Moderne Botnetze

Anton Hattendorf
Betreuer: Marc Fouquet
Seminar Future Internet SS2009
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik
Technische Universität München
E-Mail: hattendo@in.tum.de

***Kurzfassung*— Botnetze sind ein Zusammenschluss von mehreren verteilen Rechner. Sie erfüllen Aufgaben, die ihnen übertragen werden. Dies sind unter anderen der Versand von SPAM und DDoS Angriffe. Um ihre Aufgaben zu koordinieren verfügen sie über eine Kommunikationsschnittstelle. Die ersten Botnetze haben IRC zur Kommunikation verwendet. Dies hatte den Nachteil, dass sie zentrale Infrastrukturen benötigen, welche leicht deaktiviert werden konnte. Deshalb werden die IRC-basierten Netze von anderen Strukturen abgelöst. Die folgende Ausarbeitung beschäftigt sich mit den Strukturen von modernen Fast-Flux und Peer-to-Peer Botnetzen. Weiterhin werden die Slapper und Storm Botnetze genauer betrachtet. Zu Abschluss wird erläutert, wie man gegen Botnetze vorgehen kann.**

***Schlüsselworte*— Botnetze, Fast-Flux, P2P, Slapper, Storm, Conficker**

I. Einleitung

A. Was sind Bots

Bots sind Teile eines verteilten Softwaresystemen. Die Software eines Bots wird auf mehreren physisch unabhängigen Rechnern ausgeführt. Die Bots kommunizieren über Netzwerke miteinander und koordinieren sich auf diese Art zu einem sogenannten Botnetz (botnet). Ein Bot verfügt über eine Steuerschnittstelle (siehe Abschnitt II-B) und wird in der Regel von einer übergeordneten Einheit, dem sogenannten Mutterschiff (mothership), gesteuert und überwacht. Die ersten Bots sind um 1993 im IRC entstanden [1]. Die Betreiber von Botnetzen werden auch „Bot Herder“ genannt.

Das Ziel eines Botnetzes ist nicht zwangsweise bösartig: gutartige Bots sind z.B. Webcrawler wie der Googlebot oder der NickServ im IRC. Diese gutartigen Bots verhalten sich kooperativ gegenüber den Web-Applikationen. Webcrawler durchsuchen das WWW und erstellen gemeinsam eine Datenbank, auf die die jeweiligen Suchmaschinen dann zugreifen. Dabei kooperieren sie mit den Webservern, indem sie z.B. den Robots Exclusion Standard [2] befolgen oder die genutzte Bandbreite der durchsuchten Seiten beschränken.

Bösartige Botnetze sind z.B. für SPAM-Mails, Phishing und DDoS Attacken verantwortlich. Sie werden für kriminelle oder zumindest fragwürdige Zwecke eingesetzt und verhalten sich in der Regel nicht kooperativ zu Dritten. Die Bots von bösartigen Botnetzen werden in der Regel auf anderen Rechnern ohne die Genehmigung des Betreibers installiert.

Die Systeme des Internets (Server, Firewalls, ...) reagieren auf die Handlungen von gutartige und bösartige Bots verschieden. So werden gutartige Bots häufig unterstützt, während bösartige Bots bekämpft werden. Dem Googlebot werden z.B. Inhalte zur Indizierung verfügbar gemacht, welche anderen Benutzern nicht unbedingt zugänglich sein sollten [3] (z.B. Bezahlinhalte). Im Gegensatz dazu werden SPAM-Bots dadurch, dass sie auf Blacklisten eingetragen werden, an der Erfüllung ihres Zwecks gehindert.

Beide Arten von Bots benutzen zwar teilweise die gleichen grundlegenden Technologien (verteilte Systeme, Steuerschnittstelle), werden aber von den Systemen des Internets verschieden behandelt. Deshalb wird sich diese Ausarbeitung im Folgenden nur mit bösartigen Bots beschäftigen und der Begriff Bot wird synonym für bösartiger Bot verwendet.

B. Abgrenzung zu Viren, Würmern, etc.

Die Gemeinsamkeit von Viren, Würmern und Bots ist, dass es sich um unerwünschte Software handelt. Eine Schadroutinen ist bei allen nicht notwendig, aber häufig anzutreffen. Häufig integrieren sie sich auch in das Betriebssystem, um sicherzustellen, dass sie auch nach einem Neustart des Systems wieder aktiv werden.

Viren: Sie infizieren fremde Programme, indem sie ihren Code in den Code der Programme integrieren. Bei compilierten Programmen oder Bibliotheken läuft dies auf der Ebene von Maschinencode ab. Sie können sich aber auch in interpretierten Skripten, Makros von Office-Programmen oder Boot-Sektoren von Datenträgern einnisten.

Ein Virus verbreitet sich durch die Weitergabe seines Codes in den infizierten Dateien. Sobald der Virus auf einem System ausgeführt wurde, infiziert er andere Programme.

Seit Beginn dieses Jahrtausends sind Viren durch die zunehmenden Vernetzung über das Internet in den Hintergrund getreten und durch Würmer abgelöst worden [4].

Würmer: Würmer verbreiten sich, indem sie aktiv andere Rechner über das Netzwerk infizieren. Dies kann auf verschiedene Arten ablaufen: E-Mails, Sicherheitslücken aller Art oder Dateifreigaben. Zusätzlich können sich Würmer noch wie Viren durch das Infizieren von Programmen verbreiten.

Bekannte Würmer sind z.B. der „ILOVEYOU“, welcher sich im Jahr 2000 mit Hilfe von E-Mails rasant ausbreite, oder der „Code Red“, der eine Sicherheitslücke im Microsoft IIS nutzte.

Die Verbreitung von Würmern ist nicht auf das Internet beschränkt. So verbreitete sich der SymbOS.Commwarrior.A über Bluetooth und MMS auf Handys und PDAs [5].

Im Gegensatz zu Bots verfügen Würmer über keine Steuerschnittstelle (siehe II-B) und können deshalb nicht überwacht oder ferngesteuert werden. Allerdings verwenden viele Bots Wurm-Techniken um sich zu verbreiten, weshalb der Unterschied zwischen Bot und Wurm fließend ist und nicht fest definiert ist. Viele Anti-Viren-Softwarehersteller verwenden auch für Bots den Begriff Wurm.

II. Bot Techniken

A. Infektion

Die meisten Bots verbreiten sich, indem sie andere Rechner über das Internet infizieren. Dabei gibt es verschiedene Möglichkeiten. Diese Infektionsmethoden werden auch von Würmern verwendet. Typische Methoden sind:

Software-Programmierfehler: Der Bot verwendet ein Exploit für eine Sicherheitslücke in einem fremden System um auf diesem gestartet zu werden. Dieser Exploit nutzt z.B. einen Buffer-Overflow aus. Dies betrifft sowohl Server- als auch Client-Systeme.

Auf Server-Systemen wird i.d.R. der Daemon, welcher einen Dienst zu Verfügung stellt, angegriffen. Dies sind meistens Webserver, da deren Dienste für das gesamte Internet angeboten werden. Neben der eigentlichen Server Software können auch Webapplikationen angegriffen werden.

Clients werden meistens durch Webseiten mit schadhaften Inhalten angegriffen. Diese Angriffe basieren häufig auf ActiveX Controls oder JavaScript. Allerdings ist es auch möglich, einen Client direkt anzugreifen, z.B. über eine Windows-Freigabe.

Software-Designfehler: Im Unterschied zu Programmierfehlern handelt es sich hier um ein Feature, welches bewusst in ein Programm eingebaut wurde und für bösartige Zwecke missbraucht werden kann. Dazu gehört z.B. automatisches Öffnen von Dateianhängen in alten Mail-Clients.

Schwache/keine Passwörter: Auf einem Server ist ein privilegierter Dienst nur mit einem schwachen oder gar ohne Passwort abgesichert ist. Ein Bot kann z.B. durch eine Wörterbuch-Attacke das Passwort in Erfahrung bringen. Über diesen Zugang kann sich der Bot auf dem Server installieren.

Social Engineering: Hierbei überzeugt der Bot den Benutzer davon, ihn auszuführen. Dies geschieht in der Regel, indem das Interesse des Benutzers geweckt wird. Typische Formen dieses Angriffes sind Mailattachments, die eine wichtige Nachricht enthalten sollen, oder Programme, die für den Benutzer interessant sind und heruntergeladen werden können. Diese Angriffe sind, wenn sie gezielt ausgeführt werden, auch mit technischen Maßnahmen nur schwer abzuwehren. Sie greifen den Benutzer auf sozialer Ebene an, indem sie Vertrauen zu ihm aufbauen und dieses missbrauchen [6].

Zu dieser Methode gehören sowohl das Verschicken des Bots mit SPAM-Mails als auch die Integration des Bots in Anwendungen wie Spielen oder Tools (Trojaner).

B. Steuerschnittstelle

Diese Steuerschnitstelle, auch C&C (Command & Control) genannt, unterscheidet Bots von Würmern und Viren. Der Betreiber eines Botnetzes steuert über sie die Aktionen des Botnetzes.

Typische Kommandos sind:

- Rechner infizieren (vergl. II-A)
- Update herunterladen (vergl. II-C)
- Schadroutinen ausführen (vergl.II-D)

C. Modularer Aufbau

Viele Bots sind modular aufgebaut. Das heißt, dass sie über einen Update-Mechanismus mit neuen Code versorgt werden können und dadurch ihre Funktionalität erweitern können. Die Art dieser Updates ist dabei sehr verschieden. Es kann sich um ausführbare Programme, Shared Libraries oder Skripte handeln.

D. Schadroutinen/Anwendungen

Die Struktur eines Botnetzes macht dieses für viele illegale Anwendungen interessant. Viele dieser Anwendungen können auch auf den Schwarzmarkt gebucht werden.

1) SPAM: Durch den Versand von SPAM kann das Botnetz zwei Ziele erreichen:

- eigene Verbreitung durch die Infektion anderer Rechner
- kommerzielle Ziele durch den Versand von Werbemails

Die Bots bekommen vom Netzwerk ein Template für eine SPAM Mail übermittelt und versenden auf dessen Basis SPAM Mails.

Die E-Mail-Adressen, an die der Bot den SPAM versenden soll, werden entweder vom Botnetz übertragen oder der Bot durchsucht den infizierten Rechner und/oder das Internet nach E-Mail Adressen. Außerden kann er E-Mail-Adressen zufällig generieren, indem z.B. Kombinationen aus häufigen Namen und verschiedenen Domains ausprobiert werden.

Dadurch, dass die Bots den SPAM versenden ist es schwieriger, diesen zu filtern. Die verschiedenen Absender-IP-Adressen machen das Blacklisting der versendenden Hosts sehr aufwendig.

Ein Bot verbreitet sich über den Versand von SPAM, indem er entweder seinen eigenen Code im Anhang der Mail plaziert, oder die Mail einen Link zu einem Webserver enthält, auf welchem der Bot liegt. Letztere wurde z.B. vom Storm verwendet [7]. Die Ziele des SPAM Versands sind neben der eigenen Verbreitung des Bots häufig auch kommerzieller Natur. Kommerzielle Ziele sind der Versand von Werbemails für Drogen oder Medikamente (häufig Viagra etc.). Weiterhin werden mit ihnen Opfer für Finanzbetrug gesucht (z.B. Nigeria Mail oder Phishing).

2) DDoS: Bei einem Distributed Denail-of-Service(DDoS) Angriff werden von mehreren Bots Pakete zu einem Ziel-Host gesendet, so dass die gesamte Bandbreite des Ziels blockiert ist. Dadurch steht keine Bandbreite mehr für andere Dienste zu Verfügung, und das Ziel ist nicht zu erreichen. Auch wenn die einzelnen Bots häufig nur über einen sehr schmalen

Uplink (128 kBit/s bei T-DSL 1000) verfügen, so können sie, wenn sie ihre Attacke zeitlich koordinieren, mit ihrer gesamten Bandbreite durchaus beachtlichen Schaden anrichten.

Die genaue Durchführung der Attacke kann sehr vielfältig sein. Das Spektrum geht von einfachen Flooding bis zu SYN-Flood-Attacken.

Ziel dieser Angriffe ist es, konkurierende Botnetze oder Betreiber anderer Internet Dienste durch das Blockieren ihrer System zu schädigen oder Lösegeld dafür zu erpressen, dass eine solche Attacke nicht durchgeführt wird. Letztes ist dem Wettanbieter mybet.com während der Fußball Europameisterschaft 2004 passiert. Die Erpresser forderten 15000 US-Dollar, und legten, weil mybet.com nicht zahlen wollte, deren Website für 16 Stunden still [8].

3) Phishing: Ein Bot kann auf verschiedene Arten an Phishing-Angriffen beteiligt sein: Einerseits können die Bots zum Versenden der Mails verwendet werden, welche die Benutzer auf Phishing-Seiten locken. Auf den anderen Seite kann das Botnetz auch die eigentliche Phishing Seite zu Verfügung stellen und die Daten sammeln. Dafür muss der Bot aber von Internet aus erreichbar sein. Dies geschieht meistens auf der Basis von Fast-Flux Netzwerken (vergl. III-C).

Mit solchen Angriffen bekommt der Bot Herder Zugriffsdaten für Konten und kann diese „abräumen“.

4) Sammeln vertraulicher Daten: Der Bot durchsucht den infizierten Rechner nach vertraulichen Daten und übermittelt diese an das Botnetz. Bei diesen kann es sich um Adresslisten, Schlüsseln für Programme, gespeicherte Passwörter, Seriennummern etc. handeln.

5) Proxy: Der Bot fungiert als Proxy zu beliebigen anderen Zielrechnern und ermöglicht damit dritten den Zugriff auf die Zielrechner ohne das diese mit ihrer IP für den Zielrechner erkennbar sind.

6) Datenspeicher: Daten können in dem Netz gespeichert werden, und von verschiedenen Benutzer heruntergeladen werden. Dabei kann es sich sowohl um von den Bots gesammelte Daten handeln als auch um Daten die der Betreiber seinen Kunden zu Verfügung stellt. Typische Daten sind Raubkopien von Programmen und pornografisches Bildmaterial.

E. Eigenschutz

Moderne Würmer und Bots vertuschen ihre eigene Existenz. Die ersten Schädlinge haben dies versucht, indem sie ihre Dateien als versteckt markieren. Dies reicht inzwischen aber nicht aus, um Virenscanner zu überlisten.

Effektiver ist die Verwendung eines Root-Kits. Da wird so in das System so eingegriffen, dass z.B. bestimmte Dateien nicht im Verzeichnis aufgelistet werden oder bestimmte Prozesse nicht angezeigt werden. Dadurch das ein Root-Kit auch seine eigene Existenz verschleiern kann, sind solche Schädlinge oft nur durch eine Analyse des Betriebssystem durch ein spezielles von CD gebootetes System zu finden. Ein Root-Kit kommt z.B. bei einigen Versionen des Storm Bots oder dem Srizbi Bot zum Einsatz [7], [9].

Andere moderne Eigenschutz Varianten sind die Manipulation von Virenscannern, so das diese zwar scheinbar funktionieren, aber den Schädling übersehen oder die Verwendung von Hardware-Virtualisierungsfunktionen.

Hardware-Virtualisierung erlabt es, einen PC in mehrere virtuelle, voneinander unabhängige Maschinen zu unterteilen. Dadurch können Betriebssystem und Virenscanner in einer anderen Umgebungen als der Bot laufen und können die Existenz des Bots nicht feststellen [10]. Es ist allerdings noch kein Bot bekannt, welcher Hardware-Virtualisierung nutzt.

III. Botnetze

A. Einführung

Da die meisten Botnetze in der illegalen Aktivitäten nachgehen, sind zentrale Strukturen meist nicht besonders effektiv. Sobald der Server abgeschaltet wird werden die Bots nutzlos. Die Struktur des Netzwerk ist auch essentiell für die Effektivität der Steuerschnittstelle.

B. IRC

Die ersten Botnetze waren IRC [11] basiert. Bei diesen Botnetzen agierte ein IRC Server als zentrale Instanz und alle Bots haben sich sich mit einem Kanal des IRC Servers verbunden.

Zum Steuern des Netzes verbindet sich auch der Betreiber mit dem IRC Server und sendet seine Kommandos an den IRC Kanal. Diese werden von den Bots empfangen und ausgeführt [1], [12].

Der Vorteil dieser Botnetze ist, dass keine spezielle Server Software notwendig ist und entsprechenden Client Bibliotheken vorhanden sind. Weiterhin können neben den, durch den Botnetz Betreiber aufgesetzten, IRC Server auch Kanäle in öffentlichen IRC Netzen für die Kommunikation der Bots verwendet werden [13].

Nachteilig für IRC basierte Bots ist, dass falls der IRC Server bzw. der IRC Kanal abgeschaltet wird, es nicht mehr möglich ist, das Botnetz zu kontrollieren. Deshalb sind die IRC Server, die für solche Zwecke eingesetzt werden, meistens in Fernost platziert.

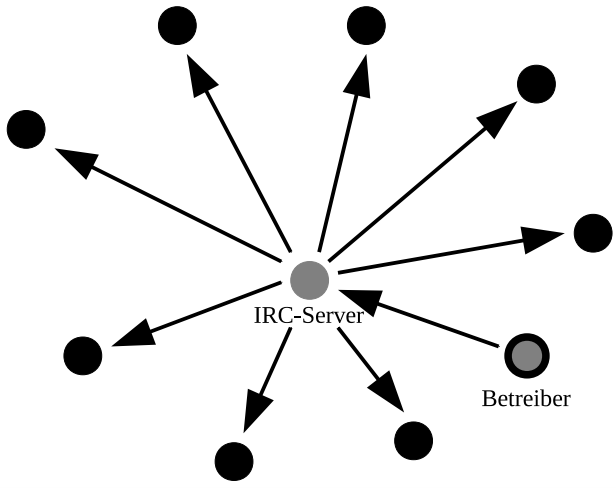

Abbildung 1. Struktur eines IRC basierten Botnetzes

C. Fast-Flux

1) Grundlagen: Fast-Flux Netzwerke nutzen das DNS [14], [15] um ihre Dienste zu Verfügung zu stellen.

Mit Hilfe eines Fast-Flux Netzwerkes soll erreicht werden, dass ein DNS Eintrag (z.B. www.fastflux.com) auf auf viele IP Adressen verweist (A Records). Die IP Adressen werden in einem Round Robin Verfahren mit hoher Frequenz getauscht und mit einer kurzen Lebensdauer (TTL) versehen. Bei dem Auswechseln der A Records berücksichtigt das Fast-Flux Netzwerk Verfügbarkeit und die Bandbreite der Hosts [16].

Ein Benutzer, der einen Host beim DNS abfragt, bekommt bei jeder Anfrage andere Ergebnisse. Diese Eigenschaft des DNS wird von großen Websites zum Lastausgleich benutzt. In Fast-Flux Botnetzen wird dieses nun gesteigert zu einem Lastausgleich zwischen Tausenden von Bots.

Die DNS Einträge verweisen auf Bots des Botnetzes. Diese Bots agieren häufig nur als Proxy für das Mutterschiff, welches dadurch im Hintergrund versteckt bleiben kann und von außen nicht gefunden werden kann. Das Mutterschiff ist das steuernde Element in dem Fast-Flux Netzwerk und übernimmt quasi die Funktion des C&C von IRC basierten Botnetzen.

Die Bots der ersten Reihe sind dabei entsorgbar, d.h. ein Verlust einzelner Bots beeinflusst das Botnetz nicht. Bei ihnen handelt es sich meisten um gehackte Maschinen, welche keinerlei Bezug zu den Betreibern des Botnetzes haben.

Ein Bot, der auf einen Rechner aktiviert wird, fragt beim DNS eine Domain ab. Er bekommt vom DNS IP Adressen von verschiedenen Proxys und baut eine Verbindung zu einem auf. Über diese Verbindung registriert er sich bei dem Netzwerk. Der Proxy leitet diese Information weiter an das Mutterschiff und schickt Kommandos zurück. Die Kommunikation mit dem Mutterschiff läuft oft über HTTP [17] da dieses von vielen Firewalls und Proxys anstandslos weitergeleitet wird.

Fast-Flux Netze könne durch deaktivieren der Domain stillgelegt werden. Dies geschah im November 2008 mit dem Srizbi Botnetz [18]. Die Zentral verwalteten Domain-Namen sind eine kritische Ressource des Botnetzes. Die meisten Bots verfügen deshalb über Ersatz-Domains, die sie in einem solchen Fall verwenden können. Weitere Details zu diesem Vorgehen sind in IV-B am Beispiel von Srizbi beschrieben.

Man unterscheidet zwischen Single-Flux und Double-Flux Netzwerken. Der Unterschied zwischen beiden ist, dass bei Double-Flux auch der DNS-Server, rotiert wird.

2) Single-Flux: Bei Single-Flux Netzwerken werden in den Name-Server für die Fast-Flux Domain regelmäßig neue Bots als A Record eingetragen. Das Eintragen erfolgt meistens auf Basis der gerade verfügbaren Bandbreiten der aktiven Bots. Die genauen Kriterien hängen von der Implementierung des Botnetzes ab und werden im Hintergrund von den Steuerroutinen getroffen.

Für die Domain wird ein sogenannter „Bullet Proof Domain Name“ benötigt. Dabei handelt es sich eine Domain, die bei einen Registrar registriert ist welcher diese nicht aufgrund von Beschwerden einfach abschaltet. Solche Domains sind bereits für 100 US Dollar pro Jahr zu bekommen und können anonym registriert werden [19].

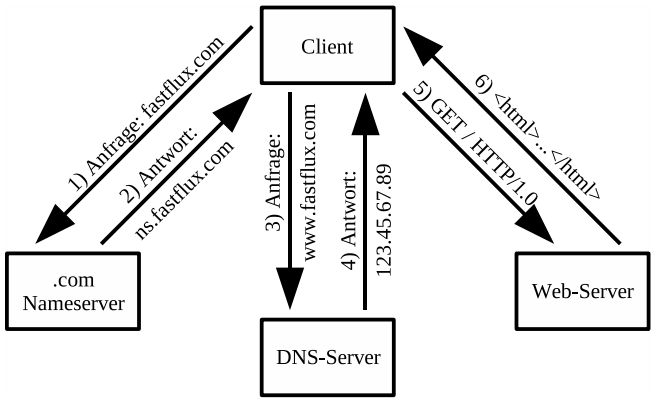

Abbildung 2. Normale Abfrage einer Website

Die Name-Server für Single-Flux Domains müssen besonders geschützt werden, da sie eine kritische Ressource sind. Deshalb befinden sie sich meistens in den Ländern wo die Strafverfolgung für „digitale Kriminalität“ nicht so ausgereift ist [19]. Dadurch wird das Abschalten solcher Domains deutlich erschwert.

Der Typische Kommunikationsverlauf für die Abfrage eine Single-Flux Domain ist in Abbildung 3 zu sehen.

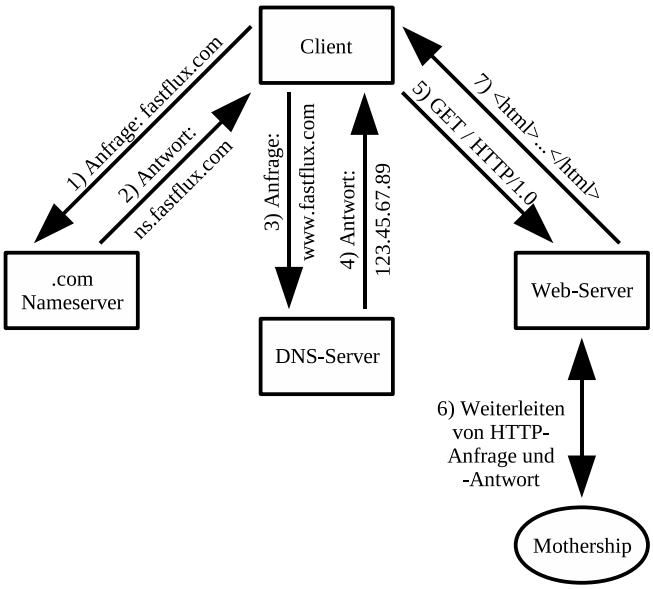

Abbildung 3. Abfrage einer Single-Flux Website

3) Double-Flux: Double-Flux Netzwerke wirken dem Problem entgegen, dass Name-Server für Single-Flux Domains eine kritische Ressource sind. Dies wird erreicht, indem auch die Nameserver rotieren. Diese Nameservern beziehen ihre Informationen dann auch vom im Hintergrund agierenden Mutterschiff.

Die Frequenz, mit der die Nameserver rotieren, ist deutlich geringer als Frequenz der Hosts. Die Frequenz liegt im Stundentakt. Es wird dennoch ein kooperativer Registrar gebraucht,

welcher eine automatisierte Schnittstelle für die Aktualisierung der NS Einträge bereitstellt und sich nicht daran stört, dass dieses im Stundentakt passiert [16], [19].

In Abbildung 4 ist ein Beispiel für die Kommunikation mit einem Double-Flux Netzwerk zu sehen.

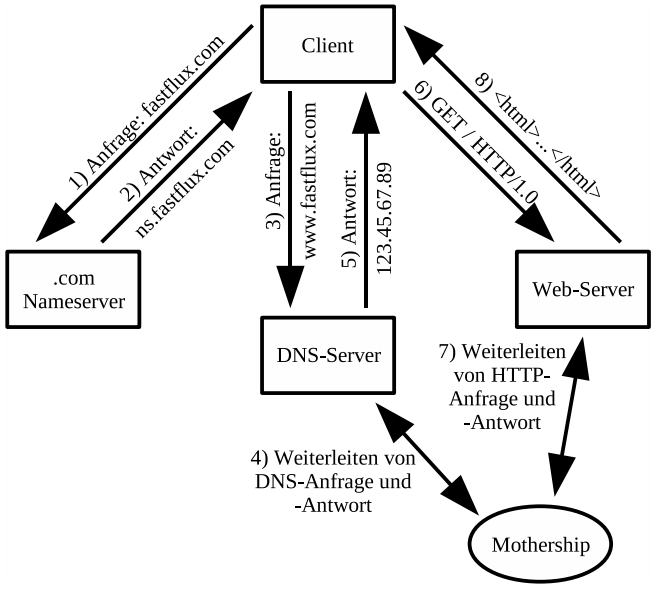

Abbildung 4. Abfrage einer Double-Flux Website

D. P2P

1) Einführung: Peer-to-Peer Netze sind Netze, welche von zentralen Strukturen unabhängig sind. In ihnen gibt es keine klassische Unterscheidung zwischen Client und Server und in der Regel können die Ressourcen von allen teilnehmenden Knoten gleichberechtigt genutzt werden. Die Teilnehmer in einen P2P Netz organisieren sich selbst, d.h. sie bauen selbst ihre Struktur auf und kümmern sich auch um deren Aufrechterhaltung. Nachrichten werden innerhalb des Netzes von Knoten zu Knoten weitergegeben. Typischerweise haben alle Knoten eine gleichartige Implementierung [20].

Für Botnetze ist die Unabhängigkeit von zentralen Strukturen von besonderen Vorteil. Netze mit zentralen Strukturen sind durch das Abschalten der zentralen Instanzen in viele kleine Inseln zufallen und die Bots können nicht mehr genutzt werden. Dadurch sind Botnetze mit zentralen Strukturen Providern und Regierungen „ausgeliefert“ und bieten auch Ansatzpunkte für die Strafverfolgung.

Ein P2P Netz ist in dieser Hinsicht deutlich robuster: es verkraftet auch den Ausfall einer größeren Menge an Knoten. Da alle Knoten gleichberechtigt sind ist auch nicht ersichtlich, hinter welchen sich das C&C verbirgt. Der Betreiber des Botnetzes bindet sich wie ein normaler Knoten in das Botnetz ein und sendet dann seine Kommandos ab. Aus Sicht des Botnetzes sieht der Betreiber wie ein normaler Knoten aus.

2) Bootstrapping: Der Vorteil, dass P2P Botnetze ohne zentrale Instanzen auskommen, bringt aber auch einen Nachteil mit: das Bootstrapping. Ein neuer infizierter Rechner, der sich

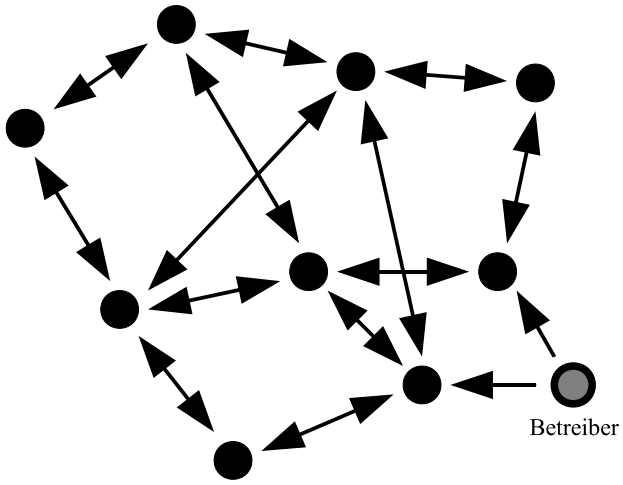

Abbildung 5. Struktur eines P2P Netzes

in das Botnetz einbinden möchte, muss erstmal Kontakt zu mindestens einen Knoten des Botnetzes aufnehmen. Von diesem kann der Knoten dann Kontaktadressen weiterer Knoten erhalten.

mitgelieferte Nachbarn: Bei der Infektion eines Rechners kann dem Schädling eine Liste mit Knoten mitgegeben werden, welche für die erste Kontaktaufnahme verwendet werden soll. Dieses Verfahren wird allerdings problematisch, wenn ein Knoten längere Zeit offline ist und alle bekannten Knoten zwischenzeitlich neue IP-Adressen bekommen haben. Dem kann durch eine hohe Anzahl von benachbarten Knoten entgegengewirkt werden. Das Slapper Botnetz verwendete ein solches Verfahren [21].

zentrale Bootstrapping Instanz: Für das initiale Bootstrapping eines Botnetzes kann auch eine zentrale Instanz verwendet werden. Diese ist dann der Rendezvous Punkt für das Botnetz.

nutzen anderer P2P Netze: Phatbot verwendet das Gnutella Netzwerk zum Bootstrapping: er verbindet sich mit dem Gnutella Netzwerk, verwendet aber einen andere Port. Dadurch ist er von anderen Gnutella Clients unterscheidbar und kann andere Knoten finden um sein eigenes Netzwerk zu erreichen [22].

IV. Beispiele für Botnetze

A. Agobot

Der Agobot wurde im Jahr 2003 von dem Deutschen Axel „Ago“ Gembe entwickelt und in Umlauf gebracht [23]. Er ist in C++ geschrieben und sehr Modular aufgebaut. Der Quelltext des Wurms wurde im April 2004 veröffentlicht [24].

Agobot ist auch unter den Namen Gaobot bekannt. Weiterhin gibt es viele Varianten wie den Phatbot, Forbot oder dem XtrmBot. Insgesamt gibt es mehr als tausend Varianten des Bots. Dies ist vor allem bedingt durch die Veröffentlichung des Quelltextes und dem modularen Aufbau [1].

Agobot infiziert ein System, indem er sich in die System Verzeichnisse kopiert und in der Registry Einträge anlegt,

die sicherstellen, dass er bei Systemstart ausgeführt wird. Er infiziert andere Systeme indem er versucht, sich auf ihre Windows-Freigaben zu kopieren oder Exploits ausnutzt [25].

Die meisten Varianten des Agobot haben eine IRC basierte Kommunikation. Er besitzt allerdings auch Funktionen für eine P2P Kommunikation. Diese war aber nicht besonders effektiv und wurde so gut wie nicht verwendet [1]. Die Agobot Variante Phatbot verwendet auch eine P2P Kommunikation, welche allerdings nicht auf der P2P Kommunikation des Agobot basiert, sondern eine WASTE Implementierung verwendet. Diese skaliert aber nicht auf große Netzwerke [22].

Nach der Infektion verbindet sich der Agobot mit mehreren einprogrammierten IRC Servern. Weiterhin ist er in der Lage, den Betreiber des Botnetzes zu identifizieren.

Die von Agobot unterstützten Kommandos sind sehr vielfältig: Es gibt Befehle um die Steuerschnittstelle der Bots zu beeinflussen. So kann der Bot angewiesen werden, einen anderen IRC Server zu verwenden oder den Kanal zu wechseln. Weiterhin gibt es Befehle um Dateien herunterzuladen und auszuführen, DoS Angriffe verschiedenster Art durchzuführen, verschiedene Proxys einzurichten und Prozesse auf dem Rechner zu beenden [13]. Der Befehlsumfang kann durch weitere Module erweitert werden.

B. Srizbi

Srizbi ist ein Schädling, der bei seinem Ausbruch im Juni 2007 als einer der innovativsten galt: Er arbeitet komplett im Kernel Mode. Weiterhin versteckt er sich mit Hilfe eines Root-Kits und versendet Spam. Das Srizbi Botnetz war im Frühjahr 2008 für 60 Milliarde SPAM Mails am Tag verantwortlich [26].

Nach der Infektion eines System verursacht Srizbi keinerlei Aktivität im User Mode. Er Installiert sich als Treiber und manipuliert die Netzwerk Treiber so dass er direkt über sie Pakete versenden kann. Dadurch kann er sogar lokale Firewalls und Sniffer umgehen und seine eigenen Pakete vor dem System verstecken [9].

Das Srizbi Netzwerk konnte im November 2008 durch die Abschaltung des Providers McColo vorübergehend stillgelegt werden. Dies hatte zu Folge, dass das SPAM aufkommen um bis zu 90 % reduziert werden konnte [18], [27].

Dadurch, dass die Bots den Kontakt zum Mutterschiff verloren hatten, war es ihnen nicht mehr möglich, neue Kommandos abzurufen. Allerdings verfügt der Srizbi Bot über eine Notfallkommunikation. Er berechnet alle 72 Stunden vier alterative Domains, welche er für einen Verbindungsaufbau ausprobiert.

Der Sicherheitsdienstleister FireEye hat diesen Algorithmus entschlüsselt und für eine gewisse Zeit diese Domains registriert. Dadurch konnte die erneute Kontaktaufnahme der Bots mit dem Mutterschiff verhindert werden. Die Registrierung der Domains verursachte allerdings Kosten in der Höhe von 4000 US Dollar pro Woche weshalb das Vorhaben aufgegeben werden musste. Kurz darauf wurde eine der Domains wieder von dem Botnetz Betreiber registriert und das SPAM Aufkommen stieg wieder an [28], [29].

FireEye hätte die Bots sogar anweisen könne, sich selbst zu deaktivieren. Dies war ihnen aus rechtliche Gründen allerdings nicht möglich. Weitere Erläuterungen zu den rechtlichen Rahmenbedingung sind in Abschnitt V-C erläutert.

C. Slapper

1) Übersicht: Der Slapper Bot wurde am 13. September 2002 in Rumänien das erste mal gesichtet. Er basiert auf dem Apache Scalper Bot, welcher ebenfalls 2002 in Umlauf kam. Slapper hat aber verbesserte Netzwerkeigenschaften und einen anderen Angriffscode.

Slapper dringt in den Apache Webserver auf IA32 Linux Systeme über einen OpenSSL Exploit ein. Es waren mindesten sechs verschiedene Linux Distributionen und neun Unterversionen des Apache Webserver betroffen.

Als Schadroutine verfügte er über die Möglichkeit, an DDoS Attacken teilzunehmen [30].

2) Architektur: Der folgende Absatz beschreibt die Architektur des Slapper Botnetzes und basiert auf einen Artikel von Iván Arce and Elias Levy [21].

Das Slapper Botnetz ist ein unstrukturiertes P2P Netz. Seine Hauptaufgabe ist, den Ursprung einer Nachricht durch mehrfaches Weiterleiten zu verschleiern. Das Protokoll basiert auf UDP und verwendet den Port 2002. Es verfügt noch über eine zusätzliche Sicherungsschicht die den Verlust von Nachrichten abfängt. Die Knoten werden in dem Netzwerk über ihre IP Adressen identifiziert.

Jede Nachricht im Slapper Netz wird durch eine zufällig gewählte Sequenznummer identifiziert. Ein Knoten verwaltet eine Liste mit den letzten 128 Nachrichten, die er empfangen hat. Wenn er eine Nachricht empfängt, deren Sequenznummer bereits in der Liste ist, wird diese verworfen. Weiterhin haben die Nachrichten eine ID, welche die Zuordnung einer Antwort zu der ursprünglich Nachricht ermöglicht.

Eine Nachricht, die durch das Netzwerk an einen bestimmten Knoten gesendet werden soll, enthält neben der IP Adresse des Ziel-Knotens auch einen Zähler, welcher die Anzahl der noch durchzuführenden Hops angibt. Wenn dieser Zähler Null ist, wird die Nachricht an die entsprechende IP Adresse weitergeleitet. Andernfalls wird die Nachricht mit um eins erniedrigten Zäher an zwei andere Knoten weitergeleitet. Die Nachricht wird also bei jeder Weiterleitung dupliziert. Der Maximalwert für den Zähler ist 16, üblicherweise wird aber fünf verwendet. Doppelt bei Empfänger ankommende Nachrichten werden verworfen.

Es ist auch möglich, ein Broadcast an das gesamte Netz zu senden. Diese Nachricht enthält keine Empfänger. Ein Broadcast wird allerdings nicht direkt an das gesamte Netz gesendet, sondern immer nur an jeweils zwei weitere Knoten. Wenn ein Knoten die Broadcast Nachricht bereits erhalten hat, wird sie nicht nochmal verteilt. Es ist nicht garantiert, dass alle Knoten eine Broadcast Nachricht erhalten.

Das Netz sogt dafür, dass sich möglichst alle Knoten kennen. Dies geschieht indem regelmäßig Nachrichten in das Netz gesendet werden und Knotenlisten ausgetauscht werden. Es gibt kein Entfernen von Knoten aus diesen Listen.

D. Storm

1) Übersicht: Das Storm Botnetz existiert seit ca. Ende 2006. Es sind seitdem viele Varianten des Bots erscheinen. Er verbreitete sich Anfangs über E-Mail Anhänge von Spam-Mails. Später beinhalteten die Mails nur noch Links, und der Benutzer wurde auf Webseiten gelockt und wo er den Bot herunterladen musste oder das Opfer von Browser Exploits wurde.

Viele Varianten des Bots verwenden Root-Kits um ihre Existenz zu verschleiern. Es gibt auch Varianten, die virtuelle Maschinen erkennen und dann entweder inaktiv bleiben oder diese zum Absturz bringen. Dies soll die Analyse des Bots erschweren.

Das Storm Botnetz kann sowohl zu Spam Versand als auch für DDoS Angriffe verwendet werden [7].

2) Architektur: Im folgenden wird die vom Storm Botnetz verwendete mehrstufige hybride Architektur beschrieben. Die Beschreibung basiert auf den Ergebnissen der Diplomarbeit von Frédéric Dahl [7].

An oberster Stelle steht Overnet, ein P2P Netz. Overnet wurde nicht nur von Storm sondern auch vom eDonkey2000 Client und seinen Nachfolgern verwendet. Overnet implementiert Kademila, eine verteilte Hashtabelle.

In Kademila ist jedem Knoten und jedem Datensatz ein zufällig generierter 160 Bit Bezeichner zugeordnet. Die Länge des gemeinsamen Präfix zweier Bezeichner gibt den Grad ihrer Ähnlichkeit an. Daten werden im Netz bei den Knoten gespeichert, denen sie am ähnlichsten sind.

Ein Knoten verwaltet für jede Ähnlichkeitsklasse eine Listen mit Knoten aus dieser Klasse. Wenn ein Knoten Kontakt zu anderen Knoten hat, wird dieser in die entsprechende Liste aufgenommen, sofern die Liste nicht voll ist. Jeder Knoten kennt aus jeder Ähnlichkeitsklasse gleichviele Knoten. Unter der Annahme, das die zufällig generierten Bezeichner sich gleichmäßig über dem Namensraum verteilen, kennt jeder Knoten viele der Konten, die ihm sehr ähnlich sind, und wenige derjenigen, die eine geringere Ähnlichkeit haben.

Bei der Suche nach einen Datensatz fragt der suchende Knoten zuerst die Knoten aus der Liste, die der Ähnlichkeitsklasse des suchenden Knoten mit den Bezeichner des gesuchten Datensatz entsprechen. Diese Knoten liefern eine Liste mit den ihnen bekannten Knoten, die dem gesuchten Bezeichner am ähnlichsten sind. Der suchende Knoten nimmt eine Teilmenge der ähnlichsten Knoten und fragt diese wiederum an. Das wird solange wiederholt, bis die Anfragen keine neuen ähnlicheren Knoten mehr liefern. Die ähnlichsten Knoten werden dann nach dem gesuchten Datensatz gefragt. Das Speichern von Daten läuft äquivalent.

Der Storm Bot berechnet jeden Tag aus dem aktuellen Datum 32 Bezeichner. Eine Suche in Netzwerk nach einem dieser Bezeichner bring eine Listen von Gateways. Gateways sind Knoten, die ohne Einschränkungen von anderen Knoten erreicht werden können und der mittleren Ebene des Storm Netzes angehören. Der Knoten baut eine Verbindung mit dem Gateways auf und fragt diesen nach Zielen von DDoS Attacke und Templates und Adressen für Spam-Mails.

Während des Verbindungsaufbaus mit dem Overnet erfährt ein Knoten, ob er ohne Einschränkungen erreichbar und damit ein Gateway ist. Ein Gateway veröffentlicht sich so in Overnet, dass er als ein solches erkennbar ist. Kurz danach wird er von der den Kontrollknoten aus der dritten Ebene des Storm Netzes kontaktiert.

Die Gateways bauen ein Fast-Flux Netzwerk (vergl. III-C) auf. Dieses wird für die Kommunikation der Gateways mit den Kontrollknoten, als auch um die in den Spam-Mails verlinkten Webseiten bereit zu stellen, verwendet.

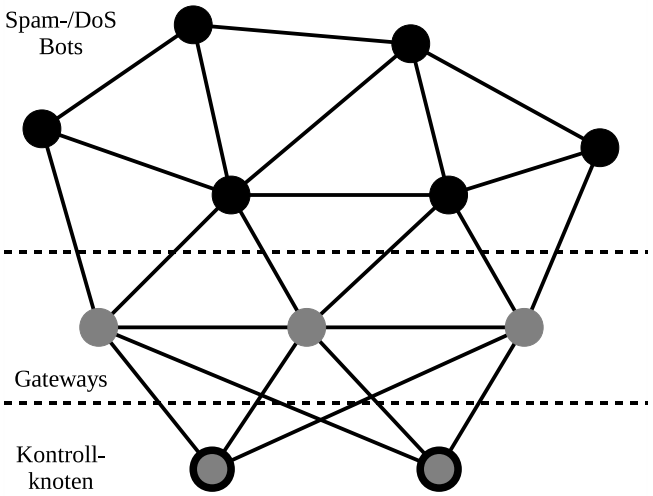

Abbildung 6. Struktur des Storm Botnetz

Seit Mitte Oktober 2007 verwendet das Botnetz eine einfache Verschlüsselung. Diese besteht aus einem 40 Bit Schlüssel, welcher per XOR mit dem Datenstrom verknüpft wird. Dadurch wurde das Storm Botnetz auch von dem restlichen, für Fileshareing benutzen, Overnet abgetrennt.

3) Sicherheitsanalyse: Das Storm Botnetz ist nicht besonders sicher aufgebaut. So wird nur eine sehr einfache Verschlüsselung mit eine kurzen sich wiederholenden Schlüssel eingesetzt. Weiterhin wird der Ursprung von Befehlen nicht überprüft.

Im Rahmen des 25C3 wurde demonstriert, wie man in das Storm Botnetz falsche C&C Server einschleust und Bots übernimmt oder deaktiviert [31], [32].

E. Conficker

1) Übersicht: Conficker, auch bekannt unter dem Namen Downadup verbreitet sich seit Mitte November 2008 über einen Windows RPC Exploit. Microsoft stellt zwar bereits seit dem 23. Oktober 2008 einen Patch für diesen zu Verfügung, dennoch konnte Conficker eine sehr große Verbreitung erlangen. Zusätzlich verbreitet er sich noch über lokale Dateifreigaben und entfernbare Medien. Der Ursprung von Conficker wird in der Ukraine vermutet [33].

Besonderes Medieninteresse erlangte Conficker am 13. Februar 2009, als bekannt wurde, dass er das Netzwerk der Deutschen Bundeswehr infiziert hatte. Zuvor waren bereits die

Netzwerke des britischen und französischen Militärs betroffen [34].

2) Architektur: Conficker berechnet täglich 250 Rendezvous-Domains und versucht von diesen ein Update herunterzuladen. Die Updates müssen mit einer Signatur versehen sein. Der öffentlichen Schlüssel für diese Signatur ist in den Code von Conficker eingebettet. Die Updates sind ausführbarer Code, welcher, sobald sein Ursprung verifiziert ist, gestartet wird.

Aktuell sind drei Varianten von Conficker in Verbreitung, welche sich aber teilweise ähneln. Alle verwenden ähnliche Algorithmen, um Domains zu generieren, aber teilweise mit verschiedene Initialisierungsparametern. Die neueren Versionen versuchen zusätzlich noch Sicherheitsmechanismen wie Virenscanner zu deaktivieren und können auch remote über eine Named Pipe der Windows IPC mit signierten Updates versorgt werden [33].

Conficker beinhaltet außer der Verbreitungsroutinen keinerlei Schadroutienen. Solche können allerdings jederzeit über ein Update nachgerüstet werden.

Aktuell ist nicht bekannt, dass der Update-Mechanismus von Conficker schon mal verwendet wurde. Außerdem verfügt er über keine C&C-Schnittstelle. Deshalb müsste man ihn zur Zeit eher als Wurm als als Bot bezeichnen. Die C&C-Schnittstelle könnte aber jederzeit über ein Update nachgerüstet werden [35].

V. Gegenmassnahmen

A. Systeme absichern

Die zuverlässigste Methode, dass Eindringen von Bots zu verhindern ist das System entsprechend abzusichern. Dazu gehören der Einsatz einer Firewall und eines Virenscanners. Insbesonders auf den Microsoft Betriebssystemen ist dieses notwendig. Durch ihre hohe Verbreitung, insbesonderes in Bereichen, in denen die Benutzer keine grundlegenden Kenntnissen über diese Problematik haben (z.B. der durchschnittliche Home-PC Anwender), bieten sie eine große Angriffsfläche für Bots.

Die Firewall verhindert, dass vom Internet aus auf das System zugegriffen werden kann. Auf den meisten Home- und Büro-Rechnern ist es nicht notwendig, dass der Rechner über das Internet erreicht werden kann. Dadurch können Lücken in den Systemdiensten nicht zum Einbruch genutzt werden. Bekannte Sicherheitslücken sollten möglichst schnell durch die Installation der entsprechenden Updates geschlossen werden.

Wichtig beim Einsatz eines Virenscanners ist, dass dieser regelmäßig aktualisiert wird. Ein Virenscanner arbeitet nur reaktiv. Er kann einen neuen Bot erst erkennen, wenn dieser dem Hersteller des Virenscanner aufgefallen ist, und das Signaturupdate dem Virenscanner verfügbar gemacht wurde. Agobot und Storm haben allerdings gezeigt, dass Virenscanner an ihre Grenzen kommen, wenn ein Schädling schnell in immer leicht verscheidenden Varianten auftaucht.

B. Abwehr von DDoS Angriffen

Präventiv kann man gegen DDoS Angriffe vor allem die eigene Infrastruktur auf des Eintreffen eines solchen Angriffs vorbereiten. Dazu gehören z.B.

- Verfügbar machen von TCP-SYN-Cookies [36]
- Einsatz von Paketfiltern
- Bereitstellen von Proxies und Reserve Servern
- Einsatz von Load Balancing

Während eines Angriffes können diese Maßnahmen dann gezielt aktiviert werden. Zusätzlich lohnt es sich, mit den Providern Kontakt aufzunehmen, damit die Pakete möglichst früh gefiltert werden [37].

C. Netz übernehmen und abschalten

Da viele Botnetze keine oder nur eine oberflächliche Authentifizierung verwenden, währe es theoretische möglich, die Bots mit einen gefälschten Befehl abzuschalten. Zusätzlich könnte z.B. auch noch Code nachgeladen werden, welcher die Bot-Software entfernt und die Sicherheitslöcher über die die Schädlinge eingedrungen sind schließt. Das konkrete Vorgehen ist von der Art des Botnetzes abhängig:

IRC: Solange keine asymmetrischen Verschlüsselungsverfahren zum Einsatz kommen enthält der Code des Bots alle Informationen um eine Verbindung zu dem IRC Kanal aufzubauen und Anweisungen an die Bots zu senden.

Domain gestützt: In diesem Fall muss die Domain übernommen werden. Dies kann z.B. durch einen Eingriff der Providers/Registras auf Anweisung einer höheren Instanz geschehen. Häufig haben die Bots auch einen Mechanismus, über welchen sie Domain-Namen berechnen und so versuchen Kontakt zu dem Netz herzustellen (vergl. Srizbi IV-B bzw. ConfickerIV-E). Da reicht es aus, einige solcher Domains zu reservieren und entsprechend auszustatten.

P2P: Hier muss mit einem manipulierten Client ein entsprechendes Kommando an das Netzwerk abgesetzt werden.

Konkret währen solche Eingriffe z.B. beim Storm [31] oder Srizbi [28], [29] möglich gewesen.

Ein solcher Eingriff ist allerdings rechtlich bedenklich: Das Entfernen des Bots könnte als eine unerlaubte Datenveränderung gemäß § 303a StGB [38] sein und währe dann unter Androhung einer Freiheitsstrafe von bis zu zwei Jahren verboten. Weiterhin könnte das Entfernen/Deaktivieren des Bots (aufgrund der hohen Verknüpfung mit den Betriebssystems) Schäden auf dem Rechner verursachen, wodurch Schadenersatzforderungen möglich wären.

VI. Zusammenfassung und Ausblick

Moderne Botnetze haben nicht mehr viel mit den alten IRC-basierten Netzen gemeinsam. Im allgemeinen zeichnet sich ein Trend zu Peer-to-Peer-basierten Botnetzen ab. Dies liegt darin begründet, dass diese Strukturen, sofern eine gute Implementierung verwendet wird, sehr robust sind. Weiterhin gibt es inzwischen auch einige freie P2P-Implementierungen, die für einen solchen Einsatz verwendet werden können.

Aber auch zentrale Instanzen kommen noch zum Einsatz. Wenn diese mit einen guten Fallback-Mechanismus ausgestattet sind, können sie sich sogar vom Abschalten ihres zentralen Servers erholen. Dies war im letzten Herbst besonders gut am Beispiel von Srizbi zu sehen.

Botnetze sind in Vergleich zu Viren eine neuere Erscheinung. Gegen sie können größtenteils ähnliche Mittel eingesetzt werden wie gegen Viren. Ein Problem ist allerdings, dass es oft viele Varianten des Bots gibt, wodurch die Bekämpfung erschwert wird. Gerade Bots mit öffentlichem Quelltext wie der Agobot haben dieses Problem verdeutlicht. Dadurch, dass Virenscanner nur reaktiv arbeiten, ist es ihnen nicht möglich, Bots mit absoluter Sicherheit aufzuspüren.

Viele der aktuellen Bots haben allerdings noch große Schwachstellen im Design. So zeigt das Beispiel Storm, dass ein Botnetz extrem angreifbar ist, wenn der Ursprung eines Kommandos nicht überprüft wird. Durch den Einsatz von verschlüsselter Kommunikation, wechselnden Ports und Anpassung des Kommunikationsverhaltens kann sich ein Botnetz noch besser tarnen. Es ist nur eine Frage der Zeit, bis auch solche Techniken kombiniert in einen Bot zum Einsatz kommen.

Eine effektive und langfristige Lösung des Problems ist aber nicht in Sicht. Jede Technik, die Botnetze verhindern soll, sorgt auf der Gegenseite für eine Anpassung, um diese zu umgehen.

LITERATUR

[1] E. Levy and I. Arce, "A Short Visit to the Bot Zoo," *IEEE Security & Privacy*, vol. ???, no. ???, pp. 76–79, 2005, abgerufen am 02.09.2008. [Online]. Available: ???

[2] I. Peacock, "Showing Robots the Door, What is Robots Exclusion Protocol?" *Ariadne*, vol. Issue 15, May 1998, abgerufen am 02.01.2009. [Online]. Available: http://www.ariadne.ac.uk/issue15/robots/

[3] Wikipedia, "Googlebot — Wikipedia, Die freie Enzyklopädie," 2008, stand 2. Januar 2009. [Online]. Available: http://de.wikipedia.org/w/index.php?title=Googlebot&oldid=51297108

[4] ——, "Computervirus — Wikipedia, Die freie Enzyklopädie," 2008, stand 2. Januar 2009. [Online]. Available: http://de.wikipedia.org/w/index.php?title=Computervirus&oldid=54744124

[5] Symantec, "SymbOS.Commwarrior.A," abgerufen am 02.01.2009. [Online]. Available: http://www.symantec.com/security_response/writeup.jsp?docid=2005-030721-2716-99

[6] K. D. Mitnick and W. Simon, *Die Kunst der Täuschung: Risikofaktor Mensch*. Mitp-Verlag, März 2006.

[7] F. Dahl, "Der Storm-Worm," Diplomarbeit, März 2008. [Online]. Available: http://pi1.informatik.uni-mannheim.de/filepool/theses/diplomarbeit-2008-dahl.pdf

[8] P. Brauch, "Geld oder Netz!" *c't*, no. 14/2004, p. 48, 2004. [Online]. Available: http://www.heise.de/ct/04/14/048/

[9] K. Hayashi, "Spam from the Kernel: Full-Kernel Malware Installed by MPack," Juli 2007, abgerufen am 11.01.2009. [Online]. Available: https://forums.symantec.com/t5/Malicious-Code/Spam-from-the-Kernel-Full-Kernel-Malware-Installed-by-MPack/ba-p/305311#A139

[10] D. Bachfeld, "Rootkit verschiebt Windows in virtuelle Maschine," *heise security*, Oktober 2006, abgerufen am 26.03.2009. [Online]. Available: http://www.heise.de/security/news/meldung/print/79676

[11] J. Oikarinen and D. Reed, "Internet Relay Chat Protocol," RFC 1459 (Experimental), May 1993, updated by RFCs 2810, 2811, 2812, 2813. [Online]. Available: http://www.ietf.org/rfc/rfc1459.txt

[12] V. Kamluk, "Botnetze - Geschäfte mit Zombies," *Kaspersky Lab*, Mai 2008, abgerufen am 04.01.2009. [Online]. Available: http://www.kaspersky.com/de/downloads/pdf/vkamluk_botnetsbusiness_0508_de_pdf.pdf

[13] M. Romano, S. Rosignoli, and E. Giannini, "Robot Wars - How Botnets Work," *WindowSecurity.com*, Okt./Nov. 2005, abgerufen am 06.01.2009. [Online]. Available: http://www.windowsecurity.com/articles/Robot-Wars-How-Botnets-Work.html?printversion

[14] P. Mockapetris, "Domain names - concepts and facilities," RFC 1034 (Standard), Nov. 1987, updated by RFCs 1101, 1183, 1348, 1876, 1982, 2065, 2181, 2308, 2535, 4033, 4034, 4035, 4343, 4035, 4592. [Online]. Available: http://www.ietf.org/rfc/rfc1034.txt

[15] ——, "Domain names - implementation and specification," RFC 1035 (Standard), Nov. 1987, updated by RFCs 1101, 1183, 1348, 1876, 1982, 1995, 1996, 2065, 2136, 2181, 2137, 2308, 2535, 2845, 3425, 3658, 4033, 4034, 4035, 4343, 2137, 2845, 3425, 3658, 4035, 4033. [Online]. Available: http://www.ietf.org/rfc/rfc1035.txt

[16] P. Bächer, T. Holz, M. Kötter, and G. Wicherski, "The Honeynet Project - Know Your Enemy: Fast-Flux Service Networks," Juli 2007, abgerufen am 02.09.2008. [Online]. Available: http://www.honeynet.org/

[17] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext Transfer Protocol – HTTP/1.1," RFC 2616 (Draft Standard), Jun. 1999, updated by RFC 2817. [Online]. Available: http://www.ietf.org/rfc/rfc2616.txt

[18] B. Ungerer, "US-Provider ziehen Spam-Schleuder den Stecker," *iX News*, November 2008, abgerufen am 06.01.2009. [Online]. Available: http://www.heise.de/ix/news/meldung/print/118804

[19] J. Schmidt, "Hydra der Moderne," *c't*, no. 18/2007, p. 76, 2007, abgerufen am 02.09.2008. [Online]. Available: http://www.heise.de/security/artikel/print/94211

[20] T. Fuhrmann, "Internet Protokolle II," SS2008.

[21] I. Arce and E. Levy, "An Analysis of the Slapper Worm," *IEEE Security & Privacy*, vol. January/February, pp. 82–87, 2003.

[22] J. Stewart, "Phatbot Trojan Analysis," *SecureWorks*, März 2004, abgerufen am 21.03.2009. [Online]. Available: http://www.secureworks.com/research/threats/phatbot/

[23] C. Bryan-Low, "How Legal Codes Can Hinder Hacker Cases," *The Wall Street Journal Online*, Januar 2007, abgerufen am 04.01.2009. [Online]. Available: http://online.wsj.com/public/article_print/SB116900488955878543-yrMHYlacFyxijV14BxFZfXeU1_8_20070216.html

[24] P. Brauch, "Superwurm mit öffentlichem Quelltext," *heise online*, April 2004, abgerufen am 04.01.2009. [Online]. Available: http://www.heise.de/newsticker/meldung/print/46634

[25] ca, "Virus Detail - Win32/Agobot Family," abgerufen am 05.01.2009. [Online]. Available: http://www.ca.com/us/securityadvisor/virusinfo/virus.aspx?id=37776

[26] K. J. Higgins, "Srizbi Botnet Sending Over 60 Billion Spams a Day," *DarkReading*, Mai 2008, abgerufen am 11.01.2009. [Online]. Available: http://www.darkreading.com/shared/printableArticle.jhtml?articleID=211201479

[27] B. Krebs, "Host of Internet Spam Groups Is Cut Off," *washingtonpost.com*, November 2008, abgerufen am 11.01.2009. [Online]. Available: http://www.washingtonpost.com/wp-dyn/content/article/2008/11/12/AR2008111200658_pf.html

[28] D. Bachfeld, "Botnetz wiederauferstanden," *heise online*, Dezember 2008, abgerufen am 06.01.2009. [Online]. Available: http://www.heise.de/newsticker/meldung/print/119703

[29] B. Krebs, "Srizbi Botnet Re-Emerges Despite Security Firm's Efforts," *washingtonpost.com*, November 2008, abgerufen am 11.01.2009. [Online]. Available: http://voices.washingtonpost.com/securityfix/2008/11/srizbi_botnet_re-emerges_despi.html

[30] Symantec Corporation, "Linux.Slapper.Worm," abgerufen am 30.04.2009. [Online]. Available: http://www.symantec.com/security_response/print_writeup.jsp?docid=2002-091311-5851-99

[31] J. Schmidt, "Sturmwurm-Botnetz sperrangelweit offen," *heise online*, Januar 2009, abgerufen am 21.03.2009. [Online]. Available: http://www.heise.de/newsticker/meldung/print/121310

[32] C. Klaß, "25C3: Storm-Botnet gekapert," *golem.de*, Dezember 2008, abgerufen am 27.03.2009. [Online]. Available: http://www.golem.de/print.php?a=64333

[33] P. Porras, H. Saidi, and V. Yegnewaran, "An analysis of conficker's logic and rendezvous points," SRI International, Tech. Rep., Februar 2009, abgerufen am 09.03.2009. [Online]. Available: http://mtc.sri.com/Conficker/

[34] S. Klettke, "Bundeswehr kämpft gegen Viren-Befall," *Spiegel Online*, Februar 2009, abgerufen am 21.03.2009. [Online]. Available: http://www.spiegel.de/netzwelt/web/0,1518,607567,00.html

[35] K. J. Higgins, “Widespread Conficker/Downadup Worm Hard To Kill,” *DarkReading*, Januar 2009, abgerufen am 22.03.2009. [Online]. Available: http://www.darkreading.com:80/shared/printableArticle.jhtml?articleID=212901489

[36] J. Schmidt, “Dämme gegen die SYN-Flut,” *heise security*, Dezember 2003, abgerufen am xx.03.2009. [Online]. Available: http://www.heise.de/security/artikel/print/43066

[37] B. für Sicherheit in der Informationstechnik, “Empfehlungen zum Schutz vor verteilten Denial of Service-Angriffen im Internet,” Juni 2000, version 1.1a vom 20.06.2000; Abgerufen am 24.03.2009. [Online]. Available: http://www.bsi.de/fachthem/sinet/gefahr/ddos.htm

[38] “Strafgesetzbuch – §303a Datenveränderung.” [Online]. Available: http://bundesrecht.juris.de/stgb/__303a.html