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.

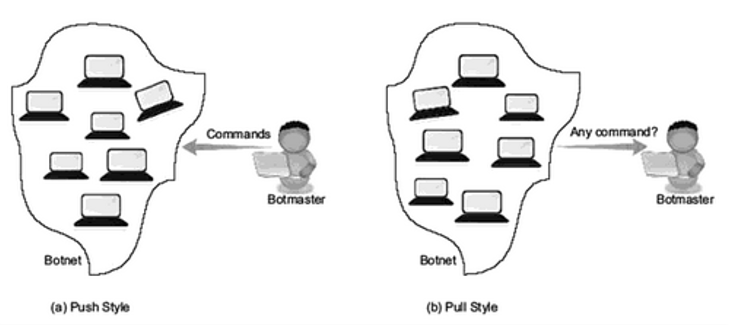
Ziel dieser Arbeit

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 resultierenden Gefahren eingegangen werden. Anhand von praxisorientierten Beispielen, wie der Code Analyse eines realen Botnetz, sollen die Funktionsweisen eines Botnetz noch genauer veranschaulicht werden. 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 Botnetz beschreibt die verschiedenen Phasen, welche das Botnetz bei der Übernahme eines Computers durchläuft. Im Folgenden Abschnitt sollen diese Phasen nun genauer erläutert werden.

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 Kampagnen über Email und soziale Netzwerke 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 aktive 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)

Phase 4 – Auflösungsphase  
In dieser Phase wird der Bot von infizierten System entfernt. Er „stirbt“ sozusagen. Dies kann mehrere Ursachen haben. Entweder wurde der Bot entdeckt und entfernt, oder das Betriebssystem des Hostrechner wurde neu installiert. Teilweise kann aber auch der Botmaster aus Sicherheitsgründen die Schadsoftware vom Computer löschen, um so nicht erkannt zu werden und keine Spuren für Forensiker und Ermittler zu hinterlassen. Dies ist die letzte Phase im Lebenszyklus.[[5]](#footnote-5)

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.[[6]](#footnote-6)

Ein Bild, das Uhr enthält.

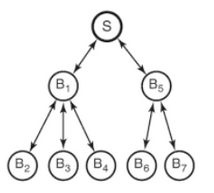
Automatisch generierte Beschreibung

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.[[7]](#footnote-7)

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.[[8]](#footnote-8)

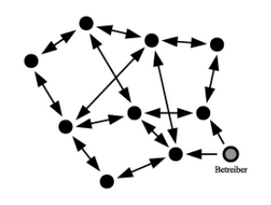


x.2 Dezentralisierte Botnetze  
Dem Aufbau und der Logik zentralisierter Botnetze stehen nun die sog. Dezentralisierten Botnetze gegenüber. Zentralisierte Botnetze haben in ihrem Aufbau ein großes Problem: Den Single Point of Failure. Alles wird zentral über einen Server geregelt. Fällt dieser aus, bricht 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.[[9]](#footnote-9)

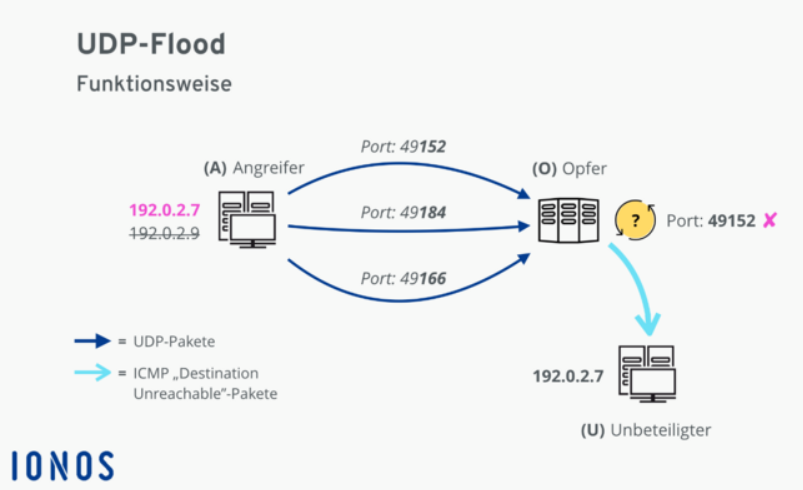
Ein Bild, das Text, Uhr, ClipArt enthält.

Automatisch generierte Beschreibung

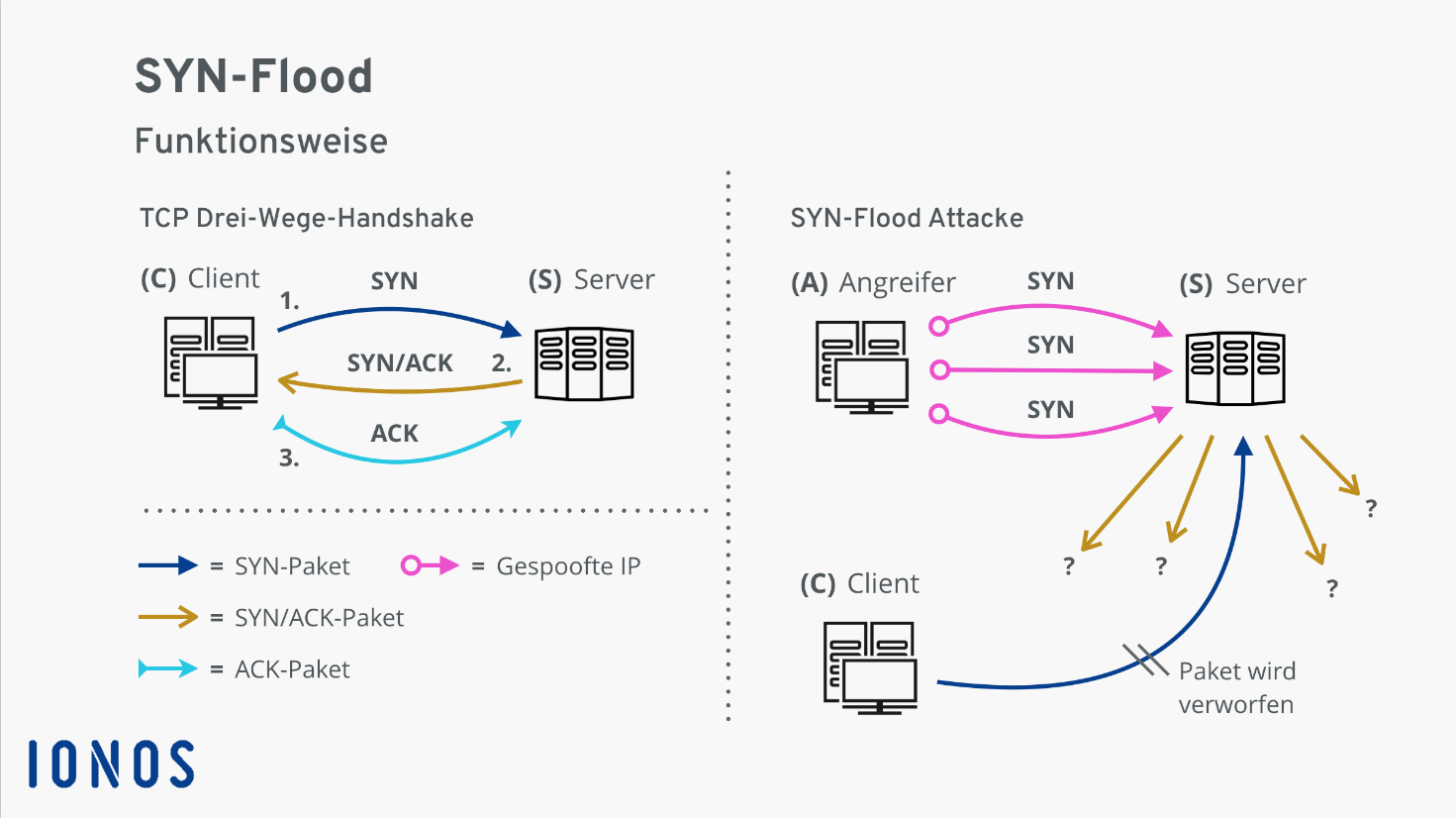
x.2.2 Peer-to-Peer Botnetze  
Peer-to-Peer stellt die bisher vorgestellten Botnetz Architekturen auf den Kopf. Bei diesem Aufbau findet die Kommunikation nicht zwischen Clients und dedizierten Servern statt. Stattdessen sind die einzelnen Netzwerkknoten gleichberechtigt. Somit sind die Clients nicht 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.[[10]](#footnote-10)  


Botnetz Angriffe:  
Hinter Botnetzen verbergen sich meist professionell organisierte kriminelle Netzwerke. Haben diese es geschafft, erfolgreich ein Botnetz aufzusetzen, so ist ihre Intention in den meisten Fällen, mit diesem einen Angriff oder andere bösartige Dinge auszuführen. In diesem Kapitel soll auf mögliche Angriffsszenarien und Attacken in Verbindung mit Botnetzen eingegangen werden.

DDOS (Distributed Denial Of Service)  
Eine DDOS-Attacke zielt darauf ab, das Zielsystem außer Betrieb zu setzen. Ein oder mehrere ausgewählte Server werden solange gezielt mit Anfragen von den verschiedenen Systemen bombardiert, bis dieser seine eigentlichen Aufgaben nicht mehr erledigen kann. Im schlimmsten Fall kann das komplette Zielsystem unter dieser Last zusammenbrechen. Gerade bei DDOS-Attacken ist es äußerst schwierig, den Angreifer zu ermitteln, da die Anfragen oft von mehreren tausenden Systemen gleichzeitig gesendet werden. DDOS-Attacken lassen sich in folgende Formen einteilen:[[11]](#footnote-11)

UDP Flood  
Bei einer UDP Flood senden ein oder mehrere Angreifer in einem sehr kurzen zeitlichen Intervall manipulierte Datenpakete an das Zielsystem. Diese sollen das Ziel derart überlasten, sodass legitime Angriffe nicht mehr beantwortet werden können und der eigentliche Service zum Erliegen kommt. Die Funktionsweise der UDP Flood Attacke baut dabei auf den Besonderheiten des User Datagram Protokoll (UDO) auf. Wenn auf einem Server ein UDP-Paket eingeht, prüft das Betriebssystem den entsprechenden Port auf lauschende Applikationen. Wenn keine Anwendungen gefunden werden, muss der Server den Absender des Pakets darüber informieren. Dafür sendet er ein „Destination Unreachable“ ICMP Paket an den Absender. [[12]](#footnote-12)  


Quelle: <https://www.ionos.de/digitalguide/fileadmin/DigitalGuide/Schaubilder/udp-flood.png>

SYN Flood   
SYN Flood ist eine sog. Protokoll-Attacke. Es zielt darauf ab, Schwachstellen des TCP-Protokolls zu missbrauchen, um sein Ziel überlasten zu können, sodass dieser keine legitimen Anfragen mehr beantworten kann. SYN Flood missbraucht den Mechanismus des drei Wege Handschake des Transmission Control Protokoll (TCP). Da es sich bei TCP um ein Verbindungsorientiertes Protokoll handelt, müssen Client und Server zunächst eine Verbindung aushandeln. Dieser Handshake läuft folgendermaßen ab:   
1. Der Client sendet ein SYN-Paket(„synchonise“) an den Server  
2. Der Server antwortet mit einem SYN/ACK(„Acknowledge“) und legt eine Datenstruktur, Transmission Protocol Block(TCB) genannt, für die Verbindung im SYN Backlog an  
3. Der Client beantwortet dieses SYN/ACK-Packet mit einem ACK-Paket. Damit wird der Handshake abgeschlossen.  
Bei dem SYN Flood Angriff senden die Bots SYN-Pakete über eine gespoofte IP-Adresse an den Server. Dieser legt eine TCB Datenstrucktur für die halboffene Verbindung im Backlog an und belegt somit Speicher auf dem Server. Der Server sendet nun ein SYN/ACK-Paket zurück. Vom Angreifer bekommt der Server aber keine Bestätigung, da es sich um eine gefälschte IP-Adresse handelt, und sendet weitere Pakete, während er die Verbindung halboffen hält. Der Server wartet weiter auf eine Antwort, währenddessen gehen aber bereits weitere SYN-Pakete des Botnetzes ein, welche ebenfalls in das Backlog eingetragen werden müssen. Irgendwann hat der Server keinen Platz im Backlog mehr und verwirft neue, eingehende SYN-Pakete. Damit ist er von außen nicht mehr erreichbar.[[13]](#footnote-13)  


Quelle: https://www.ionos.de/digitalguide/fileadmin/DigitalGuide/Schaubilder/syn-flood-funktionsweise.png

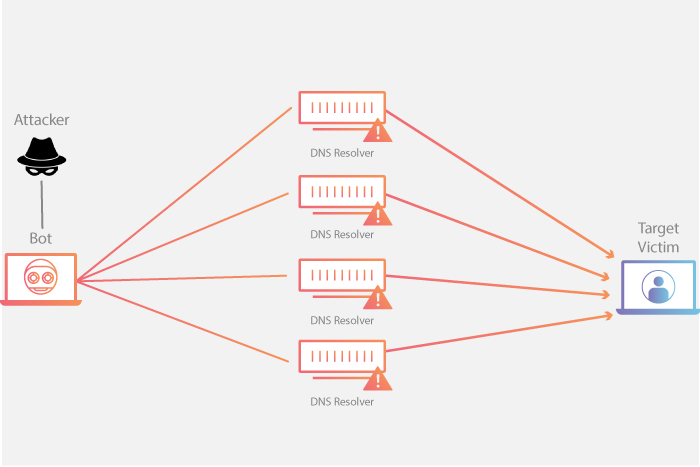
HTTP Flood  
Bei einem HTTP-Flood-Angriff senden Angreifer HTTP Anfragen an einen Webserver, welche gezielt Seiten mit großem Ladevolumen aufrufen. Durch die riesige Anfragenkapazität, welche Botnetze erreichen können, wird der Webserver überlastet und kann keine legitimen Anfragen mehr verarbeiten, wodurch der Webserver oder die Web-Applikation nicht mehr erreichbar ist. Diese Angriffsweise gehört zu der häufigsten Form der DDOS-Angriffe. Es wurden schon hTTP-Flood-Angriffe beobachtet, bei welchen die Zahl der böswilligen Anfragen bis in den mittleren dreistelligen Millionen Bereich ging.[[14]](#footnote-14)

Ein Bild, das Text, Uhr enthält.

Automatisch generierte Beschreibung  
Quelle: <https://www.myrasecurity.com/app/uploads/2021/09/Http_Flood_Attack_de.png>

DNS Amplification Attack  
Diese Art des DDOS Angriffs ist ein volumetrischer, verteilter Denial-of-Service Angriff, welcher die Funktionalität von offenen DNS-Resolvern benutzt, um so einen Zielserver mit einer verstärkten Traffic Menge zu überfluten, was in einem Ausfall der Funktionsfähigkeit des Servers und im Schlimmsten Falle in einem Ausfall des Systems resultiert. Die IP-Adresse der Bots wird bei diesem Angriff gespooft, sodass sie für den DNS Resolver wie die IP Adresse des Opfers aussieht. Anschließend stellen die Bots mit gespoofter IP-Adresse Anfragen an den DNS-Resolver. Diese Anfragen sind so strukturiert, dass die Antwort des DNS-Resolvers so groß wie möglich ausfällt.  
Der Angriff kann in vier Schritt unterteilt werden:

1. Über einen kompromittierten Endpunkt sendet der Angreifer UDP-Pakete mit gespoofter IP-Adresse an den DNS-Resolver.  
2. Die Anfragen an den DNS-Resolver sind darauf ausgelegt, eine möglichst große Antwort zu erzielen. Oft werden dabei Argumente wie „ANY“ benutzt, um die sehr große Antwortpakete zu erhalten.  
3. Nachdem er die Anfrage erhalten hat, sendet der DNS-Resolver eine große Antwort an die gespoofte IP-Adresse aka. Die IP-Adresse des Opfers  
4. Das Opfer empfängt die Antwort und seine Netzwerkinfrastruktur wird mit Traffic überschwemmt was einen Denial-Of-Serivice verursacht und das Netzwerk nicht mehr erreichbar ist bzw. komplett ausfällt.[[15]](#footnote-15)

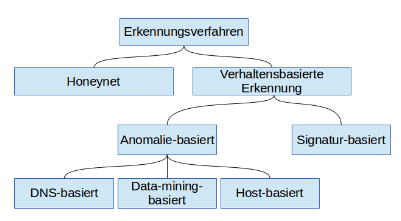


Quelle: <https://cf-assets.www.cloudflare.com/slt3lc6tev37/2JmKP07Mi6jYbACILN84VI/9a91d91ecc1f414aa89ae001dbfce393/Learning_Center_DDoS_Diagrams_clean.png>

Spam  
Auch können die Bots in einem Botnetz benutzt werden, um Spam Nachrichten zu verschicken. Sie können sich gefälschte Konten in Foren oder Social Media Pattformen anlegen und hier automatisiert verschiedene Nachrichten wie z.B. Phishing Mails versenden. Da für die Erstellung eines Benutzerkontos nur sehr wenige Felder ausgefüllt werden müssen (Name, E-Mail-Adresse etc.) können die Angreifer die Bots sehr leicht dazu Programmieren, diese Formulare automatisch auszufüllen. Die Bots werden zudem genutzt, um Webseiten zu scannen und diese nach gültigen E-Mail-Adressen zu durchsuchen. Diese validen E-Mail-Adressen werden in eine Datenbank eingetragen, welche das Botnetz benutzt, um Mails an seine Opfer zu senden.[[16]](#footnote-16)

Senden von Falschnachrichten  
Gerade in der Heutigen Zeit, in welcher wir unsere Meinung vermehrt über Social Media kundgeben, stellen Botnetze, welche gezielt auf diesen Plattformen agieren, eine immer größere Gefahr dar. Die Botnetze werden dazu genutzt, falsche Meinungen und Bewertungen abzugeben, und so beispielsweise politische Kampagnen zu untergraben oder den Ruf von verschiedenen Firmen zu beschädigen. Eine Studie der University of California hat gezeigt, das 15% alle heute aktiven Twitter Konten von Bots gesteuert werden.[[17]](#footnote-17) Durch diese gewaltige Zahl wird die Macht klar, welche die Betreiber der Botnetze haben, Meinungen online zu beeinflussen. Je weiter die Technologie voranschreitet, desto schwieriger wird es, die Bots von legitimen Usern zu unterscheiden. Deshalb ist es umso wichtiger, zuverlässigere und bessere Methoden zu entwickeln, um gegen die Botnetze vorzugehen.

Botnetz Erkennungsmethoden & Gegenmaßnahmen  
Dieses Kapitel soll sich mit der Erkennung von Botnetzen und deren Aktivitäten anhand verschiedener Indizien beschäftigen, sowie verschiedenen Gegenmaßnahmen, sollte man Indizien gefunden haben, Teil eines solchen zu sein. Je nach Architektur und Kommunikationsinfrastruktur gibt es verschiedene Ansätze und Methoden zur Erkennung. Im Allgemeinen lassen sich die verschiedenen Methoden in aktive und passive Verfahren unterteilen. Passive Verfahren sind im Allgemeinen zu bevorzugen, da aktive Verfahren dem Botmaster die Möglichkeit geben zu erfahren, dass sein Botnetz untersucht wird, wodurch er seine Konfiguration ändern und sich besser absichern kann: Die passiven Erkennungsmethoden lassen sich dagegen nicht vom Botmaster erkennen.

  
Quelle: https://www.net.in.tum.de/fileadmin/TUM/NET/NET-2014-08-1/NET-2014-08-1\_03.pdf

Honeynet  
Ein Honeynet ist ein Netzwerk, welches mehrere Honeypots enthält. Honeypots sind Computer oder Geräte, welche selbst keine produktiven Funktionen erfüllen. Ihr Zweck ist es, als Falle in Netzwerke eingebaut zu werden, um Angreifer anzulocken. Dafür greifen sie oft auf veraltete Software zurück, welche viele Sicherheitslücken besitzt, um sich für den Angreifer attraktiver zu machen. Da der Honeypot keine Funktion erfüllt, kann jeder Netzwerkverkehr in Verbindung mit dem Honeypot auf einen möglichen Angriff hindeuten. Eingehender Netzwerkverkehr könnte somit beispielsweise ein aktiver Port Scan sein. Sollte Netzwerkverkehr vom Honeypot ausgehen, so muss dies bedeuten, dass er von Schadsoftware befallen wurde, da er selbstständig keinen Netzwerkverkehr erzeugt. Aus diesen Informationen lassen sich Schüsse auf das Botnetz und wie es kommuniziert ziehen und es können beispielsweise mögliche Schwachstellen im Botnetz entdeckt werden. Des Weiteren können somit die Werkzeuge des Angreifers analysiert werden. [net.in.tum]

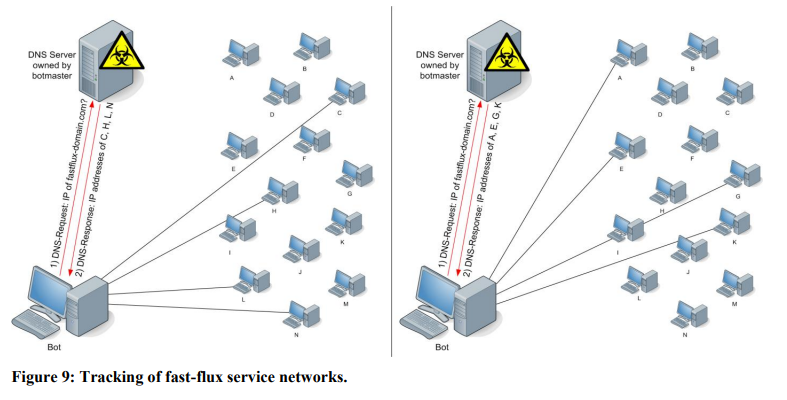
Verhaltensbasierte Erkennungsmethode  
Honeynets können nur bis zu einem gewissen Grad bei der Erkennung von Botnetzen helfen. Sollte der Computer zwar befallen worden sein, jedoch nicht versuchen andere Rechner zu infizieren, so bleibt er für das Honeynet unentdeckt, da somit kein Analysierbarer Traffic von ihm ausgeht. Hierfür kommen die Ansätze der verhaltensbasierten Erkennungsmethoden ins Spiel, welche versuchen, durch unterschiedliche Methoden die Strategien und Verhaltensweisen der Bots zu erkennen. Dies geschieht meistens durch die Analyse und Beobachtung des Netzwerkverkehrs. Dieser Ansatz lässt sich in Anomalie-basierte Erkennungsmethoden und Signaturbasierte Erkennungsmethoden unterteilen. [net.in.tum]

Anomalie-basierte Erkennungsmethode  
Die anomalie-basierten Erkennungsmethoden sind eine wichtige Art der Erkennung von Gefahren im IT Bereich. Bei dieser Methode wird das Verhalten der Computer auf Anomalien, also auf Auffälligkeiten, welche sich von der Norm abheben, untersucht. Indizien können beispielsweise hohes Netzwerkvolumen oder Verwendung unüblicher Ports sein. Zu dieser Methode zählen unter anderem:

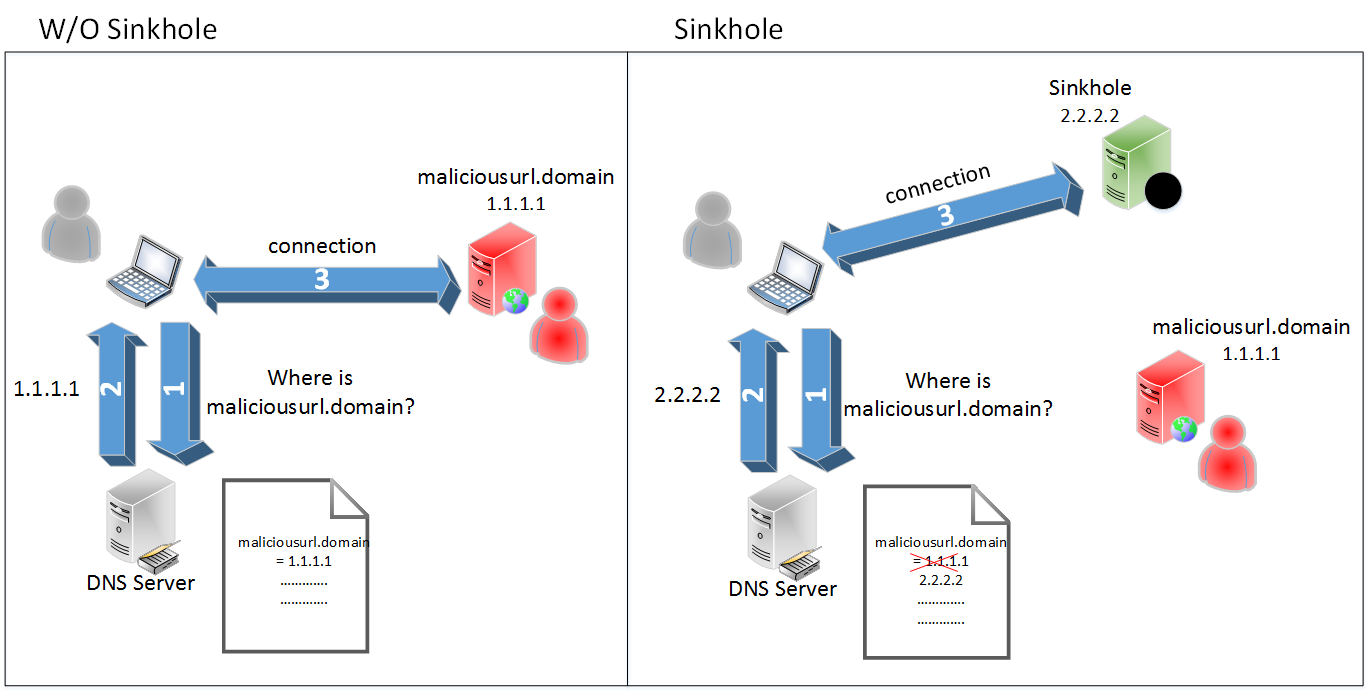
DNS-basierte Erkennungsmethode  
Hierbei wird vorwiegend der durch das **D**omain **N**ame **S**ystem erzeugte Netzwerkverkehr untersucht. Das DNS wird von den Botnetzen benutzt, um mit den Command-and-Control-Servern eine Kommunikationsverbindung aufzubauen. Durch DNS müssen IP-Adressen nicht statisch im Code des Bots stehen, sondern können dynamisch aufgelöst werden. DNS Verkehr bietet eine gute Analysegrundlage, da er in der Regel nicht sehr groß ist und sich somit so gut wie in Echtzeit scannen lässt. Ist bekannt, dass sich hinter einer Domain ein Botnetz Betreiber befindet, lässt sich diese Domain in der Regel sehr einfach sperren. Da dies den Botnetz Betreibern auch bekannt ist, nutzen diese sehr oft einen Domainname-Generation-Algorithm (DGA), welcher verwendet wird, um dynamisch neue Domains und Subdomains zu erzeugen. [net.in.tum] Sollte es doch vorkommen, dass IP Adressen im Angreifercode manuell hinterlegt sind, lässt sich auch hiergegen sehr gut vorgehen. Sollte eine Domäne, auf welche die IP Adresse verweist, bereits als bösartig identifiziert sein, so geht mit großer Wahrscheinlichkeit aller Datenverkehr zu dieser Adresse von bereits infizierten Hosts aus. Dadurch kann man die infizierten Maschinen im Netzwerk tracken, wann immer diese eine DNS Query absenden[[18]](#footnote-18)

Analyse von Logfiles und Records  
Im Allgemeinen ist es immer wichtig, Die Logfiles und Records zu analysieren, um ein genaueres Verständnis für den Netzwerkverkehr zu bekommen. Eine Logfunktion ist im Grunde in jeglichen Netzwerkgeräten sowie verschiedenen Softwarekomponenten enthalten (Beispielsweise der Firewall, Router, Switche etc), wodurch kein Mehraufwand durch das Sammeln dieser Daten entsteht. Die Logdaten bieten dem Systemadministrator ein Protokoll der Aktivitäten aller Applikationen im Netzwerk sowie ausgehende und eingehende Anfragen untereinander als auch aus dem Netzwerk. Es gibt verschiedene Techniken und Automatisierungsmethoden ,mit welchen man die Logfiles automatisch auf Unregelmäßigkeiten überprüfen kann. Erkennt man in den Logs beispielsweise, dass viele Anfragen regelmäßig an eine unbekannte IP geschickt werden, kann dies ein gutes Indiz für eine Infektion sein[[19]](#footnote-19)

Tracking Fast Flux Netzwerke  
Viele Botnetze benutzen sogenannte Fast-Flux Netzwerke(Siehe Kapitel „Botnetz Architektur“). Durch dieses ist es möglich, mithilfe von schnell wechselnden DNS Records, welche auf eine vielzahl von Hosts verweisen, eine Art Proxy Schicht zu erzeugen, um das Botnetz zu verschleiern. Obwohl das Netzwerk durch diese Große Anzahl IP Adressen, welche auf eine einzelne Domain verweisen, gut geschützt ist, ist es möglich, anhand der Eigenschaften der DNS Records, diese als bösartiges Netzwerk zu identifizieren. Die Records von Domänen, welche mit Fast Flux Netzwerken verbunden sind, haben typischer weise eine Sehr kurzen TTL(Time-to-live) Wert. Nachdem die TTL abgelaufen ist , wird eine neue Anfrage in der Regel an eine neue zugewiesene IP Adresse gesendet, welche sich komplett von der alten IP Adresse unterscheidet und häufig aus verschiedenen Netzwerken oder sogar von einem unterschiedlichen ISP stammen kann. Dies ermöglicht eine eindeutige Unterscheidung zu andern gutartigen Services und Webseiten mit niedriger TTL sowie beispielsweise google.com oder facebook.com. Diese geben für gewöhnlich IP Adressen zurück, welche starke Ähnlichkeiten zu einander aufweisen.[[20]](#footnote-20)



-Sinkhole Server  
Ein oft benutztes, DNS-basiertes Verfahren zur Identifikation von kompromittierten Systemen stellen sog. „Sinkhole Server“ dar. Es wird die schadsoftware analysiert, sodass man auf verwedete Domainnamen schließen kann, mit welchen die Schadprogramme zu kommunizieren versuchen. Danach ist es möglich, in Zusammenarbeit mit den entsprechenden Domain-Registrierungsstellen, diese betroffenen Domainnamen auf die Sinkhole Server umzuleiten. Die Aufgabe dieser besteht nun darin, die Zugriffe auf die Domainnamen zu protokollieren und die Quell IP sowie den Quell Port zu erfassen und den Zugriff mit einem Zeitstempel zu versehen. Da sich hinter den Domainnamen keine legitimen Dienstleistungen befinden und diese somit normalerweise auch nicht angesprochen werden, ist ein Zugriff auf eine solche Adresse ein gutes Indiz dafür, dass das Gerät hinter der Quell IP Adresse von einer Schadsoftware infiziert und zum Bot gemacht wurde.[[21]](#footnote-21)

  
Quelle: https://www.enisa.europa.eu/topics/incident-response/glossary/files/dns\_sinkhole

Data-Mining-basierte Erkennungsmethode  
Dieser Ansatz basiert darauf, aus gegebenen Daten neue Erkenntnisse zu sammeln. Es werden Data-Mining Strategien angewandt, um den Netzwerktraffic klassifizieren und gruppieren zu können. Die Daten sollen in zwei Kategorien unterteilt werden, um den normalen Traffic von dem, welcher vom Botnetz generiert wurde, unterschieden zu können und dadurch die infizierten Computer zu erkennen. Es werden hierbei Algorithmen für maschinelles Lernen angewandt, welche dann zum Beispiel auch für die Erzeugung von Signaturen verwendet werden können. [net.in.tum]

Host-basierte Erkennungsmethode  
Bei dieser Methode wird nicht das Netzwerk überwacht, um die Botnetz Aktivitäten zu erkennen, sondern die Computer selbst. Wenn der Computer von dem Botnetz infiziert wurde, lässt sich dies oft an ungewöhnlichen Systemaufrufen erkennen. So kann es sein dass zum Beispiel der Virenscanner plötzlich deaktiviert wird, da die Schadsoftware versucht sich zu verstecken oder es werden ungewöhnliche Verbindungen zu Zielen im Internet aufgebaut. All das kann auf eine mögliche Infizierung hinweisen.[[22]](#footnote-22)

Botnet Infiltration:  
Die Botnet Infiltration ist eine aktive Methode um sich gegen die Botnetze zur wehr setzen zu können. Diese kann in Software basierte und Hardware basierte Infiltration unterschieden werden. Die Software basierte Methode zielt darauf ab, durch reverse-engineeringe des Kommunikationsmechanismus auf verschiedene Schwachstellen im Code aufmerksam zu werden. Diese sollen im Anschluss ausgenutzt werden, um die Kontrolle über das Botnetz erlangen zu können. Mit der Kontrolle soll es möglich sein, Informationen zu sammeln, Beispielsweise über infizierte Hosts oder genaue Adresse des Botmaster, um diese Bedrohungen ausschalten zu können. Die zweite Methode ist die Hardware basierte Infiltration. Konnte man die IP eines Botmasters ausfindig machen und herausfinden, wo dieser gehostet wird, ist es möglich, in zusammenarbeit mit den Hostinganbietern über einen MirrorPort am verdächtigen Server die Kommunikation mitzuschneiden und zu Analysieren. Dies hilft nicht nur mehr über den Botmaster herauszufinden, sondern auf um beispielsweise infizierte Hosts ausfindig zu machen indem man sieht, zu welchen Clients Datenverkehr mit Anweisungen weitergeleitet wird. [[23]](#footnote-23) Die Infiltration eines Botnetzes ist jedoch rechtlich eine sehr heikle Angelegenheit. **Genauere Details werden zur rechtlichen Lage im Umgang mit Botnetzen werden in Kapitel „“ angesprochen**.(Kapitel ergänzen)

Machine Learning für die Botnetz Erkennung  
Machine Learning ist ein Teilbereich der künstlichen Intelligenz, welcher dazu beiträgt, dass IT Systeme in der Lage sind, auf Basis von vorhandenen Datenbeständen und Algorithmen Muster und Gesetzmäßigkeiten in Daten zu finden. Es wird „künstliches Wissen“ generiert und die gewonnenen Erkenntnisse können verallgemeinert und für andere Problemlösungen, sowie die Analyse von bislang unbekannten Daten verwendet werden.[[24]](#footnote-24) Die allgemeine Annahme hinter den Machine Learning basierten Methoden zur Botnetzerkennung ist, dass Botnetze ein eindeutiges, von normalem Traffic unterscheidbares Muster an Netzwerk Aktivität generieren. Diese Muster könnten mit Hilfe von Machine Learning Algorithmen entdeckt werden. Das generelle Problem, was sich hierbei auftut ist es, dass eine Verallgemeinerung des Wissens, welches aus einer begrenzten Anzahl vorheriger Erfahrungen abgeleitet wurde, hergestellt werden muss. Dafür gibt es im Gebiet des machine Learning mehrere Algorithmen, welche auf statistischen und rechnerischen Prinzipien basieren. Diese Algorithmen können, anhand des gewünschten Ergebnisses entweder in supervised oder unsupervised Algorithmen aufgeteilt werden.

Supervised Learning:  
Unter dem supervised Learning versteht man eine Klasse von klar definierten Maschine learnign Algorithmen, die eine Funktion generieren, welche die Eingaben auf gewünschte Aufgaben abbildet. Durch die Eingaben und Ausgaben werden diese Algorithmen trainiert und können verwendet werden um die Ausgaben für zukünftige Eingaben vorherzusagen. Diese können für die Botnetz-Erkennung beispielsweise zur Implementierung von Netzwerkverkehr Classifiern verwendet werden, welche eingesetzt werden können, um bösartigen von nicht-bösartigem Verkehr zu unterscheiden. Zu den bekanntesten Machine Learning Algorithmen gehören unteranderem Support Vector Machines und Artificial Neutal Networks.  
Unsupervised learning:  
Unsupervised Learning beschreibt Maschine Learning Algorithmen, bei denen die Trainingsdaten aus einer Reihe von Eingaben ohne zugehörige Ziel Ausgabewerte bestehen. Das Ziel des unsupervised Learning ist es, innerhalb der Eingabedaten Gruppen von ähnlichen Beispielen zu entdecken, was man auch als Glustering bezeichnet, die Verteilung von Daten im Eingaberaum zu bestimmen, was als Dichteschätzung bekannt ist, oder auch die Daten von einem höherdimensionalen Raum auf zwei oder drei Dimensionen für Virtualisierungszwecke zu projizieren. Im Bereich der Botnetzerkennung werden unsupervised Machine Learning Algorithmen dazu eingesetzt, Botnetz bezogenen Traffic zu Gruppieren. Die bekanntesten und beliebtesten unsupervised Machine Learning Algorithmen sind k-Means, X-Means und hierachisches Clusterung.[[25]](#footnote-25)

Für beide Varianten ist es wichtig, die Richtigen Features auszuwählen, auf welche bei der Auswertung geachtet werden soll. Der Nutzen von Maschine Learning zur identifikation von Botnetz Netzwerktraffic ist schon seit über einem Jahrzehnt von großem Intresse Innerhalb der Forschungsgemeinte, wodurch sich verschiedene Detection Methoden entwickeln konnten. In Zukunft und mit fortschreitender Entwicklung in den Bereichen Machine Learning und künstlicher Intelligenz werden sich die möglichen Erkennungsmethoden stetig verbessern.

Literaturverzeichnis  
[1] Kambourakis, Georgios; Anagnostopoulus, Marios; Meng, Weihzi; Zhou Peng(2019): Botnets: Architectures, Countermeasures, and Challenges(CRS Series in Security, Privacy and Trust) S4

[] Statista Research Department(2022) : Unternehmen, die in den letzten 12 Monaten eine Cyber-Attacke erlebt haben [online], <https://de.statista.com/statistik/daten/studie/1230157/umfrage/unternehmen-die-in-den-letzten-12-monaten-eine-cyber-attacke-erlebt-haben/> (abgerufen 4.12.22)

[2] Zipperle, Florian (2014): Überblick über Botnetz Erkennungsmethoden, , S. 19-22, [online] <https://www.net.in.tum.de/fileadmin/TUM/NET/NET-2014-08-1/NET-2014-08-1_03.pdf> (Abgerufen 02.12.22)

[] Laass, Matthis (2011) Botnetze: Aufbau, Funktion & Anwendung S6-7[online] <https://www.researchgate.net/publication/263842949_Botnetze_Aufbau_Funktion_Anwendung> (Abgerufen 5.12.22)

[]BSI (2022) Denial of Service(DoS) [Online] <https://www.bsi.bund.de/dok/6599510> (Abgerufen 06.11.2022)

[]Ionos(2022)UDP Flood[Online]<https://www.ionos.de/digitalguide/server/sicherheit/udp-flood/>

[]Ionos(2022) Syn Flood[Online] <https://www.ionos.de/digitalguide/server/sicherheit/syn-flood/>

[]Myrasecurity(2022) http Flood[Online] <https://www.myrasecurity.com/de/http-flood-attacke/>

[]Cloudflare(2022) DNS Amplification [Online] <https://www.cloudflare.com/de-de/learning/ddos/dns-amplification-ddos-attack/>

[]Cloudflare(2022) What is a Spam Bot[Online] <https://www.cloudflare.com/de-de/learning/bots/what-is-a-spambot/>

Ulf-Jost Kossol(2018) Social Bots – Gefahren und Lösungen[Online] [Social Bots – Gefahren und Lösungen (entwickler.de)](https://entwickler.de/security/social-bots-gefahren-und-losungen#:~:text=Bots%20k%C3%B6nnen%20durch%20ihre%20Aktivit%C3%A4ten,auch%20einzelne%20Produkte%20negativ%20bewerten.)

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. <https://www.net.in.tum.de/fileadmin/TUM/NET/NET-2014-08-1/NET-2014-08-1_03.pdf> [↑](#footnote-ref-5)
6. Botnetze: Aufbau, Funktion & Anwendung Matthis C. Laass Fachhochschule Aachen S2 [↑](#footnote-ref-6)
7. Botnetze: Aufbau, Funktion & Anwendung Matthis C. Laass Fachhochschule Aachen S3 [↑](#footnote-ref-7)
8. Botnetze: Aufbau, Funktion & Anwendung Matthis C. Laass Fachhochschule Aachen S3-4 [↑](#footnote-ref-8)
9. Botnetze: Aufbau, Funktion & Anwendung Matthis C. Laass (2011) Fachhochschule Aachen S6-7 [↑](#footnote-ref-9)
10. Botnetze: Aufbau, Funktion & Anwendung Matthis C. Laass Fachhochschule Aachen S7-8 [↑](#footnote-ref-10)
11. <https://www.bsi.bund.de/DE/Themen/Verbraucherinnen-und-Verbraucher/Cyber-Sicherheitslage/Methoden-der-Cyber-Kriminalitaet/DoS-Denial-of-Service/dos-denial-of-service_node.html> Zugriff 06.11.2022 [↑](#footnote-ref-11)
12. https://www.ionos.de/digitalguide/server/sicherheit/udp-flood/ [↑](#footnote-ref-12)
13. <https://www.ionos.de/digitalguide/server/sicherheit/syn-flood/> zugriff 06.11.2022 [↑](#footnote-ref-13)
14. <https://www.myrasecurity.com/de/http-flood-attacke/> Zugriff: 6.11.2022 [↑](#footnote-ref-14)
15. <https://www.cloudflare.com/de-de/learning/ddos/dns-amplification-ddos-attack/> Zugriff 6.11.2022 [↑](#footnote-ref-15)
16. <https://www.cloudflare.com/de-de/learning/bots/what-is-a-spambot/> Zugriff 6.11.2022 [↑](#footnote-ref-16)
17. [Social Bots – Gefahren und Lösungen (entwickler.de)](https://entwickler.de/security/social-bots-gefahren-und-losungen#:~:text=Bots%20k%C3%B6nnen%20durch%20ihre%20Aktivit%C3%A4ten,auch%20einzelne%20Produkte%20negativ%20bewerten.) [↑](#footnote-ref-17)
18. Enisa: [Botnets: Measurement, Detection, Disinfection and Defence — ENISA (europa.eu)](https://www.enisa.europa.eu/publications/botnets-measurement-detection-disinfection-and-defence) [↑](#footnote-ref-18)
19. Enisa: [Botnets: Measurement, Detection, Disinfection and Defence — ENISA (europa.eu)](https://www.enisa.europa.eu/publications/botnets-measurement-detection-disinfection-and-defence) [↑](#footnote-ref-19)
20. Enisa: [Botnets: Measurement, Detection, Disinfection and Defence — ENISA (europa.eu)](https://www.enisa.europa.eu/publications/botnets-measurement-detection-disinfection-and-defence) [↑](#footnote-ref-20)
21. https://www.bsi.bund.de/DE/Themen/Verbraucherinnen-und-Verbraucher/Cyber-Sicherheitslage/Methoden-der-Cyber-Kriminalitaet/Botnetze/Fragen-und-Antworten/fragen-und-antworten\_node.html [↑](#footnote-ref-21)
22. https://www.net.in.tum.de/fileadmin/TUM/NET/NET-2014-08-1/NET-2014-08-1\_03.pdf [↑](#footnote-ref-22)
23. https://www.net.in.tum.de/fileadmin/TUM/NET/NET-2014-08-1/NET-2014-08-1\_03.pdf [↑](#footnote-ref-23)
24. [Was ist Machine Learning? (bigdata-insider.de)](https://www.bigdata-insider.de/was-ist-machine-learning-a-592092/) [↑](#footnote-ref-24)
25. [On the use of machine learning for identifying botnet network traffic (riverpublishers.com)](https://www.riverpublishers.com/journal_read_html_article.php?j=JCSM/4/2/1#rsec4) [↑](#footnote-ref-25)