Einleitung

Grund der Studienarbeit

Die Digitalisierung ist aktuell ein Faktor, welcher viele Prozesse in der Welt vorantreibt. Die Menschen werden vernetzter, Unternehmen können ihre Daten besser und effizienter Speichern und wir entwickeln immer bessere Kommunikationsmöglichkeiten. Aber gerade mit dieser stetigen Verbesserung der Technik erfolgt auch eine immer größere Abhängigkeit von den Computern, welche uns helfen, die Produktivität voranzutreiben. Cyber Angriffe nehmen auf der ganzen Welt rasant zu und verursachen Schäden in Milliardenhöhe. Laut einer Umfrage der Online Plattform Statista gaben 47% der befragten deutschen Unternehmen an, im Jahr 2022 bereits Opfer eines Cyber Angriffs geworden zu sein.[[1]](#footnote-1)Hinter vielen dieser Angriffe befinden sich Botnetze, also Netzwerke aus kompromittierten PCs, welche, ohne das Wissen des Benutzer dafür genutzt werden, verschiedene bösartige Aktionen auszuführen.

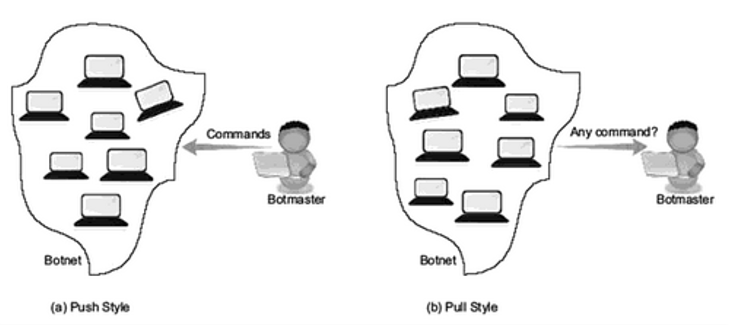
Ziel dieser Arbeit

Das Ziel dieser Arbeit soll es sein, dem Leser einen Einblick in die Funktionen und Arbeitsweisen eines Botnetzes zu geben, und ihn darüber aufklären, wie sich diese Verbreiten und Wachsen können. Es sollen zu Beginn verschiedene Grundlagen erklärt und definiert werden, welche für das Weitere Verständnis wichtig sind. Des Weiteren soll auf die Funktion der Botnetzen sowie die daraus resutierenden Gefahren eingegangen werden. Mit Hilfe des Open Source Projektes BYOB (Build your Own Botnet) sollen praxisorientiert und anhand von Codebeispielen die Einzelnen Komponenten des Systems erklärt werden, und die Funktionsweise genauer erläutert werden, indem eine virtuell aufgesetzte Testumgebung kompromittiert werden soll. Im Abschließenden Kapitel soll schließlich erklärt werden, wie man Netzwerkgeräte entsprechend sichert, um somit ausreichend Schutz garantieren zu können. Des Weiteren wird auch ein Ausblick gegeben, wie sich Botnetze in der Zukunft dank modernerer Technik weiterentwickeln könnten.

Lebenszyklus eines Botnetz  
Der Lebenszyklus eines Botnets besteht im Grunde genommen aus 3 verschiedenen Hauptphasen, welche im folgenden Abschnitt genauer erläutert werden sollen.

Phase 1 – Rekrutierungsphase:   
Am Anfang beginnt die Formierung des Botnetz damit, dass es versucht, so viele anfällige Maschinen wie möglich zu befallen, damit diese als entsprechende Bots für das Botnetz eingesetzt werden können. Hierfür werden die anfälligen Maschinen durch verschiedene Mechanismen mit dem Bot Code befallen. Einer der meistgenutzten Mechanismen bedient sich hierbei der Propagierungsmethoden klassischer Computerwürmer. Diese erfordern nicht einmal eine Interaktion mit dem Benutzer, da sich Würmer selbstständig über das lokale Netz und das Internet verbreiten, indem sie aktive Scans nach bekannten Schwachstellen durchführen und somit anfällige Maschinen finden können. Es gibt nun einige Mechanismen, diese anfälligen Maschinen für das Botnetz zu rekrutieren, diese erfordern jedoch einen gewissen Grad an Benutzerinteraktion. Das wohl mächtigste Werkzeug ist das social Engineering, welches darauf abzielt, menschliche Schwachstellen auszunutzen, um sein Ziel zu erreichen. Hierfür werden weitreichende Phishing Campagnen über Email und Social Netzworks erstellt, welche die Benutzer davon überzeugen sollen, auf bösartige Links zu klicken, welche im Anschluss eine sogenannte Bot Binary herunterladen. In anderen Fällen wird der User dazu verführt, Webseiten zu besuchen welche active Inhalte wie JavaScript oder ActiveX controls beinhalten, was dazu führt, dass automatisch Malware von diesen Seiten heruntergeladen und installiert wird, ohne dass der Benutzer etwas davon mitkriegt. Ein klassischer Weg ist es auch, die Botnetz Binaries über ein physikalisches Medium, wie beispielsweise einen USB Flash Speicher zu verbreiten. Das Problem ist hierbei, dass die Malware in Form eines Executable manuell auf dem Zielrechner installiert werden muss. Diese Methode wird eingesetzt, wenn sich der Zielrechner beispielsweise hinter einem NAT verbirgt und nicht direkt aus dem Internet angesprochen werden kann. [[2]](#footnote-2)

Phase 2- Command & Control Phase   
Die 2. Phase widmet sich nun der Kontrolle der infizierten Maschinen. Der sogenannte Botmaster kommuniziert mit diesen über einen C&C Channel. Es gibt unterschiedliche Arten der Kommunikation zwischen Botmaster und Bots. So gibt es zum einen den Push Style, aber auch den Pull Style. Wird im Push Style kommuniziert, werden die Commandos direkt an den Bot gesendet und dieser reagiert auf diese Anforderungen. Im Pull Style senden die Bots in regelmäßigen Intervallen Nachrichten an den Botmaster, und erbitten neue Commandos. Verdeutlicht wird dies in der Nachfolgenden Abbildung:[[3]](#footnote-3)



Phase 3 – Botnetz Aktivitätsphase.  
Unter Aktivität versteht sich eine Reihe von Aktionen und Angriffen wie etwa Scanning oder DDOS, welche von den Bots als Antwort auf ein Kommando ausgeführt werden. Es ist wichtig, die Bandbreite eines kompromittierten Host zu kennen, damit man weiß, wie viel Kapazitäten beispielsweise während einem DDOS Angriff zur Verfügung stehen. Die Bandbreite der Bots kann ermittelt werden, indem sie angewiesen werden, Daten an viele ausgewählte Testserver zu senden und diese Sendeleistung analysiert wird.[[4]](#footnote-4)

Botnetz Architektur  
Es gibt kein einheitliches Format, welchem ein Botnetz folgt. Jedes Netz wird individuell aufgebaut und folgt dabei verschiedenen Mustern und Architekturen. Um ein größeres Verständnis zu diesem Thema zu erlangen, wird im Folgenden Kapitel genauer auf die Architektur und Form der Botnetze eingegangen.

x.1 Zentralisierte Botnetze  
x.1.1 Stern Netz  
Bei einer Stern Topologie bekommt jeder Bot-Rechner seine Arbeitsanweisungen direkt vom C&C Server zugewiesen, da bei dieser Architektur die direkte Kommunikation zwischen dem Server und den Bots im Fokus steht. Diese Arbeitsweise erhöht den Durchsatz des Netz, da Aufträge ohne Latenz abgearbeitet werden können und vermieden wird, Daten unnötig über dritte Rechner zu leiten. Dieser vermeintliche Vorteil birgt aber auch gleichzeitig die größte Schwäche dieser Architektur. Da der Server manuell auf jedem Rechner hinterlegt ist, lässt sich dieser leicht entdecken und Übernehmen. Der Server bietet einen Single Point of Failure. Auch bei anderweitigem Ausfall des Servers zerbricht das gesamte Netzwerk.[[5]](#footnote-5)

Ein Bild, das Uhr enthält.

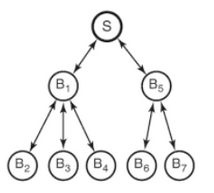
Automatisch generierte Beschreibung

x.1.2 Multi Server Netz  
Diese Architektur stellt die logische Erweiterung der Botnetz Stern Topologie dar. Das ein-Server-Modell wurde hierbei um mehrere redundante Server erweitert, es übernehmen mehrere C&C Server die Anweisungsverteilung an die Bots. Jedem Bot wird nun eine Liste der verfügbaren C&C Server mitgegeben und Unerreichbarkeit oder Ausfall einzelner Server spielen nun keine Rolle mehr. Ein dementsprechend größerer Planungsaufbau geht mit diesem Netz somit aber auch einher. Dies ist aber in Kauf zu nehmen, da sich somit der Single Point of failure vermeiden lässt und die Hauptschwachstelle der Sterntopologie somit neutralisiert wird.[[6]](#footnote-6)

Ein Bild, das Schere, Uhr, Halskettchen enthält.

Automatisch generierte Beschreibung

x.1.3 Hierarchische Netzwerke  
Eine weitere Art zentralisierter Botnetze stellen die hierarchischen Netze dar. Sie erweitern das Multiserver Netz, indem dem Netz eine weitere, aus Proxyservern bestehende Ebene, hinzugefügt wird. Die Proxys dienen der Sicherheit des Netzes, da sie den Ursprung der Command & Control Server verschleiern und somit deren Entdeckung oder Übernahme erheblich erschweren. Jedoch wird der Aufbau und die Konfiguration solcher Netze schwerer, als die bisherigen Strukturen, da dieses Netz durch die Proxyebene eine erhöhte Komplexität aufweist. Auch bremsen die Proxys den Traffic und erhöhen somit die Latenz des Netzwerks.[[7]](#footnote-7)

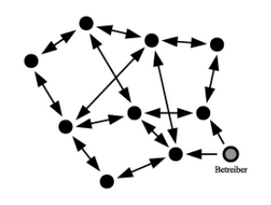


x.2 Dezentralisierte Botnetze  
Dem Aufbau und der Logik zentralisierter Botnetze stehen nun die sog. Dezentralisierten Botnetze gegenüber. Zentralisierte Botnetze haben in ihrem Aufbau ein großes Problem: Den Single Point of failure. Alles wird zentral über einen Server geregelt. Fällt dieser aus, bricht das komplette Netz zusammen. Um dieser Sicherheitslücke entgegenzuwirken wurde das Konzept der zentralen Botnetze überarbeitet. Es entstanden die dezentralisierten Botnetze, welche sich durch ein wesentlich komplexeres Design und die Vermeidung des Single Point of failure auszeichnen.

x.2.1 Fast Flux Botnetze  
Dieses Netzdesign ist auf der Round Robin Lastenverteilung via DNS aufgebaut. Hierbei existiert eine Control Domain, welche als Nameserver verschiedene Bot IP Adressen eingetragen hat. Diese agieren alle als Proxys. Um nun einen Auftrag zu bekommen, stellt ein Bot eine Anfrage an diese Control Domain. Anschließend wird eine der konstant Rotierenden IP-Adressen der Proxys zurückgegeben, an welchen nun die Anfrage geleitet wird. Dieser Proxyrechner leitet im nächsten Schritt die Anfrage an den Bot Master weiter. Der Bot Master gibt den Auftrag an den Proxy, welcher es zum fragenden Bot weiterleitet. Durch die Rotation liefert eine weitere DNS-Anfrage im Anschluss eine andere IP Adresse, da es sehr viele Proxys in der Control Domain gibt, unter welchen die Anfragen aufgeteilt werden. Zur Verschleierung des Bot Master werden durch periodische Updates der A Records im DNS-Server geändert. Wenn ein Proxy längere Zeit nicht erreichbar ist, muss sein seine IP-Adresse beim nächsten Update aus dem DNS entfernt werden.[[8]](#footnote-8)

Ein Bild, das Text, Uhr, ClipArt enthält.

Automatisch generierte Beschreibung

x.2.2 Peer-to-Peer Botnetze  
Peer to Peer stellt die bisher vorgestellten Botnetz Architekturen auf den Kopf. Bei diesem Aufbau findet die Kommunikation nicht zwischen Clients und dedizierten Servern statt. Stattdessen ist die einzelnen Netzwerkknoten gleichberechtigt. Somit sind die Clients nicht machtlos, sollte ihnen der zentrale Server eine Anfrage verweigern, oder falls der Dienst ausfällt. Es gibt nämlich keinen zentralen Server mehr. Alle Knoten übernehmen sowohl Server als auch Clientaufgaben. Sie sind ebenbürtig. Durch das Wegfallen einer zentralen Komponente wird das Überwachen als auch das Zerstören eines Peer to Peer Botnetz erheblich erschwert. Fallen einzelne Komponenten aus, können sie Problemlos ersetzt werden bzw. ein anderer Knoten angesteuert werden.[[9]](#footnote-9)  


1. <https://de.statista.com/statistik/daten/studie/1230157/umfrage/unternehmen-die-in-den-letzten-12-monaten-eine-cyber-attacke-erlebt-haben/> (15.10.2022) [↑](#footnote-ref-1)
2. Botnets: Architectures, Countermeasures, and Challenges(CRS Series in Security, Privacy and Trust)  
   Georgios Kambourakis, Marios Anagnostopoulus, Weihzi Meng, Peng Zhou 8. Oktober 2019 **S3-4** [↑](#footnote-ref-2)
3. Botnets: Architectures, Countermeasures, and Challenges(CRS Series in Security, Privacy and Trust)  
   Georgios Kambourakis, Marios Anagnostopoulus, Weihzi Meng, Peng Zhou 8. Oktober 2019 **S4** [↑](#footnote-ref-3)
4. Botnets: Architectures, Countermeasures, and Challenges(CRS Series in Security, Privacy and Trust)  
   Georgios Kambourakis, Marios Anagnostopoulus, Weihzi Meng, Peng Zhou 8. Oktober 2019 **S4-5** [↑](#footnote-ref-4)
5. Botnetze: Aufbau, Funktion & Anwendung Matthis C. Laass Fachhochschule Aachen S2 [↑](#footnote-ref-5)
6. Botnetze: Aufbau, Funktion & Anwendung Matthis C. Laass Fachhochschule Aachen S3 [↑](#footnote-ref-6)
7. Botnetze: Aufbau, Funktion & Anwendung Matthis C. Laass Fachhochschule Aachen S3-4 [↑](#footnote-ref-7)
8. Botnetze: Aufbau, Funktion & Anwendung Matthis C. Laass Fachhochschule Aachen S6-7 [↑](#footnote-ref-8)
9. Botnetze: Aufbau, Funktion & Anwendung Matthis C. Laass Fachhochschule Aachen S7-8 [↑](#footnote-ref-9)