Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

Multi-Level Erkennung von IoT-Botnetzen während der Scan-, Ausbreitungs- und Angriffsphase

Exposé zur Dissertation
zur Erlangung des Grades eines
Doktors

am Fachbereich 4 - Informatik, Kommunikation und Wirtschaft
der Hochschule für Technik und Wirtschaft Berlin

vorgelegt von
Jan Schröder

Berlin, 16. Januar 2023

Inhaltsverzeichnis

Begriff	Erklärung
Malware	Lustige Erklärung
Internet of Things	Noch eine lustige Erklärung
Botnetz	(*englisch: botnet*) Saving it for later
Software-defined Networking	
Erkennungsindikatoren	Dieser Sammelbegriff beinhaltet Variablen, Daten, Charakteristiken, Attribute etc. die die Ausgabe eines ML Modells repräsentieren.

Tabelle 1: Diese Tabelle erläutert Begriffe, welche sowohl im Exposé als auch der späteren Dissertation Verwendung finden.

1 Forschungsfragen und Aufbau des Exposés

Botnetze im Internet of Things (IoT) Gebiet haben in der modernen Welt sehr stark an Bedeutung gewonnen. Gerade im privaten Haushalt finden sich immer mehr IoT-Geräte die durch Botnetze wie Mirai und Hajime infiziert werden können. Daher ist es sehr wichtig, dass diese Botnetze durch eine Erkennung identifiziert und abgewehrt werden. Die Dissertation soll sich mit der Erkennung von Botnetzen beschäftigen und die Identifikation optimieren.

Forschungsfragen für die Dissertation

Die Dissertation soll spezifisch die folgenden Forschungsfragen beantworten:

Q1: Wie lassen sich Botnetze während der Verbreitungsphase erkennen?

Q2: Wie lassen sich Botnetze während der Scan-Phase erkennen?

Q3: Welche Kombinationen von bereits bekannten Methoden führen zu einer Verbesserung der Erkennung von Botnetzen?

Q4: Welche Kombinationen von bereits bekannten Techniken führen zu einer Verbesserung der Erkennung von Botnetzen?

Q5: Wie lässt sich die Erkennung von Botnetzen optimieren, wenn mehrere Erkennungsstufen verwendet werden?

Q6: Wie sieht ein Datensatz aus, der zur Multi-Level Erkennung verwendet wird?

In Frage **??** und **??** soll es um die Erkennung von Botnetzen während der ersten beiden Phasen gehen. Was die Phasen sind erklärt das nachfolgende Kapitel.

In den folgenden Kapitel **??** werden die theoretischen Hintergründe zum Thema der Dissertation und verschiedene Prozesse zur Erkennung von Botnetzen, sowie mehrere Botnetze die in Fallstudien eingesetzt werden sollen beschrieben. Kapitel **??** erläutert den aktuellen Stand der Forschung zur Erkennung von Botnetzen, sowie zu den Themen die in der Dissertation behandelt werden sollen. In Kapitel **??** werden die Ziele der Dissertation aufgestellt. Das Kapitel **??** erklärt wie in der Dissertation vorgegangen werden soll, um ein Konzept mit einer dazugehörigen Implementierung zu erstellen sowie den geplanten Aufbau eines Laborexperiments und mögliche Fallstudien. Das letzte Kapitel **??** stellt einen Zeitplan vor sowie die Struktur, die die Dissertation haben soll.

2 Theoretischer Hintergrund

Internet of Things (IoT) ist ein Gebiet, welches dem Alltag viele Vorteile bringt. Durch die Möglichkeit Geräte aus dem Alltag mit dem Internet zu verbinden, birgt IoT aber auch Sicherheitslücken die besagte Geräte sehr Gefährlich werden lässt. Um dem entgegen zu wirken soll sich die Dissertation im Gebiet der Malwareforschung befinden. Im speziellen soll es dabei um die Erkennung von Botnetzen gehen, welche Angriffe über IoT-Geräte ausführen.
Ein Botnetz ist der Zusammenschluss von Hosts, auch Bots oder Zombies genannt, gesteuert von einem Angreifer, auch Botmaster genannt in einem Overlay-Netzwerk [**?**]. Die Botnetze nutzen Zero-day Schwachstellen, Peer-2-Peer Netze, Phishing Angriffe, Anonyme Netzwerke, Blockchain Netzwerke und Stromnetze zur Verbreitung und ihrer Verwendungszwecke [**?**, **?**]. Auf Basis der Architektur des Botnetzes findet zu jeder Zeit ein Kommunikations- und Kontrollprozess mit dem Command und Control (C&C)-Server statt. Der C&C-Server gibt den Bots Befehle die diese dann durchführen [**?**] zum Beispiel, über das Internet Relay Chat (IRC)-Protokoll.
Botnetze durchlaufen drei Phasen wie Wazzan et al. [**?**] beschreiben, scannen, ausbreiten und angreifen. Während der Scan-Phase sucht ein Bot nach vulnerablen IoT-Geräten und infiziert das Gerät entweder durch brute force Methoden oder durch Ausnutzen einer Schwachstelle. In der Ausbreitungs-Phase ist eine lauffähige Version des Bots installiert und auf Basis der Architektur des infizierten Geräts ausgeführt. Um auf dem Gerät Malware zu verhindern die nicht vom Botnetz selbst ausgeführt wird, stoppt der Bot andere Prozesse um Ports für sich selbst zu blocken. Daraufhin rekrutiert das bösartige Programm weitere Bots um das Botnetz so schnell wie möglich zu erweitern. In der Angriffs-Phase führt das Botnetz Angriffe wie Distributed Denial of Service (DDoS), crypto mining und spam Angriffe aus. Die erläuterten Phasen beschreiben auch Studien wie [**?**, **?**, **?**, **?**].
Nach der Erläuterung wie Botnetze funktionieren wäre nun zu fragen, wie der Prozess eines Botnetzes erkennbar ist um IoT-Geräte entsprechend zu schützen. Nach Xing et al. [**?**] kann die Botnetz Erkennung in Honeypot Analyse, Signaturen aus der Kommunikation und abnormales Verhalten klassifiziert werden. Wie Abbildung **??** zeigt, unterteilen diese Klassifikationen Methoden zur Erkennung.

Die *Honeypot Analyse* erkennt Code Beispiele durch das Honeypot trapping was eine hohe Genauigkeit von bereits bekannten Botnetzen ermöglicht. Die Honeypot Methoden können verschlüsselten Netzwerkverkehr nur schlecht erkennen sowie unbekannte Botnetze. Bots die eigene Funktion zur Umgehung von Honeypots besitzen, können durch fehlende Benutzereingriffe auch nicht von der *Honeypot Analyse* erfasst werden. Weit verbreitet sind die Methoden Erkennung von *Kommunikationssignaturen* anhand von Signaturen und Muster. Dabei werden in Intrusion Detection Systems (IDS) Regeln für den Merkmalsabgleich hinterlegt um Botnetz aktivitäten zu identifizieren. Dadurch können IDS Botnetze mit bestimmten Merkmalen erkennen, aber unbekannte Funktionen werden dabei nicht erkannt sowie auch Botnetze die Techniken zur Verschleierung von Code nutzen. Bei den Methoden die durch *abnormales Verhalten* ist die Idee, Hostverhalten oder Netzwerkverkehr Auffälligkeiten zwischen gutartigem und bösartigem

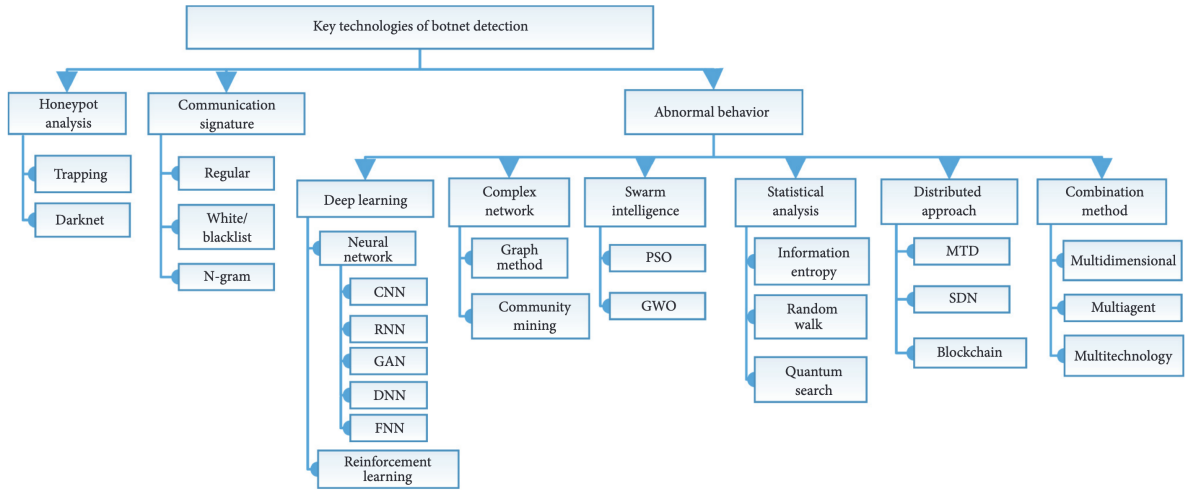

Abbildung 1: Klassifizierte Erkennungsmethoden von Botnetzen (übernommen von [?]).

Verhalten zu erkennen. Neben den erwähnten Methoden erläutern Singh et al. [?] Techniken, zur Erkennung von Botnetzen. Singh et al. klassifizieren die Techniken in Flow-, Anomalie-, Flussmittel-, Domain Generation Algorithm (DGA)-basierten [?] und Bot infizierungs Erkennung. Bei der Flow-basierten Technik findet eine Klassifizierung des Netzwerkverkehrs auf Basis von verschiedenen Parametern statt, wobei eine Aufteilung des Verkehrs in bösartig und gutartig stattfindet. Anhand von Parametern oder Mustern versucht die Anomalie-basierte Technik anomalien im Netzwerkverkehr zu finden, die sich vom regulären Verkehr unterscheiden. Über die Flussmittel-basierte Technik werden im Netzwerkverkehr IP-Flüsse gefunden. Dabei wird darauf geachtet, wie sich die IP-Larte verändert, welche in Relation zur einer Domäne steht und einen niedrigen Time To Live (TTL)-Wert hat. Die Konzentrationen auf abgefragte Domänen führt die DGA-basierte Technik durch. Mit dieser Technik soll zwischen algorithmischen bösartig erzeugten und gutartigen Domänen differentiert werden. Nach dem Stand von [?] versuchen die aktuelleren Technikansätze, anstatt C&C-Server zu identifizieren, infizierte Geräte zu erkennen.
Die Umsetzung über mehrere Erkennungslevel erklären Stevanovic und Pedersen [?]. Dabei führen Stevanovic und Pedersen eine Analyse des Netzwerkverkehrs durch, um die Kommunikation mit dem C&C-Server sowie den Angriffsverkehr anhand von TCP, UDP und DNS zu erkennen. Die Erkennung basiert auf supervised Machine Learning um bestimmte Muster zu identifizieren. Das komplette System besteht aus insgesamt drei Komponenten: die Verarbeitung, Klassifizierung und Client Analyse. In der ersten Komponente werden der Netzwerkverkehr verarbeitet durch Analyse und Extraktion anhand von statistischen Funktionen. Abbildung **??** zeigt eine Liste mit den extrahierten Informationen aus TCP und UDP.

Bei der DNS Analyse werden Fully Qualified Domain Names (FQDN) beobachtet und für jeden FQDN, statistische Eigenschaften extrahiert, welche in Abbildung **??** aufgelistet sind. Zur Klassifizierung nutzt die Botnetz Erkennung einen Random Forest Classifier

Feature	Type	Number[1]
Basic conversation features		
Port number	Numerical	2
Layer 7 protocol	Categorical	1
Duration (last pkt - first pkt)	Numerical	1
Total number of packets	Numerical	2
Total number of Bytes	Numerical	2
Mean of the number of Bytes per packet	Numerical	2
Std of the number of Bytes per packet	Numerical	2
Geographical features		
Remote IP country	Categorical	1
Remote IP continent	Categorical	1
Time-based features		
Number of packets per second	Numerical	2
Number of Bytes per second	Numerical	2
Mean of packets inter-arrival time	Numerical	2
Std of packets inter-arrival time	Numerical	2
Bidirectional features		
Ratio of number of packets OUT/IN	Numerical	1
Ratio of number of Bytes OUT/IN	Numerical	1
Ratio of inter-arrival times OUT/IN	Numerical	1
TCP specific features		
Number of three way handshakes	Numerical	1
Number of connection tear downs	Numerical	1
Number of complete conversation	Numerical	1
Average conversation duration	Numerical	1
TCP Flags	Categorical	2
Percentage of TCP SYN packets	Numerical	2
Percentage of TCP SYN ACK packets	Numerical	2
Percentage of TCP ACK packets	Numerical	2
Percentage of TCP ACK PUSH packets	Numerical	2
Percentage of TCP FIN packets	Numerical	2
Percentage of TCP RST packets	Numerical	2

Abbildung 2: TCP und UDP Informationen statistisch zusammengefasst (übernommen von [**?**]).

um dann über eine Client entitäten Analyse einen Report zu erstellen über infizierte Geräte. Für die Dissertation sollen wie bei Stevanovic und Pedersen auch mehrere Level zur Erkennung von Botnetzen eingesetzt werden. Das daraus resultierende Ziel erklärt Kapitel **??** ausführlicher. Die Botnetz Erkennung nach den unterschiedlichen Methoden und Techniken mit mehrere Level soll mehr Geräte vor der illegalen Verwendung von Botnetzen schützen.

Angriffsmöglichkeiten zum Testen

Um den Prozess der Botnetz Erkennung durch Fallstudien zu testen, sollen in der Dissertation verschiedene Botnetze Verwendung finden und alle drei Phasen durchführen. Dabei sollen bereits bekannte Botnetze implementiert und ausgeführt werden. Zusätzlich soll ein Botnetz implementiert werden um Herauszufinden, ob das System auch neue, unbekannte Botnetze erkennt. Der Fokus bei den Botnetzen soll auch weiterhin im IoT Bereich liegen. Kolias et al. [**?**] weisen in ihrer Arbeit neben Mirai auf das Botnetz Hajime hin, welches in der Dissertation für die Botnetzerkennung ausgeführt werden soll. Nach Kolias et al. besteht Mirai aus vier Komponenten, dem *Bot*, dem *C&C-Server*, der *loader* und der *report Server*. Der *Bot* und *C&C-Server* weichen nicht von der allgemeinen Funktionsweise ab, wie in **??** erklärt. Der *loader* übernimmt die Kommunikation mit neuen infizierten

Feature	Type
FQDN-based features	
Number of tokens	Numerical
Avg length of token	Numerical
Length of SLD (Second Level Domain)	Numerical
Length of Domain	Numerical
Language of SLD	Categorical
Entropy (range of characters) for SLD	Numerical
Distance from n-grams of legitimate domains (`alexa.com`) for SLD	Numerical
Distance from n-grams of dictionary words domains for SLD	Numerical
Number of dictionary words in SLD	Numerical
Ratio of numerical characters in SLD	Numerical
Ratio of vowels in SLD	Numerical
Ratio of consonants in SLD	Numerical
Number of dictionary words in domain	Numerical
Ratio of numerical characters in domain	Numerical
Ratio of vowels in domain	Numerical
Ratio of consonants in domain	Numerical
Query-based features	
Type of query	Categorical
Number of queries	Numerical
Mean of query length	Numerical
Std of query length	Numerical
Mean of queries inter-arrival time	Numerical
Std of queries inter-arrival time	Numerical
Response-based features	
Number of query responses	Numerical
Mean of query response length	Numerical
Std of query response length	Numerical
Mean of query responses inter-arrival time	Numerical
Std of query response inter-arrival time	Numerical
Number of NOERROR responses	Numerical
Number of NXDOMAIN responses	Numerical
Avg number of answers	Numerical
Avg number of authority answers	Numerical
Avg number of additional answers	Numerical
Avg number of resolved IPs	Numerical
Mean of the value of TTL (Time-To-Live) field	Numerical
Std of the value of TTL field	Numerical
Geographical features	
Number of countries resolved IPs belong to	Numerical
Number of ASs resolved IPs belong to	Numerical

Abbildung 3: DNS Informationen statistisch zusammengefasst (übernommen von [**?**]).

Geräten und verteilt direkt an sie ausführbare Dateien. Der *report Server* verwaltet Informationen über alle Geräte im Botnetz über eine Datenbank und kommuniziert mit den neu infizierten Geräten. Im folgenden Ablauf operiert und kommuniziert Mirai. Zu Beginn scannt Mirai zufällige IP Adressen über TCP ob die Ports 23 oder 2323 zuhören. Über brute-force Angriffe sucht der bot IoT Geräte, die schlecht konfiguriert sind (z.B. Standard Login Daten die nicht geändert wurden). Mit einer geöffneten Shell gibt der Bot Informationen über das Gerät an den report Server über einen anderen Port. Der botmaster prüft über den C&C-Server neu ausgewählte Geräte und anhand des report Servers den aktuellen Status des Botnetzes. Anhand der Informationen über die Geräte kann der Botmaster entsprechende Geräte zum infizieren auswählen und über ein Infect-Befehl über den loader ausführen. Der loader führt auf den ausgewählten Gerät Instruktionen aus zum herunterladen der Malware Binärdatei. Dabei stellt die Malware sicher, dass keine anderen Malware Programme auf dem Gerät ausgeführt werden und schließt sowohl Secure Shell (SSH), als auch Telnet Programme. Der neue Bot bekommt über eine Domäne vom C&C-Server nun mögliche Angriffsbefehle. Den initialen Prozess der Suche nach offenen Ports führt auch Hajime durch. Hajime ist ein Peer-to-Peer Netzwerk, welches

auf BitTorrent's Distributed Hash Table (DHT) aufbaut [**?**, **?**]. BitTorrent nutzt das Kademlia Protokoll [**?**] und zusätzlich zur direkten Peer-to-Peer Kommunikation nutzt Hajime zusätzlich das uTorrent Transport Protocol. Für weitere technische Erläuterungen zu Hajime, analysieren [**?**] die Phasen des Botnetzes.

3 Stand der Forschung

Im Zusammenhang mit der Botnetzerkennung findet ein Großteil der Forschung zu diesem Gebiet im Zusammenhang mit Maschinellem Lernen (ML) statt wie unter anderem zum Beispiel [?, ?, ?] zeigen. Wazzan et al. [?] schlägt die Kombination mehrerer Technologien vor womit sich die Dissertation auch beschäftigen soll. Daher sollen neben der Erkennung über ML auch weitere Technologien mitbetrachtet werden. Wazzan et al. fasst anhand verschiedener Studien folgende weitere Technologien zusammen: Software Defined Network (SDN), Edge Computing, Blockchain, Fog Computing und Network Function Virtualization (NFV). Im folgenden wird der aktuelle Stand der Forschung zu jeder dieser Technologien betrachtet.

Zha et al. [?] beschreiben eine Bibliothek, welche Botnetz Erkennung anhand eines SDNs durchführt. Bei dem Konzept des Frameworks handelt es sich darum, Botnetz und CC-Kanal Aktivitäten zu identifizieren, damit die Robustheit der Erkennung erhöht wird. Zum Aufbau gehört ein SDN Controller, welcher ein ML Modell enthält, dass über Erkennung und Überwachung trainiert. Die Überwachung des Netzwerks, findet über Software Switches statt, welche vollen Zugang zum Netzwerkverkehr von Virtuellen Maschinen (VM) haben. Das ML Modell wird für das komplette Netzwerk verwendet um Bot Aktivitäten zu erkennen, während für einzelne Server, dessen VMs mit einem Software Switch kommunizieren, lokale Netzwerküberwachung stattfindet. Über die einzelnen Software Switches, findet die C&C Identifizierung statt. Dabei ist speziell Peer-to-Peer Verkehr, HTTP und IRC Protokolle im Fokus. Das ML Modell ist ein Neuronales Netzwerk (NN), welches kompromitierte Hosts findet anhand von aufgebauten Verbindungen des Hosts. Die Entscheidung, ob ein Host zu einem Bot wird, entscheidet das NN über einen festgelegten Schwellenwert. Mit einer einzelnen positiven Identifizierung ist aber dennoch nicht eindeutig geklärt, dass der Host auch wirklich ein Bot ist, da er z.B. nicht unbedingt mit dem C&C-Server kommuniziert. An dieser Stelle wird der Host dennoch dauerhaft weiter beobachtet. An dieser Stelle zeigt der aktuelle Stand Forschung, dass die Erkennung nichts während der Scan-Phase durchführt. Damit ergibt sich ein Ziel, was in der Dissertation verfolgt werden soll und weiter in Kapitel **??** erläutert wird. Zusätzlich zu dem beschriebenen Framework, erläutern Chen et al. [?] ein weiteres Framework unter der Verwendung von SDNs.

4 Ziel der Dissertation

Mit der Dissertation soll die Botnetzerkennung optimiert werden. Ziel ist es, ein Konzept zu erstellen, zusammen mit einer Implementierung. Das Konzept soll eine Botnetzerkennung erläutern, welche zu jeder Phase eines Botnetzes, eine Erkennung entwirft, die aus einer Kombinationen mehrerer Erkennungsmethoden besteht. Weiterhin soll herausgefunden werden, welche Methoden und Techniken die wenigsten false positives erzielen um diese dann miteinander zu kombinieren, damit daraus eine mehrstufige Erkennung entsteht. Welche Methoden und Techniken genau eingesetz werden, soll mit dieser Dissertation zu Beginn herausgefunden werden. Ein weiteres Ziel ist, ob die Erkennung auch unbekannte Botnetze identifizieren kann.

Ziel der Dissertation ist es nicht, aus der Erkennung Maßnahmen zur Verteidigung gegen Botnetze zu treffen. Die Dissertation soll sich ausschließlich auf die Erkennung konzentrieren.

5 Forschungsdesign und Methodik

Zu Beginn des Konzepts um die Erkennung von Botnetzen zu optimieren, soll eine Zusammenfassung von bekannten Methoden durchgeführt werden. Anhand der Zusammenfassung soll erkennbar werden, welche Methoden die besten Ergebnisse liefern und welche Kombination von Methoden und Techniken am plausibelsten sind. Die Dissertation geht anschließend mit einem Laborexperiment einher. Um die Erkennung eines Botnetzes zu testen sollen verschieden typische IoT-Geräte aufgebaut werden, die von Botnetzen wie Mirai angegriffen werden. Dazu ist es nötig, dass entsprechende Geräte ausgewählt werden, die von den Botnetzen infiziert werden können.

In einer Fallstudie ist vorgesehen, Botnetze auszuführen um entsprechend Daten zu sammeln, damit eine aussagekräftige Auswertung zu plausiblen Ergebnissen führt.

Das Laborexperiment soll ein lokales Netzwerk mit gängigen Smart Home Geräten darstellen, die in einem privaten Haushalt verwendet werden. Wie schon in **??** erläutert, soll Mirai und Hajime entsprechend eingesetzt werden. Im optimalen Fall, soll für die Dissertation auch ein eigenes Botnetz implementiert werden, um zu prüfen, ob die Botnetz Erkennung auch unbekannte Botnetze erkennt.

6 Zeitplan und Struktur der Dissertation

Die voraussichtliche Dauer der Dissertation beträgt 4 Jahre und 6 Monate.

Zeit	Vorgehen
März 2023	Grundlagen, verwandte Arbeiten erläutern
August 2023	Laborexperiment starten, Konzept in der Dissertation beschreiben
Januar 2025	Implementierung der IoT Geräte beschreiben
März 2025	Fallstudien beschreiben
Juli 2025	Ergebnisse erläutern, Fazit schreiben
Februar 2026	Einleitung schreiben, Kontrolle der Dissertation
Juli 2026	Abgabe

Zum Ende der Arbeitszeit soll die Dissertation über eine externe Stelle auf Plagiate, Rechtschreibung etc. geprüft werden.

Kapitelstruktur der Dissertation

I Einführung in die Dissertation

II Grundlagen und Verwandte Arbeiten

III Konzept der Mehrstufigen Botnet Erkennung

IV Implementierung des Experiments und der Fallstudien

V Evaluierung und Ergebnisse