HOCHSCHULE DER MEDIEN
Studiengang Medieninformatik

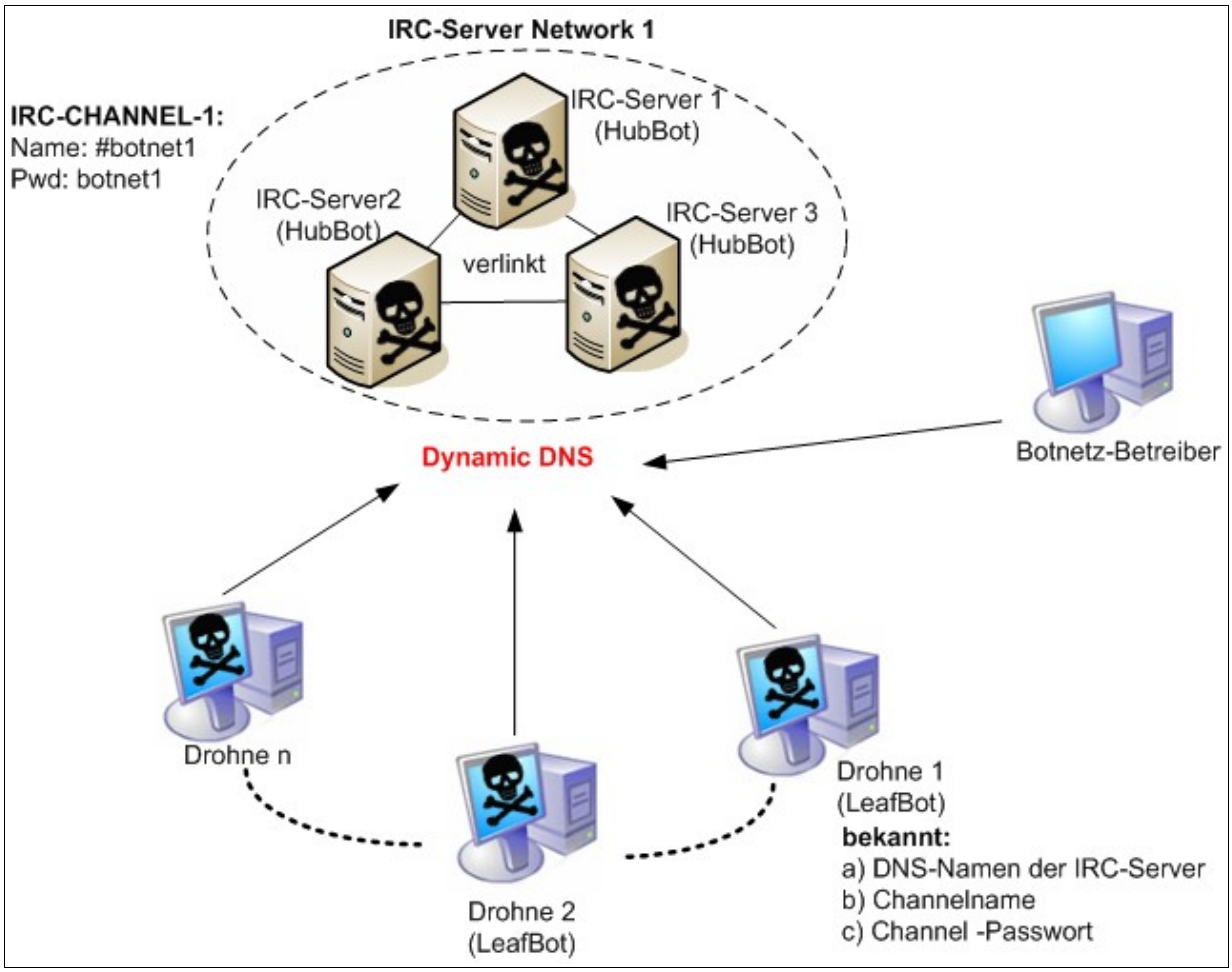
Datum	17.02.2006
Teilnehmer	Jürgen Butz (12594) Andreas Springer (13887) Sebastian Roth (13883)
Betreuer	Prof. Roland Kiefer Christoph Alscher

Inhaltsverzeichnis

1 Problematik und Motivation

1.1 Die Gefahr der Botnets

Das Internet ist ein großer Spielplatz der nicht beaufsichtigt wird und damit Raum für jeglichen Spaß läßt, den man sich nur ausdenken kann. So entstehen in Zusammenarbeit über das weltweite Netz großartige Projekte. In der Natur des Menschen liegt es aber auch diese Freiheit auszunutzen. Seit über zwei Jahrzehnten machen sich Viren auf den Rechnern der Benutzern zu schaffen. Die Zahl aktueller Erscheinungen von Schädlingen, welche sich über das Internet verbreiten ist sehr groß. Darunter finden sich nicht nur die zuvor erwähnten Viren, sondern auch Trojanische Pferde, Würmer, Spyware machen die Runde. Um einen infizierten Rechner auch nutzen zu können, öffnen einige der Schädlinge Hintertüren im System. Oft horcht dieser auf eine Verbindung seines Meisters aus dem Internet um dann Befehle entgegenzunehmen. Robots – weiterhin nur noch als Bots bezeichnet - drehen den Spieß um. Bots öffnen aktiv eine Verbindung zu einem gemeinsamen Kommunikationspunkt. In den meisten Fällen handelt es sich dabei um einen Internet Relay Chat Server. Diese Kommunikationsplattform bietet die Möglichkeit sich zu mehreren in "Kanälen" (Channels) oder individuell mit Personen zu unterhalten.



Ein solcher Verbund aus mehreren Bots im gleichen Channel nennt sich Botnet. Der Eigentümer eines Botnets ist nun in der Lage sich in den Channel einzuloggen und mit allen Teilnehmern zu reden. Dies wird allerdings nicht dazu genutzt, sich mit den Bots zu unterhalten, sondern diese ausschließlich in der Befehlsform anzureden.
Mit einer Heerschar eingenommener Rechner (sogenannte Zombies) läßt sich eine Vielzahl auf Aufgaben erledigen - deren Sinn oder Unsinn hier nicht diskutiert werden soll. So kann der Betreiber beispielweise einen verteilten Angriff auf einen wichtigen Server im Internet einleiten (DDoS).
Das ist in etwa wie eine Pizzalieferattacke. Ein Angreifer bestellt bei hunderten Lieferanten jeweils für die gleiche Zeit eine Pizza an die Adresse des Opfers. Mehrere hundert Pizzalieferanten stehen dann vor dessen Tür und wollen ihr Geld haben. Auch die Lieferenten sind Opfer, die nur benutzt wurden. Der wirkliche Angreifer kann nur sehr schwer ermittelt werden. Der Betreiber kann die vielen Maschinen auch als Weiterleiter für Spam E-Mails nutzen oder sensitive Daten der Rechner aufzeichnen.
Diese Eigenschaften lassen sich auch als Dienste an Dritte verkaufen, womit sich schnell Geld machen läßt. Skrupellose Zeitgenossen schaffen sich so einen netten Nebenverdienst. Auch Skriptkiddies finden immer mehr Interesse an Botnets.

1.2 Aktuelle Ereignisse

Um eine Vorstellung zu bekommen, welche Auswirkungen solche Netze haben können bzw. welche Ausmaße erreicht werden, haben wir einige aktuelle Ereignisse von renomierten Webseiten zusammengestellt, die uns während unseres Projektes erreicht haben.

- 28. Oktober 2005, Albert's Infoportal [1]
  Holländisches Botnet umfasst 1,5 Millionen PCs weltweit
  Das vor zwei Wochen entdeckte Botnet, das in den Niederlanden zu der Verhaftung von drei verdächtigen Männern geführt hat und nach dem bisherigen Stand rund 100.000 Rechner umfasste, ist nach neuesten Erkenntnissen offenbar doch viel größer. Jetzt ist die Rede von 1,5 Millionen PCs weltweit, die in dem Zombie-Netz zusammengeschlossen sein sollen.
- 12. Oktober 2005, ZDNet News [2]
  Botnet-Angriffe lassen Provider zittern
  Netzbetreiber und Service Provider werden immer häufiger Angriffsziel von sog. Botnets, die über koordinierte Angriffe in der Lage sind ganze Server lahmzulegen.

Diese Distributed-Denial-of-Service-Attacken (DDoS) werden zumeist von einem Netzwerk von Computern ausgeführt, die durch ein über Viren oder Trojaner eingeschleustes Programm (Bot) meist ohne Wissen der Benutzer kontrolliert werden. Angesichts des enormen Bedrohungspotenzials dieser Netzwerke, die aus über 100.000 Computer bestehen können, nehmen Experten angesichts sich häufender Erpressungsversuche immer häufiger den Begriff Cyberterrorismus in den Mund.

- 15. November 2005, PC Welt News [3]

  Botnet-Sprengung hat Spam reduziert

  Anfang November wurde in Kalifornien ein Mann verhaftet, der ein großes Botnet kontrolliert haben soll. Nach Angaben eines Mitarbeiters von Message Labs ist nach dieser Festnahme eine ganze Kategorie von Spam-Mails komplett vom Radar verschwunden...

  Der Anfang November vom FBI verhaftete Jeanson James Ancheta, 20, soll ein Botnet betrieben haben, das nach bisherigen Annahmen lediglich zur massiven Verbreitung von Adware genutzt wurde. Dabei sollen diverse Adware-Programme auf mehr als 400.000 PCs installiert worden sein.

- 18. Januar 2006, ZDNet News [4]

  Gründer der "Million Dollar Homepage" wird erpresst

  Der britische Student Alex Tew, der mit dem Verkauf von winzig kleinen Werbeflächen auf seiner «Million Dollar Homepage» zum Millionär wurde, wird erpresst. Mehrere Web-Piraten forderten nach einem Bericht des britischen Rundfunksenders BBC vom Mittwoch insgesamt mehr als 40.000 Euro. Andernfalls drohten sie damit, die Internetseite mit einer Million verkauften Pixeln durch einen Hacker-Angriff lahm zu legen. Als Tew nicht zahlte, ließen sie der Drohung Taten folgen. Die Seite war fast eine Woche lang nur noch zeitweise aufzurufen. Inzwischen funktioniert sie wieder.

- 27. Januar 2006, SC Magazine [5]

  Fastest growing malware threat: bots

  Bots are the fastest growing malware threat, with more than 10,000 new variants detected last year, security firm Panda Labs said Thursday...

  "The new focus of malware is leading to the professionalization of both the creation of malware and the search for financial returns," the company said. "For this reason, the number of variants developed in a family (of bots) could stretch into the thousands, a figure far too high for signature-based protection to cope with."

1.3 Gegenmaßnahmen

1.3.1 Unterbindung der Verbreitung

In diesem Abschnitt sollen kurz verschiedene Möglichkeiten vorgestellt und beleuchtet werden, mit deren Hilfe die Ausbreitung von Bots und Entstehung von Botnets verhindert oder zumindest eingeschränkt werden kann. Ein erster Ansatz um dieses Ziel zu erreichen ist sicherlich das Problem direkt an der Wurzel zu packen. In diesem Fall besteht die bildlich gesprochene Wurzel aus den einzelnen Bots, mit denen sich ein Botnet im Internet verankert. Die meisten annektierten Rechner innerhalb eines solchen Netzes sind Computer von Homeusern, die ihren Rechner nur notdürftig bis garnicht auf sicherheitsrelevante Aspekte überprüfen und überhaupt kein Gespür oder Feinsinn dafür entwickeln. Spricht man einen Endanwender darauf an, dass er sein System besser sichern soll, kommt meistens die Antwort: "Ich hab eh nichts wichtiges drauf was jemanden interessieren könnte!". Aber genau da liegt der Hund begraben. Ihnen ist gar nicht bewusst, dass mit Hilfe ihres Rechners im Netz Verbrechen begangen werden und sie sich somit unbewusst als Mittäter schuldig machen - tja aber momentan gilt immer noch die Devise "Wo kein Kläger, da kein Richter!". Hier könnte man wieder eine Diskussion über die Einführung eines Computerführerscheines führen. Andererseits muss man den schwarzen Peter wiederum an die Firma Microsoft weiterreichen, die ja im Grunde genommen für die Schwachstellen im System veranwortlich ist und die Benutzer nicht hinreichend über die Gefahren informiert. Diese Diskussion soll hier aber nicht weiter verfolgt werden.

Es ist aber offensichtlich, dass die Endanwender auch ihren Teil dazu beitragen müssen. Sie sollten auf alle Fälle auf das neue Problem aufmerksam gemacht und dafür sensibilisiert werden.

Eine weitere präventive Maßnahme ist das Aufspielen von Softwareupdates. Die meisten von uns getesteten Bots verwenden zur automatischen Verbreitung längst bekannte Sicherheitslücken, für die es von den Herstellern schon lange Sicherheitpatches gibt. Aus unserer Sicht könnten hiermit schon ein Großteil der Bots abgewehrt werden. Zur zusätzlichen Sicherheit können Antivirentools oder Personal Firewalls eingesetzt werden. Bei falscher Konfiguration und unregelmäßiger Pflege bieten auch diese nur eine Scheinsicherheit. Der Anwender ist zu Unrecht beruhigt. Dieses Problem liegt aber in der Usability von Securitymechanismen begründet, die bis zum heutigen Zeitpunkt leider immer noch orthogonal zu einander stehen.

Gehen wir aber von dem heutigen Standpunkt aus, dass viele User gar nicht wissen, wie sie ihren Rechner sichern können, so sollten die einzelnen ISPs mehr Verantwortung übernehmen. Denn viele ISPs, besonders call-by-call Anbieter, kümmern sich überhaupt nicht um die Rechner, die sich in ihr Netz einwählen. So ist es nicht verwunderlich, dass sehr viel Bot-Traffic über die Router läuft (automatische Attacken über verschiedene Vulnerabilities, welche für das Spreading verwendet werden). Ein solcher ISP könnte z.B. solchen Traffic unterbinden, indem die Ports gesperrt werden. Natürlich ist das nicht die beste Lösung des Problems aber es verhindert, dass der Großteil der Bots sich weiterverbreiten kann (Diskussionsstoff für Zensur).
Unter anderem ist diese Lösung aus Gründen der Freiheit, die ein User im Internet haben soll nicht gerade die ideale, denn somit ist dem User z.B. untersagt Windows Filesharing zu betreiben. Eine weitaus bessere Lösung, aber dafür auch Aufwendigere ist, dass der ISP über Intrusion Detection Systeme schädliches Verhalten von Rechnern in seinem Netz aufspürt und diese dann einfach komplett sperrt. Darüber hinaus muss der Betroffene in Kenntnis gesetzt werden, dass er Verteiler von Malware ist und erst wieder freigeschalten werden, wenn er sein System bereinigt hat.
Natürlich ist das nicht ganz so einfach wie es hier so salopp gesagt wird und wie bereits erwähnt ein sehr großer Aufwand für die Netzbetreiber. Wahrscheinlich würde es sich sogar umsatzhemmend auswirken, da dem User immer noch die Möglichkeit bleibt sich bei einem anderen Provider einzuwählen, der keine solch strikten Bestimmungen hat.
Aber man muss erwähnen, dass das BelWue hier beide dargestellten Varianten umgesetzt hat. Aus diesem Grund, hatten wir auch zuerst Probleme Malware einzufangen und unser Netz angreifbar zu machen. Zwecks der Fairness den anderen Providern gegenüber muss man hier aber auch sagen, das BelWue arbeitet in einem ganz anderen Umfeld als ein großer ISP (Benachrichtigungen des IDS werden an das Rechenzentrum der jeweiligen Forschungseinrichtung weitergeleitet).
Aber selbst wenn alle oben genannten Abwehrmechanismen verwendet werden, kann nicht davon ausgegangen werden, dass keine Botnets mehr auftreten. Ein weiterer Schutz vor Botnets besteht darin, Botnetbetreiber strafrechtlich zu verfolgen und somit Neulinge auf diesem Gebiet abzuschrecken.

1.3.2 Bekämpfung aktiver Botnets

Um diese Art der Bekämpfung ging es bei unserem Projekt. Wie bereits in der Einleitung beschrieben (Pizza-Angriff) bietet ein DDoS Angriff keine Informationen, die zu dem Ursprung bzw. dem Verantwortlichen führen könnten. Deshalb ist die einzige Möglichkeit die bleibt, die zentrale Stelle, über die Kommandos abgesetzt werden außer Gefecht zu setzen.

2 Ziele

Wir sehen die Ausbreitung von Botnets im Internet als eine sehr große Gefahr. Deshalb wollen wir mit diesem Projekt zum einen Netze und deren Betreiber ausfindig machen um die Aktivitäten dieser zu tracken und zu analysieren. Zum anderen wollten wir mit der Präsentation und der Vorstellung an der Media Night die Anwesenden auf das Problem aufmerksam machen.

Um die Ziele erreichen zu können, mussten wir ein Honeynet aufsetzen, welches den Zweck hatte Malware einzufangen. Die Bots repräsentieren den Schlüssel zu einem Botnet, da diese die Zugangsdaten zum IRC-Server gespeichert haben müssen. Durch Analyse dieser sollte es möglich sein den zentralen Punkt des jeweiligen Botnets ausfindig zu machen um somit an die im Ziel definierten Daten und Informationen zu gelangen.

3 Recherche

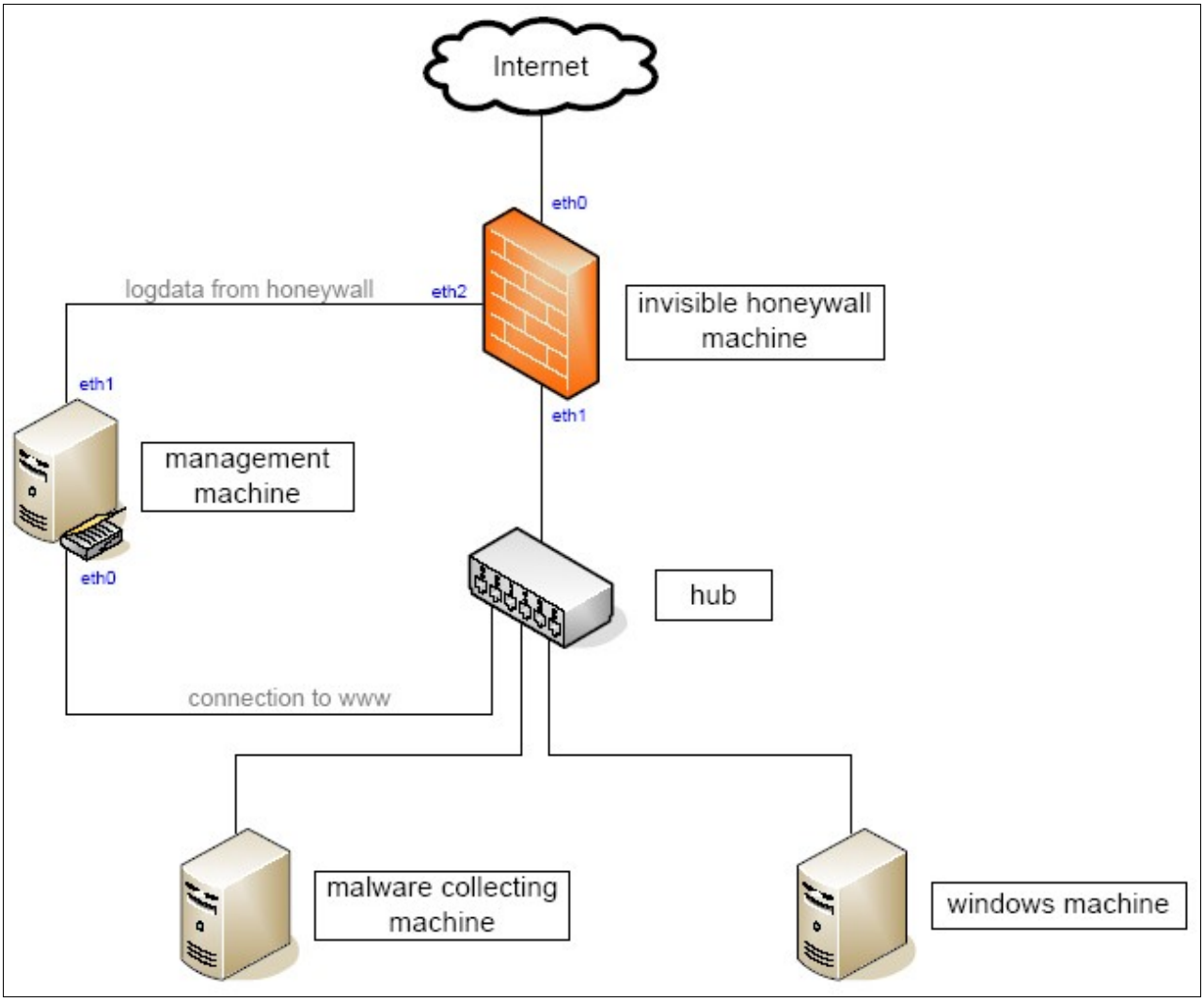
Zu Anfang unseres Projekts haben wir uns erst einmal über das Thema informiert und sind natürlich auf viele Dinge bzw. Probleme gestoßen, die uns erwarten könnten. In diesem Kapitel stellen wir nun einige Probleme vor, mit denen wir während des Projektes gerechnet hatten.

- *Bots kommunizieren verschlüsselt mit dem IRC-Server*
  Die Verschlüsselung verhindert in diesem Fall, dass man die Informationen zum Server (Channel, Benutzername, Passwort, etc.) und auch die ausgetauschten Kommandos nicht mithören kann. Des Weiteren fällt es schwer, ebenfalls eine solche Verbindung aufzubauen, um den IRC-Server belauschen zu können. Hierbei gäbe es nur die Möglichkeit, ein Rootkit einzusetzen, welches die Daten mitlesen kann bevor sie verschlüsselt werden.

- *Bots installieren eigenes Rootkit*
  Es könnte beispielsweise verwendet werden, um die Existenz von Prozessen und Dateien zu verbergen, sichere Daten mit zu sniffen oder andere Programme an ihrer Funktion zu hindern. Es gibt jedoch auch Tools, welche Rootkits erkennen, womit man dem Problem entgegnen kann.
- *Analysetools werden erkannt und geblockt*
  In vielen Berichten haben wir nachgelesen, dass einige Bots bekannte Analyzing-Tools wie z.B. Ethereal lahmlegen, um die Kommunikation vor den Augen des Anwenders verbergen zu können.
- *Einsatz der Honeywall-CD Roo*
  Die Honeywall-CD Roo des amerikanischen Honeynet Projekts haben eine CD mit Tools zusammengestellt, die das Fangen von Bots erleichtern soll. Dazu gehören auch diverse Traffic-Analyzing Tools. Zum Steuern gibt es ein Webinterface, über das alles gesteuert werden kann. Jedoch haben die Erfinder selbst angemerkt, dass man das System nicht unverändert einsetzen soll, da auch Botnetbetreiber die Chance haben, sich diese CD herunterzuladen. Dadurch könnte eine solche Honeywall erkannt und eventuell sogar kompromittiert werden.
- *Änderung des Channel Passworts*
  Es ist möglich, dass sich das Passwort im Sekundentakt durch einen Algorithmus ändert. Dadurch wird es sehr schwer, sich manuell in einen Channel einzuloggen, wenn man nicht im Besitz des Algorithmus ist.
- C*hannelhopping*
  Auch möglich ist ein ständiges Wechseln des Channelnamens, indem die Kommandos abgesetzt werden. Dies würde auch wiederum einen Algorithmus erfordern, welcher in den Bots integriert sein muss.
- *Erkennung unauthorisierter User auf dem IRC-Server*
  Wenn man sich mit einem eigenen IRC-Client in einen Server einloggt, könnte es passieren, dass man von dem Betreiber entdeckt wird. Dies kann unter Umständen böse Folgen nach sich ziehen. Man könnte schnell selber Ziel eines Angriffs werden.

4 Unser Honeynet

4.1 Aufbau



4.2 Verbindung zum Internet

Von Anfang an war klar, dass unser Netz unabhängig vom HdM-Netz bestehen sollte, um die dort bestehende Infrastruktur nicht zu gefährden oder mit zuviel Traffic das Netz zu belasten. So organisierte Herr Alscher für uns einen Zugang, der direkt zum BelWü geht und von dort ins Internet geroutet wird. Daraufhin waren wir der Meinung, wir wären nun für alle Malware dieser Welt erreichbar, um mit dem Einfangen von Bots beginnen zu können. Nach einer Woche Laufzeit des Tools Mwcollect [6], welches von mehreren Projekten auf der ganzen Welt erfolgreich eingesetzt worden ist, waren nur sehr wenige Angriffe zu verzeichnen. Des Weiteren wurde an den berüchtigten Port 445, auf den bekannterweise ein großer Teil der Angriffe abzielen, kein einziges Paket adressiert.

Dann begannen unsere Überlegungen, wo der Traffic geblockt werden könnte. Nach Überprüfung unserer Firewallregeln und Nachfragen am Rechenzentrum der HdM waren wir nicht viel schlauer. Daraufhin fanden wir auf der Homepage des BelWue heraus, dass unter anderem der Port 445 auf den Routern im Netz abgeblockt wird. Herr Alscher hat uns ermutigt mit dem Rechenzentrum Kontakt aufzunehmen.
Nach nur einer E-Mail wurden die Ports für unser Subnetz geöffnet, doch der große Ansturm blieb leider immer noch aus. Bei weiteren Nachfragen stellte sich dann heraus, dass das Problem an den Peering Points liegt. Glücklicherweise standen zu der Zeit Änderungen an diesen Maschinen an. Deshalb wurde auch diese Portöffnung sehr schnell von den Mitarbeitern umgesetzt.

4.2.1 Honeywall

Betriebssystem: Debian Linux
Die Honeywall ist das wichtigste Element in unserem Honeynet. Dort wird zum einen der komplette Traffic untersucht und limitiert und zum anderen werden Warnungen bei bekannten Angriffssignaturen mitgeloggt (Stateful Paketfilter Firewall und IDS).

4.2.2 Management PC

Betriebssystem: Debian Linux
Die Management Maschine dient zum Sammeln bzw. Auswerten verschiedener Logs und zur Kommunikation mit der Firewall von Außen (Remoteadministration über SSH).

4.2.3 Malware collecting machine

Betriebssystem: Debian Linux
Auf diesen PC sollen alle Angriffe gelenkt werden um die Bots, die normalerweise installiert werden zur Analyse herunter zu laden und zu isolieren.

4.2.4 windows machine

Betriebssystem: Windows XP, SP2, Hotfixes installiert
Um diesen PC gab es einige Diskussionen in unserem Team und auch mit Herrn Alscher, da eine Windowsmaschine immer eine größere Gefahr darstellt, auch voll gepatcht. Zuerst wollten wir die gesammelte Malware auf diesem System laufen lassen, um sehen zu können, was die Malware lokal anstellt und vor allem zuerst herausfinden zu können, ob es sich dabei um einen IRC-Bot handelt. Da jedoch auch die Honeywall nicht ganz verhindern kann, dass unsere Rechner weitere PCs im Internet angreifen, haben wir uns entschieden, die erste Analyse auf eigenen Maschinen von daheim durchzuführen.

Eine der ersten Binaries, die Jürgen zuhause auf einem Windows Rechner gefangen hatte, haben wir im Labor auf dieser Maschine analysiert, jedoch ohne Internetverbindung. In diesem Fall haben wir ein Linuxnotebook genutzt, um der Malware eine WAN-Verbindung vorzugaukeln (z.B. DNS-Server, Gateway, IRC-Server). Das Ergebnis waren die Daten zu unserem ersten gefundenen Botnet. Das Verfahren war jedoch viel zu aufwendig, um es für alle Binaries einsetzen zu können. Letztendlich haben wir diese Maschine noch genutzt, um unsere JAVA-Drohne JASirc laufen zu lassen um die entdeckten IRC-Server zu überwachen.

4.3 Traffic kontrollieren

Um die Honeywall unangreifbar zu machen, haben wir die Eigenschaft der Bridging-Funktion von Linux genutzt, wodurch die Maschine unsichtbar wird. Es operiert in diesem Fall nicht auf der IP-Ebene, sondern auf Layer 2 (Switch mit zwei Ports). Die Besonderheit ist, dass Linux trotzdem mit dem Netfilter-Framework [7] die Pakete auf höheren Schichten untersuchen und verändern/verwerfen kann!

Wie man im Aufbauschema erkennen kann, ist die Honeywall mit 3 NICs ausgestattet. Sie braucht zwei, um den ein- und ausgehenden Verkehr vom Internet mitzusniffen und gegebenenfalls weiterzuleiten und eine weitere, über die man die Honeywall gesichert erreichen kann. Könnte man die Honeywall über eine der beiden ersten NICs ansprechen, wäre die Unsichtbarkeit der Honeywall nicht mehr gewährleistet.

Da auch Herr Alscher viel Wert darauf legte, dass auf keinen Fall eine möglich kompromittierte Maschine in unserem Netz ins Internet spamt und damit großen Traffic erzeugt, haben wir viel Zeit darin investiert, die Honeywall und die Kontrollmechanismen möglichst sicher zu machen. Beispielsweise wird die Anzahl der TCP/UDP Verbindungen, die ins Internet aufgebaut werden sehr begrenzt. Damit wird verhindert, dass weitere Rechner im Internet gefährdet werden, falls es doch zur Kompromittierung eines Rechners kommen sollte. Dieser Fall trat aber im Laufe des Projekts zum Glück erst gar nicht auf. Die Regeln für das Netfilter-Framework haben wir mit dem Utility iptables gesetzt, welche in 4.3.1 genauer beschrieben werden.

Der Verkehr, der über die Honeywall läuft wird ebenfalls mitgeloggt, um einerseits Statistik betreiben zu können und um Angriffe auf unsere Rechner erkennen zu können. Die Logs des Netfilter-Frameworks werden mit dem Tool Specter [8] in die MySQL-Datenbank auf dem Management-Server geschrieben, um sie dort besser analysieren zu können.

Des Weiteren läuft auf der Maschine die IDS-Software Snort [9], welche die ankommenden Pakete mit vorhandenen Regeln vergleicht. Stimmt ein Paket mit einer dieser Regeln überein, wird dies als Angriff gewertet. Die Warnungen werden ebenfalls in die Datenbank auf dem Management-Server geschrieben.

4.3.1 netfilter-Regeln

Für die Filterregeln einer unsichtbaren Honigwand muss eine Spezialität beachtet werden. Die Optionen für eingehende und ausgehende Netzwerkkarten (-i, -o) sind durch den Match 'physdev' zu ersetzen. Unser Regelfile sieht zur Erleichterung für beide Verbindungsrichtungen jeweils eine Variable vor:

```
OUTBOUND="-m physdev --physdev-in eth1 --physdev-out eth0"
INBOUND="-m physdev --physdev-in eth0 --physdev-out eth1"
```

Das Regelwerk ist nach dem Prinzip aufgebaut, alles ist verboten, was nicht ausdrücklich erlaubt wurde (Defaultpolicy DROP). Um den Managementserver vor Zugriffen aus dem Internet über das Logdateninterface zu schützen, sind alle weitergeleiteten Pakete von und zur Schnittstelle eth2 komplett gesperrt. Ausschließlich Administrations- und Logdaten sind hier erlaubt.
Um möglichst viel Verkehr vom Internet mitzubekommen und vorallem Malware einzufangen sind alle eingehenden Pakete erlaubt. Das Aufzeichnen erlaubt später eine statistische Analyse.

```
iptables -A FORWARD ${INBOUND} -m state --state NEW -j ULOG \
   --ulog-prefix "INBOUND: " --ulog-nlgroup 1
iptables -A FORWARD ${INBOUND} -j ACCEPT
```

Für neue Verbindungen ins Internet gibt es nur eine Bedingung. Nicht zuviele in kurzer Zeit. Es ist nicht ausgeschlossen bei solch einem Szenario ungewollt Opfer eines handfesten Angriffs zu werden. Ein Traffic-Limiting verhindert dann, dass ein Rechner aus unserem Netz als Spam-Relay oder für andere groß angelegte Angriffe auf weitere Maschinen im Internet benutzt werden kann.

```
iptables -A FORWARD ${OUTBOUND} -m state --state NEW -m limit \
   --limit ${RATE}/${SCALE} --limit-burst ${RATE} -j ACCEPT
iptables -A FORWARD ${OUTBOUND} -m state --state NEW -j ULOG \
   --ulog-prefix "Drop after ${RATE} conn." --ulog-nlgroup 1
iptables -A FORWARD ${OUTBOUND} -m state --state NEW -j DROP

# allow all packets belonging to a connection
iptables -A FORWARD ${OUTBOUND} -j ACCEPT
```

Einzelne Funktionsblöcke in unserem Shell-Script ermöglichen die einfache Deaktvierung und Aktivierung der Regeln. Auch ein komplettes Neuladen ist möglich. Zusätzlich ist im iptables-Script eine Funktion für updatefreundliche Regeln. Hierbei sind alle neuen Verbindungen von außen gesperrt. Die Gegenrichtung ist komplett offen, sodass für die kurze Zeit von Softwareupdates keine Verbindungsprobleme auftreten.

4.4 Übersicht wahren

Um die Firewall im Notfall von zuhause aus abschalten zu können, haben wir die Managementmaschine aufgesetzt. Auf dieser läuft zudem ein LAMP-System (Linux, Apache, MySQL, PHP), um die Speicherung und Ansicht bzw. Analyse der gesammelten Logs zu vereinfachen.

4.4.1 Sammeln der Logdaten

Über eine der NICs (eth0) sollten Logs von den beiden anderen Maschinen (Malware collecting machine und windows maschine) gesammelt werden. Dies haben wir jedoch verworfen, da die Maschine schon zu sehr durch die Logs der Honeywall belastet wurde. Der Speicherplatz der Datenbank überschritt in den Winterferien leider trotzdem den Speicher der Festplatte. Beim Aufsetzen des Management-Servers hielten wir den Speicher für ausreichend und in den Ferien haben wir das leider nicht rechtzeitig bemerkt, um Gegenmaßnahmen ergreifen zu können. Deshalb gingen uns zu dieser Zeit einige Logdaten durch die Lappen. Über die andere NIC (eth1) werden wie oben angesprochen die Logdaten des IDS und der Firewall in die MySQL-Datenbank übertragen.

4.4.2 Zugriff von Außen

Die andere NIC (eth1) des Management PCs stellt die Verbindung zum Internet her, über die wir Projektmitglieder Zugang erhalten, um Logs auszuwerten oder die Firewall zu kontrollieren. Es kann eine gesicherte SSH-Verbindung zur Honeywall aufgebaut werden, um Logs zu empfangen oder eben die Maschine zu konfigurieren bzw. kontrollieren. Um den Management PC nicht zu gefährden ist die Kommunikation mit ihm nur über eine SSH Verbindung per Private Key Authentication über einen ungewöhnlichen Port (443) möglich. Darüber hinaus ist der Webserver nur lokal (per tunneling) erreichbar. Das heißt, es musste ein passender SSH-Tunnel erstellt werden, wodurch man auf den Webserver zugreifen kann.

Für die Analyse der Datenbank haben wir verschiedene Webinterfaces ausprobiert. Letztendlich haben wir für die Snort-Logs ACID [10] und für die iptables-Logs nulog [11] verwendet. Diese Interfaces sind aber nur bedingt vernünftig verwendbar. Einige der Statistiken haben wir per Hand (SQL Queries) generieren müssen.

4.5 Malware einfangen

Bei dem Rechner der Bots fangen soll, machen wir uns die Möglichkeit zunutze, viele IPs einer einzigen NIC zuordnen zu können, indem wir fast alle IPs des Subnetzes (von 141.62.88.3 – 141.62.88.250) auf dem Ethernetinterface dieser Maschine eingerichtet haben. Dadurch akzeptiert das Interface fast alle Pakete, die in unser Subnetz gelangen und zieht somit IP-Scans auch schneller auf sich.

Da auf dieser Maschine jedoch Linux läuft und die meisten Bots, Viren… für Windows konzipiert sind, haben wir verschiedene Tools eingesetzt, um die Windows Schwachstellen zu simulieren. Diese Werkzeuge lauschen auf den bekannten Ports, parsen die Anfragen auf Angriffsmuster und senden entsprechende Replies. Der Angreifer meint, er kommuniziert mit einer unsicheren Applikation. Kommt es nun zu der üblichen Attacke (meist Bufferoverflow Daten und angehängter Shellcode) wird die Nachricht auf Shellcode geparst und URLs werden herausgenommen. Das File (Malware) auf das die URL zeigt, wird dann heruntergeladen. Somit hat man die Dateien, welche normal ausgeführt werden isoliert und kann sie analysieren. Hier muss gesagt werden, die Tools sammeln natürlich nicht ausschließlich Bots, sondern eben auch Würmer, Trojanische Pferde, etc. Diese müssen dann später noch sortiert werden.

4.5.1 Mwcollect (v2)

Zuerst haben wir auf Mwcollect gesetzt, welches in der alten Version 2 gut funktioniert. Dieses Tool wird bei vielen Projekten auf der ganzen Welt eingesetzt und zu Beginn unseres Projektes war uns kein ähnliches Produkt bekannt. Leider speichert Mwcollect die Dateien nicht mit der MD5 Summe als Dateinamen. Viele identische Dateien müssen deshalb später von Hand aussortiert werden. Die neue Version (v3) haben wir ebenfalls getestet, jedoch war es auf unserem System nicht nutzbar, da es sich des öfteren grundlos verabschiedet hat und zudem sehr ineffektiv im Vergleich zur alten Version war.

4.5.2 Nepenthes

Nach einiger Zeit haben wir das Tool Nepenthes gefunden und ausprobiert. Das Ergebnis hat uns sehr überzeugt, weswegen wir nur noch dieses Tool laufen ließen. Mit Nepenthes haben wir - wie in den Statistiken erkennbar - deutlich bessere Ergebnisse erziehlt, da es einige Schwachstellen mehr simuliert als Mwcollect. Außerdem legt Nepenthes im Gegensatz zu Mwcollect die Malware nicht mit dem angegebenen Dateinamen ab, sondern verwendet dafür die MD5 Summe, womit kein Binary doppelt vorhanden ist. Anschließend haben wir die Dateien von der Platte kopiert, um sie auf anderen PCs analysieren zu können.

5 Die Java Drohne

5.1 Warum eigener Client

Beim Einlesen in die Thematik Botnets wurde immer wieder erwähnt, dass als Ausgangsbasis für die zentrale Einheit reguläre IRC-Server Software wie z.B. UnrealIRCd verwendet wird. Diese Software wird aber von den einzelnen Betreibern an ihre Bedürfnisse angepasst. Je nachdem wie groß das Informatikwissen ist und wie viel Programmierfähigkeiten die Person besitzt, wird die Sofware und somit das Verhalten des Server mal stark mal weniger stark modifiziert. Dies hat zur Folge, dass ein gebräuchlicher IRC-Client wie z.B. mIRC nicht mehr in der Lage ist sich mit diesem Server über das Standard IRC Protokoll zu verbinden und zu kommunizieren.

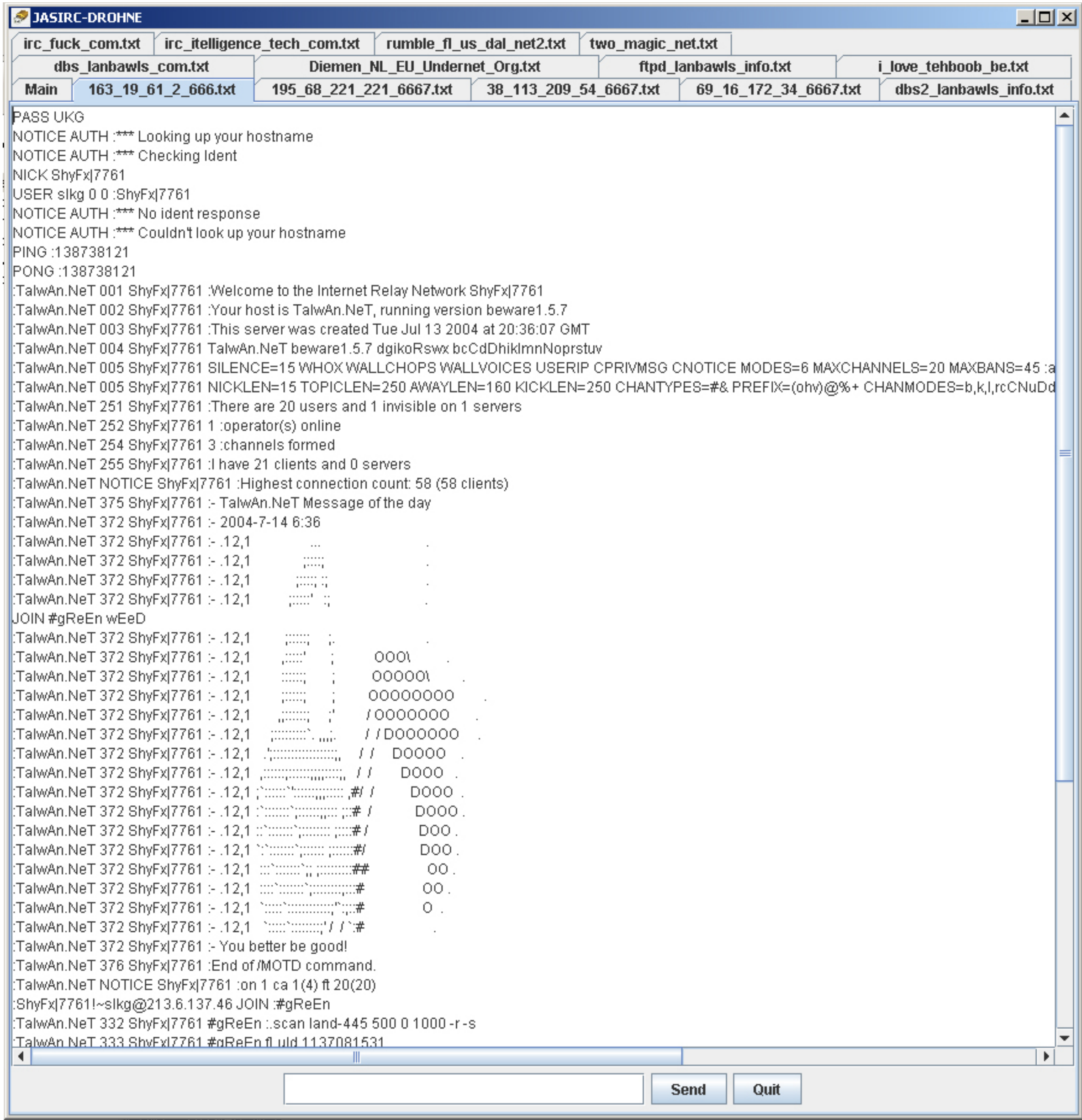
Die vorgenommenen Modifizierungen machen sich in verschiedenen Funktionen bemerkbar. Zum einen können die Loginsequenzen der einzelnen Botnets sehr unterschiedlich sind. Zum anderen wäre es auch möglich eine verschlüsselte Verbindung zu initialisieren. Es ist aber auch nicht unwahrscheinlich, dass beim "Ping Pong" - Challenge Response Verfahren verschiedene Algorithmen zum Einsatz kommen. In den meisten Fällen werden die Modifizierungen aber dazu verwendet, um die verwendbaren IRC Commands einzuschränken.

Die oben aufgeführten Modifikationen am IRC Server zielen immer darauf ab nicht authorisierten Personen (aus Sicht der Botnetbetreiber) den Zugriff auf und die Informationssammlung über den Server/Channel zu erschweren. Um sich an die gegebenen Umstände eines jeden Botnet schnell und einfach anpassen zu können, wurde ein eigener Client von uns entwickelt. Dieses Programm soll nicht nur zur schnellen Anpassung an die Modifikationen dienen sondern muss auch das Verhalten eines Bots wiederspiegeln.

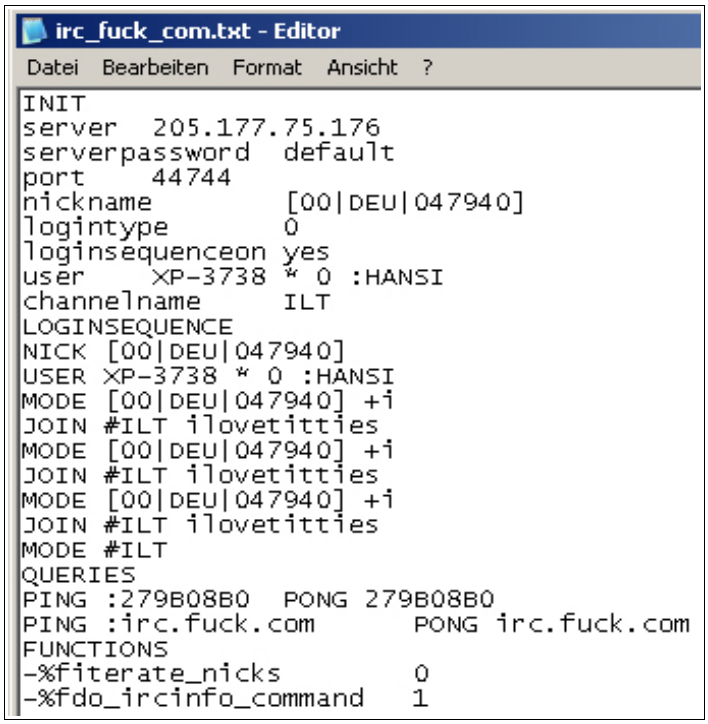
Konkret bedeutet dies, es muss in einem Command & Control Channel auf einen Command des Masters den entsprechenden Response zurückgeben. Dies ist notwendig, damit der Botnetmaster den eingeschmuggelten Botspion nicht erkennt. Würde der Master die Botatrappe doch entlarven könnte er diese aus dem Channel verbannen oder sogar sein komplettes Netz auf einen anderen Server umziehen.

5.2 Funktionsweise

Das Programm ist so konzipiert, dass es sich gleichzeitig mit mehreren Botnets verbinden kann. Die komplette Kommunikation, während einer Verbindung, wird von der Drohne mitprotokolliert und in einer Logdatei für das beobachtete Botnet gespeichert. Die Anpassung an verschiedene Botnets wurde so weit wie möglich durch Konfigurationsfiles gelöst, um eine einfache und schnelle Anpassung zu gewährleisten.



Die Abbildung zeigt die Java Drohne in Aktion. Über die Tabs können die einzelnen, geladenen Botnets ausgewählt werden. Der Textbereich des entsprechenden Tabs visualisiert die Kommunikation. Nachrichten und Kommandos die an den Server gesendet werden sollen, können über das Eingabefeld gesetzt und über den Button Send an den Server übertragen werden. Für oft verwendete Funktionen kann in der Konfiguration eine Art Alias gesetzt werden. Eine solche Funktion ist z.B. eine automatisierte, iterative whois Abfrage. Der Nickname setzt sich aus einem konstanten und einem dynamischen Teil zusammen. Der dynamische Teil wird durch eine Zahlenfolge von z.B. vier Stellen repräsentiert und kann somit über eine Schleife komplett erzeugt werden. Dadurch ist es schnell und einfach möglich genauere Informationen über alle Clients und über das Ausmaß des Botnet zu erhalten.
Das Folgende Bild zeigt die Ausführung einer Konfigurationsdatei.



Die Datei ist in vier Blöcke aufgeteilt: "INIT", "LOGINSEQUENCE", "QUERIES" und "FUNCTIONS". Innerhalb des "INIT"-Blocks werden die Userdaten wie Nickname, Passwort usw. hinterlegt. Zusätzlich werden hier auch Informationen über den Botnetserver gesetzt. Über die Option "loginsequence yes/no" kann das Verwenden des Blockes "LOGINSEQUENCE" gesetzt werden oder nicht. Die Option "loginsequence yes" kann verwendet werden, wenn kein Challenge/Response Verfahren während des Logins für das entsprechende Botnetz eingesetzt wird.

Der "LOGINSEQUENCE"-Abschnitt beinhaltet den Ablauf und die Nachrichten, die vom Client an den Server während des Logins gesendet werden. Wird ein Kommando vom Server an den Client gesendet, auf das dieser antworten muss, kann er sich aus dem "QUERIES"-Block entsprechende Antworten entnehmen. Die Kommandos stehen auf der rechten Seite und die notwendigen Antworten auf der linken Seite. Im letzten Abschnitt muss dem Client der Aufruf für die verwendeten Funkionen beigebracht werden. Mit dem Eintrag "-%fiterate_nicks 0" wird der zu verwendende Name für die oben beschriebene Funktion zum Iterieren der Namensliste gesetzt.

6 Vorgehensweise

Hier beschreiben wir nun unsere Vorgehensweise vom Fangen der Bots bis zur Analyse der IRC-Channels etwas genauer.

6.1 Fangen der Bots

Das Fangen der Bots haben, wie bereits beschrieben, propriätere Tools für uns übernommen. Wie in den Statistiken zu sehen ist, haben diese Tools gute Arbeit verrichtet (insbesondere Nepenthes) und eine große Anzahl Malware gefangen. Diese liegen nach Herunterladen auf der Festplatte und können zur Auswertung kopiert werden.

6.2 Auswerten der Bots

Die Auswertung der Bots haben wir aus Sicherheitsgründen zuhause über call-by-call Anbieter durchgeführt. Zuerst hat jeder von uns sein System sauber eingestellt, sprich Updates eingespielt, notwendige Tools installiert und dann mit einem Image/Backup-Tool ein Image erstellt, um den aktuellen Stand in wenigen Minuten wiederherstellen zu können. Um weitere Verbreitung der Bots zu unterbinden, wurden Verbindungen von typischen Ports (z.B. 445) komplett gesperrt.

Bevor die Malware ausgeführt wird, hat man die Möglichkeit durch Binärcode-Analyse Informationen über die Malware heraus zu finden. Das Herausfiltern von Klartextstrings liefert meist nur sehr wenig aufschlussreiche Daten, da sehr oft ein exe-Packer verwendet wurde, um die wichtigsten Teile des Programms zu schützen.

Hier ein kleiner Auszug einer solchen Analyse:

```
udpflood
pingflood
ping
tcpflood
email
Message sent to s.
```

Hier könnte man vermuten, dass es sich um Kommandos handelt, welche der Bot ausführen kann. Wir haben keinen Bot entdeckt, bei dem die Serverdaten so sichtbar wurden. Wir haben dann alle vorhandenen Tools gestartet, bevor wir die Malware selbst ausgeführt haben. Hier nun einige der Tools, die wir eingesetzt haben:

- *Registry Monitor* [12]
  Mit dessen Hilfe können jegliche Abfragen oder Änderungen in der Registry sichtbar gemacht und dem verantwortlichen Prozess zugeordnet werden.
- *File Monitor* [13]
  Dieser zeigt Erzeugung, Aufrufe, etc. von Files der verschiedenen Prozesse an. So kann man nachvollziehen, wohin sich ein Bot installiert.
- *TCP View* [14]
  Zeigt offene TCP und UDP Verbindungen mit den zugehörigen Prozesse an.
- *Process Explorer* [15]
  Ein sagenhaftes Tool, um Prozesse zu beobachten und zu analysieren. Da der Windows Task Manager völlig unbrauchbar ist und meistens die versteckten gefährlichen Prozesse nicht anzeigen kann, ist es unabdingbar hierbei ein anderes Tool einzusetzen.
- *Security Task Manager* [16]
  Dieser Task Manager stuft die Tasks nach ihrem Sicherheitsrisiko ein und soll somit eine Hilfe bei der Entdeckung von Malware leisten. Kriterien für die Kategorisierung sind unter anderem „Kann Tastatureingaben aufzeichnen“, „Kann andere Programme manipulieren“ und „Nicht sichtbares Fenster“
- *Ethereal* [17]
  Ein sehr bekannter Netzwerksniffer, der auch an der HdM gern verwendet wird. Mit dessen Hilfe lassen sich leicht Interaktionen mit einem IRC-Server ausfindig machen.

Diese Tools stellen nun eine Vielzahl an Informationen bereit, die wir alle in unsere Untersuchungen miteinbezogen haben. Jedoch hätte es meistens gereicht, den Netzwerkverkehr zu analysieren. Die IRC-Daten konnten wir meistens problemlos mitsniffen.

6.3 Analyse der IRC-Channels

Zu Beginn haben wir uns darin versucht, mit telnet eine Kommunikation mit dem Server aufzubauen, indem wir die IRC-Kommandos der Bots eingeschleust haben. Dies ist aber nur sehr bedingt nutzbar, da auf die Challenges relativ schnell geantwortet werden muss.
Nach der Fertigstellung unserer eigenen Java Drohne haben wir natürlich diese eingesetzt, um die Channels zu analysieren. Analog dazu haben wir zu jedem gefundenen Botnet einen Report erstellt. Bei den meisten Channels, die wir im Laufe unseres Projekts aufgespürt haben, bot uns der IRC-Dämon schon beim Login eine Vielzahl an Informationen. Dies kann man in dem Logauszug im Anhang, Kapitel 9.2 nachvollziehen.
Beispielsweise, wieviele User eingeloggt sind, die Uptime des Servers und vieles mehr. Darüber hinaus hatten wir eigentlich damit gerechnet, dass die Server sehr stark geändert werden, um herkömmliche Kommandos zu verhindern zu können, wie in Kapitel 3 vermerkt. Jedoch waren auf manchen Servern tatsächlich sehr mächtige Kommandos möglich, mit denen wir an mehr Details gekommen sind. Beispielsweise konnten wir auf einem Server die Daten von hunderten kompromittierten Rechnern abfragen bzw. mitloggen. Hierbei hätte es auch funktionieren müssen, die Daten des Betreibers auszulesen, jedoch war dieser niemals online, wenn wir live mitgeschaut haben. Die Betreiber dieser Netze sind nicht sehr vorsichtig ans Werk gegangen. Einige Mitschnitte mit Ergebnissen eingegebener Kommandos finden sich im Anhang.
Leider konnten wir nur die Interaktion eines Betreibers mitverfolgen, welcher die Topic des IRC-Channels geändert hat. Wir haben auch teilweise vermutet, dass weitere Channels bestehen, in denen sich die Betreiber verständigen können. Jedoch hatten wir nie die Möglichkeit, alle Channels anzusehen.

7 Ergebnisse

7.1 Statistiken

Hier werden einige Statistiken erläutert, welche aus den gewonnenen Daten erstellt wurden. Die dazugehörigen Diagramme befinden sich im Anhang in Kapitel 9.2.

- *Attackdiagram m*

  Dieses Diagramm wurde aus den Snortdaten erstellt. Auf diesem ist die Anzahl der jeweiligen Angriffskategorien verzeichnet, welche auf unser Honeynet gerichtet wurden.

- *TCP-Ports*

  Zeigt die Anzahl neuer Verbindungen und über welchen Port diese Verbindungen aufgebaut wurden. Daran erkennt man, warum zu Beginn ohne den Port 445 kaum Angriffe zu verzeichnen waren. Würde dieser Port von ISPs gesperrt werden, wäre ein Großteil des, durch Malware erzeugten Traffics ausgelöscht. Bei diesem Angriff handelt es sich um einen Bufferoverflow im Windows LSASS Dienst [18].
- *UDP-Ports*

  Im Vergleich zu den TCP-Verbindungen gingen sehr wenige Pakete ein. Bei den meisten Angriffen (Port 1434) handelte es sich um einen Pufferüberlauf in der MDAC-Funktion [19].
- *Connections per country*

  In diesem Diagram werden die Verbindungen den Ländern zugeordnet, von denen die Anfragen kamen. Klar zu sehen ist, dass unser liebstes Land USA mal wieder weit an der Spitze liegt, auch wenn es um die Verteilung von Malware geht. Alle Länder, von denen mehr als 20.000 Verbindungen ausgingen, sind auf der Landkarte im Anhang gekennzeichnet.
- *Malware Statistik Nepenthes*

  Hier sind zum einen die verschiedenen Malware Typen eingezeichnet und zum anderen die verschiedenen Varianten des jeweiligen Typs. Hierbei geht es auch wieder allgemein um Malware, wobei Bots den Großteil der Malware ausmachen.
- *Malware per day Nepenthes*

  Die Anzahl der Malware pro Tag, die mit Nepenthes eingefangen wurde. Hier sieht man, dass Nepenthes sehr „fleißig“ war. Der Durchschnittswert entspricht etwas mehr als 8 Binaries pro Tag.

7.2 Entdeckte IRC-Channels

Bei der Auswertung der Bots, wurden von uns 15 unterschiedliche Botnetze entdeckt. Einige davon waren bereits kurz nach der Auswertung schon gar nicht mehr zu erreichen. Effektiv konnten 4-5 aktive Botnetze bis zum Ende des Projektes verfolgt werden. Eine große Dunkelziffer der analysierten Bots versuchten sich mit Hilfe eines DNS Namens, der nicht mehr auflösbar war, mit einem Server zu verbinden. Diese Bots sind somit alle herrenlos und dienen nicht einmal mehr einem bösartigen Zweck. Der Ausbreitungsmechanismus ist aber trotzdem noch aktiv und erzeugt somit ein Grundrauschen im Internet. Einige der untersuchten Bots loggten sich sogar in einen öffentlichen Server ein.

Diese verwendeten nicht die typischen Namen mit einem statischen und einem dynamsichen Teil, sondern benutzten normale Namen aus einer vordefinierten Liste. Die meisten Botnets hatten eine Gößen von 4000-6000 Drohnen. Eine Ausnahme stellt das kleinste beobachtete Botnet mit einem Umfang von 58 annektierten Clients dar. Dieses Botnet war dafür aber schon seit dem 13 Juli 2004 online. Einige Logauszüge sind im Anhang zu finden.

7.3 Gesammelte Erfahrungen

Hier nun ein paar Erfahrungen, die wir mit den Bots bei der Auswertung gesammelt haben. Manche der Punkte beziehen sich auf die Recherche in Kapitel 3.

- *Bots sperren Anwendungen*
  Nach Ausführung mancher Bots war es nicht mehr möglich Datenverkehr mit Ethereal aufzuzeichnen. Der Sniffer und z.B. auch der Internet Explorer konnten nicht mehr gestartet werden.
- *Einfach gestrickte Bots*
  Die untersuchten Bots waren ziemlich einfach gestrickt. Sie waren nicht, wie oft im Internet zu lesen war, mit Verschlüsselungsmechanismen bestückt oder hatten Rootkits im Gepäck. Meist versteckten sie sich vor dem Windows Task Manager, aber mit anderen Tools war dies kein Hindernis.
- *Bots erschweren die norm ale Arbeit*
  Wir haben die Erfahrung gemacht, dass die Bots meist den CPU ziemlich unter Beschlag nehmen, viel Netzwerktraffic generieren und ebenso viele Abfragen und Änderungen in der Registry vornehmen. Dadurch wird das Arbeiten an diesen Maschinen manchmal fast unmöglich.
- *Botnets sind m eist sehr schlecht geschützt*
  Wir hatten bei den gefundenen Botnets keine großen Probleme, uns mit einem anderen Client bzw. unserer Java Drohne zu den Servern zu verbinden. Und auch die erwarteten schweren Modifikationen der IRC-Server blieben aus. Wie bereits beschrieben, war es des öfteren möglich, Kommandos im Channel auszuführen.
- *Botnetbetreiber schützen sich relativ gut*
  Die meisten gefundenen Server waren in der DNS Registrierungsdatenbank nicht einfach auf reale Personen angemeldet, sondern es wurden Anonymisierungs-dienste genutzt. Wie in einem der Botnetreports im Anhang zu sehen, ist hier der Dienst protectfly.com eingetragen. Dadurch kann man die eigentliche Person nicht direkt belangen.

8 Ausblick

Durch das Projekt haben wir einen sehr tiefen Einblick in das Thema und die Problematiken bekommen. Es ist klar, dass hier das Ende der Fahnenstange noch lange nicht erreicht ist. Wir hoffen auf das Interesse von Kommilitonen. Während des Semesters kamen einige interessante Ideen auf, die aus Zeitgründen leider nicht weiterverfolgt werden konnten.

8.1 Sandboxumgebung zur Auswertung von Malware

Die Analyse von Hand (mit Sniffer, Einspielen des Imagebackups) hat sehr viel Zeit in Anspruch genommen. Eine automatisierte Lösung inkl. Virenscanner, Dateisystem-, Registrymonitor würde die Produktivität bei der Auswertung erheblich steigern. Schon bestehende Möglichkeiten ([20], [21], [22]) konnten nur wenig weiterhelfen.

8.2 Informationsgewinnung durch Binärcodeanalyse

Durch die automatisierte Untersuchung des Binärcodes (ähnlich einem Virenscanner) könnten die wichtigsten Informationen auch ohne Ausführung der Malware herausgefunden werden. Geeignete Algorithmen wären bei dieser Aufgabe um vielfaches schneller als gewöhnliche Methoden zur Analyse.

8.3 Botnets lahmlegen

Nachdem der zentrale Punkt eines Netzes (IRC-Server) gefunden wurde, ist Geduld gefragt. Der Betreiber muss aktiv werden, um weitere Informationen über diesen herausfinden zu können. Kann ein Botnet übernommen werden, könnten auf allen Zombies beispielsweise Betriebssystemupdates installiert werden. Auch in Zusammenarbeit mit Behördern könnten aktive Netze stillgelegt werden. Doch wie ist dabei vorzugehen?

8.4 Allgemeine Beschreibungsform für Botnets

Die vielen Informationen eines gefundenen Bots/Botnets in einer Textdatei abzulegen ist nur wenig produktiv und fehlerträchtig. Ein ausgefeiltes Datenbankschema (inkl. Webinterface) könnte hier sehr nützliche Dienste leisten! Auch Zusammenhänge zwischen Botnets können so möglicherweise gefunden werden.

8.5 Einfaches Honeynet (Roo)

Der Aufbau eines Honeynets erfordert viel Wissen über z.B. Firewalls, IDS, Logging usw., was die Administration nicht ganz einfach macht. Eine umfassende Lösung, die einfach einzusetzen ist, existiert bereits [23]. Verbesserungen und Erweiterungen an der CDROM Roo könnten der Ausbreitung von Honeynets sicher weiterhelfen.

8.6 Was war, was wird

Die von uns erwarteten Möglichkeiten der sicheren Kommunikation in Botnets sind laut unseren Ergebnissen noch lange nicht ausgeschöpft. Alles deutet darauf hin, dass die Botnets sich sehr schnell vergrößern. Bisher sind uns nur Netze mit einem zentralen Server bekannt. Dieser Einstiegspunkt für die Zombie PCs ist ein sogenannter single-point-of-failure. Es ist gut vorstellbar, dass eine dezantrale Organisation - wie bei peer-to-peer – bald eingesetzt wird. Ein kleiner Schritt für mehr Unauffälligkeit ist es, öffentliche IRC-Server zu benutzen. Durch Algorithmen erzeugte Nicknames erregen Aufsehen und werden gesperrt. Deshalb nutzen diese Bots wohl festprogrammierte Namenslisten. Eine Spionage der Kommunikation zwischen Bot und IRC-Server kann ganz einfach auf einer dazwischen platzierten Honeywall durchgeführt werden. Eine verschlüsselte Verbindung hingegen könnte auf diese Weise nicht belauscht werden. Auch hier steckt noch gewaltiges Entwicklungspotenzial. Durch bessere Verwaltung und Sicherheit der Bots können die Netze bestimmt noch anwachsen und für noch gewaltigere Angriffe genutzt werden. Diese akute Problematik wird weiterhin für mehr Kriminalität im World Wide Web sorgen. Jeder Internetbenutzer kann unbewusst Mitglied eines Botnets werden. Mit dieser noch relativ neuen Bedrohung steht man jedoch nicht alleine da. Aber auch von Seiten der Gegener gibt es immer wieder Erfolge zu vermelden. Beispielsweise an der Entwicklung der Werkzeuge ist gut zu erkennen, dass die Community wächst und eine Zusammenarbeit statt findet.

9 Anhang

9.1 Linksammlung

[1] http://www.atanneberg.de/index.php?option=com_content&task=view&id=409&Itemid=37

[2] http://www.zdnet.de/news/tkomm/0,39023151,39137266,00.htm

[3] http://www.pcwelt.de/news/sicherheit/124176/index.html

[4] http://www.zdnet.de/news/tkomm/0,39023151,39140275,00.htm

[5] http://www.scmagazine.com/uk/news/article/538412/fastest-growing-malware-threat-bots

[6] http://www.mwcollect.org

[7] http://www.netfilter.org

[8] http://joker.linuxstuff.pl/specter/

[9] http://www.snort.org

[10] http://www.andrew.cmu.edu/user/rdanyliw/snort/snortacid.html

[11] http://www.inl.fr/Nulog.html

[12] http://www.sysinternals.com/Utilities/Regmon.html

[13] http://www.sysinternals.com/Utilities/Filemon.html

[14] http://www.sysinternals.com/Utilities/TcpView.html

[15] http://www.sysinternals.com/Utilities/ProcessExplorer.html

[16] http://www.neuber.com/taskmanager/

[17] http://www.ethereal.com

[18] http://nvd.nist.gov/nvd.cfm?cvename=CVE-2003-0533

[19] http://www.microsoft.com/germany/technet/sicherheit/bulletins/ms04-003.mspx

[20] http://sandbox.norman.no/live_4.html

[21] http://www.malwareupload.com

[22] http://virusscan.jotti.org/

[23] http://www.honeynet.org/tools/cdrom/

9.2 Auszüge und Dokumente

Die folgenden Auszüge und Dokumente wurden in die Datei „Anhang.pdf“ ausgelagert:

- Botnetreport 1 und Botnetreport 2
  Diese sind Beispiele für die Botnetreporte wie sie für die einzelnen Botnets angefertigt wurden.
- Attackdiagramm, TCP-Ports, UDP-Ports, Connections per country, Connections per country (2), Malware Statistik Nepenthes, Malware per day Nepenthes
- Logauszug
  Zeigt ein Beispiel einer Logdatei, wie sie von der Java Drohne erstellt werden.