ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ
ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
Π.Μ.Σ . ΔΙΔΑΚΤΙΚΗΣ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ ΚΑΙ ΨΗΦΙΑΚΑ ΣΥΣΤΗΜΑΤΑ
ΚΑΤΕΥΘΥΝΣΗ ΨΗΦΙΑΚΩΝ ΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ

Εντοπισμός και Διαχείριση Κακόβουλών Δικτύων(botnets) με Χρήση Honeypots

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Θεοφύλακτος Ελευθέριος

Επιβλέπων Καθηγητής: ΣΩΚΡΑΤΗΣ ΚΑΤΣΙΚΑΣ

ΑΠΡΙΛΙΟΣ 2011

ΠΕΡΙΛΗΨΗ

Η συγκεκριμένη διπλωματική εργασία αφορά την μελέτη των δικτύων botnet και την ανάλυση των μεθόδων προστασίας από επιθέσεις, εστιάζοντας στα honeypots. Στόχος είναι η κατανόηση των κακόβουλων επιθέσεων μέσω των δικτύων botnets και περιγραφή και επισκόπηση της πλέον διαδεδομένης μεθόδου προστασίας των honeypots.

Συγκεκριμένα, γίνεται μια εισαγωγική αναφορά σε μερικά γενικά χαρακτηριστικά των botnet και των honeypot. Ακολουθεί μια ποιο εκτενεστέρη αναφορα στα δίκτυα botnet , στις τεχνικές απόκρυψης και ανιχνευσής τους αλλα και εργαλεια που υπάρχουν στα οποία εχουμε την δυνατοτήτα να ανιχνευθουν.Στο συγκεκριμένο εργαλείο που επικεντρονόμαστε είναι στα Honeypots στα οποία ακολουθεί αναλυσή του. Τέλος, προτείνουμε καποία εργαλεία διαχείρισης τους για την ανίχνευση των δικτύων botnet.

ΠΕΡΙΕΧΟΜΕΝΑ

Κεφάλαιο 1. Εισαγωγή

Στη σύγχρονη εποχή, η ασφάλεια των δικτύων αποτελεί ένα από τα σημαντικότερα θέματα της επιστήμης των υπολογιστών. Η ιλιγγιώδης ανάπτυξη του Διαδικτύου συνοδεύτηκε, δυστυχώς, από την εκθετική αύξηση του αριθμού των επιθέσεων. Κύριος στόχος των επιτιθέμενων ("*black hats*") είναι να εγκαταστήσουν κακόβουλο λογισμικό (*malware*) σε υπολογιστές ανυποψίαστων χρηστών και να αποκτήσουν πρόσβαση σε ευαίσθητα και προσωπικά δεδομένα. Συνεπώς, παρουσιάστηκε η ανάγκη μελέτης και επινόησης μεθοδολογιών οι οποίες θα συντελούσαν στην αντιμετώπιση του προβλήματος.

1.1 *Honeypots*

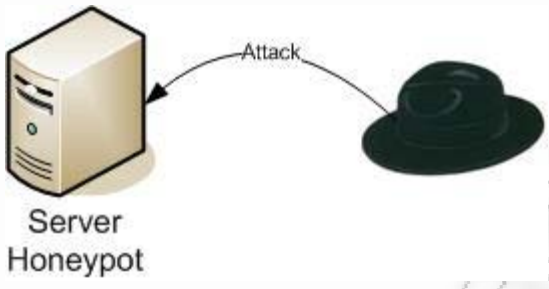
Η χρήση των *honeypots* είναι μία αποτελεσματική μέθοδος, η οποία παρέχει τη δυνατότητα συγκέντρωσης και επεξεργασίας δεδομένων που αφορούν τη δραστηριότητα των επιτιθέμενων. Σύμφωνα με τον ορισμό του Lance Spitzner, «ένα *honeypot* αποτελεί έναν πόρο πληροφοριακών συστημάτων, του οποίου η αξία έγκειται στην μη εξουσιοδοτημένη ή παράνομη χρήση του συγκεκριμένου πόρου» [1]. Τα δεδομένα συλλέγονται παρακολουθώντας τον επιτιθέμενο και εξετάζοντας τις ευπάθειες του πληροφοριακού συστήματος, τις οποίες εκμεταλλεύεται.

Η ιστορία των *honeypots* αρχίζει στα μέσα της δεκαετίας του 1980 και παρουσιάζει ιδιαίτερο ενδιαφέρον. Η ιδέα ενός συστήματος, το οποίο προσελκύει τους επιτιθέμενους των δικτύων εξελισσόταν για περισσότερο από μία δεκαετία. Σημαντικές ερευνητικές και αναπτυξιακές δραστηριότητες πραγματοποιήθηκαν από στρατιωτικούς, κυβερνητικούς και επιχειρηματικούς οργανισμούς. Ωστόσο, ελάχιστες πληροφορίες κοινοποιήθηκαν πριν το 1990 και μόλις πρόσφατα δημοσιεύθηκαν άρθρα και αναπτύχθηκε λογισμικό, τα οποία υποστηρίζουν τη συγκεκριμένη ιδέα. Στα τέλη του 20ου αιώνα, δημοσιεύσεις σχετικές με γεγονότα, ιδέες και έννοιες έθεσαν τα θεμέλια για την ανάπτυξη των *honeypots*.

Οι εξελίξεις στο πεδίο της ασφάλειας των δικτύων συνεχίστηκαν με έντονο ρυθμό στη διάρκεια της δεκαετίας του ’90. Ιδιαίτερα σημαντική υπήρξε η πραγμάτωση του Honeynet Project από τον Lance Spitzner το 1999 [2]. Ο Lance Spitzner συντέλεσε στη συγκρότηση μίας ομάδας επαγγελματιών της ασφάλειας, οι οποίοι επικεντρώθηκαν στην μελέτη και τη συγκέντρωση εξαιρετικά χρήσιμων πληροφοριών με στόχο τη γνωστοποίησή τους στους ενδιαφερόμενους. Το 2001 δημοσίευσαν το βιβλίο “Know Your Enemy: Learning about Security Threats” [3], του οποίου το περιεχόμενο αναφέρεται στην έρευνά τους και τα αποτελέσματά της. Η ερευνητική δραστηριότητα στα πλαίσια του Honeynet Project υποστηρίχθηκε από ένα βελτιωμένο και εξελιγμένο είδος *honeypot*, το οποίο αποτελεί ένα δίκτυο *honeypots*, ένα *honeynet*.

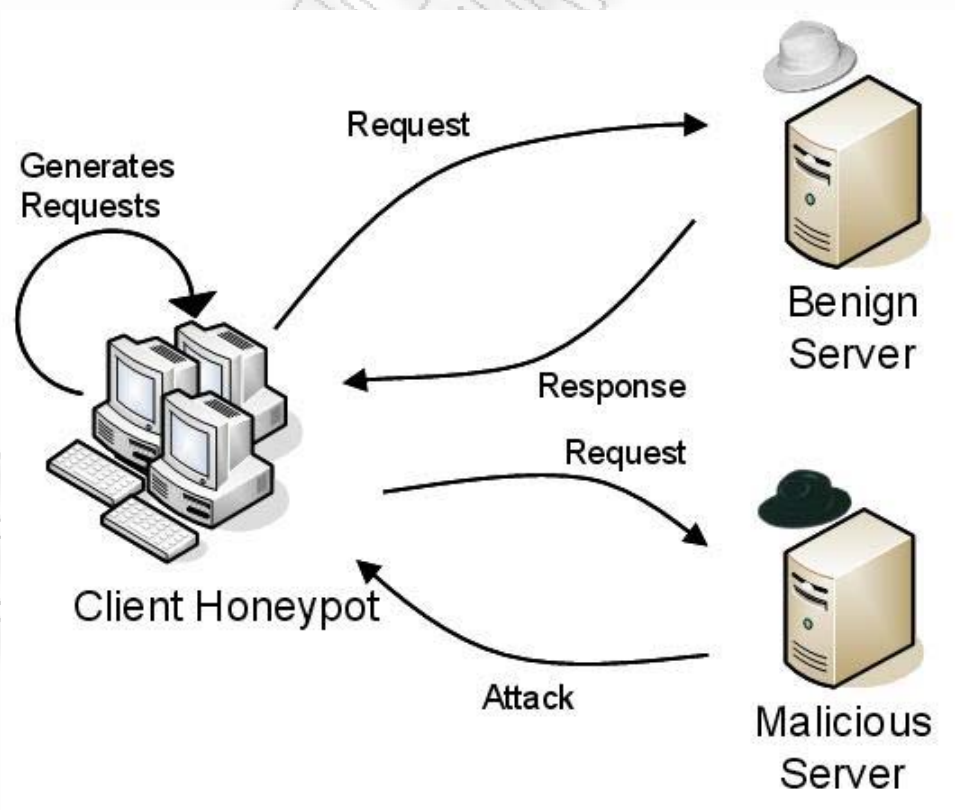
Η ανάπτυξη των *honeypots* συνεχίζεται τον 21° αιώνα με γνώμονα τις εξειδικευμένες γνώσεις που αποκτούνται σταδιακά από την παρακολούθηση των επιτιθέμενων. Ο σχεδιασμός και η υλοποίησή τους απαιτούν ιδιαίτερη προσοχή, ώστε να συντελέσουν αποτελεσματικά στην προστασία των πληροφοριακών συστημάτων. Τα *honeypots* διακρίνονται σε χαμηλής αλληλεπίδρασης (*low interaction honeypots*) και υψηλής αλληλεπίδρασης *honeypots* (*high interaction honeypots*), ανάλογα με το επίπεδο της δραστηριότητας του επιτιθέμενου. Η διαφορά τους έγκειται στο ότι ενώ τα χαμηλής αλληλεπίδρασης *honeypots* εξομοιώνουν συστήματα και υπηρεσίες, τα υψηλής αλληλεπίδρασης *honeypots* αποτελούν πραγματικά πληροφοριακά συστήματα, στα οποία ενεργούν οι επιτιθέμενοι.

Ενδιαφέρουσα είναι και η διάκριση των *honeypots* σε *server* και *client honeypots*. Η παραδοσιακή τεχνολογία των *server honeypots* (Σχήμα 1.1) εστιάζει αποκλειστικά στην προστασία των εξυπηρετητών από τις επιθέσεις των εισβολέων (*black hat*) που, εκμεταλλευόμενοι τις ευπάθειες των υπηρεσιών των εξυπηρετητών, προσπαθούν να παραβιάσουν το σύστημά τους.



Σχήμα 1.1 *Server Honeypot*

Τα *client honeypots* (Σχήμα 1.2) αποτελούν μία σύγχρονη τεχνολογία ασφάλειας, η οποία επιτρέπει την αναγνώριση κακόβουλων εξυπηρετητών που επιτίθενται εναντίον των χρηστών (*client-side attacks*). Ένα σύστημα του συγκεκριμένου είδους απαιτεί ενεργητική αλληλεπίδραση με έναν εξυπηρετητή. Τα *client-honeypots* επικοινωνούν με διάφορους εξυπηρετητές και τους διακρίνουν ανάλογα με τη φύση της δραστηριότητάς τους σε καλόβουλους και κακόβουλους (*benign/malicious servers*).



Σχήμα 1.2 *Client Honeypot*

Η αρχιτεκτονική ενός *client honeypot* διαμορφώνεται από τρία βασικά συστατικά στοιχεία [4]:

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

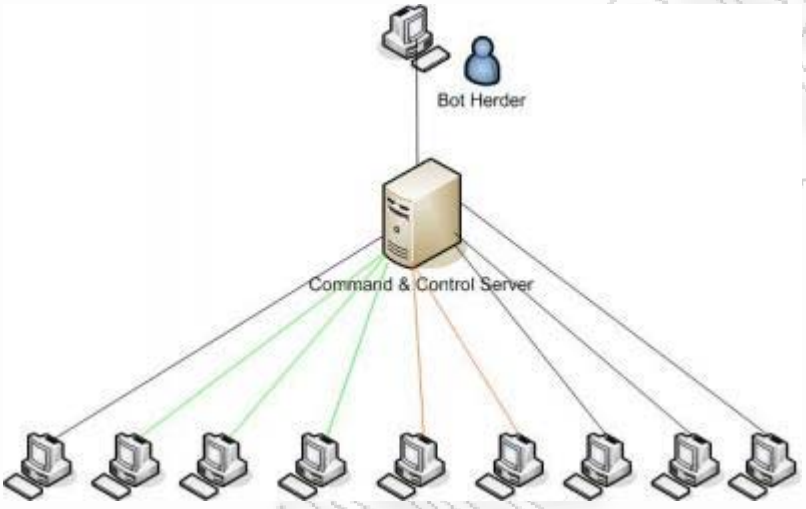
Επίσης, στα *client honeypots* εφαρμόζεται κάποια στρατηγική προστασίας, η οποία αποτρέπει οποιαδήποτε επίθεση εκτός των ορίων του. Συνήθως, αυτό επιτυγχάνεται με τη χρήση ενός τοίχους προστασίας (*firewall*) ή ενός εξειδικευμένου εικονικού συστήματος (*virtual machine*).

1.2 *Botnets*

Τα ro*BOT NET*works (*botnets*) είναι δίκτυα υπονομευμένων υπολογιστών ή αλλιώς δίκτυα προγραμμάτων ρομπότ, δηλαδή εφαρμογές που εκτελούν δράσεις για λογαριασμό χειριστή εξ΄αποστάσεως, οι οποίες εγκαθίστανται μυστικά στους υπολογιστές των θυμάτων. Πρόκειται για αυτοματοποιημένα προγράμματα που «τριγυρίζουν» στο διαδίκτυο, καταλαμβάνουν τους υπολογιστές και τους μετατρέπουν σε «ζόμπι». Τα μηχανήματα αυτά γνωστά με τον όρο *bots*, συναθροίζονται σε συστήματα που ονομάζονται *botnets* ή *zombie computers*, ενώ υπολογιστές σε σπίτια και επιχειρήσεις συνθέτουν μια τεράστια αλυσίδα κυβερνο-ρομπότ.

Τα κακόβουλα αυτά δίκτυα ενοικιάζονται συνήθως για δόλιους και αξιόποινους σκοπούς. Ενοικιαστές τους μπορεί να είναι αποστολείς ανεπίκλητων ηλεκτρονικών-μηνυμάτων (*spam*), δράστες «ηλεκτρονικού ψαρέματος» (*phishing*) και πωλητές κατασκοπευτικού ή άλλου κακόβουλου λογισμικού. Συχνά επίσης τα δίκτυα αυτά χρησιμοποιούνται για επιθέσεις μεγάλης κλίμακας που στρέφονται εναντίον συστημάτων πληροφοριών ή οργανισμών και ατόμων ακόμη και εναντίον των κρίσιμων υποδομών

πληροφόρησης ενός κράτους. Χρησιμοποιούνται επίσης για παράνομη εξόρυξη δεδομένων εν αγνοία των χρηστών, ενώ παράλληλα εγκαθιστούν κακόβουλα λογισμικά σε ακόμα περισσότερους υπολογιστές.



Σχήμα 1.3 Τυπικό Δίκτυο *Botnet*

Τα δίκτυα προγραμμάτων ρομπότ αποτελούν τη μάστιγα του διαδικτύου. Αυτά τα ενεργά ζόμπι-δίκτυα δημιουργήθηκαν από μια διαδικτυακή μαφία που συνεχώς αυξάνεται, με κίνητρο διαρκώς περισσότερο το κέρδος και όχι την πρόκληση διαταραχής και μόνο. Παράλληλα, αυξάνεται και ο αριθμός των μολυσμένων υπολογιστών, αφού οποιοσδήποτε υπολογιστής που συνδέεται στο διαδίκτυο είναι ευάλωτος. Πιο συγκεκριμένα, η προσβολή και η μετατροπή ενός υπολογιστή σε *bot* γίνεται με δύο τρόπους:

α) Προσβολή μέσω εκτέλεσης ενός κακόβουλου προγράμματος (*malware*) από τον ίδιο το χρήστη. Στην περίπτωση αυτή ο ανυποψίαστος χρήστης εκτελεί ένα πρόγραμμα χωρίς αυτός να γνωρίζει τον κίνδυνο δείχνοντας εμπιστοσύνη στην πηγή του προγράμματος.

β) Προσβολή μέσω εκμετάλλευσης (*exploit*) κάποιας ευπάθειας (*vulnerability*) του λειτουργικού συστήματος ή κάποιου άλλου προγράμματος ή υπηρεσίας του συστήματος. Στην περίπτωση αυτή κακόβουλοι χρήστες με ιδιαίτερες γνώσεις γνωστοί και ως *black*

hat hackers, χρησιμοποιούν προγράμματα τα οποία εκμεταλλεύονται τυχόν παραλείψεις στην πολιτική ασφαλείας των πληροφοριακών συστημάτων ή αδυναμίες στο λογισμικό των συστημάτων αυτών, έτσι ώστε να εκτελέσουν απομακρυσμένα κακόβουλο κώδικα και να αποκτήσουν πρόσβαση σε αυτά. Συχνά οι ευπάθειες στο λογισμικό εντοπίζονται στα λειτουργικά συστήματα, στα προγράμματα περιήγησης του παγκόσμιου ιστού (web browsers) και σε υπηρεσίες (*services*) σε εξυπηρετητές (*servers*) (π.χ. *web services*, *ftp services*, κ.τ.λ.).

Τα *botnets* έχουν εξελιχθεί σημαντικά και ο εντοπισμός τους είναι πλέον ιδιαίτερα δύσκολος. Τον τελευταίο χρόνο, τα *botnets* χρησιμοποιούν μια τεχνική που ονομάζεται *fast-flux*, με την οποία επιτυγχάνεται μια σειρά ταχύτατων αλλαγών διευθύνσεων διαδικτύου ώστε να μπορούν να εντοπίζονται δύσκολα και να προκαλούν μεγαλύτερη καταστροφή.

1.3 Στόχος της εργασίας

Το θέμα που εστιάζει η συγκεκριμένη διπλωματική εργασία, αφορά την μελέτη των δικτύων *botnet* και την ανάλυση των μεθόδων προστασίας από επιθέσεις, εστιάζοντας στα *honeypots*. Στόχος είναι η κατανόηση των κακόβουλων επιθέσεων μέσω των δικτύων *botnets* και περιγραφή και επισκόπηση της πλέον διαδεδομένης μεθόδου προστασίας των *honeypots*.

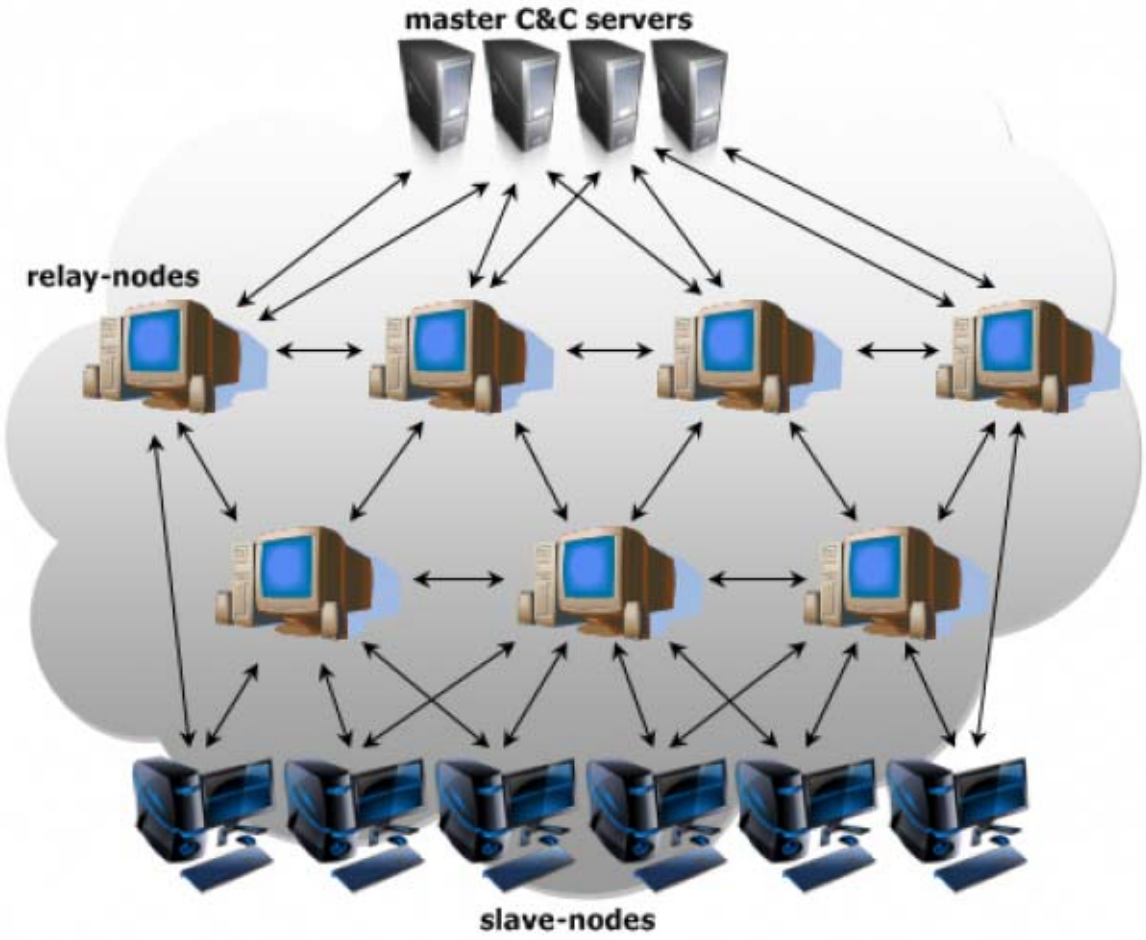
Κεφάλαιο 2. Δίκτυα Botnet

Το κεφάλαιο αυτό περιέχει μία εισαγωγή στα δίκτυα *botnet*, παρουσιάζοντας ένα σύντομο ιστορικό τους, τις χρήσεις τους και τον τρόπο λειτουργίας τους. Στο τέλος του κεφαλαίου γίνεται προσπάθεια παρουσίασης τους κλίματος που επικρατεί στο διαδίκτυο, σχετικά με τη χρήση τους μαζί με στατιστικά στοιχεία για την ανάπτυξη τους.

2.1 Τι Είναι τα Botnets

Ο όρος *Botnet* προέρχεται από τις λέξεις Ro*bot* και Network ενώ συχνά παρουσιάζεται και με τον όρο *Zombie Network*. Είναι ένα δίκτυο από υπολογιστές, αναφερόμενοι και ως *bot*s, που βρίσκονται υπό τον έλεγχο ενός ή περισσοτέρων κακόβουλων χρηστών, τους *Botmasters*. Συνήθως αποτελούνται από ένα μεγάλο αριθμό υπολογιστών, οικιακών και μη, και χρησιμοποιούνται για παράνομες δραστηριότητες, αποφέροντας στους κατόχους τους μεγάλα χρηματικά ποσά.

Για τη δημιουργία του δικτύου *Botnet*, αρχικά πραγματοποιείται η εγκατάσταση ενός προγράμματος στον υπολογιστή του χρήστη-θύμα, το οποίο δίνει τη δυνατότητα απομακρυσμένου ελέγχου, χωρίς ο χρήστης να το γνωρίζει. Στη συνέχεια ο υπολογιστής-*bot* επικοινωνεί με τον *botmaster*, με σκοπό την ανάθεση αποστολών. Αφού ολοκληρωθεί μία αποστολή το *bot* συνδέεται και πάλι με τον *botmaster* και του αποστέλλει τα αποτελέσματα. Περιοδικά το *bot* δύναται να ενημερώνεται, κάνοντας λήψη και εγκαθιστώντας νέο λογισμικό. Οι ενέργειας που αναφέρθηκαν αποτελούν τον κύκλο ζωής ενός τυπικού δικτύου.



Σχήμα 2.1 Κοινωνικοποιημένη Αρχιτεκτονική Κεντρικής Διαχείρισης ενός δικτύου *Botnet*.

2.1 Ιστορικό

Η ιστορία των *botnet* ξεκινάει σχετικά πρόσφατα, το 1999, όπου έκανε την εμφάνιση του το πρώτο *IRC bot*, το Pretty Park worm [5]. Από τότε μέχρι σήμερα έχουν εξελιχθεί, αλλάζοντας τον τρόπο επικοινωνίας τους, τον τρόπο απόκρυψης τους και τις δυνατότητες που παρέχουν στον boxmaster σχετικά με την εκτέλεση εντολών. Στον παρακάτω πίνακα φαίνεται εξέλιξη των *botnet*.

Bot	Date	Implementation language	Protocol	Propagation mechanisms	Description
eggdrop	19.12.93	C	*IRC*	Active download	First non-malicious *IRC bot*
Pretty Park	19.05.99	Delphi	*IRC*	Send Email	First malicious *Bot* using *IRC* as *C&C* protocol With worm character
Subseven 2.1	1999	Delphi	*IRC*	Send Email	First *Bot* with Trojan character
GT*Bot*	2000	M*IRC* Script	*IRC*	Binding to M*IRC*	*IRC Bot* based on m*IRC* executables and scripts
SD*Bot*	20.02.02	C	*IRC*	Free Download	First stand alone *IRC Bot* code base
Slapper	20.09.02	C	P2P	Remote Vulnerability Scan	First worm with P2P communications protocol
Ago*Bot*	20.10.02	C++	*IRC*	Remote Vulnerability Scan	Incredibly robust, flexible, and modular design
rx*Bot*	2004	C	*IRC*	Remote Vulnerability Scan	Descendant of SD*Bot*, most wildly distributed *IRC Bot* code base
phat*Bot*	2004	C++	WASTE	Remote Vulnerability Scan	First Peer-to-Peer *Bot* based on WASTE
Bobax	20.05.04	VC++	*HTTP*	Send Email/ Remote Vulnerability Scan	*Bot* using *HTTP* based command and control mechanism
Click*Bot*.A	20.05.06	PHP	*HTTP*	Binding to other malware	*Bot* for click fraud
Nuwar or Storm worm	2007	VC++	P2P	Remote Vulnerability Scan	Distributional P2P structure Based on eMule protocol
Zunker	20.04.07	PHP/CGI	*HTTP*	P2P file sharing	communication by *HTTP*, P2P propagation mechanism
Mayday	20.01.08	N/A	*HTTP*/ Icmp	Send Email	communication by *HTTP*/Icmp, P2P structure
Waledac5	12.2008	N/A	P2P over *HTTP*	Remote Vulnerability Scan	Peer-to-Peer over *HTTP* protocol

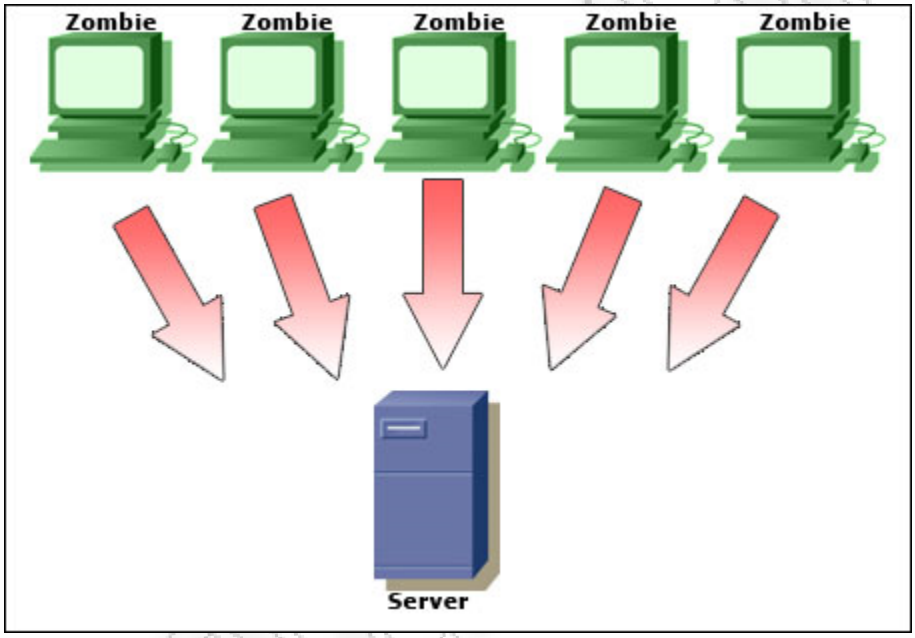
Πίνακας 1. Ιστορική Εξέλιξη των *Botnet*

2.2 Δυνατότητες των *Botnets*

2.2.1 Κατανεμημένες Επιθέσεις Άρνησης Υπηρεσιών (DDoS Attacks)

Η κατανεμημένη επίθεση άρνησης υπηρεσιών (*Distributed Denial of Service Attack*) αποτελεί ένα από τα κυριότερα είδη επιθέσεων που δύναται να πραγματοποιήσουν τα δίκτυα *botnet*. Ουσιαστικά πρόκειται για ένα μεγάλο αριθμό συνδέσεων των υπολογιστών-void προς κάποιον εξυπηρετητή ή δίκτυο που προσφέρει υπηρεσίες μέσα

από το διαδίκτυο. Αυτό έχει ως αποτέλεσμα την εξάντληση των πόρων του συστήματος και την αδυναμία εξυπηρέτησης αιτήσεων από νόμιμους χρήστες. Οι μέθοδοι πραγματοποίησης μίας τέτοιας επίθεσης είναι αρκετές, πιο συχνά εμφανιζόμενες όμως, στην υλοποίηση των *bots*, είναι οι *TCP SYN flood* και η *UDP flood* επιθέσεις, όπου πραγματοποιείται αποστολή ενός μεγάλου αριθμού *TCP SYN* και *UDP* πακέτων, αντίστοιχα.



Σχήμα 2.2 Απεικόνιση τυπικής DDoS Attack

Στόχος τέτοιου είδους επιθέσεως είναι κυρίως ιστοσελίδες με μεγάλη επισκεψιμότητα, επιφέροντας στις εταιρίες που τις φιλοξενούν απώλεια κερδών. Το γεγονός αυτό κάνει τα δίκτυα *botnet* ένα εργαλείο οικονομικής ωφέλειας, τοποθετώντας τα στο παιχνίδι μεγάλων επιχειρηματικών δραστηριοτήτων. Εκτός βέβαια από επιθέσεις σε ιστοσελίδες τα δίκτυα *botnet* είναι δυνατόν να πραγματοποιήσουν *DDoS* επιθέσεις σε οποιαδήποτε υπηρεσία διατίθεται μέσα από το διαδίκτυο. Σε αυτές συγκαταλέγονται μέχρι και κρατικές υπηρεσίες δίνοντας ακόμα περισσότερη αξία στα δίκτυα *botnet*, προάγοντας τα σε εργαλείο διακρατικών μαχών.

2.2.2 Αποστολή Ανεπιθύμητης Ηλεκτρονικής Αλληλογραφίας (Spamming)

Κάνοντας χρήση του πρωτοκόλλου *SMTP* τα *bot* εκτελούν εντολές μαζικής αποστολής ανεπιθύμητων ηλεκτρονικών μηνυμάτων, γνωστά και ως *Spam*. Έχοντας λίστες ηλεκτρονικών διευθύνσεων στη διάθεση τους, κάθε *bot* αποστέλλει ένα πλήθος από ηλεκτρονικά μηνύματα, τα οποία στο σύνολο τους αποτελούν ένα αρκετά μεγάλο όγκο δεδομένων. Τα *Spam* περιλαμβάνουν διαφημιστικά μηνύματα από οποιονδήποτε θελήσει να εκμεταλλευτεί, έναντι πληρωμής τη συγκεκριμένη λειτουργία-υπηρεσία των δικτύων *botnet*. Το γεγονός αυτό αποτελεί μία ακόμα αιτία που τα κάνει να αποτελούν σημείο ενδιαφέροντος στην οικονομία των επιχειρήσεων. Τα ανεπιθύμητα μηνύματα μπορεί επίσης να περιλαμβάνουν περιεχόμενο με τέτοιο τρόπο ώστε ο παραλήπτης να νομίζει ότι προέρχονται από κάποιο νόμιμο αποστολέα, ζητώντας του να επισκεφτεί κάποια ιστοσελίδα και να συμπληρώσει τα απαραίτητα στοιχεία. Η ιστοσελίδα αυτή είναι κατασκευασμένη με τέτοιο τρόπο, ώστε να φαίνεται αληθινή, πείθοντας τον επισκέπτη να συμπληρώσει τα στοιχεία που του ζητάει, στη συνέχεια όμως το μόνο που κάνει είναι να υποκλέπτει. Η επίθεση αυτή ονομάζεται ηλεκτρονικό ψάρεμα (*phishing*).

2.2.3 Κλοπή Ιδιωτικών Δεδομένων (Identity Theft)

Με τις λειτουργίες καταγραφής των πληκτρολογήσεων (*keylogging*) και ανίχνευσης κίνησης πακέτων δικτύου (*packet sniffing*) υπάρχει δυνατότητα, σε κάθε *bot*, υποκλοπής προσωπικών ιδιωτικών δεδομένων. Τα δεδομένα αυτά περιλαμβάνουν συνθηματικά (*passwords*) λογαριασμών, όπως ηλεκτρονικού ταχυδρομείου, αποστολής άμεσων μηνυμάτων (*ICQ*,*MSN*,*Yahoo*,*Skype*), τραπέζης, αριθμούς πιστωτικών καρτών και, ανάλογα με το σύστημα που είναι εγκατεστημένο το *bot*, δεδομένα μεγάλης αξίας και εμπιστευτικότητας όπως δεδομένα κρίσιμων υποδομών κρατών. Η δυνατότητα αυτή που έχουν τα δίκτυα *botnet* αποτελεί και τη μεγαλύτερη απειλή, αφού μεγάλοι οργανισμοί, επιχειρήσεις ακόμα και κρατικοί φορείς απειλούνται με αποκάλυψη κρίσιμων ιδιωτικών δεδομένων.

2.2.4 Απάτη Κλίκ (Click Fraud)

Εκμεταλλευόμενα τον μεγάλο αριθμό υπολογιστών με διαφορετική *IP* διεύθυνση, τα δίκτυα *botnet* συμμετέχουν στην απάτη κλικ (*click fraud*), όπου επισκέπτονται ιστοσελίδες και κάνουν κλικ πάνω σε διαφημιστικές ετικέτες (*banners*), προς όφελος των διαφημιστών που τα φιλοξενούν και οι οποίοι πληρώνονται ανάλογα με τον αριθμό των επισκέψεων, που πραγματοποιούνται μέσα από το *banner*. Η κίνηση της επισκεψιμότητας παρουσιάζεται να είναι νόμιμη, αφού προέρχεται από υπολογιστές με *IP* διευθύνσεις παγκόσμιας εμβέλειας.

2.2.5 Παράνομο Λογισμικό (WAREZ)

Τα δίκτυα *botnet* μπορούν επίσης να χρησιμοποιηθούν για τη διακίνηση παράνομου λογισμικού, είτε υποκλέπτοντας το από τον εκάστοτε υπολογιστή όπου είναι εγκατεστημένο το *bot*, είτε αποθηκεύοντας το σε αυτόν και δίνοντας τη δυνατότητα, στη συνέχεια, σε άλλους να το κατεβάσουν. Το ίδιο μπορεί να συμβεί και για τη διακίνηση οποιουδήποτε παράνομου περιεχομένου, χωρίς να ενοχοποιείται ο κάτοχος του, αλλά ο χρήστης του μηχανήματος που έχει μολυνθεί.

2.2.6 Φιλοξενία Παράνομων Ιστοσελίδων

Η φιλοξενία ιστοσελίδων από τα *bot*s επιτρέπει την ευκολότερη διακίνηση αρχείων παράνομου περιεχομένου, όπως πειρατικό λογισμικό και παράνομο φωτογραφικό υλικό. Επίσης μπορεί να φιλοξενήσει ιστοσελίδες ηλεκτρονικού ψαρέματος, όπως αναφέρθηκε παραπάνω, ιστοσελίδες για την αναφορά των *bot*s προς του *bot masters* και ιστοσελίδες για τον έλεγχο των *bot*s από τους *bot masters*.

2.2.7 Διάδοση Δικτύων *Botnet*

Για τη διάδοση τους τα δίκτυα *botnet* πραγματοποιούν επιθέσεις, σαρώνοντας ένα σύνολο από *IP* διευθύνσεις για γνωστές *TCP* και *UDP* υπηρεσίες , και κάνοντας χρήση κώδικα εκμετάλλευσης ευπαθειών (*exploit code*) αποκτούν τον έλεγχο του μηχανήματος. Στη συνέχεια μεταφορτώνουν το λογισμικό που συνιστά το *bot* και το μηχάνημα πλέον

αποτελεί και αυτό μέλος του *botnet*. Οι επιθέσεις που σχετίζονται με τη διάδοση του δικτύου μπορούν συνεχώς να εξελίσσονται, ενημερώνοντας τα *bot*s με νέα *exploit* και νέες τεχνικές.

2.3 Αρχιτεκτονικές

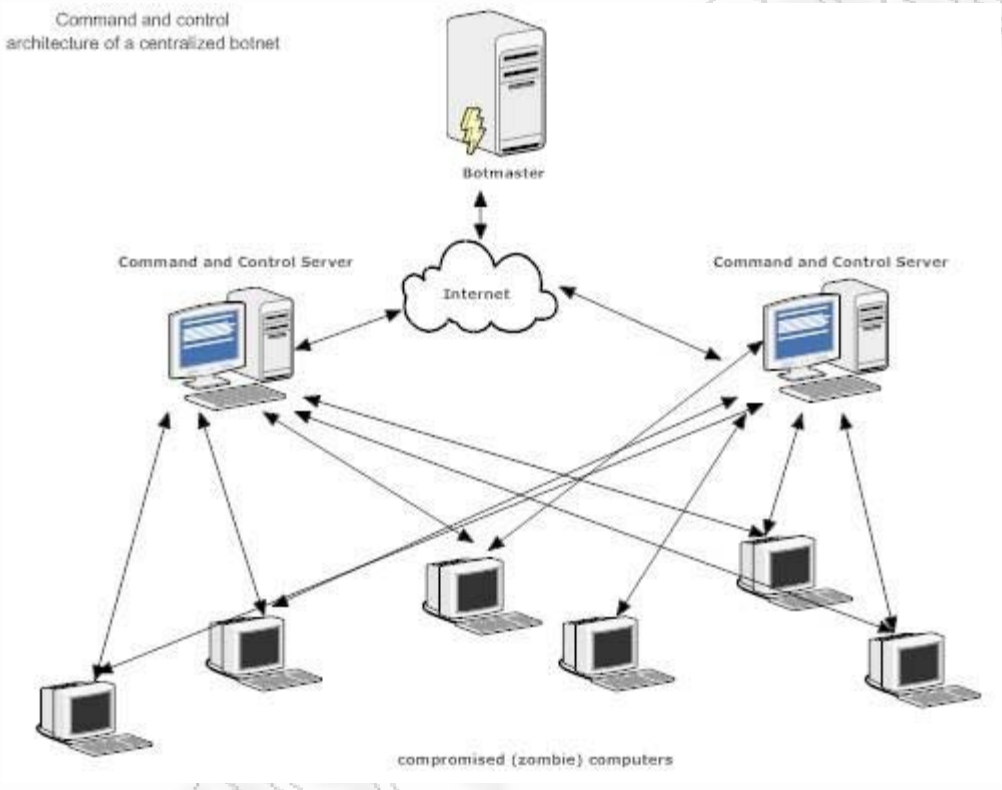
Όπως αναφέρθηκε παραπάνω βασική ιδιότητα ενός δικτύου *botnet* είναι η επικοινωνία του *bot* master με τα *bot*. Για το σκοπό αυτό χρησιμοποιείται ένα κανάλι επικοινωνίας, το *Command and Control Channel*, μέσα από το οποίο τα *bot* λαμβάνουν εντολές και επιστρέφουν αποτελέσματα. Ανάλογα με την αρχιτεκτονική του *Command and Control* καναλιού, τα *botnet* κατηγοριοποιούνται σε τρία διαφορετικά είδη αρχιτεκτονικής: την κεντρικοποιημένη , την αποκεντρικοποιημένη και την υβριδική .

2.3.1 Κεντρικοποιημένη

Σε αυτή την κατηγορία ανήκουν δίκτυα botnet που έχουν ένα κεντρικό σημείο ελέγχου. Όλα τα *bot*s λαμβάνουν εντολές από ένα *Command and Control Server* και συνδέονται πάλι πίσω για να δώσουν τις αναφορές των αποτελεσμάτων. Παράδειγμα τέτοιων δικτύων είναι τα *IRC* και τα *HTTP botnet*. Στα *IRC botnet*, *bot master* και *bot* συνδέονται σε κάποιο *IRC* κανάλι, επικοινωνώντας βάση του *Internet Relay Chat* πρωτοκόλλου. Τα *IRC botnet* είναι τα πρώτα που εμφανίστηκαν, λόγω του ότι είναι εύκολα στην υλοποίηση, αφού το *IRC* πρωτόκολλο είναι δημόσια ανοιχτό, έχουν εύκολο χειρισμό και γρήγορη ανταπόκριση των *bot*. Επίσης είναι δύσκολο να εντοπιστεί ο *botmaster*. Μειονεκτούν όμως στο γεγονός ότι είναι εύκολα ανιχνεύσιμα, ενώ κλείνοντας το κεντρικό *IRC* κανάλι επικοινωνίας σταματάει η λειτουργία όλου του δικτύου *botnet*. Στα *HTTP botnets* γίνεται χρήση του *Hyper Transfer Protocol* , αποκρύπτοντας εντολές και αναφορές μέσα σε *HTTP request - reply* πακέτα, κάνοντας την κίνηση να φαίνεται νόμιμη.

Υπάρχουν δύο τεχνικές επικοινωνίας του *botnmaster* με τα *bot*s, η *Push* και η *Pull*. Στην πρώτη τα *bot*s συνδέονται στον *C&C server*, για παράδειγμα έναν *IRC server* περιμένοντας από τον *botmaster* την ανάθεση αποστολών, ενώ στη δεύτερη αποθηκεύει

τις εντολές στον *C&C server*, συνήθως *HTTP*, περιμένοντας από τα *bot*s να συνδεθούν, για την ανάγνωση τους. Στην πρώτη περίπτωση η επικοινωνία είναι άμεση κατά την ανάθεση αποστολών, ενώ στη δεύτερη παρεμβάλλεται μία μικρή καθυστέρηση.

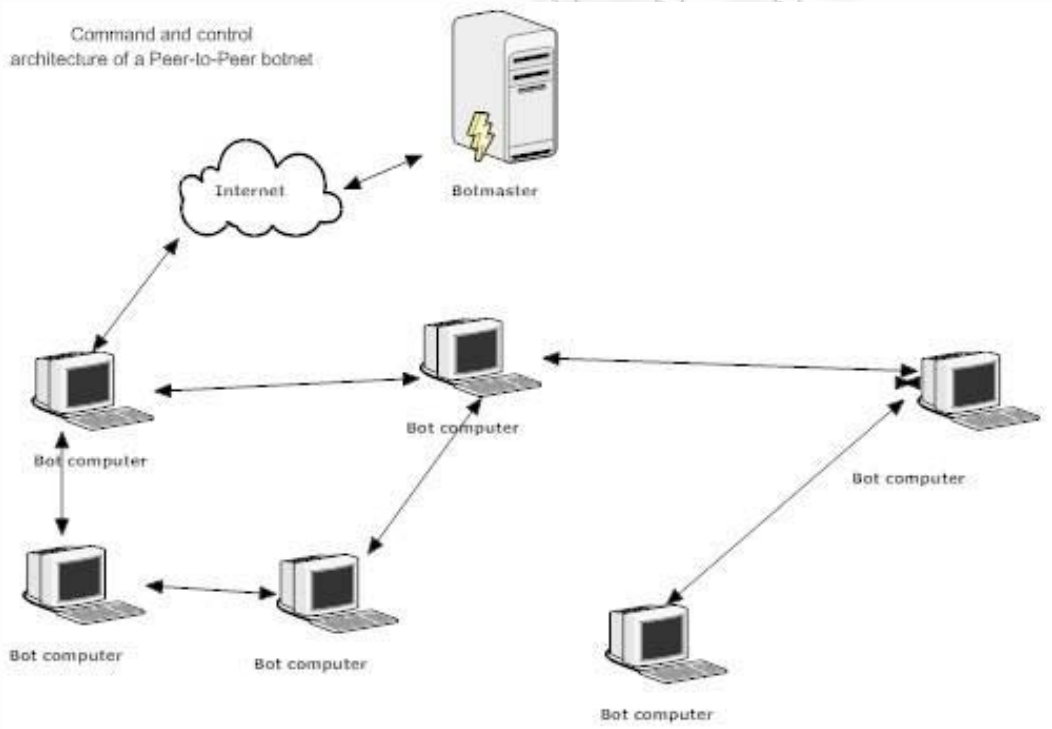


Σχήμα 2.3 Κανονικοποιημένη Αρχιτεκτονική Command & Control

2.3.2 Αποκεντρικοποιημένη – Κατανεμημένη

Στην αποκεντρικοποιημένη αρχιτεκτονική ανήκουν δίκτυα *botnet* που έχουν παραπάνω από ένα σημεία *C&C*. Η εμφάνιση τους έγινε σχετικά πρόσφατα, κάνοντας χρήση του πρωτοκόλλου *Peer to peer*. Η επικοινωνία του *botmaster* με τα *bot* γίνεται μέσα από ένα *peer to peer* δίκτυο στο οποίο συμμετέχουν όλα τα *bot* και ο *botmaster*, κάνοντας την εύρεση του καναλιού αρκετά δυσκολότερη, αφού εμφανίζεται να είναι νόμιμη *peer to peer* κίνηση. Ουσιαστικά τα *bot*s συστήνουν ένα «σύννεφο» από *peer to peer client*s και ο *botmaster* «σπρώχνει» εντολές μέσα σε αυτό. Στη συνέχεια τα *bot* επικοινωνούν μεταξύ τους για τη διάδοση των εντολών. Στην αρχιτεκτονική αυτή δεν

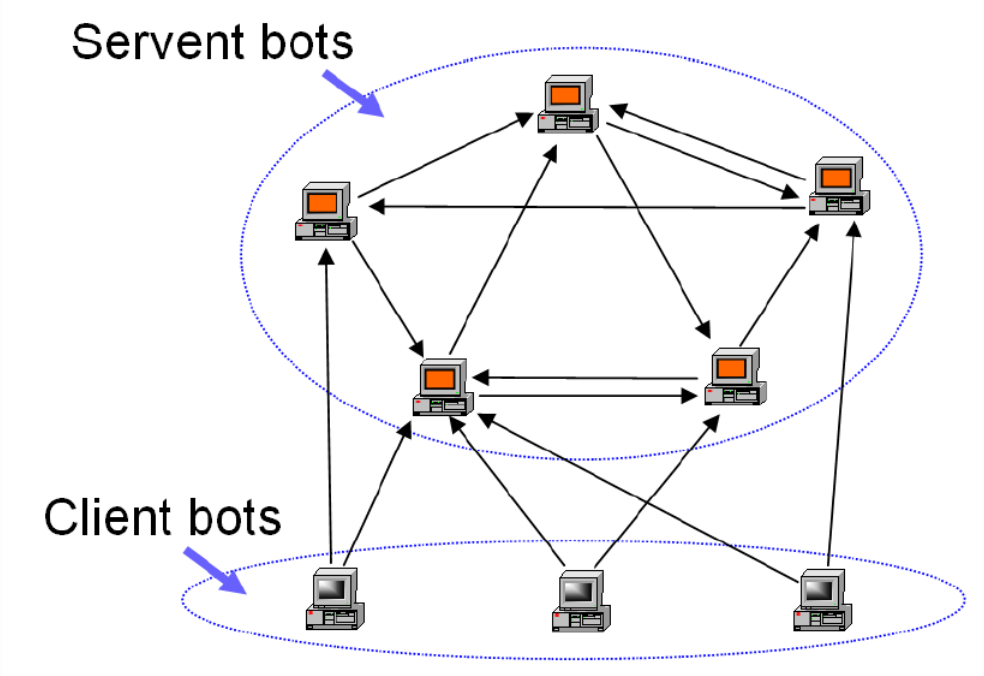
υπάρχει μοναδικό σημείο ελέγχου και ο τερματισμός λειτουργίας του δικτύου είναι αρκετά δύσκολος, αφού αρκεί η πρόσβαση σε ένα και μόνο *bot* του δικτύου, για τη μετάδοση εντολών. Η εισαγωγή νέων κόμβων μέσα στο δίκτυο πραγματοποιείται κάνοντας αναζήτηση για ήδη υπάρχοντες. Εφόσον ανακαλυφθεί έστω και ένας, ο νέος κόμβος ανακτά όλες τις διαθέσιμες πληροφορίες για το δίκτυο και στη συνέχεια συμπεριφέρεται και θεωρείται και αυτός ως μέλος του δικτύου. Παράδειγμα *Peer to peer* πρωτοκόλλου που χρησιμοποιείται από δίκτυα *botnet* είναι το *eDonkey*/*Overnet* πρωτόκολλο.



Σχήμα 2.4 Αποκεντρικοποιημένη Αρχιτεκτονική *C&C*

2.3.3 Υβριδική

Η υβριδική αρχιτεκτονική δανείζεται χαρακτηριστικά και από τις δύο που προαναφέρθηκαν. Ένα μοντέλο της αρχιτεκτονικής έχει προταθεί στο [7] και είναι περισσότερο θεωρητικό. Υπάρχει ένα *peer to peer* δίκτυο από *Servent bot* με δημόσιες *IP* διευθύνσεις, μέσα στο οποίο προωθούνται οι εντολές από τον *botmaster* και ένα *peer to peer* δίκτυο από *client bot*, με ιδιωτικές ή *NAT IP* διευθύνσεις, το οποίο επικοινωνεί με το προηγούμενο για την ανάγνωση τους. Η επικοινωνία των *client*s με τα servent *bot* πραγματοποιείται κάνοντας χρήση μίας στατικής λίστας, που είναι γνωστή από πριν. Αν ανακαλυφθεί κάποιο από τα *servent bot* η λειτουργία του δίκτυο *botnet* συνεχίζεται, μέχρι να αποκαλυφθεί και το τελευταίο.



Σχήμα 2.5 Υβριδικό *C&C* Channel

2.3.4 Ταξινόμηση

Η ταξινόμηση των δικτύων *botnet* μπορεί να γίνει με βάση πολλά από τα χαρακτηριστικά τους, όπως την αρχιτεκτονική του *C&C* καναλιού, το πρωτόκολλο επικοινωνίας, τον τύπο των επιθέσεων, τον τρόπο εντοπισμού του *C&C* δένει, τις ενέργειες που παρατηρούνται και τις τεχνικές απόκρυψης που χρησιμοποιούν. Στον

παρακάτω πίνακα παρουσιάζονται συνοπτικά τα είδη των *botnet*, με βάση τα χαρακτηριστικά που προαναφέρθηκαν [8].

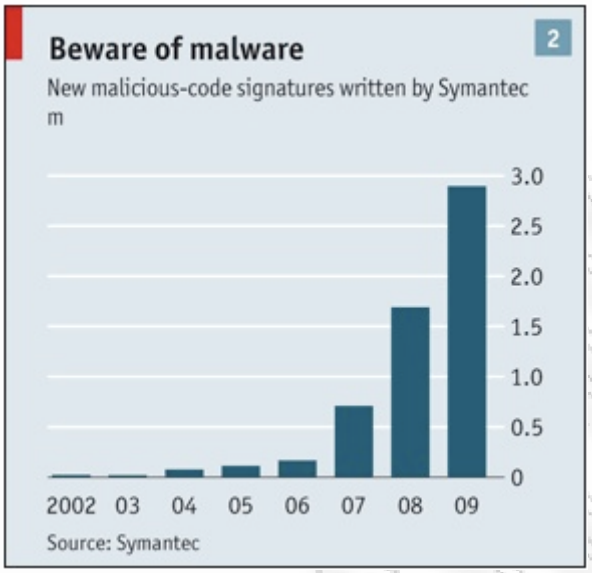
Category	Examples
Attacking Behavior	DDoS , Scan , Remote *Exploit*s , Junk Emails (phishing and virus attachments) , Phishing websites, Spyware, Identity theft etc
***C&C* Models**	Centralized, Distributed, P2P, etc
Rally Mechanisms	Hard-coded IP, Dynamic DNS, Distributed DNS etc
Communications Protocols	*IRC*, *HTTP*, IM, P2P, etc
Observable *Botnet* Activities	DNS queries, Burst short packets, Abnormal system calls, etc
Evasion Techniques	*HTTP*/VoIP tunneling, IPv6 tunneling, P2P encrypted traffic, etc

Πίνακας 2. Ταξινόμηση Δικτύων *Botnet*

2.4 Στατιστική του Προβλήματος

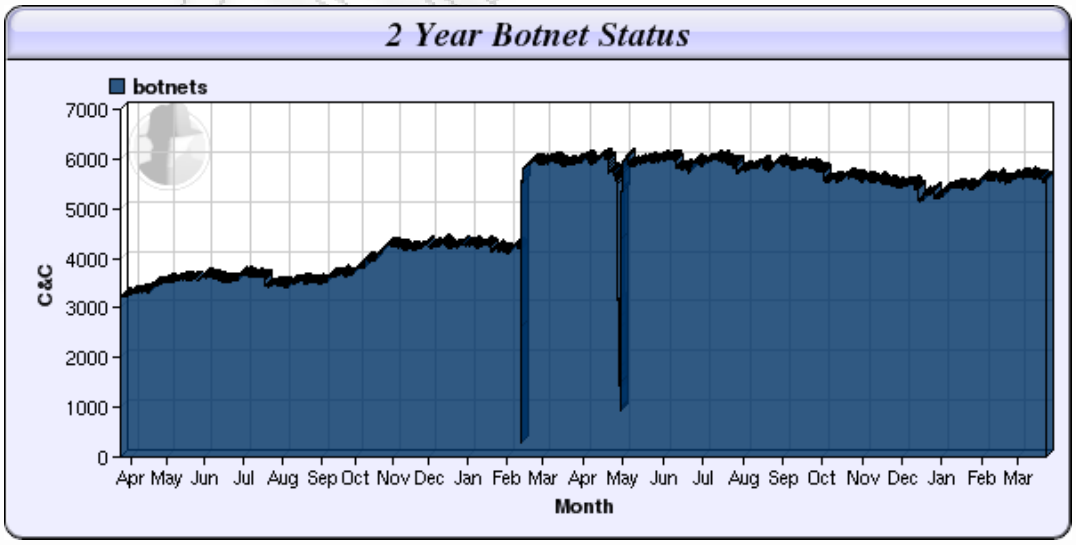
Η εξάπλωση των δικτύων *botnet* και των συνεπειών που επιφέρουν αποτελούν μία από τις μεγαλύτερες σύγχρονες απειλές του διαδικτύου. Τοποθετούνται μέσα στις κυριότερες πηγές παράνομου εισοδήματος μέσα από ηλεκτρονικό έγκλημα και είναι πανίσχυρα όπλα στα χέρια κυβερνο-εγκληματιών. Η ανάπτυξη τους είναι συνεχής, και η διαχείριση τους όλο και πιο απλή, κάνοντας τη χρήση τους αρκετά εύκολη, ακόμα και από απλούς χρήστες. Η πρόσβαση σε δίκτυα *botnet* και σε παράνομα αγαθά εξαρτάται πλέον από τα χρήματα που μπορεί κάποιος να διαθέσει, παρά τη γνώση που κατέχει, σε σημείο να κυκλοφορούν ακόμα και κατάλογοι τιμών.

Σύμφωνα με μετρήσεις της *Symantec* το 2010 δημιουργήθηκαν πάνω από 3 εκατομμύρια νέες υπογραφές κακόβουλου λογισμικού. Όπως φαίνεται και στο Σχήμα 2.6, πρόκειται για 350% αύξηση από το έτος 2007 μέχρι σήμερα, γεγονός που ενισχύει ακόμα περισσότερο το μέγεθος της απειλής.



Σχήμα 2.6 Υπογραφές κακόβουλου λογισμικού (Symantec)

Σύμφωνα και με τα στατιστικά στοιχεία, που παρουσιάζονται στην ιστοσελίδα του Shadow *Server*, τον τελευταίο χρόνο υπάρχει αύξηση του συνολικού αριθμού των *bots* . Το Σχήμα 2.7 παρουσιάζει τις μετρήσεις που έχουν γίνει σχετικά με την παραπάνω αύξηση, για διάστημα 2 χρόνων,. Αναλύοντας τα στοιχεία που παρουσιάζονται στο Σχήμα 2.7, βλέπουμε ότι ο αριθμός των ενεργών *botnet* παρουσιάζει και αυτός αυξητικό χαρακτήρα κατά το τελευταίο έτος και φτάνει τα 6,000 *botnet* περίπου. Μπορεί κανείς λοιπόν να φανταστεί ότι ο συνολικός αριθμός των υπονομευμένων υπολογιστών μπορεί να υπολογιστεί σε αρκετά εκατομμύρια



Σχήμα 2.7 Συνολικός Αριθμός *Botnet* (Shadow *Server*)

Κεφάλαιο 3. Τεχνικές Ανίχνευσης

Η ανίχνευση των *botnets* είναι μια σχετικά νέα περιοχή έρευνας πολύ μεγάλης όμως σημασίας για την αντιμετώπιση των *botnets*. Δεδομένου ότι η προσβολή και μετατροπή των υπολογιστών σε *bot*s είναι σε μεγάλο βαθμό αναπόφευκτη, η έρευνα στα *botnets* επικεντρώνεται κυρίως στην ανίχνευσή τους. Πρόσφατα μερικές δημοσιεύσεις έχουν προτείνει κάποιες προσεγγίσεις για την ανίχνευση των *botnets*, δεν έχουμε όμως συναντήσει κάποια εργασία που να συνοψίζει τις προσεγγίσεις αυτές.

Ξεκινώντας μπορούμε να διακρίνουμε δύο βασικές κατηγορίες: την ανίχνευση σε επίπεδο μηχανήματος (*host-based detection*) και την ανίχνευση σε επίπεδο δικτύου (*network-based detection*).

3.1. Ανίχνευση σε επίπεδο μηχανήματος (host-based)

3.1.1 Ανίχνευση σε επίπεδο υπολογιστή

Η ανίχνευση σε επίπεδο υπολογιστή αφορά τις μη κανονικές παρατηρούμενες συμπεριφορές σε κάποιο προσωπικό υπολογιστή ή και εξυπηρετητή. Τα *bot*s υπονομεύουν τους υπολογιστές και κρύβουν την παρουσία τους όπως ακριβώς κάνουν και οι παλιότεροι ιοί των υπολογιστών. Επομένως, μπορεί να επιδεικνύουν παρατηρήσιμες συμπεριφορές όπως και οι ιοί στους μολυσμένους υπολογιστές. Όταν εκτελούνται κάνουν μια σειρά από κλήσεις συστήματος/βιβλιοθηκών (*system*/*library calls*) οι οποίες μπορεί να αποσκοπούν σε ενέργειες όπως, οι αλλαγές στο μητρώο του υπολογιστή (*registry*), οι αλλαγές στο σύστημα αρχείων (*file system*), η δημιουργία δικτυακών συνδέσεων και η απενεργοποίηση των λογισμικών προστασίας (π.χ. *antivirus* ή *firewall*). Η σειρά των κλήσεων συστήματος/βιβλιοθηκών που γίνονται από τα *bot*s είναι συχνά διαφορετικές από αυτές που προκαλεί η φυσιολογική χρήση των προγραμμάτων και των εφαρμογών που περιέχει ο υπολογιστής. Κάποιες από αυτές τις συμπεριφορές είναι παρατηρήσιμες από ανθρώπους με λογικές γνώσεις στην ασφάλεια. Για παράδειγμα αν κάποιο λογισμικό προστασίας από ιούς αποτύχει να ενημερωθεί

(*update*), όπως θα έκανε φυσιολογικά, τότε είναι λογικό να υποψιαστούμε ότι ο υπολογιστής είναι μολυσμένος από κάποιο ιό ή κάποιο *bot* [9]. Παρακάτω αναφέρουμε κάποιους από τους τρόπους ανίχνευσης.

α) Εξέταση των αρχείων καταγραφής

Αυτός ο τρόπος ανίχνευσης περιλαμβάνει την εξέταση των αρχείων καταγραφής (log files) από το διαχειριστή του υπολογιστή ή εξυπηρετητή. Τα αρχεία αυτά καταγράφουν τη δραστηριότητα του συστήματος (συνδέσεις, αποσυνδέσεις χρηστών κτλ.), του λογισμικού προστασίας (π.χ. *anti-virus* και *anti-spyware logs*), των εγκατεστημένων εφαρμογών, καθώς και των υπηρεσιών (*services*) όταν πρόκειται για κάποιο εξυπηρετητή. Τα *bot*s συνήθως κάνουν κατάχρηση των υπηρεσιών που προσφέρουν οι εξυπηρετητές, πράγμα που μπορεί να φανεί από τα αρχεία καταγραφής, αφού κάθε υπηρεσία, όπως η *web*, *ftp*, *smtp*, *proxy* υπηρεσία, περιλαμβάνει τα δικά της αρχεία καταγραφής.

β) Συσχέτιση των αρχείων καταγραφής διαφορετικών υπολογιστών

Σύμφωνα με την τεχνική αυτή παρακολουθούνται οι κλήσεις διαδικασιών των διεπαφών προγραμματισμού εφαρμογών του λειτουργικού συστήματος (π.χ. *Windows API*), που γίνονται από τις εφαρμογές επικοινωνίας του υπολογιστή και καταγράφονται οι κλήσεις αυτές μαζί με τις παραμέτρους τους σε αρχεία καταγραφής. Τα αρχεία αυτά συγκρίνονται στη συνέχεια με τα αντίστοιχα σε κάποιους άλλους υπολογιστές του δικτύου και συσχετίζονται ως προς το μέγεθός τους. Η περίπτωση υψηλής συσχέτισης, δηλαδή ταυτόχρονης αύξησης του μεγέθους των αρχείων σε συγκεκριμένες χρονικές στιγμές μπορεί να αποκαλύψει την ύπαρξη κάποιου *botnet* [10].

γ) Συσχέτιση δραστηριότητας keylogging

Η τεχνική αυτή μπορεί να εφαρμοστεί σε ένα μεμονωμένο υπολογιστή. Η καταγραφή των πληκτρολογήσεων (keylogging) από τα *bot*s σχεδόν πάντα συνοδεύεται από αποθήκευση των πληκτρολογήσεων αυτών σε κάποιο αρχείο, ή την αποστολή τους

σε τακτά χρονικά διαστήματα σε κάποιον άλλο υπολογιστή στο διαδίκτυο. Συσχετίζονται λοιπόν, οι κλήσεις διαδικασιών του συστήματος που προκαλούνται από την πληκτρολόγηση με τις κλήσεις που προκαλούνται από τη μετέπειτα δημιουργία/ενημέρωση αρχείων ή το άνοιγμα εξερχόμενων δικτυακών συνδέσεων. Η υψηλή χρονική συσχέτιση μεταξύ των συνδυασμών κλήσεων που αναφέρθηκαν μπορεί να σημαίνει την ύπαρξη κάποιου *bot* στο σύστημά μας [11].

Παρόλα αυτά η ανίχνευση σε επίπεδο υπολογιστή μπορεί να γίνει μόνο αν εκδηλώνονται παρατηρήσιμες συμπεριφορές, πράγμα που δεν ισχύει πάντα και εξαρτάται από την εξυπνάδα του κακόβουλου λογισμικού. Αν για παράδειγμα το *bot* διαθέτει χαρακτηριστικά *rootkit* ή ακόμη χειρότερα αν εκτελείται στο επίπεδο του πυρήνα (*kernel-level*) του λειτουργικού συστήματος του υπολογιστή, τότε η ανίχνευσή του είναι ακόμη πιο δύσκολη αφού θα έχει τη δυνατότητα να αποκρύπτει εκτελούμενες διεργασίες, να τροποποιεί τα αρχεία του λειτουργικού συστήματος και τα αρχεία καταγραφής των προγραμμάτων προστασίας, ακόμη και να αποκρύπτει την εισερχόμενη και εξερχόμενη δικτυακή κίνηση στο μολυσμένο υπολογιστή.

3.1.2 Ανίχνευση σε επίπεδο δικτυακής συσκευής

Παρόμοια με την πρώτη τεχνική ανίχνευσης σε επίπεδο υπολογιστή, η εξέταση των αρχείων καταγραφής (*log files*) μπορεί να εφαρμοστεί και στις δικτυακές συσκευές για την ανίχνευση των *botnets*. Εδώ εξετάζονται τα αρχεία καταγραφής του τοίχου προστασίας (firewall) και του δρομολογητή (*router*) για εύρεση κίνησης '*Command and Control*' (*C&C traffic*). Θα πρέπει να παρακολουθούνται τα αρχεία καταγραφής του τοίχου προστασίας για συνδέσεις που δεν έγιναν δεκτές, αλλά και για αυτές που επιτράπηκαν σε συνήθη *C&C* κανάλια (π.χ. η *TCP* θύρα 6667 για εντοπισμό *IRC botnets*).

3.2 Ανίχνευση σε επίπεδο δικτύου (network-based)

Όπως αναφέρθηκε προηγουμένως, η ανίχνευση των *bot*s σε επίπεδο μηχανήματος (*host*) δεν είναι αποτελεσματική σε περίπτωση που τα *bot*s αποκρύπτουν τις παρατηρήσιμες συμπεριφορές. Αυτό που δεν μπορεί να κρυφτεί όμως είναι η δικτυακή κίνηση που δημιουργούν τα *bot*s κατά τη δραστηριότητα τους προς άλλους υπολογιστές του τοπικού ή εξωτερικού δικτύου. Στην τελευταία περίπτωση, η κίνηση αυτή σίγουρα θα περάσει από τη δικτυακή πύλη (*gateway*) του δικτύου. Οι *botmaster*s χρειάζεται να επικοινωνήσουν με τα *bot*s τους και να εκκινήσουν επιθέσεις. Έτσι δημιουργείται συγκεκριμένη παρατηρήσιμη δικτυακή κίνηση προς και από τα *bot*s η οποία μπορεί να ανιχνευτεί παρατηρώντας τις δικτυακές ροές (*network flows*) με τις τεχνικές που αναφέρουμε παρακάτω.

3.2.1 Ανίχνευση στα χαρακτηριστικά των δικτυακών ροών

Αυτή η τεχνική για την ανίχνευση των *botnets* εξετάζει τα χαρακτηριστικά των δικτυακών ροών, όπως η χωρητικότητά τους, η διάρκεια και η χρονομέτρηση (*timing*) των πακέτων για εντοπισμό ενδείξεων δραστηριότητας *C&C*. Είναι περισσότερο αποτελεσματική στην ανίχνευση *IRC botnets* τα οποία έχουν «σφιχτό» '*Command and Control*'. Για αυτό το λόγο άλλωστε, εκτιμάται ότι τα *IRC botnets* θα επιβιώσουν για αρκετό χρόνο ακόμη, σε αντίθεση με την τάση για μεταφορά της αρχιτεκτονικής του *C&C* σε άλλα πρωτόκολλα όπως το *HTTP* και το *P2P*.

Πιο συγκεκριμένα, τα χαρακτηριστικά των ροών περιλαμβάνουν τα πακέτα από ροή (*ppf*), το μέσο αριθμό *bytes* ανά πακέτο (*bpp*), το μέσο αριθμό *bytes* ανά δευτερόλεπτο (*bps*) και το μέσο αριθμό πακέτων ανά δευτερόλεπτο (*pps*). Για τις συναθροισμένες ροές τα χαρακτηριστικά περιλαμβάνουν τις ροές ανά διεύθυνση (*fpa*) και τις ροές ανά ώρα (*fph*).

Για την ανίχνευση γίνεται συσχέτιση (*correlation*) κοιτώντας τη σχέση ανάμεσα σε δύο ή περισσότερες ροές η οποία θα έδειχνε ότι οι ροές είναι μέρος του ίδιου *botnet*. Η ερώτηση για το αν μια ροή συσχετίζεται με κάποια άλλη έχει νόημα μόνο αν οι δύο ροές είναι ενεργές στον ίδιο χρόνο. Πειράματα από πραγματικά ίχνη κίνησης (*traces*) έχουν

δείξει ότι τα ακόλουθα χαρακτηριστικά έχουν υψηλή αξία στην διάκριση των *IRC* ροών από τις μη-*IRC* ροές: διάρκεια, ρόλος (αν ο πελάτης ή ο εξυπηρετητής ξεκίνησε τη ροή), μέσος αριθμός *bytes* ανά πακέτο, μέσος αριθμός *bits* ανά δευτερόλεπτο και μέσος αριθμός πακέτων ανά δευτερόλεπτο. Ανάμεσα σ' αυτές, τα *bytes* ανά πακέτο παρείχαν τη μεγαλύτερη ισχύ για τη διάκριση στις ροές [12].

3.2.2 Ανίχνευση στους headers των πακέτων

α) Ανίχνευση κίνησης DNS προς τα *C&C* domains

Πρόκειται για ένα μηχανισμό ανίχνευσης που παρακολουθεί την κίνηση *DNS* (*Domain Name System*) χρησιμοποιώντας την πληροφορία που παρέχουν τα *IP headers*. Πολλά *botnets* χρησιμοποιούν δυναμικά ονόματα *DNS domains* για να εντοπίσουν τους *C&C server*s. Επομένως, μη συνηθισμένα ερωτήματα (*queries*) *DNS* καθώς και οι διαφορές τους από τα συνηθισμένα ερωτήματα μπορεί να είναι ενδείξεις για την ύπαρξη κάποιου *botnet*. Σε κάποιες περιπτώσεις οι υπολογιστές βρίσκονται να ρωτούν για ακατάλληλα domain ονόματα (π.χ. *cheese.dns4biz.org*), πράγμα που μπορεί να δείχνει με μεγάλη πιθανότητα ότι οι υπολογιστές έχουν υπονομευθεί. Το επόμενο λογικό βήμα θα είναι προφανώς να καταγράψουμε και να μπλοκάρουμε αυτές τις *IP* που αντιστοιχούν στους *C&C* εξυπηρετητές. Επίσης αν ανακαλυφθεί ότι οι *IP* διευθύνσεις που αντιστοιχούν σε ένα συγκεκριμένο *domain* αλλάζουν συνεχώς, τότε θα έχουμε ακόμη ισχυρότερες ενδείξεις για την ύπαρξη *botnet*.

Ένα άλλο μετρικό για την ανίχνευση είναι το κατά πόσο τα ερωτήματα *DNS* γίνονται ταυτόχρονα από περισσότερους από ένα υπολογιστές του δικτύου και αφορούν το ίδιο *domain*, γεγονός που θα φανέρωνε κάποιο ύποπτο συγχρονισμό ανάμεσα στους υπολογιστές.

Το πλεονέκτημα αυτής της τεχνικής είναι ότι μπορεί να γίνει ανίχνευση ενός *botnet* ακόμη και αν η επικοινωνία με τον *C&C server* είναι κρυπτογραφημένη. Μειονεκτήματα αποτελούν: το γεγονός ότι η ανίχνευση δεν είναι αποτελεσματική για μικρά δίκτυα, τα *bot*s μπορεί να συγχύζουν το σύστημα ανίχνευσης με ψεύτικα ερωτήματα *DNS* και ότι το

σύστημα εξαρτάται από συγκεκριμένα κατώφλια (*threshold*) χρόνου μέσα στα οποία καταγράφονται τα ερωτήματα *DNS* [13].

β) Ανίχνευση ασυνήθιστης εξερχόμενης SMTP κίνησης

Σύμφωνα με την τεχνική αυτή παρακολουθείται η εξερχόμενη κίνηση του πρωτοκόλλου *SMTP* (*Simple Mail Transfer Protocol*) η οποία μπορεί να είναι υπερβολική ή να δημιουργείται από υπολογιστές που υποτίθεται ότι δεν θα έπρεπε να δημιουργούν τέτοιο είδος κίνησης. Η κίνηση αυτή διακρίνεται από τις δικτυακές συνδέσεις που χρησιμοποιούν την πόρτα 25 του πρωτοκόλλου *TCP*, αλλά και από τα ιδιαίτερα χαρακτηριστικά του πρωτοκόλλου *SMTP* (π.χ. ανίχνευση εντολών *HELO*, *EHLO*, *STARTTLS* κτλ.). Στη περίπτωση των '*spambots'* όπως το *Storm bot*, των *bot*s δηλαδή που χρησιμοποιούνται για τη μαζική αποστολή ανεπιθύμητης αλληλογραφίας (*spam*), η διάδοση της επίθεσης μπορεί εύκολα να ανιχνευθεί από τη γρήγορα αναπαραγόμενη επικοινωνία τοπικών υπολογιστών, οι οποίοι δεν είναι *SMTP* εξυπηρετητές αλλά ξαφνικά κάνουν *Smtp Mail* συναλλαγές (*transactions*) με ένα ευρύ φάσμα εξωτερικών *SMTP* εξυπηρετητών [14].

Για την αντιμετώπιση των spam*bot*s ένα αποτελεσματικό μέτρο που λαμβάνουν πλέον οι πάροχοι δικτυακής πρόσβασης και οι διαχε*ι*ριστές των Mail εξυπηρετητών είναι να μην επιτρέπουν τις εισερχόμενες συνδέσεις για αποστολή email από υπολογιστές με δυναμικές διευθύνσεις IP (εκτός της περίπτωσης που παρέχονται διαπιστευτήρια για την ταυτότητα του χρήστη), αφού σχεδόν πάντα οι Mail εξυπηρετητές έχουν στατικές διευθύνσεις. Αυτό το μέτρο βέβαια δεν έχει εφαρμογή στην περίπτωση που κάποιο *bot* έχει υπονομεύσει ένα νόμιμο Mail εξυπηρετητή.

γ) Ανίχνευση υψηλών ρυθμών συνδέσεων TCP ή UDP

Αυτή η τεχνική ανίχνευσης αφορά κυρίως τα *P2P bot*s, τα οποία λόγω της φύσης της λειτουργίας τους δημιουργούν υψηλούς ρυθμούς συνδέσεων προς εξωτερικές διευθύνσεις κάνοντας χρήση των πρωτοκόλλων *TCP* ή και *UDP*. Στην περίπτωση ενός δικτύου που δεν χρησιμοποιεί *peer-to-peer* εφαρμογές, τα *P2P bot*s θα προκαλούσαν τη

δημιουργία εκρηκτικών συνδέσεων προς πολλαπλές εξωτερικές πόρτες και διευθύνσεις *IP*, άρα και υψηλές ενδείξεις για την ύπαρξή τους.

Επίσης αυτή η τεχνική θα μπορούσε να χρησιμοποιηθεί για την ανίχνευση αποστολής όχι μόνο ανεπιθύμητων μηνυμάτων (*spam*) σε *Mail* εξυπηρετητές ως μηνύματα *email*, αλλά και ανεπιθύμητων μηνυμάτων ως δημοσιεύσεις σε ιστολόγια (*blogs*), φόρουμ (*forum*), ή *wiki*.

3.2.3 Ανίχνευση στο payload των πακέτων

Αυτή η τεχνική είναι αρκετά διαδεδομένη και εξετάζει το φορτίο (*payload*) των δεδομένων των πακέτων (*payload inspection*). Πιο συγκεκριμένα αναγνωρίζει κίνηση με ομοιότητες στο *payload*, ή ειδικότερα *payloads* για τα οποία η τιμή edit distance είναι μικρή. Διαισθητικά, η κίνηση '*Command and Control*' ανάμεσα στο *bomaster* και τα *bot*s θα έχει συγκεκριμένη δομή, και επομένως θα περιμέναμε να υπάρχει μικρή τιμή '*edit distance*' ανάμεσά τους.

Επίσης, η τεχνική αυτή μπορεί να χρησιμοποιηθεί για την ανίχνευση συγκεκριμένων χαρακτηριστικών (*patterns*) μέσα στα πακέτα του δικτύου. Χρησιμοποιείται για παράδειγμα για τον εντοπισμό *IRC bot*s με συγκεκριμένα ψευδώνυμα (*nicknames*). Συχνά τα *IRC bot*s έχουν ψευδώνυμα με κοινά χαρακτηριστικά, όπως μεγάλοι τυχαίοι αριθμοί ή κωδικοί χωρών [15]. Βέβαια, αυτή η προσέγγιση μπορεί να ανιχνεύσει *bot*s για τα οποία ο τύπος (*format*) του ψευδωνύμου είναι γνωστός.

α) Ανίχνευση με συνάθροιση κίνησης παρόμοιου περιεχομένου

Η τεχνική συνάθροισης κίνησης (*traffic aggregation*) μπορεί να χρησιμοποιηθεί για να συναθροίσουμε ροές (*flows*) που βασίζονται σε παρόμοιο περιεχόμενο. Σε κάποιες περιπτώσεις ισχύει ότι η κακόβουλη κίνηση είναι σημαντικά πιο συχνή και διεσπαρμένη σε σχέση με την υπόλοιπη κίνηση, έτσι το ίδιο περιεχόμενο θα επαναλαμβάνεται σε ένα μεγάλο αριθμό διαφορετικών πακέτων ή ροών [16].

β) Ανίχνευση εντολών *C&C* στα πακέτα δικτύου

Εκμεταλλευόμενοι το γεγονός ότι ο *botmaster* χρειάζεται ένα κώδικα επικοινωνίας που θα είναι κοινός για όλα τα *bot*s, μπορούμε να σαρώνουμε τα πακέτα του δικτύου για συγκεκριμένες εντολές που αποτελούν το 'Command and Control' [11]. Με αυτό τον τρόπο εντολές όπως οι 'scan.start', '*HTTP.download'*, '*spam.setlist'* που χρησιμοποιεί το *Agobot* μπορούν εύκολα να εντοπιστούν. Παρόλα αυτά η κρυπτογράφηση του *payload* μπορεί να είναι ανασταλτικός παράγοντας για την εφαρμογή αυτής της τεχνικής ανίχνευσης.

γ) Ανίχνευση κώδικα εκμετάλλευσης ευπαθειών

Η ανίχνευση κώδικα εκμετάλλευσης (*exploit code*) ευπαθειών (*vulnerabilities*) λογισμικού αποτελεί μια σίγουρη τεχνική ανίχνευσης κακόβουλης επίθεσης, εφόσον τα δεδομένα των πακέτων θα τηρούν συγκεκριμένα πρότυπα (*patterns*). Η επίθεση μπορεί να γίνεται από μέσα προς τα έξω από κάποιο *bot* που υπάρχει στο δίκτυό μας, ή και από έξω προς κάποιον υπολογιστή του εσωτερικού δικτύου μας.

Ο κακόβουλος αυτός κώδικας μπορεί να εμφανίζεται και εκτός του payload των πακέτων, όμως το να εντοπίζεται μέσα στο payload είναι πιο πιθανό.

Παρόλα αυτά, αυτή η τεχνική μπορεί μόνο να εντοπίσει επιθέσεις που χρησιμοποιούν γνωστές ευπάθειες των λογισμικών και δεν μπορεί να καλύψει καινούργιες ευπάθειες εκ των προτέρων, παρά μόνο μετά τη γνωστοποίησή τους.

3.2.4 Ανίχνευση βασισμένη σε υπογραφές (signature-based)

Η ανίχνευση των *bot*s βασιζόμενη σε υπογραφές (*signatures*) ή αλλιώς ανίχνευση λανθασμένης εφαρμογής (*misuse detection*), χρησιμοποιεί πληροφορίες από γνωστές πολιτικές ασφάλειας, από γνωστές ευπάθειες των λογισμικών και από γνωστές επιθέσεις. Συγκεκριμένα, συγκρίνεται η δικτυακή δραστηριότητα ή τα καταγεγραμμένα δεδομένα ενός συστήματος με γνωστές υπογραφές επιθέσεων ή άλλες ενδείξεις λανθασμένων εφαρμογών επικοινωνίας, έτσι ώστε να αναγνωριστούν πρότυπα (*patterns*) που παραπέμπουν στην ύπαρξη *bot*s. Τα περισσότερα συστήματα ανίχνευσης εισβολών

(*Intrusion Detection Systems*) χρησιμοποιούν αυτή την τεχνική ανίχνευσης, ενώ οι ερευνητές προσπαθούν να βρουν έξυπνους τρόπους αντιστοίχησης των δυναμικά εξελισσόμενων μορφών επιθέσεων σε ήδη γνωστές επιθέσεις. Ένα από τα πιο γνωστά συστήματα ανίχνευσης εισβολών (*IDS*) είναι το *Snort* που θα δούμε παρακάτω.

Εδώ θα πρέπει να αναφέρουμε ότι η ανίχνευση των *botnets* που βασίζεται στις υπογραφές μπορεί να χρησιμοποιεί τις προηγούμενες τεχνικές που αναφέραμε για την ανίχνευση στους *headers* αλλά και το *payload* των πακέτων, αφού μια υπογραφή μπορεί και έχει στοιχεία για το *header* και για το *payload* ενός πακέτου.

Η μέθοδος αυτή συνήθως εφαρμόζεται (για λόγους απόδοσης) για συγκεκριμένες εφαρμογές-πρωτόκολλα όπως είναι το *HTTP* ή το *IRC* που αποτελούν και τα πιο πιθανά κανάλια επικοινωνίας του *botmaster* με τα *bot*s. Το πρόβλημα όμως που ανακύπτει είναι ότι μπορεί το *C&C* να υλοποιείται σε κάποια άλλη από τις γνωστές πόρτες των πρωτοκόλλων που αναφέραμε. Για παράδειγμα μπορεί το κανάλι *HTTP* επικοινωνίας να μην υλοποιείται στην πόρτα 80 αλλά στην 8080 ή στην 8081 ή στην 8000 και ούτω καθ' εξής. Ομοίως μπορεί το κανάλι *IRC* επικοινωνίας να μην υλοποιείται στις γνωστές πόρτες 6666 και 6667. Μια λύση στο πρόβλημα αυτό είναι η χρησιμοποίηση υπογραφών σε επίπεδο *byte* και η σημείωση (*flag*) του πρωτοκόλλου ως τέτοιου, αν ανήκει δηλαδή ή όχι στα παρακολουθούμενα πρωτόκολλα επικοινωνίας. Η εξέταση όλων των πακέτων που διαπερνούν το δίκτυό μας για ταίριασμα με τις υπογραφές θα μπορούσε να γίνει, αλλά θα ήταν αποδοτική μόνο με εξειδικευμένο υλικό [17].

3.2.5 Ανίχνευση βασισμένη σε ανωμαλίες (anomaly-based)

Οι μέθοδοι ανίχνευσης ανωμαλιών, γνωστές και ως μέθοδοι ανίχνευσης με βάση συμπεριφορές (*behavior-based*), χρησιμοποιούν πληροφορίες για επαναλαμβανόμενη και ασυνήθιστη συμπεριφορά και προσπαθούν να ανιχνεύσουν εισβολές διακρίνοντας σημαντικές αποκλίσεις από την κανονική συμπεριφορά. Το πιο σημαντικό πλεονέκτημα των μεθόδων αυτών είναι η ικανότητά τους να ανιχνεύουν νέες επιθέσεις ενάντια στα συστήματα μας. Αυτό είναι πιθανό γιατί οι τεχνικές ανίχνευσης ανωμαλιών δεν αναλύουν τη δικτυακή κίνηση για να εντοπίσουν συγκεκριμένα πρότυπα, αλλά αντίθετα συγκρίνουν την τρέχουσα δραστηριότητα με μοντέλα προηγούμενης συμπεριφοράς. Το

μεγάλο μειονέκτημα όμως είναι ο υψηλός ρυθμός λάθος συναγερμών (*false alarms*) που παράγονται σε σύγκριση με τις τεχνικές ανίχνευσης που χρησιμοποιούν υπογραφές. Επειδή οποιαδήποτε σημαντική απόκλιση από την προηγούμενη 'μαθημένη' συμπεριφορά μπορεί να σημανθεί σαν εισβολή, είναι πολύ πιθανό ότι κάθε μη-απειλητική συμπεριφορά που πέφτει έξω από το κανονικό εύρος, να σημαίνεται ως εισβολή καταλήγοντας σε *false positive*.

Ένας άλλος περιορισμός της τεχνικής ανίχνευσης ανωμαλιών είναι ότι τα δεδομένα για την εκπαίδευση του συστήματος ανιχνεύσεων θα πρέπει να είναι ελεύθερα από οποιαδήποτε απειλητική συμπεριφορά γιατί αν μια επίθεση συμβεί κατά τη διάρκεια της περιόδου εκπαίδευσης, τότε η απειλητική συμπεριφορά θα γίνει μέρος της κανονικής συμπεριφοράς. Το πιο επιθυμητό χαρακτηριστικό όμως, ενός συστήματος ανίχνευσης εισβολέων όπως τα *bot*s, είναι η ικανότητά του να βρίσκεται ένα βήμα πιο μπροστά από τον επιτιθέμενο, η δυναμική του δηλαδή να ανιχνεύει νέες επιθέσεις. Επομένως, παρόλα τα μειονεκτήματα, οι τεχνικές ανίχνευσης ανωμαλιών είναι πολλά υποσχόμενες για την ανίχνευση νέων επιθέσεων εναντίον των υπολογιστών.

Στη συνέχεια εξετάζουμε τις βασικότερες τύπους ανώμαλης συμπεριφοράς που χρησιμοποιείται για τον εντοπισμό των *botnets*.

α) Ομοιότητα στη σχέση, τις απαντήσεις και το συγχρονισμό μεταξύ των υπολογιστών

Όπως έχει αναφερθεί όλα τα *bot*s εκτελούν κακόβουλες δραστηριότητες σύμφωνα με τις εντολές του *botmaster*. Προτείνονται τρία μετρικά (*metrics*) για την ανίχνευση τα οποία εξάγονται από τη συμπεριφορά των *botnets* [18].

Πρώτον, η σχέση (*relationship*) ανάμεσα στα *bot*s και τον *botmaster* η οποία είναι σχέση ενός-προς-πολλά. Για παράδειγμα όλα τα *bot*s θα πρέπει να συνδεθούν (*join*) στο ίδιο *IRC* κανάλι για να πάρουν εντολές.

Δεύτερον, οι απαντήσεις (responses) των *bot*s στις εντολές του *botmaster* οι οποίες είναι άμεσες και ακριβείς. Ειδικότερα για την επικοινωνία τους μέσω ενός καναλιού *IRC*, οι απαντήσεις των *bot*s είναι προγραμματισμένες με σταθερό χρόνο απόκρισης σε

αντίθεση με τις απαντήσεις των ανθρώπων οι οποίες έχουν τυχαίο χρόνο απόκρισης, αφού θα πρέπει να σκεφτούν πρώτα, αλλά και απρόβλεπτη συμβολοσειρά-κείμενο απάντησης.

Τρίτον, ο συγχρονισμός (*synchronization*) ανάμεσα στους υπολογιστές του δικτύου μας αποτελεί μια ισχυρή ένδειξη ύπαρξης *bot*s, αφού τα *bot*s ταυτόχρονα εκτελούν ενέργειες όπως για παράδειγμα μια επίθεση άρνησης υπηρεσίας (*DDoS*) ή η αναφορά τους στον *botmaster*. Η χρήση του μετρικού του συγχρονισμού είναι πολύ αποτελεσματική για την ανίχνευση όταν τα *bot*s κάνουν επιθέσεις *DDoS* ή *Spam* γιατί αυτού του είδους οι επιθέσεις πράγματι απαιτούν συγχρονισμό για να έχουν επιτυχία.

β) Χώρο-χρονική συσχέτιση όμοιας συμπεριφοράς (spatial-temporal correlation)

Αυτός ο τύπος ανώμαλης συμπεριφοράς, που στην ουσία αφορά την εκδήλωσης παρόμοιας συμπεριφοράς χωρίς αυτό όμως να είναι φυσιολογικό, αφορά κυρίως την ανίχνευση *botnets* που χρησιμοποιούν κεντρικοποιημένη αρχιτεκτονική για την υλοποίηση του *C&C*. Πρόκειται για χώρο-χρονική συσχέτιση (*spatial-temporal correlation*) στη δικτυακή κίνηση και χρησιμοποιεί στατιστικούς αλγορίθμους για την ανίχνευση των *botnets*. Η χώρο-χρονική συσχέτιση εξετάζει ομοιότητες στη συμπεριφορά σε ίδιες χρονικές στιγμές που μπορεί να συμβούν όταν τα *bot*s εκτελούν προγραμματισμένες από πριν δραστηριότητες σχετικές με το *C&C* και εφαρμόζεται από το πρόγραμμα ανίχνευσης '*BotSniffer'* [19].

Η τεχνική αυτή αποσκοπεί στον εντοπισμό πολυπληθών και όμοιων απαντήσεων στις εντολές που τα *bot*s λαμβάνουν από το *botmaster* ομαδοποιώντας χωρικά χαρακτηριστικά, όπως η διεύθυνση *IP* και η πόρτα του προορισμού, και χρησιμοποιώντας χρονικά χαρακτηριστικά, όπως συγκεκριμένα παράθυρα χρόνου.

Το πλεονέκτημα αυτής της τεχνικής είναι ότι μπορεί να χρησιμοποιηθεί για να γίνει αυτό-συσχέτιση (*autocorrelation*) έτσι ώστε να εντοπιστεί ακόμη και ένα μόνο *bot* που μπορεί να υπάρχει στο δίκτυο μας [20]. Η αυτό-συσχέτιση μπορεί να αναγνωρίσει για παράδειγμα εάν η *HTTP* δραστηριότητα επίσκεψης ενός υπολογιστή σε κάποιον

εξωτερικό *web server* εμφανίζει επαναλαμβανόμενη συμπεριφορά π.χ. περιοδική σύνδεση σε κάποιο *C&C* εξυπηρετητή.

γ) Ανίχνευση με συσχέτιση διαλόγων (dialog-based)

Η ανίχνευση των *botnets* με συσχέτιση των διαλόγων (*dialog-based*) που παρατηρούνται στις διπλής κατεύθυνσης (*two-way*) ροές επικοινωνίας ανάμεσα σε κάθε υπολογιστή του δικτύου μας και τους εξωτερικούς επιτιθέμενους, προσπαθεί να εντοπίσει στοιχεία ανταλλαγής δεδομένων τα οποία ταιριάζουν με κάποιο μοντέλο μόλυνσης (*infection model*) [21]. Τα μοντέλα μόλυνσης αποτελούνται από μια συγκεκριμένη σειρά 'μολυσματικών' γεγονότων ή αλλιώς γεγονότων επιθέσεων (*attack events*), που όταν ανιχνευθούν σε κάποιο υπολογιστή του δικτύου μας, μας επιτρέπουν με σχετική βεβαιότητα να αποφανθούμε ότι αυτός ο υπολογιστής είναι μολυσμένος, δηλαδή *bot*. Ο μολυσματικός διάλογος που αποτελείται από συγκεκριμένα ανιχνεύσιμα γεγονότα επιθέσεων, καταγράφεται για διάφορα χρονικά παράθυρα (*temporal windows*) μέσα στα οποία γίνεται η συσχέτιση για τον εντοπισμών των *bot*s αλλά και των *botmaster*s που διαχειρίζονται τα *bot*s.

Η τεχνική αυτή μπορεί να εντοπίσει κεντρικοποιημένα, αλλά και κατανεμημένα *botnets* και χρησιμοποιείται από το πρόγραμμα ανίχνευσης '*BotHunter'* το οποίο θα περιγράψουμε εκτενέστερα στα εργαλεία ανίχνευσης στο ομώνυμο κεφάλαιο.

Κεφάλαιο 4. Τεχνικές απόκρυψης

Τα *botnets* εξελίσσονται συνεχώς και γίνονται πιο έξυπνα και πολύπλοκα. Έχουν τη δυνατότητα να πολλαπλασιάζονται όπως τα σκουλήκια (worms), να κρύβονται όπως οι ιοί (viruses) και να χρησιμοποιούν την ισχύ τους για την εκκίνηση μεγάλων κατανεμημένων και συγχρονισμένων επιθέσεων.

Τα *botnets* βελτιώνονται έτσι ώστε να αποφεύγουν την ανίχνευσή τους από τα προγράμματα προστασίας, όπως τα antivirus και τα συστήματα ανίχνευσης εισβολών (*IDS*s). Γίνονται πιο ανθεκτικά σε νέες τεχνικές ανίχνευσης, όπως η ανίχνευση που βασίζεται στην ανώμαλη συμπεριφορά και γενικότερα πιο ανθεκτικά στις τεχνικές ανίχνευσης που αναφέραμε στο προηγούμενο κεφάλαιο.

Στη συνέχεια θα αναφέρουμε τις βασικότερες τεχνικές αποφυγής της ανίχνευσης (*detection evasion techniques*), αφού πρώτα εξηγήσουμε την επίδραση που αυτές έχουν στα ίδια τα *botnets* και τη δημιουργία τους.

4.1. Κόστη των τακτικών απόκρυψης

Οι τακτικές απόκρυψης ή αλλιώς αποφυγής της ανίχνευσης έχουν δύο συνδεδεμένα με αυτές κόστη: α) την πολυπλοκότητα της υλοποίησης (*implementation complexity*) και β) την επίδραση στη χρησιμότητα του *botnet* (*effect on botnet utility*) [22].

Η πολυπλοκότητα της υλοποίησης έχει να κάνει με την ευκολία που οι δημιουργοί του *bot* μπορούν επαυξητικά να το τροποποιήσουν ώστε να αποφεύγει την ανίχνευση. Οι τροποποιήσεις μπορεί να είναι χαμηλής πολυπλοκότητας χωρίς να απαιτούν την αλλαγή του πηγαίου κώδικα, αλλά απλά την αλλαγή κάποιων εντολών του *C&C* ή το πακετάρισμα (*packing*) του *binary* με νέο τρόπο. Μπορεί όμως να απαιτούνται και μεγάλες και περίπλοκες αλλαγές στον πηγαίο κώδικα, για την αλλαγή για παράδειγμα του πρωτοκόλλου υλοποίησης του *C&C* καναλιού.

Η ποσοτικοποίηση της χρησιμότητας ενός *botnet* είναι μια ουσιαστική για την έρευνα ερώτηση. Μπορούμε να αναγνωρίσουμε κάποια χαρακτηριστικά που επηρεάζουν την αγοραία αξία ενός *botnet*. Ο υπολογισμός της συνολικής χρησιμότητας ιδανικά θα συνδύαζε τα παρακάτω χαρακτηριστικά με ένα τρόπο ο οποίος αντανακλά την τιμή στην οποία το *botnet* θα μπορούσε να ενοικιαστεί για την πραγματοποίηση επιθέσεων:

- Η ποικιλία των επιθέσεων, ο αριθμός δηλαδή των διαφορετικών επιθέσεων που ένα *botnet* είναι ικανό να διεξάγει.
- Ο χρόνος που απαιτείται για να ξεκινήσει μια επίθεση, ο οποίος δείχνει αν τα *bot*s είναι διαθέσιμα σε πραγματικό χρόνο καθώς και την καθυστέρηση του *C&C* δικτύου.
- Το μέγεθος του *botnet*, ο αριθμός δηλαδή των *bot*s που μπορούν να συμμετέχουν σε κάποια επίθεση.
- Ο ρυθμός των επιθέσεων, δηλαδή ο αριθμός των επιθέσεων που μπορούν να εκκινηθούν ανά ώρα.
- Το επίπεδο συγχρονισμού, το οποίο προσδιορίζει το πάνω όριο στη διαφορά του χρόνου ανάμεσα στις ενέργειες του πρώτου και τελευταίου *bot* που συμμετέχουν σε μια επίθεση.

4.2. Χρήση μη-κεντρικοποιημένων αρχιτεκτονικών

Όπως ήδη αναφέραμε τα *bot*s επικοινωνούν με αλλά *bot*s και το *botmaster* σύμφωνα με καλά ορισμένα δικτυακά πρωτόκολλα. Οι κακόβουλοι δημιουργοί των *botnets* (*back hat hackers*) σε πολλές περιπτώσεις προτιμούν τη χρήση υπαρχόντων πρωτοκόλλων επικοινωνίας τα οποία είναι υλοποιημένα με δημόσια διαθέσιμα εργαλεία λογισμικού από το να δημιουργούν νέα πρωτόκολλα επικοινωνίας.

Παρόλο που το πρωτόκολλο *IRC* αποτελεί το κυρίαρχο πρωτόκολλο για την επικοινωνία στα *botnet* λόγω της απλότητας και ωριμότητάς του, πολλά *botnets* έχουν ήδη αρχίσει να μετακινούνται προς άλλα πρωτόκολλα για την επικοινωνία, αφού η παρακολούθηση της κίνησης *IRC* γίνεται ολοένα και πιο έντονη. Η κίνηση *IRC* μπορεί να αναγνωριστεί πιο εύκολα, αφού πολλά συστήματα ανίχνευσης εισβολών διαθέτουν φίλτρα για την αναγνώρισή της ακόμη και αν δε χρησιμοποιούνται οι γνωστές πόρτες της

υπηρεσίας αυτής (6666, 6667). Για παράδειγμα ο παρακάτω κανόνας ανιχνεύει στο περιεχόμενο την εντολή "*NICK*" που χρησιμοποιείται για τον καθορισμό του ψευδωνύμου κατά την έναρξη της επικοινωνίας με κάποιο *IRC* εξυπηρετητή:

```
alert tcp any any -> any any (msg:"IRC TRAFFIC DETECTED BY NICK CHANGE";
    flow: to_server,established; content:"NICK "; nocase; offset: 0; depth: 5;
    flowbits: set,is_proto_IRC; flowbits: noalert; sid:9000075; rev:1;)
```

Ο κανόνας αυτός θέτει ένα *flowbit* ώστε να δηλωθεί ότι η επερχόμενη κίνηση θα είναι τύπου *IRC* οπότε μόνο σ' αυτή την περίπτωση θα πρέπει να παρακολουθηθεί.

Επίσης τα τείχη προστασίας (*firewalls*) μπορούν να διαμορφωθούν ώστε να μπλοκάρουν την *IRC* κίνηση, όμως είναι πολύ πιο δύσκολο να ανιχνευθούν κανάλια *IRC* τα οποία έχουν μπει σε *HTTP* τούνελ (*tunnels*). Τα *botnets* που χρησιμοποιούν το *HTTP* πρωτόκολλο είναι δυσκολότερο να ανιχνευθούν γιατί η κίνηση που δημιουργούν αναμιγνύεται με την περισσότερη κίνηση του διαδικτύου (*HTTP* κίνηση), η οποία διαπερνά ελεύθερα τα περισσότερα firewalls.

Όμως, η επικοινωνία μέσω *IRC* αλλά και μέσω *HTTP* χρησιμοποιούν κεντρικοποιημένες αρχιτεκτονικές και συνεπώς υπάρχει ένα κεντρικό σημείο αποτυχίας. Σε περίπτωση που εντοπιστεί και εξουδετερωθεί ο *C&C* εξυπηρετητής, όλο το *botnet* βγαίνει εκτός λειτουργίας. Η λύση που προτείνεται εδώ είναι ο διαμοιρασμός των *bot*s σε πολλούς *C&C* εξυπηρετητές. Έτσι στην περίπτωση εντοπισμού κάποιου από αυτούς το *botnet* να εξακολουθεί να λειτουργεί και να έχει τη δύναμη να μεγαλώσει υπονομεύοντας νέους υπολογιστές.

Τα μη-κεντρικοποιημένα ή αλλιώς κατανεμημένα *botnets* είναι πιο ανθεκτικά και μειώνουν τις πιθανότητες εντοπισμού και εξουδετέρωσής τους. Πράγματι υπάρχει μια πρόσφατη τάση αύξησης της ανάπτυξης των P2P *botnets* και αναμένεται το επίπεδο της πολυπλοκότητας τους επίσης να αυξηθεί [23]. Στα *P2P botnets* το '*Command and Control*' ενσωματώνεται στα ίδια τα *bot*s. Οι υπονομευμένοι υπολογιστές μπορούν να

λειτουργούν και ως απλά *bot*s (πελάτες) αλλά και ως *C&C* εξυπηρετητές χωρίς ένα μοναδικό σημείο αποτυχίας κάνοντας δυσκολότερη την ανίχνευσή τους.

Κάποια ανωτέρου επιπέδου τεχνικής *botnets* χρησιμοποιούν κανάλια επικοινωνίας όπως τα *TCP* και *ICMP* τούνελ, ακόμη και *IPv6* τούνελ [24] για την ενθυλάκωση του *C&C*. Τέλος έχουν γίνει τεχνικές συζητήσεις για την πιθανότητα χρήσης του Skype και των άμεσων μηνυμάτων (*Instant Messaging*) για την υποστήριξη της επικοινωνίας του '*Command and Control*'.

4.3. Αναφορά στον *botmaster* με συνήθη πρωτόκολλα

Η αναφορά των δραστηριοτήτων των *bot*s προς το *botmaster* μπορεί να γίνεται χρησιμοποιώντας το *HTTP* πρωτόκολλο (μέθοδος *POST* του *HTTP*) ή την υπηρεσία *Email* (*SMTP* πρωτόκολλο) ανεξάρτητα αν το υπόλοιπο '*Command and Control*' είναι υλοποιημένο με κεντρικοποιημένη ή κατανεμημένη αρχιτεκτονική. Έτσι, οι πληροφορίες που συλλέγονται από τα *bot*s, όπως αρχεία ή κωδικοί των χρηστών ή τα αποτελέσματα των επιθέσεων μπορούν να αποστέλλονται στον *botmaster* χωρίς να δημιουργούν ιδιαίτερες υποψίες. Όσον αφορά την αποστολή των αναφορών με *email*, τα *bot*s σε πολλές περιπτώσεις δε χρειάζεται να χρησιμοποιήσουν δικούς τους ή υπονομευμένους διακομιστές αλληλογραφίας (*smtp server*s), αφού μπορούν εύκολα να μάθουν τους διακομιστές εξερχόμενης αλληλογραφίας του ίδιου του θύματος.

Αυτή η τεχνική μειώνει σημαντικά τις πιθανότητες εντοπισμού των *bot*s, αφού αυτή η μορφή επικοινωνίας είναι σχεδόν πάντα επιτρεπτή. Επίσης λόγω του μεγάλου όγκου της πληροφορίας που μεταφέρεται μέσω των δύο πρωτοκόλλων που αναφέραμε η ανίχνευση σ' αυτά είναι ακόμη πιο δύσκολη.

4.4. Χρήση Fast-flux DNS

Η πιο δημοφιλής τεχνική για ένα *botmaster* να 'συλλέξει' τους υπονομευμένους υπολογιστές (*bot*s) σε ένα *botnet* περιλαμβάνει το 'Σύστημα Ονομάτων Τομέα' γνωστό ως *Domain Name System* (*DNS*). Έτσι όπως ένας πάροχος υπηρεσιών διαδικτύου (ISP)

χρησιμοποιεί το δυναμικό DNS για να αντιστοιχήσει ένα όνομα τομέα (*domain*) σε ένα υπολογιστή με διαφορετική διεύθυνση IP κάθε φορά, έτσι και τα *bot*s περιλαμβάνουν στον κώδικά τους 'σκληρά κωδικοποιημένα' (*hard coded*) ονόματα τομέων τα οποία έχουν αντιστοιχηθεί από δυναμικούς παρόχους *DNS*. Μερικά *botnets* τρέχουν τη δική τους κατανεμημένη υπηρεσία *DNS* η οποία εκτελείται σε υψηλούς αριθμούς πορτών (*high port numbers*) ώστε να αποφεύγεται η ανίχνευση από τα προγράμματα ασφαλείας που βρίσκονται στην πύλη (*gateway*) του δικτύου.

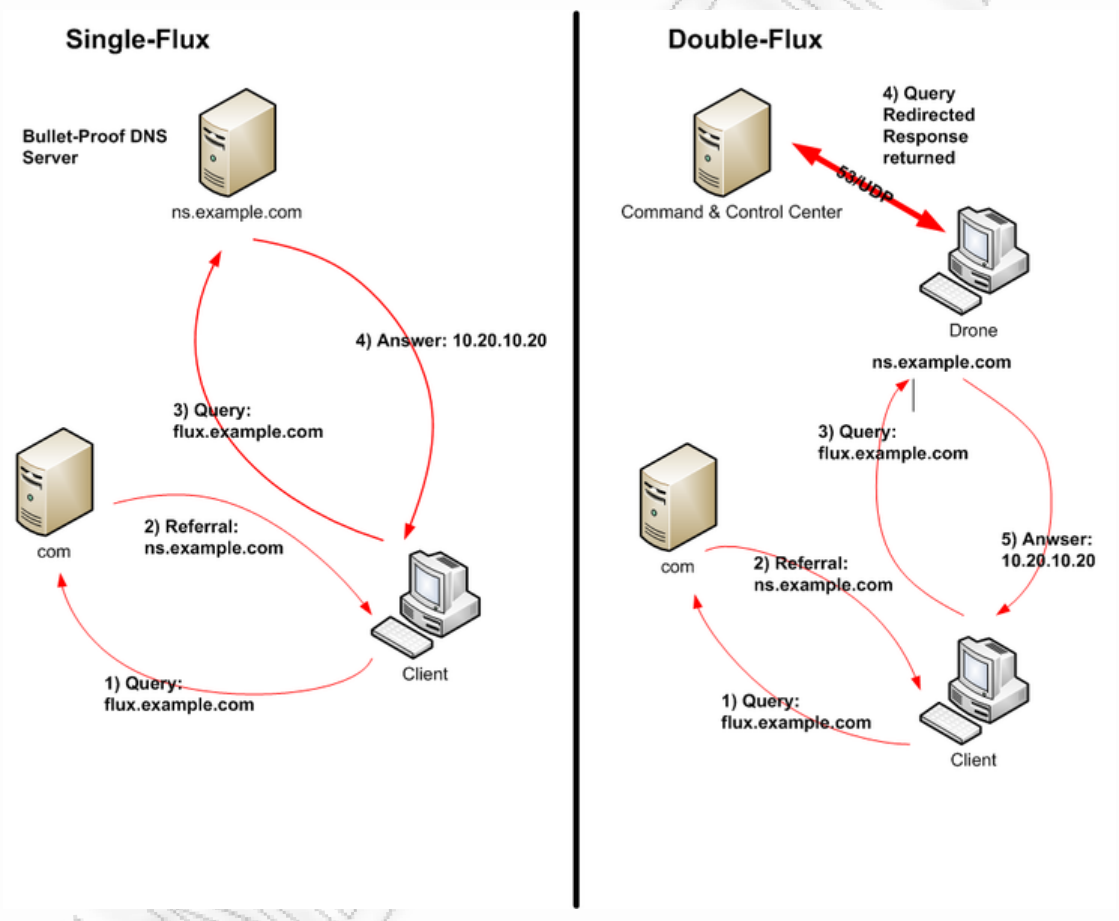
Οι νέες γενιές *botnets* όμως, πηγαίνουν ένα βήμα ακόμη πιο πέρα χρησιμοποιώντας μια νέα τεχνική που ονομάζεται *Fast-flux DNS*. Πρόκειται για μια τεχνική που σκοπό έχει την απόκρυψη των ιστοχώρων που παραδίδουν περιεχόμενο ηλεκτρονικού ψαρέματος (*phishing*) και γενικότερα κακόβουλο περιεχόμενο, χρησιμοποιώντας διαρκώς μεταβαλλόμενα δίκτυα υπονομευμένων υπολογιστών που λειτουργούν ως πληρεξούσιοι (*proxies*). Αυτή η τεχνική κάνει τα κακόβουλα δίκτυα περισσότερο ανθεκτικά στη ανακάλυψή τους και στα αντίμετρα εναντίον τους [25].

Στα *botnets* επιπλέον αυτή η τεχνική χρησιμοποιείται για να συνδέσει και να αποκρύψει τους *C&C* εξυπηρετητές όπως στο κανονικό DNS. Η μεγάλη διαφορά όμως είναι ότι τα ονόματα τομέων (*DNS domains*) που χρησιμοποιούν τα *bot*s είναι κλεμμένα, υπονομευμένα ή καταχωρημένα (*registered*) με συνήθως κλεμμένες πιστωτικές κάρτες.

Ο πιο απλός τύπος *fast-flux DNS* που αναφέρεται ως *Single-flux* χαρακτηρίζεται από πολλαπλούς μεμονωμένους κόμβους, στην περίπτωση μας τα *bot*s, οι οποίοι εγγράφουν και διαγράφουν τις διευθύνσεις τους σαν μέρος μιας λίστας εγγραφών A (*DNS* A *records*) για ένα μοναδικό όνομα τομέα (*DNS name*). Αυτό συνδυάζει το DNS εκ περιτροπής ανάμεσα στους κόμβους με πολύ μικρές τιμές (συνήθως λιγότερο από 5 λεπτά ή 300 δεύτερα) για το χρόνο ζωής (*Time To Live*) κάθε κόμβου, έχοντας ως αποτέλεσμα τη συνεχή αλλαγή της λίστας των διευθύνσεων προορισμού για ένα μοναδικό όνομα τομέα. Η λίστα μπορεί να έχει μήκος εκατοντάδων χιλιάδων καταχωρήσεων και κάθε πελάτης (*bot*) θα δοκιμάσει τις διευθύνσεις IP της λίστας μέχρις ότου συνδεθεί σε κάποια.

Ένας πιο προχωρημένος τύπος *fast-flux* που ονομάζεται *Double-flux* χαρακτηρίζεται από πολλαπλούς μεμονωμένους κόμβους, στην περίπτωση μας τα *bot*s, οι οποίοι

εγγράφουν και διαγράφουν τις διευθύνσεις τους σαν μέρος μιας λίστας εγγραφών NS (*DNS NS records*) για τη ζώνη *DNS*. Δημιουργούνται δηλαδή πολλαπλοί εξυπηρετητές ονομάτων (*name* servers) για μια ζώνη *DNS*, παρέχοντας ένα επιπρόσθετο επίπεδο πλεονασμού και βιωσιμότητας των *bot*s μέσα στο κακόβουλο δίκτυο.



Σχήμα 4.1 Σύγκριση Τεχνικών Fast Flux

4.5. Λειτουργία διακομιστών διαμεσολάβησης

Όπως αναφέρθηκε, οι επιτιθέμενοι χρησιμοποιούν τους υπονομευμένους υπολογιστές ως εξυπηρετητές για να αποφύγουν την ανίχνευση και την αναγνώρισή τους. Οι κρυφές υπηρεσίες πύλης δικτύου και διακομιστή διαμεσολάβησης (*gateway & proxy services*) στα *bot*s, επίσης υποστηρίζουν τις ενέργειες απόκρυψης των επιτιθέμενων. Οι περισσότερο συχνά παρατηρούμενες λειτουργίες proxy στα *botnets* περιλαμβάνουν [26]:

- Generic port redirection
- *HTTP* proxy
- Socks proxy
- *IRC* bounce

Η γενική ανακατεύθυνση πορτών (*port redirection*) επιτρέπει στις εισερχόμενες δικτυακές συνδέσεις που φτάνουν σε κάποιο *bot* να στέλνονται κατευθείαν σε κάποιον άλλο υπολογιστή. Μπορεί να γίνει ανακατεύθυνση σε οποιαδήποτε υπηρεσία βασίζεται στο πρωτόκολλο *IP* συμπεριλαμβανόμενων όλων των αιτημάτων *TCP* και *UDP*. Η γενική ανακατεύθυνση πορτών μετατρέπει τα *bot*s ως αναπηδήσεις (bounces) μέσω των οποίων οι επιτιθέμενοι μπορούν να κρύψουν την πραγματική τους τοποθεσία. Για παράδειγμα οι *botmaster* μπορούν να κρύβουν την τοποθεσία τους όταν συνδέονται σε *IRC* εξυπηρετητές και ελέγχουν το *botnet* τους. Έτσι αν συνδεθούν διαμέσω υπονομευμένων συστημάτων στην Αμερική, μετά στη Ρωσία, μετά στην Βόρεια Κορέα και τελικά στον *IRC* εξυπηρετητή, η εύρεση των ιχνών τους μπορεί να είναι αδύνατη. Η ίδια τεχνική μπορεί να χρησιμοποιηθεί για την αποστολή ανεπιθύμητης αλληλογραφίας, την εκκίνηση επιθέσεων ηλεκτρονικού ψαρέματος (*phishing*) κτλ.

Ένα πιο συγκεκριμένο παράδειγμα γενικής ανακατεύθυνσης είναι το τούνελ (tunneling) με χρήση του πρωτοκόλλου *GRE*. Το πρωτόκολλο *GRE* έχει το πλεονέκτημα να μην περιορίζεται μόνο στη δημιουργία τούνελ για τα πρωτόκολλα *TCP* και *UDP*

μόνο. Μπορεί να ενθυλακώσει και να παραδώσει σχεδόν κάθε μορφή πακέτου διαμέσου του δρομολογημένου τούνελ.

Οι διακομιστές διαμεσολάβησης *HTTP* και *HTTP*S εκτελούνται στα *bot*s και χρησιμοποιούνται για την πρόσβαση σε πηγές του διαδικτύου χωρίς να αποκαλύπτεται η τοποθεσία του επιτιθέμενου, εφόσον οι ιστοχώροι που καταγράφουν τις επισκέψεις μπορούν να δουν μόνο τις διευθύνσεις IP των *bot*s και όχι των επιτιθέμενων. Χρησιμοποιώντας αυτή την τεχνική οι *botmaster*s μπορούν να διεξάγουν επιθέσεις τύπου πλαστών κλικ (click fraud) χωρίς να εντοπίζονται.

Socks είναι ένα πρωτόκολλο που μπορεί να χρησιμοποιηθεί για να προσφέρει υπηρεσίες διαμεσολάβησης (*proxy*) σε άλλες υπηρεσίες βασιζόμενες στο *TCP* και το *UDP*. Όπως συμβαίνει και με τις περισσότερες υπηρεσίες διαμεσολάβησης, ο βασικός σκοπός του '*Socks proxy*' πρωτοκόλλου είναι η απόκρυψη της πραγματικής IP διεύθυνσης του επιτιθέμενου. Η πώληση ή ενοικίαση *botnets* με λειτουργίες '*Socks proxy*' για αποστολή ανεπιθύμητης αλληλογραφίας (*spam*) είναι συχνό φαινόμενο. Επειδή οι μολυσμένοι με *bot*s υπολογιστές είναι καλά κατανεμημένοι σε πολλά δίκτυα ανά τον κόσμο, οι διευθύνσεις *IP* των *proxies* είναι απίθανο να συμπεριληφθούν σε μαύρες λίστες χαρακτηρισμένες ως spam διευθύνσεις. Ακόμη όμως και αν εντοπιστούν και μπουν σε μαύρες λίστες, τα *bot*s είναι εύκολο να μετακινηθούν ή να δοθούν προς πώληση ή ενοικίαση για άλλες κακόβουλες επιθέσεις. Αυτοί οι παράγοντες συντελούν σημαντικά στις μικρές πιθανότητες εντοπισμού και μπλοκαρίσματος των αναμεταδοτών spam.

Τέλος, η λειτουργία *IRC bounce* είναι ένας τύπος υπηρεσίας διαμεσολάβησης για συνδέσεις *IRC*. Δίνει τη δυνατότητα απόκρυψης του *botmaster* όταν αυτός εκτελεί ενέργειες σχετικές με το '*Command and Control*', αφού συνδέεται στον *IRC* εξυπηρετητή διαμέσου κάποιου *bot*. Επίσης προστατεύει τον επιτιθέμενο από άλλους επιτιθέμενους αφού η ζημιά από μια επίθεση άρνησης υπηρεσίας (*DDoS*) με στόχο τον επιτιθέμενο θα επηρέαζε μόνο το σύνδεσμο του ενδιάμεσου *bot*-θύματος. Ο επιτιθέμενος θα μπορούσε να χρησιμοποιήσει κάποιο άλλο *bot* για να αναλάβει και πάλι τον έλεγχο του καναλιού.

4.6. Καθυστέρηση στις αποκρίσεις των *bot*s

Μια τεχνική αποφυγής της ανίχνευσης με καλά αποτελέσματα είναι η εφαρμογή καθυστέρησης στις αποκρίσεις των *bot*s. Τα *bot*s μπορούν να εκτελούν τις εντολές του *botmaster* εισάγοντας προηγουμένως μια τυχαία καθυστέρηση. Αυτό θα παράκαμπτε τις μεθόδους ανίχνευσης που βασίζονται στο συγχρονισμό και τη χρονική συσχέτιση. Γενικότερα, για μεθόδους ανίχνευσης που βασίζονται στο χρόνο, η τεχνική αποφυγής τους είναι η επιβολή μεγάλων διαστημάτων αποκρίσεων από τα *bot*s και η κατανομή των κακόβουλων δραστηριοτήτων στο χρόνο, έτσι ώστε να περάσουν κάτω από τα ραντάρ ανίχνευσης. Οι τεχνικές ανίχνευσης που αντιστοιχούν τα γεγονότα επιθέσεων με τις κακόβουλες δραστηριότητες των *bot*s μπορούν να παρακαμφτούν αν κατανείμουμε σε σχετικά μεγάλα χρονικά διαστήματα τις δραστηριότητες των *bot*s.

Ο χρόνος απόκρισης δεν είναι ιδιαίτερα σημαντικός στην περίπτωση που η δραστηριότητα των *bot*s αφορά την αναφορά αποτελεσμάτων από τις επιθέσεις που διεξήγαγαν ή την αποστολή πληροφοριών από τους υπολογιστές των θυμάτων τους στο *botmaster*. Όμως η εφαρμογή αυτή της τακτικής απόκρυψης μειώνει το ρυθμό και την ένταση των επιθέσεων που κάνει ένα *botnet*, αφού οι επιθέσεις είναι λιγότερο συγχρονισμένες και λιγότερο μαζικές. Σαν παράδειγμα εδώ μπορούμε να αναφέρουμε την επίθεση άρνησης υπηρεσίας (*DDoS*).

4.7. Κρυπτογράφηση καναλιού *C&C*

Μια από τις βασικότερες τεχνικές αποφυγής της ανίχνευσης που χρησιμοποιείται ολοένα και περισσότερο στα νέα *botnets* είναι η κρυπτογράφηση του καναλιού επικοινωνίας των *bot*s με το *botmaster*. Κρυπτογραφούνται όλα τα δεδομένα που ανταλλάσσονται, όπως τα αρχεία που κατεβάζουν τα *bot*s (π.χ. *binaries* ή λίστες *spam*), οι εντολές που στέλνονται στα *bot*s από το *botmaster*, αλλά και οι αναφορές που στέλνουν τα *bot*s σε αυτόν. Οι εντολείς (*botmaster*s) μπορούν να αναγνωρίζονται μοναδικά από την κατοχή ασφαλών κλειδιών κρυπτογράφησης. Τα *bot*s κρυπτογραφούν τα δεδομένα που θέλουν να αποστείλουν με ένα δημόσιο κλειδί το οποίο περιέχεται στον

ίδιο τον κώδικα (λογισμικό) του *bot*. Μόνο με το ιδιωτικό κλειδί που ο *botmaster* κατέχει μπορούν τα δεδομένα που συνέλλεξε το *bot* να διαβαστούν.

Το προφανές πρόβλημα που δημιουργεί αυτή η τεχνική στα συστήματα προστασίας και ανίχνευσης εισβολών είναι ότι αποτρέπει την ανίχνευση στο *payload* της κρυπτογραφημένης κίνησης. Τα δικτυακά συστήματα ανίχνευσης εισβολών (*NIDS*) δεν έχουν πρόσβαση στα κλειδιά κρυπτογράφησης έτσι δεν μπορούν να αποκρυπτογραφήσουν τα δεδομένα και η ανάλυση των δεδομένων δεν είναι εφικτή. Από την πλευρά του δικτύου είναι πράγματι δύσκολο να αναλύσουμε την κρυπτογραφημένη κίνηση. Ένα αντίμετρο που προτείνεται στην κρυπτογράφηση είναι η μέτρηση της εντροπίας (*entropy*) στα κρυπτογραφημένα πακέτα. Αν μετρούσαμε την εντροπία των *payloads* στις δικτυακές ροές, τότε θα παρατηρούσαμε ομοιότητες ανάμεσα στα διαφορετικά μέλη ενός *botnet* [27], εφόσον γνωρίζουμε ότι τα *bot*s έχουν ίδια χαρακτηριστικά επικοινωνίας.

4.8. Σύγχυση δικτυακών ροών

Η τεχνική αποφυγής της ανίχνευσης με σύγχυση (*obfuscation*) στις δικτυακές ροές περιλαμβάνει την εσκεμμένη εισαγωγή πακέτων ή bytes στις ροές που δημιουργούνται από τα μέλη ενός *botnet*, αλλά και την εσκεμμένη δημιουργία νέων ροών. Μια προχωρημένη τεχνική είναι η τυχαιοποίηση του μοτίβου επικοινωνίας (*communication pattern*) κάθε ξεχωριστού μέλους ενός *botnet*, τυχαιοποιώντας για παράδειγμα τον αριθμό των πακέτων ανά ροή, εισάγοντας δηλαδή τυχαία πακέτα στη ροή, και τυχαιοποιώντας τον αριθμό των bytes ανά πακέτο με την εισαγωγή τυχαίων bytes στα πακέτα. Προφανώς ο παραλήπτης γνωρίζει τον τρόπο διαχωρισμού της περιττής πληροφορίας από την χρήσιμη πληροφορία στις ροές και τα πακέτα.

Ειδικότερα στα *P2P botnets* όπου γίνονται πολλές συνδέσεις καθώς και πολλές ανεπιτυχείς προσπάθειες σύνδεσης λόγω της φύσης της επικοινωνίας, η σύγχυση είναι μεγαλύτερη. Από τη μια μεριά τα *bot*s συνδέονται και αποσυνδέονται από το *P2P* δίκτυο με μεγάλη τυχαιότητα. Από την άλλη μεριά η δημιουργία πλαστών συνδέσεων δημιουργεί μεγαλύτερη σύγχυση για την τοποθεσία των '*Command and Control*' *bot*s.

Με τη δημιουργία πολλών συνδέσεων απαιτείται περισσότερος χρόνος για την ανάλυσή τους και συνεπώς περισσότερος χρόνος από τους διαχειριστές για τον εντοπισμό των *C&C* εξυπηρετητών. Επιπλέον η εισαγωγή τυχαίων πακέτων ή bytes στις *P2P* ροές κάνει την ανάλυση και ανίχνευση ακόμη πιο δύσκολη.

4.9. Κατανεμημένη σάρωση δικτύων (scan)

Η σάρωση δικτύων (*scanning*) για εύρεση ενεργών υπολογιστών, ανοικτών πορτών και πιθανώς ευπαθών υπηρεσιών σε υπολογιστές αποτελεί το πρώτο βήμα αναγνώρισης (*reconnaissance*) των στόχων από τους κακόβουλους εισβολείς. Μετά την απόκτηση πληροφοριών για το θύμα επιχειρείται η διείσδυση χωρίς εξουσιοδότηση στο σύστημά του.

Η σάρωση είναι απαραίτητη γιατί αποκαλύπτει ποιες υπηρεσίες είναι διαθέσιμες στους υπολογιστές για εκμετάλλευση (*exploit*). Θα μπορούσε να επιχειρηθεί άμεση επίθεση σε μια υπηρεσία σε κάποιο υπολογιστή μέσω μιας διεύθυνσης IP, χωρίς να γνωρίζουμε αν ο υπολογιστής αυτός είναι ενεργός και αν πράγματι διαθέτει την υπηρεσία. Για παράδειγμα να γίνει προσπάθεια εκμετάλλευσης κάποιας γνωστής αδυναμίας του *IIS* web *server*, χωρίς να γνωρίζουμε όμως αν στη διεύθυνση *IP* του προορισμού είναι ενεργή κάποια μηχανή και πολύ περισσότερο χωρίς να γνωρίζουμε αν η μηχανή αυτή λειτουργεί ως web εξυπηρετητής. Κάτι τέτοιο όμως θα δημιουργούσε περισσότερους συναγερμούς στα προγράμματα ανίχνευσης εισβολών αφού η εκμετάλλευση μιας αδυναμίας του λογισμικού είναι σίγουρα κακόβουλη πράξη. Παρόλα αυτά η τεχνική αυτή θα μπορούσε να χρησιμοποιηθεί σε επίπεδο τοπικού δικτύου όπου ένας υπονομευμένος υπολογιστής θα προσπαθούσε να υπονομεύσει με τη σειρά του τον web εξυπηρετητή του δικτύου με την προϋπόθεση ότι δεν παρακολουθείται η τοπική ενδο-κίνηση, πράγμα που συμβαίνει συχνά.

Η σάρωση των δικτύων μπορεί να ανιχνευθεί σχετικά εύκολα από τα προγράμματα ανίχνευσης, ειδικότερα αν η τεχνική σάρωσης είναι αρκετά επιθετική όταν σαρώνονται δηλαδή πολλές διευθύνσεις *IP* και πόρτες *TCP* ή *UDP* από ένα επιτιθέμενο υπολογιστή

μέσα σε λίγο χρόνο. Μια πιο επιθετική τεχνική σάρωσης χαρακτηρίζεται από τον αριθμό των προσπαθειών σύνδεσης που επιχειρείται και από τον τύπο της σάρωσης π.χ. '*connect scan'*, '*syn scan'*, 'fin scan' κτλ. Για αυτό το λόγο η τεχνική σάρωσης που χρησιμοποιούν τα *bot*s είναι συχνά κατανεμημένη. Κάθε *bot* σύμφωνα με τις εντολές του *botmaster* αναλαμβάνει τη σάρωση ενός συγκεκριμένου τμήματος του δικτύου (ένα μικρό υποδίκτυο) προς επίθεση και ένα συγκεκριμένο μικρό εύρος πορτών. Με αυτό τον τρόπο μειώνονται οι πιθανότητες ανίχνευσης της επίθεσης αφού οι επιχειρούμενες συνδέσεις ξεκινούν από πολλούς υπολογιστές οι οποίοι βρίσκονται σε διαφορετικά γεωγραφικά σημεία.

4.10 Πολυμορφισμός των εκτελέσιμων αρχείων

Ο πολυμορφισμός (*polymorphism*) είναι μια τεχνική μετάλλαξης του κώδικα ενός εκτελέσιμου προγράμματος (*binary*), διατηρώντας όμως τον αρχικό αυθεντικό αλγόριθμο άθικτο. Ο πολυμορφισμός δεν είναι κάτι καινούριο στον κόσμο του κακόβουλου λογισμικού αφού χρησιμοποιείται για πολύ καιρό στους ιούς (*viruses*) και τα σκουλήκια (*worms*). Παρόλα αυτά τα προγράμματα προστασίας έχουν ακόμη πρόβλημα στον εντοπισμό του πολυμορφικού κακόβουλου λογισμικού.

Οι νέες γενιές *bot*s, όπως είναι το Storm *bot*, μεταλλάσσουν τα εκτελέσιμα αρχεία και οι *bot* herders εισάγουν πολλαπλές εκδόσεις στα *botnets* διαμέσου κάποιου υπονομευμένου υπολογιστή. Εκτιμάται ότι το *Storm botnet* διένειμε περισσότερες από 40.000 παραλλαγές σε διάρκεια 12 ημερών. Ο χρόνος που χρειάζονται οι εταιρείες προστασίας από το κακόβουλο λογισμικό για να φτιάξουν νέες υπογραφές ή ευριστικές τεχνικές για τα πολυμορφικά εκτελέσιμα δίνει τη δυνατότητα στα *bot*s να μολύνουν πολλούς άλλους υπολογιστές εν τω μεταξύ.

4.11 Χρήση *rootkit* s, απόκρυψη σε επίπεδο πυρήνα

Με τον όρο *rootkit* εννοούμε κάποιο κακόβουλο πρόγραμμα το οποίο είναι σχεδιασμένο ώστε να αναλαμβάνει το θεμελιώδη έλεγχο (πρόσβαση '*root'* στο *Unix* ή '*administrator'* στα *Windows*) ενός υπολογιστή χωρίς εξουσιοδότηση φυσικά. Σκοπός είναι να αρπάξει τον έλεγχο του λειτουργικού συστήματος του υπολογιστή. Τυπικά τα *rootkit* s κρύβουν την παρουσία τους ανατρέποντας ή εισβάλοντας στους μηχανισμούς ασφαλείας του λειτουργικού. Για να το πετύχουν αυτό κρύβουν εκτελούμενες διεργασίες (processes), αρχεία και δεδομένα του συστήματος όπως καταχωρήσεις στο μητρώο (registry), τροποποιούν τις διεπαφές προγραμματισμού εφαρμογών του λειτουργικού συστήματος (*Windows API*) κτλ. Τα *rootkit* s γίνονται πιο επικίνδυνα αλλά και πιο δύσκολα ανιχνεύσιμα στην περίπτωση που τρέχουν στο επίπεδο του πυρήνα του λειτουργικού συστήματος (*kernel*). Αυτό γιατί στην ουσία τρέχουν στο ίδιο επίπεδο με το λειτουργικό σύστημα. Στα Windows μπορεί να τρέχουν σαν οδηγοί συσκευών (*device drivers*) και στο *Linux* σαν '*loadable kernel modules'*.

Πρόσφατα παρατηρείται μια αύξηση στη χρήση *rootkit* τεχνικών στα κακόβουλα *bot*s. Το *Strom bot* για παράδειγμα χρησιμοποιεί ένα *rootkit* το οποίο προσπαθεί να απενεργοποιήσει τα προγράμματα προστασίας από ιούς και να αρνηθεί την εκκίνηση διεργασιών που σχετίζονται με τα προγράμματα αυτά. Ο εντοπισμός του τουλάχιστο στο ίδιο το μηχάνημα είναι ιδιαίτερα δύσκολος, αλλά μπορεί να ανιχνευθεί από τη δικτυακή κίνηση που δημιουργεί προς άλλα μηχανήματα του δικτύου.

Κεφάλαιο 5. Εργαλεία ανίχνευσης

Είδαμε ότι οι χρήσεις των *botnets* είναι πολλαπλές, το ίδιο πολλαπλές και σύνθετες είναι και οι επιθέσεις που διεξάγουν. Καμία τεχνολογία από μόνη της δεν μπορεί να ανακόψει όλες τις επιθέσεις και να μας προστατέψει από τα *botnets*. Για παράδειγμα ο στόχος της κατανεμημένης επίθεσης άρνησης υπηρεσίας (*DDoS*) είναι να βγάλει εκτός υπηρεσίας κάποιον εξυπηρετητή. Ο στόχος της επίθεσης ηλεκτρονικού ψαρέματος (*phishing*) είναι να προσελκύσει τους χρήστες να επισκεφτούν ένα πλαστό ιστοχώρο (*spoofed website*) και να τους κάνει να αποκαλύψουν τα προσωπικά τους δεδομένα. Ο στόχος του κακόβουλου λογισμικού που ένα *bot* μπορεί να ενσωματώσει ποικίλει, από τη συλλογή προσωπικών δεδομένων σε ένα υπονομευμένο υπολογιστή ως την εμφάνιση διαφημιστικών (*ads*) και την αποστολή ανεπιθύμητης αλληλογραφίας από αυτό. Μια σε βάθος προσέγγιση άμυνας είναι ουσιαστική για την ανίχνευση και το μετριασμό των αρνητικών επιδράσεων από τα *botnets*.

Το παραδοσιακό φιλτράρισμα πακέτων (*packet filtering*) και οι τεχνικές που βασίζονται στις πόρτες επικοινωνίας και τις υπογραφές (*signatures*) δεν μπορούν αποτελεσματικά να αντιμετωπίσουν τα *botnets* τα οποία δυναμικά και ταχύτατα αλλάζουν τον κώδικα εκμετάλλευσης ευπαθειών (*exploit code*), αλλάζουν το κανάλι επικοινωνίας (*control channel*) και καταφεύγουν στη ‘μεταπήδηση πορτών’ (*port hopping*) ή χρησιμοποιούν κοινές πόρτες όπως οι 80 (*HTTP*) και 443 (*HTTP*s) και χρησιμοποιούν τους υπονομευμένους υπολογιστές για διαφορετική κάθε φορά χρήση [28].

Στις μέρες μας, μια ποικιλία ανοιχτού λογισμικού και εμπορικών εργαλείων χρησιμοποιείται για την ανίχνευση των *botnets*. Πολλά από αυτά χρησιμοποιούν κάποιες τεχνικές ανίχνευσης από αυτές που αναφέραμε στο αντίστοιχο κεφάλαιο, όπως η ανάλυση των δικτυακών ροών (π.χ. το *Cisco NetFlow*), η ανίχνευση βασισμένη στη συμπεριφορά, έχοντας πρώτα αποτυπώσει τη συμπεριφορά του δικτύου υπό κανονικές συνθήκες ώστε να αναγνωρίσουν μη κανονικά μοτίβα κίνησης όπως η *DDoS* κίνηση κτλ.

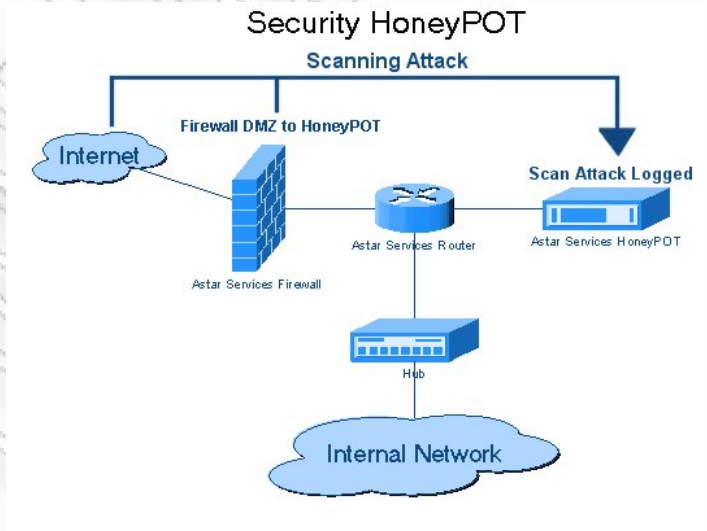
Στη συνέχεια θα αναφέρουμε τρία εργαλεία ανίχνευσης και μετρίασης των επιδράσεων από τα *botnets*, τα οποία και χρησιμοπούνται για την καταγραφή και

ανίχνευση κακόβουλων επιθέσεων. Πρόκειται για τα '*Honeypots*' τα οποία βοηθούν περισσότερο στην ανάλυση των επιθέσεων και θα αναλυθούν πλήρως στο επόμενο κεφάλαιο, το '*Snort*' και το '*BotHunter*' τα οποία έχουν δυνατότητες ανίχνευσης βασισμένες στις υπογραφές αλλά και τη συμπεριφορά.

5.1. *Honeypots*

Ένα *honeypot* είναι μία παγίδα η οποία στήνετε για να εντοπίσει, απωθήσει ή κατά κάποιο τρόπο αντιμετωπίσει προσπάθειες για μη εξουσιοδοτημένη χρήση υπολογιστικών συστημάτων. Γενικά, αποτελείτε από έναν Η/Υ ή μία δικτυακή τοποθεσία όπου εμφανίζετε σαν στοιχείο του δικτύου αλλά στην πραγματικότητα παρακολουθείτε, είναι απομονωμένο, είναι προστατευμένο, και φαίνεται να περιέχει πληροφορίες ή δεδομένα τα οποία έχουν αξία για τους εισβολείς.

Ένα *honeypot* είναι χρήσιμο σαν εποπτικό μέσο και σαν εργαλείο προειδοποίησης. Εφόσον συνήθως είναι ένας υπολογιστής, το *honeypot* μπορεί να έχει διάφορες μορφές όπως αρχεία, εγγραφές δεδομένων, ακόμη και τη μορφή διευθύνσεων IP που δεν χρησιμοποιούνται. Τα *Honeypots* μπορεί να αποτελέσουν κίνδυνο για ένα δίκτυο και για αυτό πρέπει να χειρίζονται προσεκτικά. Σε αντίθετη περίπτωση, ο εισβολέας μπορεί να τα χρησιμοποιήσει για να εισβάλει στο σύστημα.

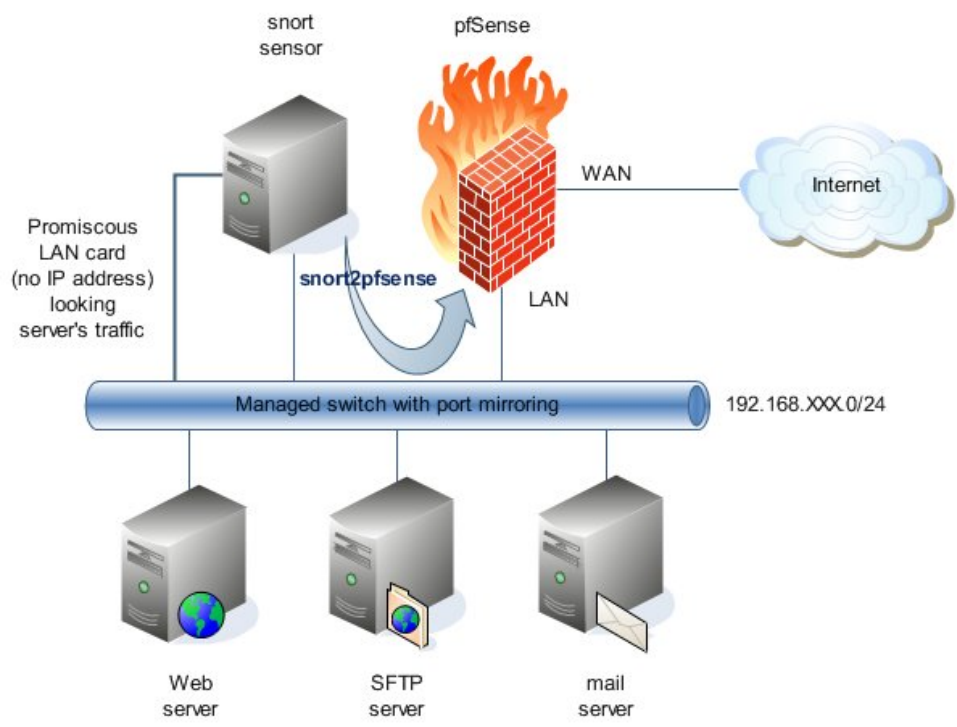


Σχήμα 5.1 Τυπικό Δίκτυο με *Honeypot*

5.2. Snort

Το Snort είναι ένα δωρεάν ανοιχτού τύπου λογισμικό που λειτουργεί ως σύστημα ανίχνευσης εισβολών (*Intrusion Detection System – IDS*) αλλά και ως σύστημα παρεμπόδισης εισβολών (*Intrusion Prevention System – IPS*). Μπορεί να κάνει καταγραφή πακέτων και ανάλυση κίνησης σε πραγματικό χρόνο σε δίκτυα *IP*. Μπορεί να εκτελεί ανάλυση πρωτοκόλλων, εύρεση και ταίριασμα (*matching*) περιεχομένου και συνήθως χρησιμοποιείται για να ανιχνεύσει παθητικά ή να μπλοκάρει ενεργά μια ποικιλία δικτυακών επιθέσεων και διερευνήσεων (*probes*), όπως υπερχειλίσεις buffer, κρυφή σάρωση πορτών (*stealth scan*), διερευνήσεις πρωτοκόλλου *SMB* (*Server Message Block*) και προσπάθειες αναγνώρισης (*fingerprinting*) του λειτουργικού συστήματος ανάμεσα σε πολλά άλλα χαρακτηριστικά [29].

Το *Snort* χαρακτηρίζεται ως μια «ελαφριά» (*lightweight*) τεχνολογία ανίχνευσης εισβολών σε σύγκριση με τα εμπορικώς διαθέσιμα αντίστοιχα συστήματα και είναι η πιο ευρέως αναπτυγμένη τεχνολογία ανίχνευσης και παρεμπόδισης εισβολών παγκοσμίως. Ο δημιουργός του είναι ο Martin Roesch και ο κώδικας του είναι γραμμένος στη γλώσσα C.



Σχήμα 5.2 Τοπικό Δίκτυο με Εγκατεστημένο Snort *Server*

5.2.1 Τεχνικές ανίχνευσης στο Snort

Αναλυτικότερα το Snort, χρησιμοποιεί για την ανίχνευση μια γλώσσα καθοδηγούμενη από κανόνες (*rule-driven*) η οποία συνδυάζει τα πλεονεκτήματα των μεθόδων ανίχνευσης βασιζόμενων στις υπογραφές (*signatures*), στην ανάλυση των πρωτοκόλλων για εύρεση ανωμαλιών, αλλά και γενικότερα στην ανώμαλη συμπεριφορά ως νεότερη τεχνική. Οι μηχανισμοί για τον εντοπισμό ανώμαλης συμπεριφοράς υλοποιούνται κατά κύριο λόγο από κάποιες μονάδες που ονομάζονται preprocessors και περιγράφονται πιο κάτω, παρόλο που δίνεται η δυνατότητα να εκφραστεί η ανώμαλη συμπεριφορά και μέσα από ένα κανόνα για την ανάλυση ενός πακέτου ή μιας ροής. Οι κανόνες (*rules*) περιγράφουν τα χαρακτηριστικά ενός πακέτου που μπορεί να είναι μέρος μια γνωστής επίθεσης. Κάθε ένας από τους κανόνες ουσιαστικά περιγράφει ποια είναι η «εικόνα» ενός βλαβερού πακέτου και επίσης πως πρέπει το Snort να αντιδράσει όταν εντοπίσει κάποια υπογραφή σε κάποιο πακέτο. Το *Snort* περιλαμβάνει πάνω από 6.000 κανόνες.

Οι κανόνες και οι υπογραφές συχνά χρησιμοποιούνται σαν συνώνυμες λέξεις. Οι διαφορές τους είναι οι εξής:

Οι υπογραφές είναι τα ειδικά χαρακτηριστικά του πακέτου που το χαρακτηρίζουν σαν ύποπτο ή βλαβερό (*malicious*). Τα χαρακτηριστικά αυτά βρίσκονται στο payload ή στο *header* του πακέτου και είναι μοτίβα από συμβολοσειρές (*string patterns*) που χαρακτηρίζονται σαν υπογραφή (*signature*) ενός «κακού» πακέτου. Γενικά η περιγραφή ενός πακέτου που είναι «κακό», όταν γίνεται με μια υπογραφή είναι στατική. Δηλαδή μια υπογραφή περιγράφει κάποιο υπαρκτό χαρακτηριστικό στο *payload* ή στο *header* του πακέτου.

Οι κανόνες περιγράφουν στο *Snort* ή άλλο *IDS* τα χαρακτηριστικά ενός πακέτου που μπορεί να είναι μέρος μίας γνωστής επίθεσης καθώς και την ενέργεια που θα εκτελεστεί κατά τον εντοπισμό του. Η περιγραφή ενός πακέτου με ένα κανόνα είναι αρκετά πιο δυναμική. Αφενός, σε ένα κανόνα μπορεί να περιγράφονται περισσότερα του ενός υπαρκτά χαρακτηριστικά στο *payload*, αφετέρου, μπορούν να περιγράφονται χαρακτηριστικά που δεν πρέπει να έχει ένα πακέτο για να θεωρηθεί ύποπτο. Τέλος, ένας

κανόνας μπορεί να περιγράφει μια ολόκληρη ροή και όχι ένα πακέτο, στις περιπτώσεις που γίνεται ανίχνευση εισβολών κρατώντας την 'κατάσταση' (*state*) των συνδέσεων [30].

Παράδειγμα ενός κανόνα για την ανίχνευση του δούρειου ίππου (*trojan*) *SubSeven* είναι το ακόλουθο:

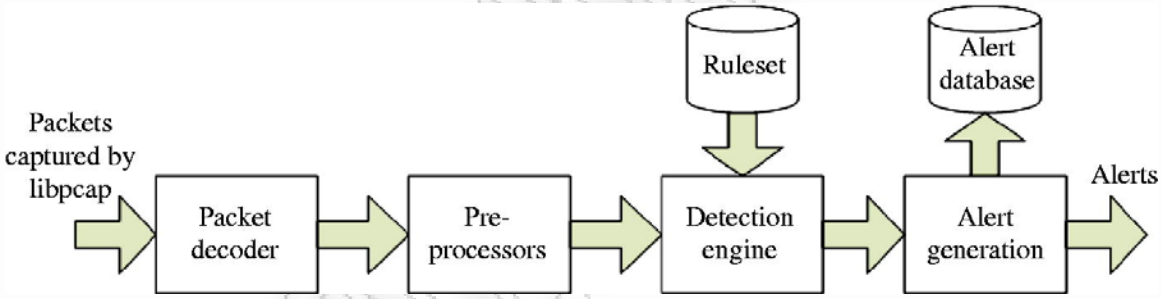
```
alert tcp $EXTERNAL_NET 27374 -> $HOME_NET any (msg:"BACKDOOR subseven
22"; flags: A+; content: "|0d0a5b52504c5d3030320d0a|"; reference:arachnIDS,485;
reference:url,www.hackfix.org/subseven/; sid:103; classtype:misc-activity; rev:4;)
```

Το *Snort* επιτρέπει την ενσωμάτωση ξεχωριστών υποπρογραμμάτων στον κώδικά του με κομμάτια κώδικα που ονομάζονται '*plug-ins*'. Τα *plug-ins* επιτρέπουν την επέκταση των δυνατοτήτων του *Snort* χωρίς να χρειάζεται αλλαγή το κύριο σώμα του κώδικά του. Τα *plug-ins* έχουν σαν κύριο στόχο να επεκτείνουν τις δυνατότητες του *Snort*. Τα *preprocessor plug-ins* για παράδειγμα προσθέτουν δυνατότητες ανίχνευσης ανώμαλης συμπεριφοράς. Οι *preprocessors* είναι σαν μία συμπληρωματική μηχανή του *Snort* όπου χωρίς αυτούς ορισμένα είδη επιθέσεων δεν θα μπορούσαν να εντοπιστούν. Οι δυνατότητες που προσφέρουν οι *preprocessors* συνοψίζονται παρακάτω:

- Αναπτύσσονται σαν '*plug-ins*' για να δίνουν στο *Snort* ευελιξία και επεκτασιμότητα. Και φυσικά για να μπορεί το *Snort* να ρυθμίζεται ανάλογα με τις ανάγκες του περιβάλλοντος δικτύου που θα χρησιμοποιηθεί.
- Δίνουν την δυνατότητα στο *Snort* να χειρίζεται δεδομένα που μοιράζονται σε πάνω από ένα πακέτα όπως τα TCP *streams* ανασυνθέτοντάς τα.
- Χρησιμοποιούνται στο Snort για να κανονικοποιούν τα δεδομένα που περιγράφονται με πολλαπλούς τρόπους ('*HTTP_decode preprocessor'*). Για παράδειγμα αν ένα *HTTP request* μπορεί να είναι σε *Unicode* ή σε *ASCII*, τότε ο αντίστοιχος preprocessor θα παράγει μια έξοδο (*output*) από *Unicode* σε ASCII, ώστε κανόνες που περιγράφονται με ASCII να μπορούν να εφαρμοστούν και σε αιτήματα (*requests*) σε *Unicode*.

- Δίνουν την δυνατότητα στο Snort να εφαρμόζει μεθόδους εντοπισμού που δεν μπορούν να εκφραστούν ακόμα και με τους πιο ευέλικτους κανόνες (π.χ. ταίριασμα ανώμαλης συμπεριφοράς με *regular expressions*. Για παράδειγμα ο '*portscan preprocessor'* που εντοπίζει τις σαρώσεις πορτών βασιζόμενος στην ανώμαλη συμπεριφορά).
- Οι *Preprocessors* του *Snort* μπορούν να προσφέρουν την δυνατότητα στο Snort να εντοπίζει κάποιες επιθέσεις που δεν έχουν γίνει ακόμα κανόνες.

Σκοπός μας δεν είναι να αναλύσουμε τον τρόπο λειτουργίας του *Snort*, αλλά να αναφερθούμε σε εκείνα τα χαρακτηριστικά του, που επιτρέπουν την κατανόηση των τεχνικών ανίχνευσης που χρησιμοποιεί όπως οι τεχνικές με βάση τις υπογραφές και οι τεχνικές με βάση την ανώμαλη συμπεριφορά. Ακολουθεί μια γενική εικόνα της ροής δεδομένων στο *Snort*.



Σχήμα 5.3 Διάγραμμα Ροής Δεδομένων Snort

5.3 BotHunter

Το *BotHunter* είναι ένα εργαλείο παθητικής παρακολούθησης ενός δικτύου σχεδιασμένο να αναγνωρίζει τα πρότυπα (*patterns*) επικοινωνίας των μολυσμένων με κακόβουλο λογισμικό υπολογιστών μέσα στην περίμετρο του δικτύου μας. Χρησιμοποιώντας μια προχωρημένη μηχανή συσχέτισης μολυσματικών γεγονότων βασισμένων στο διάλογο (*infection-dialog-based*) – εν αναμονή κατοχύρωσης με

πατέντα – το *BotHunter* αποτελεί το πιο εμπεριστατωμένο δικτυακό σύστημα διάγνωσης μολύνσεων από κακόβουλο λογισμικό που είναι διαθέσιμο σήμερα [31].

Το *BotHunter* είναι μια εφαρμογή σχεδιασμένη να παρακολουθεί τις διπλής κατεύθυνσης ροές επικοινωνίας ανάμεσα στους εσωτερικούς πόρους και τις εξωτερικές οντότητες, αναπτύσσοντας στοιχεία-ίχνη από ανταλλαγές δεδομένων που ταιριάζουν με μοντέλα ακολουθιών μολυσματικών γεγονότων, κρατώντας κατάσταση (*state*) για τις συνδέσεις δικτύου. Αποτελείται από μια μηχανή συσχέτισης οδηγούμενη από μια προσαρμοσμένη και επαυξημένη έκδοση του Snort 2, η οποία ανιχνεύει τις συνεπαγόμενες ενέργειες που συμβαίνουν κατά τη διάρκεια της διαδικασίας μόλυνσης με κακόβουλο λογισμικό. Οι ενέργειες αυτές περιλαμβάνουν την εισερχόμενη σάρωση (*inbound scanning*), την εκμετάλλευση ευπαθειών (*exploit*s), τη μεταφόρτωση εκτελέσιμων αρχείων (*egg download*), τον εξερχόμενο διάλογο συντονισμού των *bot*s, (*outbound dialog*), την εξερχόμενη εξάπλωση των επιθέσεων και την κακόβουλη επικοινωνία σε δίκτυα ομοτίμων (P2P). Το *BotHunter* συσχετίζει τους συναγερμούς εισβολής με πρότυπα εξερχόμενης επικοινωνίας, τα οποία αποτελούν υψηλές ενδείξεις μιας επιτυχούς μόλυνσης κάποιου εσωτερικού υπολογιστή. Όταν μια σειρά από στοιχεία βρίσκονται να ταιριάζουν στο μοντέλο μόλυνσης διαλόγου του *BotHunter*, μια συγκεντρωτική αναφορά παράγεται για να καταγράψει όλα τα σχετικά γεγονότα και τις πηγές των γεγονότων που έπαιξαν ρόλο κατά τη διάρκεια της διαδικασίας μόλυνσης. Αυτή η αναλυτική στρατηγική ταιριάσματος των ροών διαλόγων ανάμεσα στους εσωτερικούς πόρους του δικτύου μας και το ευρύτερο διαδίκτυο αναφέρεται ως συσχέτιση βασισμένη στο διάλογο (*dialog-based correlation*) [31].

Η διαφορά του *BotHunter* από τα παραδοσιακά συστήματα ανίχνευσης εισβολών (*IDSs*) έγκειται στο ότι τα τελευταία τυπικά εστιάζουν στις εισερχόμενες ροές πακέτων για σημάδια προσπαθειών εισβολής. Τα συστήματα ανίχνευσης εισβολών έχουν την ικανότητα να ανιχνεύουν αρχικές προσπάθειες εισερχόμενων εισβολών και η συχνότητα με την οποία παράγουν σχετικούς συναγερμούς (*alarms*) σε λειτουργικά δίκτυα είναι καλά τεκμηριωμένη. Όμως η ικανότητα να διακριθεί μια «επιτυχής» μόλυνση ενός τοπικού υπολογιστή από αναρίθμητες καθημερινές σαρώσεις (*scans*) και προσπάθειες εισβολής είναι πολύ σημαντική για την άμυνα των δικτύων και σ’ αυτό τον τομέα έρχεται να συμβάλλει το *BotHunter*.

5.3.1 Ανάλυση δικτυακών ροών στο *Bot*Hunter

Το *BotHunter* μοντελοποιεί τα στοιχεία μόλυνσης ως μια σύνθεση συμμετεχόντων και μια χαλαρά ταξινομημένη σειρά από ανταλλαγές διαλόγων στο δίκτυο:

Infection I = <A, V, E, C, P, V', {D}>

Όπου : **A** = attacker (επιτιθέμενος)

V = victim (θύμα)

E = egg download location (τοποθεσία μεταφόρτωσης εκτελέσιμου αρχείου)

C = *C&C server* (εξυπηρετητής '*Command and Control*')

P = *peer to peer* coordination points (σημεία συντονισμού ομοτίμων)

V' = the victim's next propagation targets (ο επόμενος στόχος του θύματος – εξάπλωση επίθεσης)

{D} = αναπαριστά ένα σύνολο από αλληλουχίες διαλόγων που αποτελούνται από διπλής κατεύθυνσης ροές που διαπερνούν τα όρια εξόδου του δικτύου μας.

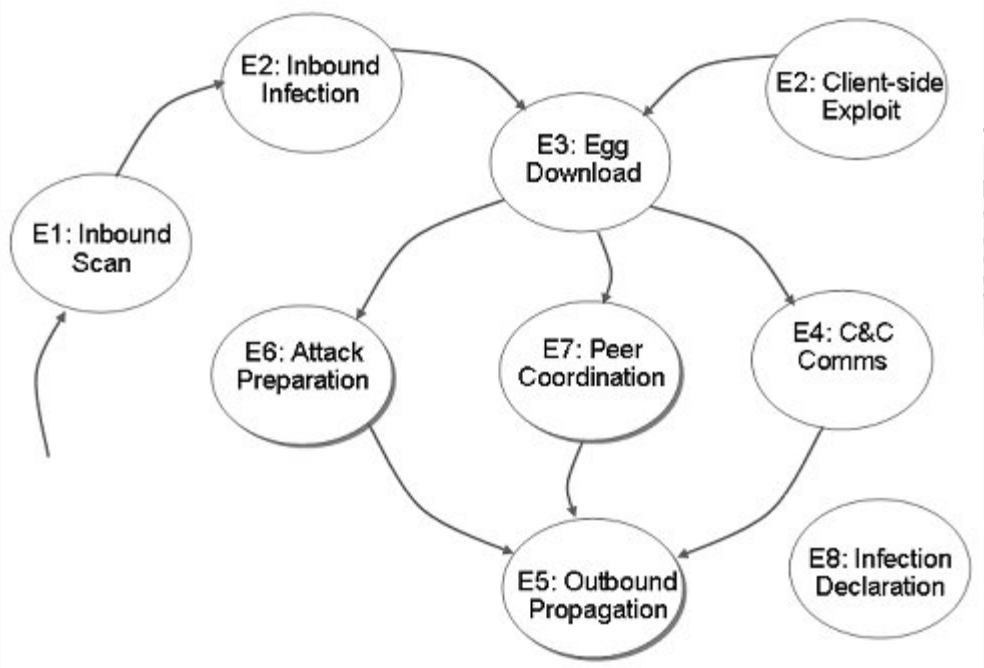
Το τρέχον σύνολο {D} με τους διαλόγους μόλυνσης παρέχει κάλυψη ανίχνευσης για τα ακόλουθα οκτώ γεγονότα-αποδείξεις (*Evidence*):

- ***E1***: Εισερχόμενη κακόβουλη σάρωση πορτών
- ***E2***: Εισερχόμενη και εξερχόμενη ανίχνευση εκμετάλλευσης ευπαθειών (μολύνσεις πελάτη μέσω web, εκμετάλλευση ευπαθειών Microsoft – RPC, Netbios, επιθέσεις OP/Shell κώδικα μέσω υπερχειλίσεων – εκμετάλλευση ειδικών πορτών, εκμετάλλευση εφαρμογών υψηλών πορτών, επιθέσεις σε προγράμματα περιήγησης, εξερχόμενη αλληλογραφία από μη-SMTP εξυπηρετητή, ανίχνευση εξωτερικών σαρώσεων)
- ***E3***: Εξαναγκασμένη μεταφόρτωση / ανίχνευση εγκατάστασης παράνομου λογισμικού (αίτηση μεταφόρτωσης από δούρειο ίππο / κακόβουλο λογισμικό, εντοπισμός στιγματισμένων δυαδικών αρχείων, επικοινωνία με κακόβουλο FTP εξυπηρετητή, μεταφόρτωση / εγκατάσταση κατασκοπευτικού λογισμικού μέσω web)
- ***E4***: Ανίχνευση επικοινωνίας '*Command and Control*' (περιοδικές συνδέσεις κατασκοπευτικού λογισμικού σε τοποθεσίες ελέγχου, αναφορές επιτυχούς εγκατάστασης κακόβουλου λογισμικού μέσω *web*, ανίχνευση εντολών

κατασκοπευτικού λογισμικού σε εγκαταστημένες συνδέσεις, συνδέσεις διαφημιστικού λογισμικού σε τοποθεσίες ελέγχου, αναγνώριση εισόδων/διαλόγων/εντολών που αντιστοιχούν σε *botnet C&C*, περιοδική επικοινωνία δούρειων ίππων κυρίως μέσω πορτών web, αναφορές επιτυχούς εισόδου/εγκατάστασης από εφαρμογές σε πόρτες δικτύου, επιστροφές κλήσεων βασισμένες στο *DNS*, επιστροφές κλήσεων βασισμένων στο SMTP για μη-SMTP υπολογιστές, ανίχνευση καναλιών ελέγχου από *IRC botnets)*.

- ***E5***/***E6***: Προετοιμασία επιθέσεων από εσωτερικούς υπονομευμένους υπολογιστές (αναζήτηση εγγραφών *DNS* τύπου *MX – spambots,* ερωτήματα DNS συνδεδεμένα με κακόβουλο λογισμικό)
- **E7**: Κανόνες επικοινωνίας υπολογιστών σε δίκτυα ομοτίμων (δραστηριότητα *P2P botnets*)
- **E8**: Κανόνες δήλωσης μολύνσεων από κακόβουλο λογισμικό (γνωστές διευθύνσεις IP που αντιστοιχούν σε *botnets*, διευθύνσεις του "Ρωσικού *business* δικτύου", ανίχνευση εξερχόμενων κακόβουλων σαρώσεων για εξάπλωση των μολύνσεων)

Τα μολυσματικά γεγονότα ή αλλιώς γεγονότα επιθέσεων ακολουθούν μια λογική σειρά η οποία αποτελεί τον κύκλο ζωής των *botnets* στο *BotHunter*. Αν και η σειρά αυτή δεν τηρείται πάντα στις ανιχνεύσεις κακόβουλων επιθέσεων του *Bot*Hunter θεωρείται ότι τα γεγονότα επιθέσεων συμβαίνουν με συγκεκριμένη σειρά. Οι δημιουργοί του *BotHunter* υποστηρίζουν ότι η ανάλυση της σειράς των διαλόγων από τα *bot*s πρέπει να είναι ανθεκτική στην απουσία κάποιων γεγονότων διαλόγου και δεν απαιτούν αυστηρή σειρά στα γεγονότα των διαλόγων. Εξάλλου, είναι πιθανό κάποια προαπαιτούμενα γεγονότα να μην ανιχνευθούν, είτε λόγω αδυναμίας ανάλυσης σε περιπτώσεις αυξημένου φόρτου στο δίκτυο, είτε επειδή οι επιθέσεις είναι καινούριες και άρα μη ανιχνεύσιμες (π.χ. η εκμετάλλευση μιας καινούριας ευπάθειας στο λογισμικό – *zero day attack*).



Σχήμα 5.4 Κύκλος Ζωής των *Botnet* στο *Bot*Hunter

Όπως φαίνεται από το γράφημα, το *BotHunter* ενσωματώνει αρχική ανίχνευση σαρώσεων και εκμεταλλεύσεων ευπαθειών, περιλαμβάνοντας τις μολύνσεις υπολογιστών-πελατών μέσω του *web* (π.χ. ευπάθειες στον πρόγραμμα περιήγησης). Η μόλυνση των υπολογιστών μετά ακολουθείται από τη μεταφόρτωση δυαδικών αρχείων, την εγκατάσταση και το συντονισμό (στην περίπτωση των *botnets*, μολύνσεις από κατασκοπευτικό ή διαφημιστικό λογισμικό). Στη συνέχεια, το μοντέλο μολύνσεων προχωρά με την εξάπλωση-διάδοση των μολύνσεων οι οποίες περιλαμβάνουν δραστηριότητα όπως οι εξωτερικές σαρώσεις, οι εκμεταλλεύσεις ευπαθειών, η αποστολή ανεπίκλητων ηλεκτρονικών-μηνυμάτων (*spam*) και η προετοιμασία επιθέσεων. Τέλος, το *Bot*Hunter περιλαμβάνει την ικανότητα να αναγνωρίζει κακόβουλες μολύνσεις όταν τα εσωτερικά συστήματα παρατηρούνται να προσπαθούν να συνδεθούν σε γνωστούς *C&C* εξυπηρετητές ή άλλες διευθύνσεις οι οποίες σχετίζονται με τον έλεγχο κακόβουλου λογισμικού (όπως το "Ρωσικό *business* δίκτυο" - RBN).

5.3.2 Ανίχνευση *bot*s στο BotHunter

Το *BotHunter* δίνει τη δυνατότητα καταγραφής των ανιχνεύσιμων μολυσματικών γεγονότων - που αναφέρθηκαν στη προηγούμενη ενότητα - και στην περίπτωση ανίχνευσης συγκεκριμένων συνδυασμών γεγονότων επιθέσεων από και προς κάποιο εσωτερικό πόρο του δικτύου μας συμπεραίνει την ύπαρξη προγραμμάτων ρομπότ (*bot*s) στο δίκτυό μας. Αυτό το πετυχαίνει κάνοντας συσχέτιση των μολυσματικών διαλόγων που παρατηρούνται ανάμεσα σε κάθε ξεχωριστό υπολογιστή του προστατευόμενου δικτύου και τους έξω από το δίκτυό μας υπολογιστές. Με άλλα λόγια το *BotHunter* αναζητεί τη σύνδεση εξερχόμενων κακών προτύπων επικοινωνίας με παρατηρούμενη εισερχόμενη δραστηριότητα εισβολών.

Επιπλέον, το *BotHunter* χρησιμοποιεί τη μηχανή του λογισμικού Snort για την ανίχνευση εισβολών, το οποίο και έχει εμπλουτίσει με κανόνες (rules) για την ανίχνευση εντολών ελέγχου και επιθέσεων που παρατηρούνται στα *botnets*. Οι συνδυασμοί γεγονότων που οδηγούν το *BotHunter* στην δήλωση ύπαρξης *bot* είναι οι τρεις ακόλουθοι:

1. Μόλυνση τοπικού υπολογιστή (γεγονότα ***E2*** ή ***E3***) και ανίχνευση εξερχόμενης κίνησης που αφορά το συντονισμό *bot*s ή την εξάπλωση των επιθέσεων (γεγονότα ***E4*** έως ***E7***).
2. Τουλάχιστο δύο διακριτά σημάδια από εξερχόμενη κίνηση συντονισμού ή κίνηση εξάπλωσης των επιθέσεων (γεγονότα ***E4*** έως ***E7***).
3. Εγκαθίδρυση επικοινωνίας με επιβεβαιωμένους κακόβουλους υπολογιστές ελέγχου *botnets* ή ιστοχώρων μεταφόρτωσης κακόβουλου λογισμικού (γεγονός ***E8***).

Η ανίχνευση *bot* στο προστατευόμενο από το *BotHunter* δίκτυο, έχει ως συνέπεια την παραγωγή συναγερμών και την ταυτόχρονη δημιουργία του προφίλ της μόλυνσης το οποίο περιλαμβάνει τα ακόλουθα στοιχεία:

- Το υπολογιζόμενο σκορ της μόλυνσης (στο εύρος από 0,8 έως 3,8). Υψηλό σκορ σημαίνει την ύπαρξη ισχυρότερων στοιχείων που οδήγησαν στην δημιουργία του προφίλ μόλυνσης
- Το χρόνο έναρξης και τη διάρκεια της επίθεσης

- Τη διεύθυνση *IP* του υπολογιστή θύματος
- Τη διεύθυνση *IP* του υπολογιστή ελέγχου *C&C* (σε περίπτωση ανίχνευσης κίνησης *C&C*)
- Τις διευθύνσεις *IP* των υπολογιστών στόχων (εξάπλωση μόλυνσης)
- Τα παρατηρούμενα γεγονότα επιθέσεων με περισσότερα στοιχεία όπως οι πόρτες επικοινωνίας, ο τύπος των εκμεταλλευόμενων ευπαθειών, ο χρόνος που ανιχνεύθηκαν κτλ.

Κεφάλαιο 6. Τα Honeypots

Τα *honeypots* είναι σχετικά μια νέα και ραγδαία εξελισσόμενη τεχνολογία. Το πεδίο εφαρμογή τους είναι αρκετά ευρύ και για αυτό το λόγο είναι δύσκολος ο ορισμός τους. Δεν είναι λύση για ένα μόνο πρόβλημα όπως είναι τα *IDS* ή τα *IPS*. Μπορούν να χρησιμοποιηθούν για ανίχνευση, αποτροπή, εκμάθηση επιθέσεων και πολλές άλλες εφαρμογές ασφαλείας. Ως ο πιο αποδεκτός ορισμός έχει καθιερωθεί ο ακόλουθος:

Τα «*honeypots*» είναι ένα πληροφοριακό σύστημα του οποίου η αξία έγκειται στην παράνομη ή χωρίς εξουσιοδότηση χρήση αυτού.

Μια άλλη έννοια είναι αυτή των *honeynets* που είναι ουσιαστικά ένα δίκτυο από *honeypots*.

Τα *honeypots* στηρίζονται στην αρχή του ότι κανένας δεν πρέπει να χρησιμοποιεί ή να έχει την παραμικρή επικοινωνία με αυτά τα συστήματα. Δεν έχουν καμία πραγματική υπηρεσία οπότε οποιαδήποτε επικοινωνία με αυτά θεωρείται εξορισμού ύποπτη.

Για την εξέλιξη των *honeypots* το *The Honeynet Project* όρισε αρχικά τις εξής φάσεις:

Φάση Ι: 1999 – 2001

Μελέτη διαφόρων επιθέσεων χρησιμοποιώντας πρώτης γενιάς (*GenI*) Honeynets. Κύριος σκοπός αυτής της φάσης ήταν η αναγνώριση και η κατηγοριοποίηση όλων των απειλών σε τυπικές (*default*) εγκαταστάσεις λειτουργικών συστημάτων αλλά και εφαρμογών. Αυτή η φάση έχει πλέον ολοκληρωθεί.

Φάση ΙΙ: 2001 - 2003

Ο σκοπός αυτής της φάσης ήταν ο εμπλουτισμός της τεχνογνωσίας των *honeynets* αλλά και η εξέλιξη τους ως προς την ευκολία εγκατάστασης τους, την ικανότητα συλλογής πληροφοριών και απόκρυψης της πραγματικής τους ταυτότητας (δυσκολότερη ανίχνευση τους).

Σε αυτή την φάση αναπτύχθηκαν και τα εικονικά *honeynets* (*virtual honeynet*), εκείνα δηλαδή που τρέχουν σε ένα μόνο σύστημα. Άλλο βήμα ήταν η οργάνωση της έρευνας γύρω από τα *honeynets* σε όλο τον κόσμο. Και αυτή η φάση έχει πλέον ολοκληρωθεί.

Φάση ΙΙΙ: 2003 - 2004

Σε αυτή την φάση αναπτύχθηκε ένα «*bootable CDROM*» το οποίο ξεκινάει κάποιον υπολογιστή κατευθείαν σε *Honeynet Gateway* ή *Honeywall*. Αυτός ο ψηφιακός οπτικός δίσκος πληρεί όλα τα κριτήρια πού είχαν τεθεί από το *The Honeynet Project* για ικανοποιητικό έλεγχο και καταγραφή της δικτυακής κίνησης. Επίσης αναπτύχθηκε και η ικανότητα καταγραφής της κίνησης σε απομακρυσμένα συστήματα έτσι ώστε να είναι πλέον εύκολη η υλοποίηση και τον κατανεμημένων *Honeynets* (*distributed Honeynets*). Και αυτή η φάση έχει ολοκληρωθεί.

ΦΑΣΗ IV: 2004 – 2005

Στην τέταρτη φάση ο στόχος είναι η ανάπτυξη ενός κεντρικού συστήματος συλλογής πληροφοριών από κατανεμημένα honeynets. Ταυτόχρονα αναπτύσσεται και ένα user interface για την ανάλυση και αξιολόγηση αυτών των πληροφοριών.

6.1. Κατηγορίες *Honeypots*

Οπώς αναφεραμέ και στη εισαγωγή της παρουσας εργασίας τα honeypots μπορούν να διακριθούν σε δύο κατηγορίες, ανάλογα με τη λειτουργία τους:

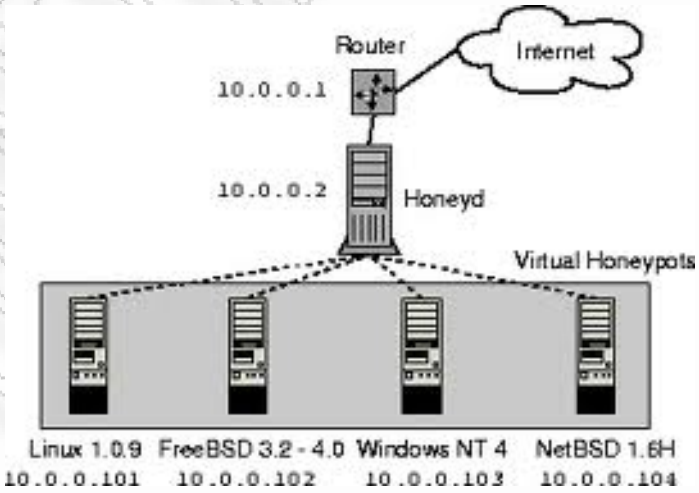
- Server ή «κλασσικά» honeypots
- Client honeypots ή honeyclients

Τα server και client honeypots, ανάλογα με τον τρόπο υλοποίησής τους, χωρίζονται σε δύο κατηγορίες:

- Χαμηλής αλληλεπίδρασης (low-interaction)
- Υψηλής αλληλεπίδρασης (high-interaction)

6.1.1 Χαμηλής Αλληλεπίδρασης Honeypots

Τα χαμηλής αλληλεπίδρασης *honeypots* (low interaction *honeypots*) που είναι συστήματα που εξομοιώνουν υπηρεσίες, αδυναμίες (vulnerabilities) και λειτουργικά συστήματα. Χρησιμοποιούν script-based languages για την περιγραφή της απόκρισης τους στις διάφορες ενέργειες του επιτιθέμενου. Εξαιτίας του ότι είναι εύκολη η εγκατάσταση τους και λόγω των περιορισμένων δυνατοτήτων τους θεωρούνται αρκετά ασφαλή. Είναι όμως πιο εύκολο να ανιχνευθούν και οι πληροφορίες που μπορούν να συλλέξουν είναι περιορισμένες. Το πιο γνωστό *honeypot* αυτής της κατηγορίας είναι το honeyd.



Σχήμα 6.1 Τυπικό Σύστημα *Honeypot* Χαμηλής Αλληλεπίδρασης με χρήση Honeyd

Παρακάτω θα δούμε μερικές υλοποιήσεις *honeypots* χαμηλής αλληλεπίδρασης.

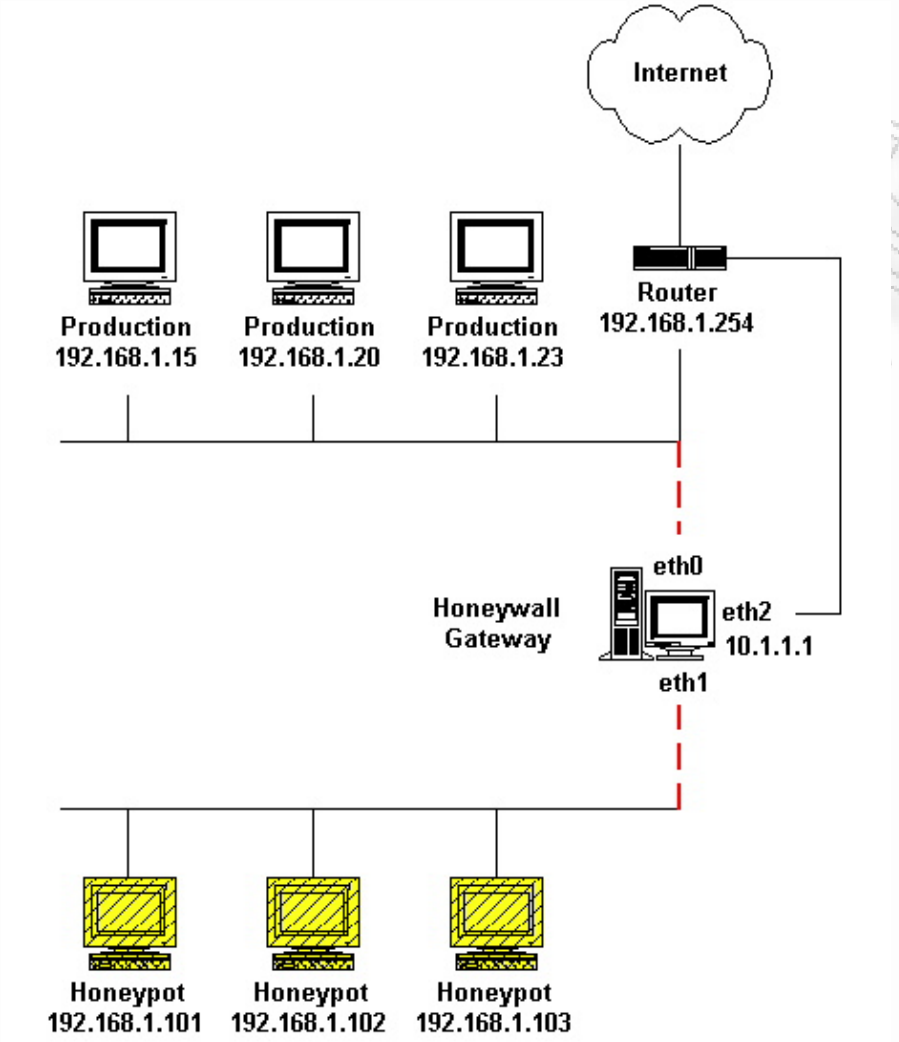
- **Honeyd:** Το *honeyd* λειτουργεί δημιουργώντας εικονικούς *hosts* σε ένα δίκτυο. Μπορεί να παραμετροποιηθεί έτσι ώστε να μιμείται λειτουργικά συστήματα και υπηρεσίες, δίνονταν έτσι τη δυνατότητα χρήσης πολλών *IP* διευθύνσεων μέσα από ένα μοναδικό *host*. Η διαδραστικότητα που προσφέρει όμως δεν είναι τόσο αυξημένη, με αποτέλεσμα να προδίδεται γρήγορα.

- **Nepenthes:** Χρησιμοποιείται για την αυτοματοποίηση της συλλογής κακόβουλου λογισμικού. Λειτουργεί μιμούμενο γνωστές ευπάθειες, ενώ ταυτόχρονα ανιχνεύει και αποθηκεύει το λογισμικό που προσπαθεί να τις κάνει *exploit*. Προσφέρει αρκετά μεγαλύτερη διάδραση απ' ότι το *honeyd*, με αποτέλεσμα να αποσπά μεγαλύτερο όγκο πληροφορίας.

- **Honeytrap:** Είναι ένα εργαλείο για την παρατήρηση νέων επιθέσεων σε υπηρεσίες δικτύου. Λειτουργεί εκκινώντας δυναμικούς servers ενώ περιλαμβάνει και εργαλεία για αυτόματη συλλογή malware και βασική ανάλυση των δεδομένων.

6.1.2 Υψηλής Αλληλεπίδρασης Honeypots

Τα υψηλής αλληλεπίδρασης *honeypots* (*high interaction honeypots*) που είναι συστήματα με πραγματικές υπηρεσίες και λειτουργικά συστήματα. Έχουν πραγματικές (όχι *virtual*) αδυναμίες και για αυτό τον λόγο θεωρούνται και τα πιο επικίνδυνα. Παρουσιάζουν αρκετές δυσκολίες στην εγκατάσταση και στην συντήρηση τους αλλά μπορούν να συλλέξουν πολύτιμες πληροφορίες με μεγάλη ακρίβεια και λεπτομέρεια. Λόγω του τρόπου υλοποίησης είναι πιο δύσκολη η ανίχνευση τους. Σε αυτήν την κατηγορία είναι και τα *honeynets* τα οποία επειδή ακριβώς είναι ένα δίκτυο από *honeypots* είναι συνήθως πραγματικά συστήματα για την όσο το δυνατόν μεγαλύτερη συλλογή πληροφοριών.



Σχήμα 6.2 Τυπικό Σύστημα *Honeypot* Υψηλής Αλληλεπίδρασης με Honeynet

❖ **Honeynets:** Ο όρος *honeynet* αναφέρεται σε ένα δίκτυο αποτελούμενο από ένα ή περισσότερα honeypots. Το πλεονέκτημα του *honeynet*, έναντι στο honeypot, έγκειται στο γεγονός ότι μπορεί να δώσει «τροφή» στον επιτιθέμενο να ασχοληθεί περισσότερο με τα honepots, αποσπώντας του, έτσι, ακόμα περισσότερες πληροφορίες.

❖ **Honeywall:** Για τη σωστή δημιουργία ενός *honeynet*, απαραίτητη προϋπόθεση είναι ένα κομβικό σημείο ελέγχου, μέσα από το οποίο θα διέρχεται το σύνολο της κίνησης, από και προς τα *honeypots*. Το *gateway* μηχάνημα που κάνει την παραπάνω δουλειά ονομάζεται *honeywall*. Ουσιαστικά αποτελεί ένα τείχος ασφαλείας αλλά και

εργαλείο καταγραφής-ελέγχου της εισερχόμενης και εξερχόμενης κίνησης, που χωρίζει το *honeynet*, από τον υπόλοιπο κόσμο. Χαρακτηριστικό του *honeywall* είναι η διαφανής λειτουργία του, πράγμα που σημαίνει ότι πρέπει να είναι αόρατο προς οποιονδήποτε, μέσα και έξω από το *honeynet*.

Σύγκριση High-interaction και Low-interaction Honeypots

High interaction	Low interaction
Πραγματικές υπηρεσίες/λειτουργικό	Προσομοίωση συστημάτων και υπηρεσιών
Υψηλότερο ρίσκο	Χαμηλότερο ρίσκο
Δύσκολη συντήρηση	Εύκολη εγκατάσταση και συντήρηση
Πιθανή εκμετάλλευση από τον επιτιθέμενο	Δεν παρέχεται πραγματικό root shell
Συγκέντρωση σημαντικής ποσότητας πληροφορίας/ κατανόηση των προθέσεων του επιτιθέμενου	Συγκέντρωση κυριώς στατιστικών στοιχείων

6.2. Υλοποιήσεις *Honeypots*

6.2.1 Φυσικά Honeypots

Φυσικά *honeypots* (*physical honeypots*), που είναι τα *honeypots* σε πραγματικά συστήματα με την δικιά τους *IP* διεύθυνση. Παρουσιάζουν αρκετές δυσκολίες στην εγκατάσταση και συντήρηση τους αφού απαιτούν την ολοκληρωμένη εγκατάσταση κάποιου λειτουργικού και όλων των μηχανισμών εκείνων που χρειάζονται για την διασφάλιση και την παρακολούθηση του.

6.2.2 Εικονικά Honeypots

Εικονικά *honeypots* (*virtual honeypots*), που είναι τα *honeypots* που τρέχουν από μηχανές που εξομοιώνουν την λειτουργία τους. Είναι κατάλληλα για υλοποίηση *honeypots* σε ένα μεγάλο εύρος διευθύνσεων *IPs* και είναι πιο απλά και οικονομικά στην εγκατάσταση και συντήρηση τους από τα φυσικά *honeypots*. Είναι όμως πιο εύκολα στην ανίχνευση τους. Σε αυτήν την κατηγορία μπορεί να υπάρξουν και τα *honeynets* τα οποία εξομοιώνουν όλα τα φυσικά τους μέρη σε ένα υπολογιστή. Σε αυτή την περίπτωση τα *honeynets* αυτά ονομάζονται *virtual honeynets*. Αυτή η κατηγορία προσφέρει περιορισμούς στο ποια λειτουργικά μπορούν να εξομοιωθούν (για παράδειγμα είναι δύσκολη η εγκατάσταση κάποιου *Cisco router*) και υπάρχει ο κίνδυνος με μια παραβίαση του *host OS* να υπάρξουν ταυτόχρονα παραβιάσεις όλων των υπόλοιπων.

6.3 Παραλλαγές των *Honeypots*

Η ιδέα αυτή της δημιουργίας κάποιου δολώματος για τον επιτιθέμενο δημιουργεί συνεχώς καινούργια πεδία εφαρμογής. Έτσι κατά καιρούς εμφανίζονται καινούργια είδη *honeypots*. Παρακάτω θα δούμε μερικά από αυτά.

6.3.1 Honeytokens

Τα *honeytokens* είναι μια έννοια παρόμοια με την έννοια των *honeypots* εκτός του ότι δεν είναι σε υπολογιστικό σύστημα. Για παράδειγμα μπορεί να αναφερθεί η εισαγωγή ψεύτικων αριθμών πιστωτικών καρτών στην βάση δεδομένων που κρατάει αυτούς τους αριθμούς. Όταν υπάρξει παραβίαση του συστήματος και ο εισβολέας επιχειρήσει να πάρει την λίστα με τους αριθμούς των πιστωτικών καρτών, τότε θα πάρει μαζί και τους ψεύτικους αριθμούς. Αυτοί οι αριθμοί μπορούν να ανιχνευθούν στο δίκτυο και να σημάνει συναγερμός από ένα *Intrusion Detection System*.

6.3.2 Fakeap

Τα *fakeap* δημιουργούν χιλιάδες «ψεύτικα access points» για την παραπλάνηση αλλά και ανίχνευση των *Wardrivers* και των *Script Kiddie*.

6.3.3 LaBrea

Εξομοιώνει διακομιστές σε αχρησιμοποίητες *IPs* στο δίκτυο και προσπαθεί να επιβραδύνει τις όποιες συνδέσεις γίνονται με αυτό. Αυτό μπορεί να έχει για παράδειγμα ως αποτέλεσμα την επιβράδυνση της μόλυνσης από *worms*.

6.3.4 HoneyMonkeys

Πρόκειται για μια ιδέα της «Microsoft Research», η οποία επεκτείνει την έννοια των "*honeypots*" και τους δίνει μια δυναμική για ενεργό εντοπισμό και ανάλυση επιθέσεων που χρησιμοποιούν διακομιστές *HTTP* για την εκμετάλλευση (*exploit*) αδυναμιών ασφάλειας στα προγράμματα πλοήγησης των χρηστών του διαδικτύου. Οι συνηθέστερες επιθέσεις που ανιχνεύονται είναι επιθέσεις σε προγράμματα πλοήγησης που δεν έχουν ενημερωθεί και διορθωθεί (*unpatched*) αλλά ανιχνεύονται και επιθέσεις σε αδυναμίες ασφάλειας που μέχρι πρότινος ήταν άγνωστες (*zero-day exploits*). Ο τρόπος που λειτουργούν τα «*honeymonkeys*» είναι χρησιμοποιώντας μια αυτοματοποιημένη διαδικασία περιπολίας στο διαδίκτυο (*Automated Web Patrol*) να επισκέπτονται διάφορους ιστοτόπους. Ο υπολογιστής που προσομοιώνει κάποιον χρήστη που επισκέπτεται αυτούς τους ιστοτόπους είναι προστατευμένος με τον κατάλληλο τρόπο

έτσι ώστε οποιαδήποτε αλλαγή πραγματοποιείται στο λειτουργικό σύστημα ή οποιαδήποτε καινούργιο αρχείο εμφανίζεται σε χώρο έξω από τον χώρο κανονικής λειτουργίας του *browse* να καταγράφεται για περαιτέρω ανάλυση. Αυτές οι αλλαγές δεν θα έπρεπε να γίνουν σε καμία περίπτωση οπότε θεωρούνται κακόβουλες. Έτσι με αυτόν τον τρόπο δεν χρειάζεται η χρησιμοποίηση κάποιων υπογραφών ανίχνευσης των επιθέσεων (*signatures*) και έτσι αυτά τα συστήματα είναι ικανά να ανιχνεύσουν *zero-day exploit*. Όπως έχει αποδειχθεί από την μέχρι τώρα λειτουργία τους τα «honeymonkeys» είναι αποτελεσματικά και είναι χρήσιμο εργαλείο για την χαρτογράφηση των κακόβουλων ιστοτόπων και τον τρόπο λειτουργίας τους. Μάλιστα τον Ιούνιο του 2005 το σύστημα αυτό ανίχνευσε το πρώτο *zero-day exploit* στο *javaprxy.dll*.

6.4 Πλεονεκτήματα των *Honeypots*

Τα *honeypots*, όπως και κάθε άλλη τεχνολογία, έχει πλεονεκτήματα και μειονεκτήματα. Παρακάτω θα αναφέρουμε τα ποιο σημαντικά πλεονεκτήματα τους. Ξεκινώντας θα αναφέρουμε ότι τα *honeypots* δεν παράγουν μεγάλα αρχεία καταγραφής (*log files*). Έτσι για παράδειγμα οργανισμοί ή εταιρίες που είχαν τεράστιο όγκο από *alerts*, με τα *honeypots* θα έχουν πολύ λιγότερο αφού καταγράφουν μόνο την πληροφορία που σχετίζεται με αυτά με αποτέλεσμα να διευκολύνει το έργο της ανάλυσης και εξαγωγής συμπερασμάτων.

Επίσης τα *honeypots* μειώνουν τους ψεύτικους συναγερμούς. Μια από τις μεγαλύτερες προκλήσεις στα *IDS* είναι η μείωση των ψεύτικων συναγερμών. Τα *honeypots* εξαιτίας του ότι οποιαδήποτε κίνηση σε αυτά είναι εξορισμού ύποπτη καταφέρνουν την μείωση αυτή.

Ένα άλλο πλεονέκτημα των *honeypots*, είναι ότι μπορούν να ανιχνεύσουν καινούργιες ή άγνωστες επιθέσεις. Τα παραδοσιακά *IDS* δεν μπορούν να ανιχνεύσουν άγνωστες επιθέσεις. Αν δεν υπάρχει η συγκεκριμένη υπογραφή της επίθεσης δεν μπορούν και να την αναγνωρίσουν. Τα *honeypot* διαφοροποιούνται εξαιτίας του ότι οποιαδήποτε κίνηση σε αυτά είναι ύποπτη.

Ακόμα ένα θετικό στοιχείο των *honeypots* είναι ότι μπορούν να καταγράψουν κρυπτογραφημένες επιθέσεις. Όσο οι διάφοροι οργανισμοί και επιχειρήσεις εφαρμόζουν μεθόδους κρυπτογραφημένων επικοινωνιών (όπως *SSH*, *IPsec*, *SSL*) το πρόβλημα της ανίχνευσης των επιθέσεων από τα παραδοσιακά *IDS* μεγαλώνει. Τα *honeypots* αντιμετωπίζουν αυτό το πρόβλημα αφού οι κρυπτογραφημένες επιθέσεις σε αυτά αποκρυπτογραφούνται αφού τα *honeypots* είναι ο στόχος αυτών.

Τα *honeypots* λειτουργούν και με το IPv6. Ολοένα και περισσότερο οργανισμοί και εταιρίες υλοποιούν το IPv6. Τα *IDS* και τα *firewalls* έχουν πολλές ιδιαιτερότητες και δυσκολίες σχετικά με την υποστήριξη του IPv6.

Τέλος τα *honeypots* δεν απαιτούν ακριβά και απαιτητικά μηχανήματα. Ένας οικονομικός ηλεκτρονικός υπολογιστής μπορεί να παρακολουθεί χιλιάδες IP διευθύνσεις χωρίς κανένα πρόβλημα.

6.5 Μειονεκτήματα των *Honeypots*

Τα *honeypots* παρά τα τόσα καλά πλεονεκτήματα που έχουν, παρουσιάζουν και μειονεκτήματα τα οποία αναφέρονται παρακάτω.

Καταγράφουν μόνο ότι αλληλεπιδρά με αυτά και δεν ανιχνεύουν καμία άλλη κίνηση. Αυτό έχει ως συνέπεια να μην μπορεί να αντιληφθεί καμία επίθεση που γίνεται σε κάποιο άλλο υπολογιστή του δικτύου του.

Στα υψηλής αλληλεπίδρασης *honeypots* υπάρχει σημαντικός κίνδυνος, όπου κάποιος επιτιθέμενος μπορεί να κυριεύσει και να χρησιμοποιήσει αυτό το *honeypot* για επιθέσεις σε άλλους στόχους. Αν και υπάρχουν αρκετά μέτρα θωράκισης των *honeypots* ο κίνδυνος δεν μπορεί να εξαλειφθεί. Στα χαμηλής αλληλεπίδρασης *honeypots* αυτός ο κίνδυνος είναι μικρότερος αν και όχι μηδενικός. Γνωρίζοντας πλέον ο επιτιθέμενος ότι σε κάποιο συγκεκριμένο IP τρέχει το *honeyd* μπορούσε να συγκεντρώσει τις επιθέσεις του σε αυτό και να εκμεταλλευτεί ένα άλλο λάθος του ότι αυτό τρέχει με δικαιώματα διαχειριστή. Για την έγκαιρη ανίχνευση τυχόν μη επιθυμητής πρόσβασης στα *honeypots* είναι απαραίτητος ο πολλαπλός έλεγχος της εξερχόμενης κίνησης. Υπάρχουν κάποιες αρχιτεκτονικές (όπως θα αναφερθούν στην συνέχεια) που επιτρέπουν τον έλεγχο αυτής

της εξερχόμενης κίνησης αλλά δυστυχώς, ούτε αυτές μπορούν να εγγυηθούν απόλυτη ασφάλεια.

Όλη σχεδόν η αξία ενός *honeypot* εκμηδενίζεται μόλις γίνει αντιληπτό. Οι επιτιθέμενοι αν δεν το αγνοήσουν, σίγουρα δεν θα χρησιμοποιήσουν τα καλύτερα τους όπλα και γενικότερα όλες οι κινήσεις τους θα είναι παραπλανητικές.

Ακόμα , υπάρχει περίπτωση ο επιτιθέμενος να καταφέρει να απενεργοποιήσει τους μηχανισμούς καταγραφής και ελέγχου με τρόπο που ο διαχειριστής να μην το αντιληφθεί. Η προστασία από αυτόν τον κίνδυνο είναι η εφαρμογή πολλαπλών επιπέδων μηχανισμών καταγραφής και ελέγχου. Κάτι τέτοιο επιλύει το πρόβλημα της βλάβης του ενός σημείου (*single point of failure*) αλλά ο κίνδυνος ούτε και εδώ εκμηδενίζεται.

Τέλος ένας άλλος υπαρκτός κίνδυνος είναι η παραβίαση ενός *honeypot* αλλά όχι για την χρησιμοποίηση του για επίθεση σε κάποιο άλλο σύστημα αλλά για να ανεβάσει (*upload*) πειρατική μουσική, πειρατικές ταινίες, αριθμούς πιστωτικών καρτών κα. Αν αυτά ανιχνευθούν τότε θα τα χρεωθεί ο διαχειριστής ο οποίος πρέπει να αποδείξει ότι δεν είναι αυτός που τα ανέβασε.

6.6 Εφαρμογές των *Honeypots*

Γενικά τα *honeypots* μπορούν να χρησιμοποιηθούν όχι μόνο σε οικιακούς χρήστες αλλά και σε οργανισμούς, ινστιτούτα, ερευνητικά ιδρύματα, ιδιωτικές εταιρείες, κτλ. Παρακάτω θα παρουσιάσουμε πως μπορούν να βρουν εφαρμογή τα *honeypots*.

Η ποιο σημαντική εφαρμογή τους είναι ότι μπορούν να εμποδίσουν συγκεκριμένες επιθέσεις. Υπάρχουν αυτοματοποιημένες επιθέσεις από «σκουλήκια» (*worms*) οι οποίες σαρώνουν ένα εύρος διευθύνσεων ψάχνοντας για συστήματα με συγκεκριμένες αδυναμίες ασφαλείας (*security holes*). Όταν βρεθούν τέτοια συστήματα τα σκουλήκια (*worms*) επιτίθενται καταλαμβάνοντας το σύστημα αυτό και αφού το σκουλήκι αντιγραφθεί σε αυτό τότε συνεχίζει την αναζήτηση για άλλα τέτοια συστήματα. Τα *honeypots* μπορούν να μειώσουν την ταχύτητα εξάπλωσης αυτών των επιθέσεων με το να καθυστερήσουν όσο το δυνατόν περισσότερο το σάρωμα που κάνει το σκουλήκι (*worm*). Κάτι τέτοιο μπορεί να γίνει με το να ρυθμιστεί ένα χαμηλής αλληλεπίδρασης

honeypot να ακούει σε ένα μεγάλο αριθμό αχρησιμοποίητων IP διευθύνσεις του Lan και να απαντάει στις διάφορες σαρώσεις με *Window size of zero* στο TCP. Χαρακτηριστικό παράδειγμα τέτοιου συστήματος είναι το *LaBrea Tarpit* [32].

Τα *honeypots* μπορούν να αποτρέψουν και επιθέσεις από *hackers* (μη αυτοματοποιημένες). Μπορεί δηλαδή ένα *honeypot* σωστά εγκατεστημένο να μπερδέψει έναν *hacker* και να επιτεθεί σε αυτό και όχι στον πραγματικό υπολογιστή παραγωγής (*production server*).

Επίσης μπορούν να χρησιμοποιηθούν για την μείωση του *spamming*. Κάτι τέτοιο μπορεί να επιτευχθεί με την χρησιμοποίηση χαμηλής αλληλεπίδρασης *honeypots* ως *open mail relays* που να καταγράφουν τα συγκεκριμένα *emails* και στην συνέχεια να ενημερώνουν το *spam filter* του *email server*. Ένας παρόμοιος τρόπος καταπολέμησης της ανεπιθύμητης αλληλογραφίας παρουσιάζεται στην συνέχεια αυτής της εργασίας.

Ακόμα μπορούν να ανιχνεύσουν μια επίθεση. Επειδή, όπως έχει προαναφερθεί, κάθε δικτυακή κίνηση στα *honeypots* είναι εξορισμού ύποπτη τότε η ανίχνευση οποιασδήποτε κακόβουλης κίνησης ακόμη και επιθέσεων με άγνωστες τεχνικές είναι πρακτικά ανιχνεύσιμη.

Σε ένα δίκτυο ο διαχειριστής συστημάτων μπορεί να ανταποκριθεί πιο αποτελεσματικά σε μια ενδεχόμενη επίθεση αφού μέσω των *honeypots* (όταν γίνεται σε αυτά) μπορεί να προσδιορίσει την επίθεση που δέχεται, χωρίς να σταματήσει κάποιο *service* βγάζοντας το από το δίκτυο για να αναλύσει την επίθεση.

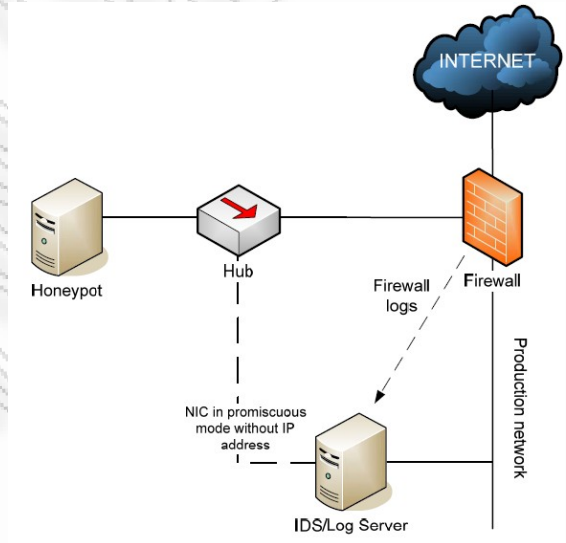
Τέλος μπορούν να χρησιμοποιηθούν ακόμα και για ερευνητικούς σκοπούς. Οι τεχνικές των *hackers* εξελίσσονται συνεχώς και γίνονται πιο καλές και πιο αποτελεσματικές. Το να γνωρίζει κανείς τις τεχνικές τους είναι αναγκαίο αν θέλει να προστατευθεί αποτελεσματικά. Ακριβώς σε αυτό το σημείο έρχονται τα *honeypots* τα οποία δίνουν μια λύση για την πιο στενή παρακολούθηση των καινούργιων τεχνικών επίθεσης.

6.7 Αρχιτεκτονική των *Honeypots*

Όπως έχει προαναφερθεί τα *honeynets* είναι ουσιαστικά ένα δίκτυο από «*honeypots*» κυρίως υψηλής αλληλεπίδρασης. Κατά συνέπεια η αρχιτεκτονική των *honeynets* είναι από τα πιο σημαντικά σημεία για την σωστή υλοποίηση τους. Χρειάζονται να πληρούνται κάποια κριτήρια για την επίτευξη όχι μόνο της σωστής και ακριβής καταγραφής (*data capture*) των διαφόρων κακόβουλων κινήσεων των επιτιθεμένων αλλά και της διασφάλισης των συστημάτων αυτών από την χρησιμοποίηση τους ως βάση για την επίθεση σε άλλους στόχους (*data control*). Επίσης ιδιαίτερη προσοχή θα πρέπει να δοθεί και στο σύστημα συλλογής πληροφοριών (*data collection*) από διαφορετικά *honeynets*. Τα κριτήρια και οι τεχνικές αυτές συνεχώς τελειοποιούνται και προσαρμόζονται στις καινούργιες τεχνικές των επιτιθεμένων, πάντα όμως η διασφάλιση έχει προτεραιότητα έναντι της καταγραφής.

6.7.1 Πρώτη Γενιά *Honeypots*

Η αρχιτεκτονική υλοποίησης της πρώτης γενιάς *honeynets* αναπτύχθηκε το 1999. Αν και αρκετά απλή μπορούσε να ανιχνεύσει και να καταγράψει επιθέσεις από «worms» και *script kiddies* (παιδιά χωρίς ιδιαίτερες γνώσεις που χρησιμοποιούν εργαλεία άλλων πιο έμπειρων, για να επιτεθούν).



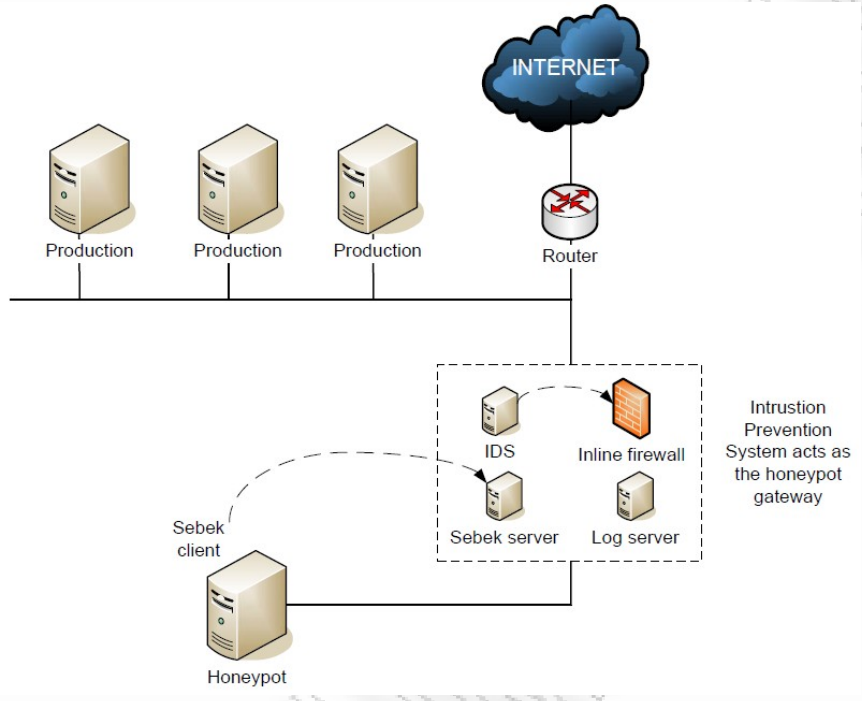
Σχήμα 6.3 Αρχιτεκτονική *Honeypot* Πρώτης Γενιάς

Η καταγραφή της κίνησης γίνονταν από ένα *IDS* χρησιμοποιώντας ένα σύστημα με δύο κάρτες δικτύου. Η μια κάρτα «άκουγε» το δίκτυο με τα *honeypots* χωρίς IP και το άλλο ήταν σε ένα άλλο δίκτυο με IP για τον απομακρυσμένο έλεγχο του συστήματος αυτού. Τα αρχεία καταγραφής συνήθως αποθηκεύονταν σε άλλο διακομιστή που ήταν στο προστατευόμενο υποδίκτυο. Ο έλεγχος της κίνησης για την αποτροπή χρησιμοποίησης των «*honeypots*» για την εξαπόλυση άλλων επιθέσεων γίνονταν από δύο διαφορετικά σημεία για περισσότερη ασφάλεια. Το ένα ήταν το *firewall* και το άλλο ένας δρομολογητής (*router*). Αυτά είχαν συγκεκριμένους κανόνες για αποτροπή κακόβουλης δικτυακής κίνησης από τα «*honeypots*» στο υπόλοιπο διαδίκτυο όπως ήταν ο περιορισμός συνδέσεων ανά πρωτόκολλο (π.χ. 5 συνδέσεις TCP την ώρα) και ο περιορισμός της ταχύτητας σύνδεσης (π.χ. 1Mbit). Είναι αυτονόητο ότι όσο πιο χαλαροί ήταν αυτοί οι κανόνες τόσο περισσότερη ζημιά μπορούσαν οι επιτιθέμενοι να προσκαλέσουν. Αντίθετα όσο πιο αυστηροί τόσες μεγαλύτερες πιθανότητες υπήρχαν για να κινήσουν υποψίες ότι κάτι περίεργο συμβαίνει. Σημαντικό μειονέκτημα αυτής της υλοποίησης *honeynets* ήταν το ότι ο έλεγχος της κίνησης γίνονταν από *firewall* που λειτουργούσε σε *layer 3* με αποτέλεσμα την σχετικά εύκολη ανίχνευση του από τους επιτιθέμενους. Ωστόσο τα συμπεράσματα, η τεχνογνωσία και η εμπειρία που αποκτήθηκε από αυτή την αρχιτεκτονική ήταν απαραίτητα για την περαιτέρω εξέλιξη και ωρίμανση αυτής της τεχνολογίας.

6.7.2 Δεύτερη Γενιά *Honeypots*

Το 2002 εμφανίζεται η δεύτερη γενιά «honeynets», όπου δίνεται βάρος σε τεχνικές αποτελεσματικότερου ελέγχου και καταγραφής της δικτυακής κίνησης και καλύτερης αντιμετώπισης των τεχνικών ανίχνευσης αυτών (αντίμετρα). Κύριο χαρακτηριστικό τους είναι η υλοποίηση ενός συστήματος ελέγχου κίνησης που λειτουργεί σε *Open System Interconnect* (OSI) *layer* 2. Αυτή η αλλαγή δυσχεραίνει την ανίχνευση αυτού του συστήματος που ονομάζεται *honeywall*, λόγω του ότι δεν υπάρχει μείωση του *Time to Live* (TTL) πεδίου της επικεφαλίδας του IP. Το σύστημα αυτό έχει και άλλη μια κάρτα

δικτύου η οποία «ακούει» σε ένα ασφαλές υποδίκτυο μέσω της οποίας μπορεί να γίνεται απομακρυσμένη διαχείριση.



Σχήμα 6.4 Αρχιτεκτονική *Honeypot* Δεύτερης Γενιάς

Το *honeywall* ελέγχει την κίνηση στο υποδίκτυο των «honeynets» με την χρησιμοποίηση εργαλείων όπως είναι τα *iptables* και καταγράφει την δικτυακή κίνηση μέσω των ενσωματωμένων του μηχανισμών όπως είναι για παράδειγμα τα *iptables logs* και το *Snort*. Το *honeywall* είναι υπεύθυνο για την αποτροπή επιθέσεων σε εξωτερικούς στόχους χρησιμοποιώντας ο επιτιθέμενος σαν βάση τα *honeypots*. Για παράδειγμα μπορεί να αναφερθεί η χρησιμοποίηση του *snort-inlin* είτε για αποτροπή των επιθέσεων σε άλλους στόχους είτε για την απλή αλλοίωση τους. Η τελευταία μέθοδος έχει το πλεονέκτημα ότι ενώ ο επιτιθέμενος νομίζει ότι κάνει σωστά τα βήματα για την εκμετάλλευση κάποιας αδυναμίας στην ουσία γίνεται μια αδιόρατη αλλαγή η οποία καταστρέφει όλη την επίθεση (π.χ. ενώ στέλνει την εντολή rm πηγαίνει η rmm). Έτσι αυτό που νομίζει ο επιτιθέμενος είναι ότι απλά δεν πέτυχε η επίθεση αυτή και όχι ότι κάποιος την σταμάτησε.

Σημαντική ήταν επίσης η ανάπτυξη του εργαλείου *Sebek* για καταγραφή όλων των εντολών που δίνει ο επιτιθέμενος αλλά και ότι αρχεία «ανεβάζει» με ένα τρόπο αρκετά πιο κρυφό και αποτελεσματικό σε σχέση με αυτά της πρώτης γενιάς των «honeynets». Το *Sebek* είναι ουσιαστικά ένα *module* το οποίο μπορεί να καταγράφει ακόμη και τις κρυπτογραφημένες κινήσεις του επιτιθέμενου αφού είναι εγκατεστημένο σε σημείο όπου η κίνηση έχει αποκρυπτογραφηθεί. Το εργαλείο αυτό αποτελείται από τρία μέρη. Αυτό του πελάτη (*client*), του διακομιστή (*server*), και του *web interface*. Το πρώτο εγκαθίσταται στο *honeypot* από όπου στέλνει ότι καταγράφει στο διακομιστή. Ο διακομιστής λαμβάνει όλες τις πληροφορίες που του στέλνουν όλοι οι πελάτες και μπορεί να το καταγράψει σε κάποια βάση δεδομένων. Το *web interface* του *Sebek* είναι μια εφαρμογή από όπου ο επόπτης έχει συγκεντρωτική εικόνα από όλα τα συστήματα που είναι εγκατεστημένη η αντίστοιχη *client* εφαρμογή.

Σημαντικό μειονέκτημα που υπάρχει στο *Sebek* λόγω του ότι είναι ανοιχτού κώδικα είναι το ότι μπορεί κάποιος κακόβουλος χρήστης να αναπτύξει κατάλληλα εργαλεία για την ανίχνευση του.

Σε αυτή την γενιά δημιουργήθηκε επίσης και το *Honeywall CDROM Eeyore*. Το *Eeyore* είναι ένας ψηφιακός δίσκος από όπου μπορεί ένας υπολογιστής να ξεκινήσει (*boot*) και να φορτώσει όλα τα απαραίτητα προγράμματα για την υλοποίηση ενός *honeynet* δεύτερης γενιάς με μεγαλύτερη ευκολία και ταχύτητα αναλαμβάνοντας την επίλυση κάποιων τεχνικών δυσκολιών με ένα αυτοματοποιημένο τρόπο. Συγκεκριμένα το *Eeyore* σχεδιάστηκε για να είναι:

- Εύκολο στην υλοποίηση και στην συντήρηση.

Ένα σωστά υλοποιημένο *honeynet* απαιτεί πολύ χρόνο για την εγκατάσταση, παραμετροποίηση και δοκιμή. Χρειάζονται αρκετά εργαλεία για τον έλεγχο και την καταγραφή της πληροφορίας αλλά και για την δημιουργία ενός μηχανισμού αυτόματης ειδοποίησης κινδύνων. Όλα αυτά τα εργαλεία χρειάζονται την δική τους εγκατάσταση, παραμετροποίηση και έλεγχο για την σωστή τους λειτουργία. Όλη αυτή την χρονοβόρα διαδικασία έρχεται και αναλαμβάνει το *Eeyore* το οποίο έχει έτοιμα όλα τα απαραίτητα εργαλεία. Η απομακρυσμένη διαχείριση γίνεται μέσω του *SSH*.

- Παραμετροποιήσιμο

Κάθε *honeynet* δεν είναι το ίδιο. Αν και το *Eeyore* έχει τα περισσότερα συνηθισμένα εργαλεία μπορεί σε κάποιες υλοποιήσεις να χρειάζονται κάποια εργαλεία τα οποία δεν είναι διαθέσιμα στο προκαθορισμένο ψηφιακό δίσκο. Για την κάλυψη αυτής της έλλειψης υπάρχει η δυνατότητα δημιουργίας παραμετροποιημένου *ISO image* και κατά συνέπεια ψηφιακού δίσκου που θα περιλαμβάνει ότι άλλο χρειάζεται για την υλοποίηση του συγκεκριμένου *honeynet*.

- Πλατφόρμα δικτυακής ασφάλειας

Επειδή μέσα στα εργαλεία που περιλαμβάνονται είναι και εργαλεία όπως τα *Snort*, *snort-inline*, *iptables bridging firewall* το *Eeyore* μπορεί να χρησιμοποιηθεί και ως ένα σύστημα ασφάλειας συστημάτων παραγωγής (*production systems*)

6.7.3 Τρίτη Γενιά *Honeypots*

Η γενιά αυτή χαρακτηρίζεται από την εξέλιξη και βελτίωση της τεχνολογίας που βρίσκουμε στην δεύτερη γενιά και κυρίως από την εξέλιξη του *Eeyore* που είναι το *Honeywall CDROM Roo*. Σημαντικές βελτιώσεις είναι η αυξημένη δυνατότητα αναγνώρισης υλικού (*hardware*), η δυνατότητα για εύκολη ενημέρωση (*updates*) και εγκατάσταση καινούργιων προγραμμάτων (*install*), η απομακρυσμένη διαχείριση από γραφικό περιβάλλον. Επίσης σε αυτή την γενιά βελτιώθηκε και το *Sebek*. Η βασική αρχιτεκτονική του δικτύου έχει παραμείνει ουσιαστικά η ίδια.

Κεφάλαιο 7. Εργαλεία Διαχείρισης *Honeypots*

Στο κεφάλαιο αυτό θα μιλήσουμε για την αποτελεσματική διαχείριση συστημάτων *honeypots* απέναντι σε επιθέσεις *botnets*. Θα γίνει αναφορά σε διάφορα εργαλεία που μπορούν να διαχειριστούν *honeypots* , καθώς επίσης και πως μπορούν τα *honeypots* να ανταπεξέλθουν σε επιθέσεις από δίκτυα *botnet*.

7.1 Τα *Honeypots* στην ανίχνευση *Botnet*

Παρόλο που η αξία των *honeypots* σε μεγάλης κλίμακας δίκτυα είναι περιορισμένη λόγω της δύσκολης εγκατάστασης αλλά και της διαδικασίας ενεργούς ανάλυσης την οποία απαιτούν, σε μικρότερα δίκτυα μπορούν να αποβούν πολύ χρήσιμα στον εντοπισμό *botnets*.

Όταν το κακόβουλο λογισμικό των *bot*s διεισδύσει σε κάποιο σύστημα *honeypot*, τότε το λογισμικό αυτό μπορεί να αναλυθεί και να παρακολουθηθεί η δραστηριότητά του. Αυτό σημαίνει ότι όταν το *bot* συνδεθεί στον *C&C* εξυπηρετητή η ταυτότητα του εξυπηρετητή θα αποκαλυφθεί. Επομένως μπορούν να συλλεχθούν πολλές πληροφορίες για το *bonet* όπως η IP διεύθυνση του *C&C* εξυπηρετητή, οι εντολές του *botmaster*, καθώς και το όνομα του καναλιού (*channel*), το ψευδώνυμο (*nickname*) και ο κωδικός με τον οποίο συνδέεται το *bot* στην περίπτωση που το κανάλι επικοινωνίας υλοποιείται μέσω ενός *IRC* εξυπηρετητή. Στην τελευταία περίπτωση του κεντρικοποιημένου *IRC* '*Command and Control*', μπορούμε να δημιουργήσουμε ένα ψεύτικο *bot* το οποίο θα συνδεθεί στο ίδιο *IRC* κανάλι με το ίδιο ψευδώνυμο και τον ίδιο κωδικό ώστε να παρακολουθεί ποιοι άλλοι είναι συνδεδεμένοι στο κανάλι. Με αυτό τον τρόπο μπορούμε να συλλέξουμε στατιστικά όπως πόσα είναι τα *bot*s κάθε στιγμή και ευαίσθητες πληροφορίες όπως το ποιοι είναι οι στόχοι των επιθέσεων και πότε γίνονται οι επιθέσεις. Τελικά θα μπορούσαμε να μπλοκάρουμε την επικοινωνία από και προς το *botnet* και να αναφέρουμε τη δραστηριότητά του ώστε να εξουδετερωθεί.

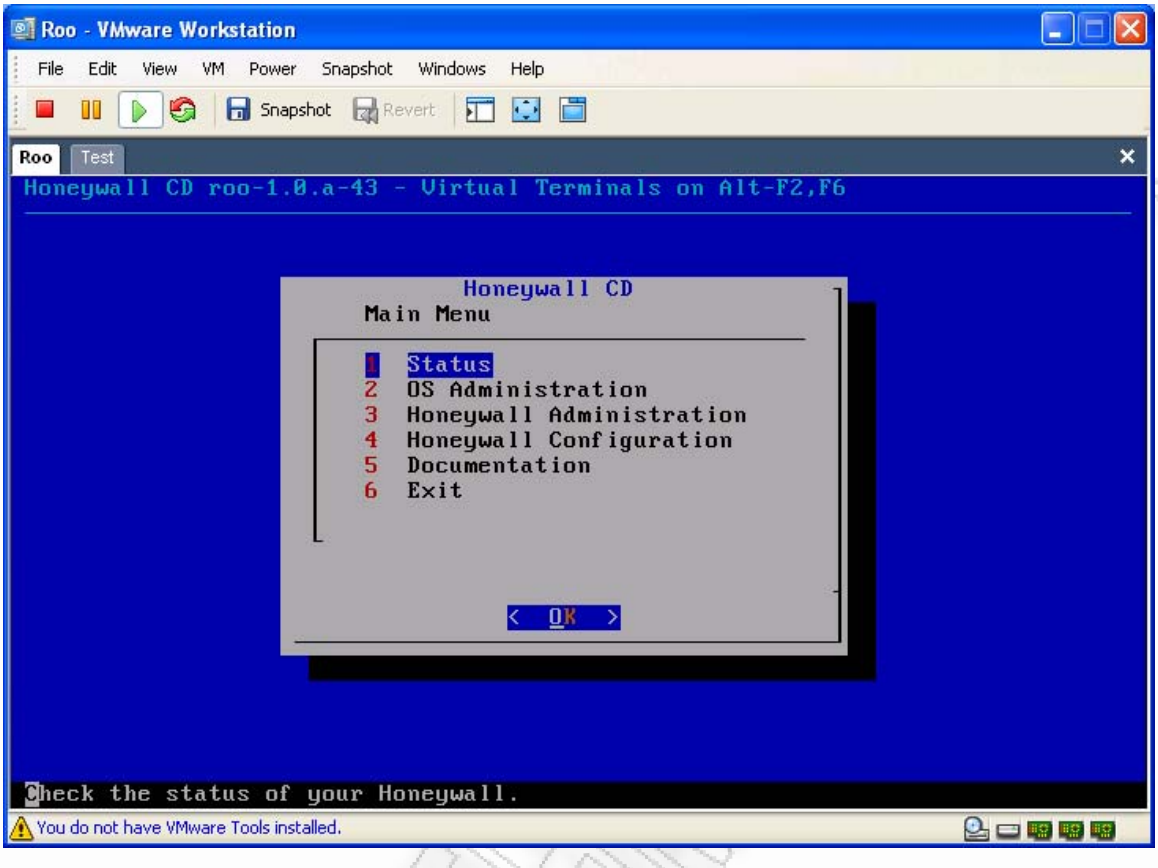
Στα πλαίσια της παρούσας διπλωματικής εργασίας παρουσιάζουμε παρακάτω διάφορα εργαλεία που μπορούν να διαχειριστούν κακόβουλες επιθέσεις από δίκτυα *botnet*. Όμως, η συλλογή των επιθέσεων, η ανάλυση τους και ο περαιτέρω πειραματισμός με την ανίχνευση *botnets* σε πραγματική κίνηση θα αποτελέσει αντικείμενο μελέτης μελλοντικής εργασίας.

7.2 Εργαλεία Διαχείρισης *Honeypot*

Για να αναλυθεί η κίνηση που προκαλούν τα *botnet* μέσω *honeypot* χρησιμοποιούνται διάφορα εργαλεία. Τα εργαλεία αυτά συμβάλουν στην καλύτερη ανάλυση των δεδομένων που θα ληφθούν από το *honeypot*. Στη συνέχεια παρουσιάζουμε τα εργαλεία αυτά, αναφέροντας τις βασικές δυνατότητες και λειτουργίες τους.

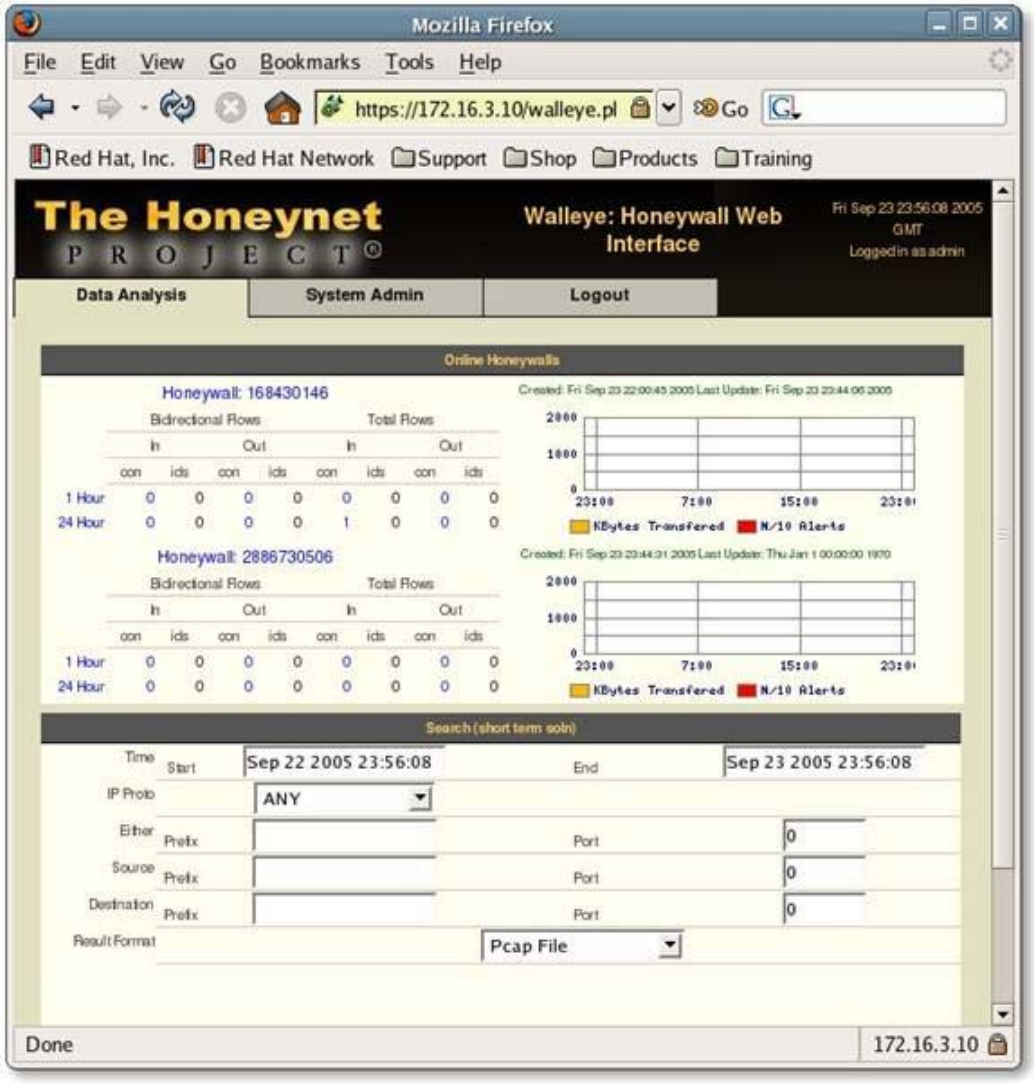
7.2.1 Honeywall CDROM

Το *Honeywall CDROM* είναι ένα εργαλείο, το οποίο διευκολύνει τη δημιουργία ενός *honeynet*, κάνοντας την εγκατάσταση του *honeywall* αρκετά πιο απλή. Πρόκειται για ένα boot cd, εγκατάστασης του λειτουργικού συστήματος Fedora Core 6, συμπεριλαμβανομένων των προγραμμάτων που απαιτούνται για τον έλεγχο, τη συλλογή και την ανάλυση των δεδομένων κίνησης. Επίσης παρέχει εργαλεία για την ευκολότερη παραμετροποίηση του *honeynet*. Για τη διαφανή λειτουργία του χρησιμοποιεί layer 2 bridging mode, σε δύο network interfaces, ένα για σύνδεση με το εξωτερικό δίκτυο και ένα για το εσωτερικό δίκτυο, όπου και βρίσκονται τα *honeypots*. Το *layer* 2 *bridging* mode κάνει προώθηση πακέτων, χωρίς να είναι απαραίτητη η δρομολόγηση προς τα συγκεκριμένα interfaces αφού δεν υπάρχει *IP address* που να αντιστοιχεί σε αυτά.



Σχήμα 7.1 Ρύθμιση Παραμέτρων Honeywall

Εκτός από τα δύο interfaces, που προαναφέρθηκαν, το *Honeywall CDROM* υποστηρίζει ακόμα ένα, με σκοπό την απομακρυσμένη διαχείριση του. Στο τρίτο αυτό interface, στο οποίο ανατίθεται κανονικά *IP* διεύθυνση, υπάρχει δυνατότητα εγκατάστασης ενός Web και ενός SSH *server*. Ο πρώτος φιλοξενεί το web-based εργαλείο Walleye, ενώ ο δεύτερος χρησιμοποιείται για απομακρυσμένη σύνδεση με το *honeywall*. Το Walleye, παρόλο που η εγκατάσταση του είναι προαιρετική, αποτελεί το κύριο εργαλείο διαχείρισης του *honeywall*. Όπως προαναφέρθηκε είναι ένα web-based εργαλείο, που παρέχει τη δυνατότητα πλήρους ελέγχου του *honeywall*. Μέσα από το Walleye είναι δυνατό να γίνει η παραμετροποίηση του συστήματος, ενώ παράλληλα άνω, ένα αρκετά εύχρηστο και πολύτιμο εργαλείο ανάλυσης των δεδομένων κίνησης.

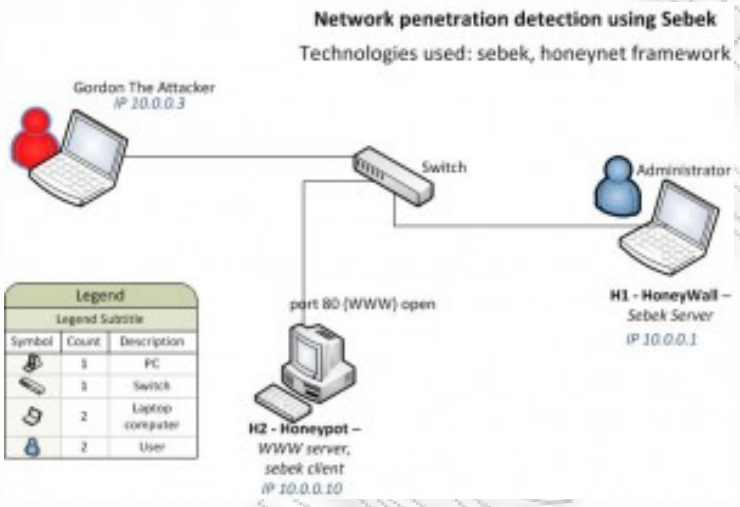


Σχήμα 7.2 Το Web Interface του Walleye

7.2.2 Sebek

Το *Sebek* είναι ένα εργαλείο, με βασικό σκοπό την καταγραφή κάθε ενέργειας που πραγματοποιείται στα *honeypots*. Χωρίζεται σε δύο επιμέρους εργαλεία, το *Sebek Client* και το *Sebek Server*. Το *Sebel Client* εγκαθίσταται σε κάθε *honeypot* και λειτουργεί σαν kernel *rootkit* , αποκρύπτοντας έτσι την ύπαρξη του. Χρησιμοποιείται για την καταγραφή δεδομένων, όπως *keystrokes*, file *uploads* και *passwords*, ακόμα και αν είναι κρυπτογραφημένα, αφού γίνεται σε επίπεδο πυρήνα. Στη συνέχεια η πληροφορία

αποστέλλεται μέσω *UDP* πακέτων στο *Sebek Server*, όπου βρίσκεται εγκατεστημένος στο *Honeywall*. Ο *Sebek Server* δείχνει αναγνωρίζει τα πακέτα, λαμβάνοντας τα σε ένα συγκεκριμένο port, το οποίο ορίζει ο χρήστης, και στη συνέχεια καταγράφει την πληροφορία που περιέχουν, ώστε να είναι εφικτή η μελέτη της μέσα από το *Honeywall*.



Σχήμα 7.3 Δίκτυο ανίχνευσης διείσδυσης χρησιμοποιώντας Sebek

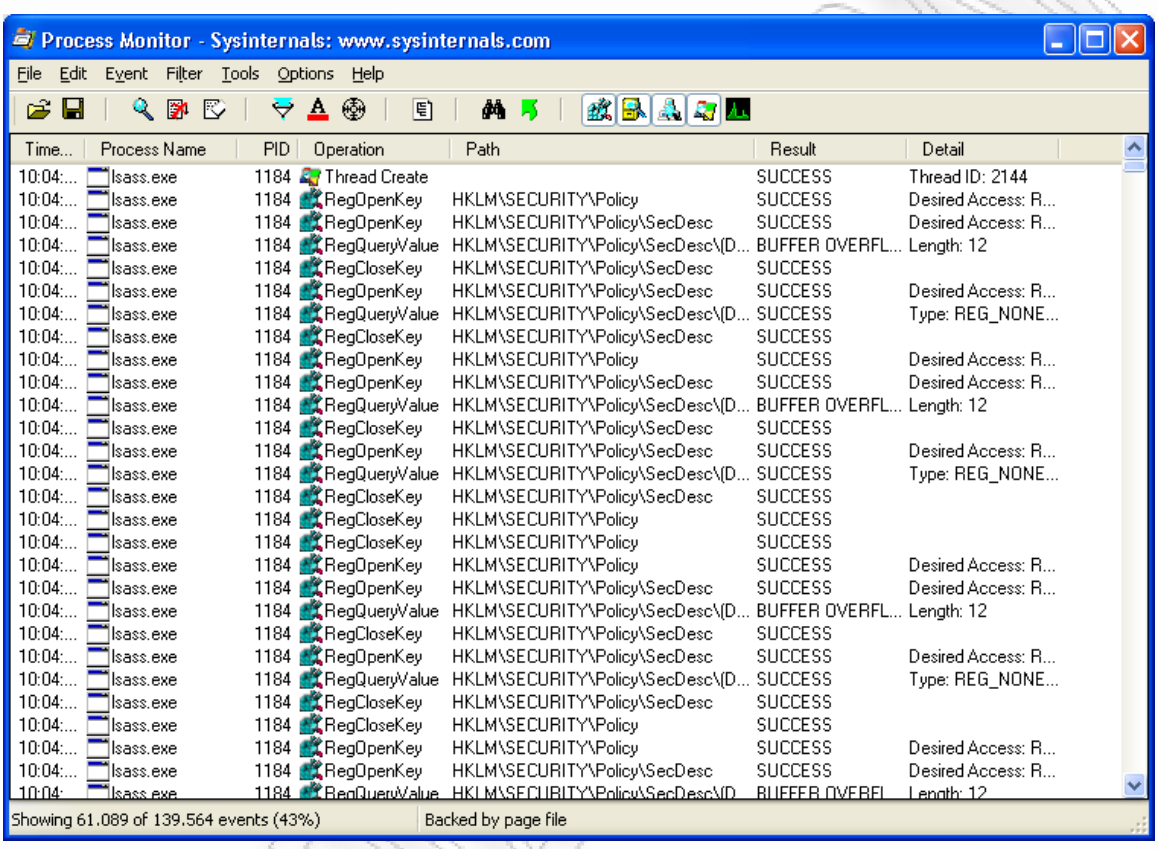
7.2.3 Honeysnap

Το *Honeysnap* είναι ένα εργαλείο περιβάλλοντος εντολών, με δυνατότητα ανάγνωσης και ανάλυσης ενός ή πολλαπλών αρχείων pcap, που περιέχουν δεδομένα κίνησης. Παρέχει μία πρώτη ανάλυση των δεδομένων που συλλέγονται από τα *honeypot*, ώστε να μπορέσει ο αναλυτής να αναγνωρίσει τις ενέργειες που το ενδιαφέρουν και να καταγράψει τη χρονική στιγμή που πραγματοποιήθηκαν. Αυτό γίνεται μέσα από ένα report που δημιουργεί, το οποίο περιλαμβάνει πληροφορίες που αφορούν εισερχόμενη και εξερχόμενη κίνηση, *HTTP* και *IRC* επικοινωνία, dns resolves, κ.α. Στη συνέχεια μπορεί να ανατρέξει στο *Walleye*, για μία πιο λεπτομερή ανάλυση.

7.2.4 Process Monitor

Για την επιμέρους μελέτη των binaries που συλλέγονται από τα *honeypot*, μπορεί να χρησιμοποιηθεί και το εργαλείο της Microsoft, το *Process Monitor*. Το εργαλείο *Process Monitor* παρέχει δυνατότητα καταγραφής, σε πραγματικό χρόνο, ενεργειών που

αφορούν το file system, τη registry και τις διεργασίες ή threads. Ουσιαστικά μας δίνει μία πλήρη εικόνα των γεγονότων που ακολουθούν την εκτέλεση ενός binary, με σκοπό τη μελέτη κάθε binary ξεχωριστά.



Σχήμα 7.4 Το Interface του Process Monitor

Επίσης το πρόγραμμα μπορεί να δείχνει όλες τις αλλαγές που γίνονται με τα αρχεία, να ενημερώνει για την διαγραφή τους ή την προσπέλασή τους. Στο Process Monitor υπάρχουν και εργαλεία για την παρακολούθηση της κατάστασης του μητρώου. Η εφαρμογή θα δείξει ποιες εφαρμογές απευθύνονται στο μητρώο και, συγκεκριμένα, σε ποια κλειδιά, τι δεδομένα προσπαθούν να διαβάσουν ή να εγγράψουν. Η ιδιαιτερότητα αυτού του προγράμματος είναι, ότι εντοπίζει όχι μόνο τις τιμές και τα κλειδιά που είχαν τροποποιηθεί, αλλά και δείχνει ποιες ακριβώς αλλαγές είχαν γίνει.

Κεφάλαιο 8. Συμπεράσματα & Μελλοντικές Προτάσεις

8.1 Συμπεράσματα

Η ασφάλεια έχει γίνει κρίσιμο συστατικό της σχεδίασης συστημάτων και δικτύων σήμερα. Στη πραγματικότητα πρόκειται για ένα αγώνα χωρίς τέλος ανάμεσα στους κακόβουλους επιτιθέμενους και τους υπεύθυνους ασφαλείας των συστημάτων. Οι επιτιθέμενοι χρησιμοποιούν ολοένα και εξυπνότερους τρόπους εκμετάλλευσης των αδυναμιών των συστημάτων, έτσι θα υπάρχει πάντα η ανάγκη για εξυπνότερες και καλύτερες λύσεις ασφαλείας, δεδομένου ότι τα συστήματα δεν μπορεί να είναι τέλεια χωρίς αδυναμίες στο λογισμικό τους.

Τα *botnets* έχουν εδραιώσει την θέση τους σαν μία από τις βασικότερες απειλές που καραδοκούν στο σημερινό διαδίκτυο. Εκατομμύρια υπολογιστών είναι υπονομευμένοι στο διαδίκτυο και μπορούν να ελέγχονται από κακόβουλους για την εκκίνηση επιθέσεων μεγάλης ισχύος και διάπραξη δραστηριοτήτων απάτης. Ταυτόχρονα τα *botnets* δημιουργούν συνθήκες για την καλλιέργεια και ανάπτυξη νέων μορφών εγκληματικότητας. Παρόλα αυτά, η γνώση μας για τη συμπεριφορά των *botnets*, τους τρόπους ανίχνευσης και αντιμετώπισής τους είναι ακόμη ελλιπής. Απαιτείται επειγόντως η κατανόηση των *botnets* και λύσεις για την ανίχνευση, το μετριασμό των συνεπειών και την αντιμετώπισή τους.

Με την εργασία αυτή συμβάλλουμε στη συνειδητοποίηση του προβλήματος και την κατανόηση των λειτουργιών των *botnets* μέσα από την παρουσίαση των αρχιτεκτονικών που αυτά χρησιμοποιούν. Παράλληλα συνεισφέρουμε στην εξεύρεση λύσεων κατηγοριοποιώντας και αναλύοντας όλες τις τεχνικές ανίχνευσης που μπορούν να χρησιμοποιηθούν για τον εντοπισμό των *botnets*. Ο πιο αποδοτικός τρόπος να εξουδετερωθεί ένα *botnet* είναι ο εντοπισμός των *C&C* εξυπηρετητών που ελέγχονται από το *botmaster*, όμως αυτό είναι και το δυσκολότερο.

Από την άλλη μεριά, οι τεχνικές ανίχνευσης μπορεί να γίνουν αποτελεσματικές μόνο αν γνωρίζουμε σε βάθος τις τεχνικές απόκρυψης που τα *botnets* χρησιμοποιούν για να αποφύγουν την ανίχνευσή τους. Η εργασία μας συγκεντρώνει και αναλύει στοιχεία

σχετικά με τα *Honeypots*, τα οποία έχουν διάφορες παραλλαγές και εμπλουτίζονται συνεχώς με νέα χαρακτηριστικά και νέες συμπεριφορές.

Τα *honeypots* έχουν και πρέπει να έχουν την θέση τους σε μεγάλες εταιρείες ως ένα επιπρόσθετο μέτρο ανίχνευσης και προστασίας απειλών και σε εταιρείες αντιμετώπισης ιών (*virus*) για τον εντοπισμό και παγίδευση *worms* και *virus* για την περαιτέρω ανάλυση και κατασκευή κατάλληλων και αποτελεσματικότερων *antivirus*. Στην αντιμετώπιση όμως απειλών της κατηγορίας *zero day exploits* δεν είναι το ίδιο αποτελεσματικά. Ωστόσο δεν μπορούμε να εκμηδενίσουμε την αξία τους και σε αυτήν την κατηγορία, την στιγμή μάλιστα που οι υπάρχουσες λύσεις αντιμετώπισης και καταγραφής αυτών των κενών ασφάλειας δεν είναι πολύ περισσότερο αποτελεσματικές. Επίσης ακόμη η τεχνολογία αυτή εξελίσσεται και καινούργιες ιδέες υλοποιούνται.

8.2 Μελλοντική Εργασία

Στα σχέδια της μελλοντικής μας δουλειάς προτείνουμε την δημιουργία ενός εικονικού δικτύου *botnet* και ενός δικτύου με *honeypots*. Εκεί θα μπορέσουμε να κάνουμε επιθέσεις και να δούμε τις τρύπες ασφαλείας που μπορεί να έχει ένα δίκτυο και να προσπαθήσουμε να τις αποτρέψουμε με τη βοήθεια του *honeypot*. Σκοπός μας θα είναι ο περαιτέρω πειραματισμός με περισσότερα λογισμικά ανίχνευσης *botnets* και με διαφορετικές τεχνικές ανίχνευσης. Θα θέλαμε να κάνουμε τα όρια της ανίχνευσης και της απόκρυψης πιο ξεκάθαρα για κάθε ξεχωριστή τεχνική που χρησιμοποιείται.

Η μελλοντική έρευνα χρειάζεται να επικεντρώσει όχι μόνο στο μετριασμό των επιδράσεων των επιθέσεων που προκαλούνται από τα *botnets*, άλλα και στο πως μπορούν τα *botnets* που πραγματοποιούν τις επιθέσεις να εξουδετερωθούν.

Τέλος θα μπορούσαμε να μελετήσουμε τα αποτελέσματα με συνεργατική ανίχνευση συνδυάζοντας τεχνικές ανίχνευσης σε επίπεδο υπολογιστή με τεχνικές ανίχνευσης σε επίπεδο δικτύου. Οι προσεγγίσεις ανίχνευσης σε επίπεδο υπολογιστή μπορούν να παράσχουν διαφορετικές ενδείξεις-στοιχεία που οι προσεγγίσεις από το επίπεδο δικτύου δεν μπορούν. Ο συνδυασμός των δύο αυτών συμπληρωματικών προσεγγίσεων μπορεί πιθανώς να αποφέρει καλύτερα αποτελέσματα στην ανίχνευση, αποκαλύπτοντας *botnets*

με ευφυείς τεχνικές απόκρυψης, *botnets* με κρυπτογραφημένο κανάλι *C&C* κτλ. Για την υλοποίηση αυτού του στόχου, θα μπορούσαμε να χρησιμοποιήσουμε *honeypots* 4ης γενιάς με διάφορες παραλλαγές για να μπορέσουμε να διαπιστώσουμε την καλύτερη δυνατή λύση. Συλλέγοντας και αναλύοντας κίνηση πραγματικών επιθέσεων θα μπορέσουμε να μελετήσουμε καλύτερα τη συμπεριφορά των *botnets* και να βελτιώσουμε τις χρησιμοποιούμενες τεχνικές ανίχνευσης μέσω των *honeypots*.

Βιβλιογραφία

[1] Spitzner, L. *"Honeypots: Tracking Hackers"*. Addison-Wesley Professional. September 20, 2002.

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

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

[4] http://www.cs.vu.nl/~herbertb/misc/shelia/shelia07.pdf

[5] Schoof, Reinier and Koning, Ralph. Detecting peer-to-peer botnets. s.l. : System and Network Engineering, University of Amsterdam, 2007.

[6] Villamarín-Salomón, Ricardo and Brustoloni, José Carlos. Identifying Botnets Using Anomaly Detection Techniques Applied to DNS Traffic. s.l. : Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE, 2008.

[7] Canavan, John. The Evolution of Malicious IRC Bots, White Paper. s.l. : Symantec Security Response, 2005.

[8] Florio, Elia and Ciubotariu, Mircea. Peerbot: Catch me if you can, White Paper. Ireland : Symantec Security Response, 2007.

[9] Trend Micro: Taxonomy of botnet threats, white paper, 2006

[10] Al-Hammadi Y. and Aickelin U., Detecting Botnets Through Log Correlation. IEEE / IST Workshop on "Monitoring, Attack Detection and Mitigation", 2006

[11] Cooke, Evan, Jahanian, Farnam and McPherson, Danny. The Zombie Roundup:Understanding, Detecting, and Disrupting Botnets. s.l. : USENIX, Steps to Reducing Unwanted Traffic on the Internet Workshop, 2005.

[12] Strayer W. T., Walsh R., Livadas C., and Lapsley D., Detecting Botnets with Tight Command and Control. IEEE, Proceedings of the 31st IEEE Conference on Local Computer Networks (LCN), 2006

[13] Choi H., Lee H., Lee H., Kim H., Botnet Detection by Monitoring Group Activities in DNS Traffic. IEEE, Proceedings of the 7th IEEE International Conference on Computer and Information Technology

[14] Porras P., Saidi H., and Yegneswaran V., A Multi-perspective Analysis of the Storm (Peacomm) Worm. Technical report, SRI International, October 2007

[15] Goebel J. and Holz T., Rishi: Identify bot contaminated hosts by IRC nickname

[16] Yen Ting-Fang and Reiter Michael, Traffic Aggregation for Malware Detection. Proceedings of the 5th international conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA), 2008

[17] Dreger H., Feldmann A., Mai M., Paxson V., Sommer R., Dynamic Application-Layer Protocol Analysis for Network Intrusion Detection. USENIX Security Symposium, Proceedings of the 15th conference, 2006

[18] Akiyama M., Kawamoto T., Shimamura M., Yokoyama T., Kadobayashi Y., and Yamaguchi S., A proposal of metrics for botnet detection based on its cooperative behaviour. International Symposium on Applications and the Internet (SAINT) 2007

[19] Gu G., Zhang J., and Lee W., BotSniffer Detecting Botnet Command and Control Channels in Network Traffic. 15th Annual Network & Distributed System Security Symposium (NDSS), 2008

[20] Botnet Research Survey. IEEE, COMPSAC '08: Proceedings of the 2008 32nd Annual IEEE International Computer Software and Applications Conference

[21] Gu G., Porras P., Yegneswaran V., Fong M., and Lee W., BotHunter: Detecting Malware Infection Through IDS-Driven Dialog Correlation. USENIX, Proceedings of 16th USENIX Security Symposium, 2007

[22] Stinson E. and Mitchell J., Towards Systematic Evaluation of the Evadability of Bot/Botnet Detection Methods, USENIX, Proceedings of the 2nd conference on USENIX Workshop on offensive technologies, 2008

[23] Grizzard J. B., Sharma V., Nunnery C., Kang B. B., Dagon D., Peer-to-Peer Botnets Overview and Case Study. USENIX, Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets, HotBots 2007

[24] Malware Tunneling in IPv6, http://www.us-cert.gov/reading_room/IPv6Malware-Tunneling.pdf

[25] Fast flux, http://en.wikipedia.org/wiki/Fast_flux

[26] Al-Hammadi Y. and Aickelin U., Detecting Bots Based on Keylogging Activities. IEEE, Proceedings of the 2008 Third International Conference on Availability, Reliability and Security

[27] Gu G., Perdisci R., Zhang J., and Lee W., BotMiner Clustering Analysis of Network Traffic for Protocol- and Structure-Independent Botnet Detection, USENIX, Proceedings of the 17th conference on Security symposium, 2008

[28] Botnets: The New Threat Landscape, white paper, Cisco 12/2007

[29] Snort (software), http://en.wikipedia.org/wiki/Snort_(software)

[30] Beale J., Foster J., Posluns J., Russell R., Caswell B., Snort 2.0 Intrusion Detection, book, 2003

[31] BotHunter®, http://www.bothunter.net/

[32] LaBrea: "Sticky" Honeypot and IDS, http://labrea.sourceforge.net/