



**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, 3. Februar 2023

# Inhaltsverzeichnis

1	Forschungsfragen und Aufbau des Exposés	2
2	Theoretischer Hintergrund	3
3	Stand der Forschung	8
4	Ziel der Dissertation	11
5	Forschungsdesign und Methodik	12
6	Zeitplan und Struktur der Dissertation	13

Begriff	Erklärung
Malware	Lustige Erklärung
Internet of Things	Noch eine lustige Erklärung
Botnetz	( <i>englisch: botnet</i> ) 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 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?

In Frage RQ1 und RQ2 soll es um die Erkennung von Botnetzen während der frühen Phasen gehen während Botnetze sich in einem Netzwerk verbreiten. Mit Frage RQ3 soll geklärt werden, welche gängigen Erkennungsmethoden am besten separat funktionieren und wie diese miteinander in Verbindung gebracht werden können. Dies steht auch mit Forschungsfrage RQ4 in Relation, da über die Methoden mehrere Techniken zur Erkennung in Verbindung gebracht werden sollen. Die Frage RQ5 soll klären, wie die Kombination von Methoden und die Kombination von Techniken zu einer möglichen Optimierung der Botnetzerkennung führen.

In den folgenden Kapitel 2 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 3 erläutert den aktuellen Stand der Forschung zur Erkennung von Botnetzen, sowie zu den Themen die in der Dissertation behandelt werden sollen. In Kapitel 4 werden die Ziele der Dissertation aufgestellt. Das Kapitel 5 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 6 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 [1]. Die Botnetze nutzen Zero-day Schwachstellen, Peer-2-Peer Netze, Phishing Angriffe, Anonyme Netzwerke, Blockchain Netzwerke und Stromnetze zur Verbreitung und ihrer Verwendungszwecke [2, 3]. 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 [4] zum Beispiel, über das Internet Relay Chat (IRC)-Protokoll.

Botnetze durchlaufen drei Phasen wie Wazzan et al. [5] 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 [6, 7, 8, 9].

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. [1] kann die Botnetz Erkennung in Honeypot Analyse, Signaturen aus der Kommunikation und abnormales Verhalten klassifiziert werden. Wie Abbildung 1 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

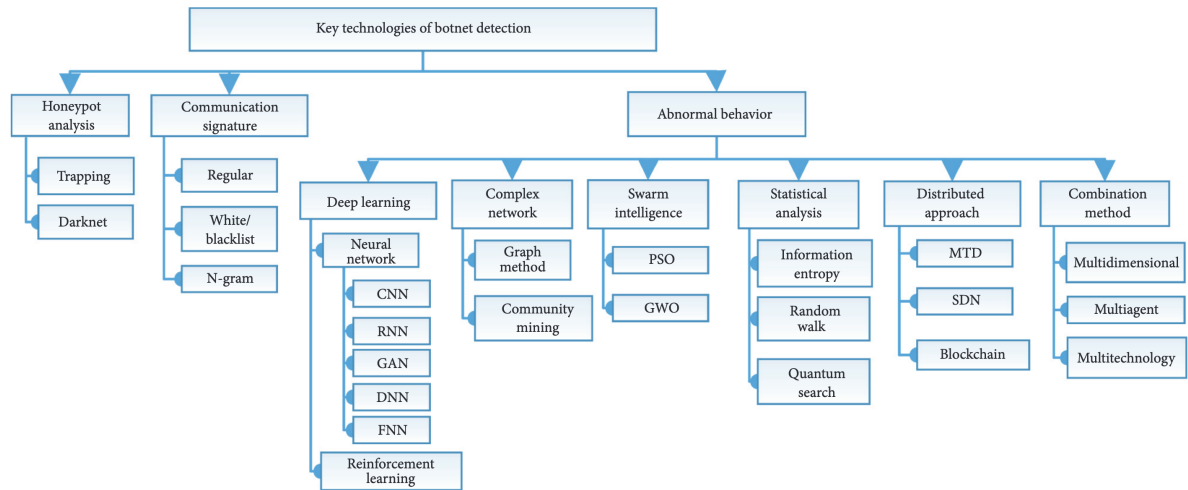


Abbildung 1: Klassifizierte Erkennungsmethoden von Botnetzen (übernommen von [1]).

Verhalten zu erkennen. Neben den erwähnten Methoden erläutern Singh et al. [10] Techniken, zur Erkennung von Botnetzen. Singh et al. klassifizieren die Techniken in Flow-, Anomalie-, Flussmittel-, Domain Generation Algorithm (DGA)-basierten [11] 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 böse 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 böse erzeugten und gutartigen Domänen differenziert werden. Nach dem Stand von [10] versuchen die aktuelleren Technikansätze, anstatt C&C-Server zu identifizieren, infizierte Geräte zu erkennen.

Die Umsetzung über mehrere Erkennungslevel erklären Stevanovic und Pedersen [12]. 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 2 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 3 aufgelistet sind. Zur Klassifizierung nutzt die Botnetz Erkennung einen Random Forest Classifier

Feature	Type	Number <sup>1</sup>
<b>Basic conversation features</b>		
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
<b>Geographical features</b>		
Remote IP country	Categorical	1
Remote IP continent	Categorical	1
<b>Time-based features</b>		
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
<b>Bidirectional features</b>		
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
<b>TCP specific features</b>		
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 [12]).

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 4 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. [13] 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 2 erklärt. Der *loader* übernimmt die Kommunikation mit neuen

Feature	Type
<b>FQDN-based features</b>	
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
<b>Query-based features</b>	
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
<b>Response-based features</b>	
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
<b>Geographical features</b>	
Number of countries resolved IPs belong to	Numerical
Number of ASs resolved IPs belong to	Numerical

Abbildung 3: DNS Informationen statistisch zusammengefasst (übernommen von [12]).

infizierten 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 [14, 15]. BitTorrent nutzt das Kademlia Protokoll [16] und zusätzlich zur direkten Peer-to-Peer Kommunikation nutzt Hajime zusätzlich das uTorrent Transport Protocol. Für weitere technische Erläuterungen zu Hajime, analysieren [14] die Phasen des Botnetzes.

## 3 Stand der Forschung

### Methoden zur Erkennung von Botnetzen

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 [17, 18, 19] zeigen. Wazzan et al. [5] schlägt die Kombination mehrerer Erkennungsmethoden vor, womit sich die Dissertation auch beschäftigen soll. Nach verschiedener Studien schlagen Wazzan et al. folgende Erkennungsmethoden vor: 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. [20] beschreiben eine Bibliothek, welche Botnetz Erkennung anhand eines SDNs durchführt. Bei dem Konzept des Frameworks handelt es sich darum, Botnetz und C&C-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. Der aktuelle Stand Forschung zeigt hier, dass die Erkennung nicht während der Scan-Phase durchgeführt wird, womit sich ein Ziel ergibt Botnetzerkennung in frühen Botnetzphase durchzuführen, was in der Dissertation verfolgt werden soll und weiter in Kapitel 4 erläutert wird. Zusätzlich zu dem beschriebenen Framework, erläutern Negera et al. [21] verschiedene Arbeiten zur Erkennung von Botnetzen in SDNs über Modelle des ML.

Neben der Erkennung mit SDNs findet in der aktuellen Forschung auch die Erkennung von Botnetzen über Edge Computing Architekturen statt, wie [22] in ihrer Arbeit zeigen. Bei dieser Erkennung verwenden Gromov et al. in einem Edge Computing System ein Convolutional Neural Network (CNN). Der Netzwerkverkehr der einzelnen IoT-Geräte wird an ein NVIDIA Jetson Nano Entwickler Kit zur Botnetzerkennung geleitet. Dabei werden anhand von Computer Vision Bilder zur Erkennung erzeugt, welche in auffällige und unauffällige Bilder klassifiziert werden um die Performance zu erhöhen sowie das Netzwerk zu verkleinern.

Bei der Blockchain basierten Erkennung geht es darum eine Blockchain zu implementieren und in diesem Netzwerk ein IoT Netzwerk auf Botnetze zu überwachen. Dabei werden unterschiedliche Methoden innerhalb der Blockchain implementiert und mit den Vorteilen der Blockchain verwendet. Sagirlar et al. [23] nutzen diese Methode indem sie innerhalb der Blockchain eine Peer-to-Peer Botnetz Erkennung implementieren. Salim et al. [24] erläutern ein Framework, welches Digital Twins und Packet Auditor in einer Blockchain zur Erkennung einsetzen.

Neben Blockchains wird auch Fog Computing zur Botnetzerkennung eingesetzt. [25] stellen ein Framework vor, welches Anomalien in einem IoT Netzwerk erkennt, aufgebaut als Fog Computing Architektur, anhand von signatur- und anomaliebasierten Erkennungsmethoden. Die signaturbasierte Methode greift auf eine Datenbank zu, welche IP Adressen enthält die als Angriffsursprung identifiziert sind. Die anomaliebasierte Methode klassifiziert anhand eines Extreme Gradient Booster (XGBoost) Algorithmus [26] zwischen auffälligem und unauffälligem Netzwerkverkehr. Der Netzwerkverkehr fließt zu Beginn durch ein signaturbasiertes IDS um die IP Adresse mit der Datenbank zu vergleichen. Sollte nichts gefunden werden, fließt der Verkehr durch ein anomaliebasiertes IDS und klassifiziert ihn entsprechend.

Das letzte Framework erläutern Kim et al. [27] in ihrer Arbeit zu einer SDN Umgebung mit integrierten Netzwerk Funktionen auf NFV, welche in den IoT-Geräten integriert sind. Das Konzept aus der Arbeit von Kim et al. beschreibt ein IoT Netzwerk, welches für jede einzelne Komponente eigene Sicherheitsrichtlinien vorgibt. Die Architektur des Netzwerks beginnt mit einer *Control Plane*, welches für das Management des Netzwerks verantwortlich ist. Die *Control Plane* besteht aus fünf Komponenten: *Datapath*, welches für die Kommunikation zwischen *Control Plane*, *Function Plane* und *Data Plane* (Verbindung zu den IoT-Geräten) verantwortlich ist. Die zweite Komponente ist der *Protocol Parser* der Nachrichten von der *Function-* und *Data Plane* empfängt und zum *Core Module* weiterleitet. Das *Core Module* verwaltet Netzwerkkomponenten, stellt Sicherheitsfunktionen zur Verfügung und erkennt Anomalien bei den Sicherheitsrichtlinien. Der *Event Manager* stellt Event basierte Kommunikation zwischen zur Verfügung. Die letzte Komponente sind *Applications* welche über ein *programmable interface* implementiert sind und sind für die Verarbeitung von den IoT-Geräten zuständig. Die *Function Plane* ist für die Sicherheit der IoT-Geräte zuständig, da diese durch begrenzte Leistung Anspruchsvolle Sicherheitsfunktionen nicht verarbeiten können. Diese Sicherheitsfunktionen werden über NFV Techniken ausgeführt damit die Funktionen virtualisiert zur Verfügung stehen. Die Kommunikation findet über das von Kim et al. vorgestellte SODA Protocol statt, damit zum Beispiel Richtlinien richtig verwaltet werden können. Wenn zum Beispiel neue Geräte dem Netzwerk hinzugefügt werden, dann werden anhand von Eventbasierten Prozessen diese auf zum Beispiel Sicherheitsbedenken geprüft. Sollte über das Netzwerk ein Angriff stattfinden, so wird das an den Richtlinien erkannt und diese dann angepasst. Bei der Implementierung wurde die *Control-* und *Function Plane* als SDN Controller implementiert, das *Data Plane* als Kommunikation zwischen dem Controller und den

IoT-Geräten. Zur vereinfachten Ressourcen Verwaltung nutzen Kim et al. NFV bei jedem IoT-Gerät.

## **Techniken zur Botnetzerkennung**

## 4 Ziel der Dissertation

Mit der Dissertation soll die Botnetzerkennung optimiert werden. Ziel ist es, ein Konzept zu erstellen, zusammen mit einer Implementierung. Das erste Ziel soll ein Konzept zur Botnetzerkennung werden, welches zu jeder Phase eines Botnetzes, mehrere Erkennungslevel integriert, die aus einer Kombinationen mehrerer Erkennungsmethoden besteht. Der aktuelle Stand aus der Forschung zu den einzelnen Erkennungsmethoden zeigt außerdem, dass die aktuellen Methoden sich eher auf die späteren Phasen eines Botnetzes fokussieren. Daher ist ein weiteres Ziel der Fokus auf die Entwicklung zur Erkennung der früheren Phasen der Botnetze. Das Ziel der Erkennungsmethoden ist es die Methoden herauszuarbeiten, welche die wenigsten false positives erzielen um diese dann miteinander zu kombinieren. Welche Methoden genau eingesetzt werden, soll mit dieser Dissertation zu Beginn herausgefunden werden. Zusätzlich sollen verschiedene Kombinationen aus Techniken eingesetzt werden, wie z.B. Flow- und Hostbasierte Techniken. Die Techniken sollen die Multi-Level Erkennung darstellen und zu jeder Phase separat ausgearbeitet werden, in deren Kombinationsmöglichkeiten. Ein weiteres Ziel ist, ob die Erkennung auch unbekannte Botnetze identifizieren kann indem Fallstudien durchgeführt werden. 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 2 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

## Literatur

- [1] Y. Xing, H. Shu, H. Zhao, D. Li, and L. Guo, “Survey on botnet detection techniques: Classification, methods, and evaluation,” *Mathematical Problems in Engineering*, vol. 2021, pp. 1–24, 2021.
- [2] M. Casenove and A. Miraglia, “Botnet over tor: The illusion of hiding,” in *6th International Conference on Cyber Conflict, CyCon 2014, Tallinn, Estonia, June 3-6, 2014* (P. Brangetto, M. Maybaum, and J. Stinissen, eds.), pp. 273–282, IEEE, 2014.
- [3] A. Kurt, E. Erdin, M. Cebe, K. Akkaya, and A. S. Uluagac, “Lnbot: A covert hybrid botnet on bitcoin lightning network for fun and profit,” in *Computer Security - ESORICS 2020 - 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14-18, 2020, Proceedings, Part II* (L. Chen, N. Li, K. Liang, and S. A. Schneider, eds.), vol. 12309 of *Lecture Notes in Computer Science*, pp. 734–755, Springer, 2020.
- [4] C. A. Schiller, J. Binkley, D. Harley, G. Evron, T. Bradley, C. Willems, and M. Cross, “Chapter 2 - botnets overview,” in *Botnets* (C. A. Schiller, J. Binkley, D. Harley, G. Evron, T. Bradley, C. Willems, and M. Cross, eds.), pp. 29–75, Burlington: Syngress, 2007.
- [5] M. Wazzan, D. Algazzawi, O. Bamasaq, A. A. Albeshri, and L. Cheng, “Internet of things botnet detection approaches: Analysis and recommendations for future research,” *Applied Sciences*, 2021. <https://pdfs.semanticscholar.org/8dc2/d3f4e1120ebd9170dc68d72b17a96189729d.pdf>, last accessed 2022-11-29.
- [6] P. Beltrán-García, E. Aguirre-Anaya, P. J. Escamilla-Ambrosio, and R. Acosta-Bermejo, “Iot botnets,” in *Telematics and Computing* (M. F. Mata-Rivera, R. Zagal-Flores, and C. Barriá-Huidobro, eds.), (Cham), pp. 247–257, Springer International Publishing, 2019.
- [7] H. Alzahrani, M. Abulkhair, and E. Alkayal, “A multi-class neural network model for rapid detection of iot botnet attacks,” *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 7, 2020.
- [8] N. Vlajic and D. Zhou, “Iot as a land of opportunity for ddos hackers,” *Computer*, vol. 51, no. 7, pp. 26–34, 2018.
- [9] H.-T. Nguyen, Q.-D. Ngo, D.-H. Nguyen, and V.-H. Le, “Psi-rooted subgraph: A novel feature for iot botnet detection using classifier algorithms,” *ICT Express*, vol. 6, no. 2, pp. 128–138, 2020.
- [10] M. Singh, M. Singh, and S. Kaur, “Issues and challenges in DNS based botnet detection: A survey,” *Comput. Secur.*, vol. 86, pp. 28–52, 2019.



- [11] A. M. Manasrah, T. Khmour, and R. Freehat, "Dga-based botnets detection using DNS traffic mining," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 5, pp. 2045–2061, 2022.
- [12] M. Stevanovic and J. M. Pedersen, "Detecting bots using multi-level traffic analysis," *Int. J. Cyber Situational Aware.*, vol. 1, no. 1, pp. 182–209, 2016.
- [13] C. Kolias, G. Kambourakis, A. Stavrou, and J. M. Voas, "Ddos in the iot: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.
- [14] S. Herwig, K. Harvey, G. Hughey, R. Roberts, and D. Levin, "Measurement and analysis of hajime, a peer-to-peer iot botnet," in *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*, The Internet Society, 2019.
- [15] A. Author(s), "Analyzing the propagation of iot botnets from dns leakage," 2017.
- [16] P. Maymounkov and D. Mazières, "Kademlia: A peer-to-peer information system based on the XOR metric," in *Peer-to-Peer Systems, First International Workshop, IPTPS 2002, Cambridge, MA, USA, March 7-8, 2002, Revised Papers* (P. Druschel, M. F. Kaashoek, and A. I. T. Rowstron, eds.), vol. 2429 of *Lecture Notes in Computer Science*, pp. 53–65, Springer, 2002.
- [17] F. S. Alrayes, M. Maray, A. Gaddah, A. Yafoz, R. Alsini, O. Alghushairy, H. Mohsen, and A. Motwakel, "Modeling of botnet detection using barnacles mating optimizer with machine learning model for internet of things environment," *Electronics*, 2022.
- [18] M. M. Alani, "Botstop : Packet-based efficient and explainable iot botnet detection using machine learning," *Comput. Commun.*, vol. 193, pp. 53–62, 2022.
- [19] G. T. Habtamu and A. Y. Kassahun, "A systematic review of botnet detection system using deep learning and machine learning approaches," *SSRN Electronic Journal*, 2022. <https://doi.org/10.53730/ijhs.v6nS2.6551>.
- [20] Z. Zha, A. Wang, Y. Guo, D. Montgomery, and S. Chen, "Botsifter: An sdn-based online bot detection framework in data centers," in *7th IEEE Conference on Communications and Network Security, CNS 2019, Washington, DC, USA, June 10-12, 2019*, pp. 142–150, IEEE, 2019.
- [21] W. G. Negera, F. Schwenker, T. G. Debelee, H. M. Melaku, and Y. M. Ayano, "Review of botnet attack detection in sdn-enabled iot using machine learning," *Sensors*, vol. 22, no. 24, p. 9837, 2022.
- [22] M. Gromov, D. Arnold, and J. Saniie, "Edge computing for real time botnet propagation detection," in *2022 IEEE International Conference and Expo on Real Time Communications at IIT (RTC)*, pp. 13–16, IEEE, 2022.

- [23] G. Sagirlar, B. Carminati, and E. Ferrari, “Autobotcatcher: Blockchain-based P2P botnet detection for the internet of things,” *CoRR*, vol. abs/1809.10775, 2018.
- [24] M. M. Salim, K. C. Alowonou, N. Tojimurotov, H. Park, and J. H. Park, “A blockchain-enabled secure digital twin framework for early botnet detection in iiot environment,” *Sensors*, vol. 22, no. 16, p. 6133, 2022.
- [25] M. A. Lawal, R. A. Shaikh, and S. R. Hassan, “An anomaly mitigation framework for iot using fog computing,” *Electronics*, 2020.
- [26] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” *CoRR*, vol. abs/1603.02754, 2016. <http://arxiv.org/abs/1603.02754>.
- [27] Y. Kim, J. Nam, T. Park, S. Scott-Hayward, and S. Shin, “SODA: A software-defined security framework for iot environments,” *Comput. Networks*, vol. 163, 2019.