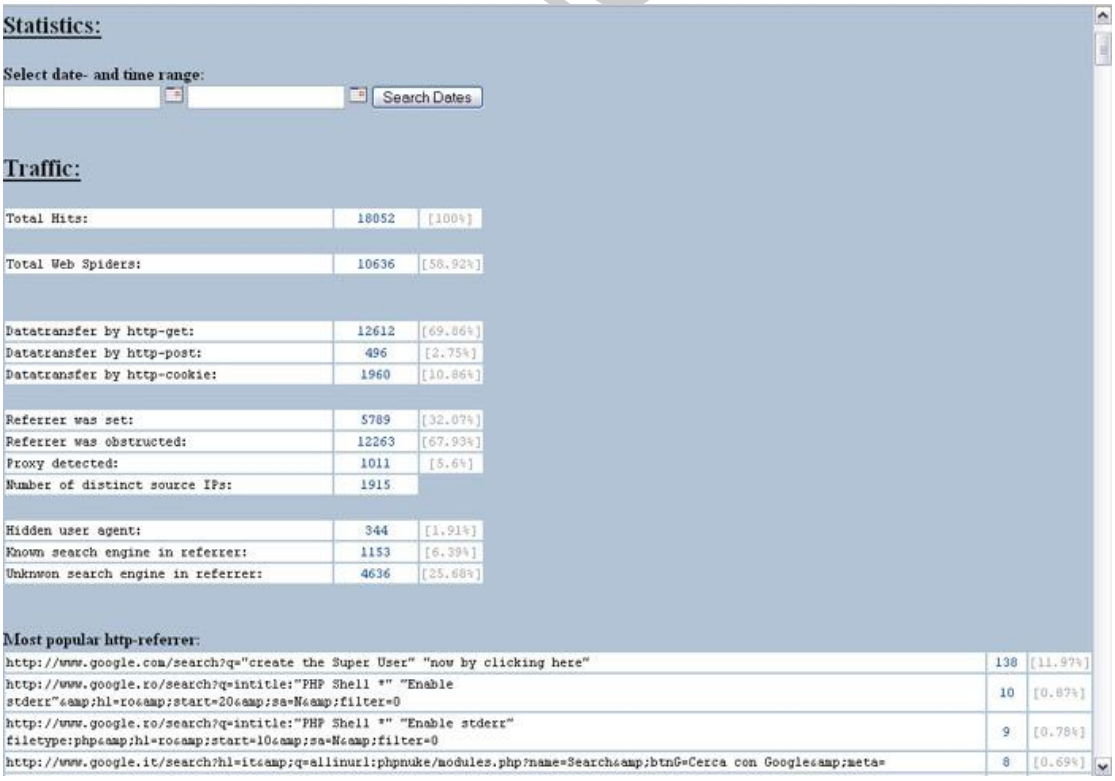
Ακολουθούν ορισμένα στιγμιότυπα της λειτουργίας του HiHAT. Η εικόνα 6-11 παρουσιάζει το HiHAT σε λειτουργία επισκόπησης, η εικόνα 6-12 δείχνει μια λεπτομερέστερη προβολή λειτουργίας επισκόπησης και η εικόνα 6-13 παρουσιάζει στατιστικά στοιχεία χρήσης HiHAT.



Εικόνα 6-11: HiHAT σε λειτουργία επισκόπησης. [36]



Εικόνα 6-12: Λεπτομερής προβολή λειτουργίας επισκόπησης HiHAT. [36]



Εικόνα 6-13: Στατιστικά στοιχεία HiHAT. [36]

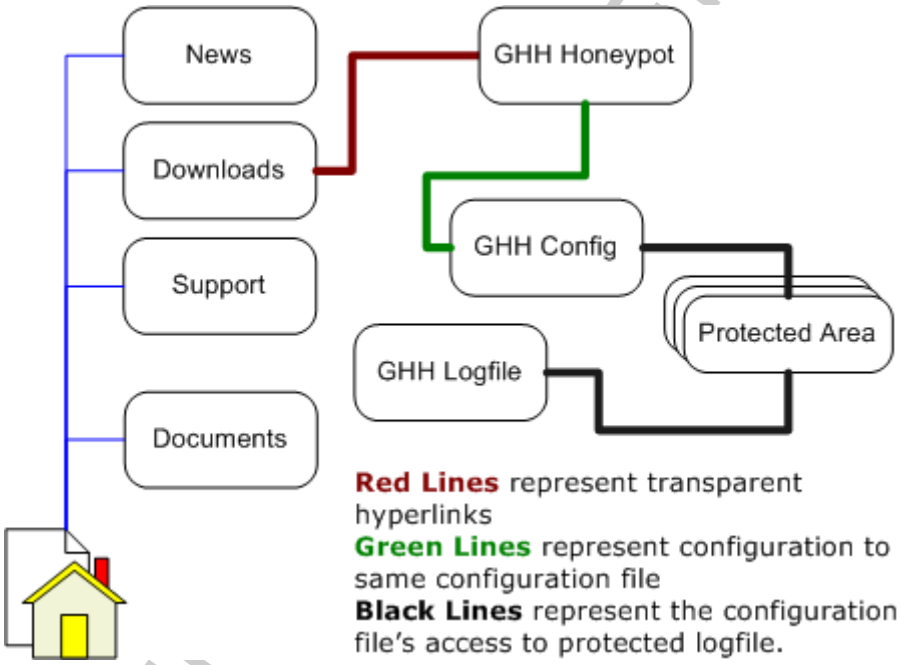
6.3.8 Google Hack Honeypot

Το Google Hack Honeypot, (GHH), αναπτύχθηκε από την Google το 2005 για την αντιμετώπιση της διαδικτυακής κίνησης που θεωρείται κακόβουλου τύπου, των hackers δηλαδή που κάνουν χρήση μηχανών αναζήτησης. Έχει σχεδιαστεί έτσι ώστε να παρέχει αναγνώριση των εισβολέων που χρησιμοποιούν τις μηχανές αναζήτησης ως βασικό εργαλείο για τις επιθέσεις τους. Υποστηρίζεται από τη μηχανή αναζήτησης της Google και τη Google Hacking Database (GHDB) η οποία διατηρείται από την κοινότητα johnny.ihackstuff.com.

Η ολοένα αυξανόμενη χρήση της Google αλλά και web-based εφαρμογών όπως message boards, υπηρεσίες instant messaging και απομακρυσμένη διαχείριση εργαλείων και εφαρμογών οδήγησε στην αύξηση των μη σωστά παραμετροποιημένων προγραμμάτων κι εφαρμογών στο διαδίκτυο και των ευπαθειών σε αυτά. Οι επισφαλείς αυτές λειτουργίες όταν συνδυαστούν με τις δυνατότητες που παρέχει η μηχανή αναζήτησης της Google, γίνονται ένας βολικός μηχανισμός εναντίον κακόβουλων χρηστών.

Η πρώτη έκδοση του GHH έγινε διαθέσιμη στις 13 Φεβρουαρίου 2005**.** Η τελευταία έκδοση ανακοινώθηκε στις 23 Φεβρουαρίου του 2007.

Έχει γραφτεί σε php και διανέμεται υπό τη γενική άδεια της GNU.



Εικόνα 6-14: Αρχιτεκτονική Google Hack Honeypot. [37]

Λειτουργία

Το GHH προσομοιώνει μία ευπαθή web εφαρμογή και της επιτρέπει να ευρετηριοποιηθεί από άλλες μηχανές αναζήτησης. Η εφαρμογή είναι κρυφή από τους κλασσικούς χρήστες διαδικτύου καθώς χρησιμοποιεί ένα διάφανο link αλλά μπορεί να βρεθεί μέσω της χρήσης ανιχνευτή μηχανών αναζήτησης αφού επιτρέπεται η ευρετηριοποίηση. Η χρήση του διάφανου link μειώνει τα ψευδώς θετικά αποτελέσματα και αποτρέπει το αποτύπωμα του Honeypot.

Το Honeypot συνδέεται με ένα αρχείο ρύθμισης παραμέτρων και το αρχείο ρύθμισης παραμέτρων γράφει σε ένα αρχείο καταγραφής, το οποίο επιλέγεται κατά την παραμετροποίηση. Το αρχείο καταγραφής περιέχει πληροφορίες για τον φορέα υποδοχής, την IP διεύθυνση προέλευσης και εν γένει οποιαδήποτε σχετική πληροφορία.

Χρησιμοποιώντας τα δεδομένα του αρχείου καταγραφής, ένας διαχειριστής μπορεί να πάρει αρκετές πληροφορίες για τους εισβολείς που δρουν κατά του ιστοτόπου του, ενώ διασταυρώνοντας πολλαπλά δεδομένα από διάφορους ιστοτόπους μπορεί να σκιαγραφήσει το προφίλ συγκεκριμένων εισβολέων. [37]

Η ανωτέρω εικόνα 6-14 αναδεικνύει τη γενική αρχιτεκτονική Google Hack Honeypot.

6.3.9 HonSSH

Το HonSSH είναι ένα Honeypot υψηλής αλληλεπίδρασης το οποίο εγκαθίσταται μεταξύ ενός εισβολέα και ενός Honeypot, δημιουργώντας δύο ξεχωριστές SSH συνδέσεις μεταξύ τους. Έχει βασιστεί στο μεσαίας αλληλεπίδρασης Honeypot Kippo και χρησιμοποιεί τους μηχανισμούς του καταγραφής επιθέσεων και αλληλεπίδρασης.

Αναπτύχθηκε από τον Thomas Nicholson τον Αύγουστο του 2013. Έχει γραφτεί σε Python και είναι ένα project ανοιχτού κώδικα το οποίο γίνεται host στο Google Code από τον συγγραφέα του.

Λειτουργικότητα

Την πρώτη φορά που θα εκτελεστεί το HonSSH, θα εγκαθιδρύσει μία σύνδεση με το Honeypot, θα λάβει την έκδοση του SSH διακομιστή του Honeypot, θα την αποθηκεύσει και θα αποσυνδεθεί. Όταν ο εισβολέας συνδεθεί στο HonSSH, το HonSSH θα εγκαθιδρύσει μία SSH σύνδεση με τον εισβολέα και μία ξεχωριστή με το Honeypot. Αφού έχει προηγηθεί η προηγούμενη σύνδεση μεταξύ Honeypot και HonSSH, όλα τα δεδομένα που μεταφέρονται από τον εισβολέα προς το HonSSH, μεταφέρονται και στο Honeypot. Απαιτείται η απενεργοποίηση αυθεντικοποίησης μέσω χρήσης δημοσίων κλειδιών στον SSH διακομιστή του Honeypot ενώ ενδείκνυται η χρήση των ίδιων κλειδιών κρυπτογράφησης από το Honeypot και από το HonSSH για τη σωστή λειτουργία του HonSSH.

Το μηχάνημα που φιλοξενεί το HonSSH παρέχει υποστήριξη NAT, (Network Address Translation), και χρησιμοποιεί τείχος προστασίας.

Κύρια χαρακτηριστικά

- Το HonSSH καταγράφει όλες τις προσπάθειες σύνδεσης σε ένα αρχείο κειμένου.
- Όταν ένας εισβολέας στείλει έναν κωδικό που έχει προβλέψει, το HonSSH αυτόματα τον αντικαθιστά με το σωστό κωδικό. Αυτό επιτρέπει στους εισβολείς να αποκτούν πρόσβαση στο μηχάνημα με οποιονδήποτε κωδικό αλλά τους προκαλεί σύγχυση όταν χρησιμοποιούν την εντολή sudo στο μηχάνημα με τον ίδιο κωδικό και αποτυγχάνουν.
- Όλη η αλληλεπίδραση καταγράφεται σε μία συνεδρία TTY, καταγράφεται δηλαδή σε δυαδική μορφή, όπως ακριβώς κάνει και το Kippo, και η συνεδρία μπορεί να αναπαραχθεί με τη χρήση του ttylog εργαλείου που περιλαμβάνεται στο Kippo.
- Μία γραπτή περίληψη κάθε συνεδρίας του εισβολέα καταγράφεται σε ένα αρχείο κειμένου.
- Όλες οι συνεδρίες μπορούν να διαβαστούν σε πραγματικό χρόνο, όπως ακριβώς και στο Kippo, με χρήση του πρωτοκόλλου telnet.

Ουσιαστικά, το HonSSH καταγράφει όλες τις SSH συνδέσεις που πραγματοποιούνται μεταξύ πελάτη και διακομιστή. Προσφέρει περισσότερη ασφάλεια και παραπάνω λειτουργικότητα από το Kippo, αφού πλέον μεταξύ του εισβολέα και του Honeypot παρεμβάλλεται το HonSSH. [56]

6.3.10 HoneyDrive

Το HoneyDrive είναι ένας εικονικός σκληρός δίσκος με λειτουργικό XUbuntu Desktop 12.04 32-bit εγκατεστημένο. Περιέχει διάφορα Honeypot πακέτα λογισμικού όπως Kippo, Dionaea, Honeyd, Glastopf, LaBrea sticky, Thug, Tiny honeypot, Kojoney, Amun, mwcrawler και Wordpot καθώς και χρήσιμα scripts και εργαλεία για την ανάλυση, οπτικοποίηση και επεξεργασία των δεδομένων που καταγράφει, όπως Kippo-Graph, Honeyd-Viz, Dionaea-FR. Περιλαμβάνει τέλος και άλλα βοηθητικά εργαλεία όπως nmap, Wireshark, ClamAV και άλλα.

Για την εκτέλεση των Honeypots που περιέχει το HoneyDrive, απαιτείται ελάχιστη ή και καθόλου παραμετροποίηση, αφού είναι ήδη προεγκατεστημένα.

Το κίνητρο για την ανάπτυξη του HoneyDrive ήρθε σαν αποτέλεσμα της δυσκολίας παραμετροποίησης κι εγκατάστασης πολλών άλλων Honeypots. Το HoneyDrive συγκεντρώνει πληθώρα Honeypots σε έναν μόνο εικονικό δίσκο που απαιτεί την εγκατάστασή του σε ένα εικονικό μηχάνημα για να λειτουργήσει.

Η έκδοση 0.1 του Honeybox αναπτύχθηκε από τον Ιωάννη Κονιάρη και διατέθηκε ως ελεύθερο λογισμικό στις 11.09.2012 στην προσωπική του ιστοσελίδα bruteforce.gr. Στις 26.10.2012 μετονομάστηκε από Honeybox που ήταν η αρχική του ονομασία σε HoneyDrive. [28]

Τα Kippo-Graph και Honeyd-Viz αναπτύχθηκαν ομοίως από τον ίδιο. Τα συγκεκριμένα εργαλεία είναι εγκατεστήμενα στη σουίτα του HoneyDrive. [55]

6.3.11 Σύνοψη

Honeypot	Χρόνος	Client	Server	Λειτουργικότητα
ManTrap	2005			Εμπορικό Honeypot. Πλήρες λειτουργικό σύστημα – χρήση εικονικών υποσυστημάτων
Capture – HPC	2008	✓		Εικονικό μηχάνημα, αλληλεπιδρά με servers ψάχνοντας κακόβουλους.
HoneyMonkey	2005	✓	✓	Microsoft –χρήση εικονικού δικτύου υπολογιστών για ανίχνευση στο Διαδίκτυο τοποθεσιών που χρησιμοποιούν server ευπάθειες
SHELIA	2009			IDS – Προσομοιώνει email client, (έλεγχος spam, instant messaging, επισυνάψεων).
UW Spycrawler	2005	✓	✓	Εικονικό – χρήση Mozilla Firefox για εύρεση web ευπαθών διευθύνσεων
Web Exploit Finder	2006	✓		Honeynet – χρήση εικονικού επιπέδου και web διεπαφής, εξετάζει λίστα διευθύνσεων URL για εύρεση web ευπαθειών
High Interaction Honeypot Analysis Toolkit, (HiHAT)	2007	✓	✓	Μετατροπή PHP εφαρμογών σε web based Honeypots, χρήση GUI για παρακολούθηση και ανάλυση δεδομένων, ανίχνευση για web ευπάθειες.

Google Hack Honeypot	2005	✓	✓	Προσομοίωση ευπαθούς web εφαρμογής η οποία ευρετηριοποιείται από άλλες μηχανές αναζήτησης – ανίχνευση browser hacking
HonSSH	2013			Ενδιάμεσο Honeypot μεταξύ εισβολέα και άλλου Honeypot, εγκαθιδρύει δύο SSH συνδέσεις, διαθέτει τείχος προστασίας. Βασίστηκε στο Kippo Honeypot.
HoneyDrive	2012			Εικονικός σκληρός δίσκος, διαθέτει προεγκατεστημένα 10 Honeypots καθώς και εργαλεία επεξεργασίας και ανάλυσης δεδομένων.

Πίνακας 6-3: Συνοπτική παρουσίαση Honeypots υψηλής αλληλεπίδρασης.

7. Προηγμενα Honeypots

7.1 Honeynets

Τα Honeynets αποτελούν ουσιαστικά ένα σύνολο από Honeypots υψηλής αλληλεπίδρασης. Εν γένει αποτελούν πολύπλοκα συστήματα που αποτελούνται από πολλαπλά Honeypots, IDS στοιχεία και στοιχεία τείχους προστασίας.

Η ανάγκη ύπαρξης ενός Honeypot που προσφέρει υψηλό επίπεδο αλληλεπίδρασης οδήγησε στη δημιουργία του Honeynet. Το Honeynet χρησιμοποιεί πραγματικά συστήματα παραγωγής, τα οποία υπόκεινται σε έλεγχο πρόσβασης. Οι εισβολείς μπορούν να επιτεθούν, να συλλέξουν, να εκμεταλλευτούν οποιοδήποτε σύστημα που βρίσκεται εντός του Honeynet. Κανένα σύστημα και καμία λειτουργία του Honeynet δεν έχει προσομοιωθεί.

Τα Honeynets αποτελούν σχετικά καινούρια έννοια στον κόσμο των Honeypots. Τα Honeypots πρωτοεμφανίστηκαν το 1990 ενώ η ανάπτυξη των Honeynets ξεκίνησε το 1999, όταν πρωτοεμφανίστηκε ο όρος Honeynets στην αναφορά «To Build a Honeypot» από τον Lance Spitzner. Η βασική ιδέα ήταν η συγκέντρωση πραγματικών συστημάτων παραγωγής πίσω από τείχος προστασίας και η παρακολούθηση του όλου συστήματος. [29]

Τα Honeynets επιτρέπουν την προσομοίωση πραγματικών περιβαλλόντων παραγωγής στο κόστος μεγαλύτερων διοικητικών και τεχνικών δαπανών. Τα αρχεία καταγραφής των Honeynets για παράδειγμα, είναι πολύ πιο δύσκολο να ερμηνευθούν σε σύγκριση με την έξοδο από ένα Honeypot. Ακόμα και σενάρια εκτός πραγματικότητας μπορούν να συμβούν, με διάφορα στοιχεία να γίνονται αντικείμενα επίθεσης, να σχετίζονται και να χρησιμοποιούνται με λάθος τρόπο από τρίτους.

Αποτελούν ένα εξαιρετικά ευέλικτο εργαλείο καθώς μπορούν να πληρώσουν πολλούς Honeypot ρόλους, να προστατεύσουν πραγματικά συστήματα και να εξαπατήσουν τους επιτιθέμενους αφού οι επιτιθέμενοι δυσκολεύονται αρκετά να αναγνωρίσουν ότι επιτίθενται σε Honeypot. Τα Honeynets χρησιμοποιούνται και για να ανιχνεύουν επιθέσεις χρησιμοποιώντας μάλιστα το μοντέλο του ManTrap, παρακολουθώντας δηλαδή την κίνηση σε κάθε θύρα και καταγράφοντας IP διευθύνσεις. Αποτελούν μία εξαιρετική λύση δε για την αντιμετώπιση επιθέσεων καθώς μπορούν να εκτελέσουν οποιοδήποτε λειτουργικό σύστημα και εφαρμογή.

Από τη στιγμή που τα Honeynets εμπεριέχουν υψηλής αλληλεπίδρασης Honeypots, χαρακτηρίζονται και αυτά ως υψηλής αλληλεπίδρασης. Επομένως, το ρίσκο που παίρνει ο χρήστης με τα Honeynets είναι σημαντικά μεγαλύτερο από το ρίσκο χρήσης ενός μόνο Honeypot. Ωστόσο, οι παρεχόμενες δυνατότητες και μέθοδοι είναι αρκετά πιο προηγμένες και υπάρχουν διάφοροι τρόποι για ελαχιστοποίηση των κινδύνων, όπως για παράδειγμα κάνοντας χρήση περιορισμού κίνησης στοιχείων τείχους προστασίας κατά την αρχική παραμετροποίηση.

Αν και τα Honeynets προσαρμόζονται εύκολα ως Honeypots παραγωγής, σπάνια χρησιμοποιούνται με αυτόν τον τρόπο επειδή είναι αρκετά πολύπλοκα. Απαιτούν πολύ χρόνο και πολλούς πόρους για να δημιουργηθούν, να λειτουργήσουν και να διατηρηθούν. Η προσπάθεια που απαιτούν οι Honeynet τεχνολογίες ως Honeypots παραγωγής συχνά δεν αξίζει τα αποτελέσματα. Ναι μεν προλαμβάνουν, ανιχνεύουν και αντιδρούν στις επιθέσεις αλλά το ίδιο ακριβώς κάνουν και απλά Honeypots. [9]

Αρχιτεκτονική Honeynet

Ο ρόλος – κλειδί σε ένα Honeynet είναι το Honeywall. Το Honeywall λειτουργεί σαν δρομολογητής μεταξύ των Honeypots και των μηχανημάτων παραγωγής. Οποιαδήποτε κίνηση προς ή από τα Honeypots περνάει από το Honeywall. Ουσιαστικά η πύλη αυτή επιτρέπει την εισερχόμενη κίνηση στα συστήματα, αλλά ελέγχει την εξερχόμενη κίνηση με τη χρήση τεχνολογιών για την πρόληψη εισβολής. Αυτό δίνει στον εισβολέα την ευελιξία να αλληλεπιδρά με τα συστήματα αλλά τα εμποδίζει από το να βλάψουν άλλους υπολογιστές.

Το Honeywall είναι αόρατο σε οποιονδήποτε αλληλεπιδρά με τα Honeypots και έχει τρία επίπεδα. Τα δύο πρώτα, (eth0 και eth1), διαχωρίζουν τα Honeypots από οτιδήποτε άλλο, δηλαδή είναι γεφυρωμένες διασυνδέσεις που δεν έχουν στοίβα IP και το τρίτο επίπεδο, (eth2), διαθέτει μία IP στοίβα που επιτρέπει την απομακρυσμένη διαχείριση.

Υπάρχουν αρκετές βασικές απαιτήσεις που πρέπει να ικανοποιεί ένα Honeywall: έλεγχος δεδομένων – Data Control, σύλληψη δεδομένων – Data Capture, ανάλυση δεδομένων – Data Analysis και συλλογή δεδομένων – Data Collection. Ο έλεγχος δεδομένων καθορίζει πώς η δραστηριότητα κατανέμεται στο Honeynet χωρίς ο εισβολέας να το γνωρίζει. Σκοπός του είναι να ελαχιστοποιηθεί ο κίνδυνος. Η σύλληψη δεδομένων καταγράφει τη συνολική δραστηριότητα του εισβολέα χωρίς ο εισβολέας να το γνωρίζει. Η ανάλυση δεδομένων αναλύει τα δεδομένα και η συλλογή δεδομένων συγκεντρώνει τα δεδομένα των Honeypots του Honeynet σε έναν μόνο πόρο. Από αυτές τις απαιτήσεις, ο έλεγχος δεδομένων είναι η πιο σημαντική καθώς σκοπό έχει τον περιορισμό του κινδύνου.

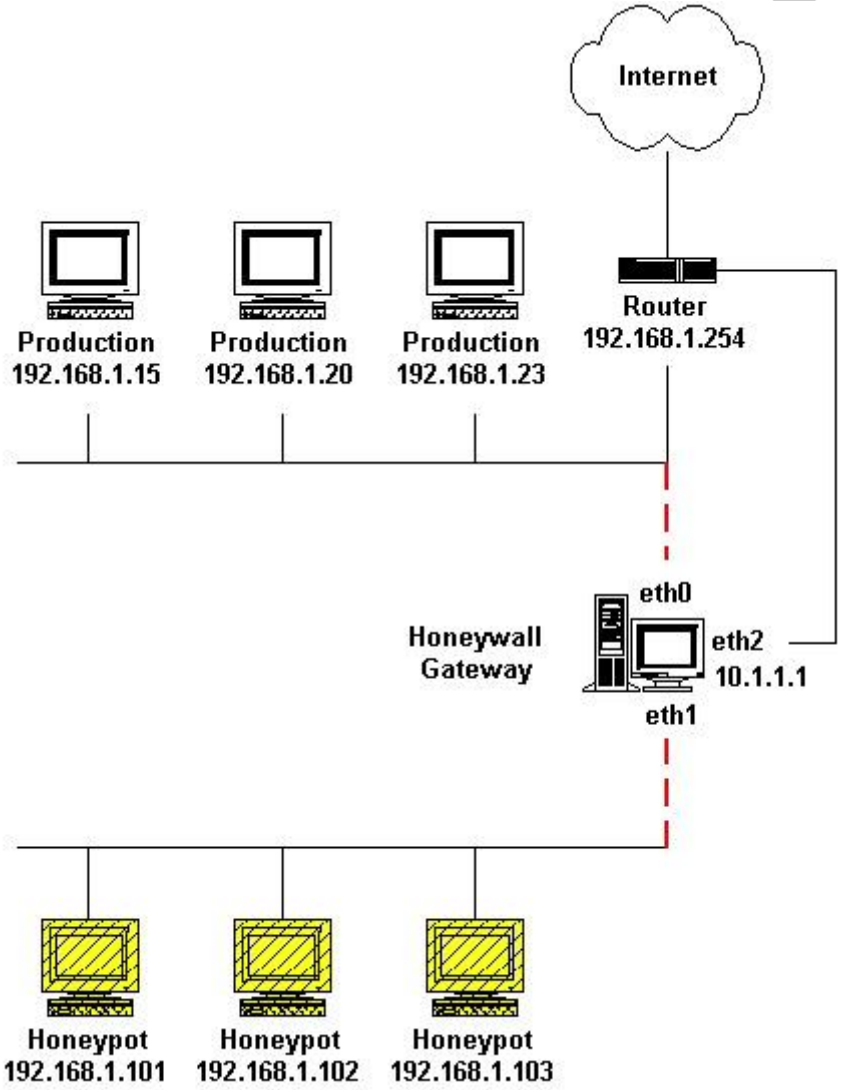
Αναλυτικότερα:

- Data Control: Ο έλεγχος δεδομένων περιορίζει τη δραστηριότητα, μετριάζει τον κίνδυνο. Με την έννοια του κινδύνου εννοείται η πιθανότητα ενός εισβολέα ή κακόβουλου λογισμικού να χρησιμοποιήσει το Honeynet για να επιτεθεί ή να βλάψει μέρη του συστήματος που δεν ελέγχονται από Honeypots ή να καταχραστεί το Honeynet με μη αναμενόμενο τρόπο. Σκοπός είναι να εξασφαλίζεται ότι από τη στιγμή που ένας εισβολέας είναι μέσα στο Honeynet, δε θα μπορεί τυχαία ή σκόπιμα να βλάψει άλλα συστήματα που δεν ανήκουν στο Honeynet. Ουσιαστικά απαιτείται η εφαρμογή δεδομένων ελέγχου με ελαχιστοποίηση των πιθανοτήτων ανίχνευσης κακόβουλου κώδικα. Για να επιτευχθεί αυτό απαιτείται κάποιος βαθμός ελευθερίας στις κινήσεις των εισβολέων. Όση περισσότερη δραστηριότητα επιτρέπεται στους εισβολείς, τόσα περισσότερα μπορούν να γίνουν γνωστά για τη συμπεριφορά και τη δραστηριότητά τους. Ωστόσο, όση περισσότερη ελευθερία έχουν οι εισβολείς, τόσο μεγαλώνει κι ο κίνδυνος παραβίασης των δεδομένων ελέγχου και πιθανής εισβολής σε συστήματα που δεν ανήκουν στο Honeynet. Η ισορροπία μεταξύ πόσης ελευθερίας επιτρέπεται στους εισβολείς και πόσης απαιτείται είναι ένα μείζον ζήτημα κατά τη δημιουργία του Honeynet. Η καλύτερη μέθοδος προσέγγισης για τα δεδομένα ελέγχου είναι να μη στηρίζονται σε ένα μόνο μηχανισμό για την εφαρμογή τους αλλά σε πολλούς, όπως καταμέτρηση εξερχόμενων συνδέσεων, πρόληψη θυρών εισβολής ή περιορισμοί στο εύρος ζώνης. Ο συνδυασμός πολλών και διαφορετικών μηχανισμών βοηθά στην προστασία ενάντια σε ένα μοναδικό σημείο αποτυχίας ειδικά όταν αφορούν σε νέες και άγνωστες επιθέσεις. Επιπρόσθετα, ο έλεγχος δεδομένων πρέπει να λειτουργεί σε έναν κλειστό κύκλο, ώστε σε περίπτωση αποτυχίας, να μην υπάρξει πλήρης κατάρρευση του συστήματος, δηλαδή να αποτραπεί όλη η εξερχόμενη δραστηριότητα. Ωστόσο, πρέπει να τονιστεί ότι η πιθανότητα κινδύνου μπορεί μόνο να μετριαστεί και όχι να εξαλειφθεί τελείως.
- Data Capture: Σύλληψη δεδομένων είναι η παρακολούθηση και καταγραφή όλης της δραστηριότητας των απειλών εντός του Honeynet. Αυτά τα δεδομένα στη συνέχεια αναλύονται ώστε να γίνουν γνωστά τα εργαλεία, οι μέθοδοι και τα κίνητρα των εισβολέων. Σκοπός είναι να συλληφθούν όσο το δυνατό περισσότερα δεδομένα χωρίς να γίνει γνωστή η διαδικασία αυτή στους εισβολείς. Όπως ακριβώς και με τα δεδομένα ελέγχου χρησιμοποιούνται κι εδώ πολλαπλά επίπεδα για τη σύλληψη δραστηριότητας. Ο συνδυασμός των επιπέδων δε βοηθά μόνο στη συγκέντρωση όλων των ενεργειών του εισβολέα αλλά αποτρέπει τη δημιουργία ενός μόνο σημείου κατάρρευσης συστήματος. Ακόμα, μία από τις προκλήσεις κατά τη σύλληψη δεδομένων είναι ότι ένα μεγάλο μέρος της δραστηριότητας του εισβολέα συμβαίνει πάνω από κρυπτογραφημένα κανάλια, όπως IPsec, SSH, SSL κ.α. Οι μηχανισμοί της σύλληψης δεδομένων πρέπει να λάβουν υπόψη τους την κρυπτογράφηση. Ομοίως με τον έλεγχο δεδομένων, πρέπει να ελαχιστοποιηθεί η δυνατότητα των εισβολέων να ανιχνεύουν τους μηχανισμούς καταγραφής. Αυτό μπορεί να επιτευχθεί με πολλούς τρόπους, όπως με την όσο το δυνατό λιγότερη προσθήκη τροποποιήσεων και την αποθήκευση των δεδομένων όχι τοπικά στο Honeypot αλλά εξωτερικά.
- Data Analysis: Η ανάλυση των δεδομένων ποικίλλει ανάλογα με τις απαιτήσεις της κάθε εταιρείας ή του κάθε οργανισμού για τα δεδομένα της και την ασφάλειά τους.
- Data Collection: Η συλλογή δεδομένων αφορά μόνο στους οργανισμούς που έχουν πολλαπλά Honeynets σε κατανεμημένα περιβάλλοντα. Οι οργανισμοί που έχουν πολλαπλά Honeynets πρέπει να συγκεντρώσουν όλα τα δεδομένα και να τα αποθηκεύσουν σε μία κεντρική τοποθεσία. Έτσι, τα δεδομένα αυτά μπορούν να

συνδυαστούν με ποικίλους τρόπους. Η συλλογή δεδομένων παρέχει όλους τους ασφαλείς τρόπους κεντρικής συγκέντρωσης των δεδομένων.

Η εφαρμογή αυτών των απαιτήσεων και ο συγκερασμός τους ώστε να λειτουργούν σαν ένα ενιαίο σύστημα είναι ιδιαίτερα δύσκολη, περίπλοκη και χρονοβόρα. Στο παρελθόν χρειαζόταν πολύς χρόνος και προσπάθεια για την εφαρμογή αυτής της αρχιτεκτονικής. Το Honeynet Project για να αποφύγει αυτές τις δυσκολίες ανέπτυξε το Honeywall CDROM, το οποίο δημιουργεί και εγκαθιστά γρήγορα το Honeywall, σημείο – κλειδί για την αρχιτεκτονική του Honeynet. [30]

Η εικόνα 7-1 παρουσιάζει τη γενική αρχιτεκτονική ενός Honeynet.



Εικόνα 7-1: Αρχιτεκτονική Honeynet. [30]

Κίνδυνοι

Το Honeynet είναι ένα ιδιαίτερα δυνατό εργαλείο αλλά η χρήση του ενέχει και ορισμένους κινδύνους, οι σημαντικότεροι από τους οποίους αναλύονται παρακάτω.

- Ένα Honeynet μπορεί να χρησιμοποιηθεί για επίθεση σε άλλα συστήματα. Για παράδειγμα, ένας εισβολέας να επιτεθεί στο Honeynet, να το καταλάβει και στη συνέχεια αφού εγκαθιδρύσει μία εξωτερική σύνδεση, να πραγματοποιήσει επιθέσεις προς άλλα συστήματα θέτοντας σε κίνδυνο το ίδιο το Honeynet. Ο έλεγχος δεδομένων είναι υπεύθυνος για τη μετρίαση αυτού του κινδύνου. Πολλαπλά επίπεδα στον έλεγχο δεδομένων χρησιμοποιούνται ώστε ο εισβολέας να δυσκολευτεί να πραγματοποιήσει

επίθεση. Ωστόσο, δεν υπάρχει εγγύηση ότι το Honeynet δε θα χρησιμοποιηθεί ως μέσο για μετέπειτα επιθέσεις ή και ότι ο εισβολέας δε θα παρακάμψει τους μηχανισμούς που θα αποτρέπουν τη δραστηριότητα αυτή.

- Υπάρχει ο κίνδυνος της ανίχνευσης. Από τη στιγμή που αποκαλυφθεί η πραγματική ταυτότητα του Honeynet, η αξία του μειώνεται δραματικά. Οι επιτιθέμενοι μπορούν να αγνοήσουν ή να παρακάμψουν το Honeynet, εξαλείφοντας έτσι την ικανότητά του να καταγράψει πληροφορίες. Ακόμα πιο επικίνδυνη είναι η απειλή ότι ένας εισβολέας μπορεί να εισάγει ψευδείς ή ανακριβείς πληροφορίες σε ένα Honeynet παραποιώντας έτσι την ανάλυση δεδομένων. Για παράδειγμα, ένας έμπειρος εισβολέας έχοντας αποκτήσει τοπική πρόσβαση σε ένα Honeynet, μπορεί να αναγνωρίσει την πραγματική ιδιότητα του Honeynet και να αναγνωρίσει ομοίως τους μηχανισμούς ελέγχου δεδομένων και σύλληψης δεδομένων.
- Υπάρχει ο κίνδυνος απενεργοποίησης της λειτουργικότητας του Honeynet μέσω επιθέσεων στις ρουτίνες του ελέγχου δεδομένων και της σύλληψης των δεδομένων και μάλιστα χωρίς ενδεχομένως ο διαχειριστής του Honeynet να γνωρίζει το γεγονός αυτό. Η χρήση πολλαπλών επιπέδων στα συστατικά του Honeynet βοηθάει στη μείωση αυτού του κινδύνου καθώς έτσι δεν υπάρχει μοναδικό σημείο κατάρρευσης.
- Οι εισβολείς μπορεί να μην προσπαθήσουν να υλοποιήσουν επιθέσεις σε άλλα συστήματα μέσω του Honeynet αλλά να πραγματοποιήσουν παράνομες δραστηριότητες εν γένει. Για παράδειγμα να ανεβάσουν και στη συνέχεια να διανέμουν παράνομο, πορνογραφικό υλικό ή περιεχόμενο που υπόκειται σε έλεγχο πνευματικών δικαιωμάτων οπότε σε αυτή την περίπτωση ο ιδιοκτήτης του Honeynet θα πρέπει να αποδείξει ότι δεν ήταν αυτός στην πραγματικότητα που πραγματοποίησε τις αξιόποινες αυτές πράξεις. [30]

Υπάρχουν διάφορα μέτρα για την αντιμετώπιση αυτών των κινδύνων. Τα σημαντικότερα είναι η ανθρώπινη παρακολούθηση και η παραμετροποίηση. Η ανθρώπινη παρακολούθηση έγκειται στην επαγγελματική παρακολούθηση και ανάλυση των δεδομένων του Honeynet σε πραγματικό χρόνο. Αυτό δίνει τη δυνατότητα ανίχνευσης αποτυχίας στην περίπτωση που αυτοματοποιημένοι μηχανισμοί αποτύχουν να την αναγνωρίσουν και να αντιδράσουν.

Η παραμετροποίηση παίζει επίσης μεγάλο ρόλο. Το Honeynet Project αποτελεί λογισμικό ανοιχτού κώδικα και επομένως έχουν ελεύθερη πρόσβαση σε αυτό οι πάντες, συμπεριλαμβανομένης της hacking κοινότητας. Οπότε ο δημιουργός ενός Honeynet οφείλει να το προσαρμόσει στις ανάγκες του και να το παραμετροποιήσει ώστε να διαφέρει από την αρχική έκδοση και να είναι πιο δύσκολο για τρίτους να ανιχνευτεί και να γίνει στόχος επίθεσης. [1]

7.2 Honeytokens

Ως Honeytoken ορίζεται το Honeypot που δεν είναι υπολογιστής αλλά κάποιου είδους ψηφιακή οντότητα. Μπορεί να είναι ένας αριθμός πιστωτικής κάρτας, ένα υπολογιστικό φύλλο Excel, μια παρουσίαση Powerpoint, μια καταχώρηση δεδομένων ή και μία ψεύτικη είσοδος. Μπορεί να έχει πολλές μορφές αλλά η βασική ιδέα που πρέπει να ικανοποιείται είναι αυτή των Honeypots, ότι δηλαδή η αξία τους έγκειται στη μη εξουσιοδοτημένη χρήση τους. Όπως ακριβώς ένας υπολογιστής που λειτουργεί ως Honeypot δεν έχει καμία εξουσιοδοτημένη χρήση, έτσι και ένα Honeytoken δεν έχει ομοίως καμία εξουσιοδοτημένη χρήση. Κανείς δεν επιτρέπεται να χρησιμοποιεί ή να προσπελαύνει ένα Honeytoken. Αυτό προσδίδει στα Honeytokens την ίδια ισχύ και τα ίδια πλεονεκτήματα των παραδοσιακών Honeypots αλλά επεκτείνει ταυτόχρονα τις δυνατότητες τους πέρα από τις δυνατότητες των φυσικών υπολογιστών.

Το σκεπτικό βάσει του οποίου λειτουργούν τα Honeytokens δεν είναι καινούριο. Για παράδειγμα, η εισαγωγή ψεύτικων πόλεων ή οδών σε χάρτες από εταιρείες χαρτογράφησης προκειμένου να διαπιστωθεί αν κυκλοφορούν αντίγραφα των συγκεκριμένων χαρτών, ακολουθεί το ίδιο σκεπτικό. Ο όρος όμως Honeytoken πρωτοεμφανίστηκε από τον Augusto Paes de Barros το 2003 στη λίστα αλληλογραφίας των Honeypots. Ο όρος περιέγραφε ακριβώς το σκεπτικό αυτό. [31]

Λειτουργικότητα

Ένα Honeytoken λειτουργεί ακριβώς όπως ένα Honeypot. Κανείς δεν πρέπει να αλληλεπιδράσει με αυτό. Τι θα χρησιμοποιηθεί ως Honeytoken και πώς, εξαρτάται από το δημιουργό του. Για παράδειγμα, σε βάσεις δεδομένων όπου απαγορεύεται η μη εξουσιοδοτημένη πρόσβαση, εισάγεται μία ψεύτικη εγγραφή που λειτουργεί σα δόλωμα και θεωρητικά εμπεριέχει ευαίσθητες πληροφορίες για τη βάση και το περιεχόμενό της. Αν κάποιος χρήστης αποκτήσει πρόσβαση σε αυτή την εγγραφή θεωρείται ότι η βάση έχει παραβιαστεί. Το σκεπτικό είναι ιδιαίτερα απλό, δεν υπάρχουν ειδικοί αλγόριθμοι, ψηφιακές υπογραφές ή κανόνες που πρέπει να ικανοποιηθούν. Φορτώνονται οι εγγραφές και εάν επιχειρηθεί πρόσβαση σε ψεύτικη εγγραφή, το σύστημα έχει παραβιαστεί.

Το σκεπτικό αυτό μπορεί να επεκταθεί και πέρα από τις βάσεις δεδομένων. Διακομιστές αρχείων ή ηλεκτρονικής αλληλογραφίας ή διαδικτυακοί μπορούν να φέρουν ενσωματωμένα Honeytokens. Οτιδήποτε φέρει δεδομένα μπορεί να φορτωθεί με ψεύτικα δεδομένα, η πρόσβαση στα οποία θα σημαίνει την παραβίαση του συστήματος από μη εξουσιοδοτημένους χρήστες.

Τα Honeytokens δε λύνουν ένα συγκεκριμένο πρόβλημα. Δεν έχουν σχεδιαστεί αποκλειστικά για την ανίχνευση ή πρόληψη επιθέσεων. Αντίθετα, είναι ένα εξαιρετικά ευέλικτο και απλό εργαλείο με πολλαπλές εφαρμογές στην ασφάλεια που ποικίλουν από την ανίχνευση μέχρι την αναγνώριση της ταυτότητας της απειλής και των κινήτρων της.

Επιπρόσθετα, ένα μεγάλο πλεονέκτημά τους είναι το σχεδόν μηδενικό κόστος τους. Δεν απαιτούν ειδικό λογισμικό, αναβαθμίσεις, έκδοση αδειών.

Το μόνο πράγμα που καθορίζει και περιορίζει τη δραστηριότητά τους είναι η φαντασία. Ένα Honeytoken μπορεί να αποτελείται από οτιδήποτε. [31]

7.3 FakeAp

Το FakeAp, (Fake Access Points), αναπτύχθηκε από την ομάδα του BlackAlchemy.to για να έλκει hackers και άλλους εισβολείς προκειμένου να συλλέξει πληροφορίες για αυτούς.

Ουσιαστικά δημιουργεί πολλαπλά σημεία πρόσβασης που ακολουθούν το ασύρματο πρωτόκολλο 802.11b και παραπλανεί και ανιχνεύει Wardrivers, NetStumblers, Script Kiddies και λοιπούς ανεπιθύμητους.

Το FakeAp εκτελείται σε Linux περιβάλλοντα και διανέμεται ελεύθερα υπό την άδεια της GPL. [38]

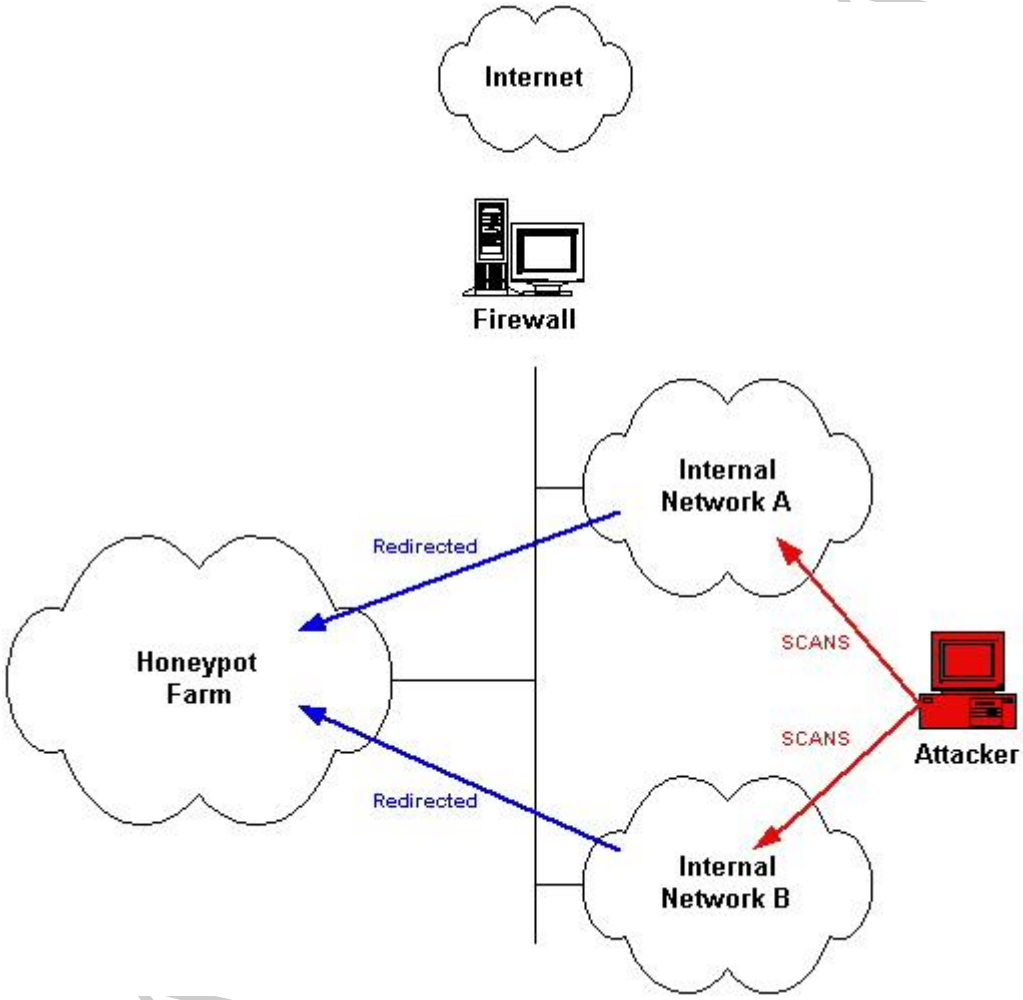
7.4 Honeyfarms

Τα Honeyfarms αναπτύχθηκαν για να απλοποιήσουν τη διαδικασία εγκατάστασης μεγάλων Honeypot συστημάτων. Αντί να εγκαθίσταται μεγάλος αριθμός από Honeypots σε κάθε δίκτυο, εγκαθίστανται όλα τα Honeypots σε μία απομονωμένη ξεχωριστή μονάδα, η οποία αποτελεί ένα Honeyfarm. Όλες οι επιθέσεις στη συνέχεια ανακατευθύνονται προς τη Honeypot φάρμα, ανεξάρτητα από το δίκτυο στο οποίο βρίσκονται τη δεδομένη στιγμή και το δίκτυο προς το οποίο κάνουν επίθεση. Τα Honeypots, όπως έχει ήδη αναφερθεί, δε χρειάζεται να είναι φυσικά εγκατεστημένα σε ένα δίκτυο. Επομένως, οι Honeyfarms χρησιμοποιούν προγράμματα ανακατεύθυνσης, τα οποία οδηγούν τους εισβολείς σε εικονικά Honeypots εντός του Honeyfarm αλλά εκτός του δικτύου παραγωγής. Ο εισβολέας επομένως θεωρεί ότι αλληλεπιδρά με ένα θύμα σε ένα τοπικό δίκτυο, ενώ στην πραγματικότητα έχει μεταφερθεί σε μία Honeypot φάρμα. [39]

Τα πλεονεκτήματα της χρήσης Honeyfarms είναι τεράστια. Η ανάπτυξη των Honeypots γίνεται μια εξαιρετικά απλή υπόθεση. Αντί της χρήσης ενός μόνο honeypot για κάθε δίκτυο, χρησιμοποιείται μία ενιαία, κεντρική εγκατάσταση πολλών Honeypots, η Honeyfarm. Η Honeyfarm μπορεί να είναι τόσο απλή όσο διάφορα χαμηλής αλληλεπίδρασης Honeypots μέχρι ένα μεγάλο Honeynet που αποτελείται από εκατοντάδες πραγματικών συστημάτων και εφαρμογών που αναμένουν να γίνουν αντικείμενο επίθεσης. Το γεγονός αυτό καθιστά τη συντήρηση, παραμετροποίηση και ανάλυση της λειτουργίας των Honeypots αρκετά ευκολότερη

αφού για την εικονική εγκατάσταση τους σε άλλα δίκτυα απαιτείται, όπως έχει ήδη αναφερθεί, απλά ένα πρόγραμμα ανακατεύθυνσης προς τους διαχειριστές των δικτύων. Τα προγράμματα αυτά λειτουργούν σα μαύρα κουτιά που συνδέονται φυσικά στα τοπικά δίκτυα και παρακολουθούν προκαθορισμένες διευθύνσεις IP. Όταν ένας εισβολέας αλληλεπιδράσει με αυτές τις διευθύνσεις κακόβουλα ή μη εξουσιοδοτημένα, το μαύρο κουτί μεταφέρει τον εισβολέα στη Honeyfarm. Έτσι, η αναβάθμιση και διαχείριση των Honeypots γίνεται ευκολότερη, αφού οι διαχειριστές ασφαλείας έχουν μόνο να αναβαθμίσουν ένα συγκεκριμένο φυσικό Honeynet. Τέλος, το μεγαλύτερο πλεονέκτημα των Honeyfarms είναι η μετρίαση του κινδύνου μη εξουσιοδοτημένης πρόσβασης, καθώς όλη η εισερχόμενη και εξερχόμενη κίνηση μεταφέρεται εκτός του δικτύου παραγωγής.

Στην εικόνα 7-2 διαφαίνεται η αρχιτεκτονική ενός Honeyfarm.



Εικόνα 7-2: Αρχιτεκτονική Honeyfarm. [39]

Παράδειγμα Honeyfarm είναι το NetBait, μία εμπορική εφαρμογή που χρησιμοποιεί Honeyfarms, οι οποίες ονομάζονται ServerFarms στην προκειμένη περίπτωση. Σε αυτές τις Honeypot φάρμες μπορεί να εγκατασταθεί οποιοδήποτε σύστημα και οποιαδήποτε εφαρμογή επιθυμεί ο χρήστης. Τα προγράμματα ανακατεύθυνσης αναλαμβάνουν να οδηγήσουν τη δραστηριότητα των εισβολέων σε προκαθορισμένα συστήματα εντός των ServerFarms. Οι εισβολείς επιτίθενται σε συγκεκριμένες διευθύνσεις και δε συνειδητοποιούν ότι επιτίθενται σε ένα σύστημα που φυσικά λειτουργεί κάπου εξωτερικά. Το NetBait προσφέρει ServerFarms ως υπηρεσίες. Δημιουργούν και συντηρούν για κάθε οργανισμό μία ServerFarm. Το μόνο που

χρειάζεται να κάνει ένας οργανισμός είναι να φορτώσει προγράμματα ανακατεύθυνσης στα δίκτυά του που κατευθύνουν την κακόβουλη ή μη εξουσιοδοτημένη πρόσβαση σε ServerFarms. Οι οργανισμοί επομένως δε χρειάζεται να συντηρούν και να αναλύουν τα δεδομένα των Honeypots ούτε να ανησυχούν για ζητήματα αυθεντικοποίησης, εμπιστευτικότητας κι εξουσιοδότησης. [39]

7.5 HoneyPages

Οι HoneyPages είναι ιστοσελίδες που βρίσκονται διασκορπισμένες σε μια διαδικτυακή τοποθεσία. Δε διαθέτουν κανέναν υπερσύνδεσμο προς πραγματικές σελίδες, ούτε έχουν κανένα συγκεκριμένο θεμιτό λόγο ύπαρξης. Οι κανονικοί χρήστες της τοποθεσίας δεν μπορούν σε καμία περίπτωση να προσπελάσουν αυτές τις σελίδες. Ωστόσο, οι σχεδιαστές των τοποθεσιών αυτών ενσωματώνουν κρυφούς υπερσυνδέσμους προς τις HoneyPages ως σχόλια ή κρυφά πεδία σε έγκυρες σελίδες. Θεωρείται δεδομένο ότι ένας εισβολέας που αναλύει τον πηγαίο κώδικα των σελίδων ή και ένα πρόγραμμα που λειτουργεί σαν ανιχνευτής αδυναμιών, θα αντιληφθούν τις υποδείξεις αυτές και θα επιτεθούν, και όταν οι HoneyPages προσπελαστούν, θα υποδείξουν τους επιτιθέμενους. [40]

7.6 client Honeypots

Τα παραδοσιακά Honeypots λειτουργούν ως εξυπηρετητές, (servers), ή ως μηχανισμοί που αποκαλύπτουν τις υπηρεσίες και τη λειτουργικότητα των εξυπηρετητών και αναμένουν παθητικά να δεχτούν επίθεση. Τα client Honeypots αντίθετα αποτελούν ενεργούς μηχανισμούς ασφαλείας που αναζητούν κακόβουλους εξυπηρετητές που επιτίθενται σε πελάτες. Προσομοιώνουν δηλαδή πελάτες και συνδέονται με εξυπηρετητές προκειμένου να διαπιστώσουν αν έχει πραγματοποιηθεί καμιά επίθεση. Ανιχνεύουν το δίκτυο ώστε να εντοπίσουν μέσω της απόκρισης που στέλνουν την ύπαρξη εξυπηρετητών που εκμεταλλεύονται τον client. Ο εντοπισμός αυτός βασίζεται στον έλεγχο της κατάστασης του λειτουργικού συστήματος του εξυπηρετητή, περιλαμβάνει δηλαδή παρακολούθηση αλλαγών στο σύστημα αρχείων, την παραμετροποίηση και τη λίστα διεργασιών. Τα client Honeypots που χρησιμοποιούν πραγματικά συστήματα, θεωρούνται Honeypots υψηλής αλληλεπίδρασης.

Συχνά, το κέντρο ενδιαφέροντος για τους client Honeypots βρίσκεται στους διαδικτυακούς φυλλομετρητές, (web browsers), αλλά το καθετί που αλληλεπιδρά με τους εξυπηρετητές σαν πελάτης, μπορεί να θεωρηθεί μέρος ενός client Honeypot, (για παράδειγμα ftp, ssh υπηρεσίες κ.α.). Για παράδειγμα, τα client Honeypots HoneyMonkey και HoneyClient με τη χρήση ενός φυλλομετρητή ανιχνεύουν το Διαδίκτυο για να εντοπίσουν εξυπηρετητές που εκμεταλλεύονται clients.

Ένα client Honeypot αποτελείται από τρία βασικά συστατικά. Το πρώτο είναι ο queuer, ο οποίος δημιουργεί μια λίστα από εξυπηρετητές να επισκεφτεί ο client. Η λίστα αυτή μπορεί να δημιουργηθεί για παράδειγμα μέσω ανίχνευσης. Το δεύτερο στοιχείο είναι ο client, που αναλαμβάνει να συνδεθεί με τους εξυπηρετητές της λίστας και το τρίτο στοιχείο είναι μια μηχανή ανάλυσης, που είναι υπεύθυνη να καθορίσει αν μια επίθεση έχει λάβει χώρα στο client Honeypot. Επιπρόσθετα, τα client Honeypots διαθέτουν και κάποια στρατηγική περιορισμού, ώστε να αποφευχθεί η διάδοση των επιτυχών επιθέσεων εκτός του client Honeypot. Αυτό επιτυγχάνεται με τη χρήση τειχών ασφαλείας ή και εικονικών μηχανημάτων sandboxes.

Σε αναλογία με τα παραδοσιακά server Honeypots, τα client Honeypots ταξινομούνται βάσει του βαθμού αλληλεπίδρασής τους σε χαμηλής, μεσαίας και υψηλής αλληλεπίδρασης.

Ως υψηλής αλληλεπίδρασης client Honeypots θεωρούνται τα Capture-HPC, HoneyMonkey, SHELIA, UW SpyCrawler και Web Exploit Finder. Ως χαμηλής αλληλεπίδρασης client Honeypots θεωρούνται τα HoneyC, Monkey-Spider, PhoneyC και SpyBye και ως μεσαίας θεωρείται το HoneySpider. [41]

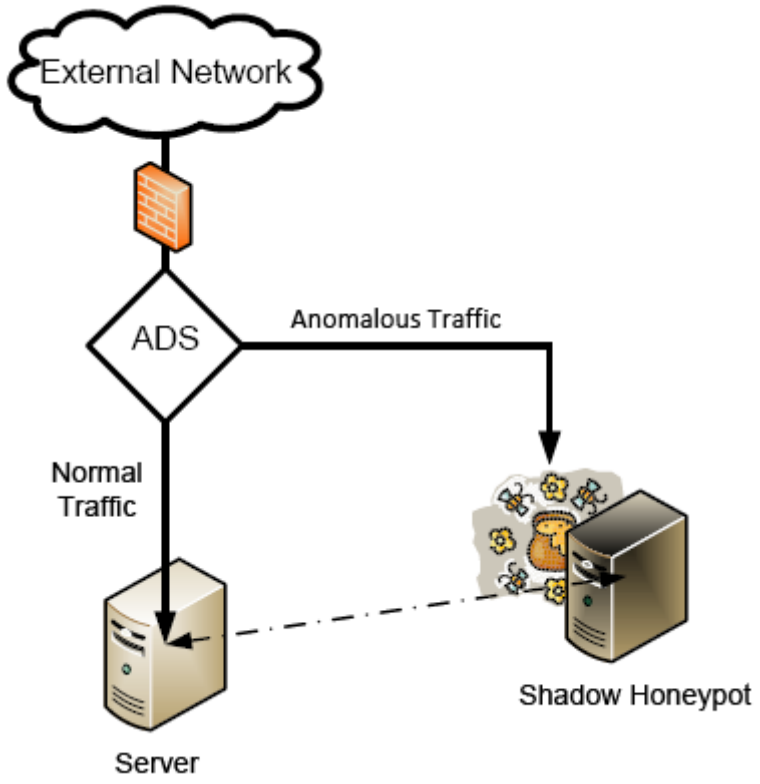
7.7 Shadow Honeypots

Τα Shadow Honeypots είναι ένας συνδυασμός κλασσικών Honeypots και συστημάτων ανίχνευσης ανωμαλιών, (Anomaly Detection Systems – ADS), που αποτελούν μία εναλλακτική

στα συστήματα ανίχνευσης εισβολών, που βασίζονται σε κανόνες. Σε υψηλό επίπεδο, χρησιμοποιείται μια ποικιλία ανιχνευτών ανωμαλιών για την παρακολούθηση όλης της κίνησης σε ένα προστατευμένο δίκτυο ή υπηρεσία. Διαχωρίζεται η κανονική από την ύποπτη κίνηση και η ύποπτη κίνηση, δηλαδή η «ανώμαλη», στέλνεται στο Shadow Honeypot, όπου υποβάλλεται σε επεξεργασία για να προσδιοριστεί η ακρίβεια της υπό πρόβλεψη ανωμαλίας. Η «σκιά» του Shadow Honeypot αποτελεί ένα στιγμιότυπο του λογισμικού του προστατευμένου δικτύου που μοιράζεται όλη την εσωτερική κατάσταση με ένα κανονικό στιγμιότυπο της εφαρμογής και έχει αναπτυχθεί έτσι ώστε να ανιχνεύει πιθανές επιθέσεις. Όταν καταγραφούν επιθέσεις εναντίον της σκιάς, τυχόν αλλαγές που προέκυψαν απορρίπτονται ενώ η νόμιμη κίνηση που ταξινομήθηκε εσφαλμένα, επικυρώνεται από τη σκιά και στη συνέχεια αντιμετωπίζεται ανάλογα από το σύστημα με διαφανή τρόπο προς τον τελικό χρήστη. Το αποτέλεσμα της επεξεργασίας της αίτησης από τη σκιά χρησιμοποιείται για να φιλτράρει μελλοντικές επιθέσεις και μπορεί να χρησιμοποιηθεί για να αναβαθμίσει τον ανιχνευτή ανωμαλιών. [42]

Η αρχιτεκτονική αυτή επιτρέπει στους σχεδιαστές συστημάτων να τελειοποιήσουν από την πλευρά της απόδοσης συστήματα, αφού τα ψευδώς θετικά αποτελέσματα φιλτράρονται από τη σκιά. Σε αντίθεση με τα κανονικά Honeypots, η αρχιτεκτονική αυτή μπορεί να χρησιμοποιηθεί και για server και για client εφαρμογές. [4]

Στην εικόνα 7-3 διαφαίνεται η αρχιτεκτονική ενός Shadow Honeypot.



Εικόνα 7-3: Αρχιτεκτονική Shadow Honeypot. [4]

7.8 Σύνοψη

Honeypot	Χρόνος	Λειτουργικότητα

Honeynet	1999	Δίκτυο Honeypots υψηλής αλληλεπίδρασης, IDS, τειχών προστασίας που συνδέεται σε πραγματικό σύστημα παραγωγής. Honeywall δρομολογητής μεταξύ Honeynet και συστήματος παραγωγής, επιτρέπει εισερχόμενη κίνηση προς το Honeynet, ελέγχει εξερχόμενη.
Honeytoken	2003	Οτιδήποτε, ένας αριθμός πιστωτικής κάρτας, ένα υπολογιστικό φύλλο Excel, μια καταχώρηση δεδομένων. Οποιαδήποτε αλληλεπίδραση με αυτό θεωρείται κακόβουλη.
FakeAp	2002	Δημιουργεί πολλαπλά σημεία πρόσβασης που ακολουθούν το πρωτόκολλο 802.11b και ανιχνεύει για Wardrivers, NetStumblers, Script Kiddies.
Honeyfarm	2003	Απομονωμένη ξεχωριστή μονάδα που αποτελείται από Honeypots και συνδέεται σε πραγματικό σύστημα παραγωγής. Όλες οι επιθέσεις προς το σύστημα παραγωγής ανακατευθύνονται στη Honeyfarm.
HoneyPage	2012	Ιστοσελίδες που βρίσκονται διασκορπισμένες σε μια διαδικτυακή τοποθεσία. Δε διαθέτουν κανέναν υπερσύνδεσμο προς πραγματικές σελίδες, ούτε έχουν κανένα συγκεκριμένο θεμιτό λόγο ύπαρξης. Όταν προσπελαστούν, υποδεικνύουν τον επιτιθέμενο.
Client	2001	Προσομοιώνουν clients και συνδέονται με servers προκειμένου να διαπιστώσουν αν έχει πραγματοποιηθεί καμιά επίθεση. Ανιχνεύουν το δίκτυο ώστε να εντοπίσουν μέσω της απόκρισης που στέλνουν την ύπαρξη κακόβουλων servers.
Shadow	2005	Συνδυασμός Honeypots και ADS. Παρακολουθείται όλη η δικτυακή κίνηση και η «ύποπτη» μεταφέρεται στο Shadow Honeypot.

Πίνακας 7-1: Συνοπτική παρουσίαση προηγμένων Honeypots.

8. Αρχιτεκτονικη – Τοποθετηση των Honeypots

Ένα Honeypot μπορεί να τοποθετηθεί σε τρία σημεία:

- εξωτερικά, εκτός τείχους προστασίας και σε σύνδεση με το Διαδίκτυο,
- εσωτερικά, εντός τείχους προστασίας και
- στην αποστρατικοποιημένη ζώνη του δικτύου, (Demilitarized Zone – DMZ).

Παρακάτω περιγράφεται η σημασία του Honeypot σε καθεμία από αυτές τις τρεις θέσεις.

8.1 Εξωτερική τοποθέτηση

Σε αυτή την περίπτωση το Honeypot βρίσκεται εκτός τείχους προστασίας και είναι συνδεδεμένο στο Διαδίκτυο. Το Honeypot αποτελεί έτσι τον πρώτο δέκτη της εισερχόμενης κίνησης στο δίκτυο, οπότε το τείχος προστασίας ή και το εσωτερικό Σύστημα Ανίχνευσης Παρείσφρησης – αν υπάρχει – γλιτώνει την παραγωγή ειδοποιήσεων για ανεπιθύμητη κίνηση, όπως ανιχνεύσεις θυρών ή εύρεση μοτίβων εισβολών. Σε αντίθετη περίπτωση, πλήθος ειδοποιήσεων θα παραγόταν από το τείχος προστασίας ή και από το εσωτερικό Σύστημα Ανίχνευσης Παρείσφρησης.

Ένα εξίσου σημαντικό πλεονέκτημα για το τείχος προστασίας ή και το εσωτερικό Σύστημα Ανίχνευσης Παρείσφρησης είναι ότι δε χρειάζονται να παραμετροποιηθούν ανάλογα ώστε η λειτουργία του Honeypot να είναι επιτρεπτή στο δίκτυο. Το Honeypot αντιμετωπίζεται σαν ένα οποιοδήποτε άλλο μηχάνημα του εξωτερικού δικτύου και η λειτουργία του επομένως δεν μπορεί να εισάγει νέους κινδύνους στο δίκτυο.

Το μειονέκτημα της τοποθέτησης αυτής είναι ότι οι εντός του δικτύου εισβολείς δεν μπορούν να γίνουν αντιληπτοί, ειδικά αν το τείχος προστασίας περιορίζει την εξερχόμενη κίνηση και συνακόλουθα περιορίζει και την κίνηση προς το Honeypot. [9]

Η επιλογή αυτή είναι κατάλληλη για μεμονωμένα ερευνητικά Honeypots, καθώς αφού δεν υπάρχει τείχος προστασίας, το Honeypot μοιράζεται το ίδιο υποδίκτυο διευθύνσεων μαζί με τα άλλα μηχανήματα του δικτύου και είναι διαθέσιμο για άμεση ανίχνευση, επίθεση και κατάληψη. [55]

Στην εικόνα 8-1 στο σχήμα 1 φαίνεται η εξωτερική τοποθέτηση ενός Honeypot.

8.2 Εσωτερική τοποθέτηση

Σε αυτή την περίπτωση το Honeypot βρίσκεται μέσα στο δίκτυο, με το τείχος προστασίας να είναι ανάμεσα στο Honeypot και το Διαδίκτυο. Το Honeypot μπορεί έτσι να εισάγει νέους κινδύνους ασφαλείας στο εσωτερικό δίκτυο, ειδικά αν το εσωτερικό δίκτυο δε διαθέτει πρόσθετο τείχος ασφαλείας για το φιλτράρισμα των δεδομένων από και προς το Honeypot.

Αντίστοιχα, το τείχος προστασίας ή και το Σύστημα Ανίχνευσης Παρείσφρησης – αν υπάρχει – αναπόφευκτα πρέπει να παραμετροποιηθούν ανάλογα ώστε να μην παράγονται ειδοποιήσεις κάθε φορά που το Honeypot γίνεται αντικείμενο επίθεσης ή ανίχνευσης από εισβολείς. [9]

Το μεγάλο μειονέκτημα αυτής της τοποθέτησης είναι ο κίνδυνος κατάληψης του Honeypot από εισβολείς. Σε αυτή την περίπτωση, οι εισβολείς έχουν τη δυνατότητα να αποκτήσουν πρόσβαση στο εσωτερικό δίκτυο μέσω του Honeypot. Οπότε σε αυτή την περίπτωση απαιτείται η απομόνωση του Honeypot από το υπόλοιπο δίκτυο, ώστε οι εισβολείς να μην καταφέρουν να αποκτήσουν πρόσβαση οπουδήποτε αλλού. Ειδικότερα, οι διαχειριστές του δικτύου θα πρέπει να αποφασίσουν ποια διαδικτυακή κίνηση θα ανακατευθύνεται στο Honeypot και ποια στο υπόλοιπο δίκτυο. Ενδείκνυται να ανακατευθύνονται στο Honeypot τα αιτήματα συνδέσεων υπηρεσιών που δεν εκτελούνται από μηχανήματα του δικτύου.

Ο κυριότερος λόγος για την εσωτερική τοποθέτηση των Honeypots είναι η ανακάλυψη εσωτερικών εισβολέων ή και εξωτερικών που έχουν καταφέρει να προσπελάσουν το τείχος ασφαλείας του δικτύου και να εισέλθουν στο δίκτυο. Παράλληλα, διαπιστώνονται συχνά με αυτό τον τρόπο και ευπάθειες στο τείχος προστασίας. [55]

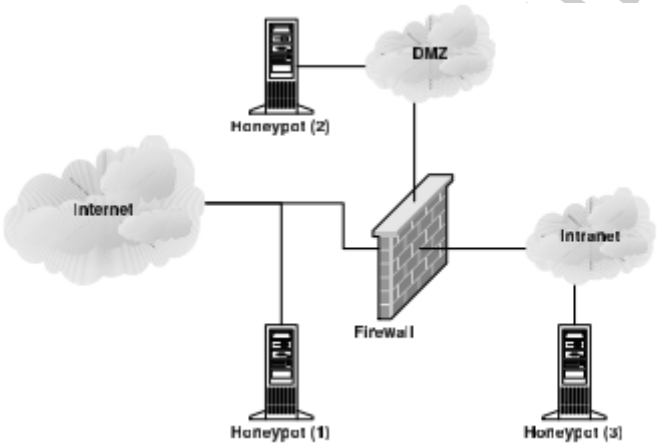
Στην εικόνα 8-1 στο σχήμα 3 φαίνεται η εσωτερική τοποθέτηση ενός Honeypot.

8.3 Τοποθέτηση στην αποστρατικοποιημένη ζώνη (DMZ)

Η τοποθέτηση αυτή είναι ιδανική λύση για μια εταιρεία ή οργανισμό, ωστόσο απαιτεί πολύπλοκη και προσεκτική σχεδίαση, καθώς οι περισσότερες αποστρατικοποιημένες ζώνες δεν είναι πλήρως προσβάσιμες και μόνο υπηρεσίες που το απαιτούν, περνούν το τείχος προστασίας. Απαιτείται επομένως τα υπόλοιπα συστήματα που βρίσκονται εντός της αποστρατικοποιημένης ζώνης να είναι προστατευμένα από το Honeypot, δηλαδή για την ακρίβεια το Honeypot να λειτουργεί απομονωμένα. [9]

Τα Honeypots στην περίπτωση αυτή λειτουργούν ως σύστημα ειδοποίησης σε περιπτώσεις επίθεσης που έχουν στόχο την αποστρατικοποιημένη ζώνη. Ομοίως, μπορούν να λειτουργήσουν και ως αντίγραφα πραγματικών συστημάτων παραγωγής, ώστε οι διαχειριστές να πάρουν πληροφορίες για πιθανούς εισβολείς. Ωστόσο, για την ανίχνευση και πρόληψη επιθέσεων σε εσωτερικά δίκτυα, απαιτείται η τοποθέτηση Honeypots και σε άλλα σημεία. [55]

Στην εικόνα 8-1 στο σχήμα 2 φαίνεται η τοποθέτηση ενός Honeypot στην αποστρατικοποιημένη ζώνη.



Εικόνα 8-1 Τοποθέτηση ενός Honeypot. [9]

9. ΝΟΜΙΚΑ ΖΗΤΗΜΑΤΑ HONEYPOTS

Δεν υπάρχει νομοθετημένο θεσμικό πλαίσιο χρήσης Honeypots. Κάθε χώρα έχει θεσπίσει τους δικούς της νόμους όσον αφορά στην Ασφάλεια Πληροφοριακών Συστημάτων και την Προστασία Ιδιωτικότητας και η χρήση των Honeypots υπόκειται στους νόμους αυτούς. Σύμφωνα με τον L. Spitzner, η νόμιμη χρήση των Honeypots εξετάζεται υπό το πρίσμα της παγίδευσης και της ιδιωτικότητας. [61]

9.1 Παγίδευση

Ως παγίδευση, (entrapment), χαρακτηρίζεται η πρόκληση ποινικού αδικήματος χωρίς να υπάρχει προηγουμένως πρόθεση από τη μεριά του δράστη.

Τα Honeypots δε θεωρείται ότι παγιδεύουν εισβολείς γιατί δεν προκαλούν επιθέσεις. Είναι συστήματα παραγωγής ή προσομοιώνουν πραγματικά συστήματα παραγωγής ώστε να λάβουν επιθέσεις. Η λειτουργικότητά τους δεν ευνοεί τη διενέργεια επιθέσεων. Οι εισβολείς βάσει δικής τους πρωτοβουλίας εντοπίζουν και εξαπολύουν επιθέσεις σε Honeypots. Οι διαχειριστές ενός δικτύου δε λειτουργούν ως όργανα επιβολής νόμων ή ως τιμωροί αλλά επιδιώκουν μέσω της χρήσης των Honeypots να ανακαλύψουν τις ευπάθειες του δικτύου τους και να τις διορθώσουν. [61]

9.2 Ιδιωτικότητα

Όσον αφορά στο ζήτημα της ιδιωτικότητας και στο δικαίωμα συλλογής πληροφοριών και προσωπικών δεδομένων εισβολέων, η χρήση των Honeypots ομοίως εξαρτάται από τη χώρα λειτουργίας καθώς και την ταυτότητα του χρήστη.

Σύμφωνα με τον L. Spitzner και με βάση την αμερικανική νομοθεσία περί ηλεκτρονικής έρευνας για εγκληματολογικούς σκοπούς, οι χρήστες που αποκτούν πρόσβαση σε ένα Honeypot σαφώς δεν έχουν καμία εξουσιοδότηση να το κάνουν και εάν απομακρυσμένα λάβουν αρχεία και τα μεταφέρουν στο Honeypot, αυτόματα χάνουν τα δικαιώματα Προστασίας Ιδιωτικών Δεδομένων που διαφορετικά θα είχαν. Ομοίως η χρήση των Honeypots για επικοινωνία αφαιρεί το δικαίωμα Προστασίας Ιδιωτικών Δεδομένων, αφού τα Honeypots δε συνιστούν υπηρεσίες παροχής υπηρεσιών και δεν υπόκεινται σε απαιτήσεις Προστασίας Ιδιωτικών Δεδομένων, τις οποίες πρέπει να ικανοποιούν εν γένει οι υπηρεσίες παροχής υπηρεσιών. Τέλος, οι διαχειριστές των οργανισμών που χρησιμοποιούν Honeypots δε λειτουργούν ως όργανα επιβολής νόμων οπότε δεν απαιτείται από αυτούς καμία ενέργεια περί δικαιωμάτων Ιδιωτικότητας. Στην ουσία τα Honeypots που έχουν στην επιχείρησή τους λειτουργούν σαν οποιοδήποτε άλλο σύστημα ασφαλείας που καταγράφει δραστηριότητα και κίνηση δικτύου. [61]

10. Εφαρμογη

Στα πλαίσια της παρούσας μεταπτυχιακής διατριβής εγκαταστάθηκαν και μελετήθηκαν τρία Honeypots, το Kippo, το Dionaea και το Glastopf. Το μεν πρώτο είναι μεσαίας αλληλεπίδρασης και τα άλλα δύο χαμηλής. Προτιμήθηκαν έναντι Honeypots υψηλής αλληλεπίδρασης ή και προηγμένων λόγω έλλειψης σύνθετων υποδομών καθώς τα Honeypots υψηλής αλληλεπίδρασης και τα προηγμένα απαιτούν την ύπαρξη δικτύου για τη λειτουργία τους.

Τα Kippo και Dionaea αναπτύχθηκαν στο ίδιο μηχάνημα ως μέρος της σουίτας HoneyDrive. Το Glastopf εγκαταστάθηκε και αναπτύχθηκε σε ξεχωριστό μηχάνημα. Από τη σουίτα HoneyDrive δεν υλοποιήθηκαν όλα τα Honeypots που προσφέρονται γιατί ορισμένα εξ αυτών χρησιμοποιούσαν κοινές θύρες, (π.χ. Dionaea, Glastopf, Amun, Nepenthes), οπότε η λειτουργία τους θα αναιρούνταν ενώ άλλα κρίθηκε χρησιμότερο να μην εγκατασταθούν λόγω παλαιότητας, (π.χ. το πολύ δημοφιλές χαμηλής αλληλεπίδρασης Honeyd ανακοινώθηκε το 2003 και η τελευταία του αναβάθμιση ανακοινώθηκε το 2007). Και τα τρία προσομοιώνουν διαφορετικά πρωτόκολλα οπότε απευθύνονται και σε εισβολείς διαφορετικού σκοπού, συνακόλουθα μπορούν να εξαχθούν σημαντικά συμπεράσματα ανά πρωτόκολλο.

Στα κεφάλαια που ακολουθούν, περιγράφονται η λειτουργία του καθενός, η ανάλυση των δεδομένων που συγκέντρωσαν και τα συμπεράσματα που προκύπτουν.

10.1 HoneyDrive

Το HoneyDrive εγκαταστάθηκε σε εικονικό μηχάνημα στο https://okeanos.grnet.gr/, μία cloud υπηρεσία του Εθνικού Δικτύου Έρευνας και Τεχνολογίας – ΕΔΕΤ που παρέχει τη δυνατότητα δημιουργίας εικονικών μηχανημάτων και δικτύων, καθώς και φιλοξενίας αρχείων.

Ο Okeanos αποτελείται από δύο υπηρεσίες, το Pithos+ και το Cyclades. Το Pithos+ αφορά στη φιλοξενία αρχείων. Είναι ο εικονικός δίσκος του okeanos. Παρέχει τη δυνατότητα διαδικτυακής αποθήκευσης αρχείων, την κοινή χρήση τους και την απομακρυσμένη πρόσβαση τους.

Το Cyclades αφορά στη δημιουργία εικονικών μηχανημάτων. Ο χρήστης μπορεί να διαλέξει κατά τη δημιουργία λειτουργικό σύστημα (FreeBSD, OpenSUSE, Windows Server, Centos, Fedora, Kubuntu, Ubuntu, Debian ή και οποιαδήποτε άλλη custom image λειτουργικού), αριθμό πυρήνων επεξεργαστή (1-4), μνήμη (512 Mb – 8Gb) και χωρητικότητα (5 – 100 Gb). Η δημιουργία του μηχανήματος διαρκεί μερικά λεπτά και η πρόσβαση στο μηχάνημα γίνεται μέσω ssh, μέσω του ίδιου του site και μέσω ειδικών προγραμμάτων απομακρυσμένης πρόσβασης, όπως το X2Go. Η υπηρεσία επιτρέπει τη δημιουργία μέχρι δύο εικονικών μηχανημάτων και τη σύνδεσή τους σε ένα εικονικό δίκτυο.

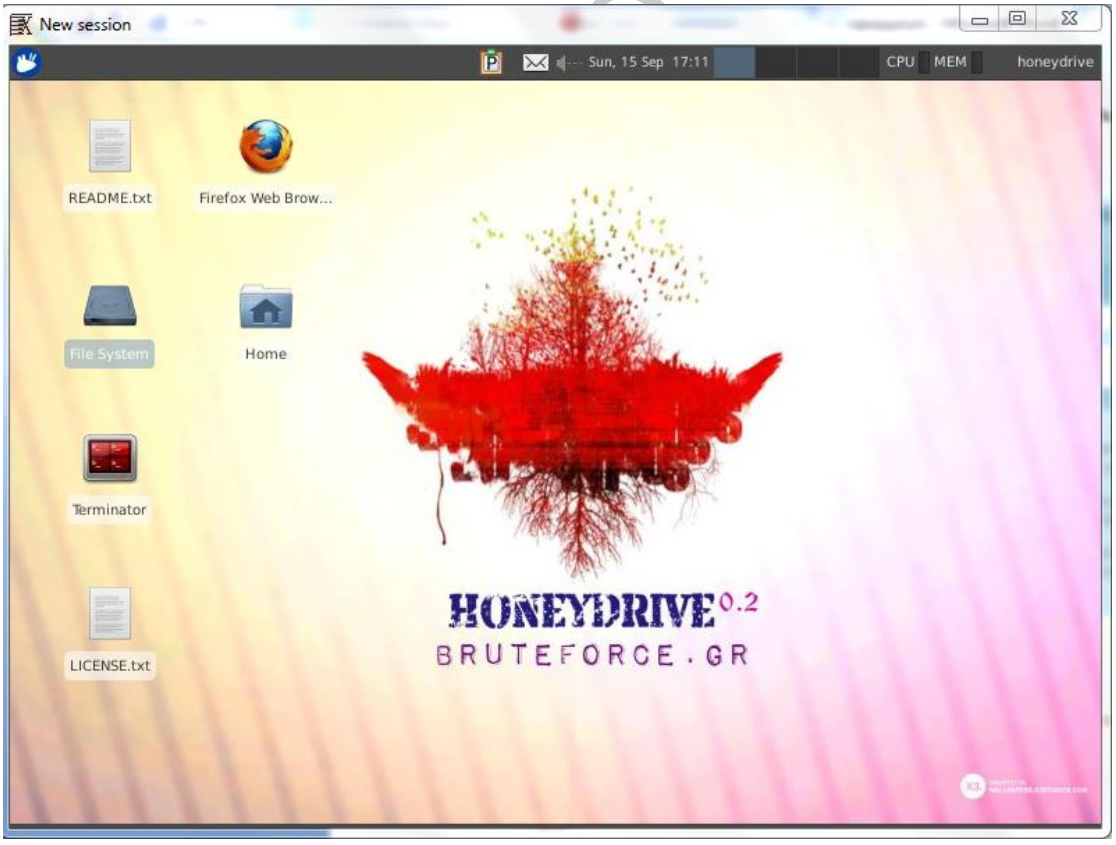
Ο Okeanos απευθύνεται σε ακαδημαϊκά μέλη και φοιτητές.

10.1.1 Εγκατάσταση

Οδηγίες εγκατάστασης του HoneyDrive περιγράφονται στο παράρτημα Π1.

Το HoneyDrive εγκαταστάθηκε σε ένα μηχάνημα με τετραπύρηνο επεξεργαστή, με μνήμη 4096 Mb και με χωρητικότητα 60 Gb. Το λειτουργικό του είναι XUbuntu Desktop 12.04 32-bit. Η ip διεύθυνσή του είναι 83.212.124.134. Στην εικόνα 10-1 από το site του okeanos διαφαίνονται τα χαρακτηριστικά του εικονικού μηχανήματος, ονόματι HoneyDrive και στην εικόνα 10-2 διαφαίνεται η επιφάνεια εργασίας του HoneyDrive.

Εικόνα 10-1: Χαρακτηριστικά εικονικού μηχανήματος HoneyDrive.



Εικόνα 10-2: Επιφάνεια εργασίας HoneyDrive.

10.2 Kippo

Το HoneyDrive αποτελεί σουίτα από Honeypots, επομένως απαιτείται ελάχιστη ή και καθόλου παραμετροποίηση για την εκτέλεση των Honeypots που περιέχει. Στην περίπτωση του Kippo, απαιτήθηκε μόνο η εκτέλεση του script start.sh από τον κατάλογο του Kippo, ώστε να ξεκινήσει η λειτουργία του. Ωστόσο για λόγους κατανόησης λειτουργικότητας του Kippo, στο παρακάτω κεφάλαιο θα αναλυθούν χαρακτηριστικά εγκατάστασης και λειτουργίας του Kippo.

10.2.1 Εγκατάσταση

Το Kippo εξομοιώνει έναν SSH διακομιστή. Εξ ορισμού, ακούει στη θύρα 2222 για συνδέσεις, ωστόσο για να λειτουργήσει σαν ένας SSH διακομιστής, απαιτείται η αλλαγή της θύρας αυτής από 2222 σε 22 και αντίστοιχα η αλλαγή του πραγματικού SSH διακομιστή του μηχανήματος από 22 σε οτιδήποτε άλλο, (στην προκειμένη 2222). Έτσι, οι εισβολείς αναζητώντας την προεπιλεγμένη θύρα 22 των SSH διακομιστών θα συνδέονται στη θύρα 22 του Kippo.

Απαιτείται επομένως αλλαγή της θύρας από 22 σε 2222 στο αρχείο ρυθμίσεων του SSH διακομιστή που βρίσκεται στον κατάλογο /etc/ssh/sshd_config και στη συνέχεια επανεκκίνηση του διακομιστή με χρήση της εντολής /etc/init.d/ssh restart.

Αναλυτικές οδηγίες εγκατάστασης και παραμετροποίησης του Kippo περιγράφονται στο παράρτημα Π2.

Κατόπιν εγκατάστασης, τα αρχεία του Kippo βρίσκονται στον κατάλογο /home/kippo. Παρακάτω περιγράφονται οι φάκελοι και τα αρχεία που δημιουργήθηκαν.

- dl: Εδώ αποθηκεύονται τα αρχεία που οι εισβολείς λαμβάνουν στο Honeypot με χρήση της εντολής wget.
- log/kippo.log: Αρχείο καταγραφής του Kippo.
- log/tty: Εδώ αποθηκεύονται τα αρχεία καταγραφής συνεδριών, (session logs), δηλαδή οι αλληλεπιδράσεις των εισβολέων με το τερματικό του προγράμματος.
- utils/playlog.py: Script με δυνατότητα αναπαραγωγής των session logs.
- fs.pickle: Ψεύτικο σύστημα αρχείων του Honeypot.
- honeyfs: Εδώ βρίσκονται κάποια πρόσθετα περιεχόμενα του ψεύτικου συστήματος αρχείων, όπως για παράδειγμα ένα ψεύτικο αρχείο κωδικών passwd, τα οποία ένας εισβολέας αφού αποκτήσει πρόσβαση στο Honeypot, μπορεί να προσπελάσει.
- data/userdb.txt: Εδώ βρίσκονται όλοι οι πιθανοί συνδυασμοί ονόματος χρήστη και Honeypot, με τους οποίους ένας εισβολέας μπορεί να αποκτήσει πρόσβαση στο Honeypot. Ο προεπιλεγμένος συνδυασμός είναι root και 123456. Εάν κάποιος επιτιθέμενος με χρήση της εντολής passwd αλλάξει τον κωδικό πρόσβασης, ο κωδικός αυτός προστίθεται αυτόματα στο εν λόγω αρχείο. [57]

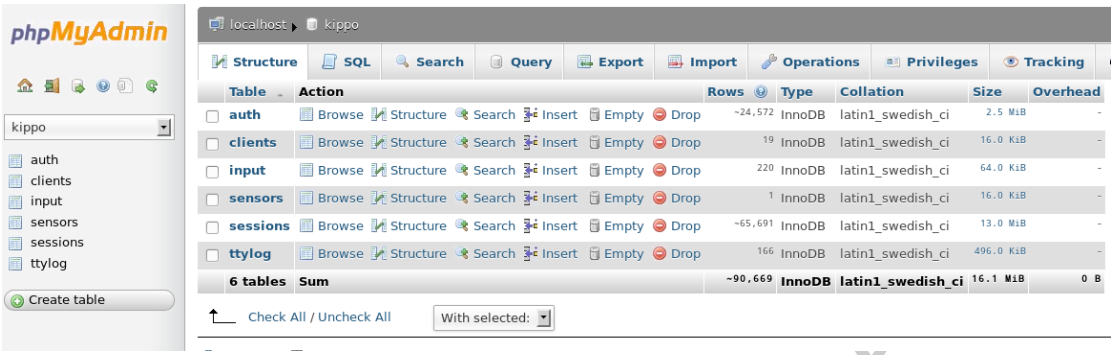
Στη συνέχεια, απαιτείται η δημιουργία ενός χρήστη και μίας βάσης δεδομένων για την αποθήκευση όλων των δεδομένων που καταγράφει το Kippo, διαδικασία η οποία περιγράφεται στο παράρτημα Π2.

Οι πίνακες της βάσης δεδομένων kippo που δημιουργήθηκαν είναι οι παρακάτω.

- auth: εδώ καταγράφονται όλες οι απόπειρες σύνδεσης στον kippo ssh διακομιστή
- clients: εδώ καταγράφονται οι διαφορετικοί SSH πελάτες που χρησιμοποιούνται για τη σύνδεση στον kippo ssh διακομιστή
- input: εδώ καταγράφονται όλες οι εντολές που πληκτρολογεί ένας εισβολές αφού έχει εισέλθει στο Honeypot
- sensors: εδώ αποθηκεύονται τα Honeypots που έχουμε δημιουργήσει, στην προκειμένη περίπτωση δηλαδή έχουμε ένα Honeypot
- sessions: εδώ καταγράφονται όλες οι συνεδρίες που έχουν δημιουργηθεί με τον kippo ssh διακομιστή
- ttylog: εδώ καταγράφονται, σε δυαδική μορφή τα δεδομένα καταγραφής του τερματικού του Honeypot.

Με την εντολή ./start.sh από τον κατάλογο του Kippo, εκκινείται η λειτουργία του Kippo. [57]

Στην παρακάτω εικόνα διαφαίνεται ένα στιγμιότυπο της βάσης δεδομένων του Kippo μέσα από την πλατφόρμα του phpmyadmin.



Εικόνα 10-3: Πλατφόρμα PhpMyAdmin, στιγμιότυπο βάσης δεδομένων Kippo.

10.2.2 Ανάλυση δεδομένων

Το Kippo SSH Honeypot λειτούργησε το διάστημα 12 Ιουνίου – 12 Σεπτεμβρίου. Κατά το διάστημα αυτό δέχτηκε 24572 αιτήματα σύνδεσης από 19 διαφορετικούς SSH πελάτες και κατέγραψε 65691 συνεδρίες, 220 εντολές από επιτιθέμενους και 166 συνεδρίες σε δυαδική μορφή. Τα αιτήματα σύνδεσης προήλθαν από 303 μοναδικές διευθύνσεις IP και τα 185 από αυτά κατέληξαν σε επιτυχή σύνδεση.

Τα δεδομένα οπτικοποιούνται με χρήση Google Charts και του εργαλείου Kippo-Graph. [55] Είναι γραμμένο σε php και χρησιμοποιεί τη βιβλιοθήκη Libchart. Εκτελεί επερωτήσεις στη βάση δεδομένων του Kippo και τα αποτελέσματα τα οπτικοποιεί με χρήση της προαναφερθείσας βιβλιοθήκης. Είναι προεγκατεστημένο στη σουίτα του HoneyDrive.

Ο κώδικας για τη δημιουργία των διαγραμμάτων με χρήση Google Charts εμπεριέχεται στο παράρτημα (Π5).

10.2.3 Ονόματα χρήστη – κωδικοί

Με χρήση του εργαλείου pipal, πραγματοποιήθηκε ανάλυση των ονομάτων χρήστη και των κωδικών που χρησιμοποιήθηκαν από τους εισβολείς και ελήφθησαν τα παρακάτω αποτελέσματα.

Top 10 ονομάτων χρήστη	Πλήθος εμφανίσεων (%)
Root	14846 (61.63%)
bin	369 (1.53%)
Test	276 (1.15%)
oracle	246 (1.02%)
Nagios	114 (0.47%)
postgres	98 (0.41%)
tomcat	96 (0.4%)
Guest	91 (0.38%)
user	82 (0.34%)
Admin	59 (0.24%)

Πίνακας 10-1: Top 10 ονομάτων χρήστη, (Kippo).

Top 10 κωδικών	Πλήθος εμφανίσεων (%)
123456	384 (1.6%)
password	143 (0.59%)
12345	78 (0.32%)
111111	78 (0.32%)
cacutza	77 (0.32%)
1qaz2wsx	75 (0.31%)
1234	74 (0.31%)
branburica	68 (0.28%)
oracle	67 (0.28%)
abc123	65 (0.27%)

Πίνακας 10-2: Top 10 κωδικών, (Kippo).

Ακολουθούν οι δέκα κορυφαίοι συνδυασμοί ονόματος χρήστη και κωδικού.

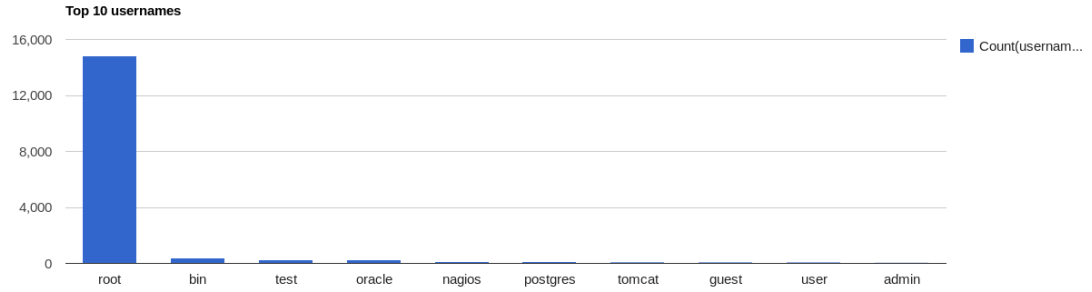
Top 10 συνδυασμοί ονομάτων χρήστη – κωδικών		Πλήθος εμφανίσεων
Όνομα χρήστη	Κωδικός	
Root	123456	178
Root	P@ssw0rd	92
Root	password	87
Root	passw0rd	69
Root	111111	68
Root	root	61
Root	12345	59
Root	qwerty	57
Oracle	oracle	56
Root	1234	55

Πίνακας 10-3: Top 10 συνδυασμοί ονομάτων χρήστη-κωδικών, (Kippo).

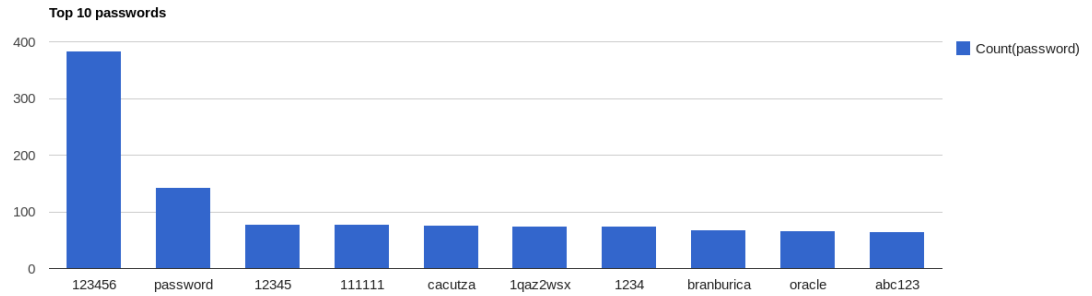
Ο προεπιλεγμένος συνδυασμός ονόματος χρήστη και κωδικού root – 123456 είναι και αυτός που χρησιμοποιήθηκε περισσότερο. Οι υπόλοιποι συνδυασμοί είναι παραλλαγές αυτού ενώ παρατηρήθηκαν συχνά και συνδυασμοί γραμμάτων πληκτρολογίου.

Όπως έχει ήδη αναφερθεί αφού κάποιος επιτιθέμενος συνδεθεί επιτυχώς στο σύστημα μπορεί με χρήση της εντολής passwd να αλλάξει τον κωδικό πρόσβασης και ο καινούριος κωδικός πρόσβασης καταχωρείται στο αρχείο userdb.txt. Στο εν λόγω αρχείο εκτός από τον προεπιλεγμένο συνδυασμό root – 123456 περιέχεται και ο συνδυασμός root – angelqwe123.

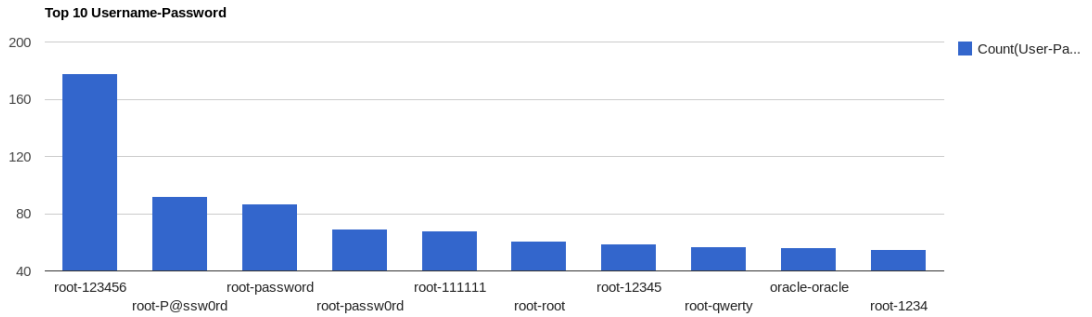
Με χρήση Google Charts τα παραπάνω δεδομένα αναπαρίστανται στα διαγράμματα που ακολουθούν.



Εικόνα 10-4: Top 10 ονομάτων χρήστη Kippo.



Εικόνα 10-5: Top 10 κωδικών Kippo.



Εικόνα 10-6: Top 10 συνδυασμοί ονομάτων χρήστη-κωδικών Kippo.

10.2.4 Επιθέσεις – Επιτυχίες, Αποτυχίες

Στον παρακάτω πίνακα παρατίθεται το πλήθος επιτυχημένων – αποτυχημένων συνδέσεων και τα ποσοστά τους ως προς το σύνολο των συνδέσεων.

Αποτυχίες	23922 (99,23%)
Επιτυχίες	185 (0,77%)

Πίνακας 10-4: Πλήθος επιτυχιών – αποτυχιών σε συνδέσεις προς το Kippo.

Το ποσοστό των επιτυχών συνδέσεων είναι εξαιρετικά χαμηλό.

Στην εικόνα 10-7 που σχεδιάστηκε με χρήση Google Charts αναπαρίστανται τα παραπάνω δεδομένα.

Εικόνα 10-7: Πλήθος επιτυχιών - αποτυχιών σε συνδέσεις προς το Kippo.

10.2.5 Επιθέσεις – Συχνότητα

Ακολουθούν οι 20 ημερομηνίες κατά τις οποίες σημειώθηκαν οι περισσότερες επιτυχείς συνδέσεις.

Πλήθος επιτυχών συνδέσεων	Ημερομηνία / Ώρα
11	2013-09-12 04:06:13
9	2013-07-02 04:22:07
7	2013-08-10 08:01:33
7	2013-08-02 00:05:09
7	2013-07-15 04:45:55
6	2013-08-09 02:09:48
5	2013-07-31 03:58:15
5	2013-08-21 02:11:46
5	2013-07-22 03:52:34
5	2013-07-23 06:13:36
5	2013-07-25 10:39:59
4	2013-08-20 16:56:21
4	2013-06-24 04:25:28
4	2013-07-13 06:08:46
4	2013-06-25 01:19:19
4	2013-08-04 05:34:05
4	2013-08-07 10:20:58
4	2013-06-20 11:19:12
3	2013-07-12 05:34:58
3	2013-06-15 07:59:31

Πίνακας 10-5: Top 20 ημερομηνιών-επιτυχών συνδέσεων, (Kippo).

Όπως διαφαίνεται δεν υπήρξε κάποια μέρα με εντυπωσιακά μεγαλύτερο αριθμό επιτυχών συνδέσεων συγκριτικά με τις υπόλοιπες, ενώ καμία μέρα δεν ξεπεράστηκαν οι 10 επιτυχείς συνδέσεις. Στις 12 Σεπτεμβρίου 2013 σημειώθηκαν οι περισσότερες επιτυχείς συνδέσεις με πλήθος 11 ενώ ακολουθούν 9 επιτυχείς συνδέσεις στις 2 Ιουλίου 2013.

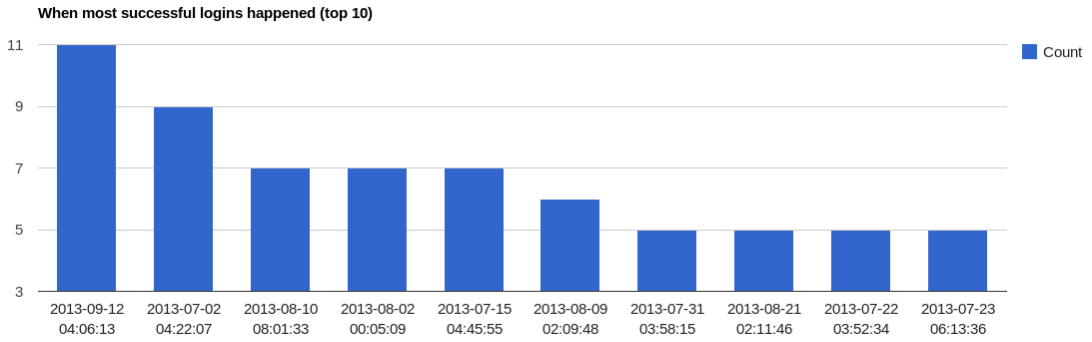
Ακολουθούν οι 20 ημερομηνίες κατά τις οποίες πραγματοποιήθηκαν οι περισσότερες επιθέσεις προς το Kippo.

Πλήθος επιθέσεων	Ημερομηνία / Ώρα
3600	2013-07-15 00:21:15
2044	2013-06-30 07:28:10
1431	2013-06-21 05:56:15
1020	2013-07-21 11:59:06
950	2013-09-10 17:28:27
916	2013-06-23 02:09:23
893	2013-08-29 03:45:21
859	2013-09-11 11:40:41
797	2013-08-18 00:39:41
710	2013-09-09 09:28:50
549	2013-07-31 03:43:05
544	2013-08-13 14:46:29
542	2013-06-24 04:25:21
487	2013-07-05 07:57:56
476	2013-09-12 04:04:51
407	2013-07-23 06:13:12
389	2013-06-18 02:28:22
366	2013-09-05 00:01:17
338	2013-06-22 09:05:22
334	2013-06-29 01:32:05

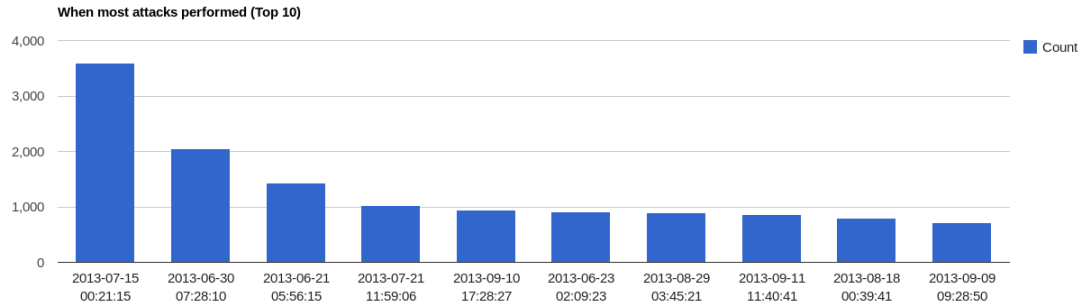
Πίνακας 10-6: Top 20 ημερομηνιών - επιθέσεων, (Kippo).

Όπως διαφαίνεται με ποσοστό 14,7% των συνολικών επιθέσεων σημειώθηκαν 3600 στις 15 Ιουλίου 2013, ενώ με ποσοστό μικρότερο του 10% των συνολικών επιθέσεων σημειώθηκαν 2044 επιθέσεις στις 30 Ιουνίου 2013.

Η εικόνα 10-8 παρουσιάζει τις ημερομηνίες κατά τις οποίες πραγματοποιήθηκαν οι περισσότερες επιτυχείς συνδέσεις προς το Kippo και η εικόνα 10-9 παρουσιάζει τις ημερομηνίες κατά τις οποίες σημειώθηκαν οι περισσότερες επιθέσεις προς το Kippo. Και τα δύο διαγράμματα σχεδιάστηκαν με χρήση Google Charts.

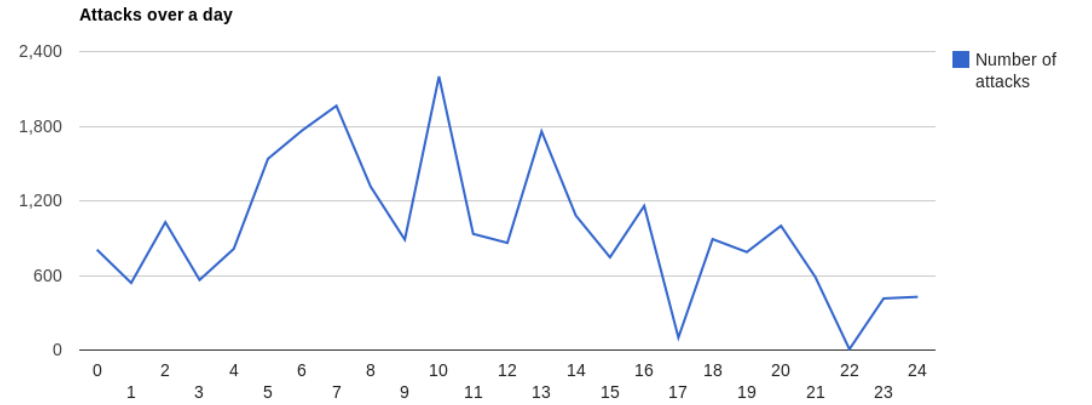


Εικόνα 10-8: Top 10 ημερομηνιών – επιτυχών συνδέσεων, (Kippo).



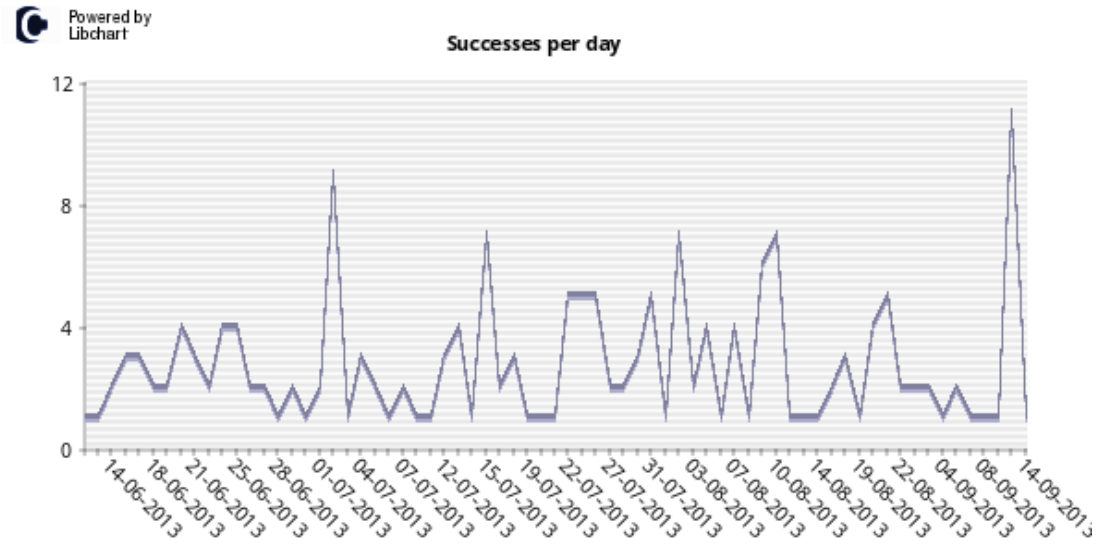
Εικόνα 10-9: Top 10 ημερομηνιών – επιθέσεων, (Kippo).

Ακολουθεί η αναπαράσταση του πλήθους των επιθέσεων που έλαβαν χώρα ανά ώρα κατά τη διάρκεια ενός 24ώρου σε διάγραμμα με χρήση Google Charts (εικόνα 10-10).

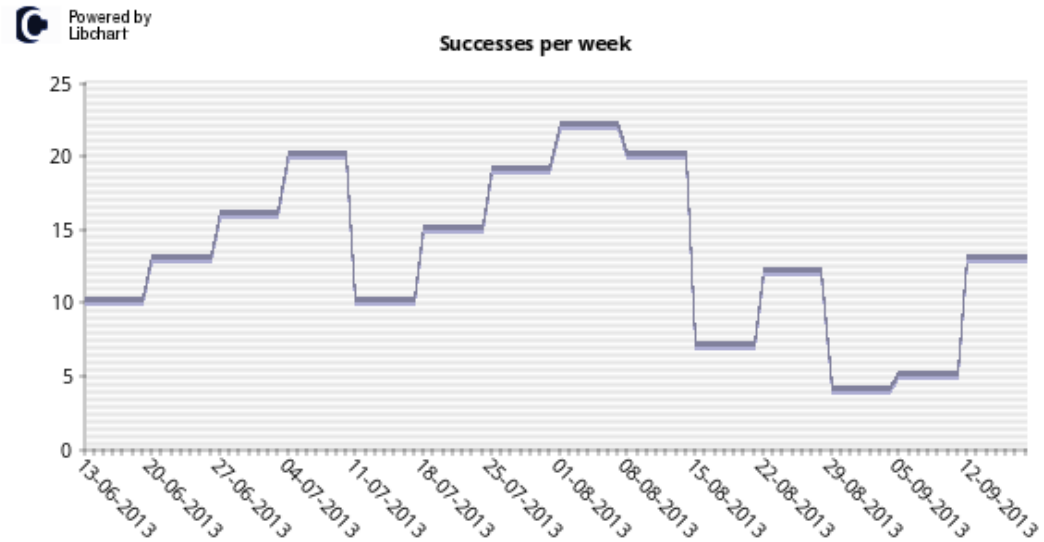


Εικόνα 10-10: Πλήθος επιθέσεων στη διάρκεια 24ωρου, (Kippo).

Στα παρακάτω διαγράμματα του Kippo-Graph διαφαίνεται στο μεν πρώτο η διακύμανση των καθημερινών επιτυχών συνδέσεων (εικόνα 10-11) και στο δεύτερο η διακύμανση των επιτυχών συνδέσεων σε εβδομαδιαία βάση (εικόνα 10-12). Ημέρες κατά τις οποίες σημειώθηκαν μηδενικές επιτυχείς συνδέσεις δε λαμβάνονται υπόψη.



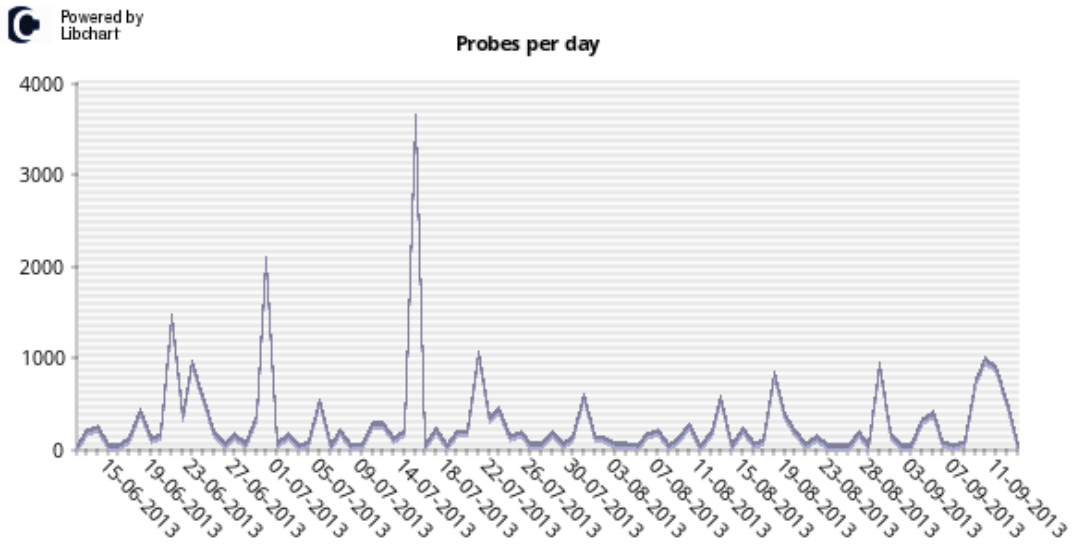
Εικόνα 10-11: Διακύμανση επιτυχών συνδέσεων σε καθημερινή βάση, (Kippo).



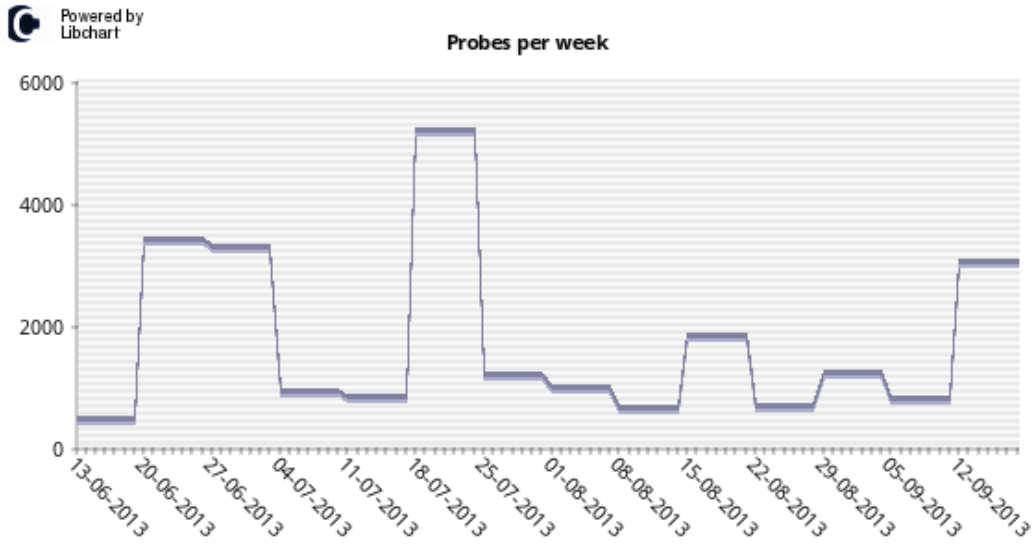
Εικόνα 10-12: Διακύμανση επιτυχών συνδέσεων σε εβδομαδιαία βάση, (Kippo).

Στα παρακάτω διαγράμματα του Kippo-Graph διαφαίνεται στο μεν πρώτο η διακύμανση των επιθέσεων σε καθημερινή βάση, (εικόνα 10-13), και στο δεύτερο η διακύμανση των επιθέσεων σε εβδομαδιαία βάση, (εικόνα 10-14).

Όπως παρατηρείται, δεν μπορεί να εξαχθεί κάποιο ασφαλές συμπέρασμα για τη συσχέτιση μεταξύ επιτυχών συνδέσεων και πλήθους επιθέσεων.



Εικόνα 10-13: Διακύμανση επιθέσεων σε καθημερινή βάση, (Kippo).



Εικόνα 10-14: Διακύμανση επιθέσεων σε εβδομαδιαία βάση, (Kippo).

10.2.6 Επιθέσεις – Προέλευση

Ακολουθούν οι 20 διευθύνσεις IP από τις οποίες προήλθαν οι περισσότερες επιθέσεις, η προέλευσή τους και το πλήθος των επιθέσεων. Να σημειωθεί ότι η προέλευση των επιθέσεων δεν είναι ακριβής καθώς η αναγνώριση των ενδιάμεσων, (proxy), διακομιστών δεν ήταν δυνατή.

Διεύθυνση IP	Χώρα	Πλήθος επιθέσεων, (%)
46.105.104.174	Γαλλία	24198, (49,8%)
31.193.140.28	Αγγλία	8678, (17,3%)
64.206.128.5	ΗΠΑ	3992, (15,9%)

61.139.54.71	Κίνα	2184, (7%)
209.105.248.190	ΗΠΑ	1438, (15,9%)
206.245.180.111	ΗΠΑ	1028, (15,9%)
212.110.129.114	Ουκρανία	953, (1,9%)
177.189.241.74	Βραζιλία	855, (1,7%)
106.240.236.226	Ν. Κορέα	823, (3,1%)
220.164.144.135	Κίνα	821, (7%)
84.246.227.10	Γαλλία	796, (49,8%)
211.53.199.6	Ν. Κορέα	713, (3,1%)
173.242.124.200	ΗΠΑ	629, (15,9%)
61.238.156.110	Χονγκ Κονγκ	523, (1%)
96.57.19.212	ΗΠΑ	490, (15,9%)
223.5.3.200	Κίνα	489, (7%)
203.211.143.185	Σιγκαπούρη	423, (0,8%)
66.199.146.126	Καναδάς	407, (0,8%)
74.118.194.220	ΗΠΑ	382, (15,9%)
83.212.113.123	Ελλάδα	381, (0,8%)

Πίνακας 10-7: Top 20 διευθύνσεων IP - χωρών επιθέσεων, (Kippo).

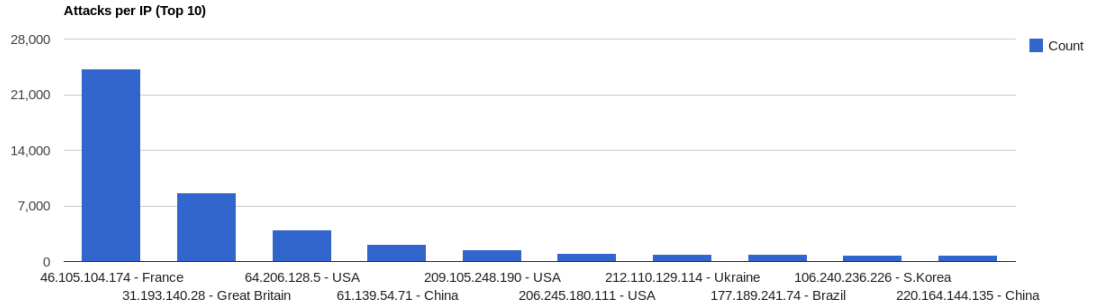
Ακολουθούν οι 10 διευθύνσεις IP από τις οποίες προήλθαν οι περισσότερες επιτυχείς συνδέσεις, η προέλευσή τους και το πλήθος τους.

Διεύθυνση IP	**Χώρα**	**Πλήθος επιτυχών συνδέσεων**
219.235.126.174	Κίνα	9, (38,8%)
94.156.12.144	Βουλγαρία	5, (10,4%)
183.232.32.24	Κίνα	5, (38,8%)
5.39.81.27	Γαλλία	5, (11,9%)
173.242.124.200	ΗΠΑ	4, (13,4%)
86.124.234.118	Ρουμανία	4, (10,4%)
86.124.233.50	Ρουμανία	3, (10,4%)
14.139.229.37	Ινδία	3, (6,8%)
223.5.3.200	Κίνα	3, (38,8%)
200.98.64.146	Βραζιλία	3, (6,8%)

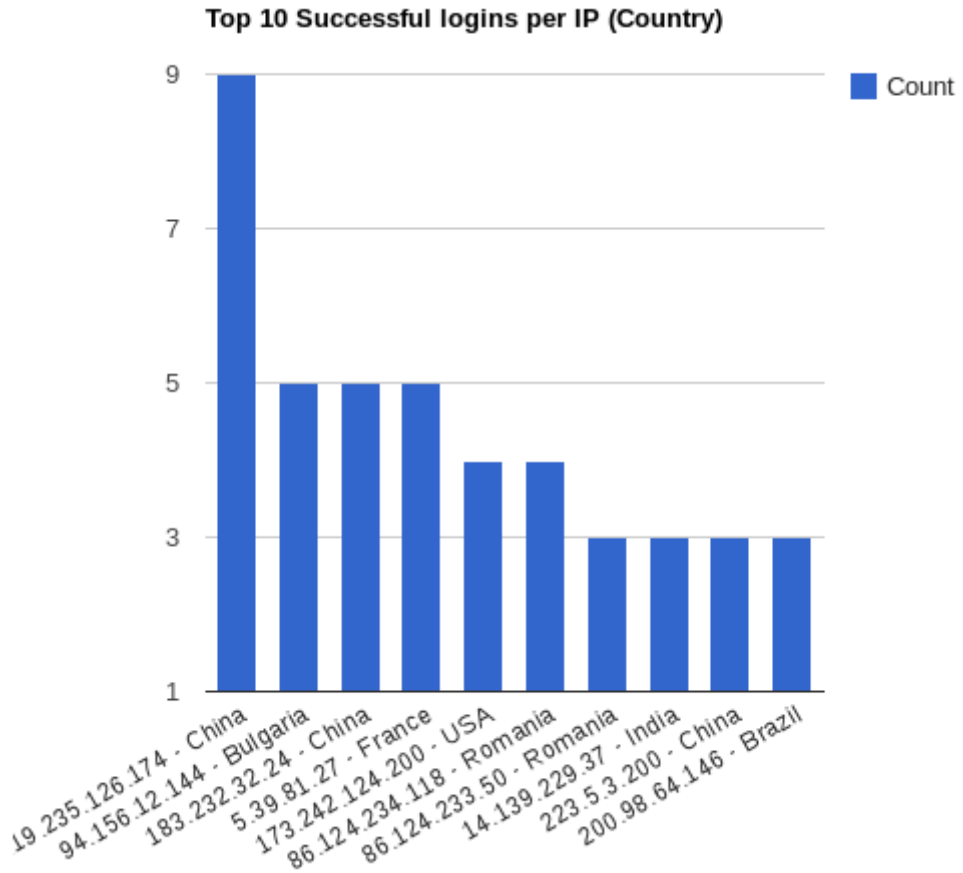
Πίνακας 10-8: Top 10 διευθύνσεων IP - χωρών επιτυχών συνδέσεων, (Kippo).

Όπως διαφαίνεται δεν μπορεί να εξαχθεί κάποιο ασφαλές συμπέρασμα για τη συσχέτιση μεταξύ πλήθους επιθέσεων και επιτυχών συνδέσεων ανά διεύθυνση IP ή και ανά χώρα.

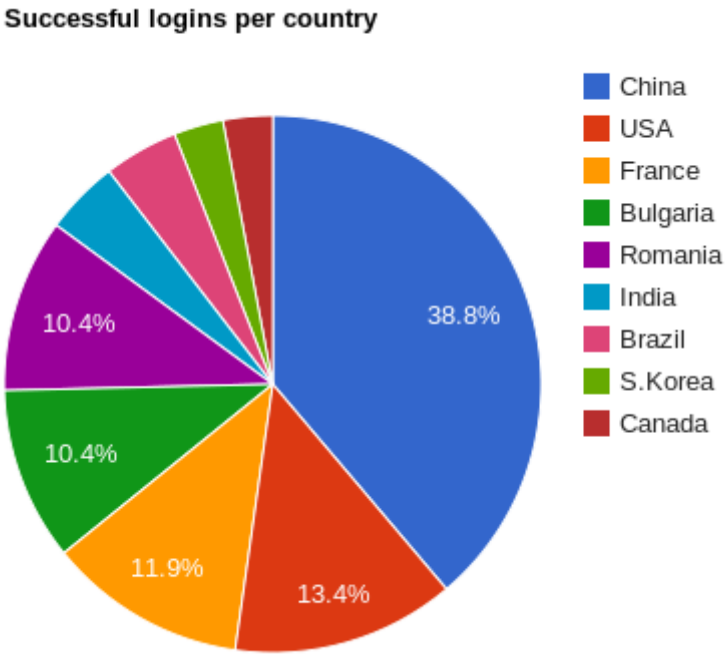
Στις εικόνες 10-15 και 10-18 διαφαίνονται αντίστοιχα οι 10 κορυφαίες διευθύνσεις IP που σημείωσαν τις περισσότερες επιθέσεις προς το Kippo και τα ποσοστά επί του συνόλου των επιθέσεων ανά χώρα. Στις εικόνες 10-16 και 10-17 διαφαίνονται αντίστοιχα οι 10 κορυφαίες διευθύνσεις IP που σημείωσαν τις περισσότερες επιτυχείς συνδέσεις προς το Kippo και τα ποσοστά επί του συνόλου των επιτυχών συνδέσεων ανά χώρα. Τα διαγράμματα αυτά σχεδιάστηκαν με χρήση Google Charts.



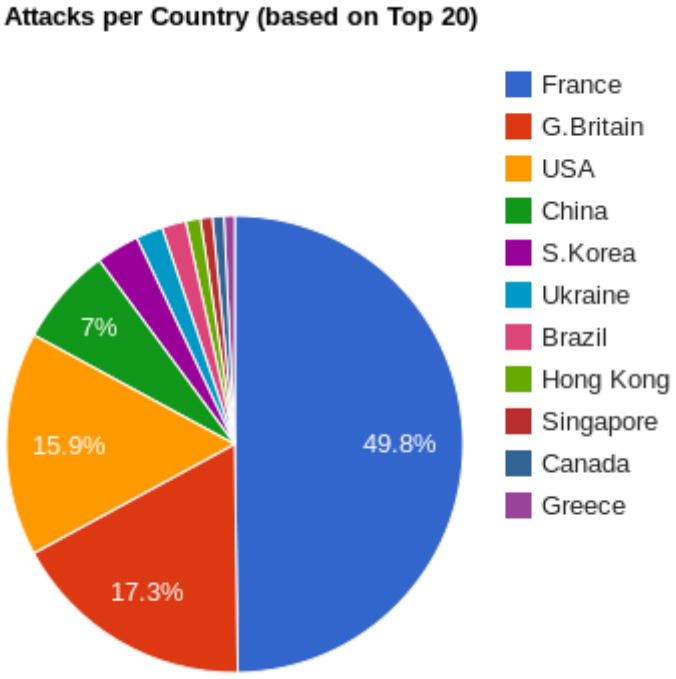
Εικόνα 10-15: Top 10 επιθέσεων ανά διεύθυνση IP, (Kippo).



Εικόνα 10-16: Top 10 επιτυχών συνδέσεων ανά διεύθυνση IP και χώρα, (Kippo).



Εικόνα 10-17: Ποσοστό επιτυχών συνδέσεων ανά χώρα, (Kippo).



Εικόνα 10-18: Ποσοστό επιθέσεων ανά χώρα, (Kippo).

Ο παρακάτω χάρτης του Kippo-Graph, (εικόνα 10-19), παρουσιάζει τη γεωγραφική κατανομή των 10 IP διευθύνσεων από τις οποίες προήλθαν οι περισσότερες επιθέσεις.

Εικόνα 10-19: Γεωγραφική κατανομή top 10 χωρών επιθέσεων, (Kippo).

10.2.7 Εντολές

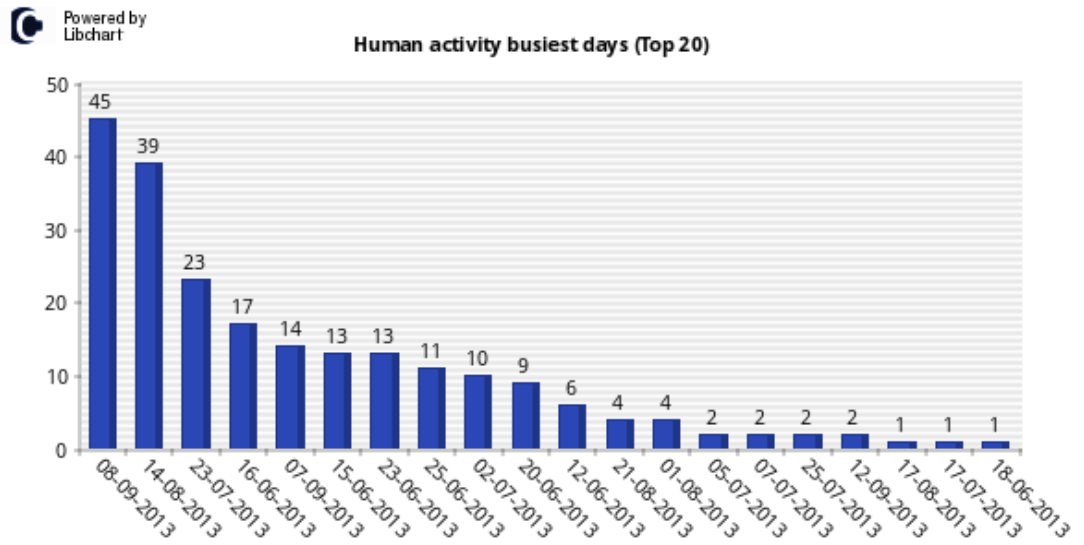
Συνολικά καταγράφηκαν 220 εντολές στη βάση δεδομένων του Kippo από τις οποίες οι 116 είναι μοναδικές.

Ακολουθούν οι 20 εντολές που χρησιμοποιήθηκαν περισσότερο από τους επιτιθέμενους που συνδέθηκαν στο Kippo.

Εντολές	Πλήθος
W	17
Ls	13
Ifconfig	7
Exit	6
ps x	6
rm -rf .bash_history	6
chmod +x *	5
echo "WinSCP: this is end-of-file:0"	5
cat /proc/cpuinfo	5
unset HISTFILE	4
unset HISTORY	4
uname –a	3
cd /usr/local/games	2
Cd	2
cd csservers_redirecte_linux_hlds	2
cd /dev/shm	2
./start	2
angelqwe123	2
wget w0rmer.altervista.org/m.txt	2
perl m.txt	2

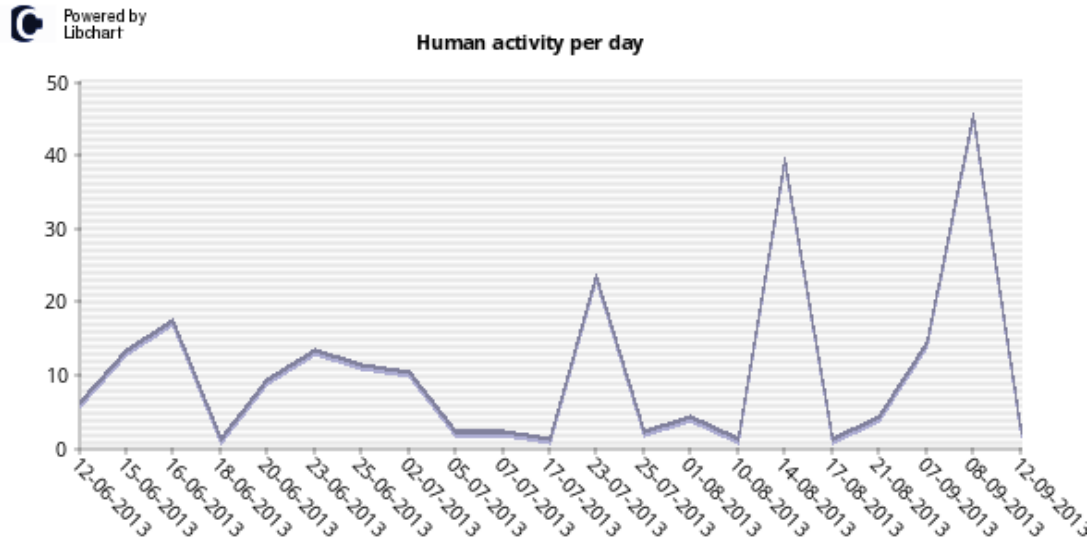
Πίνακας 10-9: Top 20 εντολών εισβολέων, (Kippo).

Η εντολή w έχει χρησιμοποιηθεί τις περισσότερες φορές (17), ενώ ακολουθεί η εντολή ls με 13 φορές. Η εντολή w παρέχει πληροφορίες συστήματος για τους χρήστες που είναι συνδεδεμένοι στο σύστημα και για τις διεργασίες που εκτελούν. Η εντολή ls εμφανίζει τα περιεχόμενα του τρέχοντος καταλόγου.

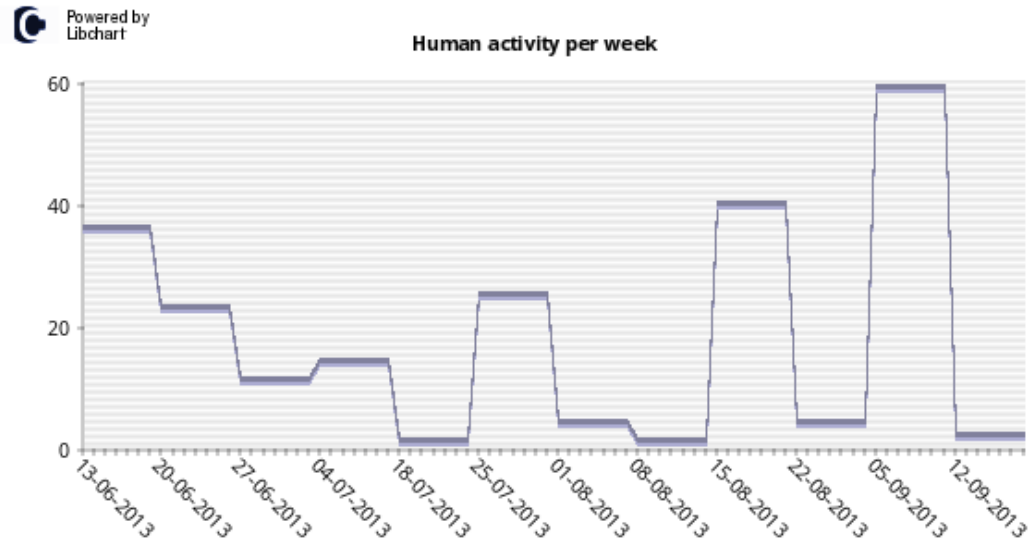


Εικόνα 10-20: Ημερομηνίες κατά τις οποίες σημειώθηκε η περισσότερη ανθρώπινη δραστηριότητα, (Kippo).

Το διάγραμμα 10-20 του Kippo-Graph παρουσιάζει τις 20 ημέρες κατά τις οποίες παρατηρήθηκε η μεγαλύτερη ανθρώπινη δραστηριότητα στο Honeypot, βάσει των εντολών που καταγράφηκαν στη βάση δεδομένων. Ακολουθούν διαγράμματα του Kippo-Graph που παρουσιάζουν τη διακύμανση της ανθρώπινης δραστηριότητας σε καθημερινή και εβδομαδιαία βάση, εικόνες 10-21 και 10-22 αντίστοιχα.



Εικόνα 10-21: Διακύμανση ανθρώπινης δραστηριότητας σε καθημερινή βάση, (Kippo).



Εικόνα 10-22: Διακύμανση ανθρώπινης δραστηριότητας σε εβδομαδιαία βάση, (Kippo).

10.3 Dionaea

Το Dionaea είναι προεγκατεστημένο στη σουίτα του HoneyDrive, επομένως για την εκτέλεσή του απαιτείται μηδαμινή παραμετροποίηση. Ωστόσο για λόγους κατανόησης λειτουργικότητας του Dionaea, στο παρακάτω κεφάλαιο θα αναλυθούν χαρακτηριστικά εγκατάστασης και λειτουργίας του Dionaea.

10.3.1 Εγκατάσταση

Η εγκατάσταση του Dionaea είναι ιδιαίτερα χρονοβόρα διαδικασία καθώς απαιτεί την εγκατάσταση πολλών πακέτων και πολλών βιβλιοθηκών.

Αναλυτικές οδηγίες εγκατάστασης περιγράφονται στο παράρτημα Π3.

Παρακάτω περιγράφονται οι φάκελοι και τα αρχεία που δημιουργήθηκαν κάτω από τον κατάλογο /opt/dionaea.

- bin: Εδώ βρίσκεται το εκτελέσιμο αρχείο του Dionaea.
- etc/dionaea: Εδώ βρίσκεται το αρχείο ρυθμίσεων του Dionaea, (dionaea.conf).
- var/log: Εδώ βρίσκονται τα αρχεία καταγραφής του Dionaea.
- var/dionaea: Εδώ βρίσκεται η SQLite βάση δεδομένων, στην οποία καταγράφονται όλα τα δεδομένα που αφορούν στις συνδέσεις με το Honeypot.
- var/dionaea/binaries: Εδώ αποθηκεύονται όλα τα αρχεία που θεωρούνται κακόβουλου περιεχομένου και λαμβάνονται από άλλους υπολογιστές που έχουν συνδεθεί στο Honeypot.
- var/dionaea/bistreams: Εδώ αποθηκεύονται σε μορφή ροής δεδομένων όλες οι συνεδρίες που εγκαθιδρύονται από και προς το Honeypot.

10.3.2 Εγκατάσταση πρόσθετων εργαλείων – Παραμετροποίηση

Ενδείκνυται η εγκατάσταση του εργαλείου p0f ώστε για κάθε σύνδεση να αναγνωρίζεται το λειτουργικό σύστημα του επιτιθέμενου.

Όπως έχει ήδη αναφερθεί, το Dionaea αφού λάβει ένα αρχείο κακόβουλου περιεχομένου, το αποθηκεύει τοπικά και παρέχεται η δυνατότητα αποστολής του αρχείου διαδικτυακά σε υπηρεσίες που πραγματοποιούν ανάλυση κακόβουλου λογισμικού, όπως VirusTotal, Anubis, Norman SandBox και CWSandbox. Η υπηρεσία VirusTotal απαιτεί από το χρήστη τη χρήση ενός κλειδιού (key) API το οποίο παρέχεται στο χρήστη άμεσα με την εγγραφή του στην υπηρεσία. Κατά την παραμετροποίηση, όπως θα αναλυθεί και παρακάτω, απαιτείται η εισαγωγή του κλειδιού αυτού στο αρχείο παραμέτρων του Dionaea.

Επιπρόσθετα, απαιτείται και η εγκατάσταση του εργαλείου phpliteadmin για την άμεση πρόσβαση στην SQLite βάση δεδομένων του Dionaea. Το phpliteadmin έχει παρόμοια λειτουργικότητα με το phpmyadmin.

Αναλυτικές οδηγίες παραμετροποίησης του Dionaea παρατίθενται στο παράρτημα Π3.

Για την εκκίνηση του p0f απαιτείται η εντολή

p0f -i any -Q /tmp/p0f.sock -l -d -o /var/log/p0f.log

Για την εκκίνηση του Dionaea απαιτείται η εντολή

/opt/dionaea/bin/dionaea -D -l all,-debug -L '*' -p /opt/dionaea/var/run/dionaea.pid

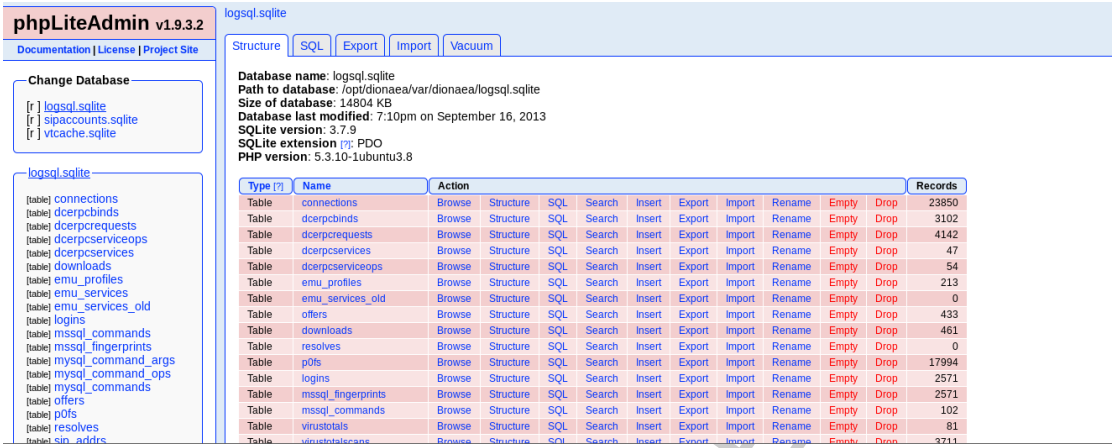
[66]

10.3.3 Ανάλυση δεδομένων

Το Dionaea Honeypot λειτούργησε το διάστημα 16 Ιουνίου – 10 Οκτωβρίου 2013. Κατά το διάστημα αυτό πραγματοποιήθηκαν 40312 συνδέσεις προς το Honeypot εκ των οποίων οι 21036 ήταν επιτυχημένες (52%) και οι υπόλοιπες 19276 απορρίφθηκαν, (48%). Καταγράφηκαν 552 απόπειρες λήψης αρχείων από επιτιθέμενους και ελήφθησαν 128 μοναδικά αρχεία ενώ

στάλθηκαν 107 αρχεία κακόβουλου περιεχομένου προς την υπηρεσία ανάλυσης κακόβουλου λογισμικού VirusTotal. Η βάση δεδομένων συνολικά έφτασε τα 14,5 Mb σε μέγεθος.

Στην εικόνα 10-23 διαφαίνεται ένα στιγμιότυπο της βάσης δεδομένων του Dionaea μέσα από την πλατφόρμα του phpliteadmin.



Εικόνα 10-23: Πλατφόρμα PhpLiteAdmin, στιγμιότυπο βάσης δεδομένων Dionaea.

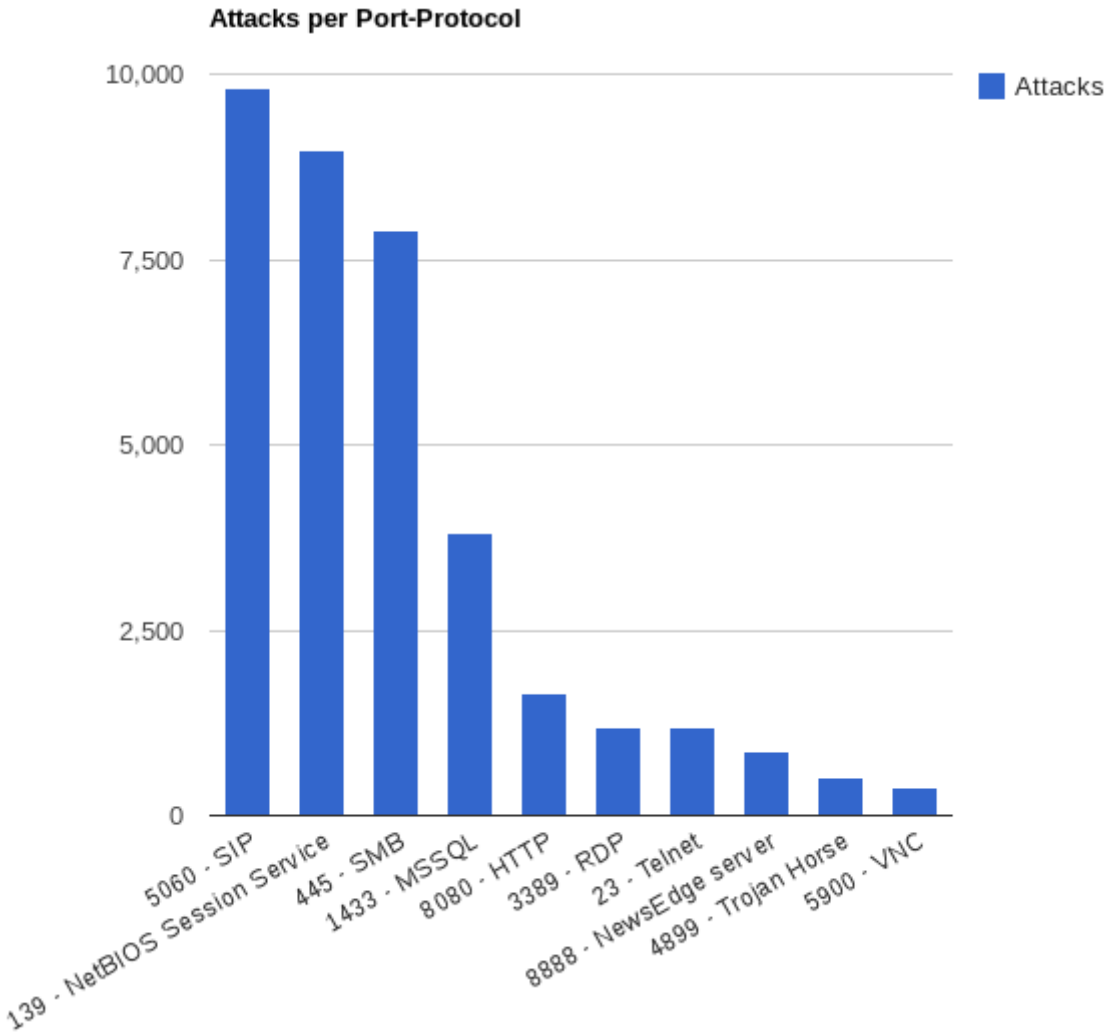
10.3.4 Επιθέσεις ανά πρωτόκολλο

Στον παρακάτω πίνακα παρουσιάζεται το πλήθος επιθέσεων ανά πρωτόκολλο.

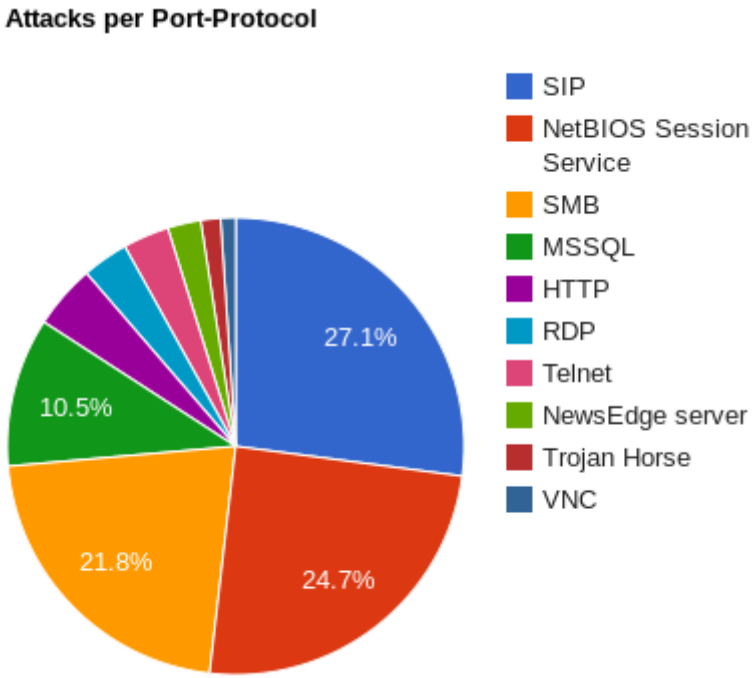
Θύρα	Πλήθος επιθέσεων, (%)
5060 (SIP)	9822, (27,1%)
139 (NetBIOS Session Service)	8970, (24,7%)
445 (SMB)	7898, (21,8%)
1433 (MSSQL)	3822, (10,5%)
8080 (HTTP)	1641, (4,5%)
3389 (Remote Desktop Protocol – RDP for Windows)	1184, (3,3%)
23 (Telnet)	1182, (3,3%)
8888 (NewsEdge server)	852, (2,3%)
4899 (Radmin remote administration tool – used as Trojan Horse)	508, (1,4%)
5900 (VNC remote desktop)	383, (1,1%)

Πίνακας 10-10: Πλήθος συνδέσεων ανά πρωτόκολλο, (Dionaea).

Τα παραπάνω αποτελέσματα αναπαρίστανται στα διαγράμματα που ακολουθούν, τα οποία σχεδιάστηκαν με χρήση Google Charts. Η εικόνα 10-24 δείχνει το πλήθος επιθέσεων ανά πρωτόκολλο και η εικόνα 10-25 δείχνει ποσοστιαία το πλήθος επιθέσεων ανά πρωτόκολλο.



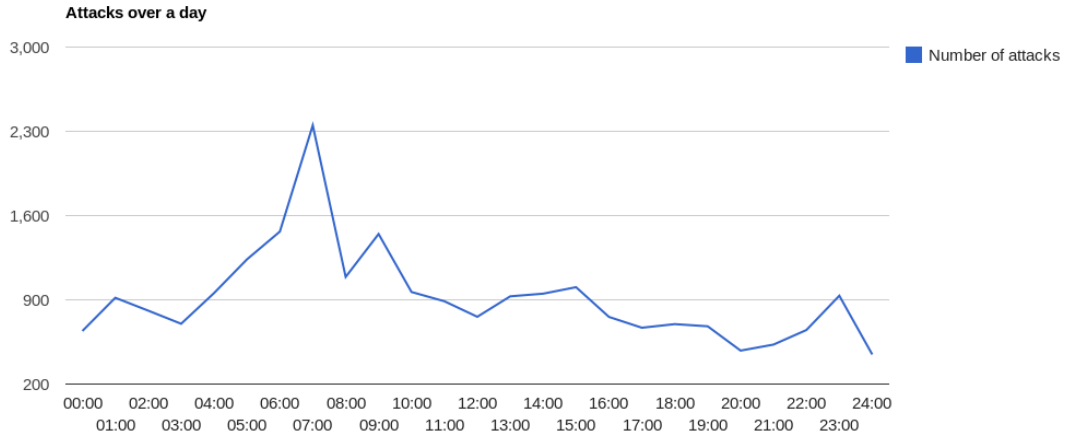
Εικόνα 10-24: Πλήθος επιθέσεων ανά πρωτόκολλο, (Dionaea).



Εικόνα 10-25: Πλήθος επιθέσεων ανά πρωτόκολλο (%), (Dionaea).

10.3.5 Πλήθος επιθέσεων ανά ώρα κατά τη διάρκεια μιας μέρας

Το διάγραμμα που ακολουθεί αναπαριστά το πλήθος επιθέσεων ανά ώρα στη διάρκεια ενός 24ώρου. Σχεδιάστηκε με χρήση Google Charts, (εικόνα 10-26).



Εικόνα 10-26: Πλήθος συνδέσεων ανά ώρα κατά τη διάρκεια 24ωρου, (Dionaea).

10.3.6 Λήψεις αρχείων κακόβουλου περιεχομένου

Συνολικά κατά την περίοδο λειτουργίας του Dionaea καταγράφηκαν 552 απόπειρες λήψης αρχείων από επιτιθέμενους και ελήφθησαν 128 μοναδικά αρχεία ενώ στάλθηκαν 107 αρχεία προς την υπηρεσία ανάλυσης κακόβουλου λογισμικού VirusTotal.

Όλα τα αρχεία ελήφθησαν μέσω του πρωτοκόλλου SMB πλην 46 αρχείων για τα οποία δεν μπορεί να προσδιοριστεί το πρωτόκολλο που χρησιμοποιήθηκε για τη λήψη ενώ 2 αρχεία ελήφθησαν μέσω TFTP.

Στον παρακάτω πίνακα περιέχονται τα 15 αρχεία που ελήφθησαν περισσότερο στο Dionaea και από διαφορετικές διευθύνσεις και στο σύνολο.

Id	md5_hash	IP / Λήψεις	Ταυτότητα – Ποσοστό αναγνώρισης
(1)	b43ad71209c5100b9ed71edb10041514	8 / 48	Gen:Trojan.Downloader.cmW@aaP5gLc (trojan) – 33%
(2)	64b4345a946bc9388412fedd53fb21cf	7 / 15	Win32.exe – 92%
(3)	9b175f5f727bcf1153e1aaf99798556a	7 / 7	Win32.exe – 96%
(4)	d41d8cd98f00b204e9800998ecf8427e	6 / 38	Δεν αναγνωρίστηκε. – 0%
(5)	4a6e5980ad7d1a4bbe71ec46fa96755e	3 / 4	Trojan.Win32.Generic – 92%
(6)	4d56562a6019c05c592b9681e9ca2737	3 / 7	Win32.exe – 90%
(7)	8b48f59fb263b1b3ed5f9f2a8cd8fd26	3 / 5	HackTool:Win32 – 47%
(8)	1e5da233df2b65238567c21ca89495ea	2 / 8	Win32.exe – 91%
(9)	23afd4ad06d26f6442fa06f4ec944513	2 / 8	Gen:Trojan.Downloader.cmW@aaP5gLc (trojan) – 36%
(10)	2eed9f6a3febd6a1c49f8edb5e60cf49	2 / 8	W32/Sality – 39%
(11)	30b5a9e46c38ffca2add7e4845a84fc1	1 / 214	Generic.Malware (trojan) / WIN.MAIL.WORM.Virus – 31%
(12)	533626e061c4cb83226f72282e1c74e3	1 / 8	Gen:Trojan.Downloader (trojan) (2) – 36%
(13)	9be443d09b25157fcfbccb953f4a2cd4	1 / 8	Gen:Trojan.Downloader (trojan) (2) – 33%
(14)	ccba8352737c90a655e1430bb1dce644	1 / 8	Win32.Virtob.8.Gen / W32/Virut-W – 24%
(15)	730498b8a6c676e2298d9b1ad7dd5d10	1 / 6	Win32.exe (4) – 91%

Πίνακας 10-11: Top 15 ληφθέντων αρχείων κακόβουλου περιεχομένου, (Dionaea).

10.3.7 Ανάλυση ληφθέντων αρχείων κακόβουλου περιεχομένου

Όπως έχει ήδη αναφερθεί το Dionaea προσφέρει τη δυνατότητα διαδικτυακής αποστολής των ληφθέντων αρχείων κακόβουλου περιεχομένου σε υπηρεσίες ανάλυσης κακόβουλου περιεχομένου, όπως VirusTotal, Anubis, Norman SandBox και CWSandbox. Επιπρόσθετα, κατά την παραμετροποίηση ο χρήστης δύναται να εισάγει τον προσωπικό λογαριασμό ηλεκτρονικού του ταχυδρομείου, ώστε να λαμβάνει ο ίδιος τα αποτελέσματα της ανάλυσης. Με

χρήση της δυνατότητας αυτής, ακολουθεί η ανάλυση των ληφθέντων αρχείων κακόβουλου περιεχομένου. Να επισημανθεί ότι συχνά κατά την ανάλυση του περιεχομένου αρχείων κακόβουλου τύπου ενδέχεται τα αποτελέσματα που δίνουν οι διάφορες εταιρείες αντιμετώπισης ιών στις υπηρεσίες ανάλυσης κακόβουλου περιεχομένου να διαφέρουν, πράγμα που δε σημαίνει απαραίτητα λανθασμένη αναγνώριση.

Για την ανάλυση που ακολουθεί χρησιμοποιήθηκε κατά κύριο λόγο η υπηρεσία VirusTotal και βοηθητικά οι υπηρεσίες Anubis και Norman Sandbox.

Στο παράρτημα εμπεριέχονται εικόνες από τις αναλύσεις των υπηρεσιών VirusTotal (Π9), Anubis (Π10) και Norman Sandbox (Π11).

Το αρχείο κακόβουλου περιεχομένου με το αναγνωριστικό (1) χαρακτηρίστηκε ως αγνώστου τύπου καθώς αναγνωρίστηκε σε ποσοστό 33%. Από την πλειοψηφία των εταιρειών αναγνωρίστηκε ως Gen:Trojan.Downloader.cmW@aaP5gLc, ως ένα trojan δηλαδή που προσπαθεί να κάνει λήψεις άλλων αρχείων κακόβουλου περιεχομένου και να αποστείλει και κρίσιμες πληροφορίες συστήματος απομακρυσμένα.

Το αρχείο κακόβουλου περιεχομένου με το αναγνωριστικό (2) χαρακτηρίστηκε ως ένα Win32.exe αρχείο που στοχεύει σε πλατφόρμες Microsoft Windows με επεξεργαστή Intel 386 και μεταγενέστερες εκδόσεις. Αναγνωρίστηκε σε ποσοστό 92% από τις εταιρείες αντιμετώπισης ιών της υπηρεσίας VirusTotal. Εμφανίστηκε για πρώτη φορά το 2012 και φέρει τις ονομασίες file-5215419_exe, lsass.exe αλλά και md5_hash.exe, όπου md5_hash το αντίστοιχο. Το κύριο χαρακτηριστικό του αρχείου είναι ότι προστατεύεται εντός ενός πακέτου ώστε να μην μπορεί με τεχνικές αντίστροφης μηχανικής να απομεταγλωττιστεί.

Το αρχείο κακόβουλου περιεχομένου με το αναγνωριστικό (3) χαρακτηρίστηκε ως ένα Win32.exe αρχείο που στοχεύει σε πλατφόρμες Microsoft Windows με επεξεργαστή Intel 386 και μεταγενέστερες εκδόσεις. Αναγνωρίστηκε σε ποσοστό 96% από τις εταιρείες αντιμετώπισης ιών της υπηρεσίας VirusTotal. Εμφανίστηκε για πρώτη φορά το 2012 και φέρει τις ονομασίες vti-rescan και lsass.exe1 αλλά και md5_hash.exe, όπου md5_hash το αντίστοιχο. Το κύριο χαρακτηριστικό του αρχείου είναι ότι προστατεύεται εντός ενός πακέτου ώστε να μην μπορεί με τεχνικές αντίστροφης μηχανικής να απομεταγλωττιστεί. Ειδικότερες λειτουργίες του μέχρι στιγμής δεν έχουν καταγραφεί και αναλυθεί καθώς και η ακριβής ταυτότητά του, αν δηλαδή αποτελεί trojan ή worm ή backdoor.

Το αρχείο κακόβουλου περιεχομένου με το αναγνωριστικό (4) χαρακτηρίστηκε ως αγνώστου τύπου καθώς δεν αναγνωρίστηκε από καμία εταιρεία αντιμετώπισης ιών της υπηρεσίας VirusTotal. Έχει μηδενικό μέγεθος, διαθέτει πληθώρα ονομάτων και έχει εμφανιστεί και ως μέρος συμπιεσμένων άλλων αρχείων κακόβουλου περιεχομένου αλλά και pcap αρχείων. Οι λειτουργίες του δεν έχουν καταγραφεί και αναλυθεί και ερευνάται προς το παρόν και η ιδιότητά του ως αρχείο κακόβουλου περιεχομένου.

Το αρχείο κακόβουλου περιεχομένου με το αναγνωριστικό (5) χαρακτηρίστηκε ως ένα trojan αρχείο που στοχεύει σε Microsoft Windows πλατφόρμες, (Win32). Εμφανίστηκε για πρώτη φορά το 2009 και φέρει την ονομασία dumpsys.exe αλλά και md5_hash.exe, όπου md5_hash το αντίστοιχο. Αναγνωρίστηκε σε ποσοστό 92% από τις εταιρείες αντιμετώπισης ιών της υπηρεσίας VirusTotal. Οι λειτουργίες του είναι οι εξής: αυτόματη εκκίνηση ανεπιθύμητων διαδικασιών κατά την εκκίνηση του συστήματος, τροποποίηση ρυθμίσεων ασφαλείας του Internet Explorer, διατήρηση αντιγράφων του ίδιου του αρχείου σε φακέλους του λειτουργικού ώστε η ύπαρξη του αρχείου να μη γίνει αντιληπτή από τους χρήστες, αποστολή μηνυμάτων ηλεκτρονικού ταχυδρομείου για μετάδοση σε άλλους υπολογιστές, σάρωση εύρους IP διευθύνσεων για εύρεση πιθανών ευάλωτων στόχων, τροποποίηση και καταστροφή μη προσωρινών αρχείων συστήματος και ανάγνωση και τροποποίηση κρίσιμων στοιχείων καταχώρησης, (registry values).

Το αρχείο κακόβουλου περιεχομένου με το αναγνωριστικό (6) χαρακτηρίστηκε ως Win32.exe αρχείο που στοχεύει σε πλατφόρμες Microsoft Windows με επεξεργαστή Intel 386 και μεταγενέστερες εκδόσεις. Αναγνωρίστηκε σε ποσοστό 90% από τις εταιρείες αντιμετώπισης ιών της υπηρεσίας VirusTotal. Αποτελεί αρχείο τύπου (3).

Το αρχείο κακόβουλου περιεχομένου με το αναγνωριστικό (7) χαρακτηρίστηκε ως παραλλαγή του HackTool:Win32/Keygen το οποίο χρησιμοποιείται για να παράγει κλειδιά για ψεύτικες παραλλαγές Windows αρχείων. Επιπρόσθετα επιδιώκει να λάβει διαδικτυακά πλήθος

αρχείων κακόβουλου περιεχομένου. Αναγνωρίστηκε σε ποσοστό 47% από τις εταιρείες αντιμετώπισης ιών της υπηρεσίας VirusTotal.

Το αρχείο κακόβουλου περιεχομένου με το αναγνωριστικό (8) χαρακτηρίστηκε ως ένα Win32.exe αρχείο που στοχεύει σε πλατφόρμες Microsoft Windows με 32-bit επεξεργαστή Intel 80386 και μεταγενέστερες εκδόσεις. Αναγνωρίστηκε σε ποσοστό 91% από τις εταιρείες αντιμετώπισης ιών της υπηρεσίας VirusTotal. Εμφανίστηκε για πρώτη φορά το 2010 και φέρει τις ονομασίες malware.exe, test.exe αλλά και md5_hash.exe, όπου md5_hash το αντίστοιχο. Οι λειτουργίες του είναι οι εξής: αυτόματη εκκίνηση ανεπιθύμητων διαδικασιών κατά την εκκίνηση του συστήματος, τροποποίηση ρυθμίσεων ασφαλείας του Internet Explorer, σάρωση εύρους IP διευθύνσεων για εύρεση πιθανών ευάλωτων στόχων, τροποποίηση και καταστροφή μη προσωρινών αρχείων συστήματος και ανάγνωση και τροποποίηση κρίσιμων στοιχείων καταχώρησης, (registry values). Επιπρόσθετα, κατεβάζει το αρχείο http://fukyu.jp/updata/ACCl3.jpg ως C:\WINDOWS\system32\msupd.exe και συνδέεται και στο αρχείο fukyu.jp στη θύρα 80.

Το αρχείο κακόβουλου περιεχομένου με το αναγνωριστικό (9) χαρακτηρίστηκε ως αγνώστου τύπου καθώς αναγνωρίστηκε σε ποσοστό 37%. Από την πλειοψηφία των εταιρειών που το αναγνώρισαν, χαρακτηρίστηκε ως αρχείο τύπου (1).

Το αρχείο κακόβουλου περιεχομένου με το αναγνωριστικό (10) χαρακτηρίστηκε ως αγνώστου τύπου καθώς αναγνωρίστηκε σε ποσοστό 39%. Η πλειοψηφία των εταιρειών αντιμετώπισης ιών χαρακτήρισε το αρχείο ως W32/Sality, ως έναν τύπο ιού που στοχεύει Windows αρχεία με κατάληξη .src ή .exe. Αφού εγκατασταθεί αυτός ο τύπος ιού, διακόπτει τη λειτουργία των αντιικών προγραμμάτων και επιδιώκει να διαγράψει κρίσιμα αρχεία συστήματος.

Το αρχείο κακόβουλου περιεχομένου με το αναγνωριστικό (11) χαρακτηρίστηκε από την υπηρεσία VirusTotal ως αγνώστου τύπου καθώς αναγνωρίστηκε σε ποσοστό μόλις 31% από εταιρείες αντιμετώπισης ιών. Από έξι εταιρείες χαρακτηρίστηκε ως Generic.Malware, δηλαδή ως ένα trojan που εξαπλώνεται σε Windows 2003/XP/2000/NT/ME/98/95 πλατφόρμες. Από άλλες εταιρείες αναγνωρίστηκε ως trojan που εξαπλώνεται ομοίως σε Windows πλατφόρμες και στόχο έχει την εκμαίευση κωδικών πρόσβασης ενώ τέλος χαρακτηρίστηκε και ως WIN.MAIL.WORM.Virus, δηλαδή ως ένα worm που εξαπλώνεται διαδικτυακά σαν επισυναπτόμενο αρχείο.

Το αρχείο κακόβουλου περιεχομένου με το αναγνωριστικό (12) χαρακτηρίστηκε ως αγνώστου τύπου καθώς αναγνωρίστηκε σε ποσοστό 36%. Από την πλειοψηφία των εταιρειών αντιμετώπισης ιών της υπηρεσίας VirusTotal αναγνωρίστηκε ως trojan τύπου αρχείου (1).

Το αρχείο κακόβουλου περιεχομένου με το αναγνωριστικό (13) χαρακτηρίστηκε ως αγνώστου τύπου καθώς αναγνωρίστηκε σε ποσοστό 33%. Από την πλειοψηφία των εταιρειών αντιμετώπισης ιών της υπηρεσίας VirusTotal αναγνωρίστηκε ως trojan τύπου αρχείου (1).

Το αρχείο κακόβουλου περιεχομένου με το αναγνωριστικό (14) χαρακτηρίστηκε ως αγνώστου τύπου καθώς αναγνωρίστηκε σε ποσοστό μόλις 24%. Από την πλειοψηφία των εταιρειών αναγνωρίστηκε ως ιός τύπου Win32.Virtob.8.Gen ή W32/Virut-W, δηλαδή ως ένας ιός που μολύνει αρχεία Windows με επέκταση .src και .exe, εγκαθιστά λογισμικό κακόβουλου περιεχομένου στο σύστημα τύπου backdoor και επιχειρεί διαδικτυακές λήψεις κακόβουλου περιεχομένου στο μολυσμένο υπολογιστή.

Το αρχείο κακόβουλου περιεχομένου με το αναγνωριστικό (15) χαρακτηρίστηκε ως ένα Win32.exe αρχείο που στοχεύει σε πλατφόρμες Microsoft Windows με επεξεργαστή Intel 386 και μεταγενέστερες εκδόσεις. Αναγνωρίστηκε σε ποσοστό 91% από τις εταιρείες αντιμετώπισης ιών της υπηρεσίας VirusTotal. Εμφανίστηκε για πρώτη φορά το 2009. Οι λειτουργίες του είναι όμοιες με αυτές του αρχείου τύπου (8).

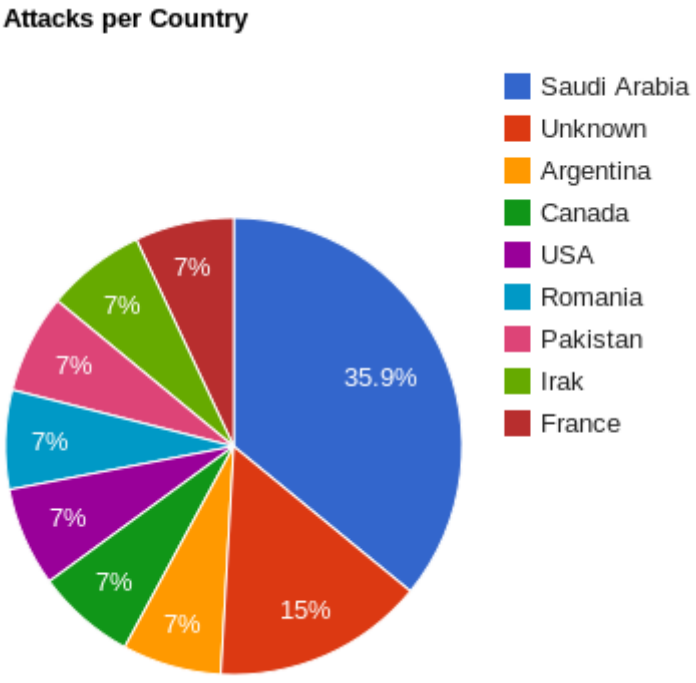
10.3.8 Επιθέσεις - Προέλευση

Ακολουθούν οι 10 διευθύνσεις από τις οποίες προήλθαν οι περισσότερες επιθέσεις προς το Honeypot, η προέλευσή τους, το πλήθος τους και το πρωτόκολλο που χρησιμοποίησαν. Δεκτές γίνονται όλες οι επιθέσεις στο Dionaea αρκεί η αντίστοιχη θύρα να είναι ανοιχτή. Στην παρούσα εφαρμογή οι συνδέσεις που στόχευαν στο http πρωτόκολλο απορρίφθηκαν γιατί η θύρα 80 ήταν κατειλημμένη από τον Apache διακομιστή. Να σημειωθεί ότι η προέλευση των επιθέσεων δεν είναι ακριβής καθώς η αναγνώριση των ενδιάμεσων, (proxy), διακομιστών δεν ήταν δυνατή.

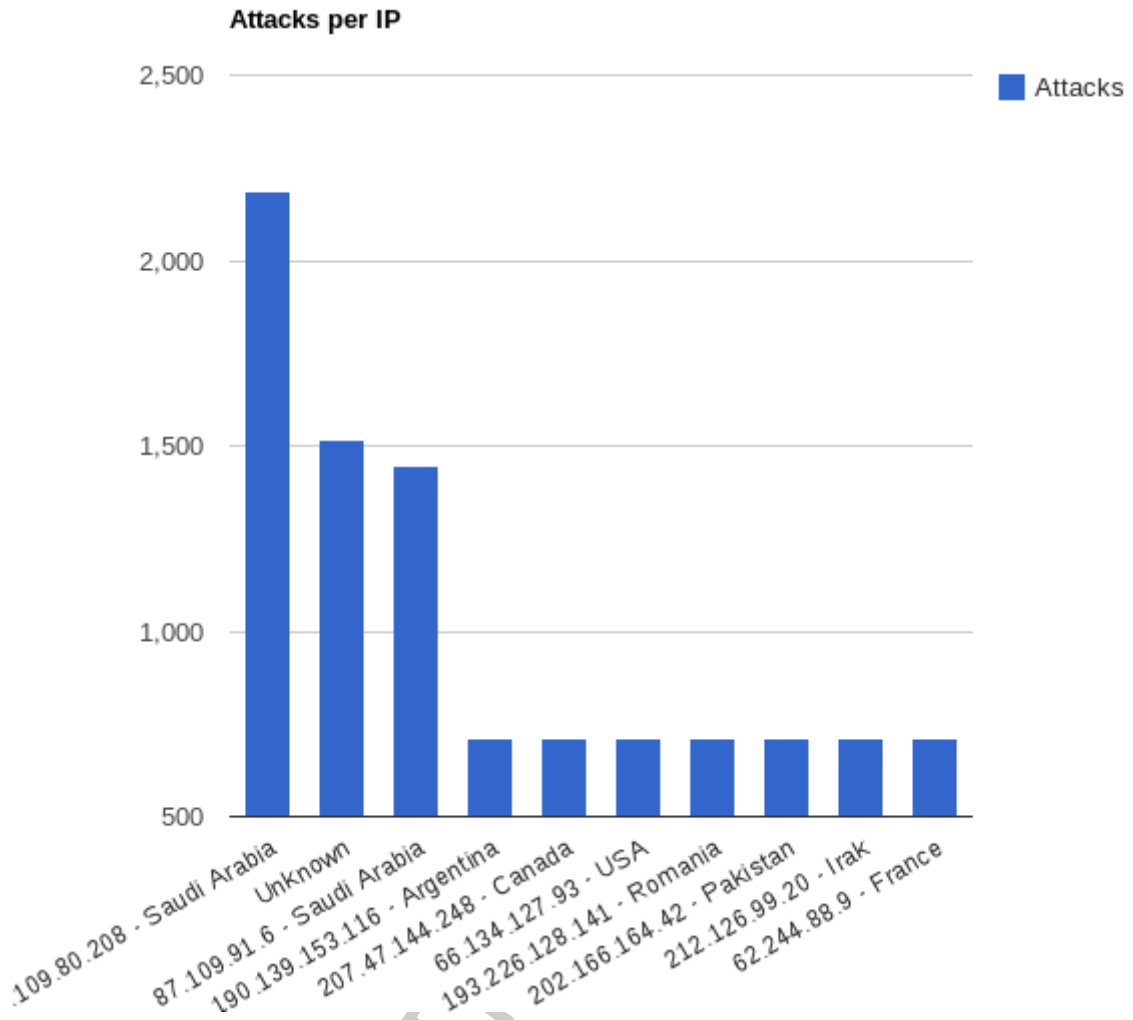
Διεύθυνση IP	Χώρα	Πλήθος επιθέσεων, (%)	Πρωτόκολλο
87.109.80.208	Σαουδική Αραβία	2186, (36%)	SipCall
Άγνωστη	Άγνωστη	1514, (15%)	FTP
87.109.91.6	Σαουδική Αραβία	1448, (36%)	SipSession
190.139.153.116	Αργεντινή	710, (7%)	SMB
207.47.144.248	Καναδάς	710, (7%)	SMB
66.134.127.93	ΗΠΑ	710, (7%)	SMB
193.226.128.141	Ρουμανία	709, (7%)	SMB
202.166.164.42	Πακιστάν	709, (7%)	SMB
212.126.99.20	Ιράκ	709, (7%)	SMB
62.244.88.9	Γαλλία	709, (7%)	SMB

Πίνακας 10-12: Top 10 διευθύνσεων IP - χωρών επιθέσεων, (Dionaea).

Στις εικόνες 10-27 και 10-28 διαφαίνονται αντίστοιχα οι 10 κορυφαίες διευθύνσεις IP που σημείωσαν τις περισσότερες επιθέσεις προς το Dionaea και τα ποσοστά επί του συνόλου των επιθέσεων ανά χώρα. Τα διαγράμματα αυτά σχεδιάστηκαν με χρήση Google Charts.



Εικόνα 10-27: Top 10 επιθέσεων ανά χώρα (%), (Dionaea).



Εικόνα 10-28: Top 10 επιθέσεων ανά διεύθυνση IP και χώρα, (Dionaea).

10.3.9 Διευθύνσεις λήψης αρχείων

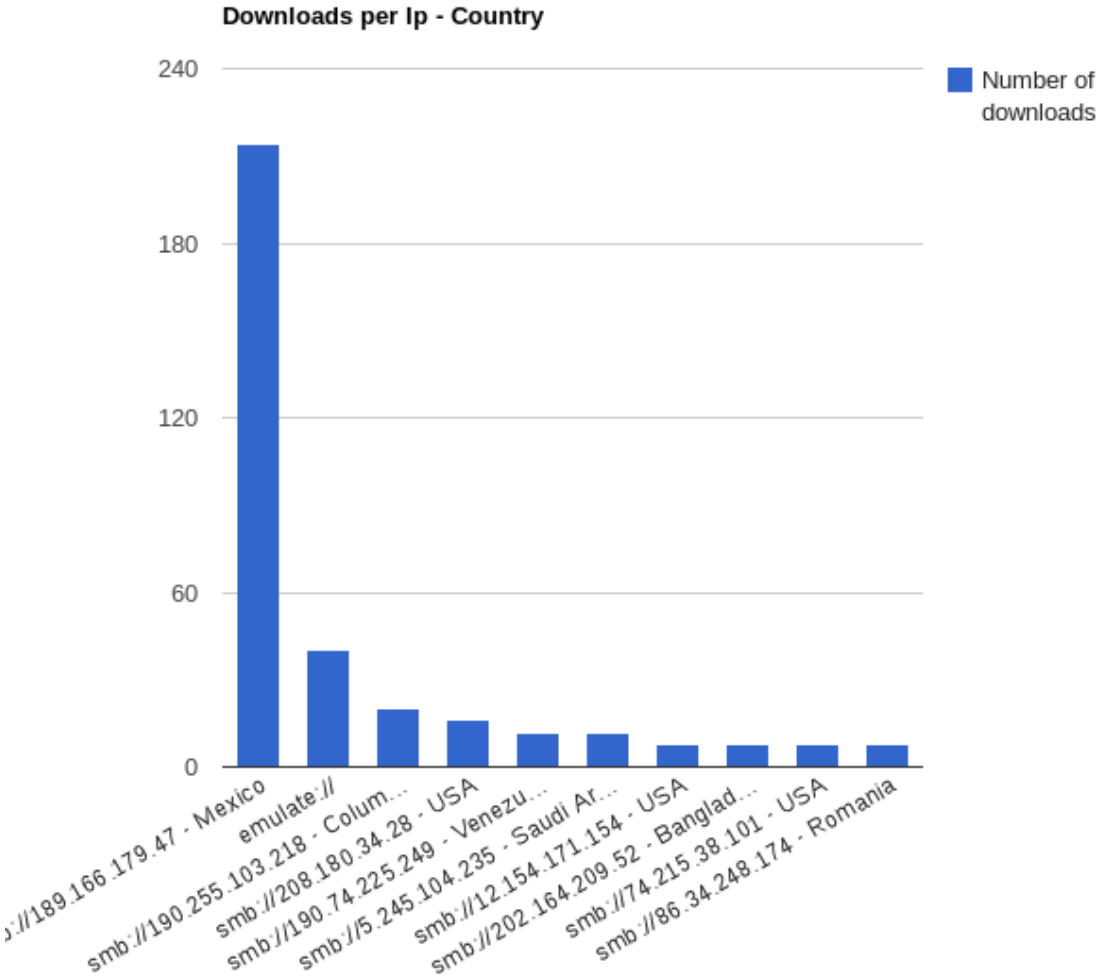
Ακολουθούν οι 10 διευθύνσεις IP από τις οποίες επιχειρήθηκαν οι περισσότερες λήψεις αρχείων, η πρόλευσή τους και το πλήθος τους.

Διεύθυνση λήψης IP	Χώρα	Πλήθος αιτημάτων, (%)
smb://189.166.179.47	Μεξικό	214, (61,8%)
emulate://		40, (11,6%)
smb://190.255.103.218	Κολομβία	20, (5,8%)
smb://208.180.34.28	ΗΠΑ	16, (9,2%)
smb://190.74.225.249	Βενεζουέλα	12, (3,5%)
smb://5.245.104.235	Σαουδική Αραβία	12, (3,5%)
smb://12.154.171.154	ΗΠΑ	8, (9,2%)
smb://202.164.209.52	Μπαγκλαντές	8, (2,3%)

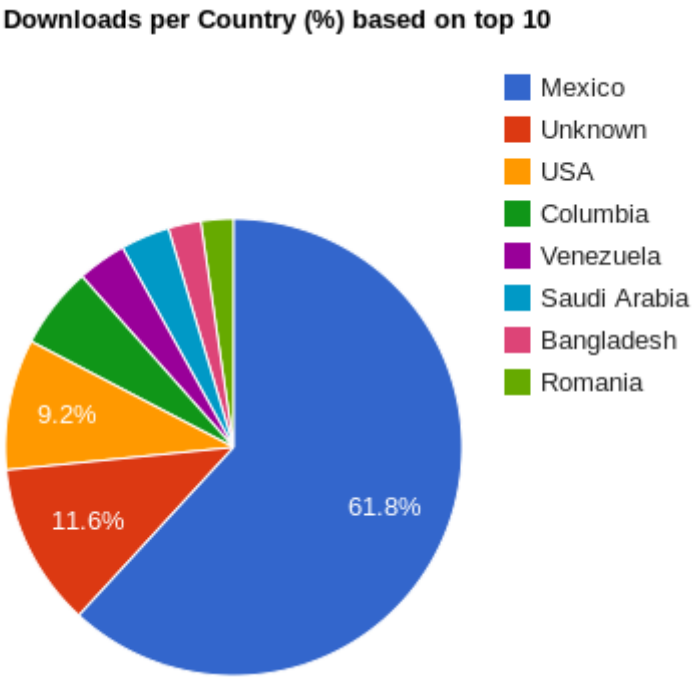
smb://74.215.38.101	ΗΠΑ	8, (9,2%)
smb://86.34.248.174	Ρουμανία	8, (2,3%)

Πίνακας 10-13: Top 10 διευθύνσεων IP λήψης αρχείων, (Dionaea).

Στην εικόνα 10-29 διαφαίνεται το πλήθος λήψης αρχείων κακόβουλου περιεχομένου ανά IP διεύθυνση και χώρα και στην εικόνα 10-30 διαφαίνεται ποσοστιαία το πλήθος λήψης αρχείων κακόβουλου περιεχομένου ανά χώρα.



Εικόνα 10-29: Top 10 IP διευθύνσεων και χωρών λήψης αρχείων κακόβουλου περιεχομένου, (Dionaea).



Εικόνα 10-30: Top 10 χωρών λήψεων αρχείων κακόβουλου περιεχομένου (%), (Dionaea).

10.3.10 Λειτουργικά συστήματα επιτιθέμενων

Ακολουθούν τα 10 λειτουργικά συστήματα μέσω των οποίων πραγματοποιήθηκαν οι περισσότερες επιθέσεις προς το Dionaea.

Λειτουργικό σύστημα	Πλήθος συνδέσεων
Windows	16925
	9203
Linux	935
SunOS	13
Novell	11
ExtremeWare	5
Solaris	4

Πίνακας 10-14: Λειτουργικά συστήματα επιτιθέμενων, (Dionaea).

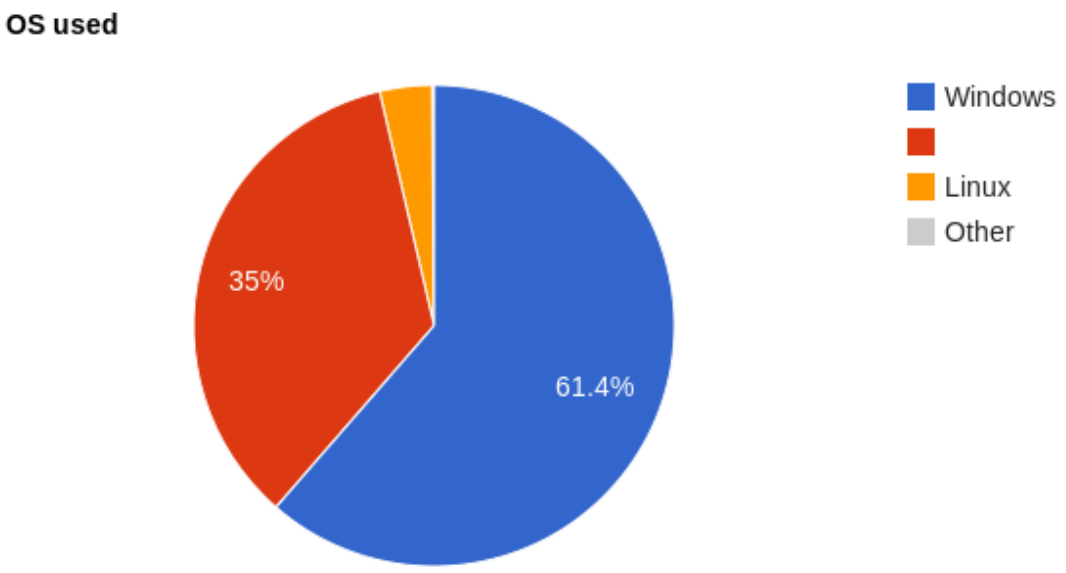
Ομοίως ακολουθεί και η ανάλυση συνδέσεων ανά τύπο λειτουργικού συστήματος.

Λειτουργικό σύστημα	Τύπος λειτουργικού συστήματος	Πλήθος συνδέσεων
		9203
Windows	2000 SP4, XP SP1+	8740

Windows	2000 SP2+, XP SP1+ (seldom 98)	4309
Windows	XP SP1+, 2000 SP3	1938
Windows	XP/2000 (RFC1323+, w+, tstamp-)	1514
Linux	2.4-2.6	321
Windows	2003 (2)	213
Linux	2.6, seldom 2.4 (older, 4)	204
Linux	2.6, seldom 2.4 (older, 2)	161
Linux	2.6 (newer, 3)	96

Πίνακας 10-15: Τύποι λειτουργικών συστημάτων επιτιθέμενων, (Dionaea).

Τα αποτελέσματα αυτά αναπαρίστανται στο διάγραμμα που ακολουθεί, το οποίο σχεδιάστηκε με χρήση Google Charts, (εικόνα 10-31).



Εικόνα 10-31: Λειτουργικά συστήματα επιτιθέμενων (%), (Dionaea).

10.3.11 Dionaea scripts

Το βασικό πακέτο του Dionaea εμπεριέχει και ορισμένα scripts, η λειτουργικότητα των οποίων θα αναλυθεί παρακάτω. Τα scripts αυτά είναι τα εξής: gnuplotsql και readlogsqltree. Επιπρόσθετα, θα περιγραφεί και η λειτουργία δύο ακόμα scripts, των dionaea-sqlquery-0_2.py και mimic-nepstats.py, τα οποία χρησιμοποιούνται ευρέως για την ανάλυση και επεξεργασία δεδομένων του Dionaea. Τα τελευταία scripts δε βρίσκονται στο βασικό πακέτο του Dionaea, ωστόσο είναι προεγκατεστημένα στη σουίτα του HoneyDrive.

Gnuplotsql.py

Το gnuplotsql.py είναι ένα script που έχει γραφτεί σε Python, το οποίο παράγει διάφορα διαγράμματα για τη συνολική κατάσταση λειτουργίας και κίνησης δεδομένων του Honeypot.

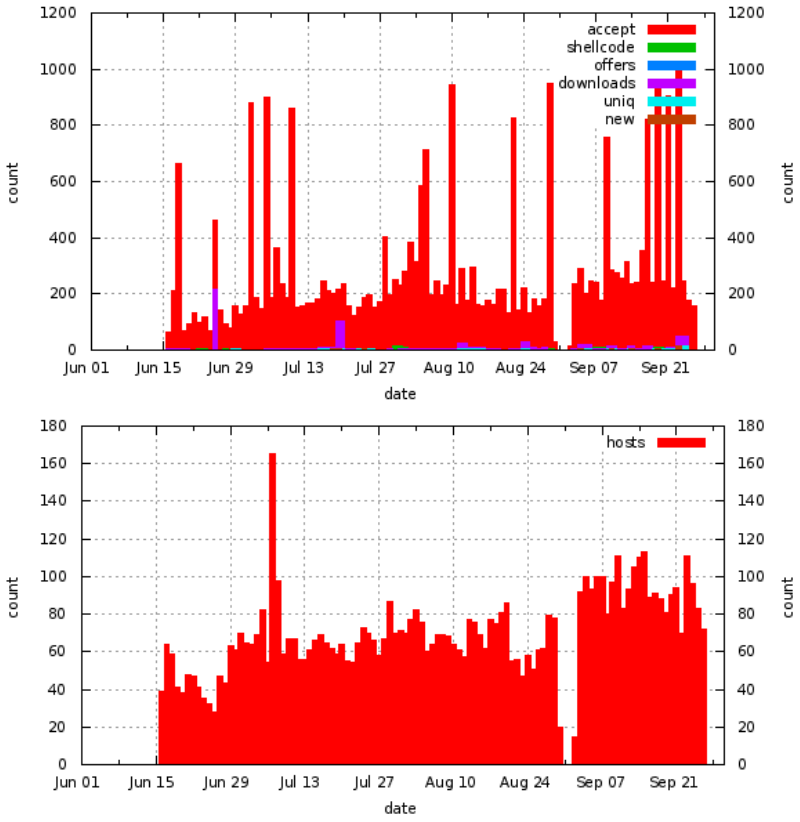
Το gnuplotsql.py βρίσκεται στον κατάλογο /opt/dionaea/bin και για την εκτέλεσή του απαιτεί την εγκατάσταση της βιβλιοθήκης gnuplot.

Στη συνέχεια με την εκτέλεση της παρακάτω εντολής, παράγονται διαγράμματα που παρουσιάζουν ανά πρωτόκολλο τη δραστηριότητα που καταγράφτηκε στο Dionaea κατά την περίοδο της λειτουργία του.

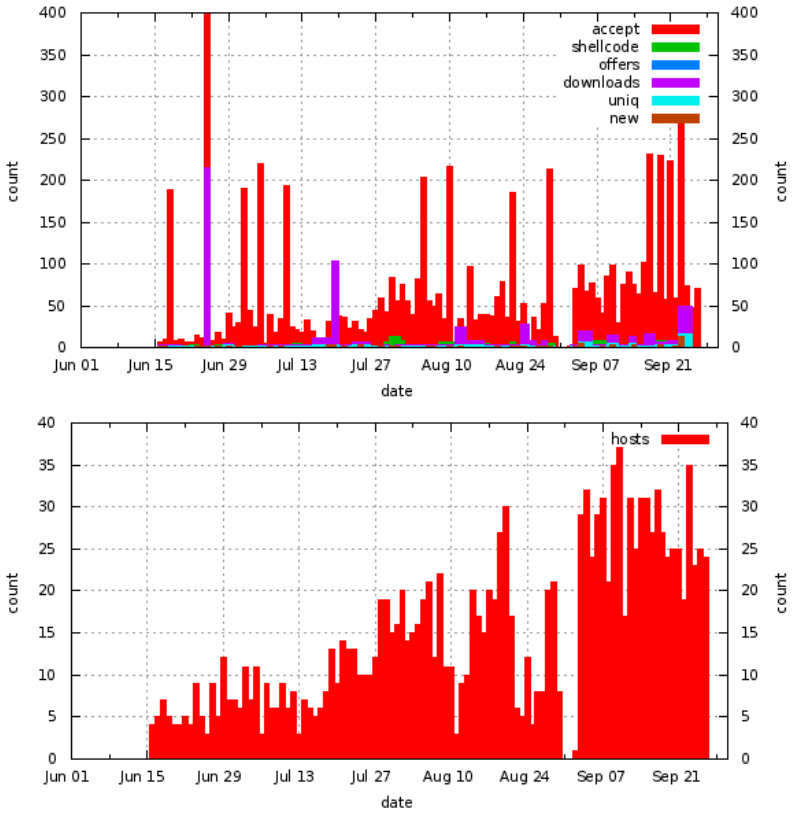
./python3.2 gnuplotsql.py -d /opt/dionaea/var/dionaea/logsql.sqlite -p smbd -p epmapper -p mssqld -p httpd -p ftpd

Τα διαγράμματα που δημιουργούνται αποθηκεύονται στο φάκελο /tmp/dionaea/gnuplot. Πρόκειται για διαγράμματα που αφορούν στην κίνηση που καταγράφτηκε στα πρωτόκολλα smbd, epmapper, mssqld, httpd και ftpd συνολικά και ανά μήνα λειτουργίας.

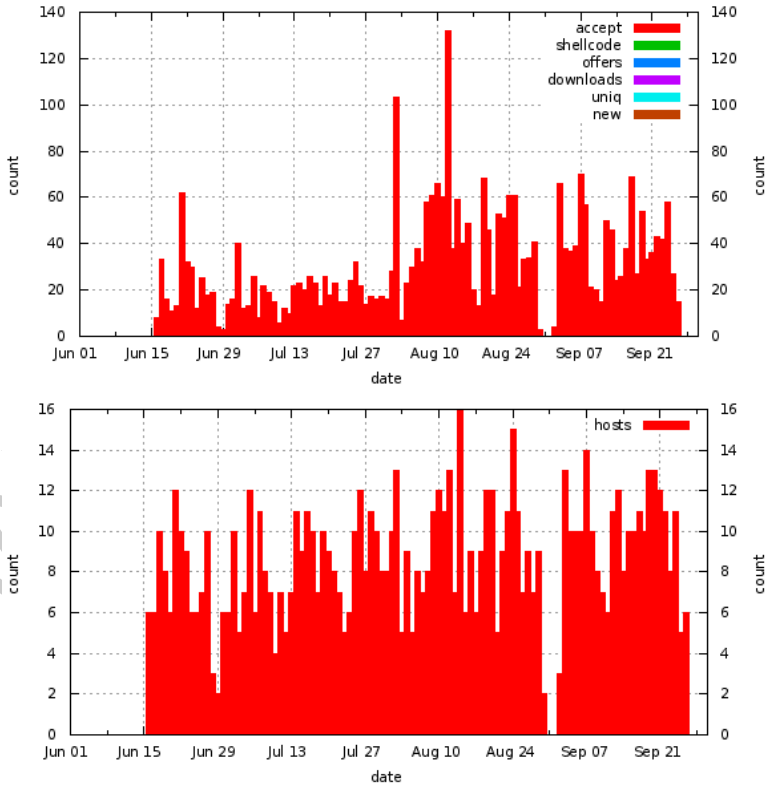
Ακολουθεί η παρουσίαση των διαγραμμάτων.



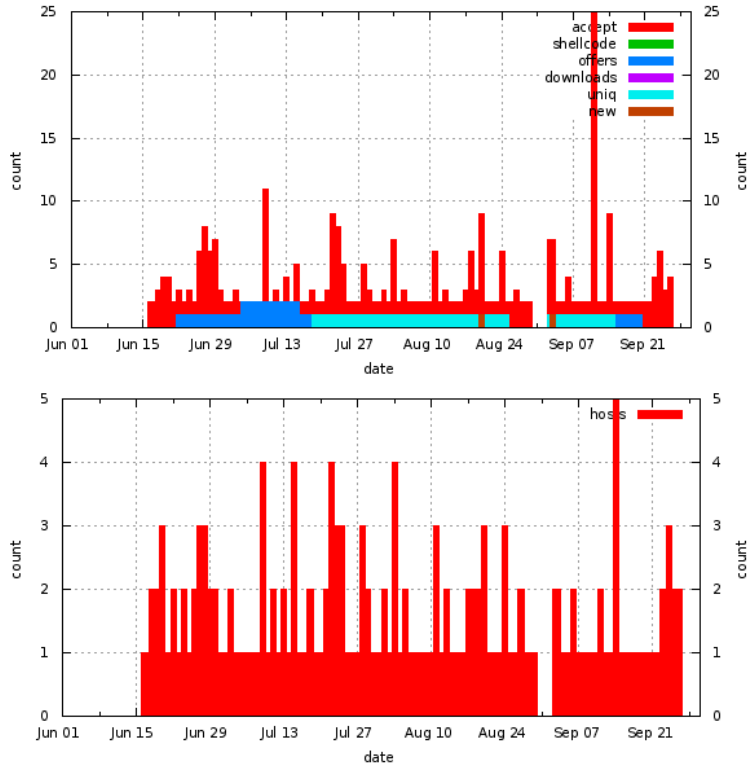
Εικόνα 10-32: Dionaea Overview.



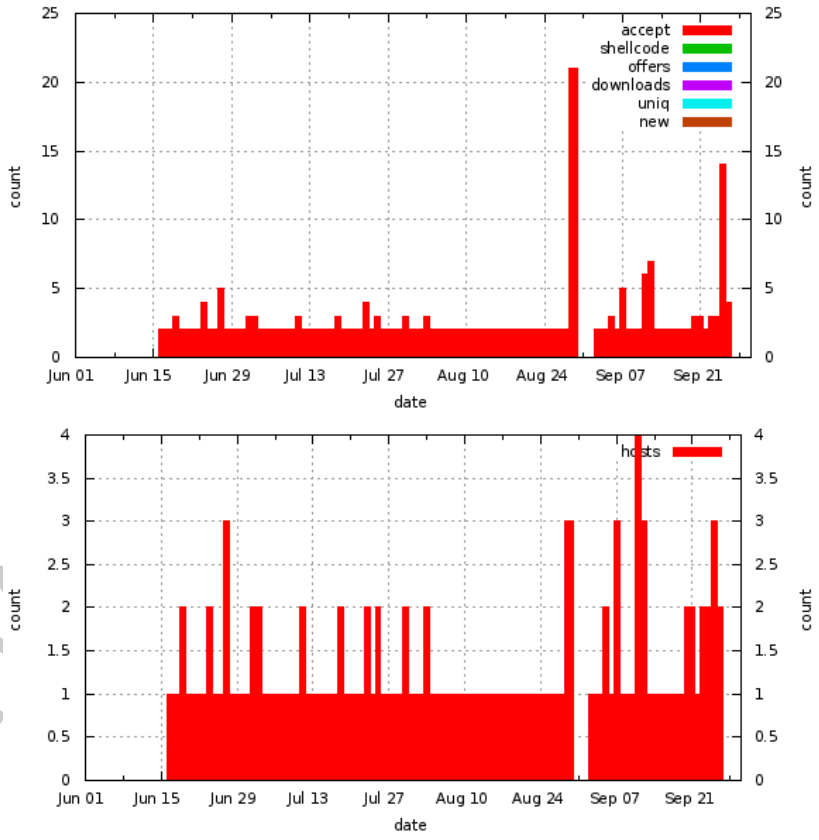
Εικόνα 10-33: smbd protocol.



Εικόνα 10-34: mssqld protocol.



Εικόνα 10-35: epmapper protocol.



Εικόνα 10-36: ftpd protocol.

Readlogsqltree.py

Το readlogsqltree.py είναι ένα script που έχει γραφτεί σε Python, το οποίο εκτελεί επερωτήσεις στην sqlite βάση δεδομένων του Dionaea και εξάγει αποτελέσματα για κάθε επίθεση. Πιο συγκεκριμένα, καταγράφει το πρωτόκολλο προς το οποίο έγινε η επίθεση, τη χρονική στιγμή της επίθεσης, τη διεύθυνση του εισβολέα, πληροφορίες γενικά για τον κώδικα κελύφους, το αρχείο που ζητήθηκε να ληφθεί από το Honeypot καθώς και τη VirusTotal ανάλυση που πραγματοποιήθηκε για το ληφθέν αρχείο.

Ωστόσο, αποτελεί ένα δύσχρηστο εργαλείο γιατί παράγει μεγάλο όγκο δεδομένων που δεν είναι εύκολο να διαβαστεί από τη γραμμή εντολής ή και από αρχείο κειμένου ενώ η εκτέλεσή του απαιτεί και πολύ καλή γνώση της γλώσσας python.

Το readlogsqltree.py βρίσκεται στο βασικό πακέτο εγκατάστασης του Dionaea.

Dionaea-sqlquery-0_2.py

Το dionaea-sqlquery-0_2.py είναι ένα script που έχει γραφτεί σε Python από τον Andrew Waite και εκτελεί έξι επερωτήσεις στην sqlite βάση δεδομένων του Dionaea. Οι επερωτήσεις αυτές εξάγουν το πλήθος επιθέσεων ανά θύρα, τη συχνότητα επιθέσεων ανά ημέρα, λίστα των επιτιθέμενων που πραγματοποίησαν τις περισσότερες επιθέσεις προς το Honeypot, λίστα των διευθύνσεων από τις οποίες πραγματοποιήθηκαν οι περισσότερες λήψεις αρχείων και το πλήθος των επιθέσεων που έλαβαν χώρα το τελευταίο 24ωρο.

Για την εκτέλεση αυτών των επερωτήσεων απαιτείται η εκτέλεση της εντολής

python dionaea-sqlquery-0_2.py –query x, όπου x ο αριθμός της επιθυμητής επερώτησης.

Το dionaea-sqlquery-0_2.py αποτελεί ελεύθερο λογισμικό. [64]

Mimic-nepstats.py

Το mimic-nepstats.py είναι ένα script που έχει γραφτεί σε Python από τον Andrew Waite και εξάγει στατιστικά στοιχεία για το Honeypot, όπως το πλήθος των συνδέσεων προς το Honeypots, το πλήθος των αρχείων που ελήφθησαν, το πλήθος των IP διευθύνσεων από τις οποίες προήλθαν οι συνδέσεις στο Honeypot, τις ημερομηνίες της πρώτης και της τελευταίας σύνδεσης που καταγράφτηκαν, τη διάρκεια λειτουργίας του Honeypot σε ημέρες καθώς και το μέσο αριθμό ληφθέντων αρχείων ανά ημέρα.

Το mimic-nepstats.py αποτελεί ελεύθερο λογισμικό. [65]

Οπτικοποίηση δεδομένων – DionaeaFR

Το DionaeaFR είναι ένα εργαλείο που παρουσιάζει σε διαγράμματα όλη τη δραστηριότητα του Dionaea. Εμφανίζει στατιστικά στοιχεία για τις συνδέσεις που υλοποιήθηκαν προς το Honeypot, για τα ληφθέντα αρχεία κακόβουλου περιεχομένου, για τις IP διευθύνσεις των επιτιθέμενων, για τις χώρες από τις οποίες πραγματοποιήθηκαν οι περισσότερες επιθέσεις. [60]

Το DionaeaFR είναι προεγκατεστημένο στη σουίτα του HoneyDrive, ωστόσο κατά την εκτέλεσή του παρουσιάστηκαν σφάλματα μεταγλώττισης, τα οποία δεν κατέστη δυνατό να επιλυθούν στην παρούσα έκδοση του HoneyDrive.

10.4 Glastopf

Το Glastopf εγκαταστάθηκε σε διαφορετικό του HoneyDrive εικονικό μηχάνημα στο https://okeanos.grnet.gr/, κι αυτό γιατί το Dionaea χρησιμοποιεί την ίδια θύρα, (80), με το Glastopf για να «ακούει» σε συνδέσεις προς το http πρωτόκολλο.

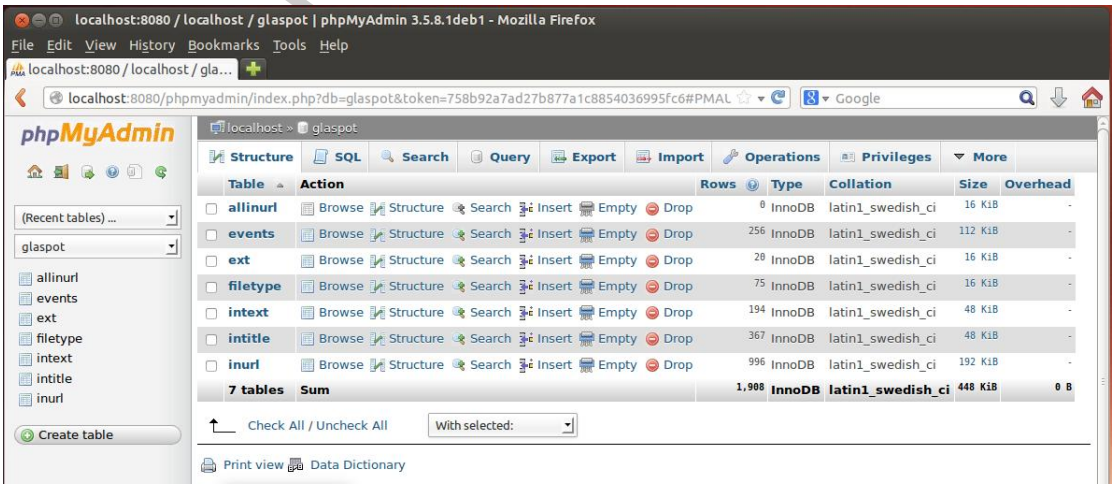
Εγκαταστάθηκε σε ένα μηχάνημα με τετραπύρηνο επεξεργαστή, με μνήμη 4096 Mb και με χωρητικότητα 40 Gb. Το λειτουργικό του είναι Ubuntu Desktop 13.04 64-bit. Η ip διεύθυνσή του είναι 83.212.108.158. Στην εικόνα 10-37 από το site του okeanos διαφαίνονται τα χαρακτηριστικά του εικονικού μηχανήματος, ονόματι Honeypots.

Εικόνα 10-37: Χαρακτηριστικά εικονικού μηχανήματος Honeypots.

Αναλυτικές οδηγίες εγκατάστασης του Glastopf παρατίθενται στο παράρτημα Π4.

10.4.1 Ανάλυση δεδομένων

Το Glastopf Honeypot λειτούργησε το διάστημα 6 Ιουλίου – 24 Αυγούστου, (50 ημέρες). Κατά το διάστημα αυτό καταγράφηκαν 256 αιτήσεις HTTP πρωτοκόλλου προς το Honeypot.



Εικόνα 10-38: Πλατφόρμα PhpMyAdmin, στιγμιότυπο βάσης δεδομένων Glaspot.

Στην ανωτέρω εικόνα 10-38 διαφαίνεται ένα στιγμιότυπο της βάσης δεδομένων του Glastopf, ονόματι Glaspot, μέσα από την πλατφόρμα του phpmyadmin.

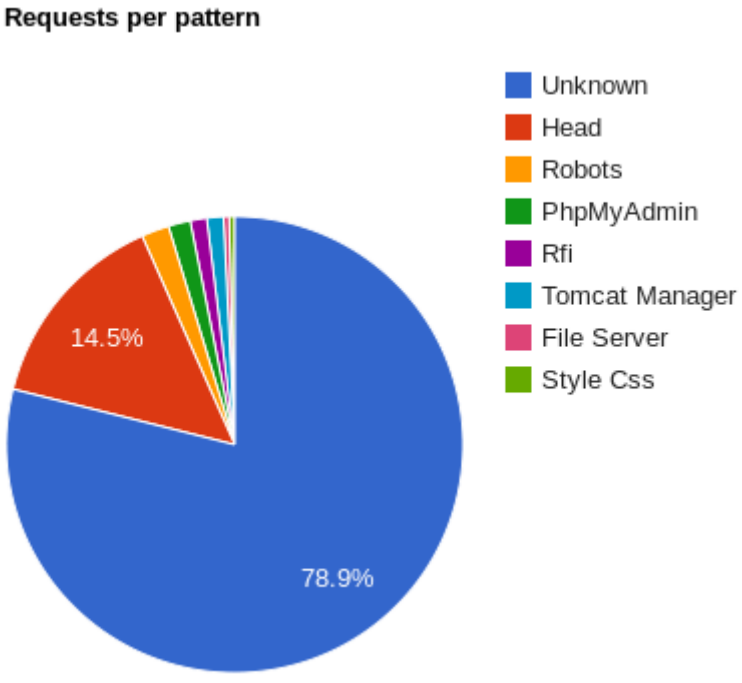
10.4.2 Πρότυπα

Στον παρακάτω πίνακα διαφαίνονται τα πρότυπα προς τα οποία στόχευαν οι αιτήσεις των εισβολέων προς το HTTP πρωτόκολλο, οι κατηγορίες δηλαδή στις οποίες το Glastopf ταξινομεί τις επιθέσεις.

Πρότυπο	Πλήθος επιθέσεων
Unknown	202, (78,9%)
Head	37, (14,5%)
Robots	5, (2%)
PhpMyAdmin	4, (1,6%)
Rfi	3, (1,2%)
Tomcat Manager	3, (1,2%)
File Server	1, (0,4%)
Style Css	1, (0,4%)

Πίνακας 10-16: Πλήθος επιθέσεων ανά πρότυπο, (Glastopf).

Τα αποτελέσματα αυτά αναπαρίστανται στο διάγραμμα που ακολουθεί, (εικόνα 10-39), το οποίο σχεδιάστηκε με χρήση Google Charts.



Εικόνα 10-39: Αιτήσεις ανά πρότυπο, (Glastopf).

10.4.3 Προέλευση

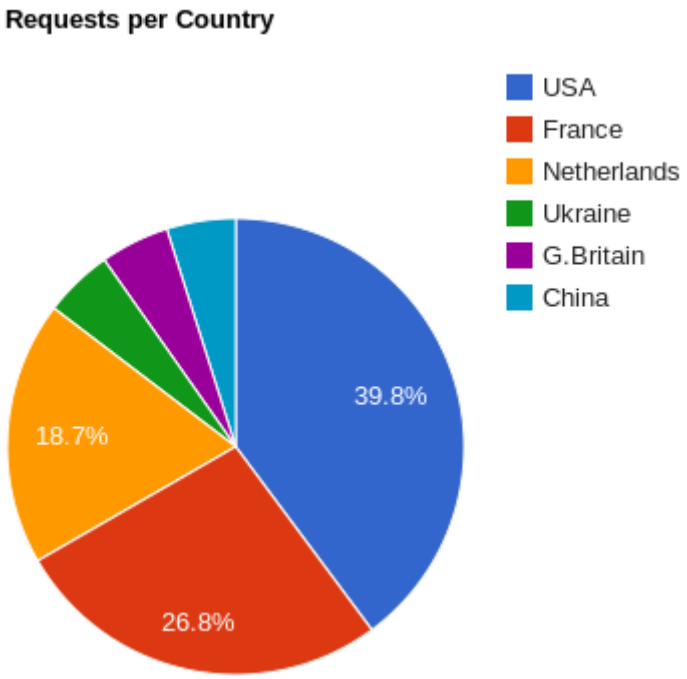
Ακολουθούν οι 10 διευθύνσεις IP από τις οποίες πραγματοποιήθηκαν οι περισσότερες επιθέσεις προς το Glastopf, η πρόλευσή τους και το πλήθος τους. Να σημειωθεί ότι η προέλευση των

επιθέσεων δεν είναι ακριβής καθώς η αναγνώριση των ενδιάμεσων, (proxy), διακομιστών δεν ήταν δυνατή.

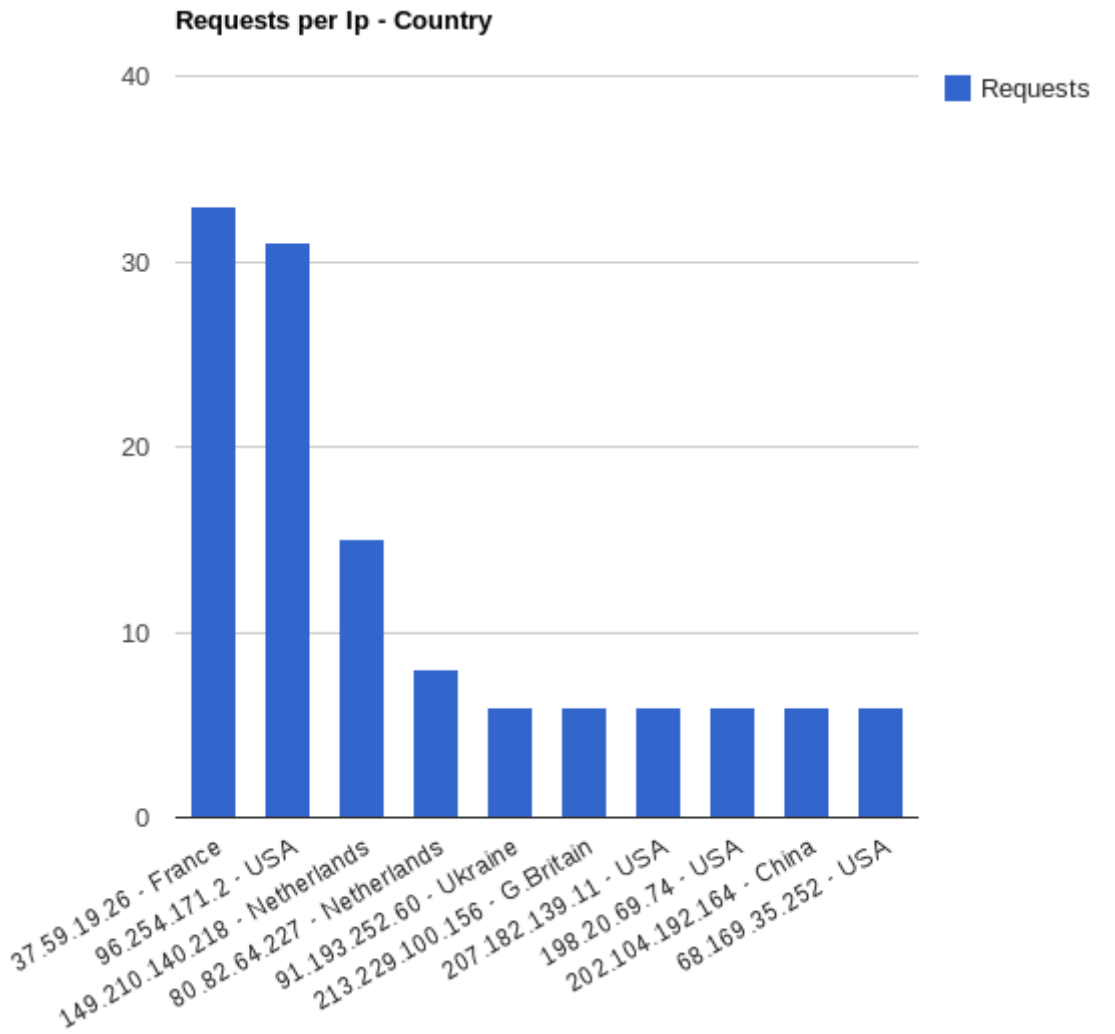
Διεύθυνση IP	Χώρα	Πλήθος επιθέσεων, (%)
37.59.19.26	Γαλλία	33, (26,8%)
96.254.171.2	ΗΠΑ	31, (39,8%)
149.210.140.218	Ολλανδία	15, (18,7%)
80.82.64.227	Ολλανδία	8, (18,7%)
91.193.252.60	Ουκρανία	6, (4,9%)
213.229.100.156	Μεγ. Βρετανία	6, (4,9%)
207.182.139.11	ΗΠΑ	6, (39,8%)
198.20.69.74	ΗΠΑ	6, (39,8%)
202.104.192.164	Κίνα	6, (4,9%)
68.169.35.252	ΗΠΑ	6, (39,8%)

Πίνακας 10-17: Top 10 διευθύνσεων IP - χωρών επιθέσεων, (Glastopf).

Στις παρακάτω εικόνες 10-40 και 10-41 διαφαίνονται αντίστοιχα το πλήθος αιτήσεων ανά IP διεύθυνση και χώρα ποσοστιαία και συνολικά. Τα διαγράμματα αυτά σχεδιάστηκαν με χρήση Google Charts.



Εικόνα 10-40: Αιτήσεις ανά χώρα (%), (Glastopf).



Εικόνα 10-41: Αιτήσεις ανά διεύθυνση IP και χώρα, (Glastopf).

10.4.4 Αιτήματα

Όπως έχει ήδη αναφερθεί, το Honeypot διατηρεί Dork List, μία λίστα δηλαδή ευπαθών αιτήσεων – διευθύνσεων, η οποία δημιουργείται κατά τις επιθέσεις προς το Honeypot και χρησιμοποιείται από τους εισβολείς μέσω μηχανών αναζήτησης όταν ψάχνουν για στόχους – θύματα. Τα αιτήματα intext, intitle και inurl που αναλύονται παρακάτω καταχωρούνται στην Dork List του Honeypot και χρησιμοποιούνται από τους εισβολείς για τη σύνδεσή τους.

Ακολουθούν τα 10 δημοφιλέστερα intext αιτήματα που πραγματοποιήθηκαν προς το διακομιστή του Glastopf.

Περιεχόμενο	Πλήθος
parent directory	19
Subject	13
robots.txt	10
Request Details	10
allow_call_time_pass_reference	10
sets mode: +p	10
Thank you for your order	10

HTTP_FROM=googlebot	10
not for distribution	9
Network Vulnerability Assessment Report	9

Πίνακας 10-18: Top 10 intext αιτημάτων, (Glastopf).

Ακολουθούν τα 10 intitle αιτήματα που πραγματοποιήθηκαν προς το διακομιστή του Glastopf.

Περιεχόμενο	**Πλήθος**
index.of	100
index of	88
Login	38
\"index	19
Index	12
Admin	12
DocuShare	12
curriculum vitae	11
Intranet	8
Hotmail	8

Πίνακας 10-19: Top 10 intitle αιτημάτων, (Glastopf).

Ακολουθούν τα 10 inurl αιτήματα που πραγματοποιήθηκαν προς το διακομιστή του Glastopf.

Περιεχόμενο	**Πλήθος**
/	35
/robots.txt	9
/manager/html	4
/appserv/main.php	3
/myadmin/scripts/setup.php	2
/ipc$	2
/phpmyadmin/index.php	2
/phpMyAdmin/scripts/setup.php	2
/phpmyadmin/translators.html	2
/modules/forums/admin/dork.php	2

Πίνακας 10-20: Top 10 inurl αιτημάτων, (Glastopf).

Η πλειοψηφία των intitle αιτημάτων περιέχει λέξεις όπως index, login, admin, τυπικές έννοιες δηλαδή ενός ιστοτόπου ή και ενός πληροφοριακού συστήματος εν γένει. Αντίστοιχα, η πλειοψηφία των inurl αιτημάτων στοχεύει στους καταλόγους phpmyadmin, manager, myadmin, τυπικά συστατικά ομοίως ενός ιστοτόπου. Ιδιαίτερο ενδιαφέρον παρουσιάζει το αίτημα /robots.txt, το οποίο θα αναλυθεί παρακάτω στα intext αιτήματα.

Ακολουθεί η ανάλυση ορισμένων εκ των intext αιτημάτων βάσει της επίσημης λίστας Dorks της Google. [67]

- robots.txt: Αρχείο που περιέχεται σε κάθε διακομιστή και περιέχει τα ονόματα των bots που έχουν ήδη επισκεφτεί τον διακομιστή. Ουσιαστικά περιέχει κανόνες που καθορίζουν σε ποιους καταλόγους διαδικτυακοί ανιχνευτές έχουν πρόσβαση και σε ποιους όχι, ποια στοιχεία του ιστοτόπου δηλαδή είναι δημόσια και ποια όχι.

- allow_call_time_pass_reference: Με το αίτημα αυτό επιστρέφονται δημόσιες σελίδες που παράγονται από την php συνάρτηση phpinfo() και η αναζήτηση αυτή δεν εξαρτάται από το αρχείο ονόματι phpinfo.php.
- sets mode: +p: Με το αίτημα αυτό επιστρέφονται κρυφά κανάλια IRC, (Internet Relay Chat), μέσω των αρχείων καταγραφής IRC.
- Thank you for your order: Κατόπιν διαδικτυακής παραγγελίας, πολλές σελίδες εμφανίζουν το μήνυμα “Thank you for your order”. Μέσω αυτού του αιτήματος οι εισβολείς προσπαθούν να λάβουν γνώση της δομής ενός e-shop.
- HTTP_FROM=googlebot: Οι σελίδες που επιστρέφονται με αυτό το αίτημα περιέχουν σημαντικές πληροφορίες που συλλέχτηκαν κατά την ανίχνευση της σελίδας από bot. Οι πληροφορίες αυτές μπορεί να περιλαμβάνουν σημαντικούς καταλόγους, πληροφορίες επικεφαλίδας, την έκδοση του διακομιστή κ.α.
- not for distribution: Ο όρος χρησιμοποιείται σπάνια και υποδεικνύει αρχείο, του οποίου η λήψη και η ανάγνωση δεν επιτρέπονται.

10.5 Συνδυαστικά αποτελέσματα

Καθένα εκ των τριών Honeypots που εγκαταστάθηκαν και μελετήθηκαν στα πλαίσια της παρούσας μεταπτυχιακής διατριβής έχει διαφορετικό σκοπό από τα υπόλοιπα, επομένως συγκεντρωτικά αποτελέσματα και για τα τρία Honeypots δεν μπορούν να εξαχθούν. Το Kippo προσομοιώνει έναν SSH διακομιστή, το Dionaea μια σειρά από πρωτόκολλα και το Glastopf έναν διαδικτυακό διακομιστή, οπότε το καθένα από αυτά προσελκύει επιτιθέμενους διαφορετικού σκοπού. Ο μόνος δείκτης που μπορεί να χρησιμοποιηθεί για την εξαγωγή συνδυαστικών αποτελεσμάτων είναι οι διευθύνσεις και οι χώρες επιθέσεων.

Ακολουθεί η ανάλυση περί κοινών διευθύνσεων και ο σχολιασμός χωρών επιθέσεων.

10.5.1 Κοινές διευθύνσεις IP επιθέσεων

Μεταξύ Kippo και Dionaea δε διαπιστώθηκαν κοινές IP διευθύνσεις επιθέσεων, αναμενόμενο καθώς προσομοιώνουν διαφορετικά πρωτόκολλα κι επομένως και οι επιτιθέμενοι είναι διαφορετικού τύπου. Ομοίως μεταξύ Kippo και Glastopf δε βρέθηκε καμία κοινή IP διεύθυνση επιθέσεων.

Μεταξύ Dionaea και Glastopf διαπιστώθηκαν δύο κοινές IP διευθύνσεις, η 198.20.69.74 προερχόμενη από τις ΗΠΑ και η 80.82.64.227 από την Ολλανδία.

Η μεν πρώτη πραγματοποίησε 9 επιθέσεις προς το Dionaea, οι οποίες απορρίφθηκαν και 6 επιθέσεις – αιτήσεις προς το Glastopf. Η δεύτερη διεύθυνση (80.82.64.227) πραγματοποίησε 147 επιθέσεις προς το Dionaea οι οποίες ομοίως απορρίφθηκαν και 8 επιθέσεις – αιτήσεις προς το Glastopf. Οι 147 επιθέσεις προς το Dionaea ταξινομούνται ως 17 στο πλήθος προς τις θύρες 1080, 3128, 8080, 8118, 8888 και 9999, 12 προς τη θύρα 7808, 9 προς τη θύρα 81 και 8 προς τις θύρες 8089, 8090 και 9000. Όλες οι ανωτέρω θύρες του Dionaea αφορούν στο http πρωτόκολλο και δεν ήταν ενεργές καθώς ο διακομιστής Apache που εκτελούνταν στο μηχάνημα του HoneyDrive «άκουγε» για συνδέσεις στη θύρα 80. Επομένως ο επιτιθέμενος της διεύθυνσης 80.82.64.227 επιδίωκε να βρει http ευπάθειες και για αυτό επιτέθηκε και στις παραπάνω θύρες του Dionaea και στο Glastopf.

10.5.2 Σχολιασμός χωρών επιθέσεων

Βάσει μίας έρευνας που εξήχθη από την αμερικανική επιχείρηση Ασφάλειας Πληροφοριακών Συστημάτων Akamai, το πρώτο τρίμηνο του 2013 η Κίνα βρίσκεται με ποσοστό 34% στην κορυφή της λίστας των 177 χωρών από τις οποίες θεωρείται δυνητικά ότι εξαπολύονται διαδικτυακές επιθέσεις. Ακολουθεί η Ινδονησία με ποσοστό 21% και στη συνέχεια οι ΗΠΑ με 8,3% και η Τουρκία με 4,5%. Σημαντικές πτώσεις ωστόσο στα ποσοστά τους είχαν η Κίνα και οι ΗΠΑ σε σχέση με το τελευταίο τρίμηνο του 2012 καθώς η Κίνα έπεσε από το 41% στο 34% και οι ΗΠΑ έπεσαν από το 10% στο 8,3%. Η πτώση αυτή οφείλεται στη μεγάλη άνοδο της Ινδονησίας από το 0,7% του 2012 στο 21% του 2013. Ακολουθούν οι Τουρκία (4,5%), Ρωσία (2,7%), Ινδία (2,6%), Ταϊβάν (2,5%), Βραζιλία (2,2%), Ρουμανία (2%) και Χονγκ-Κονγκ (1,6%).

Η έρευνα κατέδειξε ότι σχεδόν το 68% των συνολικών διαδικτυακών επιθέσεων εξαπολύεται από την περιοχή Ειρηνικού ωκεανού – Ασίας – Ωκεανίας, γεγονός που οφείλεται στη μεγάλη αύξηση των επιθέσεων από την Ινδονησία. Το αντίστοιχο ποσοστό της περιοχής το 2012 ήταν 56%. Η Ευρώπη εξαπολύει επιθέσεις σε ποσοστό κάτω του 19% ενώ η Βόρειος και η Νότιος Αμερική δεν ξεπερνούν μαζί το ποσοστό του 13%. [62]

Τα αντίστοιχα αποτελέσματα που προέκυψαν από τα τρία Honeypots Kippo, Dionaea και Glastopf που εγκαταστάθηκαν και μελετήθηκαν στα πλαίσια της παρούσας μεταπτυχιακής διατριβής εν γένει συνάδουν με τα παραπάνω αποτελέσματα. Ακολουθεί η ανάλυση των επιθέσεων ανά Honeypot. Να σημειωθεί ότι η καταγωγή των IP διευθύνσεων δεν είναι ακριβής καθώς δεν είναι δυνατή η αναγνώριση των ενδιάμεσων, (proxy), διακομιστών.

Kippo: Στις χώρες που πραγματοποίησαν τις περισσότερες επιτυχείς συνδέσεις, η Κίνα με ποσοστό 38,8% και οι ΗΠΑ με ποσοστό 13,4% καταλαμβάνουν τη μερίδα του λέοντος. Με ποσοστά γύρω στο 10% ακολουθούν οι ευρωπαϊκές χώρες Γαλλία, Βουλγαρία και Ρουμανία ενώ με ποσοστά γύρω στο 3% ακολουθούν οι Ινδία, Βραζιλία, Ν. Κορέα και ο Καναδάς.

Στις χώρες από τις οποίες πραγματοποιήθηκαν οι περισσότερες επιθέσεις προς το Kippo, η Γαλλία βρίσκεται στην κορυφή της λίστας με ποσοστό σχεδόν 50% και ακολουθεί η Μεγάλη Βρετανία με ποσοστό 17,3%. Στη συνέχεια βρίσκονται οι ΗΠΑ με 15,9% και η Κίνα με 7% και ακολουθούν οι Ν. Κορέα με ποσοστό γύρω στο 3% και οι Ουκρανία, Βραζιλία με ποσοστό γύρω στο 1%. Στο 1% βρίσκεται το Χονγκ-Κονγκ και ακολουθούν οι Σιγκαπούρη, Καναδάς και Ελλάδα με ποσοστό μικρότερο του 1%.

Από τα παραπάνω αποτελέσματα και σε σύγκριση με την έρευνα της Akamai, παρατηρείται ότι λείπουν εντελώς η Ινδονησία, η Τουρκία, η Ταϊβάν και η Ρωσία.

Dionaea: Στις χώρες που πραγματοποίησαν τις περισσότερες επιτυχείς συνδέσεις, οι ΗΠΑ και Κίνα πρωτοστατούν με ποσοστά αντίστοιχα 18% και 17%, ακολουθεί το Μεξικό με 11% και στη συνέχεια με 9% βρίσκονται οι Αργεντινή, Καναδάς, Ταϊβάν, Ρουμανία, Πακιστάν και Ν.Κορέα.

Στις χώρες από τις οποίες πραγματοποιήθηκαν οι περισσότερες επιθέσεις προς το Dionaea, η Κίνα δε βρίσκεται καν στις 10 κορυφαίες. Στην κορυφή βρίσκονται οι ΗΠΑ με το εντυπωσιακό ποσοστό του σχεδόν 70%. Ακολουθεί η Ελβετία με 12%, η Παλαιστίνη με 6%, το Ισραήλ, η Μεγ. Βρετανία και η Αυστραλία με 4%.

Στις χώρες από τις οποίες ελήφθησαν τα περισσότερα αρχεία κακόβουλου περιεχομένου, το Μεξικό καταλαμβάνει τη μερίδα του λέοντος με ποσοστό 61,8%. Ακολουθούν οι ΗΠΑ με 9% και η Κολομβία με 6%, στο 3,5% βρίσκονται η Βενεζουέλα και η Σαουδική Αραβία και τέλος κοντά στο 2,5% βρίσκονται το Μπανγκλαντές και η Ρουμανία. Να τονιστεί ότι το 12% περίπου των επιθέσεων ελήφθησαν από διεύθυνση η οποία δεν κατέστη δυνατό να αναγνωριστεί.

Από τα παραπάνω αποτελέσματα εντύπωση προκαλεί το γεγονός ότι η Κίνα απουσιάζει από τις 10 κορυφαίες χώρες από τις οποίες εξαπολύθηκαν οι περισσότερες επιθέσεις προς το Dionaea αλλά και η ύπαρξη της Ελβετίας στις 10 αυτές χώρες καθώς η Ελβετία εν γένει απουσιάζει από οποιαδήποτε άλλη στατιστική που λήφθηκε στα πλαίσια της παρούσας μεταπτυχιακής διατριβής.

Glastopf: Το δείγμα που λήφθηκε είναι κατά πολύ μικρότερο σε σύγκριση με το δείγμα των δύο άλλων Honeypots, καθώς αφορά μόνο περίοδο χρήσης 50 ημερών. Στο διάστημα αυτό, οι περισσότερες αιτήσεις προς το Glastopf έγιναν σε ποσοστό σχεδόν 40% από τις ΗΠΑ. Ακολουθούν η Γαλλία με 26,8% και η Ολλανδία με 19%. Τη δεκάδα συμπληρώνουν οι Ουκρανία, Μεγ. Βρετανία και Κίνα με 4,9%.

11. Συμπερασματα – Περιληψη

Στην παρούσα μεταπτυχιακή διατριβή για πειραματική εφαρμογή υλοποιήθηκαν τρία διαφορετικά Honeypot συστήματα, δύο χαμηλής αλληλεπίδρασης, (Dionaea και Glastopf), και ένα μεσαίας αλληλεπίδρασης, (Kippo), προκειμένου να μελετηθεί η λειτουργικότητά τους, να καταγραφούν οι επιθέσεις εναντίον τους, οι απόπειρες μη εξουσιοδοτημένης πρόσβασης και χρήσης εναντίον τους και να εξαχθούν ενδεχομένως σημαντικά συμπεράσματα για την Προστασία Πληροφοριακών Συστημάτων εν γένει. Σκοπός της εγκατάστασής τους ήταν η καταγραφή της δραστηριότητας των εισβολέων ώστε να αποκτηθεί η γνώση της μεθοδολογίας των επιθέσεών τους.

Δύο από τα Honeypots, (Kippo και Dionaea), αναπτύχθηκαν στη σουίτα HoneyDrive. Η σουίτα HoneyDrive όπως αποδείχθηκε είναι ένα πολύ βολικό εργαλείο για την εγκατάσταση, παραμετροποίηση και ανάλυση δεδομένων των Honeypots καθώς συγκεντρώνει σε μία μόνο πλατφόρμα πληθώρα Honeypots και συναφών δικτυακών εργαλείων. Στα πλαίσια της παρούσας μεταπτυχιακής διατριβής αναπτύχθηκαν δύο Honeypots από τη σουίτα HoneyDrive και όχι όλα γιατί ορισμένα εξ αυτών χρησιμοποιούσαν κοινές θύρες, (π.χ. Dionaea, Glastopf, Amun, Nepenthes), οπότε η λειτουργία τους θα αναιρούνταν ενώ άλλα κρίθηκε χρησιμότερο να μην εγκατασταθούν λόγω παλαιότητας, (π.χ. το πολύ δημοφιλές χαμηλής αλληλεπίδρασης Honeyd ανακοινώθηκε το 2003 και η τελευταία του αναβάθμιση ανακοινώθηκε το 2007).

Και τα τρία Honeypots παρέμειναν συνδεδεμένα στο Διαδίκτυο για μεγάλο χρονικό διάστημα, (Kippo και Dionaea για τρεις μήνες, Glastopf για πενήντα ημέρες), και διαπιστώθηκε η πληθώρα επιθέσεων που δέχονταν καθημερινά και καταγράφτηκε και μελετήθηκε η δραστηριότητα των εισβολέων αφού συνδεθούν επιτυχώς στο κάθε Honeypot. Στο Kippo καταγράφτηκαν οι εντολές που χρησιμοποιούσαν οι εισβολείς εντός του Honeypot, στο Dionaea μελετήθηκαν τα αρχεία κακόβουλου περιεχομένου που λήφθηκαν και στο Glastopf εξετάστηκε το είδος και η σύνταξη των αιτήσεων καθώς και οι ευριστικές μέθοδοι των εισβολέων μέσω Dork Lists για τον εντοπισμό υποψήφιων θυμάτων και συνακόλουθα και του Honeypot. Ομοίως σε κάθε Honeypot μελετήθηκε η συχνότητα, η προέλευση και η γεωγραφική κατανομή των επιθέσεων. Όλα τα ανωτέρω αποτελέσματα παρουσιάστηκαν σε πίνακες και σε γραφήματα.

Εδικότερα, στο Kippo διαπιστώθηκε ότι η ύπαρξη μίας μόνο ανοιχτής θύρας μπορεί να προσελκύσει χιλιάδες εισβολείς και ότι η χρήση εύκολων κωδικών πρόσβασης και ονομάτων χρήστη διευκολύνει την εισαγωγή των εισβολέων στο μηχάνημα – θύμα και τη διενέργεια οποιασδήποτε δραστηριότητάς τους εντός του μηχανήματος. Στο Dionaea διαπιστώθηκε ότι η ύπαρξη ανοιχτών θυρών εκτός του ότι προσελκύει χιλιάδες εισβολείς επιτρέπει και τη λήψη αρχείων οποιουδήποτε περιεχομένου και συνεπώς και κακόβουλου με άμεση συνέπεια τη μόλυνση και δυσλειτουργία του μηχανήματος – θύτη. Τέλος, στο Glastopf κατανοήθηκε ο τρόπος λειτουργίας και η μεθοδολογία που ακολουθούν διαδικτυακοί εισβολείς κάνοντας χρήση dork lists για να βρουν τα υποψήφια θύματά τους.

Συμπερασματικά, τα Honeypots μπορούν να λειτουργήσουν θετικά σε ένα δίκτυο όσον αφορά στην ασφάλειά του. Οποιαδήποτε κίνηση καταγράφεται από και προς αυτά θεωρείται κακόβουλου περιεχομένου καθώς ο μόνος λόγος ύπαρξης τους είναι η καταγραφή επιθέσεων. Παράλληλα, η καταγραφή της δραστηριότητας των επιτιθέμενων πραγματοποιείται σε βάσεις δεδομένων οι οποίες είναι εύκολο με χρήση επερωτήσεων να γίνουν αντικείμενο επεξεργασίας. Ωστόσο, απαιτούνται συνεχείς αναβαθμίσεις στο δίκτυο ώστε τα Honeypots να μη γίνουν αντιληπτά από τους εισβολείς. Ομοίως απαιτείται ιδιαίτερη προσοχή και μελέτη κατά την εγκατάσταση, τοποθέτηση και παραμετροποίηση των Honeypots ώστε τα Honeypots να μην αποτελέσουν την αφετηρία για εκκίνηση επιθέσεων προς άλλα δίκτυα. Έγκειται επομένως στους διαχειριστές δικτύων να αποφασίζουν για τον σκοπό και την ορθή χρήση των Honeypots στα πλαίσια του μεγέθους και του σκοπού του δικτύου που διαχειρίζονται.

12. ΠΑΡΑΡΤΗΜΑ

12.1 Π1 – Εγκατάσταση HoneyDrive

Το HoneyDrive είναι ένα ova αρχείο, (Open Virtual Appliance), ένας εικονικός σκληρός δίσκος δηλαδή. Ο Okeanos δεν υποστηρίζει τη δημιουργία εικονικών μηχανημάτων μέσω ova αρχείων, οπότε απαιτήθηκε η δημιουργία του HoneyDrive.ova αρχείου σε HoneyDrive.raw αρχείο.

Συγκεκριμένα, από μηχάνημα Linux, με χρήση των εντολών

tar –xvf HoneyDrive.ova

qemu-img convert –O raw HoneyDrive.vmdk HoneyDrive.raw

το αρχείο HoneyDrive.ova μετατρέπεται σε HoneyDrive.raw αρχείο.

Στη συνέχεια, με χρήση του snf-image-creator δημιουργήθηκε και καταχωρήθηκε στο Cyclades το HoneyDrive.raw αρχείο με χρήση των παρακάτω εντολών.

sudo –s

snf-mkimage HoneyDrive.raw

12.2 Π2 – Εγκατάσταση Kippo

Για την εγκατάσταση του Kippo απαιτείται η εγκατάσταση των πακέτων python-dev, openssl, python-openssl, python-pyasn1, python-twisted και python-mysqldb. Ενδείκνυται και η εγκατάσταση του phpmyadmin για την πρόσβαση στη MySQL βάση δεδομένων του Kippo.

Σε λειτουργικά Linux δεν επιτρέπεται σε κανέναν άλλο χρήστη πλην του υπερχρήστη να χρησιμοποιεί δικτυακές θύρες με αριθμό κάτω του 1024 ενώ και το Kippo για λόγους ασφαλείας δεν μπορεί να εκτελεστεί από τον υπερχρήστη. Για το λόγο αυτό χρησιμοποιείται η εφαρμογή authbind, η οποία παρέχει τη δυνατότητα σε ένα πρόγραμμα, το οποίο δεν εκκινείται χωρίς δικαιώματα υπερχρήστη να επικοινωνεί δικτυακά με χρήση θυρών που κατά τα άλλα θα απαιτούσαν δικαιώματα υπερχρήστη. Με την εκτέλεση των παρακάτω εντολών, το πρόβλημα αυτό επιλύεται.

sudo touch /etc/authbind/byport/22

sudo chown kippo:kippo /etc/authbind/byport/22

sudo chmod 777 /etc/authbind/byport/22

Στη συνέχεια, με την παρακάτω εντολή πραγματοποιείται λήψη όλων τα αρχεία του Kippo σε προσωπικό υπολογιστή.

svn checkout http://kippo.googlecode.com/svn/trunk/ ./kippo

Για τη δημιουργία ενός χρήστη και μίας βάσης δεδομένων για την αποθήκευση όλων των δεδομένων που καταγράφει το Kippo απαιτείται η παρακάτω διαδικασία.

Με χρήση της παρακάτω εντολής πραγματοποιείται σύνδεση στον MySQL διακομιστή.

mysql –u root –p

Με χρήση των παρακάτω εντολών δημιουργείται μία βάση δεδομένων kippo και ένας χρήστης kippo.

CREATE DATABASE kippo;

GRANT ALL ON kippo.* TO ‘kippo’@’localhost’ IDENTIFIED BY ‘honeydrive’;

Στη συνέχεια απαιτείται εκ νέου σύνδεση στη βάση ως χρήστης kippo και με χρήση της τελευταίας εντολής εισάγεται το σχήμα της βάσης (schema) που περιέχεται στην έκδοση του kippo.

mysql –u kippo –p;

USE kippo;

source ./doc/sql/mysql.sql;

Τέλος, στο αρχείο kippo.cfg απαιτείται αλλαγή της ssh_port από 2222 σε 22 και αφαίρεση του συμβόλου «#» από τις γραμμές που αφορούν στην καταγραφή σε βάση δεδομένων ώστε να ενεργοποιηθεί η δυνατότητα καταγραφής στη βάση δεδομένων και τέλος προσθήκη του κωδικού του χρήστη kippo της kippo MySQL βάσης δεδομένων.

12.3 Π3 – Εγκατάσταση Dionaea

Για την εγκατάσταση του Dionaea απαιτούνται τα πακέτα libudns-dev, libglib2.0-dev, libssl-dev, libcurl4-openssl-dev, libreadline-dev, libsqlite3-dev, python-dev, libtool, automake, autoconf, build-essential, subversion, git-core, flex, bison και pkg-config.

Στη συνέχεια, απαιτείται η εγκατάσταση των βιβλιοθηκών liblcfg, libemu, libnl, libpcap και libev καθώς και της έκδοσης 3.2 της γλώσσας Python αλλά και της γλώσσας Cython που αποτελεί ένα υπερσύνολο της Python και επιτρέπει τη συγγραφή επεκτάσεων σε C/C++.

Όλα τα παραπάνω εγκαθίστανται με τη βοήθεια του διαχειριστή πακέτων, (aptitude), στο μονοπάτι /opt/dionaea.

Η λήψη του κώδικα του Dionaea γίνεται μέσω του σχετικού repository με την εντολή

```
git clone git://git.carnivore.it/dionaea.git dionaea
```

και στη συνέχεια στο μονοπάτι /opt/dionaea/dionaea απαιτείται η εκτέλεση των εντολών

```
autoreconf –vi και
./configure --with-lcfg-include=/opt/dionaea/include/ \
        --with-lcfg-lib=/opt/dionaea/lib/ \
        --with-python=/opt/dionaea/bin/python3.2 \
        --with-cython-dir=/opt/dionaea/bin \
        --with-emu-include=/opt/dionaea/include/ \
        --with-emu-lib=/opt/dionaea/lib/ \
        --with-gc-include=/usr/include/gc \
        --with-ev-include=/opt/dionaea/include \
        --with-ev-lib=/opt/dionaea/lib \
        --with-nl-include=/opt/dionaea/include \
        --with-nl-lib=/opt/dionaea/lib/ \
        --with-curl-config=/usr/bin/ \
        --with-pcap-include=/opt/dionaea/include \
        --with-pcap-lib=/opt/dionaea/lib/
```

Τέλος, στο ίδιο μονοπάτι με τις εντολές make και make install το Dionaea έχει πλέον εγκατασταθεί.

Κατά την παραμετροποίηση, απαιτείται η εκτέλεση των παρακάτω ενεργειών στο αρχείο παραμέτρων του Dionaea dionaea.conf το οποίο βρίσκεται στον κατάλογο /opt/dionaea/etc/dionaea.

- Στη γραμμή 5 απαιτείται η τροποποίηση της τιμής levels, ώστε να μην καταγράφεται πληροφορία εκσφαλμάτωσης, (debug information).

  levels = “all, -debug”
- Στη γραμμή 11 απαιτείται η συμπλήρωση του προσωπικού λογαριασμού ηλεκτρονικού ταχυδρομείου ώστε τα αποτελέσματα από τις υπηρεσίες ανάλυσης κακόβουλου λογισμικού να λαμβάνονται όπου επιθυμεί ο χρήστης.
- Στη γραμμή 66 απαιτείται η τροποποίηση του mode σε manual.
- Στη γραμμή 167 απαιτείται να αναγράφεται το network interface που χρησιμοποιεί το μηχάνημα στο οποίο είναι εγκατεστημένο το Dionaea, στην προκειμένη το eth0.
- Στη γραμμή 137 απαιτείται εισαγωγή του κλειδιού API της υπηρεσίας VirusTotal.

apikey = "00be8d5efcc377cec5d8b7aec08f1f19a02cc16a9e569d26118426f00cd06e02”

- Στις γραμμές 456, 457 και 462 απαιτείται η αφαίρεση της διπλής καθέτου ώστε τα δεδομένα να μπορούν να καταγραφούν στην SQLite βάση δεδομένων αλλά και το Dionaea να συνεργάζεται με την υπηρεσία VirusTotal και το εργαλείο p0f.

Όλη η ανωτέρω διαδικασία αποτελεί τον ενδεδειγμένο τρόπο εγκατάστασης του Dionaea Honeypot, όπως αυτός περιγράφεται στην επίσημη ιστοσελίδα του Dionaea [59]. Ωστόσο από τον Απρίλη του 2012 ο οργανισμός http://www.honeynet.org/ δημιούργησε repositories όπου φορτώνονται όλα τα απαραίτητα πακέτα και οι βιβλιοθήκες του Dionaea και άλλων Honeypots ώστε να διευκολύνεται η διαδικασία εγκατάστασης.

Οπότε με προσθήκη των παρακάτω γραμμών στο αρχείο /etc/apt/sources.list

deb http://ppa.launchpad.net/honeynet/nightly/ubuntu precise main
deb-src http://ppa.launchpad.net/honeynet/nightly/ubuntu precise main

και στη συνέχεια με την εκτέλεση της εντολής

sudo apt-get install dionaea

το Dionaea εγκαθίσταται. [68]

12.4 Π4 – Εγκατάσταση Glastopf

Για την εγκατάσταση του Glastopf απαιτείται η εγκατάσταση των γλωσσών, πακέτων – βιβλιοθηκών python2.7, python-openssl, python-gevent, libevent-dev, python2.7-dev, liblapack-dev, python-chardet, python-requests, python-sqlalchemy, python-lxml, python-beautifulsoup, mongodb, python-pip, python-dev, python-numpy, python-setuptools, python-numpy-dev, python-scipy, libatlas-dev, g++, git, php5, php5-dev και pip.

Στη συνέχεια για τη δημιουργία του php sandbox απαιτείται η εκτέλεση των παρακάτω εντολών

cd /opt

sudo git clone git://github.com/glastopf/BFR.git

cd BFR

sudo phpize

sudo ./configure - -enable-bfr

sudo ./configure - -enable-bfr

sudo make && sudo make install

καθώς και η προσθήκη της παρακάτω γραμμής στο αρχείο php.ini, όπου στην τιμή του zend_extension θα εισαχθεί η τιμή που αναγράφεται στην κονσόλα μετά την εκτέλεση των παραπάνω εντολών.

zend_extension = /usr/lib/php5/20090626+lfs/bfr.so

Με την εκτέλεση της παρακάτω εντολής εγκαθίσταται το Glastopf.

sudo pip install glastopf

και με χρήση των παρακάτω εντολών δημιουργείται το Glastopf περιβάλλον το οποίο απαιτείται για την εκτέλεση στιγμιότυπου του Glastopf.

cd /opt

sudo mkdir myhoneypot

cd myhoneypot

sudo glastopf-runner

Οπότε στον κατάλογο /opt/myhoneypot έχει δημιουργηθεί ένα αρχείο παραμετροποίησης glastopf.cfg.

Προτού εκτελεστεί η εντολή εκκίνησης του Glastopf, απαιτείται σχετική παραμετροποίηση ώστε οι συνδέσεις προς τη θύρα 80 του υπολογιστή να πραγματοποιούνται προς το Glastopf και να καταγράφονται σε MySQL βάση δεδομένων.

Για τη δημιουργία MySQL βάσης δεδομένων απαιτείται η εκτέλεση των παρακάτω εντολών

create database glaspot;

create user ‘glaspot’@’localhost’ identified by ‘glaspot’;

grant all privileges on glaspot.* to ‘glaspot’@’localhost’;

flush privileges;

Στο αρχείο παραμετροποίησης glastopf.cfg απαιτείται η σύνδεση του Glastopf με τη MySQL βάση δεδομένων που μόλις δημιουργήθηκε.

```
enabled = True
#connection_string = sqlite:///db/glastopf.db
connection_string = mysql://glaspot:glaspot@localhost/glaspot
```

Τέλος, στο ίδιο αρχείο απαιτείται η αλλαγή της θύρας του Glastopf από 8080 σε 80, ενώ απαιτείται στο μηχάνημα και η αλλαγή του Apache διακομιστή από τη θύρα 80 στη θύρα 8080 ώστε στη θύρα 80 να «ακούει» για συνδέσεις μόνο το Glastopf.

Με την εκτέλεση της εντολής

```
sudo glastopf-runner > /dev/null 2>&1 &
```

το Glastopf εκτελείται στο παρασκήνιο και δέχεται συνδέσεις στη θύρα 80, τις οποίες καταγράφει σε MySQL βάση δεδομένων.

12.5 Π5 – Κώδικας Google Chart – παράδειγμα

Thesis.php

```
<html>
 <head>
  <!--Load the AJAX API-->
  <script type="text/javascript" src="https://www.google.com/jsapi"></script>
  <script type="text/javascript">
   // Load the Visualization API and the piechart package.
   google.load('visualization', '1.0', {'packages':['corechart']});
   // Set a callback to run when the Google Visualization API is loaded.
   google.setOnLoadCallback(drawChart);

   // Callback that creates and populates a data table,
   // instantiates the pie chart, passes in the data and
   // draws it.

   function drawChart() {
    // Create the data table.
    var data = new google.visualization.DataTable();
    data.addColumn('string', 'password');
    data.addColumn('number', 'COUNT(password)');
    data.addRows([

<?php
$con = mysqli_connect('localhost','root','honeydrive','kippo');
if (!$con)
{
        die('Could not connect: ' . mysqli_error($con));
}
$sql='SELECT password, COUNT(password) '
```

```
        ."FROM auth "
        ."WHERE password <> '' "
        ."GROUP BY password "
        ."ORDER BY COUNT(password) DESC "
        ."LIMIT 10" ;

$result = mysqli_query($con,$sql);
while($row = mysqli_fetch_array($result))
 {
?>
        ['<?=$row['password']?>',<?=$row['COUNT(password)']?>],
<?php
} ?> ]);
      // Set chart options
      var options = {'title':'Top 10 passwords',
                     'width':400,
                     'height':300};
      // Instantiate and draw our chart, passing in some options.
      var chart = new google.visualization.ColumnChart(document.getElementById('chart_div'));
      chart.draw(data, options);
    }
  </script>
 </head>
 <body>
  <!--Div that will hold the pie chart-->
  <div id="chart_div"></div>
 </body>
</html>

<?php
        mysqli_close($con);
?>
```

12.6 Π6 – Kippo επερωτήσεις

```
--top 10 usernames (1)
SELECT username, COUNT(username)
FROM auth
WHERE username <> ''
GROUP BY username
ORDER BY COUNT(username) DESC
LIMIT 10;

--top 10 passwords (2)
SELECT password, COUNT(password)
FROM auth
WHERE password <> ''
```

```
GROUP BY password
ORDER BY COUNT(password) DESC
LIMIT 10;

--top 10 username-password combination (3)
SELECT username+'-'+password AS USERNAME-PASSWORD, COUNT(username)
FROM auth
WHERE username <> '' AND password <> ''
GROUP BY username, password
ORDER BY COUNT(username) DESC
LIMIT 10;

--success ratio (4)
SELECT success, COUNT(success)
FROM auth
GROUP BY success
ORDER BY success;

--most successful logins per day (5)
SELECT COUNT(session), timestamp
FROM auth
WHERE success = 1
GROUP BY DAYOFYEAR(timestamp)
ORDER BY COUNT(session) DESC
LIMIT 20;

--most attacks per day (6)
SELECT COUNT(session), timestamp
FROM auth
GROUP BY DAYOFYEAR(timestamp)
ORDER BY COUNT(session) DESC
LIMIT 20 ;

--attacks during 24hour period (7)
SELECT ROUND((timestamp%(3600*24))/3600) AS hour,COUNT(*)
FROM auth
GROUP BY ROUND((timestamp%(3600*24))/3600);

--number of connections per ip (8)
SELECT ip, COUNT(ip)
FROM sessions
GROUP BY ip
ORDER BY COUNT(ip) DESC
LIMIT 20 ;

--successful attacks (9)
```

```
SELECT sessions.ip, COUNT(sessions.ip)
FROM sessions INNER JOIN auth ON sessions.id = auth.session
WHERE auth.success = 1
GROUP BY sessions.ip
ORDER BY COUNT(sessions.ip) DESC
LIMIT 20 ;

--most input commands used (10)
SELECT input, COUNT(input)
FROM input
GROUP BY input
ORDER BY COUNT(input) DESC
LIMIT 20;
```

12.7 Π7 – Dionaea επερωτήσεις

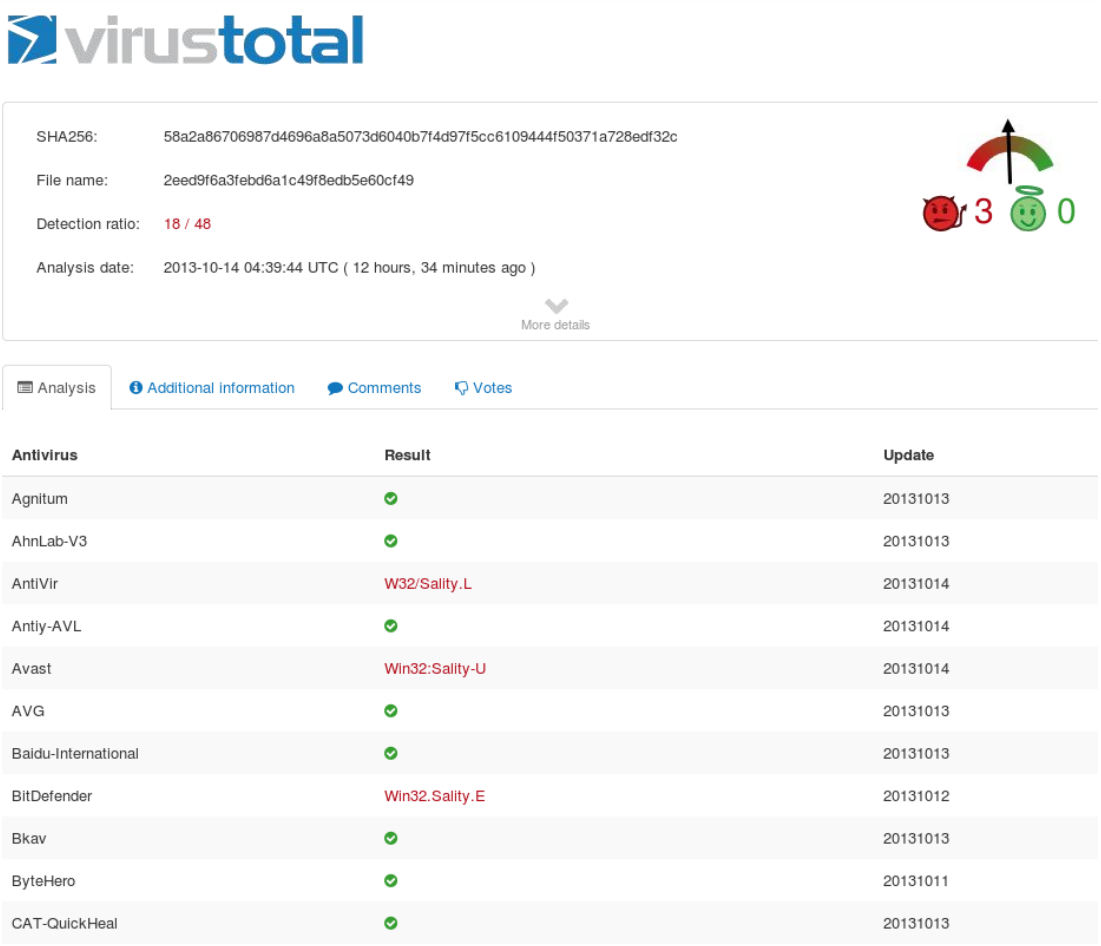
```
--attacks per protocol (1)
SELECT COUNT(local_port) AS hitcount, local_port AS port
FROM connections
GROUP BY local_port;

--attacks during 24hour period (2)
SELECT ROUND((connection_timestamp%(3600*24))/3600) AS hour,COUNT(*)
FROM connections
WHERE connection_parent IS NULL
GROUP BY ROUND((connection_timestamp%(3600*24))/3600);

--top 10 downloads (3)
SELECT COUNT(download_md5_hash), download_md5_hash
FROM downloads GROUP BY download_md5_hash
ORDER BY COUNT(download_md5_hash) DESC LIMIT 10;

--top 10 downloads group by distinct url(4)
SELECT count (distinct "download_url"), "download_md5_hash"
from downloads
group by "download_md5_hash"
order by count (distinct "download_url")
desc limit 10;

--top 10 attacks (5)
SELECT COUNT(remote_host), remote_host
FROM connections
GROUP BY remote_host
ORDER BY COUNT(remote_host) DESC LIMIT 10;

--downloads per ip (6)
SELECT COUNT(*),download_url
```

```
FROM downloads GROUP BY download_url
ORDER BY COUNT(*) DESC LIMIT 10;

--os used (7)
SELECT COUNT(*), p0f_genre
FROM p0fs
GROUP BY p0f_genre
ORDER BY COUNT(*) DESC;
```

12.8 Π8 – Glastopf επερωτήσεις

```
--requests per pattern (1)
SELECT pattern, count(pattern)
FROM `events`
GROUP BY pattern
ORDER BY count(pattern);

--requests per ip (2)
SELECT COUNT(source),
SUBSTRING(source, 1, locate( ':', source ) -1 ) AS stripped
FROM EVENTS
GROUP BY stripped
ORDER BY COUNT( stripped ) DESC
LIMIT 20;

--intext requests (3)
SELECT count, content
FROM intext
ORDER BY count DESC
LIMIT 10;

--intitle requests (4)
SELECT count, content
FROM intitle
ORDER BY count DESC
LIMIT 10;

--inurl requests (5)
SELECT count, content
FROM inurl
ORDER BY count DESC
LIMIT 10;
```

12.9 Π9 – VirusTotal Analysis Report



virustotal

SHA256: 58a2a86706987d4696a8a5073d6040b7f4d97f5cc6109444f50371a728edf32c

File name: 2eed9f6a3febd6a1c49f8edb5e60cf49

Detection ratio: 18 / 48

Analysis date: 2013-10-14 04:39:44 UTC (12 hours, 34 minutes ago)

3 0

More details

Analysis | Additional information | Comments | Votes

Antivirus	Result	Update
Agnitum		20131013
AhnLab-V3		20131013
AntiVir	W32/Sality.L	20131014
Antiy-AVL		20131014
Avast	Win32:Sality-U	20131014
AVG		20131013
Baidu-International		20131013
BitDefender	Win32.Sality.E	20131012
Bkav		20131013
ByteHero		20131011
CAT-QuickHeal		20131013

Εικόνα 12-1: VirusTotal Analysis Report.

12.10 Π10 – Anubis Analysis Report

[##]

Analysis Report for 730498b8a6c676e2298d9b1ad7dd5d10

MD5: 730498b8a6c676e2298d9b1ad7dd5d10

[##]

Summary:

- Autostart capabilities:

This executable registers processes to be executed at system start. This could result in unwanted actions to be performed automatically.

- Changes security settings of Internet Explorer:

This system alteration could seriously affect safety surfing the World Wide Web.

- Performs Address Scan:

The executable scans a range of IP Addresses. In most cases these scans identify more potential vulnerable targets.

- Performs File Modification and Destruction:

The executable modifies and destructs files which are not temporary.

- Performs Registry Activities:

The executable reads and modifies registry values. It also creates and monitors registry keys.

[==]

Table of Contents

[==]

- General information
- 730498b8a6.exe
  a) Registry Activities
  b) File Activities
  c) Network Activities
  d) Other Activities

[##]

1. General Information

[##]

[==]

Information about Anubis' invocation

[==]

Time needed: 240 s
Report created: 12/29/10, 10:44:53 UTC
Termination reason: Timeout
Program version: 1.74.3362

[==]

General information about this executable

[==]

Analysis Reason: Primary Analysis Subject
Filename: 730498b8a6.exe
MD5: 730498b8a6c676e2298d9b1ad7dd5d10
SHA-1: ac7e4883fe06b70a5cc0e7812cbcab88d9eb85ab
File Size: 45056 Bytes
Command Line: "C:\730498b8a6.exe"
Process-status
at analysis end: alive
Exit Code: 0

12.11 Π11 – Norman Sandbox Analysis Report

730498b8a6c676e2298d9b1ad7dd5d10: INFECTED with W32/Malware (Signature: win32/SB/Downloader)

[DetectionInfo]

* Filename: C:\analyzer\scan\730498b8a6c676e2298d9b1ad7dd5d10.
* Sandbox name: W32/Malware.
* Signature name: win32/SB/Downloader.
* Compressed: NO.
* TLS hooks: NO.
* Executable type: Application.
* Executable file structure: OK.
* Filetype: PE_I386.

[General information]

* File length: 45056 bytes.
* MD5 hash: 730498b8a6c676e2298d9b1ad7dd5d10.
* SHA1 hash: ac7e4883fe06b70a5cc0e7812cbcab88d9eb85ab.
* Entry-point detection: Microsoft Visual C++.

[Changes to registry]

* Accesses Registry key "HKLM\Software\Microsoft\Windows\CurrentVersion\Run".
* Creates value "PHIME2005"="C:\sample.exe /SYNC" in key "HKLM\Software\Microsoft\Windows\CurrentVersion\Run".

[Network services]

* Downloads file from http://fukyu.jp/updata/ACCI3.jpg as C:\WINDOWS\system32\msupd.exe.
* Connects to "fukyu.jp" on port 80 (TCP).

[Process/window information]

* Will automatically restart after boot (I'll be back...).

13. Πινακας Ορολογιας

Ξενόγλωσσος Όρος	Ελληνικός Όρος
Hacker	εισβολέας υπολογιστικών συστημάτων – παραβιαστής λογισμικού
Bug	προγραμματιστικό σφάλμα
Personal Digital Assistant, (PDA)	προσωπικός ψηφιακός βοηθός
Eavesdropping	Υποκλοπή
Denial Of Service attack, (DOS)	Επίθεση Άρνησης Εξυπηρέτησης
IP spoofing	παραπλάνηση με χρήση IP διευθυνσιοδότησης
Network Intrusion	δικτυακή εισβολή
malware, (malicious software)	λογισμικό κακόβουλου τύπου
Script	πρόγραμμα σεναρίου
virus	Ιός
Trojan horse	Δούρειος ίππος
worm	Σκουλήκι
Trapdoor	Κερκόπορτα
logic bomb	λογική βόμβα
phishing	ηλεκτρονική εξαπάτηση
Firewall	τείχος προστασίας
Intrusion Detection System, (IDS)	Σύστημα Ανίχνευσης Παρείσφρησης
Anomaly Detection System, (ADS)	Σύστημα Ανίχνευσης Ανωμαλιών
network sniffer	πρόγραμμα παρακολούθησης δικτύου

Spam	ανεπιθύμητη ηλεκτρονική αλληλογραφία
shell code	κώδικας κελύφους
Client	Πελάτης
Server	διακομιστής – εξυπηρετητής
web server	Διαδικτυακός διακομιστής
web browser	Διαδικτυακός φυλλομετρητής
Throughput	Ρυθμαπόδοση
sandbox	πρόγραμμα που διαθέτει τη δυνατότητα απομονωμένης εκτέλεσης σε υπολογιστή, ώστε να μην επηρεάζεται οποιοδήποτε άλλο πρόγραμμα αλλά και να μη δεσμεύεται οποιοσδήποτε άλλος υπολογιστικός πόρος
Spyware	λογισμικό κατασκοπίας κακόβουλου περιεχομένου – εμπεριέχεται κρυφά σε άλλες εφαρμογές και εγκαθίσταται όταν εγκατασταθεί και η εφαρμογή που το φέρει. Χρησιμοποιείται για να παρακολουθεί τον μολυσμένο υπολογιστή και να εκμαιεύει ευαίσθητες πληροφορίες.
Adware	λογισμικό κακόβουλου περιεχομένου που εγκαθιστά διαφημίσεις
Hoax	τύπος ανεπιθύμητης ηλεκτρονικής αλληλογραφίας
proxy server	ενδιάμεσος διακομιστής
Thread	Νήμα
Socket	φορέας υποδοχής
Module	Ενότητα

SQL injection	εκμαίευση ευαίσθητων πληροφοριών με χρήση επερωτήσεων
instant messaging	υπηρεσία άμεσων μηνυμάτων
Virtualization	Εικονικοποίηση
PHPNuke	Σύστημα διαχείρισης περιεχομένου που χρησιμοποιείται κυρίως για δημοσίευση ειδήσεων στο Διαδίκτυο. Χρησιμοποιεί PHP και MySQL.
OSCommerce	Σύστημα διαχείρισης περιεχομένου που χρησιμοποιείται κυρίως για σκοπούς ηλεκτρονικού εμπορίου. Χρησιμοποιεί PHP και MySQL.
cross-site scripting (XSS)	εκμετάλλευση διάφορων ευπαθειών υπολογιστικών συστημάτων με εισαγωγή κώδικα HTML ή Javascript σε διαδικτυακούς τόπους
message board	πίνακας μηνυμάτων
Nmap	εργαλείο ανοιχτού κώδικα που χρησιμοποιείται για την παρακολούθηση κίνησης δικτύου και τον έλεγχο ασφάλειας δικτύου
WireShark	εργαλείο ανοιχτού κώδικα που χρησιμοποιείται για την παρακολούθηση κίνησης δικτύου, εντοπισμό και επίλυση δικτυακών προβλημάτων
ClamAV	αντι-ιικό λογισμικό ανοιχτού κώδικα
Wardriver	κινούμενος χρήστης λογισμικού που ανιχνεύει την ύπαρξη ασύρματων δικτύων
NetStumbler	λογισμικό ανίχνευσης ύπαρξης ασύρματων δικτύων

Script Kiddie	χρήστης με μη εξειδικευμένες γνώσεις πληροφορικής που δύναται με χρήση scripts ή προγραμμάτων που δεν έχουν αναπτυχθεί από τον ίδιο να επιτεθεί σε υπολογιστικά συστήματα ή δίκτυα
Demilitarized Zone, (DMZ)	αποστρατικοποιημένη ζώνη δικτύου,
e-shop	ηλεκτρονικό κατάστημα

14. Συντμησεις – Αρκτικολεξα – Ακρωνυμια

DOS	Denial Of Service
DOS	Distributed Denial Of Service
TCP	Transmission Control Protocol
ICMP	Internet Control Message Protocol
IP	Internet Protocol
IPsec	Internet Protocol Security
IDS	Intrusion Detection System
ADS	Anomaly Detection System
HTTP	Hypertext Transfer Protocol
HTTPS	HyperText Transfer Protocol over SSL (Secure Socket Layer)
HTML	HyperText Markup Language
SSL	Secure Socket Layer
FTP	File Transfer Protocol
TFTP	Trivial File Transfer Protocol
IIS	Internet Information Services
ISP	Internet Service Provider
TLS	Transport Layer Security
SMB	Server Message Protocol
UDP	User Datagram Protocol
MSSQL	Microsoft SQL Server
TDS	Tabular Data Stream

VoIP	Voice Over IP
SIP	Session Initiated Protocol
RTP	Real-time Transport Protocol
API	Application Programming Interface
SMTP	Simple Mail Transfer Protocol
IMAP	Internet Message Access Protocol
DNS	Domain Name System
SSH	Secure Shell
POP	Post Office Protocol
GRE	Generic Routing Encapsulation
ARP	Address Resolution Protocol
SOAP	Simple Object Access Protocol
URL	Uniform Resource Locator
DOM	Document Object Model
RFI	Remote File Inclusion
LFI	Local File Inclusion
UML	Unified Modeling language
TTY	TeleTYpe
DMZ	Demilitarized Zone
OVA	Open Virtual Appliance
EPMAP	End Point Mapper
PDA	Personal Digital Assistant

CIFS	Common Internet File System
NAT	Network Address Translation
RDP	Remote Desktop Protocol
IRC	Internet Relay Chat

15. Βιβλιογραφία

[1] Spitzner L., *Honeypots: Tracking Hackers.* Boston, MA: Addison-Wesley Professional, 2003.

[2] PC Magazine's Encyclopedia, "Definition of Honeypot", (2009). [Online]. Available: http://www.pcmag.com/encyclopedia/term/44335/honeypot. Accessed: 20/12/2012.

[3] L. R. Even, SANS Institute, "Honeypot Systems Explained – Overview", (2000). [Online]. Available: http://www.sans.org/security-resources/idfaq/honeypot3.php. Accessed: 02/01/2013.

[4] E. Peter and T. Schiller, *A Practical Guide to Honeypots.* USA: Washington University, 2008. [Online]. Available: http://www.cse.wustl.edu/~jain/cse571-09/ftp/honey/. Accessed: 25/12/2012.

[5] Ι. Σ. Τρούλης, *Μελέτη Χαμηλής και Υψηλής Αλληλεπίδρασης Honeypots.* Πτυχιακή εργασία. Πανεπιστήμιο Πειραιώς, Τμήμα Διδακτικής της Τεχνολογίας και Ψηφιακών Συστημάτων. Ελλάδα, (Μάιος 2010).

[6] N. Provos, "A Virtual Honeypot Framework". *Proceedings of the 13th USENIX security symposium,* Vol. 132. (2004).

[7] G. Wicherski, "Medium Interaction Honeypots". *German Honeynet Project*, (2006).

[8] I. Mokube and M. Adams, "Honeypots: Concepts, Approaches and Challenges." *Proceedings of the 45th annual southeast regional conference,* (pp. 321-326). ACM, (2007).

[9] R. Baumann and C. Plattner, *Honeypots.* MSc Thesis. Swiss Federal Institute of Technology Zurich. Switzerland, (February 2002).

[10] Specter – Intrusion Detection System. (2013). [Online]. Available: http://www.specter.com Accessed: 02/01/2013.

[11] N. Provos, *Developments of the Honeyd Virtual Honeypot*. (2008). [Online]. Available: http://www.honeyd.org. Accessed: 05/01/2013.

[12] The Honeynet Project: HoneyC. (2008). [Online]. Available: https://projects.honeynet.org/honeyc. Accessed: 05/01/2013.

[13] PhoneyC – Python Honeyclient. (2009). [Online]. Available: https://code.google.com/p/phoneyc/. Accessed: 07/01/2013.

[14] N. Provos, *SpyBye – Finding malware*. (2007). [Online]. Available: http://www.monkey.org/~provos/spybye/. Accessed: 10/01/2013.

[15] Nepenthes – Finest Collection. (2005). [Online]. Available: http://nepenthes.carnivore.it/. Accessed: 07/01/2013.

[16] Angelo Dell'Area, *Thug – GitHub*. (2011). [Online]. Available: https://github.com/buffer/thug. Accessed: 10/01/2013.

[17] Google Code, *Kippo – SSH Honeypot*. (2009). [Online]. Available: https://code.google.com/p/kippo/. Accessed: 12/01/2013.

[18] F. Cohen and associates. *The Deception Toolkit Homepage and Mailing List*. (1998). [Online]. Available: http://www.all.net/dtk/index.html. Accessed 02/01/2013.

[19] G. Wicherski, "Sammeln von Malware in nicht-nativer Umgebung". (2005). [Online]. Available: http://web.archive.org/web/20090219063457/http://pixel-house.net/mwc_facharbeit.pdf. Accessed: 20/12/2012.

[20] Honeyspider Network 2. (2013). [Online]. Available: http://www.honeyspider.org/. Accessed: 02/01/2013.

[21] Australian Honeynet Project, *Tool release: Trigona*. (2010). [Online]. Available: http://honeynet.org.au/?q=node/63. Accessed: 01/06/2012.

[22] The Honeynet Project, *Capture-HPC Client Honeypot / Honeyclient*. (2007). [Online]. Available: https://projects.honeynet.org/capture-hpc/. Accessed: 05/11/2012.

[23] Microsoft Research, *Strider HoneyMonkey.* (2005). [Online]. Available: http://research.microsoft.com/en-us/um/redmond/projects/strider/honeymonkey/article.aspx. Accessed: 10/12/2012

[24] Y. Wang et al, *Automated Web Patrol with Strider HoneyMonkeys: Finding Web Sites That Exploit Browser Vulnerabilities*. Technical Report MSR-TR-2005-72. Microsoft Research, Microsoft Corporation, One Microsoft Way. Redmond, WA 98052, (2005).

[25] H. Bos, “Shelia: A client-side honeypot for attack detection”. (2009). [Online]. Available: http://www.cs.vu.nl/~herbertb/misc/shelia/. Accessed: 15/12/2012.

[26] A. Moshchuk et al., *“*A Crawler-based Study of Spyware on the Web”. *13th Annual Network and Distributed System Security Symposium (NDSS)*. San Diego, (2006).

[27] XNos Labs, *Web Exploit Finder.* (2006). [Online]. Available: http://www.xnos.org/security/web-exploit-finder.html. Accessed: 15/01/2013.

[28] I. Koniaris, “HoneyDrive”. *BruteForce Lab’s Blog*. (2012). [Online]. Available: http://bruteforce.gr/honeydrive. Accessed: 10/10/2012.

[29] L. Spitzner, *To Build a Honeypot*. (1999). [Online]. Available: http://www.spitzner.net/honeypot.html. Accessed: 08/09/2012

[30] The Honeynet Project, *Know Your Enemy: Honeynets. What a honeynet is, its value, how it works, and risks/issues involved*. (2005). [Online]. Available: http://old.honeynet.org/papers/honeynet/. Accessed: 09/07/2012.

[31] L. Spitzner, “Honeytokens: The Other Honeypot”. *Security Focus, Infocus 1713*. (2003). [Online]. Available: http://www.symantec.com/connect/articles/honeytokens-other-honeypot. Accessed: 13/08/2012.

[32] The Monkey-Spider Project. (2007). [Online]. Available: http://monkeyspider.sourceforge.net/index.html. Accessed: 11/01/2013.

[33] LaBrea: “Sticky” honeypot and IDS. (2003). [Online]. Available: http://labrea.sourceforge.net/labrea-info.html. Accessed: 03/02/2013.

[34] G. Bakos. *Tiny Honeypot – resource consumption for the good guys*. (2002). [Online]. Available: http://www.alpinista.org/thp. Accessed: 01/02/2013.

[35] J. Göbel, *Amun: A Python Honeypot*. Technical Report. Laboratory for Dependable Distributed Systems, University of Mannheim, Germany. (2008).

[36] HiHAT – High-Interaction Honeypot Analysis Tool. (2007). [Online]. Available: http://hihat.sourceforge.net/index.html. Accessed: 09/02/2013.

[37] Google, *ghh – The “Google Hack” Honeypot*. (2005). [Online]. Available: http://ghh.sourceforge.net/. Accessed: 12/02/2013.

[38] Black alchemy, *FakeAp*. (2002). [Online]. Available: http://www.blackalchemy.to/project/fakeap/. Accessed: 18/01/2013.

[39] L. Spitzner, *Honeypot Farms.* (2003). [Online]. Available: http://www.symantec.com/connect/articles/honeypot-farms.

[40] Nageshri B. Karhade, Yogini K. Kothekar, *Honeyweb: a web-based high interaction client honeypot*. International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622, National Conference on Emerging Trends in Engineering & Technology (VNCET-30 Mar’12).

[41] J. Göbel and A. Dewald, *Client-Honeypots: Exploring Malicious Websites*. Oldenbourg Verlag. (2010).

[42] Anagnostakis, K. G., et al. “Detecting targeted attacks using shadow honeypots”. *Proceedings of the 14th conference on USENIX Security Symposium*. ACM, 2005. 129-144.

[43] Σ. Κάτσικας, *Ασφάλεια Υπολογιστών: Προστασία και Ασφάλεια Συστημάτων Υπολογιστών*. Ελληνικό Ανοικτό Πανεπιστήμιο, Σχολή Θετικών Επιστημών και Τεχνολογίας, τμήμα Πληροφορικής. Πάτρα, (2001).

[44] R. Moir, “Defining Malware: FAQ”. Microsoft TechNet. (2003). [Online]. Available: http://technet.microsoft.com/en-us/library/dd632948.aspx. Accessed: 10/03/2013.

[45] F. Cohen. "Computer viruses: theory and experiments." *Computers & security* 6.1 (1987): pp. 22-35.

[46] K. Thompson. “Reflections on trusting trust.” *Communications of the ACM,* Vol. 27, No. 8 (1984): pp. 761-763.

[47] Brunner J (1975). *The Shockwave Rider*. New York: Ballantine Books.

[48] *United States v. Morris* (1991). 928 F. 2d 504, 505 (2d Cir. 1991).

[49] S. Northcutt, SANS Institute, “Security Laboratory – Methods of Attack Series: Logic Bombs, Trojan Horses and Trap Doors”, (2007). [Online]. Available: http://www.sans.edu/research/security-laboratory/article/log-bmb-trp-door. Accessed: 05/03/2013.

[50] Puri, Ramneek. "Bots & botnet: An overview." *SANS Institute 2003* (2003).

[51] P. Bacher, T. Holz et al. “Know your enemy:Tracking botnets”. *Honeynet Project*. (2008). [Online]. Available: http://www.honeynet.org/papers/bots. Accessed: 10/08/2013.

[52] MacAfee Security, *Rootkits, Part 1 of 3: The Growing Threat*. MacAfee AVERT Labs Whitepaper. (2006).

[53] Krutz, Ronald L., Russell Dean Vines, and Edward M. Stroz. *The CISSP prep Guide: Mastering the ten domains of Computer Security*. New York: Wiley, 2001.

[54] Guttman, Barbara, and Edward A. Roback. *An introduction to computer security: the NIST handbook*. DIANE Publishing, 1995.

[55] Ι. Κονιάρης, *Ανάλυση Κυβερνοεπιθέσεων με Honeypots Μεσαίας και Χαμηλής Αλληλεπίδρασης.* Πτυχιακή εργασία. Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Σχολή Θετικών Επιστημών, Τμήμα Πληροφορικής. Ελλάδα, (Οκτώβριος 2012).

[56] Google Code, *HonSSH*. (2013). [Online]. Available: https://code.google.com/p/honssh/. Accessed: 20/08/2013.

[57] Ι. Κονιάρης, «Ωραίο SSH honeypot, αλλά για το σπιτάκι». Περιοδικό DeltaHacker, τεύχος 6, σελ. 78-89, (2012).

[58] L. Rist. “Know Your Tools: Glastopf. A dynamic, low-interaction web application honeypot”. *Honeynet Project*. (2010). [Online]. Available: http://honeynet.org/papers/KYT_glastopf. Accessed: 15/08/2013.

[59] Dionaea, *dionaea catches bugs*. (2009). [Online]. Available: http://dionaea.carnivore.it/. Accessed: 09/02/2013.

[60] DionaeaFR, *DionaeaFR catches bugs.* (2013). [Online]. Available: http://rootingpuntoes.github.io/DionaeaFR/. Accessed: 20/09/2013.

[61] L. Spitzner, *The Value of Honeypots, Part Two: Honeypot Solutions and Legal Issues.* 2001. [Online]. Available: http://www.symantec.com/connect/articles/value-honeypots-part-two-honeypot-solutions-and-legal-issues. Accessed: 25/09/2013.

[62] AFP, *China, Indonesia lead sources of online Attacks: Study.* The Express Tribune. (2013). [Online]. Available: http://tribune.com.pk/story/581189/china-indonesia-lead-sources-of-online-attacks-study/. Accessed: 24/07/2013.

[63] C. Patrikakis, M. Masikos and O. Zouraкaki, *Distributed Denial of Service Attacks*. The Internet Protocol Journal, Vol. 7, No 4. (December 2004). [Online]. Available: http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_7-4/dos_attacks.html. Accessed: 23/09/2013.

[64] Andrew Waite, *InfoSanity Research – Offensive and Defensive IT Security,* (2010). [Online]. Available: http://www.infosanity.co.uk/resources/scripts/dionaea/dionaea-sqlquery-0_2.py. Accessed: 01/20/2013.

[65] Andrew Waite, *InfoSanity Research – Offensive and Defensive IT Security,* (2010). [Online]. Available: http://www.infosanity.co.uk/resources/scripts/dionaea/mimic-nepstats.py. Accessed: 01/20/2013.

[66] Ι. Κονιάρης, «Παγίδες για τα malware του κόσμου όλου». Περιοδικό DeltaHacker, τεύχος 7, σελ. 40-51, (2012).

[67] Google Hacking – *Database, Exploit Database.* (2013). [Online]. Available: http://www.exploit-db.com/google-dorks/. Accessed: 10/10/2013.

[68] devWerks IT-Security and Development, *Install Dionaea on Ubuntu 12.04 LTS (Precise Pangolin)*. (2013). [Online]. Available: http://devwerks.net/2013/03/10/install-dionaea-on-ubuntu-12-04-lts-precise-pangolin/. Accessed: 07/04/2013.