Überblick über Botnetz-Erkennungsmethoden

Florian Zipperle
Betreuer: Oliver Gasser
Seminar Future Internet SS2014
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik, Technische Universität München
Email: florian.zipperle@tum.de

KURZFASSUNG

Botnetze sind eine große Sicherheitsbedrohung des Internets, viele gefährliche Angriffe nutzen sie. Bei Botnetzen handelt es sich um von Kriminellen ferngesteuerte Computer, die zu einem Netzwerk zusammengeschlossen sind. Zur Fernsteuerung werden die Computer mit Schadsoftware infiziert. Diese Arbeit gibt eine Einführung und einen Überblick über das Thema Botnetze und wie sie anhand ihrer Architektur und ihrer Kommunikationsinfrastruktur unterschieden werden können. Hauptsächlich gibt es zentrale Infrastrukturen nach dem Client-Server-Model, Peer-to-Peer-Architektur und kombinierte hybride Architekturen. Anschließend wird in der Arbeit eine mögliche Kategorisierung der aktuellen Methoden der Botnetzerkennung gezeigt. Die Verfahren werden in signatur- und verhaltensbasierten Methoden eingeteilt. Durch Netzwerküberwachung wird nach auffälligem Verhalten und bekannten Mustern gesucht, um Bots im Netzwerk zu erkennen. Zu den einzelnen Kategorien werden aktuelle Forschungsergebnisse vorgestellt.

Schlüsselworte

Botnetze, Botnetz-Erkennung, Übersicht

1. EINLEITUNG

Diese Arbeit soll einen Überblick über Botnetze und die Methoden zu ihrer Erkennung geben. Botnetze sind wie Viren und Würmer eine große Sicherheitsbedrohung des Internets, sowie seiner Infrastruktur. Botnetze sind mitunter die gefährlichsten von ihnen [8]. Deshalb ist eine gute und zuverlässige Erkennung notwendig und Botnetz-Erkennung ein aktuelles Forschungsgebiet.

Dabei strukturiert sich diese Arbeit im weiteren Verlauf folgendermaßen: im ersten Kapitel werden Botnetze kurz beschrieben und ihre Verwendungszwecke aufgezeigt, in Kapitel zwei folgt eine genauere Kategorisierung und Beschreibung von Botnetzen. In Kapitel drei wird eine Klassifizierung der verschiedenen Ansätze zur Botnetzerkennung beschrieben. Anschließend in Kapitel vier wird ein Überblick über konkrete Forschungsergebnisse gegeben. In Kapitel fünf folgt noch der Schluss der Arbeit mit einem Ausblick.

1.1 Was sind Botnetze?

Der Name Botnetz setzt sich zusammen aus Bot und Netz, Bot kommt von Roboter. Sie sind Netzwerke bestehend aus Computern, welche von Schadsoftware befallen sind. Durch diese Schadsoftware können vom sogenannten Botmaster das Netzwerk bzw. die einzelnen Computer gesteuert werden. Die Steuerung erfolgt über eine sogenannte Command-and-Control-Infrastruktur (C & C) , welche im Verborgenen abläuft. Ein Computer, welcher sich in einem Botnetz befindet, wird meist ohne Kenntnis des Besitzers missbraucht [18].

1.2 Bedrohung durch Botnetze

Botnetze werden meist für kriminelle Zwecke missbraucht. Da Botnetze mehrere tausend Computer umfassen können, bieten sie eine große Rechenleistung sowie eine große Datenbandbreite im Internet für den Angreifer [9]. Der Angreifer ist im Falle von kriminellen Aktivitäten zusätzlich durch die Schicht der Botnetze geschützt, d.h. für das Opfer ist der Angreifer nicht sichtbar sondern nur ein übernommener Computer des Botnetzes. Als zusätzlichen Schutz vor Strafverfolgung kann der Botmaster den Angriff so planen, dass zwischen dem Befehl und dem tatsächlichen Angriff eine beliebige Zeitspanne herrscht. Dadurch wird es schwieriger die Betreiber des Botnetzes zur Rechenschaft zu ziehen [26].

Einige Bedrohungen und Verwendungszwecke von Botnetzen sind [5] [18]:

- **DDoS:** Als Distributed-Denial-of-Service, kurz DDoS, bezeichnet man eine Attacke gegen einen Server, bei welcher sehr viele Rechner gleichzeitig fehlerhafte bzw. unsinnige Anfragen an einen Server schicken. Durch die Flut der fehlerhaften Pakete kann der Server überlastet werden und gültige Anfragen nicht mehr beantworten. Dadurch kann er seine eigentliche Aufgabe nicht mehr wahrnehmen. Dies stellt insbesondere für eCommerce Anbieter ein Problem dar, da sie dadurch wirtschaftlichen Schaden erleiden können.

- **Spam:** Die übernommenen Computer können dazu missbraucht werden Spam zu versenden. Diese Spam Mails können verwendet werden, um Schadsoftware zu verteilen, welche das Botnetz vergrößern. Durch das Versenden von Spam kann ein Rechner auf eine Blacklist kommen und dadurch keine Mails mehr versenden. Laut einer Schätzung von 2009 erzeugen Botnetze 85 % der über 100 Milliarden Spamnachrichten, die täglich versendet werden [15].

- **Klickbetrug:** Durch Klickbetrug kann mit Botnetzen Geld verdient werden. Dabei richtet der Botnetzbetreiber zuerst ein Konto bei einem Online-Werbesystem wie z.B. Google AdWords ein und stellt anschließend

doi: 10.2313/NET-2014-08-1_03

eine Webseite online, welche diese Werbung anzeigt. Danach steuert er die Bots so, dass sie seine Seite besuchen und die Werbung anklicken. Da dies meist ohne des Wissens des Computerbesitzers im Hintergrund geschieht, entspricht dies nicht dem Ziel der Werbetreibenden und es wird deshalb von Klickbetrug gesprochen. Der Botnetzbetreiber kann diesen Service auch vermieten, sodass andere an der Werbung verdienen.

- **Identitätsdiebstahl:** Durch das Aufzeichnen der Tastatur des befallenen Computer können Passwörter oder Kreditkartendaten gestohlen werden. Diese Daten können für Identitätsdiebstahl missbraucht werden und Onlinekonten können dadurch übernommen werden. Es kann auch gezielt nach gespeicherten Passwörtern und Benutzerdaten auf dem Rechner gesucht werden. Die gestohlenen Informationen können vom Botnetzbetreiber, zur Aufrechterhaltung seines Botnetzes genutzt werden, z.B. um Domains zu reservieren ohne seine eigene Identität preis zu geben.

2. BOTNETZE

Die Botnetze besitzen meist den gleichen Lebenszyklus, unterscheiden sich jedoch oft in ihrer Architektur sowie in ihrer Art der Kommunikation.

2.1 Lebenszyklus

Der Lebenszyklus eines Bots bzw. dessen Schadsoftware innerhalb eines Botnetzes lässt sich grob in fünf Abschnitte einteilen[5]:

- **Infizierung:** Zu Beginn wird ein Computer durch die Botsoftware infiziert. Dies geschieht durch Sicherheitslücken im System oder durch Schadsoftware, die durch den Benutzer des Computers ausgeführt wird.
- **Sammeln:** Falls die Infizierung erfolgreich war und eine Verbindung zum Internet besteht, meldet sich die Schadsoftware über den Command-and-Control-Server beim Botmaster.
- **Sichern und Verstecken:** Die Schadsoftware muss sich im System verstecken, um nicht vom Benutzer entdeckt und entfernt zu werden. Für diesen Zweck wird auch versucht gebräuchliche Antivirensoftware zu deaktivieren bzw. sie zu manipulieren. Außerdem kann versucht werden zusätzliche Schadsoftware über das Internet herunterzuladen.
- **Befehle ausführen:** Der Bot ist einsatzbereit und kann verwendet werden. Er wartet auf Befehle vom Botmaster. Sobald er Befehle über die Command-and-Control-Infrastruktur erhält, versucht er diese auszuführen. Dies können Updates der Botsoftware sein, um die Entdeckung zu erschweren oder um neue Funktionen zu erhalten. Es können auch die in 1.2 beschriebenen Attacken ausgeführt werden.
- **Tod:** Der Tod eines Bots kann mehrere Ursachen haben. Der Bot kann entdeckt worden sein oder das Betriebssystem des Computers wird neu installiert. Es kann auch vorkommen, dass der Botmaster aus Vorsicht die Schadsoftware vom Computer löscht, um nicht erkannt zu werden und keine Spuren für Forscher und Ermittler zu hinterlassen.

2.2 Architektur

Es existieren hauptsächlich zwei verschiedene Architekturen, eine zentralisierte und eine verteilte Struktur. Die zentralisierte Struktur setzt auf das Client-Server-Modell und die verteilte Struktur auf Peer-to-Peer-Protokolle [18].

2.2.1 Zentralisierte Struktur

In der zentralisierten Struktur wird auf das im Internet üblichen Client-Server-Modell zurückgegriffen. Der Bot fungiert als Client und der Botmaster als Server. Der Botmaster betreibt einige wenige zentrale Server, die zur Steuerung und Kommunikation mit den Bots dienen. Da alle Bots mit den zentralen Servern verbunden sind, kann der Botmaster mit allen zeitgleich und schnell kommunizieren. Er erhält direkte Rückmeldungen über den Status und die Größe seines Botnetzes. Die zentralen Server bieten allerdings eine gute Angriffsfläche für Gegenmaßnahmen. Durch das Ausschalten der zentralen Server kann das ganze Botnetz lahmgelegt werden.

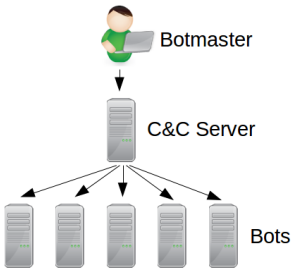

Abbildung 1: Zentralisierte Architektur mit einem Command-and-Control-Server, über welchen der Botmaster sein Botnetz steuert.

Als Beispiel für ein Botnetz mit zentralisierter Architektur kann das Botnetz Rustock genannt werden. Es wurde 2006 das erste Mal wahrgenommen und konnte 2011 mit Hilfe von Reverse Engineering unschädlich gemacht werden [7].

2.2.2 Verteilte Struktur

Um das Botnetz robuster zu gestalten, setzen Botnetzentwickler auf eine verteilte Architektur. Diese basiert meist auf dem Peer-to-Peer-Prinzip. Dabei gibt es nicht einen Server und viele Clients, sondern jeder Bot kann die Rolle eines Servers und eines Bots übernehmen. Die Erkennung, wer der Botmaster ist, fällt dadurch schwerer. Die Befehle, die der Botmaster gibt, werden im Botnetz verteilt. Die Bots tauschen sich dafür aus, der Bot mit dem älteren Befehl verwirft diesen und übernimmt den neuen Befehl. Dadurch, dass die Befehle nicht zeitgleich von einem zentralen Server geholt werden können, sondern sich erst im Netz ausbreiten müssen, kann dies zu einer geringeren Reaktionszeit führen [17][18].

In [25] werden die zwei Peer-to-Peer-Botnetze Strom und Nugache vorgestellt. Die beiden werden einander gegenübergestellt und verglichen. Beide setzten auf eine verteilte Struktur, Nugache besitzt jedoch auch eine zentralisierte Kommunikationsmöglichkeit über IRC. In der Praxis wird diese jedoch kaum verwendet. Beide Botnetze verwenden eine Verschlüsselung für die Kommunikation. Das Strom Bot-

doi: 10.2313/NET-2014-08-1_03

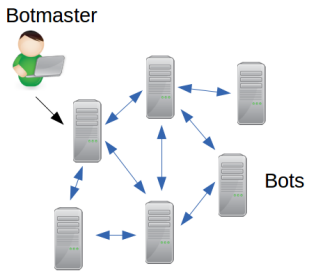

Abbildung 2: Verteilte Architektur, die Bots sind untereinander vernetzt und verteilen die Befehle des Botmasters selbstständig im Netz.

netz setzt für die Kommuniktion auf dein bestehendes Protokoll namens Overnet. Damit ein neuer infizierter Rechner sich dem Botnetz anschließen kann, braucht er einen Kontaktpunkt. Dieser Kontaktpunkt wird durch andere Bots die sich bereits im Botnetz befinden geben. Die IP-Adressen verschiedener Kontaktpunkte sind dabei hart-kodiert im Schadprogramm enthalten. Der neue Bot versucht Verbindungen zu den Kontaktpunkten herzustellen und sich dem Botnetz anzuschließen. Die Liste mit den Kontaktpunkten wird dabei aktualisiert.

2.2.3 Hybride Architektur

Die Vorteile beider Architekturen können in einem hybriden Ansatz vereint werden. Dabei existieren mehrere verteilte Netze mit jeweils einigen wenigen zentralen Servern. Dadurch ist das Netz robust gegen Gegenmaßnahmen und kann trotzdem schnell und übersichtlich gesteuert werden [22].

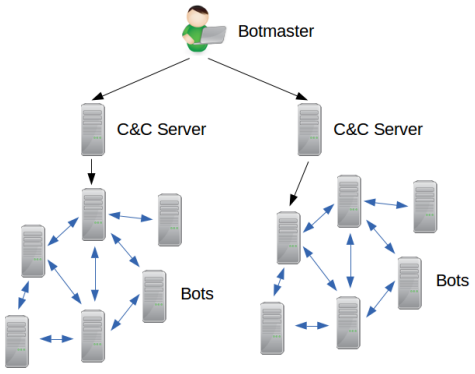

Abbildung 3: Die hybride Architektur stellt eine Kombination dar. Es gibt einige verteilte Netze, welche jeweils über einen eigenen zentralen Command-and-Control-Server verfügen. Dies kombiniert die Vorteile der zentralisierten und verteilten Architektur.

Ein Beispiel für ein konkretes Botnetz, welches auf eine hybride Architektur setzte, nannte sich Waledac. Es setzte einige zentrale Server ein. Diese Server steuerten ein darunterliegendes selbst entwickeltes Peer-to-Peer-Protokoll. Waledac wurde im April 2008 das erste Mal entdeckt und wurde 2010 im Februar erfolgreich ausgeschaltet [7].

2.3 Kommunikation

Zur Kommunikation mit dem Botnetz und zur Realisierung der Command-and-Control-Struktur können verschiedene Infrastrukturen und Methoden verwendet werden. Die ersten Botnetze verwendeten meist Internet-Relay-Chat (IRC) Protokolle zur Kommunikation. IRC ist für große Chaträume ausgelegt und es existieren einige quell-offene Implementierungen. Dies eignet sich, um als Botmaster schnell und einfach mit dem Botnetz zu kommunizieren [10]. Es existieren auch weiterhin viele Botnetze, die auf IRC setzen. Viele der komplexen und gefährlichen Botnetze setzen jedoch verschlüsselte Kommunikation und keine zentralen Server ein [7][29].

Ein anderer Ansatz zur Kommunikation ist es normalen HTTP-Netzwerkverkehr zu verwenden. Dabei können ganz normal per GET und POST Daten ausgetauscht werden. Die Kommunikation lässt sich aber auch verstecken, indem z.B. über eine öffentliche Kommentarfunktionen eines sozialen Netzwerks die Daten getauscht werden [12]. Befehle lassen sich ebenfalls in Bildern oder anderen Dateien verstecken. Diese können über soziale Netzwerke verteilt werden. Dadurch wird die Erkennung erschwert, da der Netzwerkverkehr kaum von normalem unterscheidbar ist [30].

Peer-to-Peer-Netze verwenden teilweise bestehende Peer-to-Peer-Netze, die für Filesharing ausgelegt sind. So wird Beispielsweise das Framework WASTE, welches ein verschlüsseltes Peer-to-Peer-Protokoll zur privaten Kommunikation implementiert, verwendet [12]. Durch die Verwendung von Peer-to-Peer-Protokollen kann weitestgehend auf einen zentralen Server verzichtet werden und es kann eine verteilte Architektur verwendet werden. Die Übersicht für den Botmaster über sein Botnetz, welche zum Beispiel bei IRC gegeben ist fehlt, eben weil der zentrale Punkt fehlt.

3. ERKENNUNGSVERFAHREN

Die verschiedenen Methoden zur Erkennung von Botnetzen lassen sich wie in diesem Kapitel dargestellt einteilen. Dabei werden verschiedene Methoden und Ansätze je nach Architektur und Kommunikationsinfrastruktur verwendet. In Abbildung 4 ist eine mögliche Kategorisierung der Erkennungsverfahren gegeben. Die einzelnen Kategorien werden in den folgenden Unterkapiteln näher betrachtet.

Erkennungsmethoden lassen sich in aktive und passive Erkennungsmethoden unterteilen. Aktive Verfahren bieten dem Botmaster die Möglichkeit zu erfahren, dass sein Botnetz untersucht wird. Dadurch kann er sein Botnetz ändern und es durch bessere Methoden absichern und schützen. Passive Verfahren hingegen sind für den Botmaster nicht erkennbar [3].

3.1 Honeynet

Honeynets sind Netzwerke, die ein oder mehrere Honeypots enthalten. Als Honeypot wird in diesem Zusammenhang ein Computer bezeichnet, der keine produktive Funktion hat außer Schadsoftware anzulocken. Um mit Schadsoftware infiziert zu werden, wird oft auf veraltete mit Sicherheitslücken behaftete Software verwendet. Jeder Netzwerkverkehr, der durch ein Honeypot verursacht wird, deutet auf Schadsoftware hin. Eingehender Netzwerkverkehr kann z.B.

doi: 10.2313/NET-2014-08-1_03

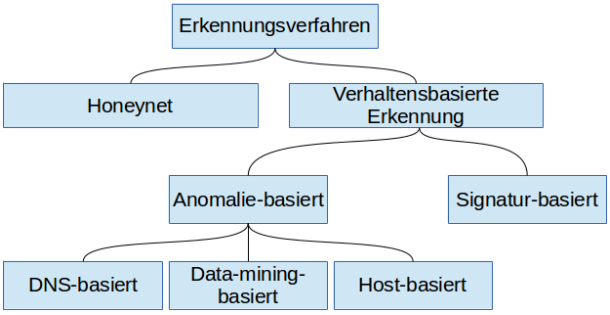

Abbildung 4: Einteilung der verschiedenen Erkennungsmethoden für Botnetze [1].

ein Portscan sein oder sonstige Attacken. Entsteht ausgehender Netzwerkverkehr, bedeutet dies dass der Computer von Schadsoftware befallen ist [13]. Aus diesen Daten lassen sich Schlüsse über die Art des Botnetzes ziehen und wie es kommuniziert. Es können neue Schwachstellen entdeckt werden, welche es ermöglichen in das System einzudringen. Außerdem können die Vorgehensweise und die verwendeten Werkzeuge des Angreifers analysiert werden. Honeynets werden von Forschern eingesetzt oder auch von Internet Service Providern, um zu erkennen, ob in ihrem Netz Bots eines Botnetzes vorhanden sind [19].

3.2 Verhaltensbasierte Erkennung

Honeynets sind nützlich, um bekannte Botnetze zu verstehen und zu analysieren. Falls ein Computer infiziert ist, aber nicht versucht andere Computer ebenfalls zu infizieren, bleibt er für ein Honeynet unentdeckt. Für die Erkennung solcher Bots sind die Ansätze von verhaltensbasierten Verfahren besser geeignet [10]. Unter verhaltensbasierter Erkennung werden Methoden zusammengefasst, welche versuchen das besondere Verhalten eines Bots zu erkennen. Dies geschieht meist durch Analyse und Beobachtung des Netzwerkverkehrs, das ein Computernetzwerk generiert. Dies kann z.B. durch einen Internet-Service-Provider geschehen.

3.2.1 Anomalie-basierte Erkennung

Das Verhalten von Computern wird auf Anomalien untersucht. Dies können hohes Netzwerkverkehrsvolumen, Verwendung von unüblichen Ports oder sonstiges unübliches Systemverhalten sein. Diese Ansätze können nochmals feiner gegliedert werden [1]: **DNS-basierte Erkennung**
Bei DNS-basierten Erkennungsverfahren wird vor allem der durch DNS erzeugte Netzwerkverkehr untersucht. Als Domain-Name-System (DNS) wird der Dienst bezeichnet, welcher Domains in IP-Adressen auflöst. Dies wird von Botnetzen verwendet, um mit dem Command-and-Control-Server eine Verbindung aufzubauen. Durch die Zuhilfenahme von DNS müssen die IP-Adressen der Server nicht im Code des Bots stehen, sondern können dynamisch aufgelöst werden. Dies ist nützlich für einen Serverumzug.

Ein Grund weshalb DNS-basierte Methoden zur Erkennung verwendet werden, ist dass der durch DNS erzeugte Netzwerkverkehr im Verhältnis sehr klein ist. Dadurch fällt das Analysieren und Auswerten der Daten an einem großen Netzwerkknoten einfacher. Es wird also weniger Rechenleistung benötigt, wodurch unter Umständen der Netzwerkverkehr in Echtzeit durchsucht werden kann[23].

Domains können relativ einfach gesperrt werden, falls bekannt ist, dass sich dahinter ein Botnetzbetreiber befindet. Um die daraus resultierende Gefahr für Botnetze zu vermeiden, werden sogenannte Domainname-Generation-Algorithmen (DGA) verwendet, um dynamisch neue Domains bzw. Subdomains zu erzeugen. Möchte der Bot mit dem Comand-and-Control-Server kommunizieren, generiert er solange neue Domains mit Hilfe eines DGA bis er einen gültigen Server findet. Dies liefert einen Angriffspunkt gegen Botnetzbetreiber [23].

Data-Mining-basierte Erkennung
Data-Mining bedeutet aus gegebenen Daten neues Wissen zu sammeln. Bei Data-Mining-basierten Erkennungsmethoden werden Verfahren aus dem Bereich Data-Mining angewandt, um den überwachten Netzwerkverkehr zu klassifizieren und zu gruppieren. Gruppierung wird bei vielen Erkennungsverfahren eingesetzt. Dies ist oft der erste Schritt, bei welchem versucht wird der von einem Computer verursachte Netzwerkverkehr in zwei Klassen einzuteilen. Es wird versucht der Netzwerkverkehr, der von Bots generiert wird, vom normalen Netzwerkverkehr zu unterscheiden und die infizierten Computer in einem Netzwerk zu erkennen. Dadurch ist es möglich große Datenmengen auf kleiner Datenmengen zu reduzieren, wodurch weniger Rechenleistung erforderlich ist. Dies ermöglicht unter Umständen Echtzeituntersuchgen oder komplexere Algorithmen. Es werden Algorithmen für maschinelles Lernen angewandt, diese können zum Beispiel zur Erzeugung von Signaturen verwendet werden [1] [3].

Host-basierte Erkennung
Bei dieser Herangehensweise zur Erkennung von Botnetzen wird nicht das Netzwerk überwacht und untersucht, sondern die Suche findet auf den Computern statt. Ist ein Computer von Schadsoftware befallen und der Bot aktiviert, lässt sich dies oft durch ungewöhnliche Systemaufrufe erkennen. Die Schadsoftware versucht sich im System zu verstecken und z.B. Antivirenscanner zu deaktivieren. Ein Bot versucht auch Verbindungen in das Internet aufzubauen [14].

3.2.2 Signaturbasierte Erkennung

Signaturbasierte Verfahren sind ähnlich zu Anomalie-basierten Methoden [28]. Sie verwenden vorhandenes Wissen über Botnetze, dieses Wissen kann mit Hilfe von Honeynets gesammelt werden. Die daraus resultierende Signatur bzw. das typische Verhalten wird in einfache Regeln übertragen. Ein Computer gilt als verdächtig, wenn er diese Regeln erfüllt [4]. Das Prinzip ist ähnlich wie bei einem Virenscanner auf einem Computer [6]. Da vorhandenes Wissen verwendet werden muss, um die Regeln aufzustellen, eignen sich reine signaturbasierte Methoden schlecht, um unbekannte Botnetze zu erkennen.

3.3 Hybride Verfahren

Die konkrete Einteilung in diese genannten Kategorien ist für reale Erkennungsverfahren schwierig. Da die meisten Verfahren Methoden aus den verschiedenen Bereichen verwenden. Die Kategorisierung gibt aber trotzdem einen groben Überblick welche Verfahren eingesetzt werden. Diese Verfahren welche Kombinationen enthalten werden als hy-

doi: 10.2313/NET-2014-08-1_03

bride Verfahren bezeichnet werden. Da durch die Kombination der einzelnen Methoden die Stärken der verschiedenen Ansätze vereint und die Schwächen kompensiert werden können. Diese hybriden Verfahren haben deshalb meist bessere Erkennungsraten oder bessere Laufzeiten [1].

In [2] wird ein hybrider Ansatz präsentiert, der Host-basierte und Data-mining-basierte Methoden vereint, um Peer-to-Peer-Botnetze zu finden.

4. FORSCHUNGSERGEBNISSE

Hier werden einige konkrete Forschungsergebnisse präsentiert. Zunächst werden verschiedene Kriterien gezeigt, die zur Bewertung der Güte einer Erkennungsmethode verwendet werden können.

4.1 Bewertung von Erkennungsverfahren

Eine hohe Erkennungsrate ist wünschenswert, es werden jedoch weitere Ansprüche an Erkennungsverfahren gestellt. Sie sollten eine kleine falsch-positiv Erkennung haben. Eine falsch-positiv Erkennung bedeutet in diesem Fall, dass ein Computer zu unrecht beschuldigt wird ein Bot zu sein. Dies kann negative Folgen für einen unschuldigen Internetnutzer haben, beispielsweise können alle sein Mails zu unrecht als Spam verworfen werden.

Es ist wünschenswert, dass sie unabhängig von der Struktur und der Art der Kommunikation sind, um vielfältig einsetzbar zu sein. Das heißt auch, dass sie verschlüsselte Kommunikation erkennen sollen.

Methoden, die auf eine Netzwerkverkehrsanalyse setzen, sollten mit großen Datenmengen umgehen können und echtzeitfähig sein.

Die Erkennung von Bots sollte idealerweise geschehen bevor diese größeren Schaden anrichten können.

	Signatur-basiert	DNS-basiert	Mining-basiert	Honeynet
Unbekannte Botnetze	Nein	Ja	Ja	Ja
Protokoll und Struktur unabhängig	Nein	Nein	Ja	Ja
Verschlüsselte Kommunikation	Nein	Ja	Ja	Ja

Tabelle 1: Eine grobe Übersicht für welche Zwecke, welche Art von Erkennungsverfahren prinzipiell geeignet ist [20].

In Tabelle 1 wird eine grobe Übersicht gegeben für welche Botnetze welche Erkennungsverfahren eher geeignet sind und für welche nicht. Dies ist im allgemeinen schwierig zu sagen, aber es können Tendenzen aufgezeigt werden. So sind zum Beispiel signatur-basierte Ansätze schlecht geeignet um neue noch unbekannte Botnetze zu erkennen. Da die typische Signatur erst erarbeitet werden muss. Ebenso ist es bei DNS-basierten Verfahren, diese setzen voraus, dass DNS verwendet wird und sind somit nicht Protokoll und Struktur unabhängig.

4.2 Konkrete Erkennungsverfahren

DNS-basierte Erkennung

Botnetzbetreiber verwenden häufig dynamische DNS-Dienste (DDNS), dies bietet einen Ansatz Bots zu erkennen. Computer machen sich verdächtig, wenn sie häufig DDNS-Domains auflösen. DDNS-Dienste verwenden meist kurze TTL-Werte (Time-to-live), der Trend geht jedoch dazu, auch für normale DNS-Einträge kleine TTL-Werte zu wählen. Dies hat zur Folge, dass viele falsche Computer verdächtigt werden [27].

Ein anderer Ansatz nutzt DNS-basierte Black-hole Lists (DNSBL). DNSBL enthalten IP-Adressen von Spammern bzw. Computern, die Schadsoftware verteilen. Diese Listen können über DNS Abfragen genutzt werden z.B. folgendermaßen: <IP-Adresse>.spamhaus.org. Wird diese Anfrage korrekt aufgelöst, ist die IP-Adresse in der Datenbank aufgelistet [24]. In [21] wird beschrieben wie dies zur Erkennung genutzt werden kann. Falls Botnetzbetreiber bzw. die Bots selbst prüfen, ob sie in den Datenbanken gelistet sind, können sie entdeckt werden. Durch Überwachung dieser Abfragen können verdächtige Computer gefunden werden. Dies hat zudem den Vorteil, dass Bots meist vor Attacken ihren Status überprüfen und dadurch eine Erkennung vor der Attacke selbst möglich ist.

Wie in 3.2.1 beschrieben bietet ein eventuell eingesetzter Domainname-Generation-Algorithmus eine Möglichkeit Botnetze zu erkennen. So hat zum Beispiel Conficker.C 50000 Domains pro Tag generiert. Um zukünftige Domains zu sperren, muss der DGA des Bots durch Reverse Engineering untersucht werden. Ein Update der Bots kann die ganze Arbeit zunichte machen, falls ein neuer Algorithmus eingesetzt wird. Die Erkennung solcher Bots basiert darauf, dass die Domains sich signifikant von normalen Domains unterscheiden. Dabei können Methoden aus dem Bereich Data-Mining für die Gruppierung verwendet werden. Die alphanumerische Verteilung unterscheidet sich meist stark zwischen maschinell generierten Domains und normalen Domains. Die Bots erzeugen viele gleiche DNS-Abfragen und viele davon können nicht aufgelöst werden, da die Domains nicht registriert sind, dies macht einen Computer verdächtig [23].

Data-Mining-basierte Erkennung

BotHunter ist eine konkrete Anwendung zur Botnetzerkennung, das Verfahren welches eingesetzt wird beruht auf Korrelation [11]. Die Idee dahinter ist es, typische Muster für die verschiedenen Abschnitte des Lebenszyklus zu erkennen. In [29] wird auf BotHunter aufbauend ein Ansatz geliefert, der auch mit verschlüsselter Kommunikation zurecht kommt.

In [31] wird ein Data-Mining-basiertes Verfahren vorgestellt. Der Ansatz geht auch von Netzwerkverkehrsüberwachung eines Netzwerks aus. Dabei wird der Netzwerkverkehr in Zeitscheiben eingeteilt und die einzelnen Zeitscheiben unabhängig von den anderen analysiert. Der Inhalt der Pakete wird ignoriert, d.h. es kann auch verschlüsselte Kom-

munikation erkannt werden. Zur Analyse werden Algorithmen für maschinelles Lernen eingesetzt. In diesem Fall wird auf Entscheidungsbäume gesetzt. Es gibt zwei Phasen, einmal die Lernphase, dabei wird das System mit normalen Netzwerkverkehr und Netzwerkverkehr der Botnetze enthält, trainiert. Die zweite Phase bildet die Erkennungsphase, dabei werden die gelernten Muster gesucht. Das bedeutet der Entscheidungsbaum wird abgearbeitet, um am Ende zu entscheiden, ob ein Bot im Netzwerk ist.

Host-basierte Erkennung
In [2] wird ein kombinierter Ansatz aus Host- und Netzwerküberwachung beschrieben, um speziell Peer-to-Peer-Botnetze zu erkennen. Auf dem einzelnen Computer wird dabei das Dateisystem des Betriebssystems überwacht, um eine Einnistung in das System zu erkennen. Außerdem wird überwacht, ob sich ein Programm in die Autostartliste einträgt und der Netzwerkverkehr wird mitgeloggt.

Signaturbasierte Erkennung
In [4] wird ein konkretes signaturbasiertes System vorgestellt. Dabei wurde der ausgehende Netzwerkverkehr eines Netzwerks mit Hilfe einer Signaturdatenbank analysiert. Bekannte Botnetze konnten in einem Versuch in realen Netzwerken erkannt werden.

Ein weiterer signaturbasierter Ansatz wird in [16] beschrieben. Dabei werden einfache Regeln aufgestellt, die eine klare Syntax haben. Sie bestehen aus Header, Inhalt und Botnetzname. Der Inhalt besteht aus dem speziellen Verhalten des Botnetzes z.B. das Auflösen bestimmter URLs, das Scannen anderer Computer im Netz mittels TCP SYN Paketen oder das Herunterladen von Dateien mit bestimmten binärem Inhalt. In einem Experiment wurde der Ansatz mit bekannten Bots getestet. Diese Bots setzten auf verschiedene Architekturen z.B. Peer-to-Peer. Auch Botnetze mit zentralem Command-and-Control-Server, die über IRC oder HTTP kommunizieren, wurden getestet. Es konnten Erkennungsraten mit durchschnittlich über 80 % erreicht werden. Mit den gleichen Regeln konnten auch unbekannte Botnetze mit über 50 % erkannt werden, ohne unschuldige Computer zu melden, d.h. ohne falsch positive Verdächtigung.

5. SCHLUSS

Botnetze stellen eine immer größer werdende Bedrohung für das Internet dar, durch DDoS-Attacken, Spam-Mails oder Datendiebstahl. Deshalb ist es sehr wichtig sie zu erkennen und auszuschalten. Es gibt zahlreiche Ansätze diese Ziele zu erreichen und durch ihre Kombination werden immer bessere Erkennungsraten erreicht. Diese verschiedenen Ansätze zu vergleichen ist jedoch relativ schwierig, da sie mit unterschiedlichen Voraussetzungen getestet wurden. Zum Teil wurden reale Netzwerke überwacht zum Beispiel in Zusammenarbeit mit Internet-Service-Providern oder Universitätsnetzwerken. Es wurden jedoch auch im Labor bekannte Botnetze aufgesetzt und dabei die Erkennungsrate ermittelt. Das Feld Botnetzerkennung ist und bleibt auch weiterhin ein spannendes Forschungsfeld, da sich die Botnetzbetreiber den neuen Möglichkeiten anpassen und dadurch versuchen die Erkennung ihrer Botnetze zu verhindern.

Zukünftige Geldeinnahmequellen für Botnetzbetreiber könnten Bitcoins sein. Die virtuelle Währung wird durch aufwenden von Rechenleistung erzeugt. Hier könnten Botnetzbetreiber anonym die Rechenleistung ihrer Bots zu Geld machen. Für die Zukunft könnten mobile Smartphones zu beliebten Angriffszielen werden. Die Rechenleistung steigt immer weiter und häufig läuft veraltete Software auf den Geräten. Zusätzlich sind viele Smartphones ständig mit dem Internet verbunden und im Betrieb. Dadurch kann ein Botnetzbetreiber zu jeder Zeit auf seine Bots zugreifen.

6. LITERATUR

[1] R. S. Abdullah, M. F. Abdollah, M. Noh, Z. Azri, M. Z. Mas' ud, S. R. Selamat, and R. Yusof. Revealing the criterion on botnet detection technique. *International Journal of Computer Science Issues (IJCSI)*, 10(2), 2013.

[2] R. S. Abdullah, M. F. Abdollah, Z. A. M. Noh, M. Z. Mas' ud, S. Sahib, and R. Yusof. Preliminary study of host and network-based analysis on p2p botnet detection. In *Technology, Informatics, Management, Engineering, and Environment (TIME-E), 2013 International Conference on*, pages 105–109. IEEE, 2013.

[3] E. Alparslan, A. Karahoca, and D. Karahoca. Botnet detection: Enhancing analysis by using data mining techniques, 2012.

[4] S. Behal, A. S. Brar, and K. Kumar. Signature-based botnet detection and prevention. *http://www.rimtengg. com/iscet/proceedings/pdfs/advcom p/148. pdf*, 2010.

[5] D. Bhatt, N. Garg, and R. Rawat. A survey on botnet detection. In *Proceedings of National Conference on Trends in Signal Processing & Communication (TSPC '13)*, volume 12, page 14th, 2013.

[6] N. Davis. Botnet detection using correlated anomalies. *Technical University of Denmark Informatics and Mathematical Modelling*, 2012.

[7] D. Dittrich. So you want to take over a botnet. In *Proceedings of the 5th USENIX conference on Large-Scale Exploits and Emergent Threats*, pages 6–6. USENIX Association, 2012.

[8] M. Eslahi, R. Salleh, and N. B. Anuar. Bots and botnets: An overview of characteristics, detection and challenges. In *Control System, Computing and Engineering (ICCSCE), 2012 IEEE International Conference on*, pages 349–354. IEEE, 2012.

[9] M. A. R. J. Z. Fabian and M. A. Terzis. My botnet is bigger than yours (maybe, better than yours): why size estimates remain challenging. In *Proceedings of the 1st USENIX Workshop on Hot Topics in Understanding Botnets, Cambridge, USA*, 2007.

[10] M. Feily, A. Shahrestani, and S. Ramadass. A survey of botnet and botnet detection. In *Emerging Security Information, Systems and Technologies, 2009. SECURWARE'09. Third International Conference on*, pages 268–273. IEEE, 2009.

[11] G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee. Bothunter: Detecting malware infection through ids-driven dialog correlation. In *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, page 12. USENIX Association, 2007.

[12] N. Hachem, Y. Ben Mustapha, G. G. Granadillo, and

H. Debar. Botnets: lifecycle and taxonomy. In *Network and Information Systems Security (SAR-SSI), 2011 Conference on*, pages 1–8. IEEE, 2011.

[13] honeynet Project. Know your enemy: Honeynets. `http://old.honeynet.org/papers/honeynet/`.

[14] O.-R. Jeong, C. Kim, W. Kim, and J. So. Botnets: threats and responses. *International Journal of Web Information Systems*, 7(1):6–17, 2011.

[15] J. P. John, A. Moshchuk, S. D. Gribble, and A. Krishnamurthy. Studying spamming botnets using botlab. In *NSDI*, volume 9, pages 291–306, 2009.

[16] W. M. Li, S. L. Xie, J. Luo, and X. D. Zhu. A detection method for botnet based on behavior features. *Advanced Materials Research*, 765:1512–1517, 2013.

[17] K. Muthumanickam and E. Ilavarasan. P2p botnet detection: Combined host-and network-level analysis. In *Computing Communication & Networking Technologies (ICCCNT), 2012 Third International Conference on*, pages 1–5. IEEE, 2012.

[18] D. Plohmann, E. Gerhards-Padilla, and F. Leder. Botnets: Detection, measurement, disinfection & defence. *European Network and Information Security Agency (ENISA)*, 2011.

[19] N. Raghava, D. Sahgal, and S. Chandna. Classification of botnet detection based on botnet architechture. In *Communication Systems and Network Technologies (CSNT), 2012 International Conference on*, pages 569–572. IEEE, 2012.

[20] N. Raghava, D. Sahgal, and S. Chandna. Classification of botnet detection based on botnet architechture. In *Communication Systems and Network Technologies (CSNT), 2012 International Conference on*, pages 569–572. IEEE, 2012.

[21] A. Ramachandran, N. Feamster, and D. Dagon. Revealing botnet membership using dnsbl counter-intelligence. *Proc. 2nd USENIX Steps to Reducing Unwanted Traffic on the Internet*, pages 49–54, 2006.

[22] R. A. Rodríguez-Gómez, G. Maciá-Fernández, and P. García-Teodoro. Survey and taxonomy of botnet research through life-cycle. *ACM Computing Surveys (CSUR)*, 45(4):45, 2013.

[23] R. Sharifnya and M. Abadi. A novel reputation system to detect dga-based botnets. In *Computer and Knowledge Engineering (ICCKE), 2013 3th International eConference on*, pages 417–423. IEEE, 2013.

[24] T. Sochor and R. Farana. Improving efficiency of e-mail communication via spam elimination using blacklisting. In *Telecommunications Forum (TELFOR), 2013 21st*, pages 924–927. IEEE, 2013.

[25] S. Stover, D. Dittrich, J. Hernandez, and S. Dietrich. Analysis of the storm and nugache trojans: P2p is here. *USENIX; login*, 32(6):18–27, 2007.

[26] W. T. Strayer, D. Lapsely, R. Walsh, and C. Livadas. Botnet detection based on network behavior. In *Botnet Detection*, pages 1–24. Springer, 2008.

[27] R. Villamarín-Salomón and J. C. Brustoloni. Identifying botnets using anomaly detection techniques applied to dns traffic. In *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE*, pages 476–481. IEEE, 2008.

[28] K. Wang, C.-Y. Huang, L.-Y. Tsai, and Y.-D. Lin. Behavior-based botnet detection in parallel. *Security and Communication Networks*, 2013.

[29] H. Zhang, C. Papadopoulos, and D. Massey. Detecting encrypted botnet traffic. In *Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on*, pages 163–168. IEEE, 2013.

[30] Z. Zhang, X. Cui, and C. Liu. Web2bot: Botnet in web 2.0 era.

[31] D. Zhao, I. Traore, B. Sayed, W. Lu, S. Saad, A. Ghorbani, and D. Garant. Botnet detection based on traffic behavior analysis and flow intervals. *Computers & Security*, 39:2–16, 2013.