Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

Fachbereich Informatik
Department of Computer Science

Botnetze
Funktion, Erkennung und Entfernung

von

Daniel Baier

Für die Veröffentlichung angepasste Seminararbeit in der Veranstaltung „Penetration Testing 2“, 5. Semester BCS, bei Herr Prof. Dr. Hartmut Pohl.

Eingereicht am: 15.02.2010

Angepasst am: 13.06.2010

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

3LD	Third Level Domain
BSI	Bundesamt für Sicherheit in der Informationstechnik
C&C	Command-and-Control-Server
CPU	Central Processing Unit
DDoS	Distributed Denial of Service
DNS	Domain Name System
DynDNS	Dynamisches Domain Name System
ICMP	Internet Control Message Protocol
ICQ	I Seek You
GHDB	Google Hacking Database
HIDS	Host-Based Intrusion Detection Systeme
IFrame	Inline Frame
IRC	Internet Relay Chat
IP	Internet Protocol
P2P	Peer-to-Peer
PHP	PHP: Hypertext Preprocessor
RIPE NCC	Réseaux IP Européens Network Coordination Centre
RIR	Regional Internet Registry
RR	Resource Record
SLD	Second Level Domain
SPAN	Switched Port Analyzer
TTL	Time-to-Live
VoIP	Voice over IP

1 Einführung

1.1 Motivation

Bei einem Bot handelt es sich um ein Computerprogramm, das auf einen kompromittierten System installiert wird. Dieser bietet einem Angreifer einen Kommunikationskanal um den Bot und somit das System zu kontrollieren. Werden mehrere Bots zu einem Netzwerk zusammengeschlossen, spricht man von einem Botnetz.

Botnetze stellen eine immer größere Bedrohung der Internetkriminalität unserer heutigen Zeit dar [Binkley, Schiller 2007]. So dienen diese dem Versenden von Spam oder dem Durchführen von koordinierten Distributed Denial of Service (DDoS) Angriffen. Weitere Verwendungszwecke sind Kapitel 1.4 zu entnehmen.

Die Befehle erhalten die Bots von einer meist zentralisierten Kontrollstruktur, dem sogenannten Command-and-Control-Server (C&C).

Die Funktionalität mit dem der Betreiber eines Botnetzes (Bot-Herder) mit seinen Bots kommuniziert sind verschieden. In der Vergangenheit wurden IRC-Channel benutzt. Der Betreiber eines Botnetzes richtet einen eigenen IRC-Server ein und eröffnet einen so genannten Channel, auf dem er seine Befehle veröffentlicht. Die Bots verbinden sich autonom zu diesem Channel, um auf ihre Befehle zu lauschen. Andere Betreiber nutzen das HTTP-Protokoll zur Kommunikation mit seinen Bots. Analog zum IRC-Server wird ein HTTP-Server eingerichtet. Im Vergleich zum IRC findet keine persistente Verbindung statt, sondern die Bots verbinden sich periodisch zum HTTP-Server. Neben eigenen HTTP-Server werden auch Web 2.0 Dienste benutzt. In Twitter wurde jüngst (August 2009) erst ein Account entdeckt, der als C&C diente [Nazario 2009]. Alternativ werden Peer-to-Peer-Protokolle (P2P) sowie selbst entwickelte Netzwerkprotokolle genutzt.

Um diesen Gefahren entgegen zu Wirken, werden unterschiedliche Botnet Detection Mechanismen benutzt, u.a. sogenannte Honeypots. Dies sind Systeme die Netzwerkdienste oder ein ganzes Rechnernetz simulieren.

1.2 Definition

1.2.1 Bot

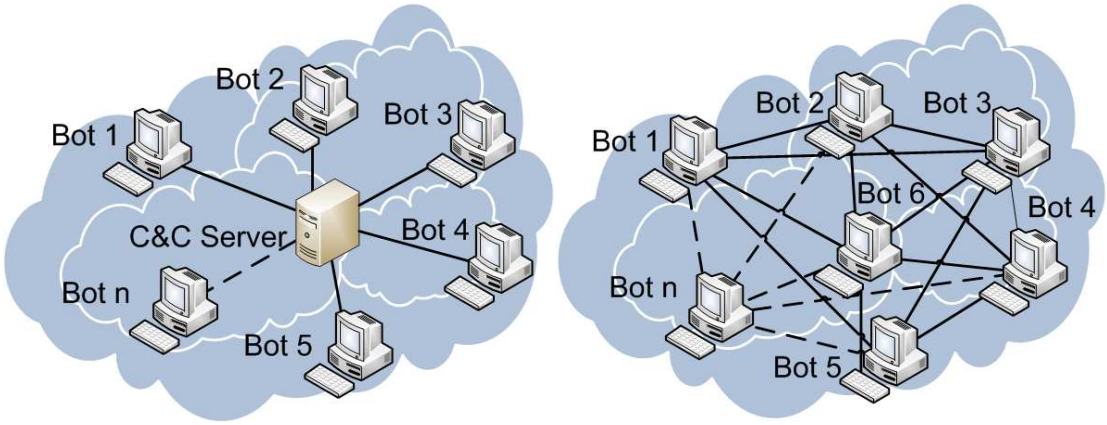
Nach Dunham und Melnick definiert sich ein Bot als

“Malicous code that acts like a remotely controlled „robot“ for an attacker, with both Trojan und worm capabilities. This term may refer to the code itself or to an infected computer, also known as a drone or zombie.“ [Dunham, Melnick 2009].

Also definiert sich ein Bot als bösartiger Code, der sich wie ein fernsteuerbarer “Roboter” für einen Angreifer verhält, mit sowohl Trojaner- als auch Wurm-Eigenschaften. Dies kann sowohl den Code selbst oder den infizierten Computer betreffen, welcher auch als Drohne oder Zombie bezeichnet wird.

1.2.2 Botnetz

Als Botnetz wird der Zusammenschluss von mehreren Bots zu einem Netzwerk bezeichnet.



a) **Zentralisierte Topologie (C&C):** Die Bots verbinden sich zum zentralen C&C Server und erwarten dort die Befehle des Bot-Herders.

b) **Dezentralisierte Topologie (P2P):** Es existiert keine zentrale Kontrollstruktur, jeder Bot kann Befehle erteilen und entgegen nehmen.

Abbildung 1: Vergleich der Botnetz Topologien

1.3 Übersicht von aktiven Botnetzen

In nachfolgender Tabelle ist eine Übersicht der aktiven Botnetze aus bzw. bis zum Jahre 2009 zu sehen und deren geschätzte Größe. Dabei kann ein Computer mit verschiedenen Bots infiziert sein.

Name	Geschätzte Botzahl	Aliases
Conficker	9.000.000 [F-Secure 2009]	DownUp, DownAndUp, DownAdUp, Kido
Rustock	1.200.000-2.000.000	RKRustok, Costrat
Cutwail	1.000.000-1.500.000	Pandex, Mutant
Grum	600.000-800.000	Tedroo
Bagle	600.000-800.000	?
Maazben	200.000-300.000	?
Festi	100.000-200.000	?
Kraken	80.000-120.000	Bobax
*Mega-D**	*< 100.000*	*Ozdok*
Xarvester	20.000-36.000	?

Tabelle 1: Übersicht von aktiven Botnetzen [Zors 2009]

Das Mega-D Botnetz existiert weiterhin, ist allerdings seit November 2009 in der Kontrolle des Sicherheitsdienstleister FireEye [FireEye 2009]

1.4 Verwendung

Ein Botnetz kann zu allem genutzt werden, was man sich unter Nutzung von einem Zusammenschluss von mehreren Bots (meist infizierte Computer) zu einem Netzwerk vorstellen kann. Tabelle 2 listet die bekanntesten Verwendungszwecke auf.

Verwendungszweck	Beschreibung
Data Mining	Die Bots haben Tools implementiert um Informationen von den Opfer-System (dem mit dem Bot infizierten System) zusammeln und auszuwerten.
DDoS	Durch verteilte kontinuierliche Verbindungsanfragen der Bots an einem Host (Server) werden die Dienste des Servers blockiert.
Datendiebstahl	Die Daten auf den Opfer-System werden von den Bots kopiert oder verschoben.
Löschen von Hinweisen	Alle Hinweise die eine Kompromittierung des betroffenen Systems erkennen lassen, werden gelöscht.
Proxy & Anonymisierer	Der Bot-Herder stellt alle seine Verbindungen über den Bot infizierten System her, so dass seine wahre Identität verborgen bleibt.
Ransomware	Bots verschlüsseln (private) Daten auf den Opfer-Systemen und verlangen zum Entschlüsseln ein „Lösegeld“.
Rekrutieren von Bots	Der Bot sucht nach verwundbaren Systemen und versucht diese zu infizieren.
Reporting	Es werden Berichte über die Ressourcen der Opfer-Systeme erstellt.
Spamming & Phishing	Die Bots werden zum Versenden von Spam- oder Phishing-Nachrichten benutzt.
Verteilung von illegalen Inhalten	Das Botnetz wird als Speicherort von illegalen Inhalt genutzt und zur Verteilung dessen.

Tabelle 2: Verwendung von Botnetzen

2 Techniken von Botnetzen

2.1 Infektions-Mechanismen

Die Infektions-Mechanismen unterteilen sich in 2 Klassen.

2.1.1 Social Engineering: Infektion mit Interaktion

Bei Infektionen mittels Social Engineering findet eine direkte Interaktion mit dem Benutzer des zu infizierenden Computers (im weiteren Verlauf „Opfer-System“ genannt) statt. Das Opfer-System wird dazu animiert sich z.B. Codecs für Videos (siehe Abbildung 2) oder besondere ActiveX-Controlls zu installieren bzw. runterzuladen. Diese Technik wird als Drive-by-download bezeichnet. Webseiten oder Blogs wie die von F-Secure informieren kontinuierlich über solche Angriffsversuche [Krogoth 2008a]. Eine weitere Methode, die sich Angreifer zu Nutze machen, ist das Opfer-System zum Ausführen zu verleiten (z.B. zum Öffnen des E-Mail Anhangs). Es sei noch anzumerken, dass beim Drive-by-download die Infektion auch am Benutzer vorbei erfolgen kann (Drive-by-infection). Durch z.B. Ausnutzen von Sicherheitslücken, welche unter der Klasse der automatisierten Infektionen fällt.

Abbildung 2: Bsp. Social Engineering (Drive-by-download)

2.1.2 Automatisiertes Exploiting

Bei automatisierten Infektionen werden Sicherheitslücken ausgenutzt. Unterschieden wird zwischen Lokalen-Exploits, zum ausweiten der Privilegien auf dem Opfer-System und Remote-Exploits um ein Opfer-System zu infizieren. Beliebt sind hier so genannte Malware Kids bzw. Web Exploit Toolkits (kurz WET) auf PHP-Basis, um den Browsers des Opfer-Systems automatisiert zu infizieren. MPack ist ein solches WET, welches 2006 der Öffentlichkeit bekannt wurde [Martinez 2007]. Angreifer benutzen MPack um es in einem IFrame auf einer kompromittierten Webseite zu hinterlegen, so dass jeder Besucher über das IFrame von MPack angegriffen wird. Abbildung 3 zeigt die Statistik-Webseite von MPack in der Version 0.99.

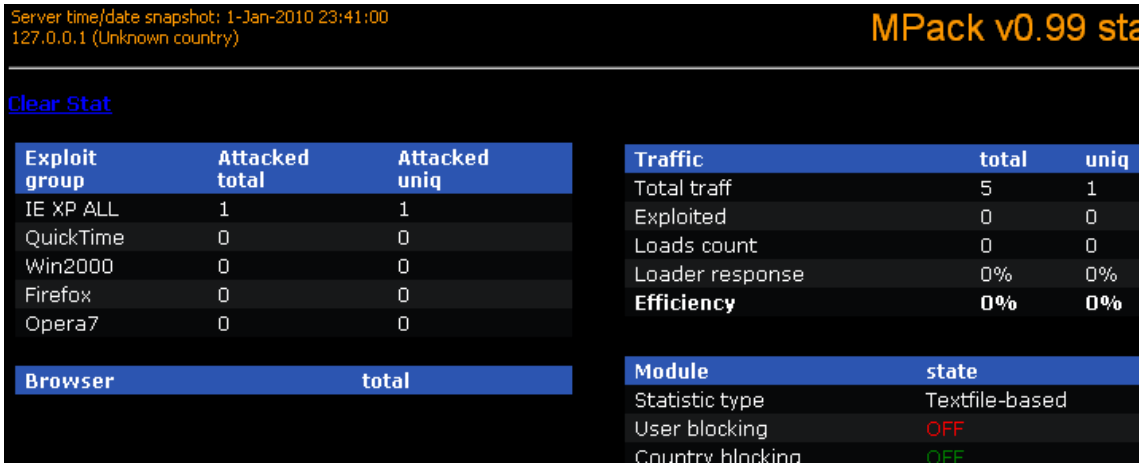


Abbildung 3: Ausschnitt aus der MPack (v. 0.99) Management Webseite

2.2 Scanning-Mechanismen

Ein Bot benötigt Strategien um sich unentdeckt möglichst schnell zu verbreiten. Obwohl jeder Bot zum Scannen genutzt werden könnte, werden nur einige ausgewählt, damit nur die Bots entblößt werden, die zur Verbreitung des Botnetzes benötigt werden.

2.2.1 Hit-list Scanning

Der Bot versucht nicht selbst Opfer-Systeme zu sichten, sondern nutzt eine vorkompilierte Liste an verwundbaren Zielsystemen. Diese Liste wird dann sequentiell abgearbeitet und jedes enthaltene System wird angegriffen. Die Hit-list ist entweder Bestandteil des Bots selbst oder kann z.B. über einen Webserver nachgeladen werden.

2.2.2 Topological Scanning

Diese Scanning-Strategie ist eine Alternative zum Hit-list Scanning. Der Bot verbreitet sich mit der Hilfe eines P2P Netzes, in dem die Liste der bekannten Peers benutzt wird. Dies ermöglicht eine schnelle Verbreitung.

2.2.3 Flash Scanning

Es besteht keine Liste an verwundbaren Zielsystemen, sondern eine schon bestehende Liste an Computersystemen, welche als Angriffsliste benutzt wird. Hier ist die Verbreitung der Bots am schnellsten, da kein Scann durchgeführt werden muss. Dabei kann diese Liste auch dynamisch erstellt werden in dem Google Dorks dazu missbraucht werden. Google Dorks sind vorgefertigte Suchpattern für Google, die z.B. in der Google Hacking Database (GHDB) zu finden sind [Long 2010].

2.2.4 Permutation Scanning

Keine Scanning- sondern viel mehr eine Reinfektions-Strategie. Das bedeutet, dass verhindert wird, das ein schon kompromittiertes System nochmals gescannt bzw. infiziert wird. Dies erfordert eine verteilte Koordination des Botnetzes.

2.2.5 Passive Scanning

Ein Bot der sich nicht aktiv verbreitet, sondern auf einem System ausgeführt wird und auf Verbindungen von dritt Anwendungen wartet. Diese Scanning-Strategie wird meist bei Malware genutzt, welche Vorteile von anderen Würmern bezieht. CRClean ist ein solcher Wurm der auf Reaktion des Code Red II Wums reagiert hat. Denn er entfernte

zunächst den Code Red II Wurm und infizierte das System dann selbst mit seinem eigenen Payload [Krogoth 2008b].

2.3 Kommunikationsmechanismen

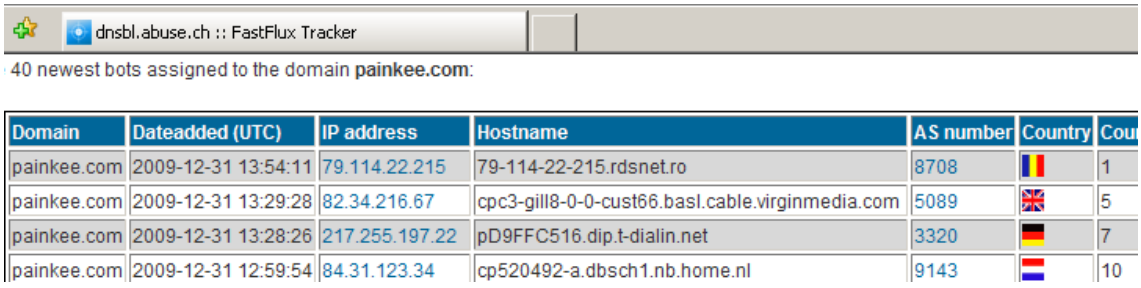
Die Kommunikation bei Botnetzen teilt sich in zwei Klassen. Die Verbindungsaufnahme, also die verschiedenen Methoden die ein Bot hat Verbindung mit dem C&C Server aufzunehmen und die dazu verwendeten Kommunikationsprotokolle.

2.3.1 Verbindungsaufnahme

Die einfachste Form bietet eine hart kodierte IP Adresse. Damit der C&C Server hinter der IP Adresse nicht einfach vom Netz genommen werden kann, machen sich Angreifer sogenannte “Bullet Proof Hosting“-Angebote zu nutze. “ Bei dieser Art des Hostings sorgen sich die Betreiber der Plattformen nicht um ihren guten Ruf und lassen Regelverletzungen bewusst zu.“ [Kaspersky Lab 2009].

Alternativ wird auf DynDNS zurückgegriffen. Ändert sich die IP-Adresse, wird der entsprechende DNS-Record auf die IP-Adresse aktualisiert. Dieses Prinzip wird auch eingesetzt sobald ein C&C erfolgreich vom Netz genommen wurde. Der Bot-Herder braucht nur zu einer neuen IP zu wandern und den entsprechenden DynDNS-Record anzupassen.

Die neueste und komplexeste Form ist die Verwendung von Fast-Flux Netzwerken. Dies sind Netzwerke, bei denen einer Domain viele verschiedene IP Adressen zugeordnet werden. Doch anstelle das diese Assoziation bestehen bleibt, ändert sich diese einem nur dem Entwicklers des Bots bekannten Algorithmus. Fast-Flux Netzwerke unterscheiden sich in Single Flux Netzwerken und Double Flux Netzwerken. Bei Single Flux Netzwerken ändert sich ausschließlich der DNS A Resource Record (die IP-Adresse zur Domain) welche durch den Time-to-Live (TTL) Wert gesteuert wird (z.B. alle 3 Minuten eine neue IP). Abbildung 4 zeigt ein Beispiel eines Single Flux Netzwerkes, welches durch den FastFlux Tracker der Webseite abuse.ch beobachtet wird.


dnsbl.abuse.ch :: FastFlux Tracker

40 newest bots assigned to the domain **painkee.com**:

Domain	Dateadded (UTC)	IP address	Hostname	AS number	Country	Cou
painkee.com	2009-12-31 13:54:11	79.114.22.215	79-114-22-215.rdsnet.ro	8708		1
painkee.com	2009-12-31 13:29:28	82.34.216.67	cpc3-gill8-0-0-cust66.basl.cable.virginmedia.com	5089		5
painkee.com	2009-12-31 13:28:26	217.255.197.22	pD9FFC516.dip.t-dialin.net	3320		7
painkee.com	2009-12-31 12:59:54	84.31.123.34	cp520492-a.dbsch1.nb.home.nl	9143		10

Abbildung 4: Beispiel eins Single Flux Netzwerkes

Double Flux Netzwerke gehen noch einen Schritt weiter und ändern neben dem DNS A Resource Records auch den NS Resource Records (hiermit wird der autoritative DNS Server bestimmt).

2.3.2 Kommunikationsprotokolle

Die Protokolle zur Inter-Botnet-Kommunikation lassen sich in “Push“- und “Pull“-Protokolle unterscheiden. Bei Push-Protokollen besteht eine persistente Verbindung zum C&C Server und der C&C Server entscheidet den Zeitpunkt neuer Kommandos. Anders bei Pull-Protokollen, dort besteht nur eine gelegentliche Verbindung zum C&C Server. Der Bot erfragt neue Kommandos. Tabelle 3 zeigt die einzelnen Protokollarten und ihre entsprechende Zuordnung mit einem Beispiel. So wurde im August 2009 erst ein Botnetz gefunden, welches Twitter als C&C Server nutzte [Nazario 2009].

Protokoll	Push	Pull	Beispiel
HTTP		X	Twitter, eigene Webseite
Instant Messaging	X		ICQ, Jabber
IRC	X		UnrealIRCD
P2P	X	X	Overnet (Kademlia)
VoIP	X	X	Bisher nur theoretisch
Andere	X	X	Eigenes (z.B. auf Basis von ICMP

Tabelle 3: Kommunikationsprotokolle bei Botnetzen

2.4 Update-Mechanismen

Erlaubt Bot-Herdern das Patchen von Fehlern (Bugfixing) in ihren Bots. Zu dem besteht durch “Feature Adding“ die Möglichkeit den Bot nachträglich zu erweitern. Dies kann beispielsweise die Aktualisierung des Exploit-Archives vom Bot sein oder das Ändern des Kommunikationsprotokolls. Auch können neue Mechanismen nachgeladen werden.

Solche Updates wirken sich auf das gesamte oder nur einer Teilmenge des Botnetzes aus. Eine Gruppierung in Einheiten ähnlich wie beim Militär wird ermöglicht [Krogoth 2008c].

Als Folge der Updates geht auch eine Veränderung der Bot-Signatur einher. Somit werden Entdeckungen durch Antiviren-Software erheblich erschwert.

2.5 Verteidigungstechniken

Ein Bot liefert Informationen über das Botnetz und dessen Funktionsweise, welche aus Entwicklersicht zu schützen sind. Dazu bedienen sich Entwickler verschiedenster Techniken - auch kombiniert - um diese Informationen zu bewahren.

So gibt es den Ansatz, ungewollte Programme (z.B. Antiviren Software) zu beenden. Abbildung 5 zeigt eine vereinfachte Darstellung einer Antiviren Detection bzw. eines Antiviren Processkillers anhand F-Secure’s BugBear.B Analyse [F-Secure 2003].

```
void KillAV()
{
 #ifdev Win32
 const char AV_FilenamesToKill = { "_AVP32.EXE", "_AVPCC.EXE", "_
                                   "ACKWIN32.EXE", "ANTI-TROJAN.EX
                                   [...]
                                   "ZONEALARM.EXE", NULL }
  for(int i = 0; AV_FilenamesToKill[i] != NULL ;++i)
    KillProcess(AV_FilenamesToKill[i]);

 #else
 KillProcess("tcpdump");
 KillProcess("ethereal");
 KillProcess("wireshark");

 #endif
}
```

Abbildung 5: Beispiel einer AV Detection/ eines AV Processkillers

Einen anderen Ansatz verfolgen Rootkit-Techniken, bei dem sich der Bot vor dem System bzw. dessen Nutzer und seinen Anwendungen versteckt.

Auch Angriffe auf die Forscher eines Bots in Form von DDoS sind möglich [Krogoth 2008d].

3 Bot(netze) Detection

3.1 Infektionen von Bots erkennen

Infektion von Bots können lokal (local detection) und im Netzwerk (network detection) entdeckt werden.

Bei der lokalen Infektionserkennung werden alle ausführbaren Möglichkeiten des Computers in Betracht gezogen. Die einfachste Form ist die manuelle Erkennung, bei dem Benutzer Systemanomalien auffallen (z.B. hohe CPU Last).

Automatisiert wird dies von Antiviren-Software übernommen. Durch den Einsatz von reaktiven Erkennungen (signatur-based detection) und proaktiven Erkennungen (z.B. durch heuristic-based detection) werden bekannte sowie alte Bots aufgespürt. Antiviren-Software scheitert allerdings oft beim Erkennen von aktuellen Schädlingen [RUS-CERT 2010].

Neben der manuellen Erkennung und der durch Anti-Viren Software, können Host-Based Intrusion Detection Systeme (HIDS) ergänzend eingesetzt werden.

Bei der Infektionserkennung übers Netzwerk wird die klassische Netzwerkverkehrs Analyse genutzt. Über einen Mirroring-Port eines Switches, wird der Verkehr mit Programmen zur Netzwerküberwachung (z.B. mit Wireshark) analysiert.

Einen Schritt weiter geht die Kommunikationsanalyse, in dem nur ein ausgewähltes Protokoll des Netzwerkverkehrs interessiert. Netzwerk Intrusion Detection Systeme (wie z.B. Snort) stellen eine Kommunikationssignaturanalyse bereit. In dieser werden Signaturen genommen, die bekannte Muster von Bots erkennen. Der Nachteil ist, das unbekannte Bots meist völlig transparent für die Kommunikationssignaturanalyse sind.

3.2 Aufspüren von Botnetzen

Zur Erkennung von Botnetzen existieren Botnet-Detection Frameworks. Diese unterscheiden sich in der Art der betrachteten Kommunikationsprotokolle (siehe Kapitel 2.3.2 auf Seite 11) wie auch der untersuchten Netze (z.B. nur Bereiche des Internets wie das RIPE NCC ≙ europäisches Netz).

Abbildung 6 zeigt das IRC basierte Botnet-Detection Framework Rishi, welches an der RWTH Aachen eingesetzt wird.

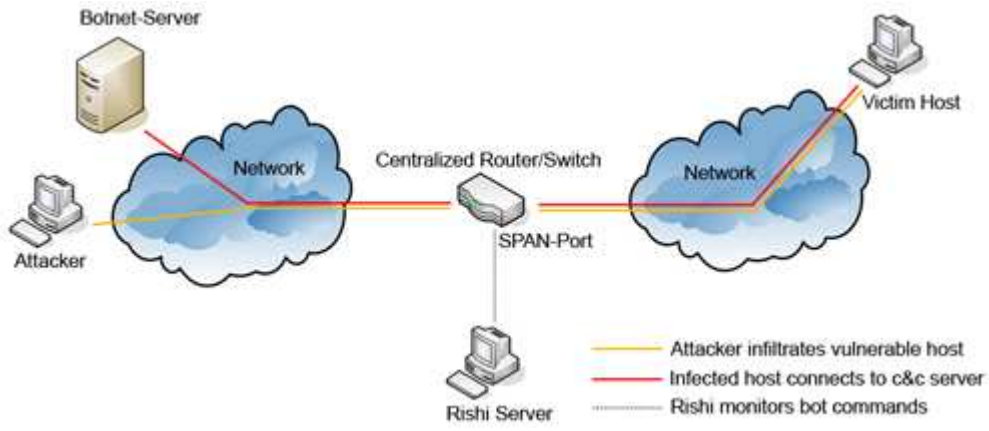


Abbildung 6: IRC-Botnetz-Erkennung an der RWTH Aachen mit Rishi [Göbel 2008]

Rishi wird an einem Switched Port Analyzer (SPAN = Mirroring Port) betrieben und wertet durch passives Sniffen des Netzwerkverkehrs Informationen, die über das IRC-Protokoll übertragen werden, anhand von regulären Ausdrücken aus. „Über ein Punktesystem, welches einzelnen Verbindungen aufgrund der gefundenen

Informationen eine Punktzahl zuordnet, wird entschieden, ob es sich um eine Verbindung zu einem IRC-Botnet handelt oder nicht.“ [RWTH Aachen 2008].

Einen ganz anderen Weg gehen die Honeypots, welche Server, Netzwerkdienste oder ein ganzes Rechnernetz simulieren. Ziel ist es Angreifer, u.a. auch Bots, dazu verleiten dieses System anzugreifen, um den Bot anschließend zu untersuchen.

Das Problem dieser Botnet-Detection Frameworks ist, dass diese teuer zu überwachen sind (dabei meint teuer auch Verbrauch von CPU-Zeit, Auslastung des Netzwerkverkehrs etc.) und in vielen Fällen auch überlistet werden können [Dagon 2005a].

Als alternativen Lösungsweg sah Dagon (im Jahr 2005) DNS basierte Erkennung, insbesondere von DnyDNS. Hierzu wurde eine Formel zur Bestimmung der Canonical DNS Request Rate aufgestellt:

$$C_{SLD_i} = R_{SLD_i} + \sum_{j=1}^{|SLD_i|} R_{3LD_j}$$

Die Formel bezeichnet das Aufsummieren der Kinder eines Second Level Domain (SLD_i) Baumes als Wurzel. Nach einer empirischen Untersuchung des Verkehrs von DynDNS stellte sich heraus, das Botnetze als DynDNS-Kunden dazu neigen viele Subdomains zu nutzen[Dagon 2005b].

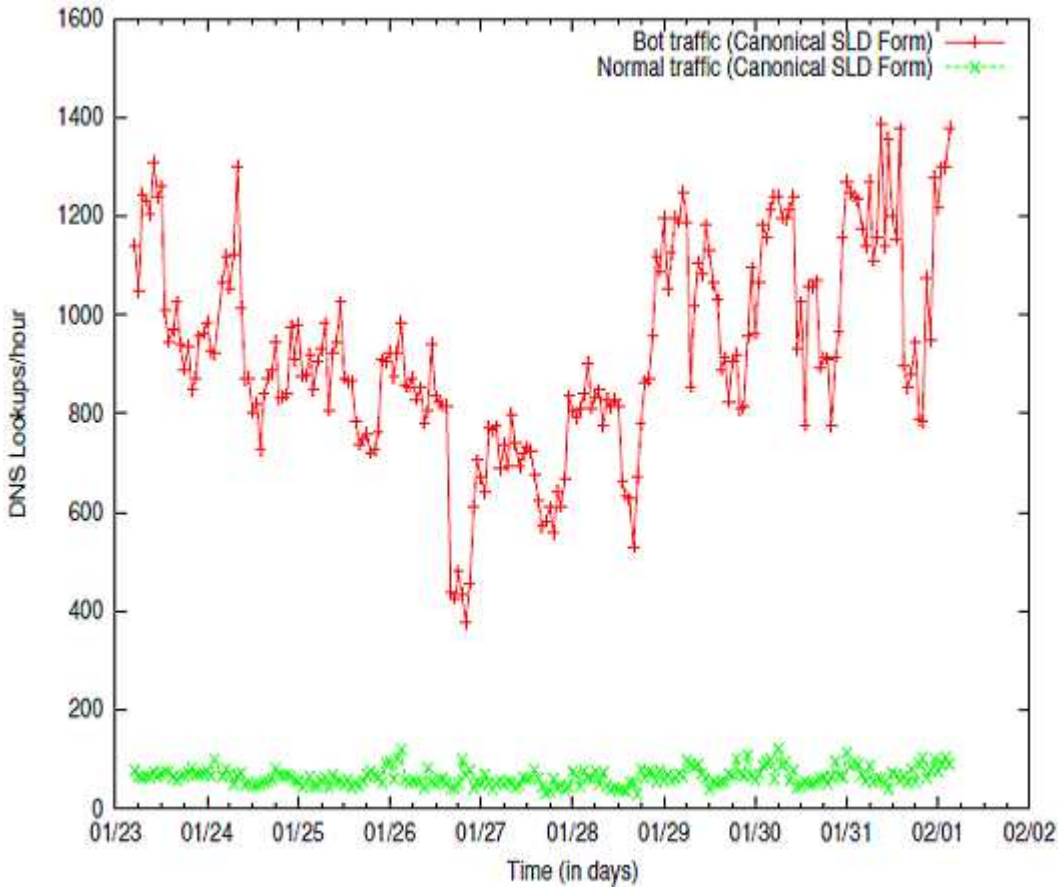


Abbildung 7: DynDNS Based Detection [Dagon 2005c]

Der Botnetz-Verkehr wird durch Division der Second Level Domain durch die Third Level Domain-Kennzahl identifiziert.

4 Entfernen der Bots

Neben Antiviren-Software existiert spezialisierte Software zum Entfernen von Bots. Tabelle 4 zeigt die von mir im Rahmen der Seminararbeit identifizierte Software zum gezielten entfernen von Bots.

Dennoch sollte beachtet werden, ein System das einmal infiziert wurde, muss als nicht vertrauenswürdig angesehen werden. Eine Neuinstallation des Systems wird dringend angeraten.

Programm	Preis	Download-Link
McAfee Stinger	Kostenfrei (Freeware)	Download
Microsoft Windows Defender	Kostenfrei (Freeware)	Download
Spybot – Search & Destroy	Kostenfrei (Freeware)	Download
TrojanHunt	Kostenpflichtig (Shareware oder 39.95 USD)	Download
Trojan Defence Suite	Kostenpflichtig (Teile des Pakets als Freeware; Preis abhängig vom konkreten Produkt)	Download

Tabelle 4: Spezialisierte Software zum Entfernen von Bots

5 Ausblick

Botnetze bleiben nach wie vor ein aktuelles Thema und werden sich in Zukunft nicht alleinig die PC Architektur als Opfersystem aussuchen. So wurde am 23. November 2009 das erste iPhone Botnetz entdeckt mit dem Namen iKeeb.B [SRI Inernational 2009].

Cloud Computing rückt immer mehr ins Licht von Botnetz-Entwicklern, so das Malware as Service sich immer weiter durchsetzen wird. So wurde am 10. November 2009 bei Googles App Engine ein Botnet von Abor Networks [Bachfeld 2009] und am 9. Dezember 2009 bei Amazon's EC2 Cloud Computing-basierten Dienst ein C&C Server des Zeus Bots entdeckt [Ferrer 2009].

Neben den neuen Entwicklungen der Bedrohung der Botnetze wurde am 8. Dezember vom BSI bekannt gegeben, dass es dem eco-Verband bei der Errichtung eines Callcenters zur Botnetz Bekämpfung unterstützt.

"Das Bundesministerium des Innern hat für die "Anti-Botnet-Initiative" des Verbandes der Deutschen Internet-Wirtschaft eco jetzt 2 Millionen Euro aus dem IT-Investitionsprogramm bereitgestellt.

Bei der Gestaltung des Beratungsangebots für Bürgerinnen und Bürger, deren Rechner ohne ihr Wissen Teil eines Bot-Netzes geworden sind, wird das Bundesamt für Sicherheit in der Informationstechnik (BSI) mit seiner technischen Expertise dem eco-Verband, dem die alleinige Federführung für das Projekt zukommt, Hilfe leisten." [BSI 2010].

6 Literaturverzeichnis

Bachfeld, D.: Neues von der Botnet-Front. Hannover 2009 http://www.heise.de/Impressum-4862.html

Binkley, J.; Schiller, C. A.: Botnets. The Killer Web Applications. Burlington 2007

BSI (Hrsg.): IT-Gipfel: BSI unterstützt eco bei Errichtung eines Callcenters zur Bekämpfung von Botnetzen. Bonn 2009 https://www.bsi.bund.de/cln_174/ContentBSI/Presse/Pressearchiv/Presse2009/ITGipfel_081209.html

BSI (Hrsg.): Bundesinnenministerium unterstützt Provider bei Botnetz-Bekämpfung. Bonn 2010 https://www.bsi.bund.de/cln_174/ContentBSI/Presse/Kurzmitteilungen/BMI_Provider_Botnetz_090210.html

Dagon D.: Botnet Detection and Response. The Network is the Infection. OARC Workshop. Georgia Tech Campus 2005a S. 23, Georgia Tech Campus 2005b S. 28, Georgia Tech Campus 2005c S. 30

Dunham, K.; Melnick J.: Malicious Bots. An Inside Look into the Cyber-Criminal Underground of the Internet. Broken Sound Parkway NW 2009

F-Secure (Hrsg.): F-Secure Virus Descriptions : Bugbear.B. Helsinki 2003 http://www.f-secure.com/v-descs/bugbear_b.shtml

F-Secure (Hrsg.): Calculating the Size of the Downadup Outbreak. Helsinki 2009 http://www.f-secure.com/weblog/archives/00001584.html

Ferrer, M. C.: Zeus “in-the-cloud”. New York 2009 http://community.ca.com/blogs/securityadvisor/archive/2009/12/09/zeus-in-the-cloud.aspx

FireEye (Hrsg.): Smashing the Mega-d/Ozdok botnet in 24 hours. Menlo Park 2009 http://blog.fireeye.com/research/2009/11/smashing-the-ozdok.html

Göbel, J: Rishi – Identifizierung von Botsdurch Auswerten von IRC Nicknamen. 2008 Mannheim S.5 http://pi1.informatik.uni-mannheim.de/filepool/publications/goebel-dfn-rishi.pdf

Kaspersky Lab (Hrsg.): Ökosystem Botnetze. Ingolstadt 2009 http://www.kaspersky.com/de/whitepaper_analysen?chapter=207717164

Krogoth (Pseudonym des Autors. Mehr ist von ihm nicht bekannt): Botnet construction, control and concealment. Looking into the current technology and analysing tendencies and future trends. New Jersey 2008a S. 24, New Jersey 2008b S. 27, New Jersey 2008c S. 32, New Jersey 2008d S. 33 http://www.shadowserver.org/wiki/uploads/Information/thesis_botnet_krogoth_2008_final.pdf

Long, J.: Welcome to the Google Hacking Database (GHDB)!. Seattle 2010 http://www.hackersforcharity.org/ghdb/

Martinez, V.: PandaLabs Report: MPack uncovered. Gran Vía de Don Diego López de Haro 2007 http://blogs.pandasoftware.com/blogs/images/PandaLabs/2007/05/11/MPack.pdf

Nazario, J.: Twitter-based Botnet Command Channel. Ann Arbor 2009 http://asert.arbornetworks.com/2009/08/twitter-based-botnet-command-channel/

SRI International (Hrsg.): An Analysis of the iKee.B (Duh) iphone Botnet. Menlo Park 2009 http://mtc.sri.com/iPhone/

RUS-CERT (Hrsg.): Entfernung von Bots. Universität Stuttgart 2010 http://cert.uni-stuttgart.de/doc/netsec/bot-entfernung.php

RWTH Aachen (Hrsg.): Rishi - IRC-Botnet-Erkennung. Aachen 2008 http://www.rz.rwth-aachen.de/aw/cms/rz/Themen/unsere_dienste/kommunikation/netzbetrieb/allgemeine_informationen/~qbb/rishi/?lang=de

Zors, Z.: Top 10 botnets and their impact. Rijeka 2009 http://www.net-security.org/secworld.php?id=8599