Multi-Level Erkennung von IoT-Botnetzen in frühen Botnetzphasen

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, 2. März 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) haben in der modernen Welt sehr stark an Bedeutung gewonnen. Gerade IoT-Geräte die durch Botnetze wie Mirai und Hajime infiziert werden können sind davon stark betroffen [1]. 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 Verbreitungs-Phase erkennen?

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

Q3: Welche Kombinationen von Methoden zur Botnetzerkennung führen zu einer Optimierung der Erkennung?

Q4: Welche Methoden bringen zu den jeweiligen Phasen passenden Ergebnisse?

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 Erkennungsmethoden am besten kombiniert werden können. Die Frage RQ4 soll aus den vorherigen Ergebnissen klären, welche Kombination von Methoden zu welchen Phasen eingesetzt werden können.

In 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, wie die Dissertation aufgebaut sein soll.

2 Theoretischer Hintergrund

Die Dissertation soll im Gebiet der Malwareforschung eingegliedert werden. Im speziellen soll es dabei um die Erkennung von Botnetzen in IoT-Netzwerken 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 [2]. Die Botnetze nutzen Zero-day Schwachstellen, Peer-2-Peer Netze, Phishing Angriffe, Anonyme Netzwerke, Blockchain Netzwerke und Stromnetze zur Verbreitung ihrer Verwendungszwecke [3, 4]. 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 [5], zum Beispiel, über das Internet Relay Chat (IRC)-Protokoll. Der Botmaster kann aber auch ohne den C&C-Server Nachrichten an die Bots leiten, wie später die verschiedenen Architekturen zeigen.
Zum Aufbau und Angriff durchlaufen Botnetze drei Phasen wie Wazzan et al. [6] 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 wird auf Basis der Architektur des infizierten Geräts ausgeführt. Um auf dem Gerät Malware zu verhindern die nicht vom Bot 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 unter anderem Angriffe wie Distributed Denial of Service (DDoS), krypto mining und spam Angriffe aus. Die erläuterten Phasen arbeiten auch Studien wie [7, 8, 9, 10] aus. Dennoch lassen sich die drei Phasen in weitere Phasen aufteilen und können je nach Verwendungszweck des Botnetzes varriieren [6].
Botnetze können nach verschiedenen Architekturen aufgebaut sein: zentralisiert, Peer-to-Peer(P2P) [11] und hybride Architekturen [12]. Bei zentralen Botnetzen kommuniziert der Botmaster über einen zentralen C&C-Server mit den Bots, während bei den dezentralen P2P Architekturen der Botmaster die Befehle über das gesamte P2P-Overlay-Netzwerk verteilt und hybride Architekturen kombinieren die beiden vorangegangen Architekturen. IoT Botnetze fallen dabei nicht in eine der drei Architekturen. Für IoT Botnetze sind alle Architekturen möglich [13].
Nach der Erläuterung wie Botnetze funktionieren und welche es gibt, ist nun zu klären, wie der Prozess eines Botnetzes erkennbar ist um IoT-Geräte entsprechend zu schützen. Nach Xing et al. [2] 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-Fragmente durch das Honeypot trapping was eine hohe Genauigkeit von bereits bekannten Botnetzen ermöglicht. Die Honeypot Methoden können dafür verschlüsselten Netzwerkverkehr nur schlecht erkennen und keine unbekannten Botnetze. Bots die eigene Funktionen zur Umgehung von Honeypots besitzen,

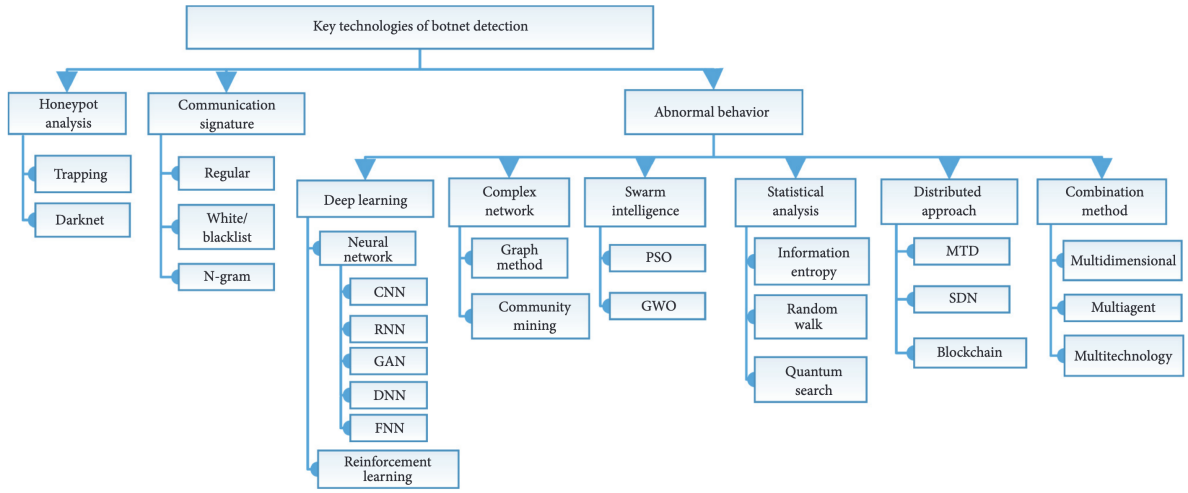

Abbildung 1: Klassifizierte Erkennungsmethoden von Botnetzen (übernommen von [2]).

können ebenfalls nicht von der *Honeypot Analyse* erfasst werden. Daher ist diese Technik ausschließlich sinnvoll für die Erkennung bereits bekannter Botnetze. Dennoch sollte diese Methode in Kombination mit anderen Methoden eingesetzt werden um auch unbekannte Botnetze zu erkennen [2]. Eine weitere Methode ist die Erkennung von *Signaturen*. 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 Methoden zur Verschleierung von Code nutzen. Bei den Methoden durch Erkennung vom *abnormalen Verhalten*, ist die Idee, Hostverhalten- oder Netzwerkverkehr-Anomalien zwischen gutartig und bösartig zu klassifizieren. Der Vorteil der letzten Methode ist die Möglichkeit zur Erkennung von unbekannten Botnetzen. Laut Xing et al. sollten diese Methoden des *abnormalen Verhaltens* in Kombination mit den anderen Methoden genutzt werden um passendere Ergebnisse zu erzielen. Zusammengefasst hat jeder dieser Methoden Nachteile die über eine Kombination mit den jeweils anderen ausgebessert werden können.
Aus der Erkenntnis zur Verwendung mehrerer Techniken soll sich die Dissertation mit einer Kombination aus mehreren Methoden beschäftigen. Konkret bedeutet das die Verwendung einer Hybriden Erkennung durch mehrere Erkennungsmethoden während jeder Phase eines Botnetzes wie Wazzan et al. [6] vorschlagen. Die Umsetzung über mehrere Erkennungslevel erklären Stevanovic und Pedersen [14]. Dabei führen Stevanovic und Pedersen eine Analyse des Netzwerkverkehrs durch, um die Kommunikation mit einem C&C-Server sowie den Angriffsverkehr anhand von TCP, UDP und DNS separat zu klassifizieren. Die Erkennung erfolgt mit supervised Machine Learning, um bestimmte Muster der Botnetzkommunikation zu identifizieren auf Basis von bereits bekannten Netzwerküberwachungsdaten. Die komplette Architektur 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.

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 [14]).

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 Botnetzerkennung einen Random Forest Classifier, um dann über eine Client Entitäten Analyse einen Report zu erstellen über infizierte Geräte. Stevanovic und Pedersen zeigen mit der Arbeit, dass es möglich ist, mehrere Level zur Erkennung von Botnetzen zu implementieren. Das daraus resultierende Ziel erklärt Kapitel 4 ausführlicher.

Angriffsmöglichkeiten zum Testen

Um den Prozess der Botnetzerkennung durch Fallstudien zu testen, sollen in der Dissertation verschiedene Botnetze ausgeführt werden. Dabei sollen bereits bekannte Botnetze ausgeführt werden und zusätzlich soll ein Botnetz implementiert werden um herauszufinden, ob das System auch neue, unbekannte Botnetze erkennt. Das Kriterium zur Auswahl sind Botnetze die den Fokus auf IoT Geräte legen. Kolias et al. [15] weisen in ihrer Arbeit auf Mirai und Hajime hin, welche in der Dissertation für die Botnetzerkennung ausgeführt

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 [14]).

werden sollen. 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. Der *loader* übernimmt die Kommunikation mit neuen infizierten Geräten und verteilt 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. 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 Programme die den Secure

Shell (SSH), als auch Telnet Port blockieren. 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 [16, 17]. BitTorrent nutzt das Kademlia Protokoll [18] und zusätzlich zur direkten Peer-to-Peer Kommunikation nutzt Hajime zusätzlich das uTorrent Transport Protocol. Für weitere technische Erläuterungen zu Hajime, analysieren [16] die Phasen des Botnetzes.

3 Stand der Forschung

Methoden zur Erkennung von Botnetzen

Xing et al. [2] stellen eine zusammengefasste Unterteilung von Methoden zur Botnetzerkennung vor, wobei sie nicht konkreter auf die Methoden eingehen. Speziell auf Botnetze im IoT Gebiet fassen Wazzan et al. [6] Methoden anhand von Studien zusammen, die zur Erkennung eingesetzt werden. Für den aktuellen Stand der Forschung nutzt dieses Exposé deshalb die ausgearbeiteten Zusammenfassungen von [6] um einen Ansatz zu finden wie die Botnetzerkennung stattfinden könnte. 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 [19, 20, 21, 22] zeigen. Wazzan et al. schlagen die Kombination mehrerer Erkennungsmethoden vor, womit sich die Dissertation auch beschäftigen soll. Nach den genannten Studien schlagen Wazzan et al. folgende Erkennungsmethoden vor: Software Defined Networks (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. [23] beschreiben eine Bibliothek, welche eine Botnetzerkennung anhand eines SDNs durchführt. Bei dem Konzept des Frameworks handelt es sich darum, das Botnetz und C&C-Kanal Aktivitäten zu identifizieren, damit die Robustheit der Erkennung erhöht wird. Zum Aufbau gehört ein SDN Controller, welcher ein Modell des ML enthält, dass über Erkennung und Überwachung trainiert wird. Die Überwachung des Netzwerks findet über Software Switches statt, welche vollen Zugang zum Netzwerkverkehr von Virtuellen Maschinen (VM) haben. Das 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 sind speziell Peer-to-Peer Verkehr, HTTP und IRC Protokolle im Fokus. Das Modell ist ein Neuronales Netzwerk (NN), welches kompromittierte 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 zum Beispiel nicht unbedingt mit dem C&C-Server kommuniziert. An dieser Stelle wird der Host dennoch dauerhaft weiter beobachtet. Der aktuelle Stand der Forschung zeigt hier, dass die Erkennung nicht während der Scan-Phase durchführt 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. [24] 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 [25] in ihrer Arbeit zeigen. Bei dieser Erkennung verwenden Gromov et al. in einer Edge Computing Architektur 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 Blockchainbasierten 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. [26] nutzen diese Methode indem sie innerhalb der Blockchain eine Peer-to-Peer Botnetzerkennung implementieren. Salim et al. [27] erläutern ein Framework, welches Digital Twins und Packet Auditor in einer Blockchain zur Erkennung einsetzen.

Neben Blockchains wird auch Fog Computing zur Botnetzerkennung eingesetzt. [28] 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 [29] 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. [30] 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 zur Verfügung. Die letzte Komponente sind *Applications* welche über ein *programmable interface* implementiert sind und für die Verarbeitung von den IoT-Geräten zuständig sind. 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 Ressourcenverwaltung nutzen Kim et al. NFV bei jedem IoT-Gerät.

Aus den Arbeiten der aktuellen Forschung geht hervor, dass bei der Verwendung von ML die eingesetzten Algorithmen immer zu Ergebnissen führen bei denen das Modell eine Genauigkeit von über 90% erreicht. Dabei gehen die Arbeiten nicht weiter darauf ein warum der entsprechende Algorithmus verwendet wird und welche Vorteile dieser bringt. Zudem erkennen die Arbeiten die Botnetze immer zu den späteren Phasen eines Botnetzes womit die Arbeiten keine Prävention von Botnetzen durchführen. Desweiteren nutzen die Botnetzerkennungen nur einzelne Methoden und kombinieren diese nicht miteinander, um zum Beispiel die Nachteile einer Methode mit einer weiteren auszugleichen. Wenn die Netzwerkarchitekturen wie zum Beispiel SDNs mit in die Erkennung von IoT Botnetzen mit einbezogen werden, zeigt die aktuelle Forschung, dass dies zu einer besseren Leistung der Botnetzerkennung führt, um so zum Beispiel Latenzen zu verringern. Aus Kapitel 2 sowie dem aktuellen Stand der Forschung geht hervor, dass [2] und [6] als mögliche Ansätze für die Botnetzerkennung verwendet werden können, da diese Arbeiten konkretere Methoden und Vorschläge machen um die Botnetzerkennung zu optimieren.

Multi-Level Botnetzerkennung

Neben der Forschung zur Botnetzerkennung anhand einzelner Methoden soll dieses Kapitel den aktuellen Stand der Forschung zur Botnetzerkennung anhand kombinierter Methoden vorstellen. Rahal et al. [31] beschreiben ein Framework, welches auf Netzwerkebene und Aktiviäten in Fahrzeugen beobachtet. Eine weitere Studie untersucht die Kombination eines Algorithmus um Merkmale in einem Suchraum zu finden mit einem Modell des ML zur Klassifizierung von Bedrohungen in einem Flying ad hoc network [32]. Almutairi et al. [33] implementieren eine Botnetzerkennung auf Basis einer Hostbasierten und Netzwerkbasierten Erkennung. Der Fokus bei der Erkennung liegt dabei auf den frühen Phasen eines Botnetzes, was genauer bedeutet, die Bots bei der Verbreitung zu erkennen. Die hybride Botnetzerkennung wird dazu genutzt, um *abnormales Verhalten* zu erkennen und falls über die Netzwerkanalyse nichts erkannt wird, kann die hostbasierte Analyse zusätzlich über weitere Regeln unterstützen. Um Schwachstellen zu sammeln, setzen die Autoren einen Honeypot ein, welcher über das Internet Angreifer dazu bringen soll, dass System anzugreifen. Ein Controller überwacht das Netzwerk und steuert VMs. Zudem ist eine Datenbank für Malware Signaturen eingesetzt. Um das Verhalten von Bots zu beobachten, wird in einer VM ein Analysewerkzeug genutzt. Eine weitere VM führt eine dynamische Analyse durch. In der ersten Phase sammelt das System Malware Signaturen, welche in einer zweiten Phase klassifiziert werden an von ML. Almutairi et al. erläutern, dass die Netzwerkanalyse wichtig ist, da die Bots über das Internet kommunizieren um

zum Beispiel Nachrichten an den C&C- Server zu schicken.

Auch bei der Multi-Level (auch hybriden) Botnetzerkennung liegt der Fokus auf einer Phase und auch überwiegend auf die späteren Phasen. Zusätzlich legen die Arbeiten auch in den meisten Fällen separat den Fokus auf C&C-Server Erkennung oder der Erkennung von Bots.

4 Ziel der Dissertation

Mit der Dissertation soll die Botnetzerkennung optimiert werden. Ziel ist es ein Konzept zu erstellen, mit einer darauffolgenden Implementierung. Das erste Ziel soll die Ausarbeitung und Zusammenfassung verschiedener aktueller Methoden sein, sowie deren Vor- und Nachteile. So soll sich herausarbeiten, welche Methoden miteinander kombiniert werden können. Das darauf folgende Ziel ist die Erstellung eines Konzepts zur Botnetzerkennung, welches zu den früheren Phasen vor der Ausbreitung des Botnetzes, mehrere Erkennungslevel integriert, die aus einer Kombinationen mehrerer Erkennungsmethoden besteht. Die verschiedenen Methoden sollen zu jeder Phase separat ausgearbeitet werden. Aus dem aktuellen Stand der Forschung zu den einzelnen Erkennungsmethoden geht hervor, dass die aktuellen Methoden sich eher auf die späteren Phasen eines Botnetzes fokussieren bei denen sich das Botnetz bereits in einem Netzwerk ausgebreitet hat. Daher ist ein weiteres Ziel, der Fokus auf die Entwicklung zur Erkennung der früheren Phasen. Ein weiteres Ziel ist, herauszufinden, ob die Erkennung auch unbekannte Botnetze identifizieren kann indem Fallstudien durchgeführt werden.
Gesamtziel der Dissertation ist die Implementierung einer Technik zur Verbesserung der Botnetzerkennung woraus ein Datensatz abgeleitet werden soll. 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 soll eine Ausarbeitung und Zusammenfassung von Methoden zur Botnetzerkennung durchgeführt werden. Anhand der Zusammenfassung soll erkennbar werden, welche Methoden die bestmöglichen Ergebnisse liefern und welche Kombinationen von Methoden am plausibelsten sind. Daraus soll ein Konzept gebildet werden, welches Hybride Erkennungsmethoden für die frühen Phasen eines Botnetzes in der Theorie beschreibt. Anschließend soll mit einem Laborexperiment das Konzept implementiert werden. Das Laborexperiment soll ein lokales Netzwerk mit gängigen Smart Home Geräten darstellen, die in einem privaten Haushalt verwendet werden. Um die Erkennung eines Botnetzes zu testen sollen verschiedene typische IoT-Geräte aufgebaut werden, die von Botnetzen wie zum Beispiel Mirai angegriffen werden. Dazu ist es nötig, dass entsprechende Geräte ausgewählt werden, die von bekannten Botnetzen infiziert werden können.

In einer Fallstudie ist dann vorgesehen, Botnetze auszuführen, um entsprechende Daten zu sammeln, damit eine aussagekräftige Auswertung zu plausiblen Ergebnissen führt. Um die Ergebnisse aus der Fallstudie überprüfen zu können, sollen diese mit Ergebnissen aus anderen vergleichbaren Arbeiten in Relation gesetzt werden.
Wie schon in 2 erläutert, sollen unter anderem Mirai und Hajime 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 ungefähr 4 Jahre und 3 Monate.

Zeit	Vorgehen
März/April 2023	Literarur Recherche zur hybriden Botnetzerkennung durchführen
August 2023	Verwandte Arbeiten und Konzept in der Dissertation erarbeiten
Januar 2025	Implementierung der Botnetzerkennung und des Smart Homes
März 2025	Fallstudie durchführen
Juli 2025	Ergebnisse zusammenfassen und Datensatz zusammenstellen
Februar 2026	Einleitung und Ende der Dissertation schreiben
April 2026	Kontrolle der Dissertation
Juli 2026	Abgabe

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] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou, "Understanding the mirai botnet," in *26th USENIX Security Symposium, USENIX Security 2017, Vancouver, BC, Canada, August 16-18, 2017* (E. Kirda and T. Ristenpart, eds.), pp. 1093–1110, USENIX Association, 2017.

[2] 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. Semantic Scholar, last accessed 2023-02-17.

[3] 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. DOI.

[4] 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. DOI.

[5] 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. Science Direct.

[6] 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. Semantic Scholar, last accessed 2022-11-29.

[7] 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. Barría-Huidobro, eds.), (Cham), pp. 247–257, Springer International Publishing, 2019. Springer.

[8] 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. DOI.

[9] N. Vlajic and D. Zhou, "Iot as a land of opportunity for ddos hackers," *Computer*, vol. 51, no. 7, pp. 26–34, 2018. DOI.

[10] 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. Science Direct.

[11] S. N. T. Vu, M. Stege, P. I. El-Habr, J. Bang, and N. Dragoni, "A survey on botnets: Incentives, evolution, detection and current trends," *Future Internet*, vol. 13, no. 8, p. 198, 2021. DOI.

[12] I. Apostol, A. Tica, and V. Patriciu, "Design and implementation of a novel hybrid botnet," in *14th International Conference on Electronics, Computers and Artificial Intelligence, ECAI 2022, Ploiesti, Romania, June 30 - July 1, 2022*, pp. 1–6, IEEE, 2022. DOI.

[13] L. McNulty and V. G. Vassilakis, "Iot botnets: Characteristics, exploits, attack capabilities, and targets," in *13th International Symposium on Communication Systems, Networks and Digital Signal Processing, CSNDSP 2022, Porto, Portugal, July 20-22, 2022*, pp. 350–355, IEEE, 2022. DOI.

[14] 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. DOI.

[15] 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. DOI.

[16] 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. Link.

[17] Anonymous-Author(s), "Analyzing the propagation of iot botnets from dns leakage," 2017. University of Maryland.

[18] 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. DOI.

[19] 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. MDPI.

[20] M. M. Alani, "Botstop : Packet-based efficient and explainable iot botnet detection using machine learning," *Comput. Commun.*, vol. 193, pp. 53–62, 2022. DOI.

[21] 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. Science Scholar.

[22] S. D, A. P, C. S. Barboza, B. S, and C. B. Katoti, "A report on botnet detection techniques for intrusion detection systems," *International Journal for Research in Applied Science and Engineering Technology*, 2022. IJRASET.

[23] 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. DOI.

[24] 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. DOI.

[25] 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. IEEE.

[26] G. Sagirlar, B. Carminati, and E. Ferrari, "Autobotcatcher: Blockchain-based P2P botnet detection for the internet of things," *CoRR*, vol. abs/1809.10775, 2018. arXiv.

[27] 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. DOI.

[28] L. Muhammad Aminu, R. Shaikh, and R. Hassan, "An anomaly mitigation framework for iot using fog computing," *Electronics*, vol. 9, p. 1565, 09 2020. Research Gate.

[29] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," *CoRR*, vol. abs/1603.02754, 2016. arXiv.

[30] 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. DOI.

[31] R. Rahal, A. A. Korba, N. Ghoualmi-Zine, Y. Challal, and Y. Ghamri-Doudane, "Antibotv: A multilevel behaviour-based framework for botnets detection in vehicular networks," *J. Netw. Syst. Manag.*, vol. 30, no. 1, p. 15, 2022. DOI.

[32] N. F. Abdulsattar, F. Abedi, H. M. A. Ghanimi, S. Kumar, A. H. Abbas, A. S. Abosinnee, A. Alkhayyat, M. H. Hassan, and F. H. Abbas, "Botnet detection employing a dilated convolutional autoencoder classifier with the aid of hybrid shark and bear smell optimization algorithm-based feature selection in fanets," *Big Data Cogn. Comput.*, vol. 6, no. 4, p. 112, 2022.

[33] S. Almutairi, S. Mahfoudh, S. Almutairi, and J. S. Alowibdi, "Hybrid botnet detection based on host and network analysis," *J. Comput. Networks Commun.*, vol. 2020, pp. 9024726:1–9024726:16, 2020.