Data-Driven Approaches for Detecting Malware-Infected IoT Devices and Characterizing Their Unsolicited Behaviors by Leveraging Passive Internet Measurements

Sadegh Torabi

A Thesis

in

The Concordia Institute

for

Information and Systems Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy (Information and Systems Engineering) at

Concordia University

Montréal, Québec, Canada

April 2021

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Mr. Sadegh Torabi**

Entitled: **Data-Driven Approaches for Detecting Malware-Infected IoT Devices and Characterizing Their Unsolicited Behaviors by Leveraging Passive Internet Measurements**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Information and Systems Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Chair
Dr. Constantinos Constantinides

_____ External Examiner
Dr. Nur Zincir-Heywood

_____ External to Program
Dr. Dongyu Qiu

_____ Examiner
Dr. Lingyu Wang

_____ Examiner
Dr. Amr Youssef

_____ Thesis Co-Supervisor
Dr. Chadi Assi

_____ Thesis Co-Supervisor
Dr. Elias Bou-Harb

Approved by _____
Dr. Abdessamad Ben Hamza, Director
Concordia Institute for Information and Systems Engineering

April 6, 2021

_____
Dr. Mourad Debbabi, Dean,
Gina Cody School of Engineering and Computer Science

Abstract

Data-Driven Approaches for Detecting Malware-Infected IoT Devices and Characterizing Their Unsolicited Behaviors by Leveraging Passive Internet Measurements

Sadegh Torabi, Ph.D.

Concordia University, 2021

Despite the benefits of Internet of Things (IoT) devices, the insecurity of IoT and their deployment nature have turned them into attractive targets for adversaries, which contributed to the rise of IoT-tailored malware as a major threat to the Internet ecosystem. In this thesis, we address the threats associated with the emerging IoT malware, which utilize exploited devices to perform large-scale cyber attacks (e.g., DDoS). To mitigate such threat, there is a need to possess an Internet perspective of the deployed IoT devices while building a better understanding about the behavioral characteristic of malware-infected devices, which is challenging due to the lack of empirical data and knowledge about the deployed IoT devices and their behavioral characteristics.

To address these challenges, in this thesis, we leverage passive Internet measurements and IoT device information to detect exploited IoT devices and investigate their generated traffic at the network telescope (darknet). We aim at proposing data-driven approaches for effective and near real-time IoT threat detection and characterization. Additionally, we leverage a specialized IoT Honeypot to analyze a large corpus of real IoT malware binary executable. We aim at building a better understanding about the current state of IoT malware while addressing the problems of IoT malware classification and family attribution. To this end, we perform the following to achieve our objectives:

First, we address the lack of empirical data and knowledge about IoT devices and their activities. To this end, we leverage an online IoT search engine (e.g., Shodan.io) to obtain publicly available device information in the realms of consumer and cyber-physical system (CPS), while

utilizing passive network measurements collected at a large-scale network telescope (CAIDA), to infer compromised devices and their unsolicited activities. Indeed, we were among the first to report experimental results on detecting compromised IoT devices and their behavioral characteristics in the wild, while demonstrating their active involvement in large-scale malware-generated malicious activities such as Internet scanning. Additionally, we leverage the IoT-generated backscatter traffic towards the network telescope to shed light on IoT devices that were victims of intensive Denial of Service (DoS) attacks.

Second, given the highly orchestrated nature of IoT-driven cyber-attacks, we focus on the analysis of IoT-generated scanning activities to detect and characterize scanning campaigns generated by IoT botnets. To this end, we aggregate IoT-generated traffic and performing association rules mining to infer campaigns through common scanning objectives represented by targeted destination ports. Further, we leverage behavioural characteristics and aggregated flow features to correlate IoT devices using DBSCAN clustering algorithm. Indeed, our findings shed light on compromised IoT devices, which tend to operate within well coordinated IoT botnets.

Third, considering the huge number of IoT devices and the magnitude of their malicious scanning traffic, we focus on addressing the operational challenges to automate large-scale campaign detection and analysis while generating threat intelligence in a timely manner. To this end, we leverage big data analytic frameworks such as Apache Spark to develop a scalable system for automated detection of infected IoT devices and characterization of their scanning activities using our proposed approach. Our evaluation results with over 4TB of IoT traffic demonstrated the effectiveness of the system to infer scanning campaigns generated by IoT botnets. Moreover, we demonstrate the feasibility of the implemented system/framework as a platform for implementing further supporting applications, which leverage passive Internet measurement for characterizing IoT traffic and generating IoT-related threat intelligence.

Fourth, we take first steps towards mitigating threats associated with the rise of IoT malware by creating a better understanding about the characteristics and inter-relations of IoT malware. To this end, we analyze about 70,000 IoT malware binaries obtained by a specialized IoT honeypot in the past two years. We investigate the distribution of IoT malware across known families, while

exploring their detection timeline and persistent. Moreover, while we shed light on the effectiveness of IoT honeypots in detecting new/unknown malware samples, we utilize static and dynamic malware analysis techniques to uncover adversarial infrastructure and investigate functional similarities. Indeed, our findings enable unknown malware labeling/attribution while identifying new IoT malware variants. Additionally, we collect malware-generated scanning traffic (whenever available) to explore behavioral characteristics and associated threats/vulnerabilities.

We conclude this thesis by discussing research gaps that pave the way for future work.

Dedication

This thesis is dedicated to the memory of my father, Nemat Torabi. I miss him every day and I will never forget his kindness and unconditional support.

To my mother, Hajar Mahwan. She has been a source of inspiration, motivation, and strength during moments of despair and discouragement.

To my wife, Zainab. Without her unconditional love and ever-lasting support nothing of this would have been possible.

To all my family, especially my older brother, Ebrahim Torabi. I will never forget your empathy and support.

Acknowledgments

First and foremost, all my gratitude goes to my parents and my wife for their continuous support and encouragement that allowed me to finish this work. Additionally, I would like to thank all my family members, especially my older brother Mr. Ebrahim Torabi, for his limitless support.

I would like to thank my esteemed supervisor Prof. Chadi Assi for his invaluable supervision, support and tutelage during the course of my PhD degree. My gratitude extends to the Faculty of Engineering and Computer Science for the funding opportunity to undertake my studies at the Concordia Institute for Information Systems Engineering (CIISE). I would like to also thank the Natural Sciences and Engineering Research Council of Canada (NSERC) and Concordia University for their financial support. Additionally, I would like to express gratitude to my co-supervisor Dr. Elias Bou-Harb for his treasured support which was really influential in shaping my experiment methods and critiquing my results. My sincere thanks must also go to the members of my thesis advisory and exam committee who generously gave their time to offer me valuable comments toward improving my work.

I would like to also thank Prof. Mourad Debbabi for sharing his expertise with me and providing guidance and support throughout my academic studies. Additionally, I would like to express gratitude to my fellow researchers Dr. Amine Boukhtouta and Dr. ElMouatez Billah Karbab for their collaboration during my research.

Finally, I would like to thank my friends, lab mates, colleagues and research team for a cherished time spent together in the lab, and in social settings.

Contents

List of Figures **xiii**

List of Tables **xvii**

1 Introduction **1**

1.1 Problem Scope and Motivation 1

1.2 Objectives and Research Questions 2

1.3 Methodology 3

1.4 Contributions 5

1.4.1 Inferring, Characterizing, and Investigating Internet-Scale Malicious IoT Device Activities: A Network Telescope Perspective 5

1.4.2 Inferring and Investigating IoT-Generated Scanning Campaigns Targeting A Large Network Telescope 5

1.4.3 A Scalable Platform for Enabling the Forensic Investigation of Exploited IoT Devices and their Generated Unsolicited Activities 6

1.4.4 A Strings-Based Similarity Analysis Approach for Large-Scale IoT Malware Analysis, Characterization, and Family Attribution 7

1.5 Thesis Organization 8

2 Background, Related Work, and Leveraged Datasets **9**

2.1 Background 9

2.1.1 Network Telescope 9

2.1.2 IoT-Driven Cyber Attacks . . . 11
2.1.3 Malware Analysis Techniques . . . 12
2.2 Related Work . . . 15
2.2.1 IoT Security and Vulnerability Analysis . . . 15
2.2.2 IoT Data Capturing Initiatives . . . 16
2.2.3 Passive Internet Measurements . . . 16
2.2.4 IoT Malware Data Collection and Analysis . . . 18
2.3 Leveraged Datasets . . . 20

3 Inferring, Characterizing, and Investigating Internet-Scale Malicious IoT Device Activities: A Network Telescope Perspective 22

3.1 Overview . . . 22
3.2 Contributions . . . 24
3.3 Identifying Unsolicited Internet-Scale IoT Devices . . . 25
3.3.1 Obtained Data . . . 25
3.3.2 Inferring and Characterizing Unsolicited IoT Devices . . . 28
3.4 Characterizing Unsolicited Traffic From Internet-Scale IoT Devices . . . 32
3.4.1 Unsolicited UDP Traffic . . . 32
3.4.2 Unsolicited Backscatter Traffic . . . 36
3.4.3 Unsolicited Scanning Traffic . . . 39
3.5 Analyzing the Maliciousness of Unsolicited Internet-Scale IoT Devices . . . 45
3.5.1 IoT Illicit Activities . . . 45
3.5.2 IoT-Centric Malware Families . . . 46
3.6 Summary and Concluding Remarks . . . 48

4 Investigating IoT-Generated Scanning Campaigns Targeting A Large Network Telescope 50

4.1 Overview . . . 50
4.2 Contributions . . . 52
4.3 Approach . . . 53

4.3.1 Data Collection . . . 53
4.3.2 Data Processing . . . 55
4.3.3 Preliminary Analysis (Initial Data Set) . . . 56
4.3.4 Limitations . . . 57
4.4 IoT-Generated Scanning Campaigns . . . 59
4.4.1 Scanning Objective(s) . . . 59
4.4.2 Campaign Detection . . . 67
4.4.3 Results Summary . . . 76
4.4.4 IoT Malware Attribution . . . 77
4.5 Campaign Persistence and Evolution . . . 80
4.5.1 Scanning Activities . . . 81
4.5.2 Persistence . . . 83
4.5.3 Evolution . . . 85
4.6 Characterizing IoT-Generated Internet Scanning Activities Using Their Packet Inter-Arrival Times . . . 86
4.6.1 Proposed Model . . . 87
4.6.2 Empirical Analysis of Packet Inter-Arrival Times . . . 88
4.7 Summary and Concluding Remarks . . . 92

5 A Scalable Platform for Enabling the Forensic Investigation of Exploited IoT Devices and their Generated Unsolicited Activities 95
5.1 Overview . . . 95
5.2 Contributions . . . 96
5.3 Design and Implementation . . . 97
5.3.1 IoT Data Collection Module . . . 97
5.3.2 Darknet Data Collection Module . . . 98
5.3.3 IoT Traffic Analysis Module . . . 99
5.3.4 IoT Threat Repository . . . 102
5.4 Experimental Results and Evaluation . . . 102

5.4.1 Data Collection and Sampling . . . 103
5.4.2 Results (Applications) . . . 103
5.4.3 Performance Evaluation . . . 115
5.5 Summary and Concluding Remarks . . . 120

6 A Strings-Based Similarity Analysis Approach for Large-Scale IoT Malware Analysis, Characterization, and Family Attribution 121

6.1 Overview . . . 121
6.2 Contributions . . . 123
6.3 Approach . . . 124
6.3.1 IoT Malware Data Collection and Labeling . . . 125
6.3.2 Extracting IoT Malware Strings . . . 126
6.3.3 Packed/Obfuscated Malware Binaries . . . 126
6.3.4 Strings-Based Similarity Analysis . . . 127
6.3.5 Limitations . . . 130
6.4 IoT Malware Data . . . 131
6.4.1 Identified IoT Malware Families . . . 131
6.4.2 IoT Malware Detection Timeline . . . 133
6.4.3 IoT Malware Activity Duration (Threat Persistence) . . . 135
6.5 Static Malware Analysis Results . . . 136
6.5.1 Adversarial IP Addresses . . . 136
6.5.2 Adversarial Similarity Analysis (IP-Based) . . . 136
6.5.3 Evolution of the Adversarial Infrastructure . . . 138
6.6 Unknown/Unseen IoT Malware Labeling . . . 141
6.6.1 Functional Similarity Analysis . . . 142
6.6.2 Malware Family Labeling . . . 146
6.7 Experimental Results: Dynamic Analysis . . . 148
6.7.1 IoT Malware Sandboxing Environment . . . 148
6.7.2 Obtained Network Traffic . . . 149

6.8 Summary and Concluding Remarks 151

7 Conclusion and Future Work **153**

7.1 Conclusion 153

7.2 Future Work 155

Bibliography **156**

List of Figures

Figure 2.1 An overview of the darknet data (scanning and backscatter traffic). 10
Figure 2.2 The overall architecture of the `Mirai` botnet [1]. 11
Figure 3.1 Countries with the largest number of deployed IoT devices. 27
Figure 3.2 The cumulative number of daily discovered compromised CPS and consumer IoT devices at the darknet over the 6 days analysis interval. 28
Figure 3.3 Countries with the largest percentage of compromised IoT devices. 29
Figure 3.4 Percentage of compromised consumer IoT devices by type/category. 30
Figure 3.5 Percentage of TCP, UDP, and ICMP traffic generated by compromised IoT devices in CPS and consumer realms. 33
Figure 3.6 Overall UDP packets sent by compromised (a) CPS and (b) consumer IoT devices to destination IP addresses and ports. 34
Figure 3.7 Distribution of the scanning and backscatter packets generated by compromised IoT devices and DDoS victims, respectively. 37
Figure 3.8 Distribution of the generated backscatter packets by CPS and consumer IoT devices (143 hours). 38
Figure 3.9 Countries with the largest number of DoS victims. 39
Figure 3.10 Countries with the largest number of backscatter packets. 39
Figure 3.11 Overall TCP scanning packets generated towards destination IP addresses and ports by exploited (a) CPS and (b) consumer IoT devices. 41
Figure 3.12 The distribution of TCP scanning packets generated by exploited IoT devices towards the top 5 targeted protocols/services. 44

Figure 3.13 Distribution of received packets from the top 8,839 IoT devices and the malicious devices flagged by Cymon ($N = 816$). 46
Figure 4.1 The overall approach for detecting and characterizing IoT threats. 54
Figure 4.2 The distribution of all TCP-SYN scanning packets generated by compromised IoT devices during the analysis interval (143 hours) [2]. 57
Figure 4.3 The distribution of unique scanning objectives over the number of scanned ports. 60
Figure 4.4 Top 17 scanned destination ports by the highest number of IoT devices within strobe scans. 62
Figure 4.5 Distribution of compromised IoT device types per scanning class. 66
Figure 4.6 Countries with the largest number of exploited IoT devices from each class (initial data–April 2018). 67
Figure 4.7 An example of K-NN distance graph for IoT devices classified within strobe scans. 70
Figure 4.8 Clustering results for Range scans ($MinPts = 3$, $\varepsilon = 0.15$, clusters=3) . . . 73
Figure 4.9 The distribution of IoT device type in the largest clusters within Range scans. 73
Figure 4.10 Clustering results for Strobe scans ($MinPts = 3$, $\varepsilon = 0.2$, clusters=12) . . . 74
Figure 4.11 The distribution of IoT device type in the largest clusters within Strobe scans. 74
Figure 4.12 Clustering results for Wide scans ($MinPts = 3$, $\varepsilon = 0.3$, clusters=3) 75
Figure 4.13 The distribution of IoT device type in the largest clusters within Wide scans. 76
Figure 4.14 The created environment for analyzing IoT malware. 79
Figure 4.15 The distribution of all TCP-SYN scanning packets generated by compromised IoT devices during the new analysis intervals (108 hours). 81
Figure 4.16 Countries with the largest number of exploited IoT devices from each class (new data–May 2018). 83
Figure 4.17 Visualizing scans targeting Telnet port 23 with correlated groups of devices using HDBSCAN. 90

Figure 4.18 Visualizing scans targeting port 23 with the distribution of IAT CDF for all correlated groups of sources. The solid black lines represent the mean (center) of all CDFs in each group. The fitted CDFs from Propositions 1.1 and 1.2 are marked with asterisks (x). 91
Figure 4.19 Visualizing scans generated by IoT and non-IoT devices targeting HTTP ports 80/8080 using a two-dimensional representation of their IAT distributions. . . 92
Figure 4.20 Visualizing scans targeting HTTP ports 80/8080 with correlated groups of devices using HDBSCAN. 93
Figure 4.21 Visualizing scans targeting HTTP ports 80/8080 with the distribution of IAT CDF for all correlated groups of sources. The solid black lines represent the mean (center) of all CDFs in each group. 94
Figure 5.1 Overall architecture of the implemented system. 98
Figure 5.2 Macroscopic views of the various IoT-generated packets towards the darknet over 24 hours of analysis interval (1,440 minutes). 105
Figure 5.3 Compromised IoT device types. 106
Figure 5.4 Countries with the largest number of compromised devices. 107
Figure 5.5 The commutative distribution of the total number of scanned destination ports by the exploited IoT devices. 109
Figure 5.6 Cumulative number of exploited IoT devices within the top 10 scanning campaigns targeting S_1–S_{10}. 110
Figure 5.7 Examples of scanning campaign evolution over the analysis interval (20x6 hours of aggregated data). The campaigns target ports specified in S_1, S_3, and S_4. . 111
Figure 5.8 Targeted device types (DDoS victims). 114
Figure 5.9 Top 15 countries with the highest number of DDoS victims. 115
Figure 5.10 Backscatter packets generated by DDoS victims. 116
Figure 5.11 Execution time analysis for (a) data parsing and aggregation, and (b) the correlation of execution times to the number of flowtuples in parsed/aggregated data files. 117

Figure 5.12 Correlation of execution time with the accumulative number of IoT devices in the merged data files. 118
Figure 5.13 CPU and memory usage for a sample of four consecutive hours of aggregated/profiled darknet data. 119
Figure 6.1 Overall approach for strings extraction and similarity analysis. 125
Figure 6.2 The distribution of the analyzed IoT malware samples as seen on `VirusTotal` (last updated on October 30, 2020). 133
Figure 6.3 Detection timeline and duration as seen from `VirusTotal` reports. 134
Figure 6.4 Total number of active days for the analyzed malware samples within every family. 135
Figure 6.5 The distribution of the adversarial IP addresses across 54 countries. 137
Figure 6.6 Identified connected components (CCs) based on similar adversarial IP addresses. 138
Figure 6.7 The three largest IP-based connected components (CCs). 139
Figure 6.8 Evolution of the largest connected components as perceived from their detection times on `VirusTotal`. 141
Figure 6.9 Top 20 identified string sequences and their frequencies across the IoT malware samples. 142
Figure 6.10 Similarity analysis results in terms of the correlated sub-components within C#1–C#3. 144
Figure 6.11 Minimum, maximum, and average centrality measures for nodes within the remaining 55 sub-components withing CC#1–CC#3. 146
Figure 6.12 Distribution of the mislabeled IoT malware samples ($n = 235$). 147
Figure 6.13 The implemented dynamic IoT malware analysis environment. 149
Figure 6.14 Functional similarity analysis results for the IoT malware samples targeting top 5 scanned port sets. 151

List of Tables

Table 2.1 A list of leveraged datasets and resources throughout this thesis. 21

Table 3.1 Top 5 ISPs hosting the highest number of compromised consumer IoT devices. 30

Table 3.2 Top 5 ISPs hosting the highest number of compromised IoT devices in CPS realms. 31

Table 3.3 Top 10 CPS realms hosting compromised IoT devices. 31

Table 3.4 Top 10 targeted UDP protocols/ports. 35

Table 3.5 Top 14 protocols/ports with the most TCP scanning packets generated by exploited IoT devices (CP=93.3%). 42

Table 3.6 Identified threats summary. Note that the identified threats are not mutually exclusive. 47

Table 3.7 Identified, previously unreported malware families exploiting IoT Devices. . 48

Table 4.1 Top 15 scanned services/ports (CP=98.7%). Source and destination IP counts represent the number of IoT devices and scanned IP addresses. 58

Table 4.2 Top 20 frequent scanning objectives within S_{strobe} generated by about 94% of all devices in strobe scans class. 64

Table 4.3 Association rules related to the scanned ports identified in Table 4.2. 65

Table 4.4 The selected flow features for analysis using DBSCAN ($\beta = 10$). 69

Table 4.5 Summary of the clustering results and evaluation ($MinPts$=3 and $\beta = 10$ features). 71

Table 4.6 Analyzed IoT malware samples and their targeted ports. 80

Table 4.7 Top 18 scanned services/ports (CP=99%). 84

Table 4.8 Frequent scanning objectives within strobe scanning class (CP=89.4%). . . . 85
Table 5.1 Compromised IoT devices and their generated scanning traffic type(s). 104
Table 5.2 Compromised IoT device models (scanning). 108
Table 5.3 Top 10 identified scanning objectives ($S^{'}$). 108
Table 5.4 Aggregated flow features for device d_i within interval I. 113
Table 5.5 Clustering results for the top 5 scanning campaigns. 113
Table 5.6 DDoS Victims' device models. 114
Table 6.1 A summary of the analyzed IoT malware samples. 132
Table 6.2 Identified IP-based CCs with underlying malware samples from unique malware families. 140
Table 6.3 The largest IP-based CCs and their underlying malware family distribution. . 141
Table 6.4 Summary of the identified sub-connected based on the strings-based similarity analysis of the top 3 largest CCs. 143
Table 6.5 Top 10 scanned destination port sets ($n = 24$). 150

Chapter 1

Introduction

1.1 Problem Scope and Motivation

Despite the benefits of using Internet-of-Things (IoT) devices in different aspects of our lives, the analysis of IoT-driven cyber attacks, which leveraged compromised IoT devices to perform Internet-scale malicious activities, demonstrate the insecurity of these devices at scale [3, 4]. In addition, the significant number of deployed IoT devices, and the fact that compromised IoT devices are utilized to enable orchestrated Internet-scale distributed attacks, have led to the rise of IoT-tailored malware as a major threat in recent years [5]. Moreover, with the rapid implementation and integration of high speed and scalable 5G networks and Mobile Edge Computing (MEC) technologies, the number of deployed IoT devices is also expected to increase significantly, which will amplify the threats associated with the deployment of insecure IoT devices within 5G networks. This is considered as a major concern for Internet and telecommunication service providers and operators, especially with the central role of IoT in the operation of future "smart" technologies IoT networks that support everyday activities within the consumer space and the public/private sectors.

To mitigate such threats, there is an utmost need to develop effective tools and techniques for prompt detection and characterization of IoT threats by inferring malware-infected IoT devices and characterizing their unsolicited activities. This is an extremely challenging task due to: (i) the world-wide deployment of large number of insecure IoT devices, (ii) the lack of information about exploited IoT devices within the user space, and (iii) the lack of control over user-owned

IoT device operations. In this thesis, we aim at addressing these challenges by leveraging passive Internet measurements along with data-driven methodologies to infer, characterize, and investigate malware-infected IoT devices and their generated malicious activities.

1.2 Objectives and Research Questions

The main objective of this thesis is to leverage passive Internet measurements to implement and evaluate data-driven approaches for inferring malware-infected IoT devices and characterizing their generated scanning activities. Indeed, given the threats associated with the rise of IoT malware, and the challenges associated with the lack of empirical data about malware-infected IoT devices and their unsolicited activities in the wild, the capability to infer and characterize malware-generated activities is considered essential to build a better understanding about the threat landscape while aiding the development of effective detection and mitigation measures.

Specifically, we aim at answering the following research questions (RQs):

(1) *How can we leverage publicly available information about IoT devices along with passive Internet measurements to infer malware-infected IoT devices at scale while characterizing their unsolicited activities?*

(2) *Given the inferred compromised IoT devices and their generated network traffic, how can we detect and characterize orchestrated scanning campaigns generated by well-coordinated IoT botnets?*

(3) *How can we leverage technological advances in terms of big data analytics frameworks along with passive Internet measurements to develop a scalable and effective IoT threat detection and analysis framework/system that produces IoT-specific threat intelligence in near real-time?*

(4) *How can we build a better understanding about the current IoT malware threat landscape, family distribution, detection timeline, and evolution?*

(5) *How can we utilize static and dynamic malware analysis techniques to address malware labeling and family attribution problems? How can we benefit from such analysis to identify*

possibly new, unknown malware families/variant?

1.3 Methodology

To achieve the aforementioned objectives, in this thesis, we device data-driven methodologies that leverage IoT device information along with two main resources of passive Internet measurements: (i) one-way traffic captured at a large-scale network telescope (darknet), and (ii) IoT malware binary executables files detected by a well specialized IoT honeypot. In general, given the passive Internet measurements and IoT malware binaries, we perform exploratory data analysis to analyze and characterize the IoT threat landscape. Additionally, we utilize the passive measurements along with the generated insights to implement multi-level methodologies for effective IoT threat detection and analysis. We summarize the devised methodologies for answering our research questions as following:

- To answer our first research question (RQ1), we present a first empirical look at the magnitude of compromised IoT devices that have been deployed in both consumer and CPS realms. Initially, large-scale correlations between passive measurements and IoT-relevant information is conducted to shed the light on Internet-wide unsolicited IoT devices. Subsequently, empirical measurements, characterization, and analysis is presented to thoroughly investigate IoT-generated unsolicited traffic, including backscattered traffic from IoT devices that have been targeted by DoS attacks, and scanning activities from exploited IoT devices. Finally, an attempt is made to uncover the maliciousness of such unsolicited IoT devices by utilizing a publicly available threat repository and an in-house built malware database.

- Motivated by the fact that IoT malware/botnets heavily rely on coordinated scanning activities to propagate through the Internet, to answer our second research question (RQ2), we leverage macroscopic, empirical passive network telescope data to execute a multi-level methodology for inferring malware-infected IoT devices and investigating their generated scanning activities. In addition, we leverage data mining methods to unveil common scanning objectives among compromised IoT devices, which reflect the targeted ports/services. More importantly, we demonstrate a meaningful approach for identifying scanning campaigns by

clustering correlated IoT devices based on their scanning objectives and similarities in their scanning behaviors over time.

- To answer our thirs research question (RQ3), we leverage a big data analytics framework (`Apache Spark`) to design and develop a scalable system for automated detection of compromised IoT devices and characterization of their unsolicited activities. The system utilizes IoT device information and passive network measurements obtained from a large network telescope, while implementing an array of data-driven methodologies rooted in data mining and machine learning techniques, to provide a macroscopic view of IoT-generated malicious activities. We evaluate the system with more than 4TB of passive network measurements and demonstrate its effectiveness in the network forensic investigation of compromised devices and their activities, in near real-time.

- To answer RQ4, we perform a large-scale characterization of real IoT malware binaries/executables that were detected by a specialized IoT honeypot (IoTPOT [6]) during the past two years. We leverage a publicly available threat repository (`VirusTotal`) to characterize known IoT malware samples in terms of their family distribution, detection timeline, and activity duration (threat persistent). Moreover, we demonstrate the effectiveness of the IoT honeypot towards early detection of IoT malware samples while highlighting new, possibly undetected malware samples.

- To answer our last research question (RQ5), we utilize reverse-engineering techniques to extract meaningful strings from the analyzed IoT malware binaries including IP addresses associated with adversaries (e.g., C&C servers), targeted destinations (e.g., scanned ports and/or IP addresses), and command strings that reflect the underlying behaviors of the analyzed malware. Furthermore, we perform a multi-level similarity analysis to uncover underlying relationships among the analyzed malware samples. Finally, we leverage dynamic malware analysis to corroborate the findings of our static malware analysis while extending our knowledge about the behavioral characteristics of the IoT malware by analyzing its generated scanning traffic (whenever available).

1.4 Contributions

We present a summary of the main contributions maid throughout this thesis in the following sub-sections:

1.4.1 Inferring, Characterizing, and Investigating Internet-Scale Malicious IoT Device Activities: A Network Telescope Perspective

Considering the challenges in terms of lack of empirical data and knowledge about IoT devices and their unsolicited behaviors online, we draw-upon close to 5TB of recent Internet measurement data collected at a large network telescope (darknet) and execute correlations with a near real-time IoT database to empirically characterize the magnitude of Internet-scale IoT exploitations in both, consumer and critical CPS realms. The generated insights not only render a first attempt ever to empirically shed the light on the large-scale insecurity of the IoT paradigm, but are also intended to contribute to operational/actionable cyber security by providing Internet-wide, IoT-tailored notifications of such exploitations, thus permitting rapid remediation. Moreover, we execute a first-of-a-kind, large-scale empirical characterization and analysis of IoT-centric unsolicited activities as perceived by a large network telescope. We uncover the nature of such traffic, its sources, employed protocols, targeted ports, upon various others. Given the lack of IoT-specific attack signatures, we postulate that the analyzed traffic from this work could be leveraged to design such signatures, in addition to promoting and facilitating further IoT-tailorted forensic investigations by making the captured unsolicited empirical traffic available to the research and operations communities at large.

The outcome of this contribution is published/presented at the DSN conference (2018) [2].

1.4.2 Inferring and Investigating IoT-Generated Scanning Campaigns Targeting A Large Network Telescope

We leverage over 6TB of passive Internet measurements collected at a network telescope along with IoT device information from `Shodan` to obtain about 172M TCP-SYN scanning packets generated by more than 8,000 compromised IoT devices over 11 days. We extend our previous work [2]

by introducing a stratified methodology, which utilizes passive darknet data for investigating emerging IoT malware/botnets through inferring compromised IoT devices and characterizing their underlying scanning campaigns. Additionally, we device a feasible approach for uncovering IoT-generated scanning campaigns, which is based on frequent pattern analysis to identify common scanning objectives (targeted ports) and unsupervised clustering of correlated IoT devices with similar behavioral characteristics over a period of time. Moreover, we highlight the evolution of IoT-generated scanning campaigns towards targeting new, or previously uncommon vulnerabilities, which indeed corroborate on the evolutionary nature of IoT malware/botnets. Finally, we propose stochastic models for low-rate, IoT-generated scanning campaigns and leverage empirical analysis to characterize campaigns targeting Telnet and HTTP ports.

The outcome of this contribution is published/presented at the IEEE TDSC and IEEE Networking Letters (2020) [7, 8]

1.4.3 A Scalable Platform for Enabling the Forensic Investigation of Exploited IoT Devices and their Generated Unsolicited Activities

Given the lack of scalable cyber-threat intelligence reporting and analysis capabilities that can trigger informed decisions for in-depth forensic investigations in near real-time, we implement a scalable system that enables scalable and timely network forensic investigations through inferring compromised IoT devices and characterizing their unsolicited activities. The system, which utilizes IoT device information and passive network traffic captured at a large network telescope, leverages the capabilities of a big data analytics framework (`Apache Spark`) to implement multi-level data-driven methodologies rooted in data mining and unsupervised machine learning. The system is evaluated using 4TB (120 hours) of IoT-generated unsolicited traffic captured "in the wild," to identify exploited IoT devices that generated millions of scanning packets on the Internet. Moreover, while the system supports various views for macroscopic and fine-grained monitoring and analysis of the detected activities, it utilizes behavioral characteristics of IoT devices in terms of aggregated flow features to support the implementation of a number of network forensic applications such as detecting and fingerprinting scanning campaigns, investigating campaign persistence and evolution, inferring IoT botnets, and identifying IoT DDoS victims.

The outcome of this contribution is presented at the DFRWS EU conference and published in the Elsevier's Digital Investigations journal (2020) [9]

1.4.4 A Strings-Based Similarity Analysis Approach for Large-Scale IoT Malware Analysis, Characterization, and Family Attribution

We are among the first to perform a large-scale characterization of real IoT malware binaries/executables that were detected by a specialized IoT honeypot (IoTPOT [6]) during the past two years. We leverage a publicly available threat repository (`VirusTotal`) to characterize known IoT malware samples in terms of their family distribution, detection timeline, and activity duration (threat persistent). Moreover, we demonstrate the effectiveness of the IoT honeypot towards early detection of IoT malware samples while highlighting new, possibly undetected malware samples.

Additionally, we utilize reverse-engineering and static malware analysis techniques to extract meaningful strings from IoT malware binaries in terms of adversarial IP addresses and embedded commands. More importantly, we execute a multi-level strings-based malware similarity analysis approach to correlate IoT malware executable binaries and investigate their underlying correlations. Indeed, we uncover adversarial infrastructure and shared resources, which are used to operate malware-driven malicious activities. Subsequently, we leverage functional similarity analysis to address the problems of unknown malware family labeling and attribution by correlating IoT malware samples into groups with common malicious implementations. Interestingly, we uncovered groups of unknown malware samples that evolved over time to form possibly new malware families/variants.

We also leverage an implemented dynamic malware analysis testbed to extend our findings by executing malware binaries and extracting behavioral characteristics in terms of connection attempts to adversarial IP addresses and scanning traffic, whenever available. Indeed, our findings corroborate the static analysis results in terms of the extracted adversarial IP addresses while extending knowledge about the behavioral characteristics of IoT malware in terms of the scanned destination ports/services, which shed light on the associated vulnerabilities and IoT-specific threats.

The outcome of this contribution is submitted/published at the IEEE TDSC and IEEE Networking Letters (2021) [10, 11]

1.5 Thesis Organization

The remainder of this thesis is structured as follows: In Chapter 2, we present background information along with a review of related literature. In Chapter 3, we present our approach for inferring and characterizing IoT devices and their unsolicited activities using passive Internet measurements and known IoT devices information. Chapter 4 presents the multi-level approach followed towards investigating IoT-generated scanning campaigns on the Internet. In Chapter 5, we detail the implementation and evaluation of a scalable platform for enabling the forensic investigation of malware-infected IoT devices and their unsolicited activities. In Chapter 6, we summarize our findings after performing static/dynamic malware analysis on a large corpus of IoT malware while proposing a strings-based similarity analysis approach for IoT Malware analysis, characterization, and family attribution. Finally, we summarize concluding remarks and discuss possible future research directions in Chapter 7.

Chapter 2

Background, Related Work, and Leveraged Datasets

In what follows, we provide background information about network telescope, IoT-driven cyber atatcks, and common malware analysis techniques, followed by a review of literature with respect to various concerned topics as presented in recent published work. We also provide an overview of the different datasets that were leveraged throughout the presented contributions in this thesis.

2.1 Background

2.1.1 Network Telescope

Network telescope data or darknet data, consists of one-way traffic targeted towards routable, allocated yet unused IP addresses (dark IP addresses). Since these IP addresses are not bound to any services, any traffic targeting them is characteristically unsolicited [12, 13]. Typically, darknet data consists of scanning and backscatter activities, in addition to other less common traffic such as misconfiguration and reflection attacks data, to name a few [12–15]. As shown in Figure 2.1, an adversary sends scanning packets to perform reconnaissance activities and identify vulnerable devices on the Internet. Given that the darknet implements many sensors across the Internet address space, a proportion of these scanning packets will be captured at the darknet and stored for further use. There are several ways for scanning the Internet, among which sending TCP-SYN packets

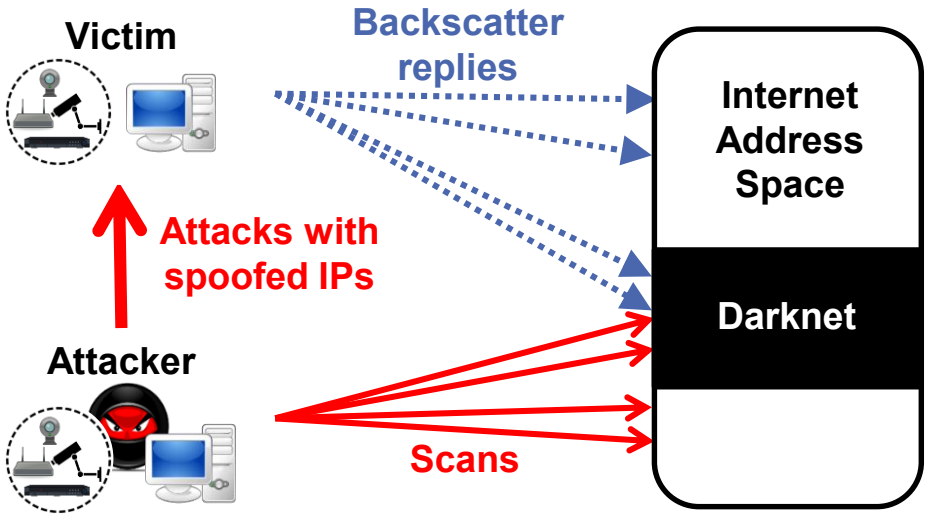


Figure 2.1: An overview of the darknet data (scanning and backscatter traffic).

is the most prominent [16, 17]. In this proposal, we focus our analysis on TCP-SYN requests, as they represent a largest portion of darknet data. It is worth noting that ICMP Echo requests, which are also commonly used for network scans, are not considered for further analysis due to their negligible magnitude in the overall darknet traffic. On the other hand, while UDP packets are also used for scanning the Internet, their stateless nature makes them indistinguishable from non-scanning packets without further analysis of the packet payload. Therefore, UDP packets, which represent a relatively small portion of the overall darknet traffic, are also excluded from further analysis throughout this proposal.

On the other hand, a considerable portion of the darknet traffic represents backscatter replies, which is a byproduct of (D)DoS attacks that target IoT devices. When a victim IoT device is attacked by a flood of packets generated from spoofed source IP addresses (that happened to be belonging to the network telescope IP space), the device will generate reply packets destined to the darknet, which can then be collected and extracted. These packets are mainly TCP (SYN-ACK and RST) or ICMP reply packets (e.g., Echo Reply) [18].

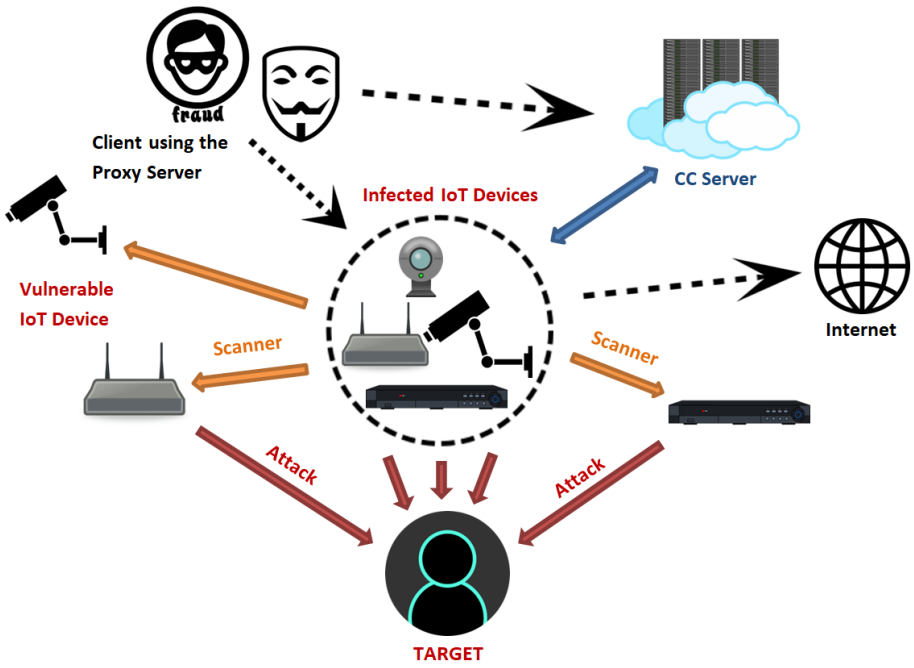


Figure 2.2: The overall architecture of the `Mirai` botnet [1].

2.1.2 IoT-Driven Cyber Attacks

The large-scale DDoS attacks caused by the `Mirai` botnet highlighted the power of exploited IoT devices as effectiveness attack enablers that can evade detection while performing malicious activities at scale [3]. As shown in Figure 2.2, an adversary can utilize millions of exploited devices from all around the world to send synchronized, small, benign-like packets to a victim without being detected by conventional security measures. Furthermore, given that these packets come from what are supposed to be benign devices deployed in many geographical locations, it is very difficult, if not impossible, to apply timely measures for preventing them from reaching their destinations. In addition to the IoT-generated DoS attacks, IoT malware/threats propagation through the cyber space by scanning the Internet for vulnerable IoT devices that could be exploited and mined for use is future malicious activities. This however, represents an important signature of recently discovered IoT attacks, which was reflected through their heavy involvement in Internet-scale scanning activities generated by compromised IoT devices [2, 3, 19].

The effectiveness of the `Mirai` botnet and its consequent variants, along with the insecurity

of IoT devices at scale, have lead to the rise of IoT-tailored malware, which aim at exploiting vulnerable IoT devices that will be utilized within coordinated botnets to perform further malicious activities [4, 19–21]. These IoT malware/botnets have gained much popularity among adversaries due to the insecurity of the IoT paradigm and the wide range of existing vulnerabilities. Moreover, the availability of the malicious source code of the `Mirai` botnet have lead to the evolution of new `Mirai`-like variants, which follow a similar overall approach as presented in Figure 2.2 to exploit a variety of existing and/or new vulnerabilities. For instance, while the `Mirai` botnet focuses on scanning common services such as HTTP and Telnet on TCP ports 80/23/2323, the analysis of recent IoT malware/botnets highlighted scanning activities towards TCP port 40, which is associated with a threat caused by `Midnight Commander` [22], a visual file manager which sometimes access FTP servers running at this port. In addition, TCP port 37215 was scanned by several IoT malware. This port is used by newer `Mirai` variants such as the `Satori` botnet [23,24], to remotely exploit specific models of Huawei routers by invoking a firmware upgrade action through the vulnerable Universal Plug and Play (UPnP). Furthermore, TCP port 52869 was also found to be associated with a similar vulnerability on UPnP, which can be reached via SOAP requests [25] via TCP port 52869 [26]. Note that both aforementioned vulnerabilities were found to be widely associated with `Huawei` routers, which gives a clear indication of targeted attack towards such devices.

2.1.3 Malware Analysis Techniques

A number of common techniques are used to detect malicious executable files and malware, which relies on analyzing various aspects of the malicious such as its content, implementation, execution, functionalities, and behavioral characteristics, to name a few. The main objective of all techniques is to build a better understanding about the malware and develop effective counter measures and mitigation techniques. While there are several ways to analyze malware, there are three common techniques:

Static: Also known as code analysis, refers to analyzing the malicious software without executing it. The aim is to extract information from the binary/source code to determine either the software contains malicious code or content. Static malware analysis is often used to fingerprint the analyzed malware binaries and create unique signatures and hashes, which can be used for further

malware detection purposes. Given that the malware binaries are not easily readable, static analysis often requires reverse engineering the malicious executable files using different tools to obtain information such as the structure of the malicious code, control flow graphs (CFG), embedded strings, attack commands, malicious domains/IP addresses, and attack payload, to name some. Different tools that can be used to perform static analysis are debugger, dissembler, decompiler and source code analyzers. Methods that are used in performing static analysis include File Format Inspection, String Extraction, Fingerprinting, AV scanning and Disassembly.

Static malware analysis techniques are considered as simple, yet effective methods to analyze malware samples without actually executing them. Therefore, static analysis can be scalable and automated, which is the case with signature based malware detection and analysis techniques. Nevertheless, malware authors often deploy various malware obfuscation techniques to avoid static analysis [27]. Some of the common malware obfuscation techniques include but not limited to compress/encrypt parts or the whole malware binary code, false code injection, arbitrary jumps, and code/function misplacement.

Dynamic: Following this approach, the malicious software is executed in a controlled environment, which is normally created using virtual machines to emulate a target host, to observe its functional and behavioral characteristics. The aim of dynamic analysis is to analyze the behaviors of the malicious code to build a better understanding about the malware and design appropriate defenses and countermeasures. Dynamic analysis is used to trace the function calls, obtain control flows, analyze the instructions, extract parameters of functions. The tools that are used for dynamic analysis are malware sandbox, simulator, emulators, and Process Explorer, to name a few [28].

The main advantage of dynamic analysis over static analysis is that it can be effectively used to provide an in-depth analysis of the malware by observing its actual behavior, while recording all the actions taken by the malware to change the host system/environment. In addition, dynamic analysis can be used to overcome problems with malware obfuscation, which limit the static-based techniques. On the down side, dynamic analysis requires more time and effort towards building the special execution environment (sandbox), which is not a straight forward operation. More importantly, a considerable number of existing malware deploy detection evasion techniques [29], which

are employed specifically to avoid execution and dynamic analysis in such virtual sandboxing environments.

Hybrid: As implied by their name, hybrid techniques combine both static and dynamic analysis techniques to benefit from both approaches. Following hybrid techniques, the malicious code is first analyzed to obtain useful information and signatures. Then, the malware is executed in the dynamic analysis environment to observe its behaviors and complement the static analysis outcomes.

Hybrid malware analysis techniques are often used to combine the benefits of both static and dynamic malware analysis. despite their benefits, by default, such techniques inherit the challenges of both static and dynamic analysis techniques. Moreover, implementing such in-depth analysis techniques required thorough analysis of the malware, which is costly and can hamper the scalability of the approach.

Next-Generation Techniques: Considering the limitations of conventional malware analysis techniques (e.g., static and dynamic), a series of next-generation methods have been proposed to benefit from multi-modal views of the malware and its behavioral characteristics in the wild. For instance, AI-based techniques such as Machine and Deep learning models have been utilized to perform static malware analysis by extracting features from different representations of the malware binaries (e.g., image-based and text-based features) [30–32]. Additionally, end-point detection and analysis can be deployed on the host to monitor and analyze software execution and their associated processes, while making decisions towards anomaly detection, application white-listing, signature-based detection, and network traffic fingerprinting and analysis.

A main advantage of next-generation analysis techniques is that they extend the analysis beyond the typical malware analysis techniques, while benefit from the advantages of AI-based methods to perform scalable malware analysis and detection. In general, AI-based techniques have been shown to be scalable and effective in different contexts, which are transferred to malware analysis and detection as well. Additionally, deep learning models/networks, which do not require feature engineering/extraction, can be leveraged to support process automation by performing end-to-end analysis. Despite their benefits, AI-based techniques have a number of main limitations including but not limited to: require domain knowledge to perform feature selection/engineering (in the case of machine learning); require costly training and testing to adapt to the changing threat landscape;

require labeled data in the case of supervised learning; overfitting and underfitting; and detection evasion techniques (e.g., mimicking benign behaviors).

2.2 Related Work

2.2.1 IoT Security and Vulnerability Analysis

IoT device vulnerabilities have been discussed in the literature from different perspectives. For instance, Cui et al. [33], performed large-scale Internet scans of IoT devices and provided quantitative evidence on the vulnerable devices. They found over half a million publicly accessible embedded devices configured with factory default root passwords. Interestingly, this vulnerability was in fact one of the main reasons behind the large-scale outbreak of the `Mirai` botnet in late 2016 [3]. Considering the impact of vulnerability analysis in identifying and addressing IoT threats, Sachidananda et al. [34] deployed a testbed of IoT devices in an experimental setting and demonstrated a preliminary effort towards building a feasible and usable platform for IoT vulnerability analysis and testing.

From a different perspective, Costin et al. [35] provided an extensive assessment of IoT device firmware. Similarly, FIRMADYNE was proposed by Chen et al. [36] to automatically analyze Linux-based firmware images and identify vulnerabilities. A noticeable number of IoT security research work has been dedicated to synthesizing IoT context-aware permission models. For instance, Yu et al. [37] proposed a policy abstraction language that is capable of capturing relevant environmental IoT contexts, security-relevant details, and cross-device interactions, to vet IoT-specific network activities. Along the same research direction, Jia et al. [38], proposed ContextIoT, a system that is capable of supporting complex IoT-relevant permission models through efficient and usable program-flow and runtime taint analysis. Fernandes et al. [39] proposed a similar program-flow tracking approach that used taint arithmetic to detect policy violations and restrict traffic generated from exploited IoT applications. In the context of protocol vulnerabilities, Ur et al. [40] studied numerous types of home automation IoT devices and unveiled various insights with regards to the security and usability of the implemented access control models. Ronen and Shamir [41] demonstrated information leakage attacks by instrumenting a set of IoT smart lights.

2.2.2 IoT Data Capturing Initiatives

Given the rareness of IoT-relevant empirical data, passive network traffic analysis has been introduced as an effective approach towards studying Internet-wide cyber threats associated with IoT devices. For instance, IoTPOT, was deployed by Pa et al. [6] as a honeypot that emulates Telnet services of various IoT devices running on different CPU architectures. In alternative work, Guarnizo et al. [42] presented the Scalable High-Interaction Physical Honeypot platform for IoT devices (SIPHON). The authors demonstrated an approach for imitating various IoT devices on the Internet to attract significant malicious traffic by leveraging worldwide wormholes and a few physical devices. Luo et al. [43] implemented a machine learning approach to create an intelligent honeypot that automatically learns the behavioral responses of IoT devices through active scanning in order to mimic realistic interactions with attackers. Vervier et al. [5] deployed a honeypot that captured a wider range of emerging IoT threats as compared to previous honeypots (e.g., IoTPOT). They used 6 months of collected data along with multiple sources of cyber-intelligence to explore current IoT malware and their emerging behavioral characteristics.

A number of ongoing projects have been implemented to perform active scanning of the Internet in order to locate and profile Internet connected devices on frequent basis. For instance, `Censys` was created by security researchers at the University of Michigan as an online tool for discovering devices, networks, and infrastructure on the Internet while monitoring changes over time [44]. `Shodan` on the other hand [45], performs IP banner analysis to provide a more specialized online IoT device search engine that indexes different types of IoT devices. In line with the same approach, Feng et al. [46] proposed a rule-based IoT device detection model that addresses the limitations of conventional banner grabbing/analysis techniques (e.g., insufficient device information) by utilizing device information from multiple online resources.

2.2.3 Passive Internet Measurements

Passive network traffic analysis is introduced as an effective approach towards studying Internet-wide cyber threats. For instance, passive DNS data, which consist of historic replicas of DNS queries and responses, was utilized to detect various threats associated with DNS abuse/misuse [47].

Furthermore, given the rareness of IoT-relevant empirical data, several recent efforts were proposed to create honeypots for collecting, curating, and analyzing IoT data. The first IoT-tailored honeypot, coined, IoTPOT, was designed and deployed by Pa et al. [6]. IoTPOT emulates Telnet services of various IoT devices running on different CPU architectures. In alternative work, Guarnizo et al. [42] presented the Scalable High-Interaction Physical Honeypot platform for IoT devices (SIPHON). The authors demonstrated how by leveraging worldwide wormholes and a few physical devices, they were able to mimic various IoT devices on the Internet and to attract significant malicious traffic. Luo et al. [43] implemented a machine learning approach to create an intelligent honeypot that automatically learns the behavioral responses of IoT devices through active scanning in order to mimic realistic interactions with attackers. U-Pot was introduced by Hakim et al. [48] as an interactive open-source framework for emulating IoT devices that support Universal Plug and Play (UPnP) protocols/services. In addition to the promising evaluation results in terms of emulating real IoT devices, the usability of the framework and its ability to automatically create honeypots from device description documents is worth noting. In a recent work, Vervier et al. [5] deployed a honeypot that captured a wider range of emerging IoT threats as compared to previous honeypots (e.g., IoTPOT). They used 6 months of collected data along with multiple sources of cyber-intelligence to explore current IoT malware and their emerging behavioral characteristics.

In addition to IoT-tailored honeypots, passive network telescope or darknet data, which represents one-way network traffic collected at unused IP addresses, has been adopted to analyze cyber activities and obtain cyber-intelligence [49, 50]. The idea of leveraging darknet to monitor unused IP addresses for security purposes was first brought to light in the early 1990's by Bellovin for AT&T's Bell Lab's Internet-connected computers [51, 52]. Since then, the focus of network telescope studies has shifted several times, closely following the volatile nature of new adversaries. More importantly, with the rise of IoT-driven cyber attacks, passive network telescope data was leveraged to capture and analyze unsolicited IoT-generated activities. For instance, some of the important contributions include the discovery of the relationship between backscattered traffic and DDoS attacks in 2001 [53], worm propagation analysis between 2003 and 2005 [54, 55], the use of time series and data mining techniques on telescope traffic in 2008 [56], the monitoring of large-scale cyber events through telescopes in 2014 [57], and more recently, the study of amplification

DDoS attacks using telescope sensors [58, 59].

From a different perspective, Fachkha et al. [50] presented a probabilistic model for sanitizing network telescope data and inferring orchestrated probing campaigns towards cyber-physical systems (CPS). Furthermore, Antonakakis et al. [3] used unique `Mirai` traffic signatures to capture `Mirai`-related scans at the network telescope for further analysis of the botnet. Torabi et al. [2] proposed a data-driven methodology to infer compromised IoT devices by leveraging IoT device information and darknet data through the execution of correlation algorithms on IP header information. In line with the same line of research, Safaei Pour et al. [60] proposed a data dimensionality reduction technique to infer and characterize Internet-scale IoT probing campaigns by analyzing passive network measurements collected from the darknet. In addition, they utilized several shallow and deep learning models to sanitize telescope data and infer probing activities generated by compromised IoT devices based on a number of flow features.

2.2.4 IoT Malware Data Collection and Analysis

The rise of large-scale IoT-driven cyber attacks (e.g., the `Mirai` botnet [3]), demonstrated the evolution of IoT malware as a major threat. Consequently, several studies have been performed in recent years to infer IoT malware and characterize the behaviors of exploited IoT devices [7, 60]. To facilitate IoT malware analysis, researchers implemented several specialized IoT honeypots to address the lack of empirical data/knowledge on IoT device deployment [5]. For instance, Pa et al. [6] introduced IoTPOT, which is one of the pioneering IoT honeypots for detecting and collecting malware targeting IoT devices. Luo et al. [43] proposed using machine learning techniques to create an intelligent and interactive honeypot, which leverages the various behaviors of IoT devices. Moreover, Wang et al. [61] presented IoTCMal, which is a hybrid framework for capturing more comprehensive IoT malware samples.

The various IoT malware/data collection initiatives introduced over the past years provided fundamental knowledge and data about IoT malware, which can be effectively utilized to create a better understanding about the state of IoT malware and its evolution. This is typically done through static malware analysis techniques [62] to extract information on the structural and behavioral similarities of the analyzed executable binaries and their underlying code structure, which can leverage further

clustering, classification, and similarity analysis using different approaches such as data-mining, graph-based analysis, and AI-based learning techniques, to name a few.

For instance, Cozzi et al. [63] leveraged about 90K IoT malware from VirusTotal to perform binary malware analysis and explore the lineage of IoT malware families, and track their relationships, and evolution. Alasmary et al. [64] leveraged malware control flow graphs (CFGs) to demonstrate that IoT malware samples have richer flow structure and higher complexity, which can be utilized for effective graph-based detection using deep learning classifiers. Moreover, strings-based malware analysis, which relies on extracting meaningful text-based indicators from the binary files, have been leveraged for malware analysis. For instance, Alhanahnah et al. [65] leveraged N-Gram strings for correlating and clustering IoT malware samples based on their strings similarities. In addition, recent work has shown promising results in terms of AI-based malware detection and classification accuracy by integrating multiple outcomes of the static malware analysis. For instance, Gibert et al. [30] introduced a multi-modal deep learning approach for effective malware classification by combining three separate static analysis outcomes (API calls, sequence of assembly code mnemonics, and the sequence of bytes).

While previous research showed promising results in terms of effective malware detection, characterization, and classification, the deployed techniques come with a number of limitations. For instance, AI-based approaches require feature engineering/learning, which is not a straightforward task and requires domain knowledge along with multi-level processing steps that can be costly. In addition, there is no one-size-fits-all approach that can serve security analyst towards achieving their objectives, and thus, different techniques can be employed accordingly and complementary.

Along this line of research, Haq et al. [66] surveyed more than 60 static malware analysis and binary code similarity approaches that were presented in the past 20 years and highlighted a range of strengths, weaknesses, and open problems. Interestingly, while the majority of the surveyed approaches were focused on extracting various features from the binary code, almost none of them focused on static strings-based analysis. This raises the question of reliability of such approach as a core malware analysis techniques. However, in this thesis, we showed that strings-based analysis can be in fact used as a discount approach for fast and reliable IoT malware analysis.

Moreover, to the best of our knowledge, we are among the few researchers who performed such

a large-scale characterization of IoT malware. Furthermore, while the recently published work by Cozzi et al. [63] presents a series of IoT-related assumptions and characterization analysis results similar to our work, the main difference between our work is that we solely rely on strings-based malware analysis as an effective and lightweight approach, while their analysis was based on raw binary data and/or available source code. Moreover, while they provide a systematic approach for studying malware lineage, we on the other hand, leverage the extracted malware strings to explore and characterize adversarial IP-based networks and the correlation of the underlying malware samples. Finally, we studied a dataset of recently detected IoT malware samples by a specialized IoT honeypot, which was shown to be more effective in detecting possibly new IoT malware samples. Therefore, our analysis can provide a more recent view of the rapidly evolving and changing IoT malware threat landscape.

2.3 Leveraged Datasets

In this thesis, we leveraged a number of resources to obtain various datasets for further empirical data analysis and experimentation purposes. We summarize the leveraged datasets throughput different chapters in Table 2.1. More specifically, we relied on the CAIDA's network telescope [67] to obtain several instances of passive traffic and Internet measurements, which were used in Chapters 3–5. Additionally, we obtained two instances of IoT device information from Shodan [45], which provided information about deployed devices in both the consumer and CPS realms (leveraged in Chapters 3–5). Finally, we utilized IoTPOT [6], which is a specialized IoT honeypot, to obtain recently detected IoT malware binaries and ELF files. We use these IoT malware dataset in Chapter 6.

Table 2.1: A list of leveraged datasets and resources throughout this thesis.

Data	Source	Size	Collection Dates	Description	Chapter
D1	CAIDA [67]	5TB	April 12–18, 2017	141M packets generated by 27K CPS and Consumer IoT devices	Ch3–4
D2	CAIDA [67]	6TB	April 12–18, 2017 & May 21–25, 2018	161M packets generated by 8K Consumer IoT devices	Ch4
D3	CAIDA [67]	4TB	November 1–5, 2018	308M packets generated by 28K Consumer IoT devices	Ch5
S1	Shodan [45]	200GB	April, 2017	331K CPS and Consumer IoT device information	Ch3–4
S2	Shodan [45]	200GB	November, 2018	400K Consumer IoT device information	Ch5
H1	IoTPOT [6]	4.4GB	September, 2018–May, 2020	49K IoT malware binaries and ELF files	Ch6

Chapter 3

Inferring, Characterizing, and Investigating Internet-Scale Malicious IoT Device Activities: A Network Telescope Perspective

3.1 Overview

Internet of Things (IoT) devices have been widely adopted in various parts of our lives. IoT devices and corresponding technologies facilitate efficient data collection, monitoring, and information sharing for consumers (e.g., Internet routers, smart TVs, health monitoring wearables), and Cyber-Physical Systems (CPS) (e.g., power utilities, manufacturing plants, factory automation) [68]. These IoT devices have brought many benefits to individual users and industries, including but not limited to: efficient data collection, monitoring, and information sharing capabilities through the Internet [68]. Despite their benefits, the always-connected nature of IoT devices and the

The work has been presented at the 48^{th} IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2018) [2].

inadequate security measures implemented by some manufacturers [69], have turned these devices into attractive targets for cyber adversaries. Moreover, the wide adoption of IoT devices by consumers and industries made Internet users more vulnerable to large-scale cyber attacks that could reach a wider range of targets through the connected IoT devices [69, 70].

In fact, the increasing number of IoT devices have contributed toward the significance of large-scale cyber attacks by increasing the attack surface to include new targets. Large-scale cyber attacks, which can lead to service disruption, denial of service (DoS), security threats, and privacy leakage, have become a major source of concern in recent years [69, 70]. These orchestrated attacks could lead to service disruption, denial of service (DoS), security threats, and privacy leakage for people, organizations, and governmental agencies. For instance, attacks on the Ukrainian power centers caused persistent damage to the power grid that was providing electricity to hundreds of thousands of consumers [71]. More recently, unverified WikiLeaks documents reported that the U.S. Central Intelligence Agency (CIA) was working on a surveillance program that utilized some technical bugs in specific smart TV models to hack them and create a back door for spying on people by recording their conversation when necessary [72].

Furthermore, IoT devices could be used as enablers for orchestrating large-scale attacks towards a variety of targets. The `Mirai` botnet for instance, utilized millions of compromised IoT devices (e.g., CCTV cameras) to launch Distributed Denial of Service (DDoS) attacks on several DNS servers, resulting in service disruption for millions of Internet users across the globe [3]. Very recently, the `Reaper` botnet extended `Mirai` by exploiting IoT-specific vulnerabilities rather than simply guessing credentials [20].

In order to adopt proper mitigation measures and prevent large-scale, IoT-related cyber attacks, security researchers and operators need to assess the magnitude of Internet-scale IoT exploitations, in addition to characterizing and analyzing their malicious activities. Nevertheless, given the lack of empirical data related to IoT devices [37], in addition to their excessive Internet-wide deployments in consumer and CPS, there is an utmost need to explore data-driven methodologies to shed the light and comprehend the characteristics of such compromised IoT devices and their malicious behaviors. To address the lack of knowledge about compromised IoT devices, there is a need to posses an Internet-scale perspective of IoT devices and their unsolicited activities over a period of

time. This indeed is quite challenging as it requires authorization from different entities who own and operate these IoT devices in their local realms. Furthermore, monitoring IoT traffic would come with underlying privacy implications. Moreover, there are tremendous variants of IoT devices operating from all around the world and monitoring them would require scalable systems and significant resources.

An effective approach to gain Internet-wide cyber threat intelligence is to study passive measurements gathered using designated sensors or traps that collect traffic from the Internet [49, 50]. These sensors collect traffic targeted towards routable, yet unused Internet Protocol (IP) addresses, which are known as darknets or network telescopes [13]. Characteristically, traffic destined to these inactive hosts is likely to represent suspicious and unsolicited activities. Furthermore, traffic captured at the darknet mainly consists of scanning [16, 73], backscatter traffic resulting from DDoS attacks [15, 18, 74], and misconfiguration [12, 13]. Therefore, by carefully studying darknet traffic, one can generate useful insights on a portion of unsolicited traffic related to different sources including compromised machines (e.g., malware-infected) and victims of DDoS attacks, to name a few.

3.2 Contributions

To this end, we aim at addressing the problems of inferring Internet-scale compromised IoT devices and analyzing their unsolicited/malicious activities by exploring auxiliary, macroscopic, empirical passive darknet data obtained from a large network telescope. Specifically, we frame the contributions of this work as follows:

- We draw-upon close to 5TB of recent darknet data and execute correlations with a near real-time IoT database to empirically characterize the magnitude of Internet-scale IoT exploitations in both, consumer and critical CPS realms. The generated insights not only render a first attempt ever to empirically shed the light on the large-scale insecurity of the IoT paradigm, but are also intended to contribute to operational/actionable cyber security by providing Internet-wide, IoT-tailored notifications of such exploitations, thus permitting rapid remediation.

- We execute a first-of-a-kind, large-scale empirical characterization and analysis of IoT-centric

unsolicited activities as perceived by a large network telescope. To this end, we uncover the nature of such traffic, its sources, employed protocols, targeted ports, upon various others. Given the lack of IoT-specific attack signatures, we postulate that the analyzed traffic from this work could be leveraged to design such signatures, in addition to promoting and facilitating further IoT-tailorted forensic investigations by making the captured unsolicited empirical traffic available to the research and operations communities at large.

- Motivated by the rise of new malware families/variants that specifically target and exploit IoT devices such as `Persirai, Hajime` and `BrickerBot`, to name a few, we execute non-intrusive correlations between passive measurements and malware threat intelligence to uncover new, previously unreported malware families targeting the IoT paradigm. In this context, we explore a publicly available threat repository and an in-house built malware database facilitated by instrumenting a large corpus of malware samples in a controlled sandbox. The results not only alarm about the severity of this malware issue in the context of the IoT, but also paves the way for future work for addressing the rise of IoT-centric, orchestrated botnets.

3.3 Identifying Unsolicited Internet-Scale IoT Devices

We initiate our work by addressing the problems of identifying and characterizing Internet-scale unsolicited IoT devices. We refer to IoT devices as being unsolicited (or compromised) if they were found to be generating any network packets towards the network telescope. Please note that Section 3.4 will detail the nature of such unsolicited traffic and provide an in-depth characterization of its modus-operandi. We herein initially elaborate on the employed datasets and subsequently provide the methodology and results towards the goal of inferring compromised IoT devices.

3.3.1 Obtained Data

IoT Device Information

It is quite difficult, if not impossible, to obtain technical information related to Internet-wide IoT devices that have been deployed in consumer and CPS environments due to privacy and logistic

reasons. In addition, there is a lack of knowledge about effective fingerprinting approaches for identifying IoT devices by solely observing network traffic. Considering these challenges however, in this work, we leverage a near real-time IoT database provided by `Shodan` [45]. This service executes large-scale active measurements to identify and index Internet-facing IoT devices.

To this end, we obtained information related to 331,000 IoT devices from `Shodan`. These IoT devices, which were deployed in more than 200 countries all around the world, belong to consumer and CPS realms. On one hand, consumer IoT devices represent wireless access points and routers, IP cameras (e.g., webcams and CCTV cameras), printers, network storage media, satellite TV box and digital video recorders (DVRs), and electric hubs/outlets. On the other hand, IoT devices in CPS realms are involved in monitoring, controlling, and managing industrial/automation operations. They represent programmable logic controllers (PLC), remote terminal units (RTU), or other smart equipment that are used in industrial control systems (ICS), supervisory control and data acquisition systems (SCADA), and/or distributed control system (DCS). We obtained information related to approximately 181,000 consumer IoT devices, including, routers (46.9%), printers (29.1%), IP cameras (18.3%), and network storage media (4.6%). The remaining consumer IoT devices accumulate to only 1.1% of the total devices. We also obtained data related to 150,000 IoT devices in CPS that supported 31 industrial/control automation protocols/services. These CPS devices belong to a number of industries including but not limited to: building automation, power generation and distribution, control systems, plant/factory automation, oil and gas transportation, and embedded IoT communications.

As depicted in Figure 3.1, the U.S. hosted the largest number of IoT devices (25%), followed by a significantly less number of devices hosted in the U.K. (6%), Russia (5.9%), and China (5%), respectively. Furthermore, by looking at the top 15 countries with the most number of IoT devices (Figure 3.1), which account for about 69% of all IoT devices, we noticed that the number of consumer IoT devices were relatively higher than those deployed in CPS for the listed countries except for China, France, Canada, Vietnam, Taiwan, and Spain.

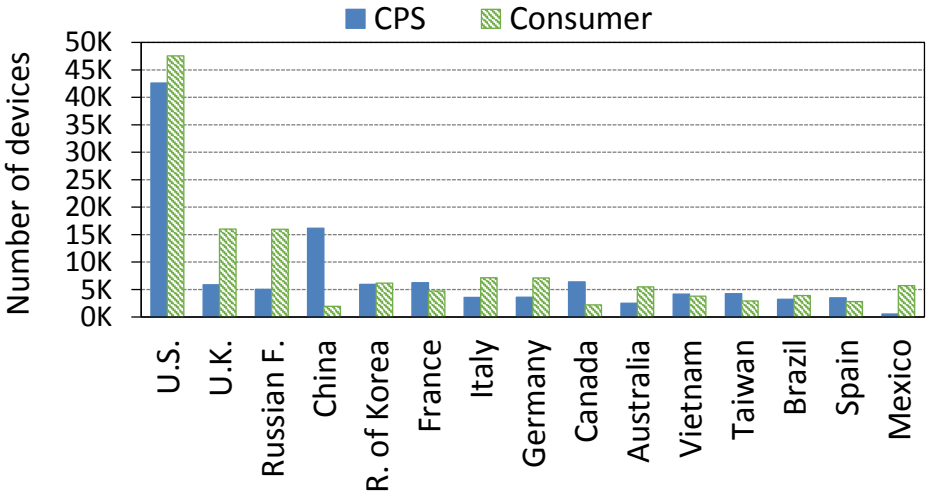


Figure 3.1: Countries with the largest number of deployed IoT devices.

Network Telescope Data

Darknet data consists of one-way traffic targeted towards routable, allocated yet unused IP addresses (dark IP addresses). Since these IP addresses are not bound to any services, any traffic targeting them is characteristically unsolicited [12, 13]. Typically, darknet data consists of scanning, backscatter, and misconfiguration traffic [12–16, 74]. We explored over 5TB of darknet traffic between April 12-18, 2017 (about 80 GB of daily traffic). The darknet traffic is obtained from the UCSD real-time network telescope data maintained by the Center for Applied Internet Data Analysis (CAIDA) [67]. It is one of largest available sources of passive darknet traffic with about 16.7 million globally routed destination IPv4 addresses (i.e., /8 network) capturing over a billion packets every hour. The processed darknet traffic is stored in "flowtuple" files. Each file represents incoming flows towards the darknet that consist of the following flowtuple information: source/destination IP addresses and used ports, protocol, time to live (TTL), TCP flags, IP length, and total number of packets. The daily darknet traffic consists of unique compressed files representing hourly traffic (maximum of 24 files per day). We found that the available data for April 18 was incomplete, with only 15 hours of collected traffic (data might be missing due to technical issues at the telescope). To maintain consistency, we decided to remove the incomplete data from further analysis throughout the work, resulting in 143 hours of darknet data that was obtained between April 12-17, 2017.

3.3.2 Inferring and Characterizing Unsolicited IoT Devices

To infer compromised IoT devices, we executed a correlation algorithm that leverages IP header information to associate the obtained IoT device information with darknet flows. A significant 26,881 IoT devices were found interacting with the darknet, representing relatively more compromised consumer IoT devices (57%) than CPS (43%). As shown in Figure 3.2, slightly over 12,000 (46%) unsolicited IoT devices were correlated with the darknet data at the first day of the analysis (April 12, 2017). For the remaining time period, we discovered an average of about 2,900 newly compromised IoT devices per day. Considering the overall and cumulative numbers of uncovered unsolicited IoT devices, we definitely anticipate that an extended analysis period would result in discovering even more compromised IoT devices.

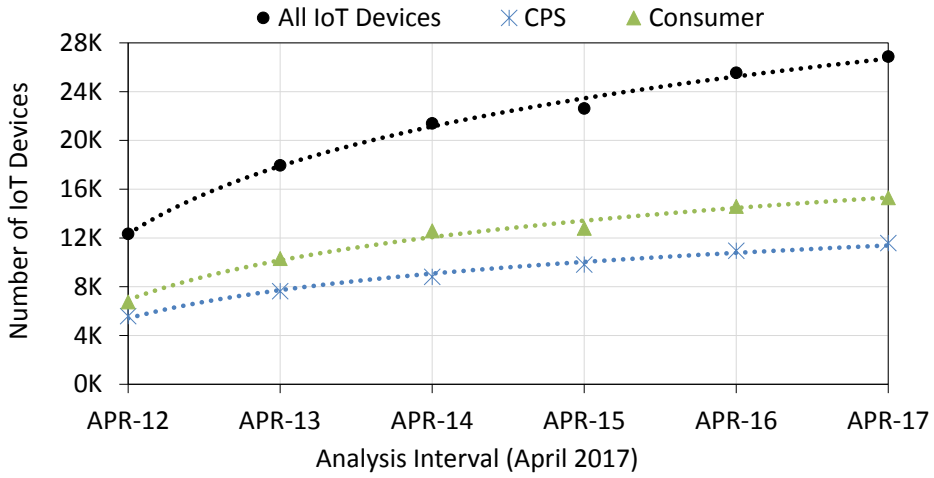


Figure 3.2: The cumulative number of daily discovered compromised CPS and consumer IoT devices at the darknet over the 6 days analysis interval.

The compromised IoT devices were located across 161 countries, with the largest number of devices to be hosted in Russia (24.5%), followed by China (8.6%), and the U.S. (8.1%), respectively (Figure 3.3). It is worth noting that while the U.S. and the U.K. hosted more number of IoT devices as compared to Russia and China (Figure 3.1), the latter countries were found to host a relatively higher number of unsolicited IoT devices, as illustrated in Figure 3.3. Furthermore, while Thailand, Indonesia, Singapore, Turkey, Ukraine, and India were not listed among the top 15 hosts with the most deployed IoT devices (Figure 3.1), it is interesting to find them among the top 15 countries with

the most number of uncovered compromised IoT devices. In fact, Figure 3.3 illustrates a significant difference in the percentage of unsolicited IoT devices found in Russia (31%) and Ukraine (30%), as compared to countries such as the U.S. (2.4%) and the U.K. (2.5%). While the actual reason behind this significant difference is quite obscured, this might indicate the enforcement of a stronger and more effective IoT security measures and policies in the U.S. and the U.K. in comparison to other countries.

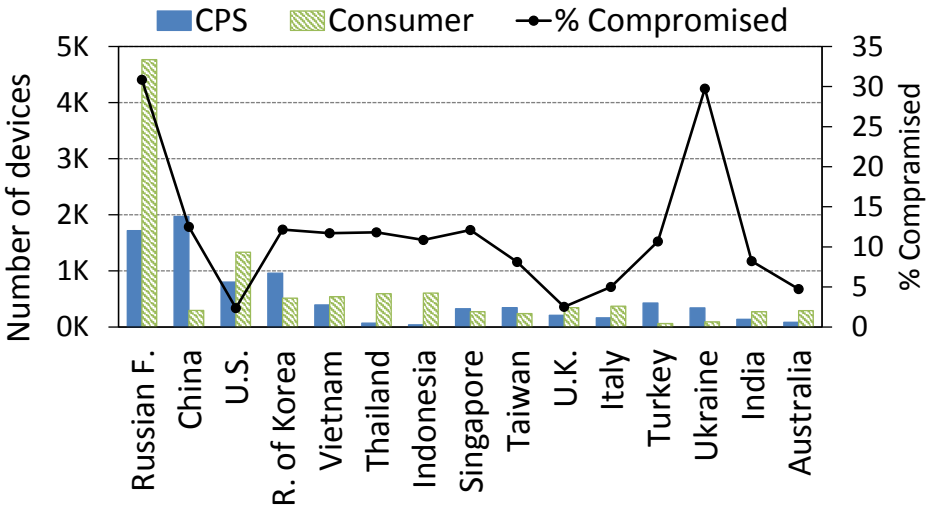


Figure 3.3: Countries with the largest percentage of compromised IoT devices.

Compromised IoT Devices in Consumer Realms

We identified 15,299 unsolicited consumer IoT devices that were correlated with the darknet over the analysis period. These IoT devices were located across 145 countries, with Russia hosting the highest percentage of compromised consumer IoT devices (32%), followed by the U.S. (9%), Indonesia (4%), and Thailand (4%), respectively. These IoT devices were connected to the Internet via 1,762 different Internet Service Providers (ISP), with the Russian "JSC ER-Telecom" hosting the highest percentage of compromised consumer IoT devices (27.6%), as summarized in Table 3.1. In addition, about 52.4% of compromised consumer IoT devices were Internet routers, followed by IP cameras (25.2%), printers (18%), and network storage media (3.6%), respectively.

As illustrated in Figure 3.4, these aforementioned devices accounted for about 99.4% of all consumer IoT devices, while TV boxes/DVRs and electric hubs/outlets represented less than 0.6%

Table 3.1: Top 5 ISPs hosting the highest number of compromised consumer IoT devices.

ISP	Country	Devices	%
JSC ER-Telecom	Russian F.	4,205	**27.6**
PT Telkom	Indonesia	542	3.6
Korea Telecom	R. of Korea	339	2.2
PLDT	Philippine	311	2.0
TOT	Thailand	277	1.8

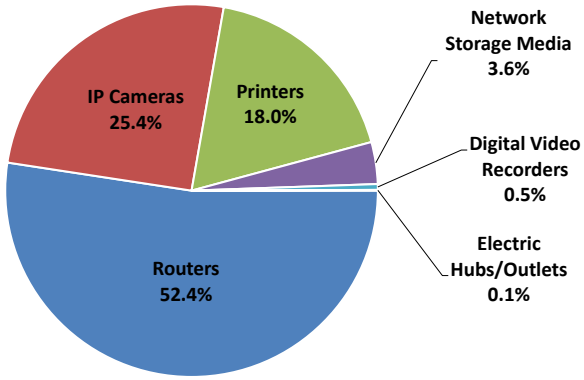


Figure 3.4: Percentage of compromised consumer IoT devices by type/category.

of all compromised consumer devices.

Compromised IoT Devices in CPS Realms

We identified 11,582 compromised IoT devices in CPS environments that were located in 136 countries, with China, Russia, Korea, and the U.S. hosting about 17%, 14.8%, 8.3%, and 6.9% of all the compromised devices respectively. The IP addresses of these devices were associated with 2,279 different ISP across the identified countries. As pinpointed in Table 3.2, "Rostelecom" hosts the highest percentage of compromised IoT devices (about 4%), followed by "Korea Telecom" (3.8%) and "Turk Telekom" (3.2%), respectively.

Furthermore, a range of 31 services/protocols were operated by such compromised IoT devices. These services are not mutually exclusive, and therefore, an IoT device in a specific CPS might support one or more of these services. The top 10 operated services/protocols by the most number of unsolicited IoT devices are summarized in Table 3.3. Among all the supported services/protocols,

Table 3.2: Top 5 ISPs hosting the highest number of compromised IoT devices in CPS realms.

ISP	**Country**	**Devices**	**%**
Rostelecom	Russian F.	461	**4.5**
Korea Telecom	R. of Korea	429	3.8
Turk Telekom	Turkey	347	3.2
HiNet	Taiwan	261	2.5
JSC ER-Telecom	Russian F.	277	1.8

Table 3.3: Top 10 CPS realms hosting compromised IoT devices.

Service/Protocol	**Common applications**	**Devices**	**%**
Telvent OASyS DNA	Oil and Gas transportation pipelines and distribution networks	**2,328**	**20.0**
SNC GENe	Control systems	2,126	18.3
Niagara Fox	Building automation systems	1,554	13.4
MQ Telemetry Transport	IoT communications, sensory networks, safety-critical communications	1,497	12.9
Ethernet/IP	Manufacturing automation	1,490	12.8
ABB Ranger	Power generating plants, transmission lines, mining operations, and transportation systems	1,061	9.1
Siemens Spectrum PowerTG	Utility networks	685	5.9
Modbus TCP	Power utilities	639	5.5
Foxboro/Invensys Foxboro	Plant automation systems, flowmeters, single-loop controllers, and product support services	590	5.1
Foundation Fieldbus HSE	Plant and factory automation	354	3.0

Telvent OASyS DNA (20%), which operates in critical oil and gas CPS, and *Niagara Fox* (13.4%), which is common in building automation systems, appear among the most prevalent. CPS hosting compromised IoT devices also include those related to power utilities and manufacturing plants. Having noted this, it is indeed alarming (to say the least) to infer over 11,000 compromised IoT devices operating in such critical and error-sensitive environments.

3.4 Characterizing Unsolicited Traffic From Internet-Scale IoT Devices

The aim of this section is to dissect, thoroughly comprehend, and characterize the unsolicited traffic generated by the inferred compromised IoT devices as perceived by the network telescope. We observed about 141.3M packets that were sent to the darknet from the 26,881 compromised IoT devices (daily $mean = 23.5$M and $\sigma = 0.92$M packets). On average, we captured 10,889 unsolicited IoT devices generating traffic towards the darknet on a daily basis, with slightly larger number of active consumer IoT devices (53%) on a daily basis. In general, consumer IoT devices, which represent 57% of all compromised IoT devices, generated more packets towards the darknet as compared to compromised devices in CPS realms, with approximately 62M packets (daily $mean =$ 10M and $\sigma = 1.01$M), and 50M packets (daily $mean = 8.3$M and $\sigma = 1.05$M) for each device type respectively.

Considering the critical CPS contexts in which the compromised IoT devices operate in, it is worrisome to observe their aggressive role in generating significant amount of unsolicited activities. Interestingly, the statistical analysis using a Mann-Whitney U test indicated that the number of packets generated towards the darknet was significantly greater for devices in CPS than for consumer IoT devices ($p < 0.0001$). The higher activities however, might be attributed to the nature of the compromised IoT devices in CPS, which might have access to more powerful processing capabilities, as compared to other IoT devices, which typically have limited processing and memory resources. By contrast, the lower activity rate of compromised consumer IoT devices might be due to the stealthy nature of their generated activities, which aim at maximizing reachability while attempting to avoid detection. In what follows, we further explore the natures and characteristics of unsolicited traffic that have been generated by Internet-scale compromised IoT devices.

3.4.1 Unsolicited UDP Traffic

The analyzed UDP packets represent about 10.4% of all traffic generated by the unsolicited IoT devices, with slightly more UDP packets generated by compromised consumer IoT devices as compared to those in CPS, as illustrated in Figure 3.5. Indeed, it is well known that UDP packets

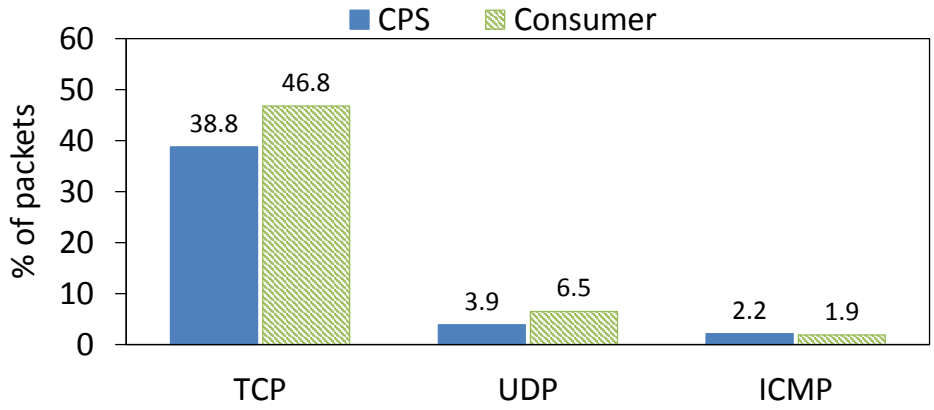


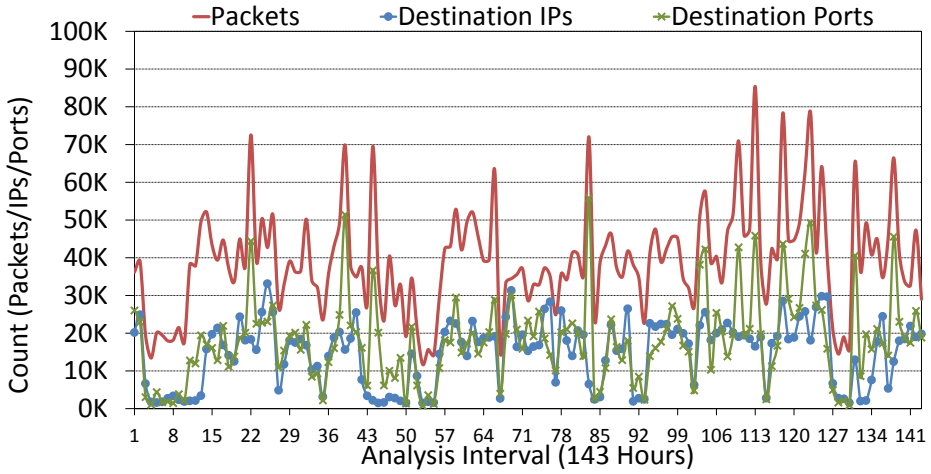
Figure 3.5: Percentage of TCP, UDP, and ICMP traffic generated by compromised IoT devices in CPS and consumer realms.

have been used to scan the Internet for open ports/services [16, 75], in addition to being employed to perform DoS attacks by flooding destination IP addresses or by exploiting open resolvers, causing amplification DoS attacks [59]. Thus, due to the stateless nature of UDP packets, it is quite challenging to classify them into a specific traffic category without further packet inspection. To maintain the focus of this work, we do not address this challenging objective herein, though we will explore methodologies similar to [18] in future work to achieve this task. Nonetheless, to gain insights related to IoT-generated UDP traffic, we provide an overall characteristic analysis of the observed UDP packets in the following sub-sections.

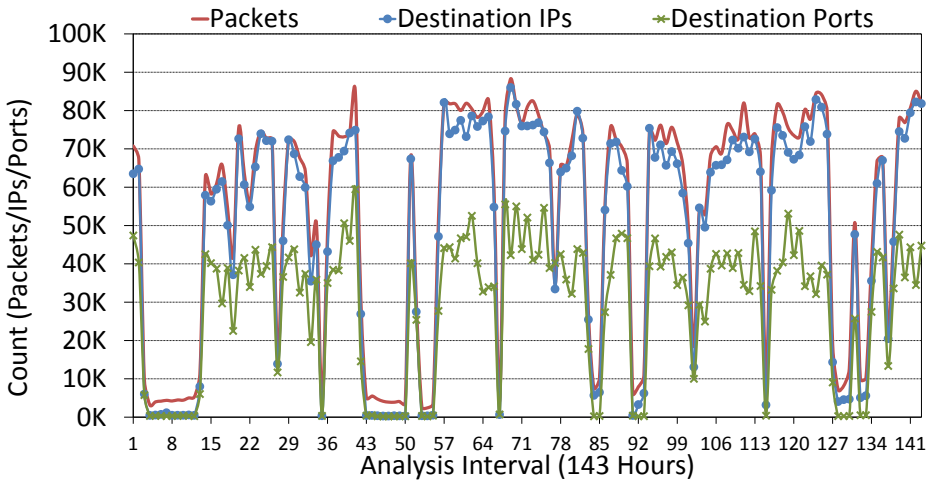
UDP Packets

Overall, we observed about 13M UDP packets generated by a total of 25,242 compromised IoT devices, among which, about 60% were compromised consumer IoT devices, generating 63% of all UDP packets. Such IoT devices also targeted a significantly higher number of ports and destination IP addresses on an hourly average, as compared to those compromised IoT devices deployed in CPS (Figure 3.6). More specifically, compromised consumer IoT devices targeted an average of about 29,000 ports on more than 48,000 destination addresses, while compromised IoT devices in CPS targeted less ports (about 18,000) on significantly less number of destination addresses (14,700).

The comparison of the overall behavior of compromised IoT devices in CPS and consumer



(a) CPS



(b) Consumer

Figure 3.6: Overall UDP packets sent by compromised (a) CPS and (b) consumer IoT devices to destination IP addresses and ports.

realms in terms of the generated UDP packets, and targeted destination addresses and ports, illustrates a number of differences. First, the compromised consumer IoT devices were actively sending UDP packets during repeated intervals that lasted for longer hours than those compromised in CPS. Second, the total number of generated UDP packets per hour by compromised consumer IoT devices was very close to the total number of targeted destination IP addresses (Figure 3.6(b)), and therefore, very few packets were sent towards each destination IP. We also found a strong positive correlation between the number of targeted ports and destination IP addresses by compromised consumer IoT devices (Pearson's correlation $r = 0.95$ and $p < 0.0001$), which may indicate the effort

Table 3.4: Top 10 targeted UDP protocols/ports.

Protocol/Port	Packets (K)	%	Devices
Not Assigned/37547	329.6	2.52	10,115
NetBIOS/137	269.9	2.06	144
Not Assigned/53413	268.1	2.05	91
Not Assigned/32124	141.2	1.08	9,488
Not Assigned/28183	122.5	0.94	9,710
mDNS/5353	99.4	0.76	165
Not Assigned/4605	50.3	0.38	150
DNS/53	42.6	0.33	158
Teredo/3544	34.4	0.26	226
OpenVPN/1194	34.0	0.26	96

of such devices to reach a wider range of new destination IP addresses on various ports at each interval. Finally, the compromised CPS devices generated a significantly larger number of UDP packets per hour towards the targeted destinations, with packets possibly sent to a relatively larger number of ports on the same destinations, as illustrated by the recurring spikes in the number of contacted destination ports per hour (Figure 3.6(a)).

UDP Ports

The compromised IoT devices generated UDP packets towards all available UDP ports (65,535). About 10.7% of all UDP packets were targeting the top 10 ports (Table 3.4), while the remaining packets (89.3%) were distributed among over 60,000 ports. As shown in Table 3.4, port 37547 received about 329,000 UDP packets (2.5% of all), followed by port 137 (NetBIOS) and port 53413 with 2.06% and 2.05% of all UDP traffic respectively. In addition, while destination ports 37547, 32124, and 28183 were targeted by more than 9,000 compromised IoT devices, the remaining ports received UDP packets from significantly less number of compromised IoT devices (Table 3.4). We identified 5 assigned (well-known) services/protocols that correspond to the top 10 targeted ports. Nevertheless, although the remaining ports were not officially assigned to any services/protocols, some of them are known to be associated with known vulnerabilities. For instance, port 37547 has been associated with a backdoor to exploit and control "Netcore/Netis" routers [76].

3.4.2 Unsolicited Backscatter Traffic

Backscatter traffic, in the context of this work, is a byproduct of (D)DoS attacks that target IoT devices. When a victim IoT device is attacked by a flood of packets generated from spoofed source IP addresses (that happened to be belonging to the network telescope IP space), the device will generate reply packets destined to the darknet, which can then be collected and extracted. These packets are mainly TCP (SYN-ACK and RST) or ICMP reply packets (Echo Reply, Destination Unreachable, Source Quench, Redirect, Time Exceeded, Parameter Problem, Timestamp Reply, Information Reply, or Address Mask Reply) [18].

Our analysis revealed a total of 839 IoT devices that have fallen victims of DoS attacks, with about 10.3M backscatter packets generated towards the darknet (8.2% of total traffic). Approximately half of the victim IoT devices generated less than 170 backscatter packets towards the darknet, while about 17% of the IoT devices generated 10,000 or more backscatter packets (Figure 3.7). Moreover, only 7 devices generated 100,000 or more backscatter packets, among which 5 of them were likely to be operating in critical CPS realms. In general, about 73% of all backscatter packets were generated by IoT devices in CPS, which represent slightly more than half of the DoS victims (53%). In fact, 5 IoT devices in CPS contributed to about 43% of all backscatter traffic, with two devices that generated about 1.1 and 3.4 million packets respectively. These observations may reflect the nature of the inferred DoS attacks that were focused on target devices in CPS with higher intensity as compared to consumer IoT devices.

IoT DoS Victims

By investigating the distribution of backscatter packets as illustrated in Figure 3.8, we observed few instances with noticeable increase in the number of generated backscatter packets by the IoT devices (e.g., between intervals 6 and 8). These sudden spikes indicate a large magnitude of DoS attacks against CPS and consumer IoT devices during the specified time intervals. It is apparent that IoT devices in CPS realms were attacked more often and with higher intensity as compared to consumer IoT devices. In fact, a conducted Mann-Whitney U test showed a statistically significant difference between the number of generated backscatter packets per hour when comparing IoT

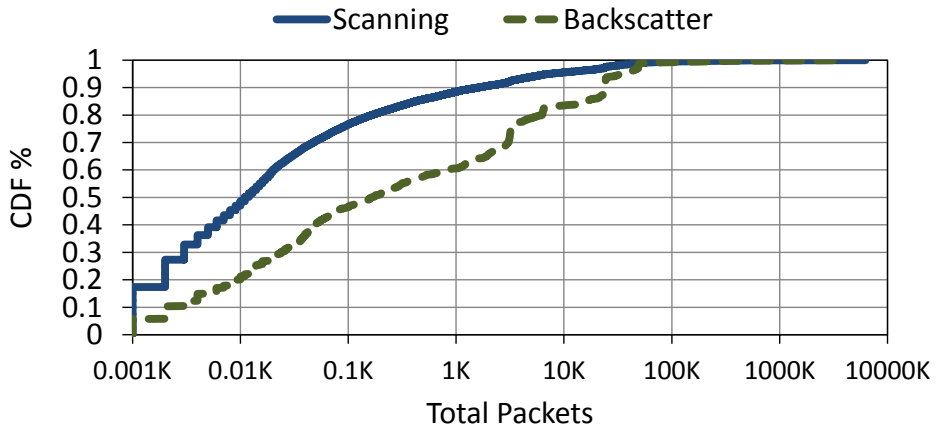


Figure 3.7: Distribution of the scanning and backscatter packets generated by compromised IoT devices and DDoS victims, respectively.

devices in CPS and consumer realms ($p < 0.0001$, $U = 6061$, and $Z = -5.95$).

To further investigate targeted IoT devices as DoS victims, we focused at intervals with sudden spikes in the number of backscatter packets. Interestingly, a single victim IoT device generated almost all the packets during every DoS attack interval. For instance, an IoT device in a CPS realm located in China was responsible for more than 99% of all backscatter traffic during intervals 6-8 and 53-55, and about 89% of traffic at interval 56. Similarly, a different CPS device from China was found to be under DoS attacks during intervals 99 and 127, generating about 91% and 97% of all backscatter traffic at those intervals respectively. Both of the aforementioned IoT device operated Ethernet/IP on TCP/UDP port 44818, which is used in manufacturing automation. After some investigations, we inferred that this service is associated with "Rockwell Automation Control Logix PLC" vulnerabilities, which can cause DoS on the targeted IoT devices.[1] Finally, an IoT device in a CPS from Switzerland, which supports Telvent OASyS DNA (used in oil and gas transportation pipelines and distribution networks), contributed towards about 85% of the backscatter traffic at interval 94, indicating another instance of targeted DoS attacks.

Analyzing the DoS events for the consumer IoT devices resulted in similar behavior as the CPS devices. For instance, a printer located in the Netherlands, generated over 104,000 backscatter packets at interval 49, contributing towards 98% of all packets at this interval. In addition, another printer from the U.K. was found to be under targeted DoS attacks as it generated about 85% of all

[1] https://ics-cert.us-cert.gov/advisories/ICSA-13-011-03

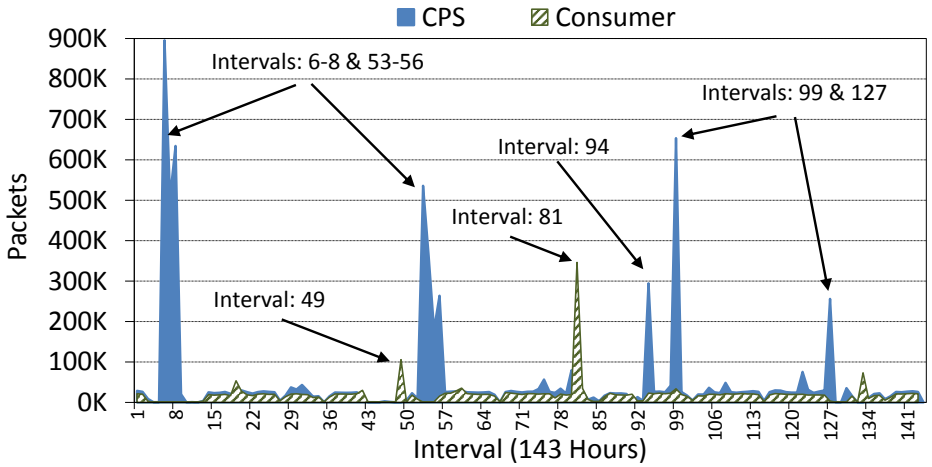


Figure 3.8: Distribution of the generated backscatter packets by CPS and consumer IoT devices (143 hours).

backscatter packets at interval 81.

Targeted Countries

The targted IoT devices were found to be located in 80 countries, with China, Singapore, and the U.S. hosting the highest number of DoS IoT victims, as shown in Figure 3.9. Moreover, China and the U.S. hosted the most number of targeted IoT devices in CPS realms (103 and 49 devices, respectively), while Singapore and Indonesia hosted the highest number of consumer IoT device victims (64 and 52 devices). From a different perspective, about 52% of all backscatter traffic was generated by IoT devices hosted in China, followed by devices in the U.S. (5.9%) and the U.K. (4.1%), respectively. In addition, we noticed that the U.K, Brazil, Switzerland, and Argentina, were among the top 15 countries with the highest number of generated backscatter packets (Figure 3.10), while hosting relatively few victim IoT devices (10, 16, 4, and 5 devices, respectively). This corroborates our previous findings regarding the nature of the observed DoS attacks during the analysis intervals, which represent intensive targeted attacks on a small number of victim IoT devices.

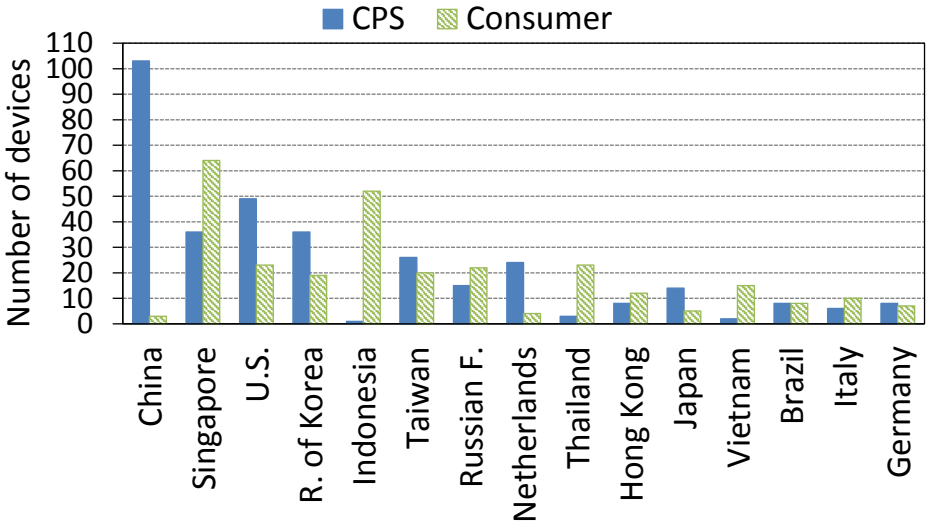


Figure 3.9: Countries with the largest number of DoS victims.

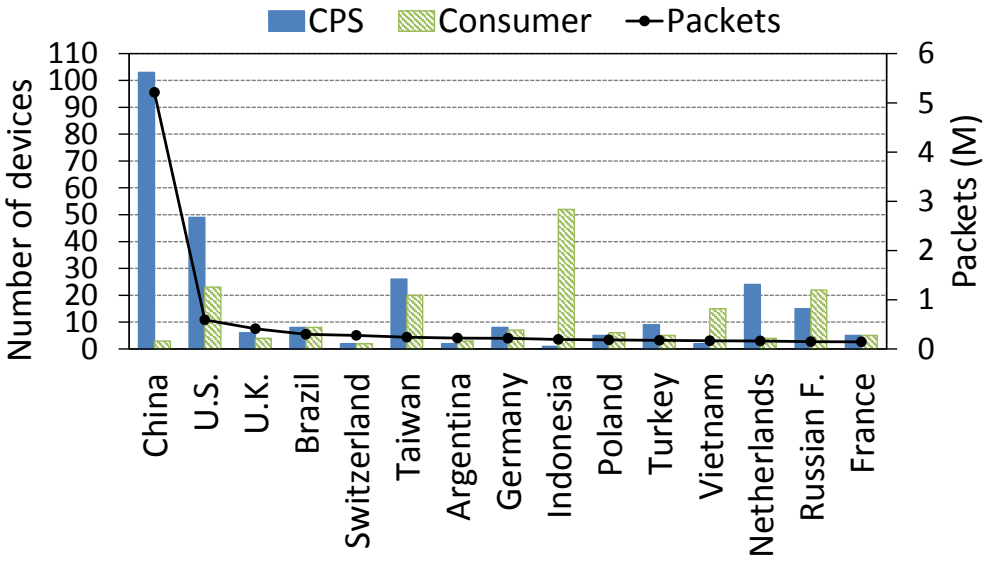


Figure 3.10: Countries with the largest number of backscatter packets.

3.4.3 Unsolicited Scanning Traffic

Probing traffic generated from unsolicited IoT devices that target the network telescope is an indicator of exploitations of such IoT devices. Such compromised devices would typically be scanning the Internet looking to exploit vulnerable hosts or other IoT devices. In order to identify IoT-generated scanning traffic, we first looked at the remaining non-backscatter ICMP packets,

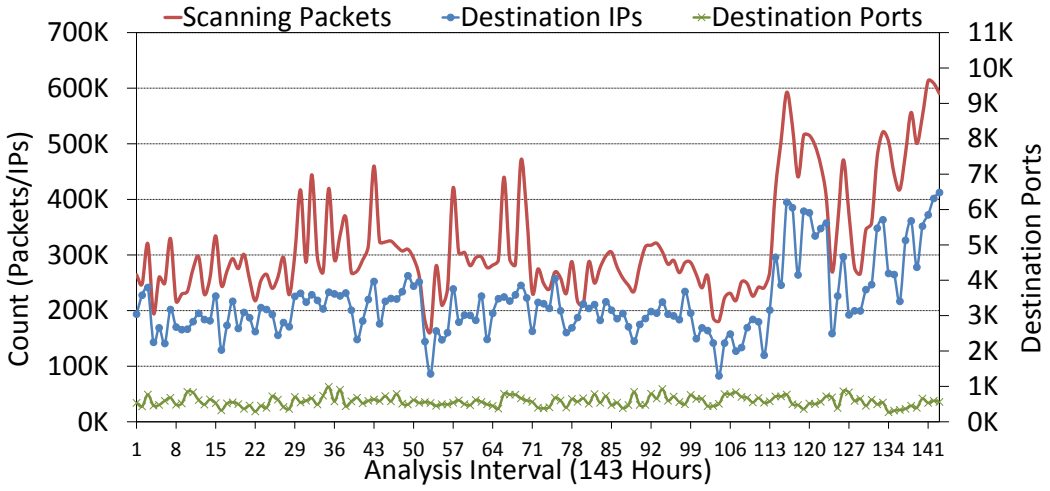
which represented a very small percentage of the total generated traffic by compromised IoT devices (0.23%). More than 99.9% of these packets were ICMP Echo requests, which are typically used for remote network scans (e.g., ping). Moreover, these packets were originated from 56 exploited IoT devices, among which 32 consumer IoT devices generated the majority of the ICMP scanning packets (93%).

We also identified slightly over 100M TCP packets that were not classified as backscatter. These TCP packets were mainly TCP-SYN packets (99.97%), which are commonly used for scanning the Internet [16]. The TCP scanning packets were generated by a total of 12,363 compromised IoT devices (55% consumer IoT devices). We illustrate the overall distribution of the TCP scanning packets generated by compromised IoT devices in both, CPS and consumer realms in Figure 3.11.

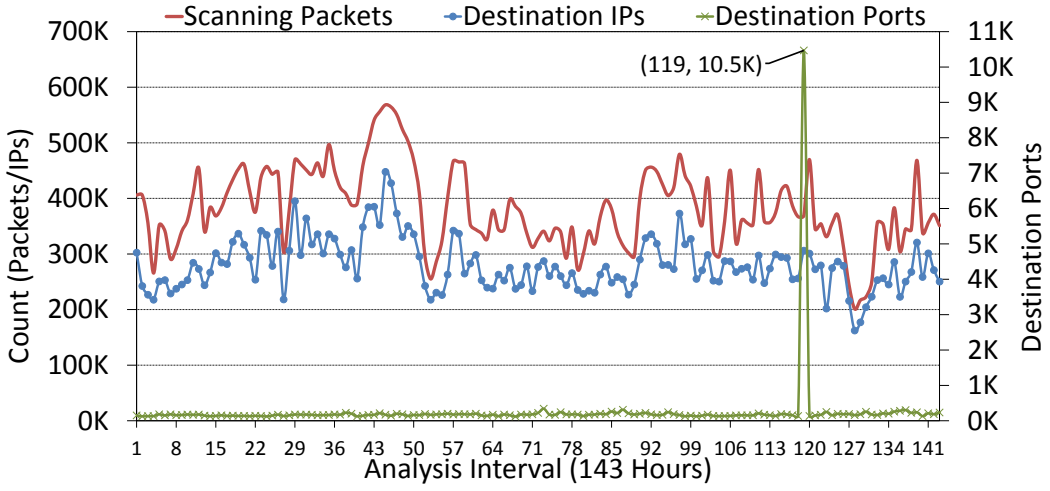
On average, exploited consumer IoT devices generated more TCP scanning packets per hour, as compared to exploited IoT devices in CPS, with about 382,000 and 318,000 packets for each device type respectively. Nevertheless, while the compromised IoT devices included more consumer IoT devices (55%), the analysis showed no linear correlation between the number of compromised IoT devices and the total generated scanning packets per hour (Pearson's $r \approx 0$ and $p > 0.05$). In addition, exploited consumer IoT devices targeted relatively more destinations per hour, as compared to those deployed in CPS, with an average of 280,000 and 215,000 destinations respectively. Interestingly, while the exploited IoT devices in CPS scanned relatively less number of destinations, they seemed to be scanning a wider range of destination ports as compared to consumer IoT devices, with an average of 576 scanned ports per hour ($min = 271$ and $max = 987$). Exploited consumer IoT devices on the other hand, scanned a smaller range of ports per hour (average of 246 ports), except at interval 119, where a sudden increase in the number of scanned ports is clearly observed (Figure 3.11(b)). Investigating the data at interval 119 revealed 734 IoT devices that were generating TCP scanning packets. Among those devices, a single IP camera hosted in the Dominican Republic was responsible for scanning 10,249 ports on 55 destination addresses.

Scanned Protocols/Services

We summarize the top 14 protocols/services that received the most scanning activities from the exploited IoT devices in Table 3.5. Telnet received the highest portion of all TCP scanning packets



(a) CPS



(b) Consumer

Figure 3.11: Overall TCP scanning packets generated towards destination IP addresses and ports by exploited (a) CPS and (b) consumer IoT devices.

(about 50%), followed by HTTP and SSH, which received significantly less number of packets, with about 9.4% and 7.7% of all TCP scanning packets respectively.

Overall, we observed that HTTP, Telnet, Kerberos, and iRDMI, were scanned by a noticeably larger number of compromised IoT devices as compared to other protocols (Table 3.5). In addition, a significantly larger number of compromised consumer IoT devices were scanning HTTP, Kerberos, and iRDMI protocols, contributing towards the majority of generated TCP scanning packets at these ports (Table3.5). On the other hand, while only one exploited IoT device in a CPS realm was actively scanning port 3387 (BackroomNet), almost all scanning packets generated towards port 21677 were

Table 3.5: Top 14 protocols/ports with the most TCP scanning packets generated by exploited IoT devices (CP=93.3%).

Protocol/Port	Packets		Consumer		CPS	
	(M)	(%)	(%)	IP	(%)	IP
Telnet /23/2323/23231	50.08	50.2	63.4	643	36.6	553
HTTP /80/8080/81	9.41	9.4	94.5	1418	5.5	345
SSH /22	7.68	7.7	33.7	64	66.3	80
BackroomNet /3387	6.2	6.2	–	–	100	1
CWMP /7547	4.49	4.5	44.8	169	55.2	244
WSDAPI-S /5358	4.05	4.1	59	94	41	48
MSSQLServer /1433	3.33	3.3	36.2	8	63.8	13
Kerberos /88	2.67	2.7	99	1061	1	23
MS DS /445	2.49	2.5	45.3	43	54.7	330
EtherneIP IO /2222	0.68	0.7	41.6	50	58.4	65
iRDMI /8000	0.67	0.7	98.5	1055	1.5	18
Unassigned /21677	0.57	0.6	0	1	100	87
RDP /3389	0.51	0.5	46.8	42	53.2	61
FTP /21	0.29	0.3	46	20	54	33

also found to be generated by compromised CPS IoT devices (negligible TCP traffic was generated by a single compromised consumer IoT device).

The distribution of the TCP scanning packets targeting the top 5 protocols/services is illustrated in Figure 3.12. It is important to note that most of these protocols were also associated with the recent IoT-initiated cyber attacks (e.g., the `Mirai` botnet and its variations) [3]. In fact, our analysis revealed a number of compromised IoT devices that were actively involved in scanning these protocols. Moreover, these compromised devices were corroborated to perform malicious scanning by comparing them against a publicly available threat repository (`Cymon` [77]), as elaborated in Section 3.5. In what follows, we present further analysis with regards to the top scanned protocol/services.

Telnet. It is clearly observed that Telnet received the highest amount of TCP scanning packets from 1,196 exploited IoT devices. In addition, slightly more compromised consumer IoT devices (54%) were scanning Telnet as compared to those deployed in CPS, generating about 63% of all

TCP scans. Moreover, a total of 7 compromised IoT devices contributed towards 55% of all TCP packets targeting Telnet. These exploited IoT devices, which were hosted in different countries, represent three IP cameras, one router, DVR, and printer, and two devices in CPS associated to power utilities and utility networks. Interestingly, these compromised IoT devices were also associated with malicious scanning as indexed by `Cymon`.

SSH. We noticed sudden increases in the overall scanning activities towards SSH at intervals 32 and 69 (Figure 3.12), with about 242,000 and 253,000 TCP packets generated by compromised IoT devices respectively. Surprisingly, only a hand full of compromised IoT devices, mainly those in CPS, were generating the majority of the TCP scans at these intervals. In particular, two exploited routers hosted in Russia and Australia, and three compromised IoT in CPS (two hosted in China and one in Brazil), generated about 93% of the scans at interval 32. Interestingly, the three exploited IoT in CPS, which generated about 80% of the scanning packets at interval 32, were also found to generate the majority of all scanning traffic at interval 69 (about 90%). In fact, all of the five aforementioned compromised IoT devices were also associated with malicious scanning and/or SSH brute force attacks by `Cymon`.

BackroomNet. We noticed that BackroomNet was scanned by a single compromised IoT in CPS located in Canada, which operated *BACnet/IP* (used in building automation). As shown in Figure 3.12, the intensive scanning activity started at interval 113 (April 16), generating over 6.2 million packets during the next 30 hours (average of approximately 200,000 TCP scanning packets per hour). We also compared this suspicious activity against `Cymon`, and confirmed that it was being involved in malicious scanning activities.

HTTP. A total of 1,763 compromised IoT devices scanned HTTP ports, among which about 80% are consumer IoT devices. Moreover, these compromised consumer devices contributed towards the the majority (94.5%) of the scanning packets targeting HTTP ports, with an hourly average of about 62 thousand generated scanning packets from 415 exploited devices. It is interesting to see that despite the gradual increase in the number of generated scanning packets towards HTTP ports after interval 92 (Figure 3.12), the overall distribution of the scanning packets illustrates a more organized and uniform scanning behavior that does not involve noticeable behavioral changes from the compromised IoT devices. This behavior however, might be resulting from orchestrated

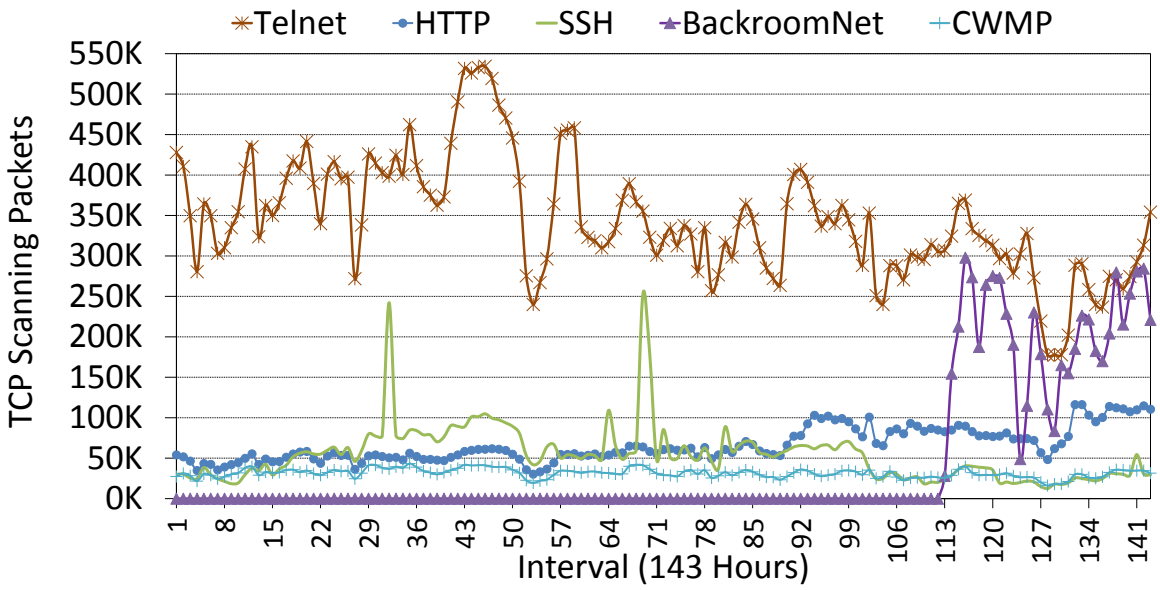


Figure 3.12: The distribution of TCP scanning packets generated by exploited IoT devices towards the top 5 targeted protocols/services.

stealthy scans generated by compromised IoT devices towards the Internet. Proving this would require further investigations and will be considered for future work.

CWMP. The CPE WAN Management Protocol (CWMP) is a web-based protocol that enables remote configuration and management of routers, gateways, and other IoT devices [78]. More importantly, CWMP was utilized by some variants of the `Mirai` botnet to exploit routers [3]. A total of 413 compromised IoT devices, among which 59% were CPS-related, generated more than 4M TCP scanning packets towards CWMP (Table 3.5). On average, CWMP was scanned by 36 compromised IoT devices, generating more than 31,000 scanning packets per hour. As illustrated in Figure 3.12, these scans had the least variations in terms of magnitude of the generated packets during the analysis interval. Despite that, we noticed an exploited router, which was located in Australia, to generate relatively more scanning packets (10.6%), as compared to other compromised IoT devices. Moreover, a total of 5 exploited CPS-related IoT devices were also found to generate relatively more packets than those others deployed in other CPS, representing a total of about 25% of all scanning traffic on CWMP. Three of these devices, which supported *Ethernet/IP* (used in manufacturing automation), were hosted in Korea. The remaining two devices, which were used in control systems (*SNC GENe*) and oil and gas transportation pipelines and distribution networks

(*Telvent OASyS DNA*), were located in China and South Africa, respectively. Finally, all but two of the aforementioned compromised IoT devices were confirmed to be performing malicious scanning of the Internet using `Cymon`.

3.5 Analyzing the Maliciousness of Unsolicited Internet-Scale IoT Devices

In this work, we identified a large number of compromised/exploited IoT devices while characterizing their unsolicited traffic, which pinpointed to some malicious scanning activities. Motivated by these findings, and the plethora of IoT-centric malware that are currently "in the wild" (e.g., `Mirai` and `Hajime`), in this section, we aim at exploring the maliciousness of the inferred IoT devices by investigating: (1) whether such IoT devices are involved in other illicit activities, and (2) whether there exists other malware families and variants that could possibly be exploiting such IoT devices.

3.5.1 IoT Illicit Activities

To investigate the involvement of such IoT devices in malicious activities, we relied on a publicly available cyber-threat intelligence service provided by `Cymon` [77]. The latter renders a service to track and aggregate Internet-scale events related to IP addresses and domains, which are involved in malware, phishing, botnets, spamming, DNS blacklisting, scanning, and web attacks. We investigated the malicious activities associated with 8,839 exploited IoT devices, which represent all devices that generated backscatter traffic (839 DoS victims), and the top 4,000 compromised IoT devices with the most generated scanning and UDP packets from each IoT device category (consumer and CPS). As presented in Figure 3.13, about 10% of the explored IoT devices sent 50 or less packets to the darknet, while only 15% of them sent 10,000 packets or more during the analysis interval. In fact, while less than 2% generated 100,000 packets or more, only 15 devices sent more than 1M packets ($max = 6.25M$ packets). By correlating the explored IoT devices against those IP addresses indexed by `Cymon`, we uncovered 816 IoT devices (9.2%) that were linked to one or more malicious activities.

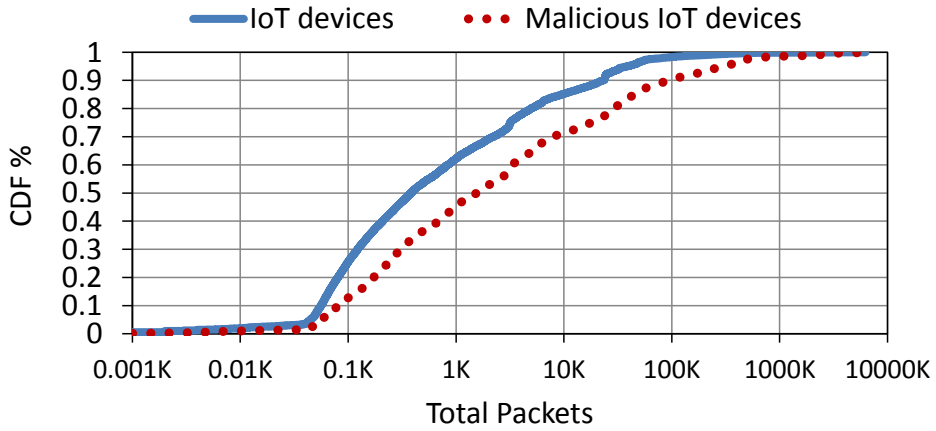


Figure 3.13: Distribution of received packets from the top 8,839 IoT devices and the malicious devices flagged by `Cymon` ($N = 816$).

We amalgamated the identified activities into 6 illicit categories, as summarized in Table 3.6. It is worthy to note that the threat categories are not mutually exclusive, since different sources of cyber threat intelligence (within `Cymon`) might flag a given host/IP with multiple malicious activities. The majority of the identified malicious IoT devices were associated with illicit scanning (96.3%). Furthermore, about 70% of the IoT devices were flagged as miscellaneous (e.g., Web attacks), while about 31% and 28% were associated with SSH brute force attacks and spamming respectively. Interestingly, a total of 117 IoT devices were linked to malware-related activities (14.3%), while only 5 devices were associated with phishing activities. Specifically, our findings identified a significant 91 IoT devices operating in various CPS realms that were indeed associated with malware, with the majority (85 devices) to be involved in TCP scanning activities. Furthermore, 26 consumer IoT devices were linked to malware, with 23 devices performing scanning activities. It is interesting to observe that a total of 9 devices (generating the DoS peaks of Figure 3.8), were found to be related to malware as well.

3.5.2 IoT-Centric Malware Families

Given the fact that we have identified 117 IoT devices that were related to malware, we attempted herein to further explore this matter. In this context, we relied on an in-house built database of malware information. The database is an artifact of conducting large-scale dynamic malware

Table 3.6: Identified threats summary. Note that the identified threats are not mutually exclusive.

Threat Category	**IoT Devices**	**%**
Scanning	786	**96.3**
Miscellaneous (Web/FTP attacks, DNSBL, Malicious domains, VoIP)	574	70.3
Brute force (SSH)	252	30.9
Spam (Mail, IMAP)	227	27.8
Malware (Virus, Worm, Bot/Botnet, Trojan)	117	14.3
Phishing	5	0.6

instrumentation. Indeed, we have been receiving a malware feed on a daily basis with an average of 30,000 malware samples from `ThreatTrack Security`.[2] XML reports are produced by analyzing the malware binaries in a controlled environment. It is worthy to mention that these reports contain the executed activities by the malware samples at the network and system levels. On one hand, the network level activities refer to the connections and the exchanged packets, including IP addresses, port numbers, URLs, visited domains and the actual payload data that has been sent. On the other hand, the system level activities constitute the list of Dynamic-link Library (DLL) files that are utilized by the malware, the key registry changes, and the memory usage. The malware database is built by parsing and indexing such XML malware reports. We executed correlations (using IP address information) between all the inferred unsolicited IoT devices from Section 3.3 (i.e., 26,881 devices) and the malware database. The outcome is intended to demonstrate if any malware variant has communicated with (possibly exploited) IoT devices.

Our findings revealed 33 domain names and 24 unique malware hashes/variants associated with the identified IoT devices. Given the extracted malware hashes from the malware database, we leveraged `VirusTotal` to unveil 11 malware families that were found to be associated with the IoT devices. The uncovered malware families are summarized in Table 3.7. While some of the families are already quite popular, such as the `Ramnit` as a backdoor, and `Zusy` for generating email spam, our results demonstrate that new variants of such families are already being empowered to target the IoT paradigm. To the best of our knowledge, such results (*i*) render a first attempt ever to shed the light on IoT-centric malware families by correlating passive measurements and malware

[2] www.threattrack.com

Table 3.7: Identified, previously unreported malware families exploiting IoT Devices.

#	Identified Malware Families
1	Ramnit
2	Starman
3	Kryptik
4	Nivdort
5	Razy
6	Zusy
7	Bayrod
8	Artemis
9	MSIL
10	Vupa
11	Allaple

samples facilitated through dynamic analysis, (*ii*) highlight on new, previously unreported families (and variants) that have empirically been demonstrated to target the IoT paradigm, and (*iii*) alarm about the rise of new malware variants, which undoubtedly would facilitate the establishment of ever-evolving, IoT-tailored, malware-orchestrated botnets.

3.6 Summary and Concluding Remarks

The Internet of Things (IoT) is an emerging paradigm of technical, social, and economic significance. Nevertheless, the initial priorities of IoT vendors have been focused on providing novel functionality, getting products to market sooner, and making IoT devices more accessible and easier to use. Unfortunately, security concerns have not received as much attention. To this end, this work presented a first empirical look at the magnitude of compromised IoT devices that have been deployed in both consumer and CPS realms. Initially, large-scale correlations between passive measurements and IoT-relevant information is conducted to shed the light on Internet-wide unsolicited IoT devices. Subsequently, empirical measurements, characterization, and analysis is presented to thoroughly investigate IoT-generated unsolicited traffic, including backscattered traffic from IoT devices that have been targeted by DoS attacks, and scanning activities from exploited IoT devices.

Finally, an attempt is made to uncover the maliciousness of such unsolicited IoT devices by utilizing a publicly available threat repository and an in-house built malware database. Some of the outcomes include more than 15,000 compromised consumer IoT devices and more than 11,000 compromised IoT devices operating in critical CPS (including oil and gas, manufacturing plants and power utilities). The results also demonstrate the aggressiveness of more than 5,000 compromised IoT devices in CPS in exploiting other services. The outcome also pinpoints the involvement of a large number of IoT devices in malevolent activities as well as the rise of new malware variants that specifically target the IoT paradigm. Overall, the presented measurements from this work highlight, at large, the insecurity of the IoT paradigm. As for future work, apart from addressing a number of tasks and issues that have been pinpointed throughout this work, we are working on addressing the challenging problem of identifying and clustering IoT botnets and their illicit activities by solely scrutinizing passive measurements.

Chapter 4

Investigating IoT-Generated Scanning Campaigns Targeting A Large Network Telescope

4.1 Overview

Despite the benefits of using IoT devices and their wide spread adoption, the increasing number of IoT-driven cyber-attacks illustrate the rise of IoT-tailored malware, which aim at exploiting vulnerable IoT devices that will be utilized within coordinated botnets to perform further malicious activities [19–21]. In fact, these IoT malware/botnets have gained much popularity among adversaries due to the insecurity of the IoT paradigm and the wide range of existing vulnerabilities. In addition, adversaries have been utilizing compromised IoT devices as effective attack enablers, which can be leveraged to evade detection while performing large-scale malicious activities (e.g., `Mirai` [3]).

In order to mitigate and prevent large-scale IoT-driven cyber attacks, there exists an utmost need

The main work done in this chapter is published at the IEEE Transactions on Dependable and Secure Computing (TDSC) [7]. The work presented in Section 4.6 is published at the IEEE Networking Letters [8].

to detect and characterize emerging IoT malware/botnets, which tend to spread over the Internet by searching for vulnerable IoT devices that could be exploited for future use. This cannot be done without possessing an Internet-scale perspective of IoT devices and their unsolicited activities over a period of time, which is indeed a challenging task as it requires addressing the following problems: (i) the lack of empirical data related to the widespread deployment of IoT devices [37], and (ii) the insufficient knowledge about compromised IoT devices and their underlying malicious activities [79].

To this end, an effective approach to gain Internet-wide cyber threat intelligence is to study passive measurements gathered using designated sensors or traps that collect traffic from the Internet [49, 50]. These sensors collect one-way traffic targeted towards routable, yet unused Internet Protocol (IP) addresses, which are known as darknets or network telescopes [13]. Characteristically, traffic destined to these inactive hosts is likely to represent suspicious and unsolicited activities. Moreover, a large portion of traffic captured at the darknet represents Internet reconnaissance activities [16, 73]. Therefore, motivated by the fact that IoT malware/botnets heavily rely on coordinated scanning activities to propagate through the Internet [3, 5], in this work, we leverage macroscopic, empirical passive network telescope data to execute a multi-level methodology for inferring malware-infected IoT devices and investigating their generated scanning activities. In addition, we leverage data mining methods to unveil common scanning objectives among compromised IoT devices, which reflect the targeted ports/services. More importantly, we demonstrate a meaningful approach for identifying scanning campaigns by clustering correlated IoT devices based on their scanning objectives and similarities in their scanning behaviors over time. Finally, we investigate stochastic modeling of IoT-generated scanning campaigns based on the distribution of the packet Inter-Arrival Time (IAT), which can characterize low-rate scanning campaigns generated by compromised IoT devices.

We leverage over 6 TB of passive darknet data with IoT device information from `Shodan`, and obtain about 172M TCP-SYN scanning packets generated by 8,444 compromised IoT devices over 11 days. Our initial data analysis revealed emerging IoT malware/botnets, illustrated by 18 clusters of correlated compromised IoT devices with similar characteristics of the underlying scanning campaigns. The majority of these IoT botnets (12 out of 18), were found to utilize IoT devices to target

short lists of commonly used ports/services, which are associated with known vulnerabilities (e.g., Telnet/23). Moreover, our results shed light on an emerging IoT malware/botnets, represented by a large scanning campaign towards a distinctive destination port range (e.g., 19328–19622), which to the best of our knowledge, are not associated with any known vulnerabilities. In addition, by analyzing and comparing two instances of IoT scanning traffic that were collected on well-separated time periods (13 months), we highlight the persistence of few well-known IoT malware/botnets, especially those targeting Telnet and HTTP. Additionally, we highlight the evolution of IoT-generated scanning campaigns towards targeting new, or previously uncommon vulnerabilities, which indeed corroborate on the evolutionary nature of IoT malware/botnets. Finally, we propose stochastic models for low-rate, IoT-generated scanning campaigns and evaluate it using simulation and empirical analysis while characterizing campaigns targeting Telnet and HTTP ports.

4.2 Contributions

In this context, we frame the contributions of this work as follows:

- We extend our previous work [2] by introducing a stratified methodology, which utilizes passive darknet data for investigating emerging IoT malware/botnets through inferring compromised IoT devices and characterizing their underlying scanning campaigns.
- We demonstrate a meaningful approach for uncovering IoT-generated scanning campaigns, which is based on frequent pattern analysis to identify common scanning objectives (targeted ports) and unsupervised clustering of correlated IoT devices with similar behavioral characteristics over a period of time.
- We explore the persistence of IoT-generated scanning campaigns by analyzing and comparing two instances of collected data over a course of one year. We also corroborate the evolutionary nature of IoT malware/botnets by highlighting newly targeted destination ports, which tend to include a larger set of possibly vulnerable destination ports/services.
- Employing stochastic processes for modeling scanning packets' Inter-Arrival Time (IAT), while considering network-specific factors such as random packet sampling, path delay, and

jitter. The proposed model is validated and shown effective and accurate in modeling different employed scanning modules for groups of correlated IoT devices.

4.3 Approach

In this work, we aim at answering the following main research question:

How can we leverage passive network measurements to identify exploited IoT devices and infer distinctive characteristics of the underlying scanning campaigns induced by IoT-tailored malware/botnets?

To answer the above question, we follow a multi-stage approach (Figure 4.1), which consists of two main components. First, we correlate IoT device information with darknet traffic to identify exploited devices and their scanning traffic (Section 4.3.3). Second, we identify IoT-generated scanning campaigns and investigate their characteristics by: (i) performing first-level clustering of compromised IoT devices using frequent pattern analysis and association rules mining to group devices that have similar objectives in terms of targeted ports/services (Section 4.4.1), and (2) implementing unsupervised learning techniques to perform second-level clustering of the grouped devices by leveraging a set of aggregated flow features (Section 4.4.2). Finally, while the outcomes represent characteristics of the IoT-generated scanning campaigns, we explore the persistence and evolution of these campaigns over time by analyzing newly collected IoT traffic and comparing results with our initial findings (Section 4.5). Further details on the used methodology is provided in the following sub-sections.

4.3.1 Data Collection

It is worth mentioning that in this work, we utilize the proposed data collection approach presented in Chapter 3 to obtain IoT-generated scanning traffic captured at the darknet. In addition, we leverage our previously collected data sample in terms of the compromised (consumer) IoT devices and their scanning traffic towards the darknet (Section 3.4.3). Nevertheless, to maintain consistency throughout this thesis, we re-iterate our data collection approach in the following sub-section:

IoT Device Information. We leverage a near real-time IoT database provided by `Shodan` [45].

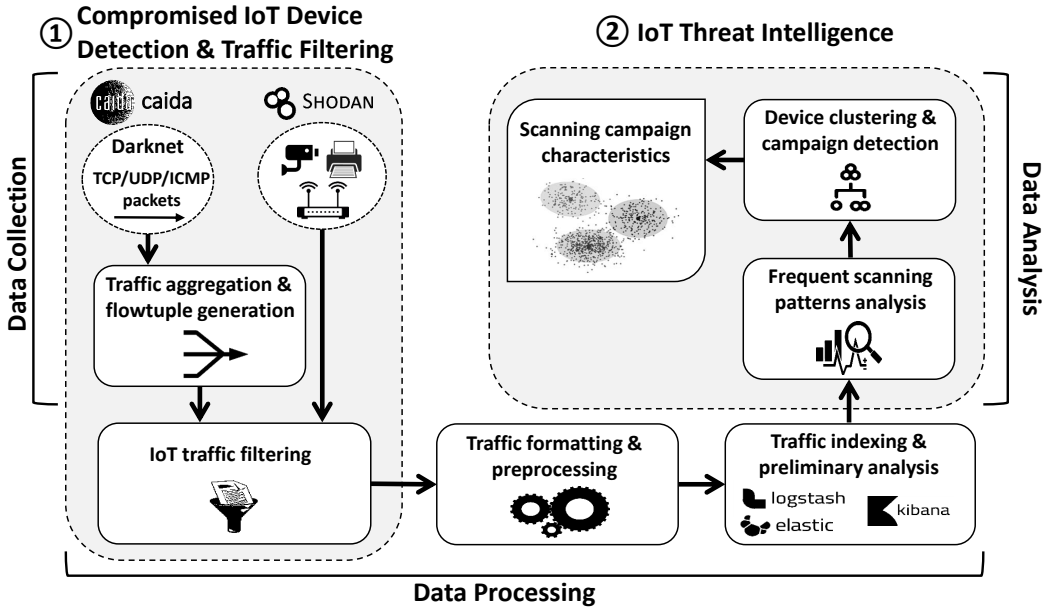


Figure 4.1: The overall approach for detecting and characterizing IoT threats.

This service executes large-scale active measurements to identify and index Internet-facing IoT devices. To this end, we obtained information related to 331,000 IoT devices from `Shodan` (identified in Chapter 3). These IoT devices were deployed in more than 200 countries. In this work, we focus our analysis on a subset of all the IoT devices, namely stand alone devices that are deployed in the consumer realm. We obtained information related to approximately 181,000 IoT devices, including routers (46.9%), printers (29.1%), IP cameras (18.3%), and network storage media (4.6%). The remaining consumer IoT devices equate to only 1.1% of the total devices. These devices were deployed across 202 countries, with the U.S. hosting over 47,000 (24%) IoT devices, representing the country with the largest number of deployed IoT devices in our data. Both the U.K. and Russia followed the U.S. by a significantly less number of hosted devices (about 16,000), representing about 8% of all devices in each country.

Network Telescope Data (Darknet). Darknet data consists of one-way traffic targeted towards routable, allocated yet unused IP addresses (dark IP addresses). Since these IP addresses are not bound to any services, any traffic targeting them is characteristically unsolicited [12, 13]. Typically, darknet data consists of scanning and backscatter activities, in addition to other less common traffic such as misconfiguration and reflection attacks [12–15]. In this work, we initially explored 6 days of

passive darknet traffic between April 12-17, 2017, representing 143 hours of darknet data (over 50 GB of hourly traffic). The darknet traffic is obtained from the UCSD real-time network telescope maintained by the Center for Applied Internet Data Analysis (CAIDA) [67]. It is one of largest available sources of passive darknet traffic with about 16.7 million globally routed destination IPv4 addresses that capture over a billion packets every hour. We processed about 3 TB of darknet data to obtain more than 65M IoT-generated packets that were captured at the darknet during the initial analysis interval (April 2017–As described in Section 4.3.2). In addition, we utilized the darknet to collect new IoT-generated traffic over 5 days in May, 2018 (Section 4.5). Overall, more than 6 TB of darknet data was processed during both analysis intervals, resulting in capturing approximately 172M IoT-generated packets.

4.3.2 Data Processing

The packets captured at the darknet are processed using the `Corsaro` tool, which is a software suite for performing large-scale analysis of trace data [80]. We used `Corsaro` to obtain hourly "flowtuple" files, representing information about incoming flows towards the darknet. Each flow illustrates incoming packets from a source IP to a darknet IP address during one minute time intervals, encompassing the following flow information: source/destination IP addresses and used ports, protocol, Time To Live (TTL), TCP flags, IP length, and total number of packets (per minute). To infer compromised IoT devices, we executed a correlation algorithm that leverages IP header information IoT device information with darknet data to filter out IoT-generated traffic. Finally, the acquired hourly traffic via filtering is prepared in tabular format (flowtuple files) and fed into the search and analysis engine (Figure 4.1), which is implemented using the `ELK Stack` [81]. More specifically, we used `Logstash` for importing data into `Elasticsearch`, which is utilized for flow indexing and analysis. We also used the `Kibana` visualization and navigation tool to run queries and generate corresponding data sets that are used for further analysis throughout the work (Figure 4.1). In what follows, we provide further information on our methodology and obtained results.

4.3.3 Preliminary Analysis (Initial Data Set)

We identified 15,299 unsolicited IoT devices that were correlated with the darknet during the initial analysis period (April 2017). The identified devices generated different types of traffic towards the darknet [2], among which about 80% were TCP-SYN flows. While there are several ways for scanning the Internet, in this work, we focus our analysis on TCP-SYN scans, as they represent the most prominent method of scanning [16, 17]. It is also important to understand that ICMP Echo requests, which are also commonly used for network scans, are excluded from further analysis due to their negligible magnitude in the overall data (0.23%). In addition, the stateless UDP packets (about 8%), which require further investigation of the packet payload to identify their nature (e.g., scanning vs. non-scanning), are also excluded from further analysis throughout the work.

Compromised IoT Devices. The analysis of recent large-scale cyber attacks caused by the `Mirai` botnet and its later variants [3, 20], demonstrated the role of malware-infected IoT devices within coordinated botnets, which are used for scanning the Internet for vulnerable hosts. Given that a benign IoT device has no justifiable reason for scanning the Internet, from here onwards, we label these unsolicited devices as "compromised" or "exploited" IoT devices. Accordingly, we identified 6,802 compromised IoT devices that generated about 54.6M TCP-SYN scanning packets towards the darknet, as illustrated in Figure 4.2. In general, these exploited devices scanned less than 200 unique destination ports per hour, except at interval 119, where we noticed an abrupt increase in the total number of scanned ports (Figure 4.2). Further analysis at interval 119 showed that a single compromised IP camera located in the Dominican Republic was performing a typical vertical scan of over 7,400 ports on 55 destination addresses.

Scanned Ports and Services. The analysis of the number of scanned destination ports indicates that the majority of the compromised IoT devices (90.6%) tend to scan less than 10 unique destination ports. In fact, about half of all devices were found to scan no more than 2 ports, while on the other hand, only about 5% of all IoT devices scanned more than 20 ports. Indeed, this behavior reflects a unique characteristic of the majority of the compromised consumer IoT devices, which were utilized to scan a handful of known ports/services, something that was different in comparison to other IoT devices in the CPS [2]. In addition, the analysis of the top 15 scanned destination ports,

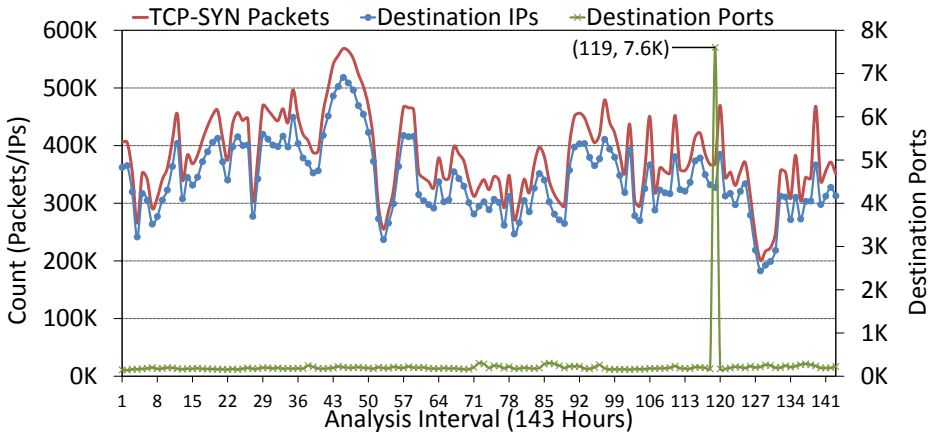


Figure 4.2: The distribution of all TCP-SYN scanning packets generated by compromised IoT devices during the analysis interval (143 hours) [2].

which contribute towards 98.7% of all scanning packets, indicates that Telnet/23 was scanned by the highest number of packets (about 54.7%), as presented in Table 4.1. Despite that, we notice that some ports such as 80, 81, 88, 8000, and 8080, which received smaller number of scanning packets, were in fact targeted by a relatively larger number of compromised devices, as compared to Telnet.

These differences in terms of the number of involved IoT devices in scanning certain ports, along with the scanning rate and total targeted destinations, might indicate distinctive characteristics of the malware-infected IoT devices and their generated scanning activities, which are investigated further throughout the work.

4.3.4 Limitations

The generalizability of our findings might be hampered by the nature of our data (IoT device information and darknet). Considering the limited empirical data on existing IoT devices, we used our resources to sample data from `Shodan` by focusing on consumer IoT devices [45]. Nevertheless, while our sample (about 300,000 IoT devices) is not representative of the overall population of the IoT devices on the Internet, it serves well in supporting our research objectives and generating insights that could be used as a basis for future work towards understanding the activities of malicious/compromised IoT devices. In addition, the darknet represents one-way traffic captured at a slice of the entire IPv4 Internet address space. Despite that, the UCSD network telescope data used

Table 4.1: Top 15 scanned services/ports (CP=98.7%). Source and destination IP counts represent the number of IoT devices and scanned IP addresses.

Service/Port	Packets		IP count	
	(M)	%	Source	Destination (M)
Telnet/23	**29.88**	**54.71**	640	12.75
HTTP/80	5.62	10.29	1,223	4.25
Unassigned/81	2.61	4.83	1,079	2.40
Kerberos/88	2.64	4.78	889	2.40
SSH/22	2.59	4.74	64	2.32
WSDAPI-S/5358	2.39	4.37	89	2.18
CWMP/7547	2.01	3.7	169	1.88
Alt. Telnet/2323	1.84	3.37	199	1.69
MS-SQL-S/1433	1.21	2.21	7	0.71
SMB/445	1.13	2.06	51	0.67
iRDMI/8000	0.66	1.21	875	0.65
HTTP/8080	0.66	1.20	1,053	0.61
EthernetIP/2222	0.28	0.52	53	0.28
RDP/3389	0.24	0.44	39	0.12
FTP/21	0.13	0.24	21	0.06

in this work provides about 16.7 Million destination IP addresses (/8 address space), which is one of the largest available sources of darknet data for research purposes [67].

Another limitation of the work is that the initial data was collected in April 2017, and some of the compromised IoT devices might have been already cleansed. Furthermore, due to DHCP churn [82], the associated IP addresses to those IoT devices might have changed over time. Nevertheless, a comparison of the new list of IoT device information from `Shodan` with the initial IoT device information shows that about 99% of the devices in our initial data were still actively connected to the Internet (on May 2018). Finally, the identification of the exact IoT device type is a challenging task as some of these IoT devices are assigned with dynamic IP addresses. Further, it is common to have IoT devices operating behind a gateway or router (using port forwarding), and therefore, while the associated IP addresses might depict an IoT device, they might be also representing the public IP address of the gateway.

4.4 IoT-Generated Scanning Campaigns

In this work, we propose an approach for detecting malware-infected IoT devices and characterizing the underlying IoT-generated scanning campaigns. The assumption is that compromised IoT devices are likely to perform similar malicious reconnaissance activities within orchestrated scanning campaigns [3, 5, 17, 83]. Given our initial data set (April 2017), we follow a multi-stage clustering/classification approach to identify groups of IoT devices that tend to behave in a similar manner. Our aim is twofold. Firstly, to identify scanning objective(s) by finding unique sets of scanned ports, thus contributing towards campaign intent analysis. Secondly, given groups of compromised IoT devices with common objectives, we perform clustering using a set of raw and aggregate flow features to identify compromised IoT devices with similar objectives and behavioral characteristics.

4.4.1 Scanning Objective(s)

We identify scanning objective(s) by exploring the targeted destination port sets by compromised IoT devices. This is considered as the first step towards inferring scanning campaigns, as discussed in the next sub-section. Furthermore, given that IoT-tailored malware are likely to target a small number of vulnerable ports/services, identifying the scanning objectives is key to attributing the inferred scanning campaigns to known IoT malware, as discussed in Section 4.4.4. Scanning objectives are identified as follows:

Let $D = \{d_1, d_2, \ldots, d_N\}$ be a set of N identified compromised IoT devices that sent TCP-SYN scanning packets to the darknet during the analysis interval E. Let $P = \{p | 0 \leq p \leq 65535\}$ be a set of all TCP ports. For every compromised IoT device $d_i \in D$, we determine scanning objective S_i as a set of all scanned ports $P_{Si} \subseteq P$. Note that these port sets do not account for the order in which the ports were scanned. Let $S = \{S_1, S_2, \ldots, S_N\}$ be a set of N identified scanning objectives for all compromised IoT devices. Given that IoT devices infected by the same malware are likely to produce similar scanning objectives, we define $S_{unique} = \{S_1, S_2, \ldots, S_k\}$ as a set of all distinct scanning objectives ($S_{unique} \subseteq S$).

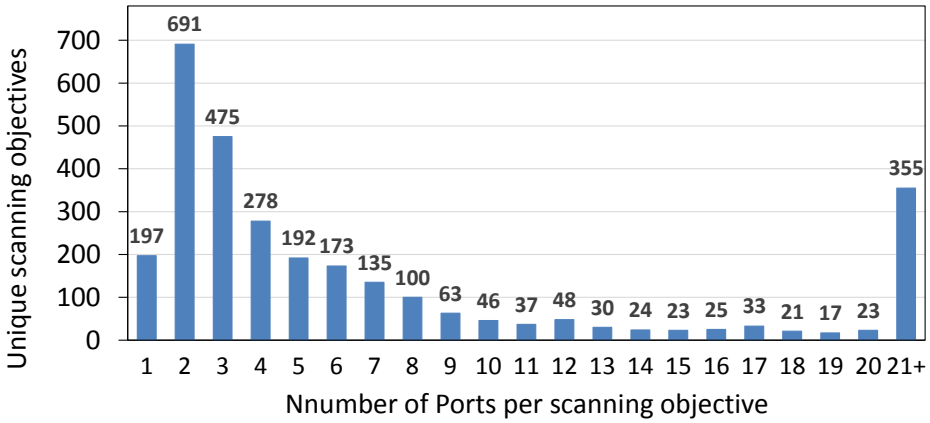


Figure 4.3: The distribution of unique scanning objectives over the number of scanned ports.

We identified $k = 2,986$ combinations of targeted destination ports, representing unique scanning objectives. As shown in Figure 4.3, the distribution of the unique scanning objectives over the number of targeted ports indicates that compromised IoT devices are likely to target a small number of vulnerable ports/services, with about 88% of all scanning objectives containing less than 21 destination ports. This is an interesting characteristic of consumer IoT devices, especially when compared to other IoT devices, such as those deployed in the CPS, which tend to target a larger number of ports/services [2]. It is also important to understand that each scanning objective may correspond to the behavior of one or more compromised IoT devices. In fact, we found that about 60% of all compromised IoT devices produced 227 scanning objectives that were common among two or more devices. On the other hand, while each one of the 2,759 remaining IoT devices was associated to its unique scanning objective, these scanned port sets were very similar, and in many cases they represented subsets/supersets of other scanning objectives. This provides yet another indication that many IoT devices are in fact following similar scanning behaviors in terms of the targeted ports/services throughout the analysis interval.

Scanning Classes

We examined the identified scanning objectives S_i to find similarities in the behavior of compromised IoT devices. The results reflect three classes of mutually exclusive scanning behaviors, as described in the following sub-sections:

Range Scans. This class represents the behaviors of IoT devices that targeted destination ports within distinctive ranges. For instance, the analysis revealed about 4,536 (66.7%) compromised IoT devices that were mainly scanning ports within the following ranges: 19328–19622 and 36224–36582. Given the distinctive behavior in terms of scanning these uncommon port ranges, it is highly likely that the involved compromised IoT devices were in fact driven by similar IoT malware/botnet. To the best of our knowledge, these port ranges are not associated with known IoT malware/botnets, and therefore, the behaviors of the compromised IoT devices might indicate an emerging IoT malware that targets new, or uncommon vulnerabilities.

Moreover, only a handful of known services are registered within the identified port ranges [84]. For instance, TCP ports 19410–19412 are associated with HP services, while the Java Control Panel (JCP) Client is registered on port 19541. Interestingly, ports 19539–19540 are associated with Silex wireless and USB drive adapters [85], which enable wireless connection and network sharing capabilities on many devices such as printers, scanners, and disk drives, to name a few. These adapters use the "SX-Virtual Link" software developed by Silex Technologies to add sharing capabilities on different operating systems (e.g., Windows and Linux), in addition to other embedded devices (e.g., wireless routers). While we do not have conclusive evidence on the actual targets of the scanning campaigns as they targeted different ports within the specified ranges, these findings may shed light on possible intentions of the emerging IoT malware and its targeted devices/vulnerabilities.

Furthermore, about 5% of the IoT devices within this class were also scanning other known ports, with about half of them scanning one or more of the following ports: HTTP/80/8080, Unassigned/81, Kerberos/88, iRDMI/8000, and HTTPS/443. In addition, a small number of devices (16) scanned Telnet/23 along with other known services such as Alternative Telnet/2323, SSH/22, WSDAPI-S/5358, CWMP/7547, and EthernetIP/2222. It is worthy to note that these ports are associated with known IoT malware/botnet (e.g., Mirai [3] and Hajime botnets [4]). Nevertheless, having these ports scanned along with the specified port ranges in this scanning class gives us a clear indication of an evolving IoT malware/botnet, which is targeting new vulnerabilities. However, proving this requires further investigation, which is beyond the scope of this work and will be considered in future work.

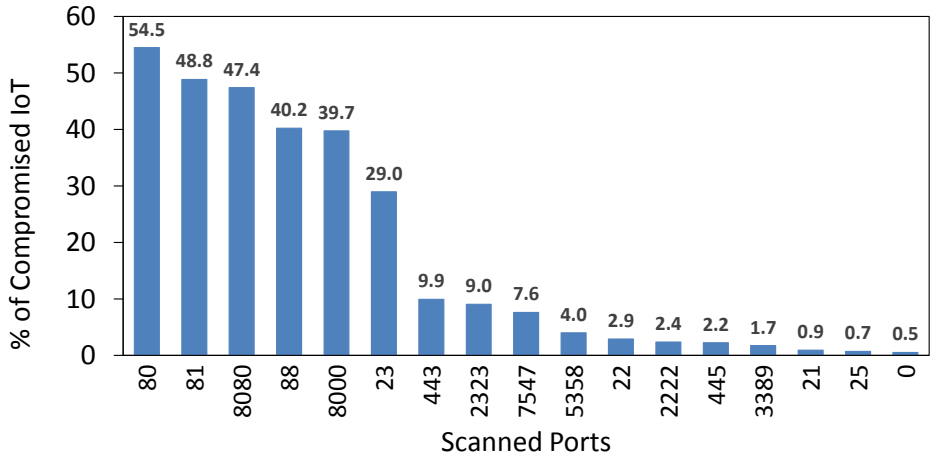


Figure 4.4: Top 17 scanned destination ports by the highest number of IoT devices within strobe scans.

Strobe Scans. The analysis of recently discovered IoT malware/botnets showed that compromised IoT devices were utilized to scan a relatively small number of vulnerable ports/services. We classify these scanning behaviors as strobe scans. In line with that, about 31.5% (2,144) of the compromised devices within our initial data were performing strobe scans, targeting less than 7 ports. In general, these devices targeted 40 different ports/services, among which, HTTP/80 was scanned by the largest number of exploited devices (54.5%). In addition, as illustrated in Figure 4.4, almost all of the top scanned ports are associated with known services that run on IoT devices to enable common operations such as information sharing (e.g., HTTP/80/8080), remote login (e.g., Telnet/23/2323), and communication (e.g., SSH/22), to name a few. It is also clearly observed that a significantly larger number of compromised devices were scanning the first six destinations ports (80–23), as compared to the remaining destination ports.

In addition, the analysis highlights 89 unique scanning objectives within the identified strobe scans (S_{strobe}). A summary of the most frequent scanning objectives within S_{strobe} is presented in Table 4.2 . It is worth noting that the first scanning objective was common among a large number of IoT devices (38.3%), followed by a relatively less number of IoT devices that targeted the remaining port sets. At this stage, the analysis of the frequent scanning objectives highlights specific intentions of the compromised IoT devices and their targeted ports/services, which represent unique characteristics of the underlying IoT malware/botnets. Furthermore, it is clearly observed that some of the

scanned ports were likely to be scanned together as they appeared in several scanning objectives (e.g., HTTP ports 80 and 8080). To explore the correlations between the identified scanned ports (S_{strobe}), we perform association rules mining [86].

Let X and Y be two scanned port sets and T a set of transactions that represent port sets $S_i \in S_{strobe}$. An association rule X→Y describes the probability of port set Y being scanned given that port set X was probed. The *Support* of a rule X is the count of the patterns in T that contain X (Equation 4.1). The *Confidence* of a rule is the support of the rule divided by the number of patterns that contain only X (Equation 4.2).

$$Supp(X) = \frac{|\{t \in T; X \subseteq t\}|}{|T|} \tag{4.1}$$

$$Conf(X \rightarrow Y) = \frac{Supp(X \bigcup Y)}{Supp(X)} \tag{4.2}$$

As shown in Table 4.3, we provide a sample of the association rules related to the frequent scanning objectives (S_{strobe}) identified in Table 4.2. As described through rules 1 to 5 in Table 4.3, there is a high correlation ($Conf. > 99\%$) between scanned ports within S_1 (ports 80 81 88 8000 8080). Furthermore, association rules 6 and 7 show a strong correlations between both ports 7547 and 2323, and port 23 (Table 4.3). This means that if either ports 7547 or 2323 is probed, there is a high chance that port 23 is also going to be probed. Nevertheless, the opposite rules (e.g., 23→7547) were not significant ($Conf. < 85\%$), which means that having port 23 scanned does not necessarily require scanning ports 7547 and/or 2323. The remaining association rules presented in Table 4.3 corroborate the high correlation among scanned ports within the scanning objectives presented in Table 4.2. Therefore, we may conclude that these frequent scanning objectives S_{strobe}, which represent the targeted destination ports/services by compromised IoT devices over a period of time, could reflect unique characteristics of the underlying scanning campaigns. We will elaborate on this in Section 4.4.2.

Wide Scans. In contrary to the range and strobe scanning classes, the remaining identified scanning activities were targeting a variable number of destination ports and IP addresses. We classify these scans as wide scans, as they tend to target a large number of randomly scanned destination

Table 4.2: Top 20 frequent scanning objectives within S_{strobe} generated by about 94% of all devices in strobe scans class.

S_i	**No. of Devices**	**%**	**Scanned Ports**
1	**821**	**38.3**	80 81 88 8000 8080
2	187	8.7	23
3	160	7.5	81
4	152	7.1	23 7547
5	139	6.5	23 2323
6	110	5.1	80 443 8080
7	82	3.8	80 443
8	80	3.7	23 5358
9	74	3.5	80
10	43	2.0	445
11	40	1.9	22 23 2222 2323
12	32	1.5	3389
13	23	1.1	80 8080
14	21	1.0	80 81 8080
15	14	0.7	21
16	13	0.6	25
17	12	0.6	443
18	8	0.4	0
19	7	0.3	81 88 8000 8080
20	7	0.3	8080

ports over the analysis interval. Furthermore, the scanned ports span over all existing ports, including reserved well-known ports that are assigned to widely used services (0–1023), other less commonly used registered ports (1024–49151), and dynamic ports (49152–65535).

We identified 117 IoT devices that implemented different strategies to perform wide scans. It is clearly observed that utilizing exploited IoT devices to perform wide scans is not very likely, as illustrated by the significant difference in the number of involved IoT devices when comparing wide scans with other scanning classes. Despite that, we detected an IP camera from the Dominican Republic that scanned more than 7,000 ports during a short period of time (interval 119-Figure 4.2). These typical port scanning behaviors (e.g., vanilla or sweep scans) might be easily detected by

Table 4.3: Association rules related to the scanned ports identified in Table 4.2.

ID	Association Rule	Support	Confidence (%)
1	81 88 8000 8080 → 80	829	99.2
2	80 88 8000 8080 → 81	829	99.9
3	80 81 8000 8080 → 88	829	99.5
4	80 81 88 8080 → 8000	829	99.5
5	80 81 88 8000 → 8080	829	99.3
6	7547 → 23	161	98.8
7	2323 → 23	193	99.5
8	443 8080 → 80	113	98.3
9	443 → 80	198	93.0
10	5358 → 23	83	96.5
11	23 2222 2323 → 22	44	100.0
12	22 2222 2323 → 23	44	97.8
13	22 23 2323 → 2222	44	100.0
14	22 23 2222 → 2323	44	97.8
15	8080 → 80	997	98.1
16	80 → 8080	997	85.4
17	81 8080 → 80	859	99.0
18	80 8080 → 81	859	86.2
19	80 81 → 8080	859	98.7
20	88 8000 8080 → 81	836	99.9
21	81 8000 8080 → 88	836	99.5
22	81 88 8080 → 8000	836	99.4
23	81 88 8000 → 8080	836	98.8

existing defensive measures as they tend to target a large set of ports and IP addresses. On the other hand, adversaries try to evade detection by implementing a combination of scanning techniques in a randomized and stealthy manner. For instance, a printer located in Taiwan scanned 1,122 ports on 1,132 destination IP addresses throughout the analysis intervals. In fact, almost all exploited devices within this class (except the IP camera from the Dominican Republic) were performing scans with a relatively small average scanning rate (about 88 packets per hour). This however, might reflect the behaviors of the majority of compromised IoT devices that were performing wide scans, as they were active (undetected) for a relatively long period of time.

Involved IoT Devices

The analysis of the involved IoT devices per scanning class illustrates that range and strobe scans contribute to the largest number of compromised IoT devices, with about 66.7% and 31.6% of all devices, respectively. More importantly, the distribution of the IoT device types per scanning classes highlights a noticeable difference between range and strobe scans, with range scans to contain a significantly larger number of routers and printers, while strobe scans containing a relatively larger number of IP cameras, as illustrated in Figure 4.5.

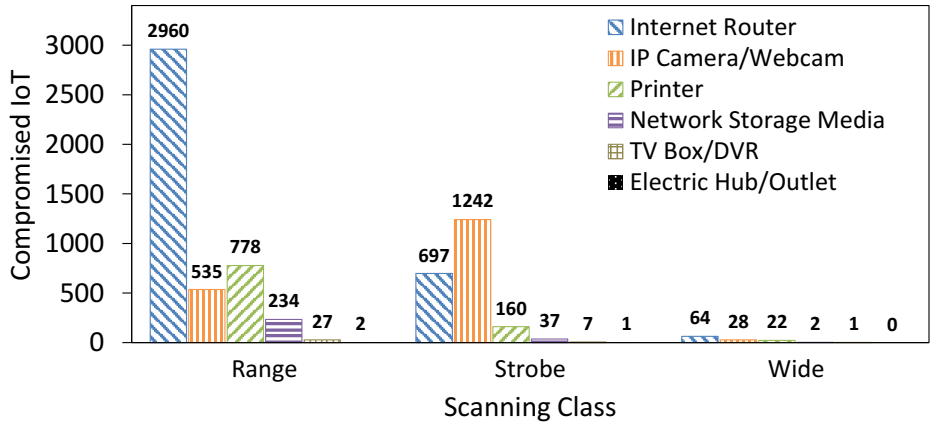


Figure 4.5: Distribution of compromised IoT device types per scanning class.

From a different perspective, while Russia hosted the largest number of overall compromised IoT devices (2,169 devices), it also contributed to the largest number of devices that performed range (46%) and wide scans (29%), respectively. As illustrated in Figure 4.6, it is also clearly observed that the majority (96.4%) of the devices hosted in Russia belong to range scanning class. Similarly, the majority of devices located in China (93.3%), S. Korea (85.4%), and the Philippines (84.4%), were performing range scans, while the behaviors of most of the devices hosted in Thailand (80.3%) and Singapore (75.7%) were classified as strobe scans. Indeed, the distribution of IoT devices per scanning classes, device types, and hosting countries (Figures 4.5–4.6), reveals differentiating characteristics of the underlying scanning activities generated by compromised IoT devices. Nevertheless, while it is difficult to find the exact reason for such dominant scanning behaviors in different contexts, the analysis shed light on important characteristics of the underlying IoT malware in terms of the targeted vulnerable device types (or services), and the countries in which these

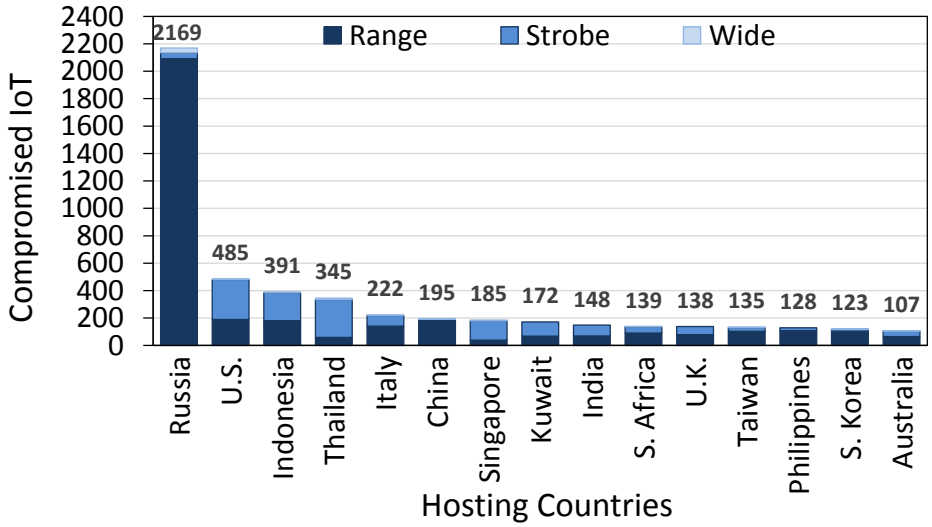


Figure 4.6: Countries with the largest number of exploited IoT devices from each class (initial data–April 2018).

devices are deployed the most. Confirming this assumption requires further investigations, which is considered for future work.

4.4.2 Campaign Detection

To detect scanning campaigns, which represent the behaviors of well-coordinated botnets operating "in the wild," we leverage our knowledge on compromised IoT devices with similar scanning objectives and classes, and utilize their behavioral characteristics to perform clustering following an unsupervised learning approach. We leverage the Density Based Spatial Clustering of Application with Noise (DBSCAN) [87]. This algorithm is widely adopted as it does not require a priori knowledge about the number of clusters, while it can detect arbitrary shaped clusters and outliers [88]. Given a set of points in a specified space, DBSCAN groups neighboring points if they form a cluster with a minimum number of points *MinPts* that are reachable within a predefined radius ε. The overall steps of DBSCAN is presented in Algorithms 1.

DBSCAN can be used with any distance function, however, in this work, we adopt the *Euclidean distance* for further analysis using `R` statistical analysis tools. It is worth noting that using DBSCAN requires adjusting the initial values of ε and *MinPts*, which is not a straight forward task as it requires extra measures to select the appropriate values in different settings. In what follows, we

Algorithm 1: The DBSCAN algorithm.

Input: A set of points D, distance threshold ε, and the minimum number of points in a cluster $MinPts$.

Output: A set of clusters C

```
1  procedure DBSCAN(D, ε, MinPts)
2      for each unvisited point d ∈ D do
3          mark d as visited
4          NeighborPts ← FindNeighbors(d, ε)
5          if |NeighborPts| < MinPts then
6              mark d as noise
7          else
8              C ← newCluster
9              ExpandCluster(d, NeighborPts, C, MinPts, ε)
10         end
11     end
12     return C
13 procedure ExpandCluster(d, NeighborPts, C, MinPts, ε)
14     C← d
15     for each point d′ ∈ NeighborPts do
16         if d′ is not visited then
17             mark d′ as visited
18             NeighborPts′ ← FindNeighbors(d′, ε)
19             if |NeighborPts|′ ≥ MinPts then
20                 NeighborPts ← NeighborPts ∪ NeighborPts′
21             end
22         end
23         if d′ is not in any cluster then
24             C← d′
25         end
26     end
27 procedure FindNeighbors(d, ε)
28     return all points with ε distance from d
```

provide further information on the feature selection process and the results.

Flow Features

Features selection and extraction is a complicated part of unsupervised learning approaches, which has no unique prescribed solution. Let $F_d = \{f_{d1}, f_{d2}, ..., f_{dN}\}$ be a set of aggregate flows corresponding to N compromised IoT devices in the analysis time interval E. Each aggregate flow $f_{di} \in F_d$ is described by a set of β flow attributes or features. It is important to understand that

when using unsupervised classification approaches, we can not apply standard feature extraction methods to validate the optimal number of required features. Therefore, we leveraged the literature to obtain a set of widely used traffic features (e.g., packet rate) [89,90], along with raw and aggregate flow features from our data analysis. Our analysis resulted in selecting $\beta = 10$ features that are summarized in Table 4.4. These features are extracted from the raw flow information and aggregated throughout the analysis period, which represents 143 hourly intervals (6 days). Note that the list of features is not conclusive and we can always add or remove features to improve the clustering results.

Table 4.4: The selected flow features for analysis using DBSCAN ($\beta = 10$).

β	**Selected Features**
1	number of active intervals (hours)
2	per hour packet rate
3	ratio of TCP-SYN packets to non-backscatter packets
4	per destination address packet rate
5	per source port packet rate
6	average number of used source ports per hour
7	average length of the IP packet (from IP header)
8	number of TCP-SYN packets
9	number of scanned destinations
10	number of scanned destination ports

Procedure

We use DBSCAN for inferring scanning campaigns within the identified scanning classes (range, strobe, and wide). To reduce noise and enhance the overall results, we filter out IoT devices that sent less than 10 packets to the darknet during the analysis period. The extracted features are then normalized and prepared to be used in DBSCAN by applying unitization with zero minimum ($x_{norm.} = (x - min)/range$). Moreover, we set $MinPts = 3$ as we assume that a campaign consists of three or more IoT devices that scan the Internet for certain vulnerabilities. To identify the values of ε, we perform the K^{th}-Nearest Neighbor (K-NN) distance analysis with $K = MinPts$. Given a sample of N points, we calculate the distances between every point and its K nearest neighbors.

The resulting NxK calculated distances are then sorted in ascending order to illustrate the K-NN distance plot (Figure 4.7), with the Y-axis to represent the calculated distance values for all NxK data points (X-axis). Note that choosing a very small ε will cause a big portion of the sample to be unreachable via other points, and thus not clustered. On the other hand, choosing a very large ε will result in grouping the majority of the sample into a single cluster. Therefore, to ensure covering the majority of the data points in the clustering analysis, a reasonable value for ε is selected at the point where we observe the beginning of a sharp increase in the values of the calculated K-NN distances, as depicted by the provided example in Figure 4.7.

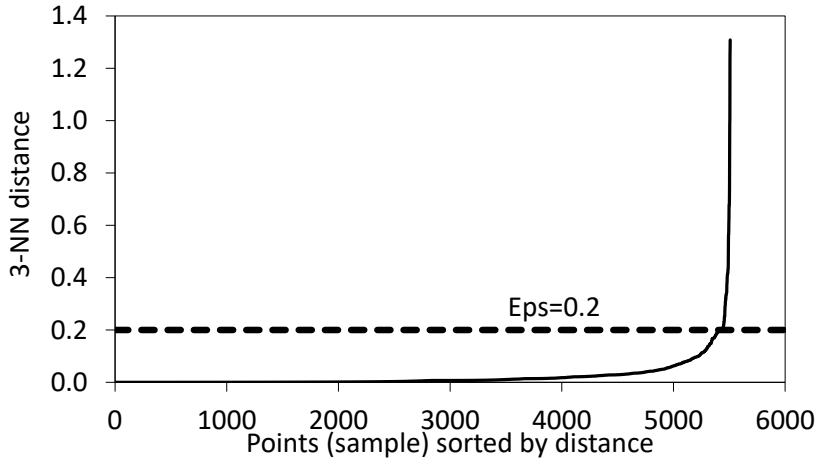


Figure 4.7: An example of K-NN distance graph for IoT devices classified within strobe scans.

We set ε to be 0.15, 0.2, and 0.3 for range, strobe, and wide scans, respectively. Given the selected values of *MinPts* and ε, we perform DBSCAN clustering analysis on devices within each scanning class and report results in the following sub-sections.

Cluster Evaluation

In contrary to supervised learning approaches, cluster evaluation or validation methods are not well developed for unsupervised learning approaches. There are a number of common approaches that are traditionally used to evaluate/validate the clustering results. Nevertheless, clustering evaluation is highly application dependent and thus, subjective. In this work, we use "Internal Measures"

Table 4.5: Summary of the clustering results and evaluation (*MinPts*=3 and $\beta = 10$ features).

Clustering Results						Cluster Evaluation	
Class	**Device**	ε	**Cluster**	**Cluster Size**	**Outliers**	**Within Dist.**	**Between Dist.**
Range	2,688	0.15	3	2604, 3, 18	63	0.379	0.859
Strobe	1,836	0.2	12	3, 822, 58, 4, 4, 4, 865, 11, 11, 4, 3, 17	30	0.457	0.879
Wide	71	0.3	3	13, 46, 6	6	0.363	0.889

such as cluster *cohesion* and *separation* to evaluate our clustering results. Cluster cohesion measures how closely related are objects in a cluster. It is represented by the average within distances among objects of clusters. Cluster separation, on the other hand, measures how distinct or well separated a cluster is from other clusters, and is presented as the average between distance among different clusters.

We analyzed the intrinsic characteristics of the clustering and summarized the evaluation results in Table 4.5 (Cluster Evaluation). Considering that the results are normalized (0.0–1.0), we want the average within distance to be as small as possible, while having a larger average between distance is always preferable. As summarized in Table 4.5, the resulting average within distances for all evaluated scanning classes is reasonable, with values equal to about 0.38, 0.46, and 0.36 for the three classes, respectively. In addition, the average between distances show that the resulting clusters are well distanced from each other in all classes, with an average of about 0.86 (range), 0.88 (strobe), and 0.89 (wide). Overall, while it is difficult to have perfect clustering, the evaluation of the resulting clusters in terms of *cohesion* and *separation* is reasonable. In what follows, we present detailed results in terms of the identified clusters and the underlying IoT-generated scanning campaigns.

Clustering Results

As summarized in Table 4.5, we identified 18 clusters of exploited IoT devices that participated in scanning campaigns. These clusters, which represent groups of correlated IoT devices with similar scanning objectives and behaviors, are illustrated in Figures 4.8–4.12. Note that the clustering is performed based on 10 feature (dimensions), among which features 3 and 4 (Table 4.4) were selected to illustrate the clusters. Therefore, although the clusters are mutually exclusive, they

might look overlapping in the 2-dimensional Figures 4.8–4.12. In addition, outliers, which were not grouped with any of the existing clusters, are represented as isolated black dots in Figures 4.8–4.12. In what follows, we discuss the the characteristics of the identified scanning campaigns with respect to each scanning class.

Range Scans. The majority of the exploited IoT devices (about 96.8%) within the range scanning class were correlated under cluster #1, as depicted by the largest cluster in Figure 4.8. These flow similarities confirm our initial classification according to common scanning objectives (Section 4.4.1), which highlight the correlation among compromised IoT devices that target similar port ranges (range scans). Moreover, considering that these port ranges are not associated with commonly used services or targeted vulnerabilities, they may in fact reflect a unique characteristic of the underlying IoT malware/botnet.

In addition, we noticed differences in the distribution of device types when comparing cluster #1 (Figure 4.9), with clusters #2 (3 IP cameras) and #3 (about 70% IP cameras and 30% routers), respectively. Furthermore, the scanning behaviors were also found to be slightly different when comparing the clusters, with devices within clusters #2 and #3 to be mainly scanning objectives of known destination ports along with the identified port ranges (19328–19622). In fact, 13 out of the 21 devices within clusters #2 and #3 were scanning ports 80, 81, 88, 8000, and 8000, representing the first frequent scanning objective (S_1) from Table 4.2, while the remaining were scanning a combination of Telnet/23 and other ports. This however, gives us yet another clue about the characteristics of the underlying IoT malware/botnet, which behave differently, as reflected by the common scanning objectives within the campaigns. Another interesting characteristics that may differentiate between the identified clusters is the average ratio of TCP-SYN to non-backscatter packets, with a value of about 0.59 for cluster #1, and about 0.96 for clusters #2 and #3. This indicates that on average, devices within cluster #1 were involved in sending a noticeably higher ratio of non-backscatter packets, such as ICMP-REQ and/or UDP packets, as compared to clusters #2 and #3.

Strobe Scans. As summarized in Table 4.5, the analysis resulted in identifying 12 clusters within the strobe scanning class, with clusters #7, #2, and #3 having the largest populations, respectively. The initial analysis of the identified clusters in Figure 4.10, showed that the clustering

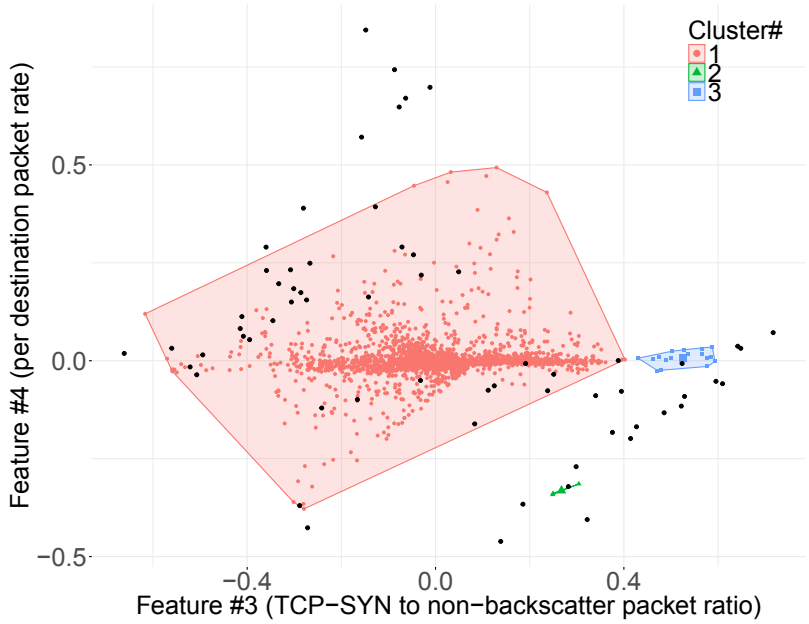


Figure 4.8: Clustering results for Range scans ($MinPts = 3$, $\varepsilon = 0.15$, clusters=3)

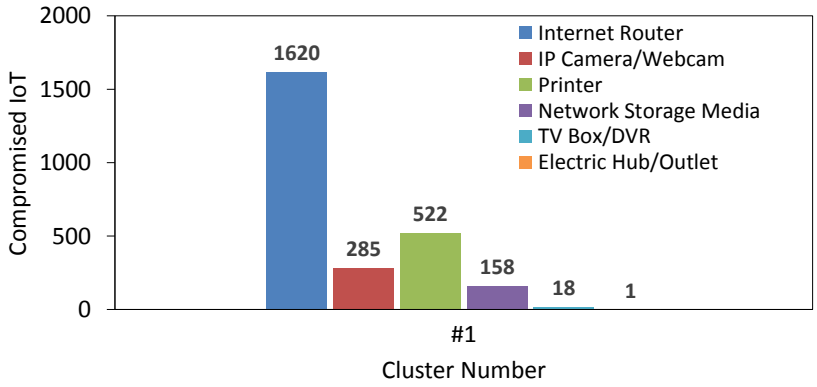


Figure 4.9: The distribution of IoT device type in the largest clusters within Range scans.

results are highly dependent on the number of scanned destination ports within the scanning objectives (feature 10). For instance, cluster #1 consists of IoT devices that scanned 6 destination ports, while devices in cluster #2 scanned 5 ports. Moreover, almost all clusters consist of devices that scanned equal number of ports, except for cluster #7, which contained devices with variable number of scanned ports (1–3 ports). Therefore, although the number of scanned ports specified in the scanning objective might not be a characterizing factor by itself, it can reflect an abstract view of the scanning behavior in terms of the total number of targeted ports/services, which is an important characteristic of the IoT-generated scanning campaigns.

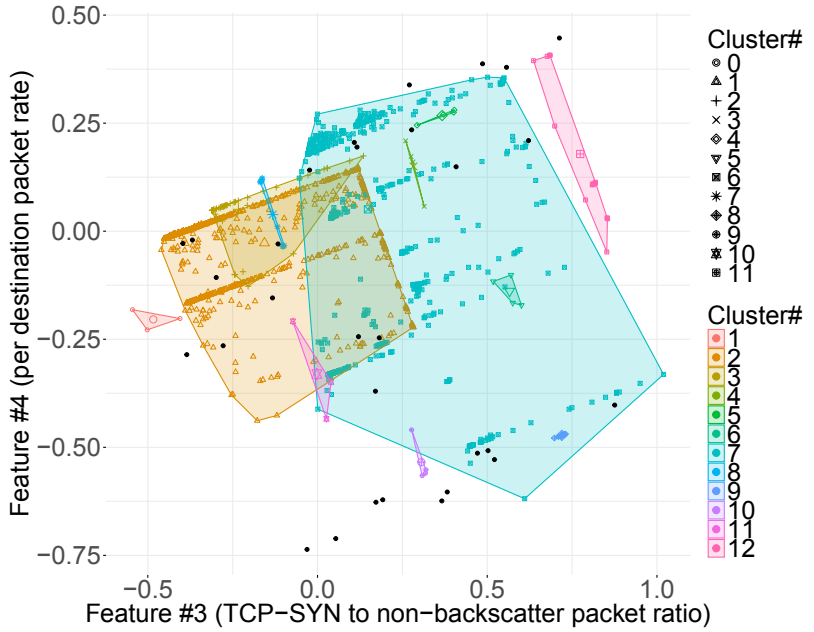


Figure 4.10: Clustering results for Strobe scans ($MinPts = 3$, $\varepsilon = 0.2$, clusters=12)

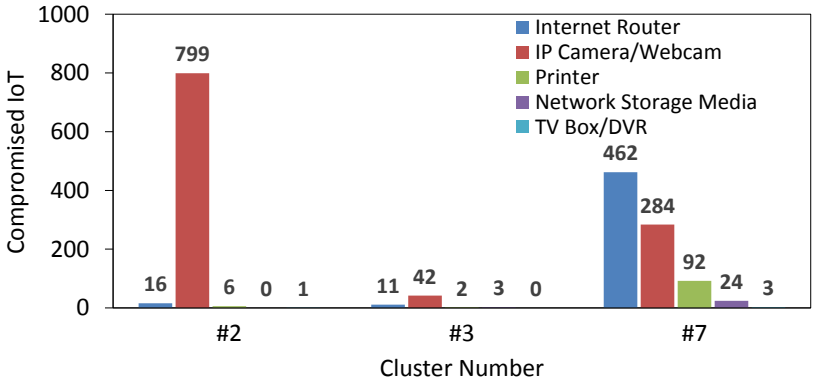


Figure 4.11: The distribution of IoT device type in the largest clusters within Strobe scans.

In addition, the investigation of the targeted ports/services highlighted similar scanning objectives among a considerable number of the exploited devices within most of the identified clusters (Figure 4.11). For instance, the vast majority (99.3%) of devices within cluster #2 were only scanning ports 80, 81, 88, 8000, and 8080, represented by S_1 in Table 4.2. Furthermore, about 66% of IoT devices within cluster #3 scanned S_{11} (ports 22, 23, 2222, and 2323), while about 33% of the devices scanned combinations of ports that are subsets of S_1 (e.g., ports 80, 81, 8000, and 8080). On the other hand, devices within cluster #7, which represents the largest cluster within the strobe

scanning class, generated over 30 different scanning objectives, among which, about 56% were associated with Telnet (e.g., ports 23 and 7547). These results indicate that despite the reasonable grouping of correlated IoT devices based on their aggregate flow features, the clustering algorithm will not be always sufficient to detect distinctive scanning campaigns within strobe scanning class. Therefore, to overcome this limitation and group IoT devices into meaningful scanning campaigns, it is necessary to consider a combination of the clustering and common scanning objectives.

Wide Scans. The analysis of IoT devices within the wide scanning class, which involved a significantly fewer number of compromised devices (71), resulted in three correlated clusters (Figure 4.12). These clusters of IoT devices, which were grouped based on similarities in their aggregate flow features, are illustrated in Figure 4.13.

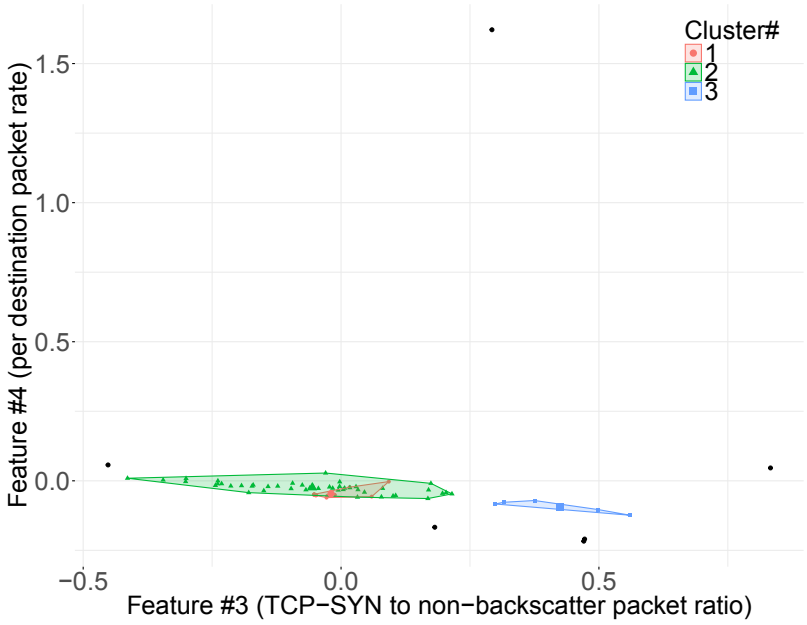


Figure 4.12: Clustering results for Wide scans ($MinPts = 3$, $\varepsilon = 0.3$, clusters=3)

It is worthy to note that the nature of the underlying scanning campaigns in terms of variable length of the scanning objectives, along with the randomness in the targeted destination ports, makes it extremely difficult to associate these IoT devices with unique IoT malware/botnet. Nevertheless, by analyzing the aggregate features with respect to the IoT devices within each cluster, we found a significant difference in the ratio of TCP-SYN packets to non-backscatter packets, with an average value of about 0.98, 0.60, and 0.20, for the three clusters respectively. In addition, while cluster #2,

which represents the largest group of exploited IoT devices within the wide scanning class (about 65%), consist of a relatively larger number of routers and IP cameras, cluster #1 contained slightly more infected printers (50%), as illustrated in Figure 4.13. These results corroborate that exploited devices from the same type are likely to generate similar scanning behaviors and therefore, forming clusters of correlated devices that operate within different scanning campaigns.

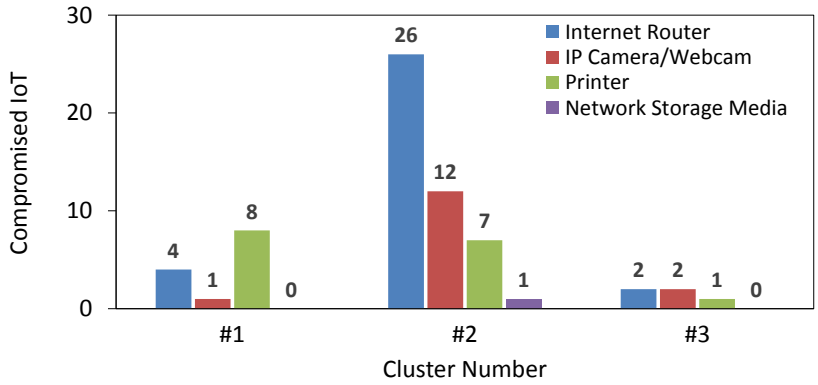


Figure 4.13: The distribution of IoT device type in the largest clusters within Wide scans.

4.4.3 Results Summary

The analysis of the identified scanning campaigns generated by compromised IoT devices provides insights on the behavioral characteristics of the underlying IoT malware. For instance, the analysis revealed common scanning objectives that represent possibly vulnerable destination ports/services. Furthermore, the analysis highlighted behavioral similarities in terms of the aggregated flow features. Together, these similarities were effectively used to uncover groups of IoT devices that were likely to be infected by similar IoT malware, as reflected from their scanning activities on the darknet.

Moreover, the analysis revealed that the ratio of the generated TCP-SYN scanning packets to non-backscatter packets (e.g., ICMP-REQ and UDP) is in fact a differentiating feature when characterizing the behaviors of exploited IoT devices within different scanning classes/campaigns. For instance, a Kruskal-Wallis rank sum test with pair-wise comparison tests (Bonferroni adjustment $p_{bonf.} = 0.0167$), showed statistically significant differences ($p < 0.0001$) of means in the ratio of

TCP-SYN to non-backscatter packets when comparing strobe scanning class with other classes, respectively. Indeed, devices within the strobe scanning class were mainly sending TCP-SYN packets (average ratio of about 0.99), and therefore, highlight a unique characteristic that can distinguish them from other devices.

From a different perspective, the prevalence of certain IoT device types within the identified scanning campaigns determine a feature of the underlying IoT malware/botnet, which is tailored to exploit certain vulnerable devices. For instance, clusters #2 and #3 within the strobe scanning class consist of mainly IP cameras. Nevertheless, it is interesting to see that devices withing these clusters targeted different destination ports, with the majority of devices within cluster #2 and #3 to target S_1 and S_{11} (Table 4.2), respectively. While we do not have concrete information on the actual malware/botnet that is generating these scanning campaigns, these behaviors can in fact illustrate the emergence and evolution of IoT-tailored malware, which tend to target multiple vulnerabilities on the targeted devices.

A main characteristic that differentiates between scanning campaigns is the scanning objective, which reveals the targeted ports that relate to existing vulnerabilities. More importantly, while these targeted ports are usually associated with known malware/botnets, the identification of scanning campaigns that target uncommon ports (e.g., range scanning class), which are not associated with known vulnerabilities can be utilized to predict and mitigate emerging IoT malware.

4.4.4 IoT Malware Attribution

To validate our approach in terms of detecting scanning campaigns based on common scanning objectives, we collected more than 9,000 real IoT malware executables and performed multiple experiments to extract real IoT malware traffic. Our objective herein is to corroborate findings from analyzing the darknet and attribute the identified scanning campaigns to known IoT malware/botnets. In what follows, we elaborate on the data collection methodology, experimental setup, and results.

Data Collection

We leveraged the data collected by an IoT-based honeypot (IoTPOT [6]) to acquire about 8,000 samples of IoT-specific malware. We also extracted about 1,000 samples of IoT-related Mirai and Bashlite malware executables from a generic online malware repository (VirusShare.com). It is important to realize that we performed a number of pre-processing steps to filter out corrupted malware samples from our experiments. Furthermore, due to our sand-box environment limitations, we had to discard malware samples that did not work on the used instruction set architectures (e.g., malware samples for SH4). Finally, given that malware family names might not be conclusive, we leveraged VirusTotal to obtain reliable malware family names/information, while excluding samples with unreliable/insufficient information.

Experimental Setup

Given the collected IoT malware samples, we developed two experimental environments for executing and analyzing the malware binaries, as illustrated in Figure 4.14. First, considering the fact that IoT malware are found to target almost all existing CPU architectures, we setup a multi-architecture environment that emulates the most common CPU architectures using a virtual sand-boxing environment on Qemu systems [91]. Second, we created an experimental testbed to mimic the behaviors of IoT devices connected to a wireless access point using three Raspberry Pi3 (Model B+) boards with Rasbian OS [92]. It is worth noting that the created testbed, which supports the execution of ARM-based malware only, was utilized to validate the actual behaviors of the IoT malware by testing for employed sandbox detection/evasion techniques. In fact, our analysis showed almost identical traffic generated by the tested malware on both environments (virtual and physical), which indicates the absence of employed evasion techniques. Finally, we utilized the created testing environments to execute IoT malware samples for thirty minutes each while capturing the exchanged traffic at the gateway using TShark.

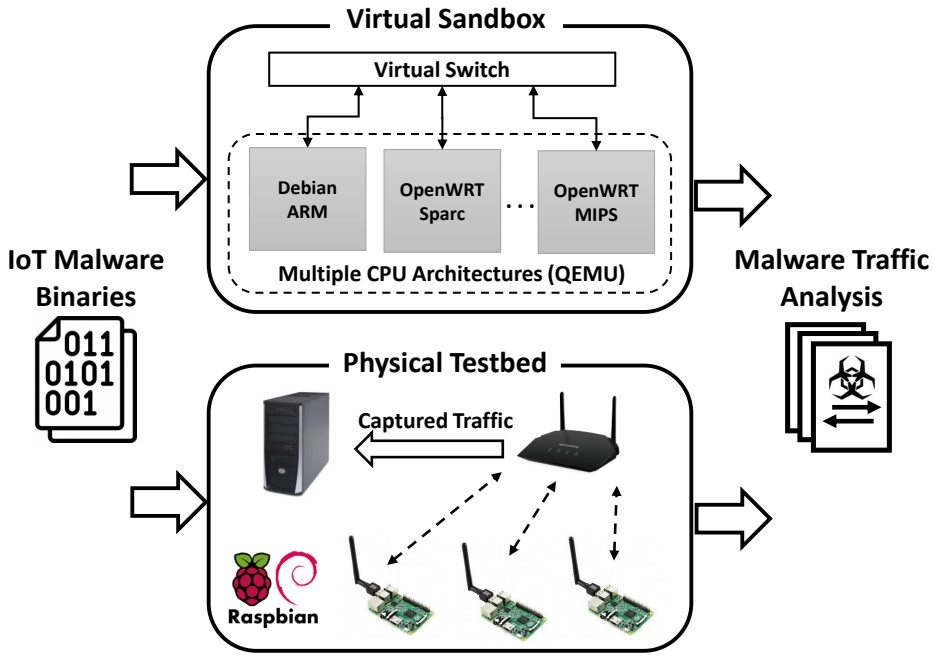


Figure 4.14: The created environment for analyzing IoT malware.

Results

As summarized in Table 4.6, the experimental analysis resulted in identifying a number of IoT malware variants, which generated scanning campaigns towards ports similar to those identified in our initial darknet data (Tables 4.2). For instance, the `Mirai-Satori` was targeting port 81, which matches one of the common scanning objectives in our initial data set (S_3 from Table 4.2). In addition, the `Mirai-A` was found to be targeting ports identified within S_5 (Table 4.2), which corresponds to the behavior of 139 compromised IoT devices in our initial data. Furthermore, while the targeted ports by some of the analyzed IoT malware such as `Lightaidra` and `Mirai-G` were less prevalent among the identified scanning objectives in our initial data set, we identified a relatively larger number of exploited devices that scanned these ports when analyzing a recent sample of darknet data, as described next in Section 4.5. This might be justified by the evolving nature of IoT malware, which are tailored to target new combinations of ports that are associated with emerging vulnerabilities.

It is important to understand that our experimental results are bound to the limited number of analyzed malware samples, which do not represent the activities of all existing IoT malware families.

Table 4.6: Analyzed IoT malware samples and their targeted ports.

IoT Malware MD5	Targeted Ports	Malware Family
807a15c2c87c7bb21d7660251e0db6f8	81	Mirai-Satori
05a8435816bb768761fdc893e79dc988	23 2323	Mirai-A
0540e803f1788f75369f434ace742346	445	Lightaidra
215e366b75e8998e214dcc2094f7c95d	443	Tsunami
67609e719aca8bfce3ac8c2500cfdacf	80 81 8080	Gafgyt-A
62a907378286e3fa431279dc2df948a4	23 80 8080	Mirai-G
d14d3483aac0032f37a9b3c42722e51a	5555	Mirai-B/ADB.Miner
4cf9d9961da97c204b303bbfe874a035	2000	Bashlite

Nevertheless, our results can indeed validate our methodology in terms of identifying malware-infected IoT devices and attributing their generated scanning campaigns to the overall behaviors of known malware families. More importantly, given the fact that IoT malware are rapidly evolving towards targeting new discovered vulnerabilities, our approach can be leveraged to infer the behaviors of emerging IoT malware through the detection of scanning campaigns that target new/uncommon ports. Finally, it is important to realize that despite the identified behavioral similarities among real IoT malware and the exploited devices involved in scanning the darknet, finding the exact malware variant/family that infected these devices requires further in-depth investigation and fingerprinting, which is considered for future work.

4.5 Campaign Persistence and Evolution

In order to investigate the persistence and evolution of IoT-generated scanning campaigns, we compared our findings from analyzing the initial data that was collected during April 2017, with newly collected data from the darknet. We followed the steps described in Section 4.3 to process over 3 TB of newly collected IoT traffic from the darknet between May 21–25, 2018 (108 hours). The new data represents about 107M packets generated by 2,902 IoT devices towards the darknet, among which, about 99% (over 106M packets) were TCP-SYN packets. These TCP-SYN packets were generated by 1,647 compromised IoT devices, with an average of about 390 IoT devices that were generating approximately 988,000 TCP-SYN packets towards the darknet per hour

(Figure 4.15). In what follows, we compare the IoT-generated scanning campaigns from the two collected data sets and investigate scanning activities, campaign persistence, and evolution.

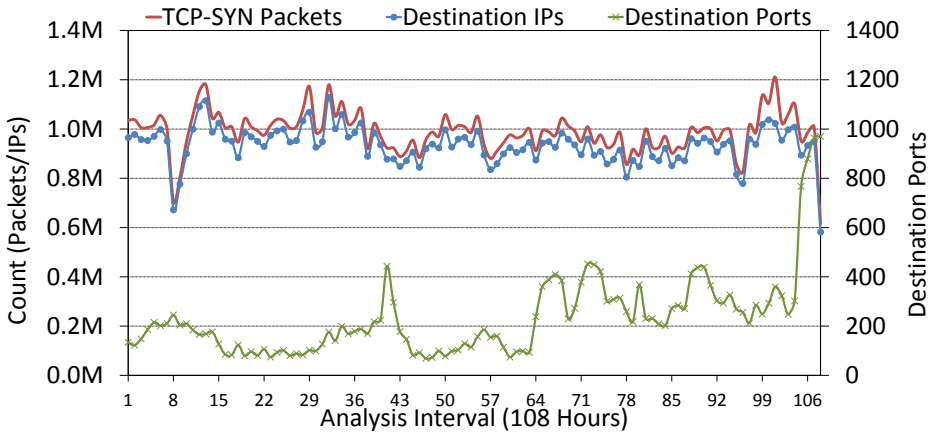


Figure 4.15: The distribution of all TCP-SYN scanning packets generated by compromised IoT devices during the new analysis intervals (108 hours).

4.5.1 Scanning Activities

The analysis of the newly collected data resulted in identifying a significantly less number of compromised IoT devices (1,647), as compared to the 6,797 devices that were discovered in our initial data collection. Nevertheless, these fewer IoT devices were found to be more active in scanning the darknet, sending significantly more TCP-SYN packets (106M packets) towards the darknet over a relatively shorter period of time (Figure 4.15), as compared to the 54.6M TCP-SYN packets that were generated by the IoT devices in our initial data (Figures 4.2). In fact, a Mann-Whitney U Test confirms that the number of generated scanning packets by compromised IoT devices was significantly greater ($p < 0.0001$) for the newly identified devices ($median = 993,931$ packets) than for the devices identified in the initial data ($median = 371,486$ packets).

Moreover, the compromised IoT devices in the new data set scanned an average of 245 unique destination ports per hour, with Telnet/23 to be scanned by the highest number of TCP-SYN packets, followed by HTTP ports 80 and 8080 (Table 4.7). It is important to note that these ports have been continuously targeted by different variants of IoT malware/botnets (e.g., `Mirai`). Moreover, while these ports were scanned by less than 33% of all IoT devices, port 445, which is associated

to the Server Message Block (SMB) protocol, was scanned by a relatively larger number of IoT device (44.5%), among which the majority (705 out of the 773) did not scan any other ports. Further investigation shows that the SMB protocol has been vulnerable to the `EternalBlue` exploit, which was leveraged by `WannaCry` ransomware to perform large-scale attacks towards computers running Windows OS in May 2017 (one month after our initial data collection). Interestingly, our findings indicate that compromised IoT devices have been used to perform reconnaissance activities to identify different types of vulnerable hosts, including non-IoT devices. Furthermore, we observe a considerable increase in the number of IoT devices that scanned port 445 in the new data set (Table 4.7), as compared to the initial data (Table 4.1). While the real reason behind the increased scanning activities towards port 445 is not known to us, we believe that our findings may provide an early indication of large-scale malware outbreaks, which target the vulnerable SMB protocol on port 445. Indeed, our findings have been corroborated by other reports, which highlight the growing number of scanning activities and malware-driven attacks towards port 445 in recent years [93].

The comparison of the total number of compromised devices hosted in different countries across the two analyzed data sets indicates a significant drop in the number of exploited devices hosted in Russia, followed by relatively smaller drops in the number of devices hosted in the U.S. and Thailand (Figures 4.6 and 4.16). In addition, while routers contributed to the largest portion of the IoT devices in our initial data (about 65%), IP cameras represented the largest population in the new data (about 50%). These changes can be justified by the significant decrease in the proportion of IoT devices within the range scanning class, which consist of mainly routers that were largely hosted in Russia. However, while the real reason behind the temporal change is unknown to us, we can only assume that these scanning campaigns have faded as a result of remediation and patching processes that took place after detecting the malware-infected devices and their malicious activities.

On the other hand, there is a noticeable increase in the total number of wide scanners hosted in Russia, as compared to other countries. Moreover, the number of devices that performed strobe scans almost doubled in Indonesia to reach slightly over 400 devices in the new data set. These temporal changes may in fact raise attention towards a number of points such as the weak security measures and/or remediation efforts put by consumers in those countries. Also, it may reflect the emergence of specific IoT malware variants, which target/exploit vulnerable devices that are widely

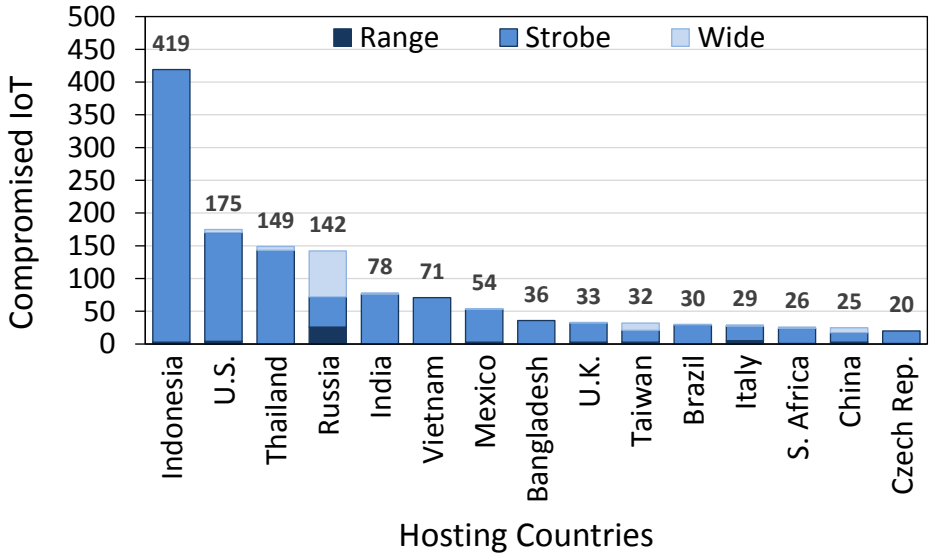


Figure 4.16: Countries with the largest number of exploited IoT devices from each class (new data–May 2018).

deployed in those countries.

4.5.2 Persistence

The analysis of new IoT-generated data revealed scanning classes similar to those identified in Section 4.4.1. For instance, we found 44 IoT devices that were scanning the exact port ranges as specified in the range scans (e.g., 19328–19622), among which 19 devices were also common in both data sets. The persistence of such scanning activities after one year of initial observation might be justified in different ways: first, there are adversaries that are still interested in scanning possibly vulnerable hosts on these port ranges. Second, these compromised IoT devices were able to successfully evade detection and perform unsolicited scanning activities over a long period of time. Third, these exploited IoT devices were not updated/patched to receive the necessary remediation.

Furthermore, the majority of the compromised IoT devices in the new data set (1,421 out of 1,647) were performing strobe scans that are mainly targeting known services such as Telnet/23/2323, HTTP/80/8080, RDP/3389, SMB/445, and HTTPS/443. The scanned ports also include other less common/known services (i.e., ports 81–85, 8001, 8081, 8088, and 8888), which are thought to be used as alternative ports for HTTP by a number of online applications. More

Table 4.7: Top 18 scanned services/ports (CP=99%).

#	Service/Port	Packets		IoT Devices	
		(M)	%	Source IP	%
1	*Telnet*/23	**14.32**	**13.42**	465	28.2
2	*HTTP*/80	8.14	7.64	556	33.8
3	*HTTP*/8080	8.06	7.56	508	30.8
4	*Unassigned*/81	6.14	5.76	126	7.7
5	*Kerberos*/88	5.96	5.59	119	7.2
6	*iRDMI*/8000	5.91	5.55	120	7.3
7	*Alt. Telnet*/2323	5.90	5.54	118	7.2
8	*XFER*/82	5.90	5.53	104	6.3
9	*MIT-ML-DEV*/85	5.90	5.53	102	6.2
10	*SUNPROXYADMIN*/8081	5.90	5.53	110	6.7
11	*DDI-TCP-1*/8888	5.90	5.53	104	6.3
12	*MIT-ML-DEV*/83	5.90	5.53	103	6.3
13	*RADAN-HTTP*/8088	5.90	5.53	108	6.6
14	*VCOM-TUNNEL*/8001	5.90	5.53	105	6.4
15	*CTF*/84	5.90	5.53	103	6.3
16	*SMB*/445	2.86	2.68	**773**	**44.5**
17	*RDP*/3389	0.66	0.62	20	1.2
18	*SSH*/22	0.45	0.42	15	0.9

importantly, 8 out of the top 10 identified scanning objectives in the new data set (Table 4.8), which account for about 90% of all devices within strobe scanning class, also appeared among the top scanning objectives identified in the initial data set (Table 4.2). The similarities in terms of the identified scanning objectives and targeted ports demonstrate the persistence of IoT-generated campaigns over time, which tend to target a short list of vulnerable services using strobe scans.

In addition to range and strobe scanning classes, we identified 152 devices that performed wide scans, which consist of mainly routers (63%), followed by printers (19.7%), and IP cameras (15%). Furthermore, Russia hosted the largest number of these devices (about 46%), with significantly fewer number of devices distributed among other countries (Figure 4.16). Despite the fact that wide scans are less prevalent among compromised IoT devices in our data, the slight increase in the number of involved devices in the new data as compared to our initial data indicates the persistence

Table 4.8: Frequent scanning objectives within strobe scanning class (CP=89.4%).

S_i	**Frequency**	**%**	**Scanning objective (ports)**
1	**705**	**49.6**	445
2	228	16.0	23 80 8080
3	96	6.8	23 80 81 82 83 84 85 88 2323 8000 8001 8080 8081 8088 8888
4	79	5.6	80 443 8080
5	46	3.2	23
6	29	2.0	80 443
7	28	2.0	80 8080
8	26	1.8	80
9	19	1.3	3389
10	15	1.1	23 2323

of such campaigns. Confirming this however, requires further investigations that is beyond the scope of this work and might be considered for future work.

4.5.3 Evolution

The analysis of the scanning objectives and classes within the newly analyzed data revealed 30 IoT devices (20 IP cameras and 10 routers) that were targeting a new range of destination ports (2–10000). These devices, which contributed to the high peaks in terms of the number of scanned destination ports throughout the analysis intervals (Figure 4.15), were performing distributed scans by targeting ports within the identified ranges on many destination addresses, resulting in a maximum rate of 5 packets per destination. Given the distinct scanned port ranges, we classify them as yet another variation of range scans, which reflect the behaviors of new or evolving IoT malware/botnets. It is also interesting to see that almost all of the devices scanned Telnet/23 and HTTP/80/8080 ports, which is another sign of underlying correlation among these devices (i.e., scanning campaign).

Moreover, despite the similarities in the majority of the identified scanning objectives within the strobe scans when comparing both data sets (Tables 4.2 and 4.8), we observed the emergence of new scanning objectives that were in fact associated with recently discovered vulnerabilities. For instance, 12 compromised IoT devices were actively scanning port 5555, which is associated

with ADB.Miner [19], the first Android worm to utilize port scanning code borrowed from Mirai. Similarly, we found traces of scans towards port 3333, which is associated with Fbot [23], a Satori variant that exploited various hosts on the Internet through their management port that runs the Claymore Miner software. Moreover, our results indicate possible traces of the Hajime botnet [4], which searches for vulnerable routers by scanning a list of ports including but not limited to 80–82, 8080, and 8081. Interestingly, while these ports appear in one of most frequent scanning objectives (S_3 from Table 4.8), they were also associated with other scanned ports (e.g., 8088 and 8888), which might reflect the behaviors of emerging IoT malware/botnets. In addition, we also noticed scans towards port 81, which is associated with a malware variant that extends Satori to exploit Goahead IP cameras [94]. Other newly scanned ports that were also related to a range of vulnerable services include: ports 83–85, 2000 (Cisco SSCP enabled phones [95] and Bashlite), 3389 (Mirai on RDP [96]), 8600, and 9000. It is important to understand that given the distinctive characteristics of the IoT devices in terms of the scanned ports, it is not anomalous to consider those devices to be correlated. In other words, they might be exploited by similar IoT malware, and therefore, involved in scanning campaigns as a part of a bigger botnet.

4.6 Characterizing IoT-Generated Internet Scanning Activities Using Their Packet Inter-Arrival Times

Motivated by the prevalence of low-rate stealthy scans generated by compromised IoT devices within well-coordinated botnets [3, 60], we draw upon the IoT-generated scanning activities captured at the network telescope to infer and characterize low-rate scanning activities based on the distribution of their packet Inter-Arrival Times (IAT). While packet IAT has been previously used as an effective feature for characterizing network scans [97, 98], we aim at providing a better understanding of the scanning activities through empirical analysis and probabilistic modeling of the perceived IAT, which would pave the way for exploring IoT-centric open research problems and much needed diverse applications, including IoT device fingerprinting, malware attribution and

The work done in this sub-section is published at the IEEE Networking Letters [8].

campaign detection.

To achieve such objectives, we leverage about 3.6 TB of data collected at the network telescope (darknet) to infer scanning traffic generated by compromised devices. We then obtain device information/labels by performing instantaneous scanning and banner analysis of such devices to identify various information such as device type (e.g., IoT/non-IoT) and known malware signatures (e.g., Mirai) [60]. Furthermore, we perform empirical analysis of the scanning activities by measuring the IAT Probability Density Functions (PDF) for all devices through implementing a series of dimension reduction techniques, while clustering correlated devices into meaningful groups. Indeed, the obtained results demonstrate the effectiveness of our approach towards classifying IoT and non-IoT devices based on the distribution of their IAT, while showing that devices infected by the same IoT malware family are likely to be correlated due to their similar scanning behaviors. Finally, while we introduce novel stochastic processes for modeling low-rate scanning activities based on observed packet IAT, we provide empirical evidence to support the accuracy of the theoretical model in estimating the behaviors of different groups of correlated devices that perform low-rate stealthy scanning activities.

4.6.1 Proposed Model

We use stochastic modeling to formulate the probability density function of IAT for randomly sampled packets from a given source towards a vantage point on the Internet. The proposed model is founded on three main hypotheses and assumptions: (1) The scanners/infected IoT devices generate stationary behaviors which allow us to model their longitudinal activities; (2) Scanners send scan packets following a burst-idle model; and (3) We are only able to observe small sample of darknet-received packets which are randomly selected.

In general, we assume scanners send batch of n packets and go dormant/idle for a deterministic or a random period. This dormant period can be due to imposing rate limiting, time required to process response packets, or performing other tasks. We model the scan traffic process as a modulated stochastic point process, where a batch of scanning packets are sent within a fixed inter-arrival time (Proposition 1.1) or following an exponential distribution (Proposition 1.2). We invite interested readers to refer to our work in [8] for further details about the theoretical formulation and validation

of the model.

Proposition 1.1. In case of a precise batch inter-arrival $f(t) \sim \delta(t-T)$:

$$g(t) = (1 - \frac{q}{n\rho})\delta(t) + \frac{q^2}{n\rho}\sum_{i=1}^{\infty} p^{i-1}\delta(t-iT) \tag{4.3}$$

Proposition 1.2. In case of batches of packets, which are sent out with Poisson distribution (inter-arrival time of batches are following exponential distribution $f(t) = \lambda e^{-\lambda t}u(t)$), we observe exponential shape distribution with rate $q\lambda$:

$$g(t) = (1 - \frac{q}{n\rho})\delta(t) + \frac{q}{n\rho}(q\lambda)e^{-q\lambda t}u(t) \tag{4.4}$$

4.6.2 Empirical Analysis of Packet Inter-Arrival Times

Data Collection

To this end, we utilized the algorithms developed in [60] to analyze about 3.6 TB of darknet data (Oct-08-2019), identify scanning traffic generated by compromised hosts (112,851), and infer device labels (IoT/non-IoT). It is worth noting that the network telescope represents a large destination IP address block, where it captures packets from a given source following similar paths with equal time delays. With that in mind, we employed the Dvoretzky-Kiefer-Wolfowitz (DKW) inequality [99] to estimate the minimum sample size $s \geq (\frac{1}{2\varepsilon^2})\ln(\frac{2}{\alpha})$, which represents the amount of scanning packets required by each source for achieving acceptable accuracy in the estimated empirical probability functions. Our analysis resulted in selecting the error $\varepsilon = 0.02$ and confidence level $\alpha = 0.1$ (i.e., 90% confidence), with a minimum sample size $s > 3,744$ packets.

In general, we identified 112,851 compromised hosts that generated more than 4,000 packets, among which, about 82.4% were IoT. Further, we leveraged Mirai's traffic signatures [3] to identify IoT scanners with Mirai infections (47.8% of all).

Experimental Results

To compare the IAT Probability Density Functions (PDF) and detect different classes of scans, we adopt the ℓ Wasserstein distance measure, which is an extension of the Euclidean distance metric

for comparing distributional-valued data. Subsequently, we leveraged a tailored technique rooted in Principle Component Analysis (PCA) to analyze the probability Density functions [100]. This method reduces the dimensions of the data by measuring differences in a number of characteristics such as position, scale, and shape of their observed distributions. Following this approach, we transform the obtained IAT distributions to a 5-Dimensional space using the `HistDAWass` R package [101]. We used the data2hist function with manual break points, with 0.01 intervals to convert vectors of arrival times to histograms. Finally, we perform subsequent HDBSCAN clustering [102] with minimum number of neighbor points=3, min cluster size=100, and outliers threshold=0.4, to explore further correlations. Note that results are illustrated using 2 main PCA components in 2-D plots.

In what follows, we demonstrate the applicability and added-value of the proposed stochastic model through use cases of scanning activities that target two prominent destination ports/services representing Telnet and HTTP.

Telnet port 23. Telnet ports (e.g., 23/2323) have been heavily targeted by compromised IoT devices in recent years. We investigate traffic generated by 7,957 devices that targeted Telnet port 23 by transforming their packet IAT to a 5D space. Furthermore, we perform subsequent clustering using HDBSCAN to identify correlated devices, as presented in Figure 4.17. The results highlight three main groups of correlated devices, representing mainly IoT devices. Moreover, it is interesting to see that the majority of devices clustered in group #3 are labeled with Mirai signatures [60]. Furthermore, the vast majority of devices within every group follow almost the same distributions of packet IAT, as illustrated in Figure 4.18. Moreover, the scanning signatures for each group represented by the mean value of their given distributions, confirm the theoretic derivations for packet IAT, as presented in Section 4.6.1. We also fit these distributions to the theoretical models (Figure 4.18) and show that packet IAT can indeed be a distinguishable feature when comparing scanning activities generated by compromised devices, especially those performing low-rate scans.

Our analysis resulted in identifying an inter-batch arrival distribution $f(t) = 72\delta(t - 1.05)$ for devices within group #3, which translates to sending 72 packets every 1.05 seconds. Similarly, devices within group #2 were sending 25 packets every 1 second ($f(t) = 25\delta(t - 1)$). Furthermore, each group of scanners might exhibit different overall target size (e.g., number of IP addresses).

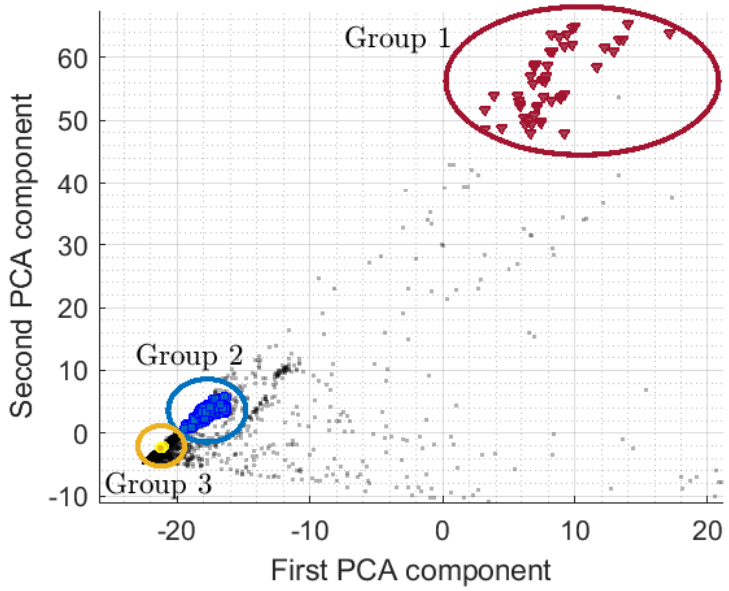


Figure 4.17: Visualizing scans targeting Telnet port 23 with correlated groups of devices using HDBSCAN.

Therefore, we had to slightly adjust the ρ values to account for these differences while fitting the distributions. Group #1 on the other hand exhibits slightly different IAT distribution, with devices sending batches of 1,100 packets following $f(t) = 0.016e^{-0.016t}$ with $\lambda = 0.016$. This very low rate approximately equals to $\frac{1}{60}$, which means that these scanners send about 1,100 packets before going idle for some random time (average of 60 seconds), possibly due to processing the response packets, before sending the next batch of packets.

In addition to packet IAT, we obtain the scanner rates from the darknet, with groups #1 to #3 having rates equal to 0.0782, 0.0983, and 0.2860, respectively. It is important to note that the observed rates do not provide sufficient details for comparing behavioral characteristics among different scanners. Nevertheless, considering that we observe similar distributions of packet IAT in each identified cluster (Figure 4.18), we may have a chance to accurately fingerprint these groups based on their IAT distributions, which in turn, characterize the implementation of their underlying scanning modules and parameters. Indeed, this highlights the importance of packet IAT analysis, which can be used along with other packet/flow features for further clustering/classification of scanning activities generated by different malware variants. In addition, our analysis shed light on

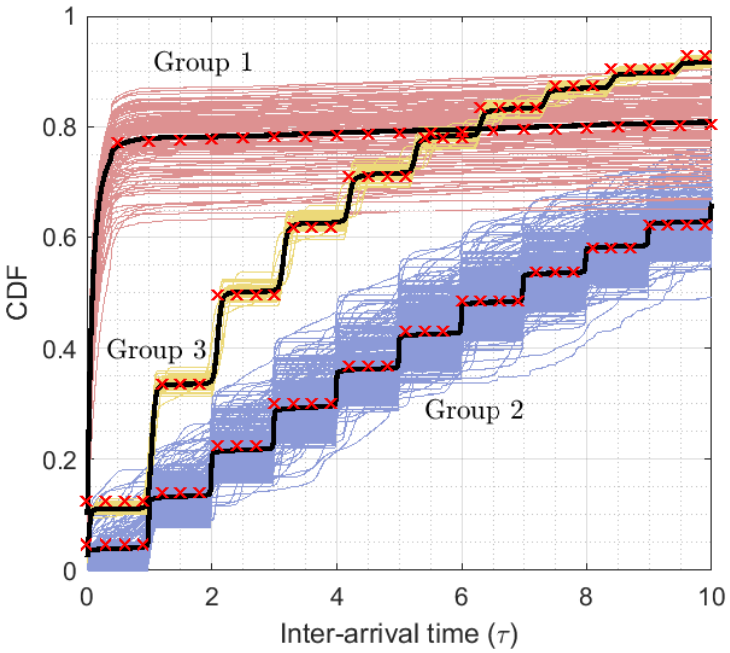


Figure 4.18: Visualizing scans targeting port 23 with the distribution of IAT CDF for all correlated groups of sources. The solid black lines represent the mean (center) of all CDFs in each group. The fitted CDFs from Propositions 1.1 and 1.2 are marked with asterisks (x).

an important characteristics of the majority of the explored scanners, which are found to generate low-rate scanning behaviors with some kind of rate limiting techniques, resulting in sending scanning packets in batches separated by specific idle times that can be leveraged to characterize and distinguish between them.

HTTP ports 80 and 8080. We investigate IAT PDFs related to 2,743 compromised devices that targeted HTTP ports 80/8080, which are also among the most targeted ports when analyzing scanning campaigns [2]. Interestingly, our analysis highlights two distinguishable dense areas in Figure 4.19, which illustrate high similarities in the scanning packets IAT distributions of the inferred IoT and non-IoT devices, respectively.

We also follow the same approach used for analyzing port 23 by performing HDBSCAN clustering on the obtained IAT distributions, and identify two distinctive groups of scanners with correlated IAT probability distributions, with group A to contain mainly IoT devices and group B to correspond to non-IoT (Figure 4.20).

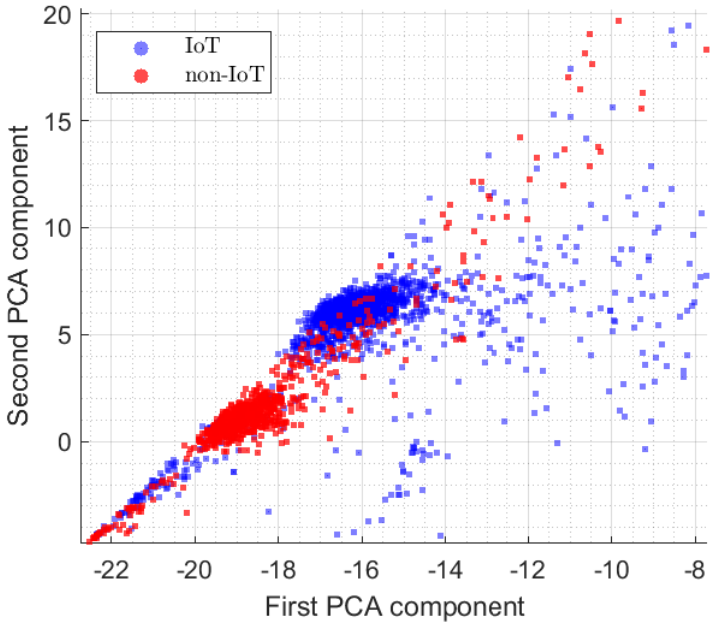


Figure 4.19: Visualizing scans generated by IoT and non-IoT devices targeting HTTP ports 80/8080 using a two-dimensional representation of their IAT distributions.

In addition, the analysis of the distributions of packet IAT for both groups show distinguishable characteristics, where devices in group A send a larger number of scans every 3 seconds, while devices in group B send relatively fewer packets per second. Further, while it is clearly observed that the distribution of IATs within group A do not follow any of the proposed models in this work (Figure 4.21), we can still estimate the corresponding IAT distributions $f(t)$ by leveraging the numerical approaches explained in [8]. Given this distinguishable behavior, it is clear that the analysis of packet IAT can in fact help in meaningfully classifying IoT and non-IoT devices based on characteristics of their scanning activities.

4.7 Summary and Concluding Remarks

We introduced a practical approach for detecting and characterizing IoT-generated scanning campaigns. More specifically, by leveraging IoT device information and over 6 TB of passive network traffic collected at a large-scale network telescope, we identified over 8,000 compromised IoT devices that were involved in a number of distinct scanning campaigns on the Internet. In fact,

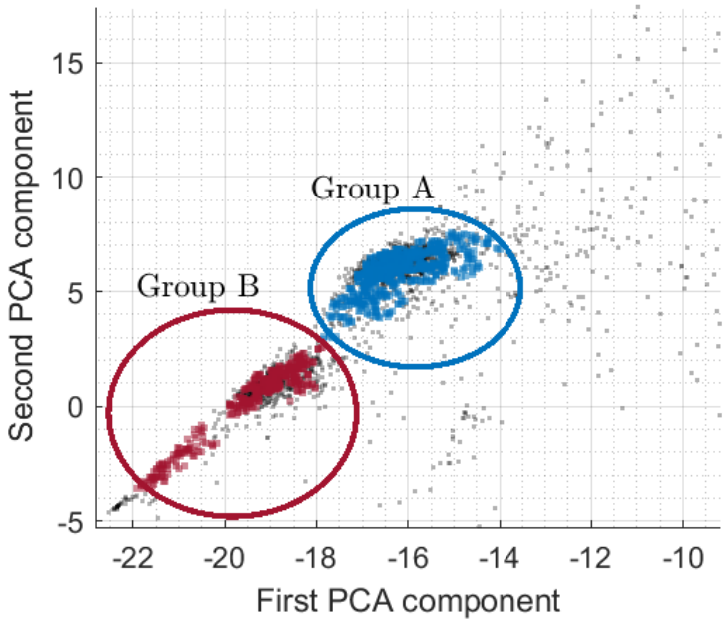


Figure 4.20: Visualizing scans targeting HTTP ports 80/8080 with correlated groups of devices using HDBSCAN.

the multi-stage investigation of the devices and their generated scanning campaigns shed light on behavioral characteristics of the underlying IoT malware/botnets. Moreover, while our results corroborate findings with respect to known IoT malware/botnets, they extend our knowledge towards discovering emerging malware/botnets, which tend to target new vulnerabilities. In addition, we provide insights on the persistence and evolution of IoT-generated scanning campaigns over time. After all, while our findings shed light on the current state of exploited IoT devices, we also lay the foundation for future work towards building scalable, Internet-wide IoT threat detection systems that can help in building a better understanding of the threats landscape while developing proper countermeasures to limit their impact on future operations.

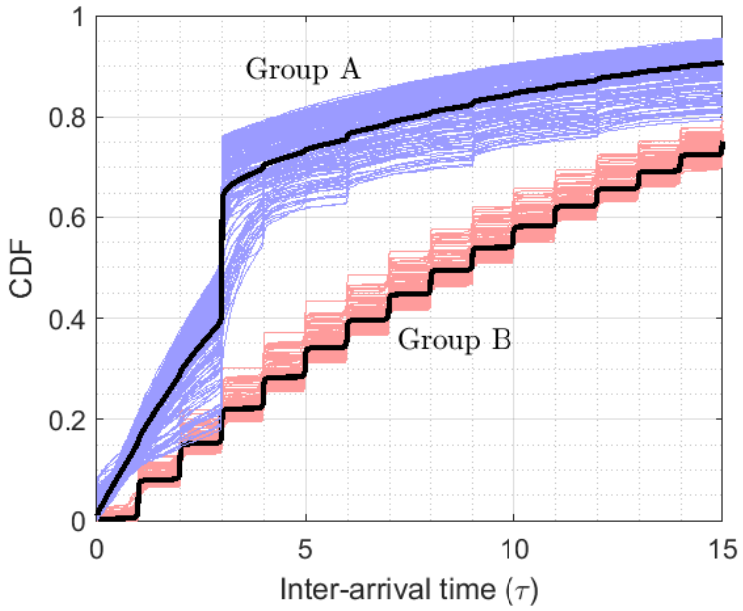


Figure 4.21: Visualizing scans targeting HTTP ports 80/8080 with the distribution of IAT CDF for all correlated groups of sources. The solid black lines represent the mean (center) of all CDFs in each group.

Chapter 5

A Scalable Platform for Enabling the Forensic Investigation of Exploited IoT Devices and their Generated Unsolicited Activities

5.1 Overview

Recent large-scale attacks unveiled an important role of compromised IoT devices as effective attack enablers, which can be utilized to generate unsolicited activities within well-coordinated botnets [3, 4, 103]. For instance, the `Mirai` botnet utilized millions of compromised IoT devices to execute one of the largest targeted Denial-of-Service (DoS) attacks [3]. On the other hand, while the `Hajime` botnet was not used to perform such attacks yet, the in-depth analysis of the botnet reveals its sophisticated design and extended capabilities, which makes it more powerful than previously detected botnets in terms of infiltrating IoT devices at scale [104].

The work was presented at the Digital Forensics Research Workshop EU (DFRWS 2020) and published at the Forensic Science International: Digital Investigation Journal [9].

In order to mitigate and prevent large-scale IoT-driven cyber attacks, there is a need to possess an Internet-scale perspective of the exploited IoT devices and their unsolicited activities over a period of time. This however, is challenging due to the shortage of empirical data on the deployment of IoT devices, and the lack of scalable cyber-threat intelligence reporting and analysis capabilities that can trigger informed decisions for in-depth forensic investigations in near real-time [79]. Furthermore, given that IoT-tailored malware heavily rely on large-scale Internet reconnaissance activities to propagate by exploiting vulnerable IoT devices at scale [3, 4], detecting and analyzing these scanning activities can provide useful insights on the compromised IoT devices and the characteristics of their underlying malicious operations and infrastructure (e.g., IoT botnets).

In this work, we address these challenges by developing a system that facilitates effective, efficient, and cyber forensic research in the context of IoT devices by providing an infrastructure for enabling a number of operations for detecting exploited IoT devices and fingerprinting their unsolicited activities. The automated system leverages a multi-stage, data-driven methodology by utilizing passive network telescope data (darknet) along with IoT device information obtained from an online IoT device search engine (`Shodan` [45]). Furthermore, the system leverages `Apache Spark`, a big data analytics framework that supports distributed computing to achieve scalable and near-real time operations.

The system is evaluated using 4TB (120 hours) of IoT-generated unsolicited traffic captured "in the wild," to identify 27,849 exploited IoT devices that generated over 308M packets, among which, the majority were scanning packets (about 300M). Moreover, while the system supports various views for macroscopic and fine-grained monitoring and analysis of the detected activities, it utilizes behavioral characteristics of IoT devices in terms of aggregated flow features to support the implementation of a number of network forensic applications such as detecting and fingerprinting scanning campaigns, investigating campaign persistence and evolution, inferring IoT botnets, and identifying IoT DDoS victims.

5.2 Contributions

Along this line of thoughts, we frame the contributions of this work as follows:

- We implement a scalable system that enables network forensic investigations through inferring compromised IoT devices and characterizing their unsolicited activities. The system, which utilizes IoT device information and passive network traffic captured at a large network telescope, leverages the capabilities of a big data analytics framework (`Apache Spark`) to implement multi-level data-driven methodologies rooted in data mining and unsupervised machine learning.

- We discuss the network forensic capabilities of the implemented system to support several operations including but not limited to: monitoring and fingerprinting unsolicited IoT-generated activities, inferring compromised IoT devices and characterizing the generated scanning campaigns, identifying IoT devices that have fallen victims of DDoS attacks, inferring IoT botnets, and performing temporal network forensic analysis.

- We evaluate the effectiveness of the system by analyzing over 4TB IoT-generated traffic over 5 days and identifying more than 27,000 IoT devices that generated about 300 million unsolicited packets. More importantly, the results of our performance evaluation affirm the scalability of the system with respect to large amount of analyzed network traffic, while generating results in near real-time.

5.3 Design and Implementation

In this section, we present the design and implementation of our proposed system, which consists of four main components, a shown in Figure 5.1.

5.3.1 IoT Data Collection Module

The aim of this module is to obtain IoT device information and process it for further use in the system. A common approach for detecting IoT devices is to perform active scanning of the Internet address space and subsequent banner analysis. Indeed, we leverage `Shodan` [45], which is one of the largest online IoT device search engines that utilizes a similar approach to infer information on different types of Internet-connected hosts, including IoT devices. `Shodan` provides an API

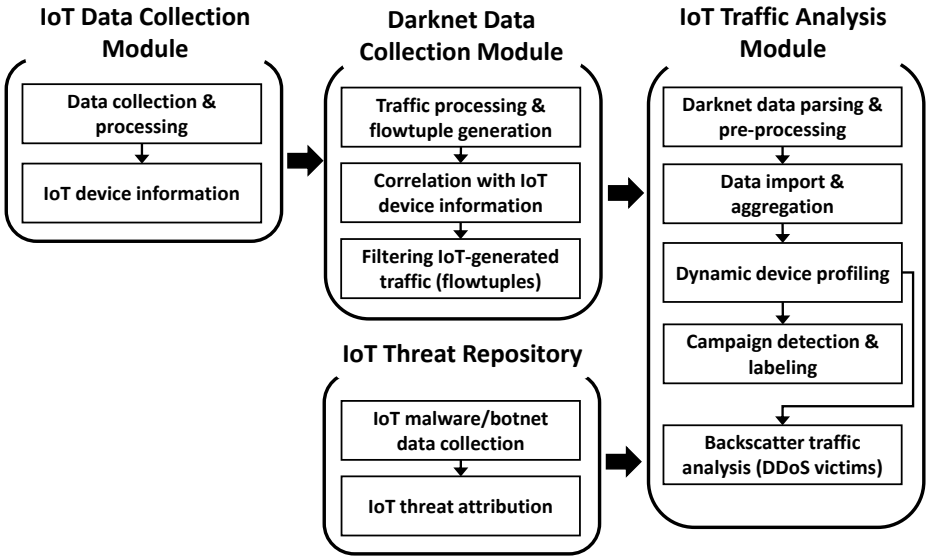


Figure 5.1: Overall architecture of the implemented system.

for searching and accessing information related to connected devices. In this work, we focus on information such as device IP address, type, operator, and location information, to name a few.

5.3.2 Darknet Data Collection Module

The system utilizes the UCSD real-time network telescope (darknet), which is one of largest available sources of passive traffic with about 16.7 million IPv4 addresses that receive over a billion packets per hour [67]. Darknet traffic represents one-way packets captured at unused, yet routable IP addresses that belong to the darknet operators. Given that traffic received at the darknet is likely to be unsolicited, the module aims at correlating the obtained IoT device information (i.e., device IP address) with darknet traffic to identify suspicious IoT-generated activities. Furthermore, depending on the implementation of the darknet, these packets undergo several pre-processing and filtering operations to eliminate noise (e.g., unnecessary traffic/information) and classify traffic categories (e.g., Internet scanning and backscatter packets). The system processes the obtained IoT-generated traffic as flowtuples, which illustrate incoming packets from a source IP to a darknet IP address during one minute time intervals. A flowtuple consists of the following nine information fields: source/destination IP addresses, source/destination ports, used protocol, time to live (TTL), TCP flags, IP length, and total number of packets sent from a source IP to a destination IP address (per

minute).

5.3.3 IoT Traffic Analysis Module

The IoT traffic analysis module, which utilizes `Apache Spark`, consists of the following main components:

Darknet Data Parsing and Pre-Processing

The IoT-generated flowtuples obtained from the darknet are pre-processed using the darknet traffic parser to identify different types of traffic according to the protocol and used flags (Algorithm2). We identify backscatter traffic [18], which represent reply packets (e.g., SYNACK) generated by IoT devices as a result of denial of service (DoS) attacks using spoofed IP addresses that belong to the darknet address space. Indeed, the analysis of backscatter packets can reveal information on benign IoT devices that were victims of DoS attacks. Moreover, we identify scanning traffic, which represents a significant portion of the darknet traffic. Given that benign IoT devices have no justifiable reason to continuously send scanning packets towards the darknet, we label these devices as compromised or exploited. The scanning traffic contains mainly TCP-SYN scanning packets, followed by a relatively smaller number of ICMP Echo requests (Ping). We also identify UDP traffic, which is less commonly used for scanning the Internet due to the stateless nature of the packets [73, 105]. Finally, the parsed/processed data (flowtuples) will be fed into the aggregation module for further analysis.

Data Import and Aggregation

The system utilizes *Apache Spark*'s *DataFrame* API to import processed darknet flowtuples into distributed collections of data organized into named columns (DataFrame) [106]. Given the imported flowtuples, the data aggregation module is implemented by utilizing a set of methods to group IoT generated traffic per source IP address, while aggregating IoT-generated traffic over specified discrete time interval(s) to obtain different views of the compromised IoT devices and their behaviors over various analysis periods. For instance, a macroscopic view of the data is presented

Algorithm 2: Parse flowtuples.

Input: Darknet-specific hourly flowtuple files F
Output: A set of dataframes F' representing parsed flowtuple files

```
for f ∈ F do
    if f ∉ Processed then
        Processed = Processed ∪ {f}
        Open f for reading
        line = f.readNextLine()
        while line do
            key = parseLine(line)
            if key == startBlock then
                minute = getMinute(line)
            else
                if key == startFlow then
                    while line ≠ EndFlow do
                        line = f.readNextLine()
                        flowInfo = getFlowData(line)
                        if flowInfo ∈ {TCPSYN, UDP, ICMPREQ, BACKSCATTER}
                          then Data.append(flowInfo) ;
                        line = f.readNextLine()
                    end
                end
            end
            line = f.readNextLine()
        end
        f'.writeData(Data)
        F' = F' ∪ {f'}
    end
end
return F'
```

through summarizing IoT-generated traffic over the analysis intervals (i.e., generated packets, number of compromised IoT devices, etc.). In addition, IoT traffic is combined to identify aggregated flow features per IoT device with different levels of interval granularity (e.g., per minute or per hour), which is utilized to infer temporal characteristics of IoT devices. This feature can be handy when analyzing scanning campaigns and their evolution over time, as described in Section 5.4.2.

Dynamic Device Profiling

As summarized in Algorithm 3, the systems utilizes the data aggregation outcomes to create a dynamic profile for every active IoT device over accumulative analysis intervals. These profiles

contain a list of IoT device information including but not limited to: source IP, targeted destination ports and IP addresses, aggregated flow features, traffic statistics and summaries, and device info (e.g., type location, ISP). The device profiles are dynamically updated after processing every input file over the accumulative time intervals. However, in order to maintain scalability and avoid accumulating unnecessary data, the system maintains a last seen flag to clean out IoT devices after a number of inactive intervals. These device profiles, which consist of device-specific measurements and information, are stored in JSON files in order to be used for further analysis when necessary.

Algorithm 3: Device Profiling.

Input: Parsed hourly flowtuple files F', aggregate interval I
Output: Device profiles D

$configureSpark()$
$spark = buildSparkSession()$
$t = 1$
$merData = \emptyset$
for $f' \in F'$ **do**
    $d = spark.readData(f')$
    if $t \ mod \ I == 1$ **then**
        $Data = d$
    else
        $Data = Data.append(d)$
    end
    if $t \ mod \ I == 0$ **then**
        $aggData = aggregat(Data)$
        $merData = merge(merData, aggData)$
    end
    $t = t + 1$
end
$spark.writeData(D, merData)$
$spark.stopSession()$
return D

Traffic aggregation and device profiling can result in several outcomes. In terms of backscatter traffic, the aggregated traffic reveals the intensity and duration of inferred DoS attacks towards the IoT devices. On the other hand, given that adversaries leverage controlled botnets to perform Internet-scale probing activities to identify hosts that run certain vulnerable services, the outcome of the module is used to profile IoT devices based on their scanning objectives (targeted ports) and overall scanning behaviors (aggregate flow features) over a period of time.

Campaign Detection and Labeling

Given that compromised IoT devices are utilized to scan certain vulnerable services/ports, the system groups the identified devices into correlated scanning campaigns according to their scanning objectives. Furthermore, given that orchestrated scanning campaigns performed by botnets tend to generate similar behavioral characteristics over a period of time, the system implements subsequent clustering using unsupervised learning techniques to identify IoT botnets. These botnets are then labeled for use in further investigations. It is also important to note that data aggregation and campaign detection/labeling processes are performed continuously over specified time intervals, and therefore, the inferred campaigns and botnet labels are updated periodically to account for any changes in the involved IoT devices and their behavioral characteristics. This is an important feature of our implemented system as it enables detecting temporal changes in the behaviors of the compromised IoT devices acting within a coordinated botnet.

5.3.4 IoT Threat Repository

The system maintains a local threat repository, which is built by compiling various publicly available information about recently discovered IoT malware/botnets such as malicious devices' IP addresses, targeted vulnerabilities, exploited services/ports, targeted device types, and botnet/malware family, to name a few. These information will be utilized by the system to create partial labels for the identified IoT devices and their malicious activities (e.g., scanning campaigns). In addition, the system will automatically update the created threat repository with information related to new, previously undetected malicious IoT behaviors and/or exploitations using feedback loops to adapt with the evolving nature of IoT threats.

5.4 Experimental Results and Evaluation

The system is built by deploying `Apache Spark` using *PySpark* in a standalone mode on a single node, with Debian Operation System (Ubuntu 18.04 version), 8 CPU cores (`Intel Xeon(R) CPU E3-1240 v5 @ 3.50GHz`), 64GB memory, and 5TB storage space. In what follows, we describe details of the data collection, analysis results, and performance evaluation results.

5.4.1 Data Collection and Sampling

In this study, we obtain information about more than 400,000 IoT devices from `Shodan` [45]. The collected data belongs to different types of IoT devices deployed in the consumer realm such as routers, IP cameras, printers, and DVRs, to name a few. Furthermore, we processed more than 4TB of passive darknet data, which represents traffic generated by millions of IoT and non-IoT hosts towards the darknet. We correlate the collected IoT device IP addresses from `Shodan` with the processed darknet traffic to obtain traffic generated by 27,940 unsolicited devices towards the darknet. Note that while the implemented system is generic and can be fed with hourly traffic from the darknet at any time frame, for the sake of experimentation, we analyzed a large sample of darknet traffic representing 120 hours (5 days) of traffic that was captured in November, 2018. We obtained about 324.6M IoT-generated packets (308M flows), with a mean of about 2.7M packets per hour. These packets represent mainly TCP-SYN traffic (87.7%), followed by UDP (10.9%), ICMP Echo requests (0.5%), and backscatter (0.3%) traffic. Other packets such as misconfiguration, account for about 0.6% of the IoT-generated traffic.

5.4.2 Results (Applications)

In what follows, we present experimental results with respect to leveraging the developed system to analyze data and enable a number of network forensic applications and investigations.

Monitoring Unsolicited Activities: A Macroscopic View

The system outputs multiple high-level macroscopic views of IoT-generated traffic over the analysis intervals. For instance, Figures 5.2(a–c) provide an Internet-scale perspective of the IoT devices and their online behaviors over a 24-hour analysis interval. These views are useful for enabling early threat detection through monitoring the overall IoT activities on the Internet, while highlighting trends and temporal changes in the overall activities of IoT devices in near real-time. For instance, we found a strong correlation between the number of IoT-generated packets and the targeted destination IP addresses in the darknet (Figure 5.2(a)), which reflects typical Internet reconnaissance activities. In fact, over 97% of the IoT-generated traffic at the majority of the observed

Table 5.1: Compromised IoT devices and their generated scanning traffic type(s).

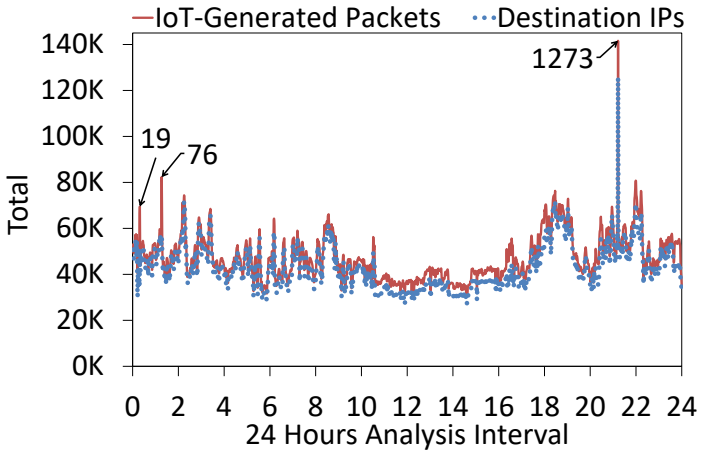
Scanning Traffic	Devices		Packets	
	Count	(%)	Count (M)	(%)
UDP	**14,314**	**51.40**	33.21	10.32
TCP-SYN	3,770	13.54	**167.88**	**52.19**
ICMP-REQ	23	0.08	0.71	0.22
TCP-SYN/UDP	9,728	34.93	118.38	36.80
UDP/ICMP-REQ	40	0.14	1.83	0.57
TCP-SYN/ICMP-REQ	36	0.13	0.97	0.30
All types	31	0.11	1.05	0.32

time intervals were TCP-SYN packets (Figure 5.2(b)), which are commonly used for scanning the Internet.

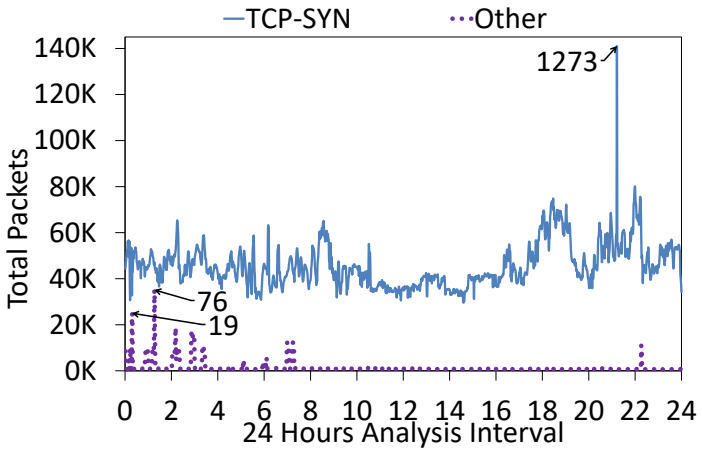
Moreover, by looking at the abrupt increases in the total number of IoT-generated packets (e.g., minutes 19, 76, and 1273 in Figure 5.2(a)) and comparing them to the detailed distribution of the packets as illustrated in Figure 5.2(b), we note that TCP-SYN packets contributed towards the majority of packets at minute 1273, while other packets such as UDP, ICMP-REQ, and backscatter, contributed towards the majority of packets at minutes 19 and 76. The system can also be used to find the number of active IoT devices that generate packets towards the darknet, which could be useful for estimating the magnitude of IoT exploitations over time. Finally, by looking at the sudden increase in the number of targeted destination ports (e.g., minutes 308, 356, and 366), we detect traces of intensive port scanning activities related to the behaviors of compromised IoT devices (Figure 5.2(c)).

Detecting Compromised IoT Devices

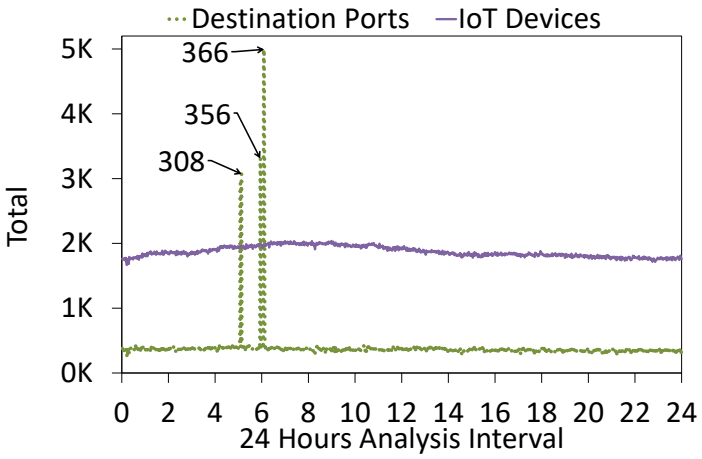
We leveraged the proposed system presented in Section 5.3 to identify 27,849 compromised IoT devices that were sending scanning packets (TCP-SYN, UDP, and ICMP-REQ) towards the darknet during the 5 days analysis interval. As summarized in Table 5.1, slightly over half of these devices (51.4%) were sending only UDP packets (32.21M packets). Furthermore, while a relatively smaller number of devices (13.54%) were generating only TCP-SYN packets, they account for significantly



(a)



(b)



(c)

Figure 5.2: Macroscopic views of the various IoT-generated packets towards the darknet over 24 hours of analysis interval (1,440 minutes).

more scanning traffic, with about 167.8M TCP-SYN packets (52.19% of total scans). In addition, about 35% of all devices generated both UDP and TCP-SYN scanning packets, with a total of about 118.4M scanning packets, representing about 36.8% all scanning packets. On the other hand, only 68 IoT devices were generating ICMP-REQ packets (about 0.2% of the scanning traffic).

Given the identified IoT devices, we utilize our system along with device information collected from Shodan to shed light on a number of properties associated with the exploited IoT devices such as device type, model, and location (hosting countries). These properties can be used to infer large-scale exploitations affecting vulnerable devices over the Internet. Indeed, the distribution of the compromised IoT devices per device type (Figure 5.3) shows about 33.7% of the devices to be routers, followed by WAP (24.4%), Firewalls (22.7%), and Webcams (10.5%), respectively. In addition, as summarized in Table 5.2, about 29% of the exploited devices were MikroTik routers, followed by a relatively smaller number of SonicWALL firewalls (16.7%), and Linksys WAPs (7%). Moreover, these devices were hosted across 192 countries (Figure 5.4), with the largest number of devices to be found in Russia (3,650), the U.S. (3,454), Ukraine (1,417), and China (1,288), respectively. The distribution of compromised IoT devices per device type, model, and country can reveal information about the overall threat landscape that targets vulnerable IoT devices.

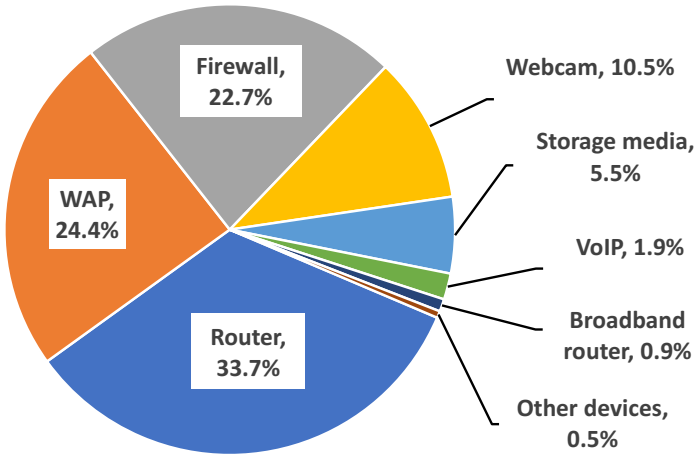


Figure 5.3: Compromised IoT device types.

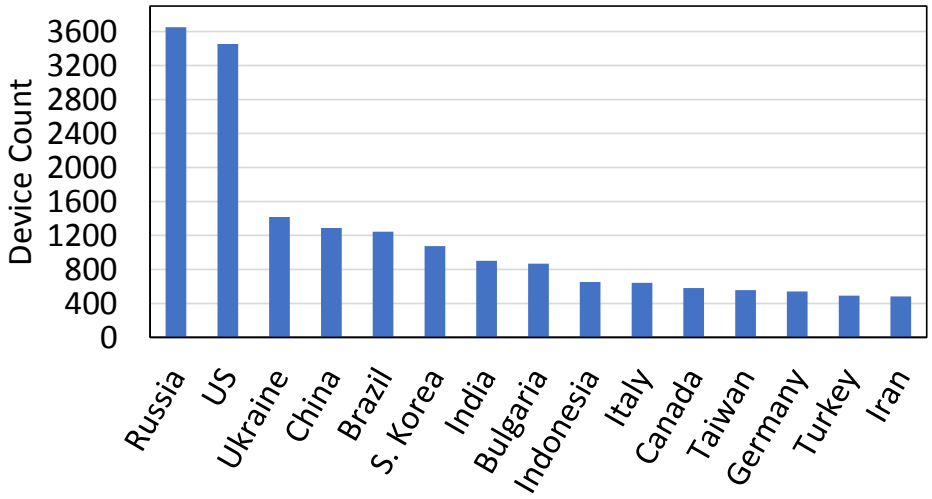


Figure 5.4: Countries with the largest number of compromised devices.

Inferring and Monitoring Scanning Campaigns

Our analysis showed that the majority of IoT-generated traffic towards the darknet consists of scanning packets (99.1%), among which about 88.5% were TCP-SYN scans, followed by UDP (11%), and ICMP Echo requests (0.5%). To identify scanning campaigns, we explored orchestrated scanning activities generated by compromised IoT devices that targeted similar destination ports/services, which represent unique scanning objectives (S_i). Prior to analyzing the scanning objectives, we filtered IoT devices that generated scanning packets less than a pre-determined threshold q. As shown in Figure 5.5, about 85% of IoT devices were found to scan less than 18 destination ports. Therefore, given the fact that the majority of IoT devices tend to scan a small number of destination ports over the analysis interval, we set the threshold $q = 20$ packets. We leveraged our system to analyze the targeted ports and identified $S^{'} = 9,523$ unique scanned destination port sets (scanning objectives) that were targeted by 14,731 compromised IoT devices.

As shown in the top 10 most common scanning objectives (Table 5.3), 932 devices (6.3%) were targeting UDP ports 28183, 32124, and 37547, while 835 devices targeted TCP port 445. Moreover, the majority of scanning packets ($> 99.5\%$) sent to ports 28183, 32124, and 37547, were UDP packets, while on the other hand, the remaining ports were almost entirely scanned by TCP packets (e.g., 23, 80, and 5555). From a different perspective, while S_1 was scanned by the largest number of IoT devices, scanning objectives associated with Telnet (e.g., S_7, S_3, and S_4) were scanned by a

Table 5.2: Compromised IoT device models (scanning).

Device Model	**Count**	**%**
MikroTik router	**8,035**	**28.9**
SonicWALL firewall	4,654	16.7
Linksys wireless-G WAP	1,944	7.0
DD-WRT supported routers	1,380	5.0
TP-LINK WR740N WAP	1,238	4.4
Cisco router	923	3.3
Talk Talk YouView box	751	2.7
TP-LINK WR841N WAP	681	2.4
Avtech AVN801 network camera	671	2.4
ZyXEL ZyWALL	618	2.2

significantly larger number of packets. This is justified by the fact that Telnet is the most targeted service, especially in the context of compromised IoT devices.

Table 5.3: Top 10 identified scanning objectives ($S^{'}$).

S_i	**TCP/UDP Ports**	**Devices (%)**	**Packets (M)**
1	28183, 32124, 37547	**932 (6.33)**	0.300
2	445	835 (5.67)	7.687
3	23, 80, 8080	735 (4.99)	11.200
4	23, 80, 8080, 37547	403 (2.74)	15.809
5	28183, 32124	209 (1.42)	0.007
6	37547	182 (1.24)	0.015
7	23, 2323	180 (1.22)	**16.849**
8	80, 8080	118 (0.80)	1.122
9	80	100 (0.68)	1.607
10	80, 443, 8080	89 (0.60)	0.019

The identified scanning campaigns highlight an important characteristics of the underlying compromised IoT devices, which targeted TCP/UDP ports that might be associated with known vulnerable services. In fact, the identified scanning objectives, which consist of a handful of common TCP services such as Telnet (23/2323), HTTP(80/8080), and HTTPS (443), are reported to be associated

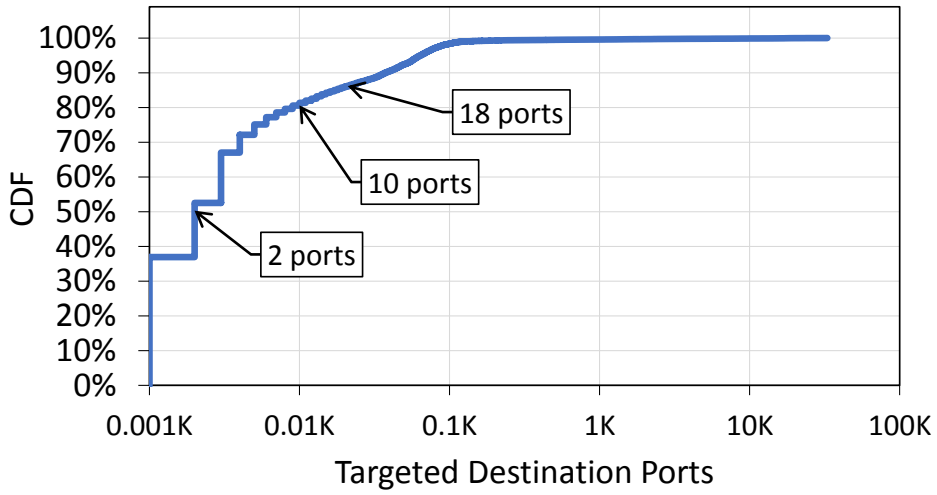


Figure 5.5: The commutative distribution of the total number of scanned destination ports by the exploited IoT devices.

with known IoT malware/botnets (e.g., `Mirai`). We also observed other targeted ports that are associated with emerging IoT malware/botnets (e.g., port 5555/`ADB.Miner` [19] and port 445/MS-DS and SMB [107]). Similarly, the remaining TCP ports in Table 5.3 are all associated with an array of known exploits that have been associated with orchestrated scanning activities generated by IoT botnets [103]. On the other hand, a considerable number of IoT devices generated scanning campaigns towards UDP ports (28183, 32124, and 37547), which to the best of our knowledge, are not associated/registered with known services. This however, implies suspicious activities that require further investigation to determine the underlying services and associated exploits (if any).

Temporal Analysis and Campaign Evolution

An important feature of the developed system is to provide the ability to monitor compromised IoT devices and their unsolicited activities over a long period of time. This feature can be used to support operational cyber security research through the identification and inferences of behavioral patterns, while enabling the analysis of temporal changes with respect to the detected scanning campaigns and their evolution over time. For instance, we leveraged the developed system to analyze campaign evolution by finding the cumulative number of compromised IoT devices within the

campaigns targeting the top 10 scanning objectives over the analysis interval, as illustrated in Figure 5.6. While these findings highlight the evolving nature of the campaigns, we also notice variable rates in terms of the number of newly detected IoT devices within the campaigns. For instance, the campaign targeting S_5, reached a steady stage early during the analysis, while the evolution of other campaigns (e.g., S_3) indicates an increasing device discovery trend. The increasing trend is likely to be justified by: (i) the spread of an infection, which exploits further devices over time, and/or (ii) the fact that adversaries may distribute a scanning campaign over controlled IoT botnets, which tend to be active in disperse time intervals while performing partial scanning tasks as part of the bigger campaign.

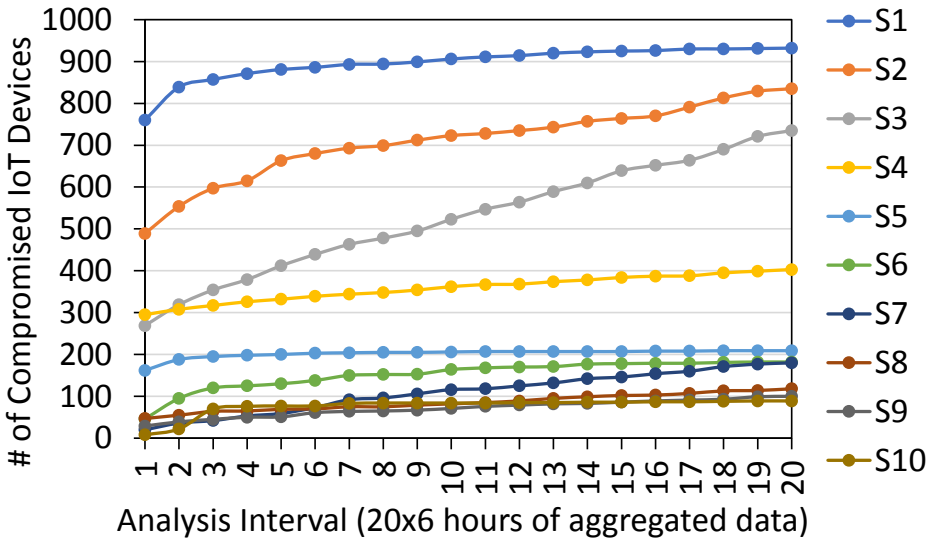


Figure 5.6: Cumulative number of exploited IoT devices within the top 10 scanning campaigns targeting S_1–S_{10}.

To investigate the latter, we looked at a sample of scanning campaigns and explored the campaign evolution in terms of the number of scanned ports by the involved IoT devices during each interval. As illustrated in Figure 5.7, while the number of exploited IoT devices that scanned all 3 destination ports within S_1 increased gradually by time, a considerable number of them were scanning a subset of the destination ports from S_1 throughout the analysis intervals. Similarly, the majority of devices targeting S_4 were scanning 3–4 ports after each analysis interval (Figure 5.7). This indicates that the devices within these two scanning campaigns did not target all specified destination ports at every time interval. Instead, they distributed the task by scanning subsets of the

final scanning objective over time, resulting in an evolving scanning campaign that targeted a fixed set of destination ports over a longer period of time.

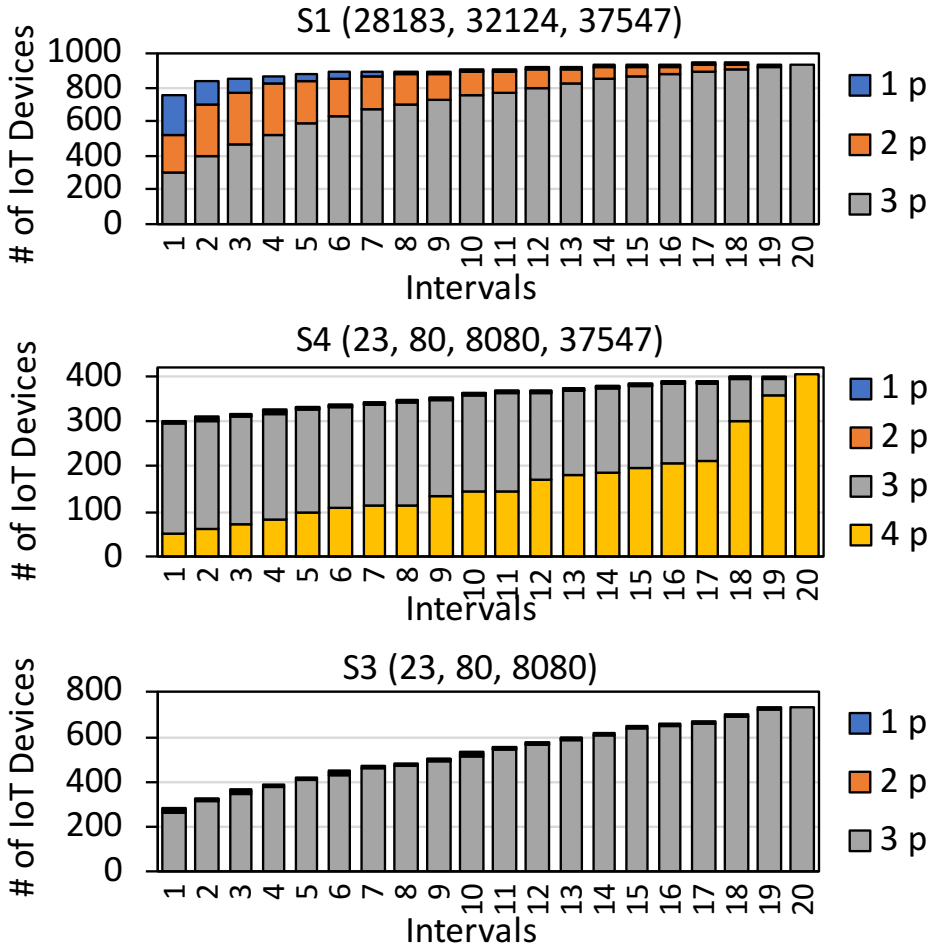


Figure 5.7: Examples of scanning campaign evolution over the analysis interval (20x6 hours of aggregated data). The campaigns target ports specified in S_1, S_3, and S_4.

In contrary, as shown in Figure 5.7, almost all devices within the identified scanning campaign were targeting the entire destination ports within S_3 at every interval. This however, reflects the used scanning strategy, which resulted in targeting all three ports during every interval (6 aggregated hours). After all, our implemented data-driven methodology, which is based on identifying scanning campaigns by finding unique scanning objectives over aggregated time periods, was indeed successful in uncovering the campaign intentions, even when the tasks were distributed among multiple devices and/or over several intervals.

Inferring IoT Botnets

It is important to realize that the identified scanning campaigns in Section 5.4.2 may reflect the behaviors of compromised IoT devices as a part of co-opted botnets, which are utilized to scan a set of predefined ports for vulnerabilities. The assumption is that different exploited IoT devices will produce similar scanning behaviors when infected by the same malware. Moreover, given that IoT malware target specific vulnerable devices, it is likely that these devices share device and firmware-specific features (e.g., TTL values). Therefore, to correlate these devices, the system is utilized to extract aggregated flow features, which represent the overall behaviors of the IoT devices over time. The system also leverages these features towards subsequent clustering of IoT devices within the scanning campaigns to infer groups of correlated IoT devices with similar behavioral characteristics (i.e., IoT botnets).

We leveraged the system to extract 16 features (Table 5.4), which consist of raw flow information from the IoT-generated packets, along with features related to the aggregated traffic over time. Note that the extracted features can always be modified to add or remove features, if necessary. The system leverages these features in a number of ways to cluster/classify compromised IoT devices into correlated groups. For instance, we leveraged the system to perform clustering within the identified scanning campaigns to detect IoT devices that produced similar flow features over the entire analysis period. The system utilizes the density-based spatial clustering of applications with noise (DBSCAN) [87], which is widely adopted due to the fact that it does not require a priori knowledge about the number of clusters, while it can detect arbitrary shaped clusters and outliers by grouping sufficiently dense regions into clusters in a spatial database [88].

The clustering analysis results for the campaigns targeting the top 5 scanning objectives (Table 5.5) highlight 7 clusters within S_1, with cluster #1 to have the largest number of members (753 out of 932). Similarly, while the analysis revealed variable number of clusters within the remaining groups (S_2–S_5), with each group to contain a main cluster with the largest number of IoT devices. This is not surprising as the majority of devices within the identified groups had similar types and models. Furthermore, given that an IoT malware might in fact target specific types/models of IoT devices, the clustering results will indeed shed light on similarities among the exploited devices

Table 5.4: Aggregated flow features for device d_i within interval I.

f_i	**Selected Features**
1–3	$U_{i,m}$: number of scanning packets from each type (m)
4	$S_P = \sum_m U_{i,m}$: combined scanning packets
5–7	$\alpha_{i,m}$: discrete prob. dist. representing the fraction of each scanning packet to scans
8	$N^{'}$: number of active intervals (minutes)
9	$A_R = \frac{b_i - a_i}{N_i^{'}}$: activity rate
10	$S_R = \frac{S_P}{N_i^{'}}$: scan rate
11	$\overline{TTL}$: average TTL value
12	$\overline{P}_{size}$: average packet size
13	Src_{Ports}: number of source ports
14	Dst_{IPs}: number of destination IP addresses
15	$Dst_R = \frac{S_P}{Dst_{IPs}}$: per destination packet rate
16	Dst_{Ports}: number of scanned destination ports

based on their correlated behavioral characteristics and aggregated flow features.

Table 5.5: Clustering results for the top 5 scanning campaigns.

S_i	ε	**#Devices**	**#Clusters**	**Clusters' Size (#Outliers)**
1	0.1	932	7	753, 45, 53, 9, 6, 3, 3 (60)
2	0.15	835	7	677, 57, 5, 13, 7, 3, 5 (68)
3	0.15	735	8	659, 3, 3, 3, 3, 3, 3, 3 (55)
4	0.15	403	7	301, 34, 15, 6, 5, 3, 3 (36)
5	0.15	209	2	179, 5 (25)

Identifying DDoS Victims

Another aspect of monitoring IoT-generated traffic is to identify devices that send backscatter packets towards the darknet. These devices are likely to be victims of DDoS attacks using spoofed IP addresses [18]. As summarized in Figure 5.8, the analysis of backscatter traffic identified 437 IoT devices, among which, the majority were routers (68%). Furthermore, slightly over half of these routers were MikroTik routers, followed by a significantly smaller number of devices from other models. This might be justified by the fact that a considerable number of the routers within

the identified DDoS victims were in fact MikroTik routers (59%), as summarized in Table 5.6.

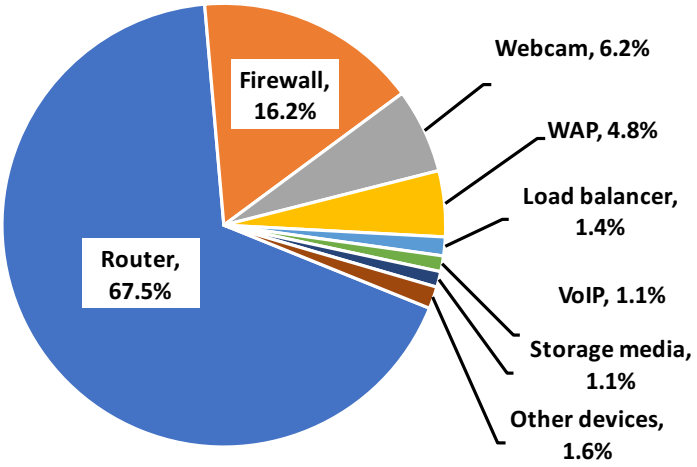


Figure 5.8: Targeted device types (DDoS victims).

Table 5.6: DDoS Victims' device models.

Device Model	**Count**	**%**
MikroTik router	**258**	**59.0**
SonicWALL firewall	33	7.4
Cisco router	19	4.3
Radware load balancer and ADC	16	3.6
Avtech AVN801 camera	15	3.4
Huawei VRP	14	3.2
WatchGuard firewall	12	2.7
Linksys wireless-G WAP	9	2.0
D-Link DCS webcam	8	1.8
Haproxy load balancer	6	1.4

Moreover, the distribution of DDoS victims over the hosting countries, as illustrated in Figure 5.9, shows that Iran was hosting the largest number of targeted devices in our data, with the majority of these devices to be MikroTik routers (102 out of 106). Considering the fact that our data contained significantly less number of IoT device that were located in Iran, this finding highlights a period of targeted DDoS attacks towards an increasing number of devices located in Iran, as perceived from the darknet.

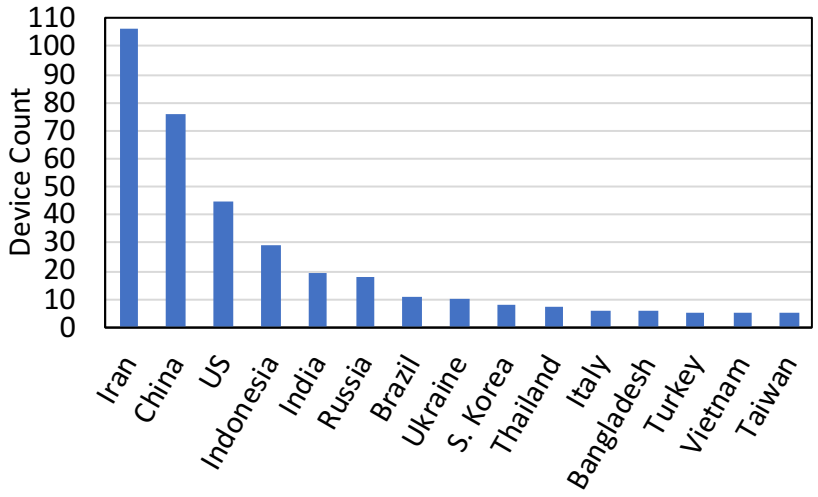


Figure 5.9: Top 15 countries with the highest number of DDoS victims.

Overall, the IoT devices (DDoS victims) generated different amount of backscatter packets towards the darknet, with the top 40 victim devices to account for about 93% of all generated backscatter packets. As illustrated in Figure 5.10, these DDoS victims are found at the high spikes, with device #120 (Radware firewall located in China) to be responsible for generating the largest number of backscatter packets (246K). On the other hand, other DDoS victims, such as device #265 (MikroTik router from Iran), generated relatively fewer number of backscatter packets (<65K). In addition to backscatter packets, about 68% (298/437) of these IoT devices were also generating scanning packets during the analysis intervals. We suspect that these devices were targeted by DDoS attacks while already being involved in scanning activities due to existing exploitations. However, confirming this phenomena is considered for future work.

5.4.3 Performance Evaluation

To evaluate the performance and scalability of the system using real life data, we sampled 24 hours of IoT-generated traffic from the collected darknet data, representing a total of 63.5M flows ($mean = 2.65M$ and $\sigma = 4.3M$) generated by 13,603 IoT devices ($mean = 4061.6$ and $\sigma = 183.9$). The performance of the system is measured during darknet data parsing, data aggregation, and device profiling processes, as described throughout Section 5.3.3. In what follows, we provide further information on the performance analysis in terms of execution time and CPU/Memory usage.

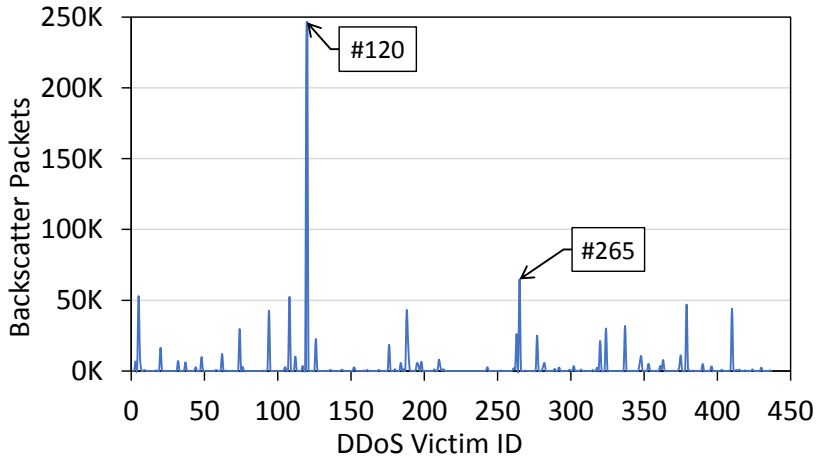
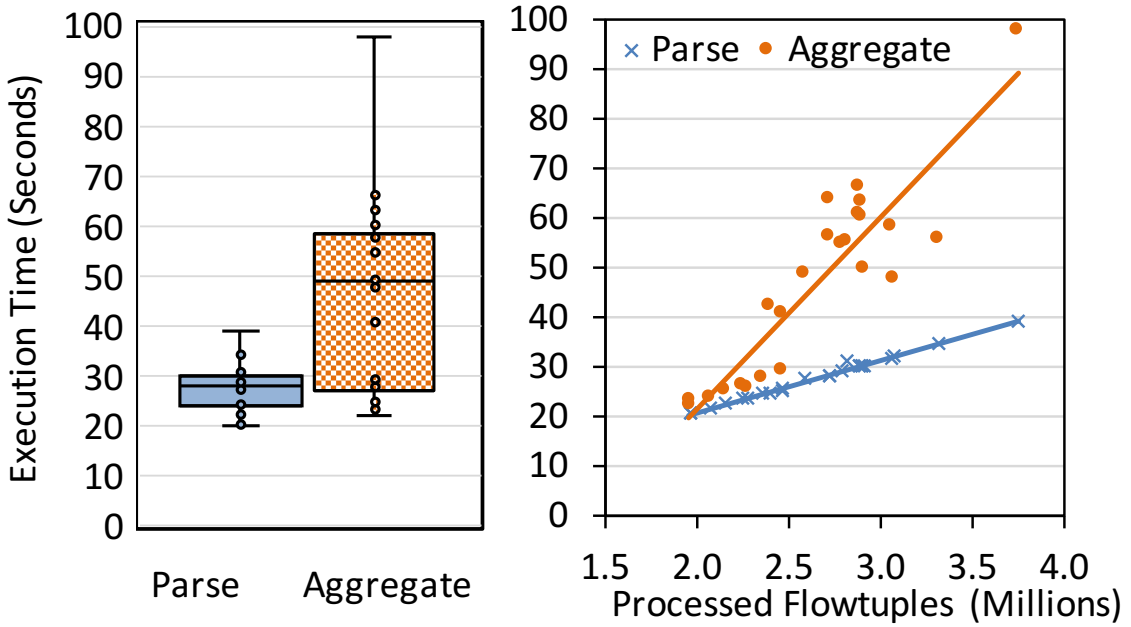


Figure 5.10: Backscatter packets generated by DDoS victims.

Execution Time

The overall execution times required to perform darknet data parsing and IoT data aggregation (Figure 5.11(a)) shows that hourly darknet data files were parsed in less than 40 seconds each, with an average of about 27.6 seconds to prepare formatted flowtuple files ($min = 20.4s$, $max = 38.9s$, and $\sigma = 4.5s$). Moreover, we observe a strong positive correlation ($r \approx 1$) between the required execution time and the number of processed flowtuples in every file, as illustrated by the Least-Squared regression lines in Figure 5.11(b). The regression analysis indicates high accuracy of the model in predicting over 99% of the variance observed in the analyzed data ($R^2 = 0.999$). This indeed can be used to predict the execution time for parsing a given data file by knowing the number of flowtuples.

Meanwhile, aggregating the parsed flowtuple files required relatively more time (Figure 5.11(a)), with an average of 46.7s per file ($min = 22.5s$, $max = 97.7s$, and $\sigma = 18.35s$). Interestingly, while we also observe a strong positive correlation between the execution time and the number of flowtuples per aggregated file ($r = 0.90$), the regression analysis indicates that the linear model can describe about 82.5% of the variance in the data ($R^2 = 0.825$). In other words, the required execution time for the aggregation processes cannot be accurately predicted by the number of flowtuples only as it depends on other factors such as the number of identified IoT devices and their associated flowtuples per analysis interval. These factors can indeed invoke a series of Spark operations (e.g.,



(a) (b)

Figure 5.11: Execution time analysis for (a) data parsing and aggregation, and (b) the correlation of execution times to the number of flowtuples in parsed/aggregated data files.

`groupBy()` and `agg()`) on subsets of data with variable length, resulting in further processing and execution overhead, respectively.

In addition, we analyze the execution time required for creating the dynamic device profiles at the end of every hourly analysis interval (recall Section 26). Device profiles are expected to grow in terms of the number of records (IoT devices) over an accumulative period of time as they depend on merging the aggregated IoT device information at any interval with previously obtained device profiles. This result in increasing the required execution time by a range between 1–59 minutes for intervals 1 to 24, as shown in Figure 5.12, respectively. In fact, the correlation analysis indicate a strong positive correlation that is modeled almost accurately by an exponential linear regression line ($R^2 = 0.99$). In is worth noting that we performed our experiments using a single node implementation of `Apache Spark`. Therefore, we can address the exponentially increasing execution times for the device profiling by increasing our resources and implementing the system on a cluster of nodes, which will significantly reduce the execution times over the accumulated IoT devices at each step.

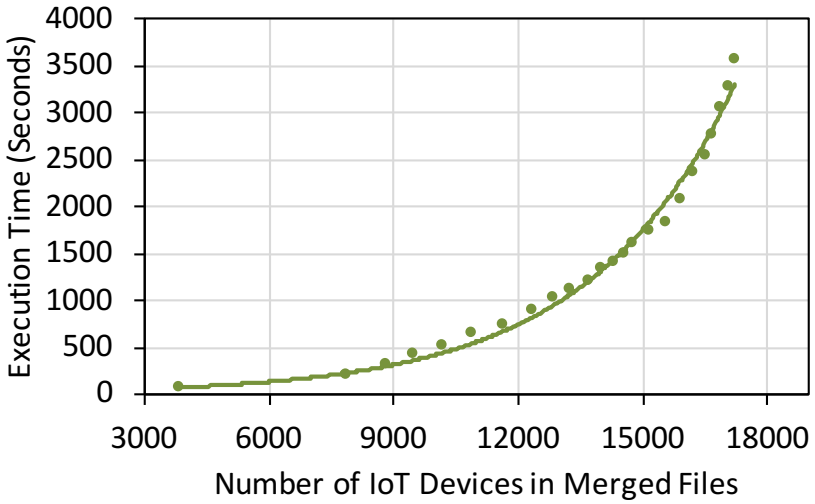


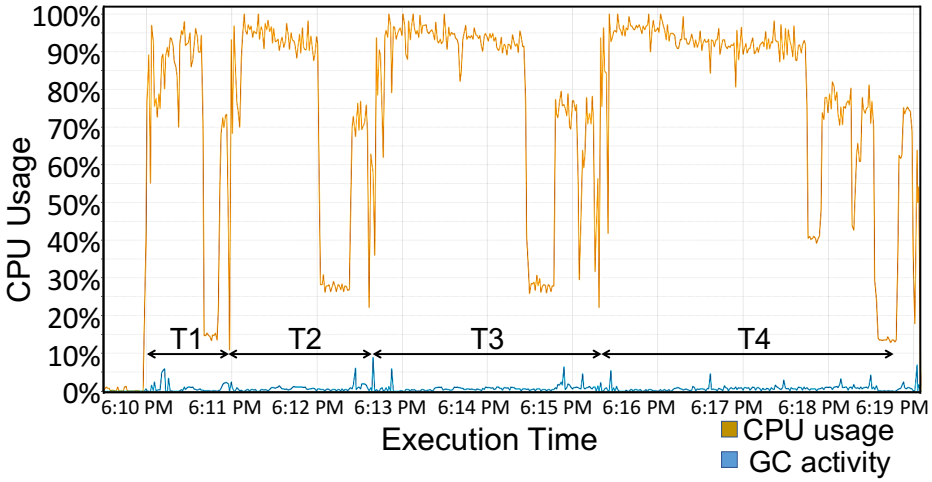
Figure 5.12: Correlation of execution time with the accumulative number of IoT devices in the merged data files.

CPU and Memory Usage

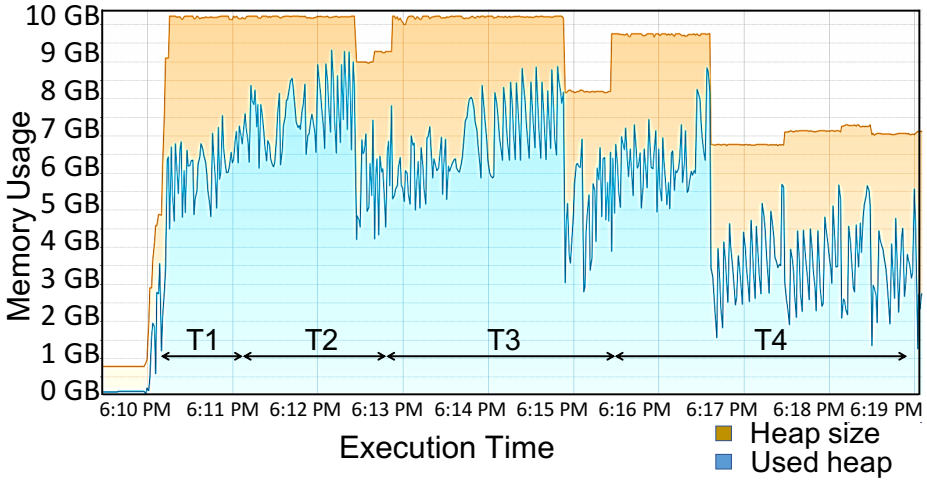
We analyzed the CPU and memory usage for different parts of the system. The darknet data parser is used for reading flowtuples from input files, parsing them and writing parsed flowtuples back into output files. These I/O operations tend to be CPU intensive and can usually use maximum CPU power. On the other hand, the memory usage for the darknet data parser stayed almost constant, with about 88MB of needed memory throughout the operations.

Moreover, we summarize the analysis of the CPU and memory usage for the data aggregation and profiling processes by illustrating the results for a sample of four consecutive aggregated and then profiled hourly darknet data files, as shown in Figures 5.13 (a–b). At every hourly time interval (T1–T4), the operations start by reading large amount of data from input files, followed by aggregation and merging (profiling) operations, which result in intensive CPU usage (Figure 5.13(a)). At the end of every interval, the data/results are written back to output files (JSON), which justifies the noticeable drop in the CPU usage ($< 30\%$) during the write operations. The sequence of operations is repeated at every hourly interval, which explains the recurring high/low CPU usage throughout the analysis. More importantly, due to the accumulated number of detected IoT devices after every analysis interval, the device profiling/merging operations required more time to process the data, as observed by the extended intervals of intensive CPU usage in Figure 5.13(a). In addition, while

the heap size was set to 10GB for this experiment, the memory usage stayed below 9.5GB, with an average of about 6.7GB of used memory over the analysis interval (Figure 5.13(b)). This is in fact very reasonable due to the size of the input data files and the number of processed flowtuples during the aggregation and device profiling operations.



(a) CPU usage



(b) Memory usage

Figure 5.13: CPU and memory usage for a sample of four consecutive hours of aggregated/profiled darknet data.

5.5 Summary and Concluding Remarks

In this work, we contribute towards empirical IoT forensics by designing, developing, and thoroughly evaluating a scalable infrastructure to enable the development of supporting technologies that help in building a better understanding of compromised IoT devices and their unsolicited activities. The developed system, which leverages the power of big data analytics frameworks, was utilized to process more than 4TB of passive network traffic collected at a large-scale network telescope (darknet) to identify 27,849 compromised IoT devices that generated more than 300 million unsolicited packets. Furthermore, we demonstrate the effectiveness of the system through a number of applied security operations to infer and fingerprint IoT-generated activities, which enable future work towards IoT-centric remediation, cyber-situational awareness, malware detection and evolution, to name a few. Finally, while the performance evaluation shows that the system can indeed execute large-scale data analysis effectively and efficiently, the implemented system is also scalable by design, as it can be extended through the implementation of `Apache Spark` on a multi-node cluster architecture.

Chapter 6

A Strings-Based Similarity Analysis Approach for Large-Scale IoT Malware Analysis, Characterization, and Family Attribution

6.1 Overview

Inferring and mitigating threats associated with the IoT paradigm requires developing a better understanding about the behavioral characteristics of the rising number of IoT malware and their underlying relationships. This is a challenging task due to the lack of information about deployed IoT devices in the user space and the insecurity of such IoT devices at scale. To address these challenges, various IoT malware data collection initiatives have been introduced over the past years [6, 45, 61], which provided fundamental knowledge and data about IoT malware. More specifically, to create a better understanding about the state of IoT malware and its evolution, various static malware

The work done in this chapter has been submitted to the IEEE Transactions on Dependable and Secure Computing (TDSC) [10]. The preliminary results has been published in the IEEE Networking Letters [11].

analysis techniques were proposed to extract information related to the structural and behavioral characteristics of the analyzed executable binaries and their underlying code structure [62]. Furthermore, such static malware analysis techniques can be effectively utilized to generate different representation of the analyzed malware binary executables, which facilitate further investigation using data-mining [63], graph-based analysis [64], and AI-based learning approaches [30].

Indeed, the analysis of the rising number of IoT malware/botnets [3,5,60] indicates that a considerable number of them are designed with two main objectives: (i) malware propagation and botnet expansion by identifying and exploiting vulnerable IoT devices, and (ii) orchestrating large-scale DDoS attacks by leveraging compromised devices as attack enablers. To fulfill their objectives, IoT malware are designed to communicate with adversarial resources such as drop zones and command and control (C&C) servers to obtain malicious command/payload and upload gathered information accordingly. Therefore, malware authors tend to embed a series of commands and IP addresses to ensure successful post-infection communication with the exploited devices, which are used in further malicious activities. On the other hand, inferring these information from malware binaries is instrumental towards understanding IoT malware interrelationships and similarities.

In this work, we leverage static malware analysis techniques to perform strings-based analysis on a large-corpus of real IoT malware samples. Our objective is to present an approach for uncovering unique characteristics and underlying interrelationships among the analyzed malware binaries. To achieve our objective(s), we leverage a specialized IoT Honeypot (IoTPOT [6]) to obtain more than 70,000 IoT malware binaries/executables that were detected over a period of 20 months (Sept. 2018 to May 2020). Moreover, we use reverse-engineering techniques to extract meaningful strings from the analyzed binaries including IP addresses associated with adversaries (e.g., C&C servers), targeted destinations (e.g., scanned ports and/or IP addresses), and command strings that reflect the underlying behaviors of the analyzed malware. Furthermore, we perform a multi-level similarity analysis to uncover underlying relationships among the analyzed malware samples. Indeed, our IP-based similarity analysis uncovered large clusters of correlated IoT malware samples that shared common adversarial resources and IP addresses. Consequently, by performing functional similarity analysis on the identified clusters of IoT malware, we were able to uncover correlated sub-components, which were leveraged to identify mislabeled malware samples. In addition, our

findings contribute towards labeling the unknown IoT malware samples while uncovering possibly new IoT malware variants that are not detected or labeled by major antivirus vendors. Finally, we leverage dynamic malware analysis to corroborate the extracted IP addresses through static strings-based malware analysis. Moreover, we extend our knowledge about the behavioral characteristics of the IoT malware by analyzing its generated scanning traffic (whenever available) and discussing its relation to the underlying functional similarities.

6.2 Contributions

To this end, we summarize the main results/contributions of this work in the following:

- We are among the first to perform a large-scale characterization of real IoT malware binaries/executables that were detected by a specialized IoT honeypot (IoTPOT [6]) during the past two years. We leverage a publicly available threat repository (`VirusTotal`) to characterize known IoT malware samples in terms of their family distribution, detection timeline, and activity duration (threat persistent). Moreover, we demonstrate the effectiveness of the IoT honeypot towards early detection of IoT malware samples while highlighting new, possibly undetected malware samples.

- We utilize reverse-engineering and static malware analysis techniques to extract meaningful strings from IoT malware binaries in terms of adversarial IP addresses and embedded commands. More importantly, we execute a multi-level strings-based malware similarity analysis approach to correlate IoT malware executable binaries and investigate their underlying correlations. Indeed, we uncover adversarial infrastructure and shared resources, which are used to operate malware-driven malicious activities. Subsequently, we leverage functional similarity analysis to address the problems of unknown malware family labeling and attribution by correlating IoT malware samples into groups with common malicious implementations. Interestingly, while our analysis enabled us to determine family labels for unknown malware samples, we uncovered groups of unknown malware samples that evolved over time to form possibly new malware families/variants.

- We leverage an implemented dynamic malware analysis testbed to extend our findings by executing malware binaries and extracting behavioral characteristics in terms of connection attempts to adversarial IP addresses and scanning traffic, whenever available. Indeed, our findings corroborate the static analysis results in terms of the extracted adversarial IP addresses for the analyzed malware samples. Moreover, we extend knowledge about the behavioral characteristics of IoT malware in terms of the scanned destination ports/services, while discussing associated vulnerabilities and IoT-specific threats.

6.3 Approach

In this work, we utilize static/dynamic malware analysis techniques to analyze a large corpus of real IoT malware binaries. Indeed, we aim at addressing the following research questions (RQs):

(1) *How can we leverage strings-based malware analysis to characterize IoT malware samples and explore their hidden interrelationships?*

(2) *How can we utilize the underlying strings-based similarities among IoT malware to address the problem of IoT malware labeling and family attribution?*

(3) *How can we leverage dynamic malware analysis techniques to validate the static-analysis results and extend knowledge about the behaviors of IoT malware in terms of scanned destination ports/services?*

To answer these RQs, we propose a multi-level approach, which consists of two main components (Figure 6.1). First, we perform static malware analysis through reverse-engineering and extraction of meaningful strings from the executable binaries to build a better understanding about the IoT malware and infer network-related characteristics and features. This component consists of data pre-processing, malware binary unpacking, and strings extraction, as illustrated in Figure 6.1. Consequently, we perform strings-based similarity analysis by leveraging the extracted adversarial IP addresses to infer hidden interrelationships among the analyzed samples and detect adversarial infrastructure. Moreover, we employ the extracted strings in terms of command sequences to

explore functional similarities among the grouped IoT malware samples within the adversarial networks. Our aim is to identify clusters of correlated malware samples, which can be used to extend the knowledge about the characteristics of the underlying malware samples and their corresponding family labels. In what follows, we elaborate further on the implemented data collection and analysis methodology.

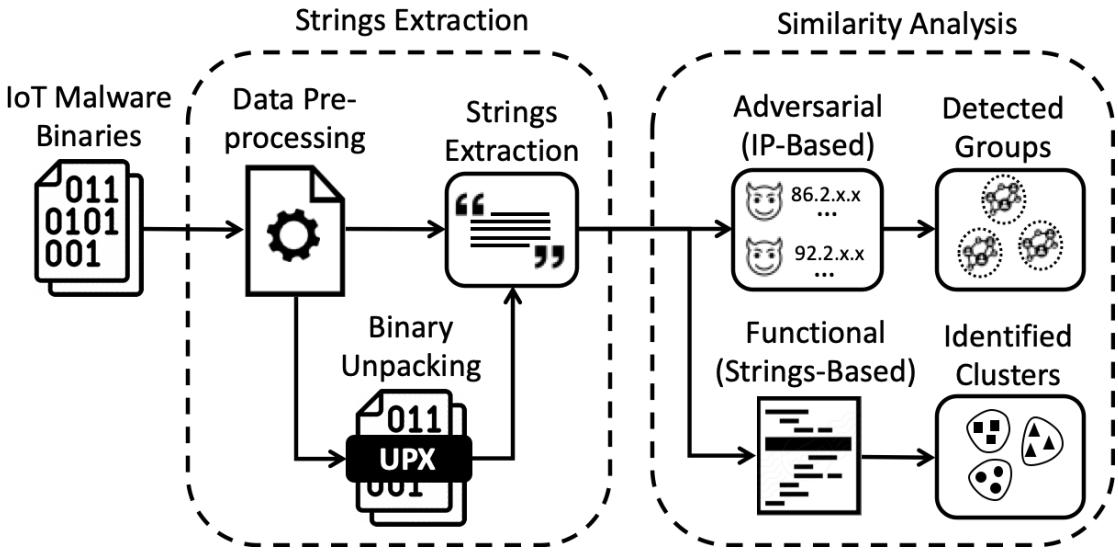


Figure 6.1: Overall approach for strings extraction and similarity analysis.

6.3.1 IoT Malware Data Collection and Labeling

In this work, we leverage a known IoT-based honeypot (`IoTPOT` [6]) to obtain over 70,000 detected IoT-tailored malware samples between September 2018 and May 2020. Among these samples, we performed pre-processing steps to filter out corrupted files and/or samples with no executable data (e.g., HTML/text), ending up with a large corpus of 49,004 IoT malware samples. Furthermore, to have a consistent malware attribution and labeling procedure, we leveraged `VirusTotal` and `AVClass` to obtain malware information such as family name, whenever available. `AVClass` is an open-source tool that uses a ranking/voting system to select the most likely family name for a given malware sample based on reported information/labels (e.g., `VirusTotal`) by multiple antivirus vendors [108]. Note that `AVClass` cannot assign malware family names when no family name/labels are associated to them by antivirus vendors, or when they are labeled with

generic (e.g., linux) malware names. We label those samples as `Unknown` for further analysis. In addition, it is worth noting that some malware samples might have not been detected by antivirus vendors, and thus, are not seen within `VirusTotal` reports. These samples are labeled as `Unseen` throughout the work.

6.3.2 Extracting IoT Malware Strings

In general, the majority of known IoT malware rely on scanning the entire IPv4 address space to identify vulnerable devices and propagate by infiltrating them. Furthermore, the updated scanning commands/payload are likely to be downloaded from malicious hosts upon malware execution, which normally store several versions of the malicious code that are intended to be executed on different CPU architectures (e.g., ARM and MIPS). Therefore, in this work, we utilized reverse-engineering techniques and static malware analysis to extract meaningful strings such as commands, payloads, and other identifiable information from the executable binaries [65]. More specifically, we use regular expressions and text-based analysis techniques to obtain IP addresses associated with possibly malicious hosts controlled by adversaries (e.g., C&C servers). In addition, we used a list of keywords (e.g., wget, tftp, cd, etc.) to search for commands associated with IoT malware operations. For instance, as shown in Listing 6.1, the IoT malware sample is trying to use an HTTP get request to download malicious payload (bins.sh) from the specified host (`http://103.*.*.*/`). Furthermore, it is clearly observed that the malware is using different techniques to download malicious scripts/payloads, as presented by the consequent instructions/commands using the TFTP protocol (e.g., `tftp 103.*.*.* -c get tftp1.sh`).

6.3.3 Packed/Obfuscated Malware Binaries

Malware packing/obfuscation is a common practice, which aims at scrambling the actual code of the malware to evade detection and prevent automatic detection and analysis using conventional methods. As a result, we were unable to extract useful strings and IP addresses from about 20,727 IoT malware binaries, representing about 42.7% of the analyzed samples. In an attempt to mitigate this malware packing/obfuscation issue, we searched the malware binaries for common indicators of packing and obfuscation methods. Interestingly, we identified about 52% of the obfuscated samples

```
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /;
wget http://103.*.*.*/bins.sh;
chmod 777 bins.sh;
sh bins.sh;
tftp 103.*.*.*5 -c get tftp1.sh;
chmod 777 tftp1.sh;
sh tftp1.sh; tftp -r tftp2.sh -g 103.*.*.*;
chmod 777 tftp2.sh;
sh tftp2.sh;
ftpget -v -u anonymous -p anonymous -P 21 103.*.*.* ftp1.sh ftp1.sh;
sh ftp1.sh;
rm -rf bins.sh tftp1.sh tftp2.sh ftp1.sh;
rm -rf *
```

Listing 6.1: Extracted strings with adversary IP address.

(10,894 samples) to be packed using `UPX` [109], which is an open source program for compressing executable files. We leveraged a combination of manual and automatic reverse-engineering methods using tools such as `UPX` [109] and `IDA Pro` [110] to unpack the `UPX`-packed binaries for further analysis. In fact, we were able to extract useful strings and IP addresses from about 84.6% of the `UPX`-packed samples (9,145 out of 10,938 samples). Finally, we were unable to unpack/decrypt about 20% of the analyzed malware binaries and thus, no useful strings/IPs were extracted from them for further analysis. One reason could be due to the fact that the adversaries leverage unique techniques for obfuscating/packing their code. It is also possible that the collected malware binaries were corrupted and therefore, contained no useful information. We consider the implementation of further malware de-obfuscation techniques for analyzing those samples for future work.

6.3.4 Strings-Based Similarity Analysis

Given the extracted strings-based information, we perform similarity analysis to explore the adversarial and functional similarities among the analyzed samples. Our objective is to find correlated groups of IoT malware samples that are owned or operated by the same adversary (adversarial similarity) by leveraging common resources such as IP addresses to operate various malware samples. Additionally, we aim at investigating functional similarity among malware samples by identifying samples that implement similar sequences of commands. To achieve our objective, we follow the steps presented in Algorithm 4 to create a graph representation of the analyzed malware samples and their pair-wise similarities. Given a set of IoT malware samples as vertices $V = \{V_1, \ldots, V_n\}$,

we create an undirected graph $G(V,E)$ such that for every pairs of vertices V_i and V_j, we identify an edge $E(V_i,V_j)$ only if the similarity between A_i and A_j is greater or equal to a minimum similarity threshold Sim_{min}. Note that the selection of the similarity analysis function/measure $Sim()$ may vary, as presented in the following sub-sections.

Algorithm 4: Similarity graph.

Input: A set of extracted strings from the malware samples A, similarity function $Sim()$, and minimum similarity threshold Sim_{min}
Output: An undirected graph of correlated malware samples $G(V,E)$

```
for A_i ∈ A do
    Create a graph node V_i
    V ← V_i
end
for each node V_i ∈ V do
    Visited ← V_i
    for each node V_j ∈ V do
        if V_j ∉ Visited then
            E(V_i,V_j) = Sim(A_i,A_j,Sim_min)
        end
    end
end
return G(V,E)
```

Adversarial Similarity

Given a set of adversarial IP addresses obtained from the malware binaries, we leverage the Jaccard similarity coefficients (Equation 6.1) to identify pair-wise similarity across all samples.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{6.1}$$

To find groups of IoT malware samples that are owned or operated by the same adversary, we leverage Algorithm 4 with the Jaccard similarity coefficients (Equation 6.1) as our $Sim()$ measure. We set the similarity threshold Sim_{min} to zero to account for all common adversarial IP addresses. As a result, for every pairs of vertices V_i and V_j representing two malware samples, we identify an edge $E(V_i,V_j)$ between them only if there is at least one common IP addresses in A_i and A_j.

Functional Similarity

We extract all strings that are associated with possible sequences of commands (e.g., `wget http://13.*.*.*/bins.sh`). The objective is to identify IoT malware samples that tend to perform similar malicious behaviors as presented by the extracted strings/commands. To achieve this, we leverage Natural Language Processing (NLP) techniques (e.g., word tokenization) to process the identified strings/commands and extract syntactic and semantic features, which are used to perform subsequent similarity analysis. The outcomes of such analysis can be used to highlight granular similarities among groups of correlated IoT malware samples based on their characteristics and underlying implementations. In what follows, we elaborate further on the feature extraction and the similarity analysis approaches.

Feature Extraction. We leverage the identified strings sequences to devise a generalized similarity analysis approach. We used NLP techniques to tokenize/vectorize the identified string sequences and extract a vocabulary of unique terms. Given a term t in every identified malware strings document d, we leverage the Term Frequency-Inverse Document Frequency (TF-IDF) approach to obtain the weighted feature vectors. TF-IDF is a common NLP processing technique, which associates each identified term in a given document with a number that represents how relevant the term is to that document. In particular, for every term t, the $tf\text{-}idf(t,d)$ (Equation 6.2) is calculated by multiplying the term-frequency $tf(t,d)$ by its inverse document-frequency $idf(t)$ (Equation 6.3), where n is the total number of malware string documents, and $df(t)$ is the number of malware string documents that contain the term t [111].

$$tf\text{-}idf(t,d) = tf(t,d) \cdot idf(t) \tag{6.2}$$

$$idf(t) = log\frac{1+n}{1+df(t)} + 1 \tag{6.3}$$

The resulting $tf\text{-}idf(t,d)$ feature vectors are then normalized to have unit Euclidean norm using the following formula:

$$v_{norm} = \frac{v}{||v||_2} \tag{6.4}$$

As a result, documents with similar terms will produce similar feature vectors, which can be used for further investigation using various similarity analysis techniques and unsupervised learning algorithms.

Similarity Measure (Cosine Similarity). Given the tf-idf vector representation of the identified command strings from the IoT malware samples, we leverage the cosine similarity measures to calculate the pair-wise distances among the identified strings. Mathematically, in contrast to the Euclidean distance, which measures the magnitude of the difference between two vectors, cosine similarity measures the orientation of the documents by calculating the cosine of the angle between two vectors projected in a multi-dimensional space irrespective of their size [112]. In general, given the $tf\text{-}idf$ vector representation of two text documents $\vec{V}(d_1)$ and $\vec{V}(d_2)$, the cosine similarity is calculated as follows:

$$Cosine\_Sim(d_1,d_2) = \frac{\vec{V}(d_1).\vec{V}(d_2)}{|\vec{V}(d_1)||\vec{V}(d_2)|} \tag{6.5}$$

Given the calculated cosine similarity, the cosine distance between two documents is simply calculated as:

$$distance(d_1,d_2) = 1 - Cosine\_Sim(d_1,d_2) \tag{6.6}$$

To find groups of IoT malware samples with functional similarities, we leverage Algorithm 4 with the cosine similarity (Equation 6.5) as our $Sim()$ measure. We set the similarity threshold $Sim_{min} = 0.8$ to account for high functional similarities in terms of common command strings. As a result, for every pairs of vertices V_i and V_j representing two malware samples, we identify an edge $E(V_i,V_j)$ only if there is more than 80% similarity between A_i and A_j.

6.3.5 Limitations

The generalizability of our results might be hampered by the fact that we rely on a single source, namely IoTPOT [6], to obtain real samples of IoT malware binaries. Furthermore, IoTPOT is deployed on a limited number of IP addresses, which mainly interact with Telnet-specific requests. Despite these limitations, it is worthy to mention that the deployed IoT honeypot (IoTPOT) have been shown to be more robust towards capturing various IoT-tailored attacks as compared to other

honeypots such as Honeyd.[1] Furthermore, the analysis of Internet-scale scanning activities generated by compromised IoT devices showed that Telnet ports (e.g., TCP 23/2323) are indeed among the most predominantly targeted ports/services by IoT malware [5, 7, 60]. In addition to that, we tried to address the generalizability of our findings through collecting and analyzing a large and representative sample of real IoT malware executables, which covers a variety of detected attacks by different IoT malware variants/families over the past two years.

Another limitation is that adversaries can employ sophisticated malware obfuscation techniques to make the malicious executable resilient to conventional static malware analysis techniques such as the used strings-based analysis. Despite that, considering the limitations within the IoT paradigm, it is assumed that such complicated obfuscation techniques, which require further resources and processing time, may hamper the malware operation and thus, less likely to be used by adversaries. Indeed, our analysis shows that we were able to extract strings from a significant portion of the IoT malware binaries without extra deobfuscation efforts. Moreover, a considerable portion of the possibly obfuscated samples were found to be implementing well-known techniques (e.g., `UPX`), which are conveniently reverse-engineered using off-the-shelf tools.

6.4 IoT Malware Data

In what follows, we summarize our analysis results with respect to a sample of 49,004 IoT malware executable binaries.

6.4.1 Identified IoT Malware Families

We leveraged `AVClass` and `VirusTotal` to obtain known malware family names/labels (Section 6.3.1). As summarized in Table 6.1, the analyzed samples belong to a handful of IoT malware families, with majority of the detected IoT malware samples (about 86.8%) to be labeled as `Mirai`, followed by a significantly fewer detected samples as `Gafgyt` (2.09%). The significant number of newly detected `Mirai` variants might be resulted from the availability of the source code, which

[1] `http://www.honeyd.org/`

Table 6.1: A summary of the analyzed IoT malware samples.

Malware Family	Count (%)	Packed Samples	Packed UPX	Adversarial IP Addresses
Mirai	**42,537 (86.80)**	**18,552**	**10,409**	**33,146**
Gafgyt	1,024 (2.09)	593	185	605
Tsunami	73 (0.15)	19	19	73
Ircbot	39 (0.08)	39	8	–
Silex	6 (0.01)	–	–	6
Bricker	4 (0.01)	–	–	4
Other	4 (0.01)	2	–	1
Unknown	4,005 (8.17)	1,029	26	3,002
Unseen	1,312 (2.68)	493	247	1,066
Total	49,004 (100)	20,727	10,894	37,903

enables code reuse [63]. Moreover, the Mirai-like malware gained much popularity among adversaries due to the effectiveness of the Mirai botnet attacks, which can be leveraged to exploit a wide range of devices with weak/default Telnet credentials [3].

Malware Family Distribution. We noticed that the distribution of Mirai and Gafgyt malware samples in our data set is not inline with the recent findings of Cozzi et al. [63], where they recently reported a larger number of Gafgyt samples in their data (obtained from VirusTotal between January-2015 to August-2018). To justify the difference, it is important to realize that we obtained the IoT malware samples from a specialized IoT honeypot (IoTPOT [6]), which is mainly focused on detecting Telnet-specific requests, and thus attracting more Mirai-like malware attacks/samples that try to exploit Telnet ports (e.g., TCP 23/2323). Moreover, our data represents a pool of more recent IoT malware samples that were collected between September-2018 and May-2020. Therefore, considering our more recent and non-overlapping data collection periods, and the fact that the Mirai malware represents a relatively newer and more popular IoT malware family as compared to Gafgyt, the prevalence of Mirai malware samples in our data is not an anomaly. More importantly, the prevalence of Mirai-like malware have been also confirmed by recent studies, which analyzed the behaviors of infected IoT devices in the wild [7, 60, 113].

Unknown/Unseen Malware Families. In addition to the identified families, we identified 5,317

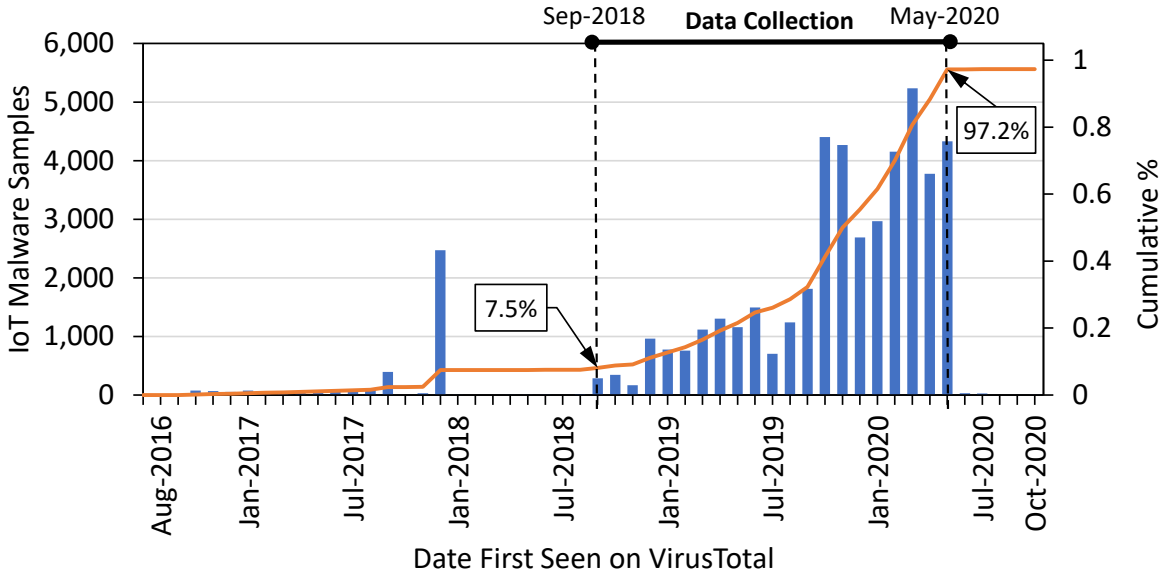


Figure 6.2: The distribution of the analyzed IoT malware samples as seen on `VirusTotal` (last updated on October 30, 2020).

IoT malware samples (10.85%) that were not associated with known IoT malware families within `VirusTotal` reports, as shown in Table 6.1 (`Unseen/Unknown`). Among these samples, 1,312 (2.68%) IoT malware samples were never found in `VirusTotal` reports (`Unseen` samples in Table 6.1). This indicates that the identified malware are either new, or have not been yet detected by major AV vendors. The remaining 4,005 (about 8.17%) IoT malware samples, which are labeled as `Unknown`, correspond to samples with generic, undecided, or unknown IoT malware family names, as perceived from `VirusTotal`.

6.4.2 IoT Malware Detection Timeline

We present the IoT malware detection timeline as perceived from `VirusTotal` reports in Figure 6.2. In general, the overall number of monthly detected/seen IoT malware samples highlights an increasing pattern during the data collection period, with a noticeable growth in the number of monthly detected IoT malware over time. Moreover, as shown by the cumulative percentage of the detected IoT malware in Figure 6.2, about 7.5% of the analyzed IoT malware samples were seen on `VirusTotal` before the data collection period. The majority of these malware samples were

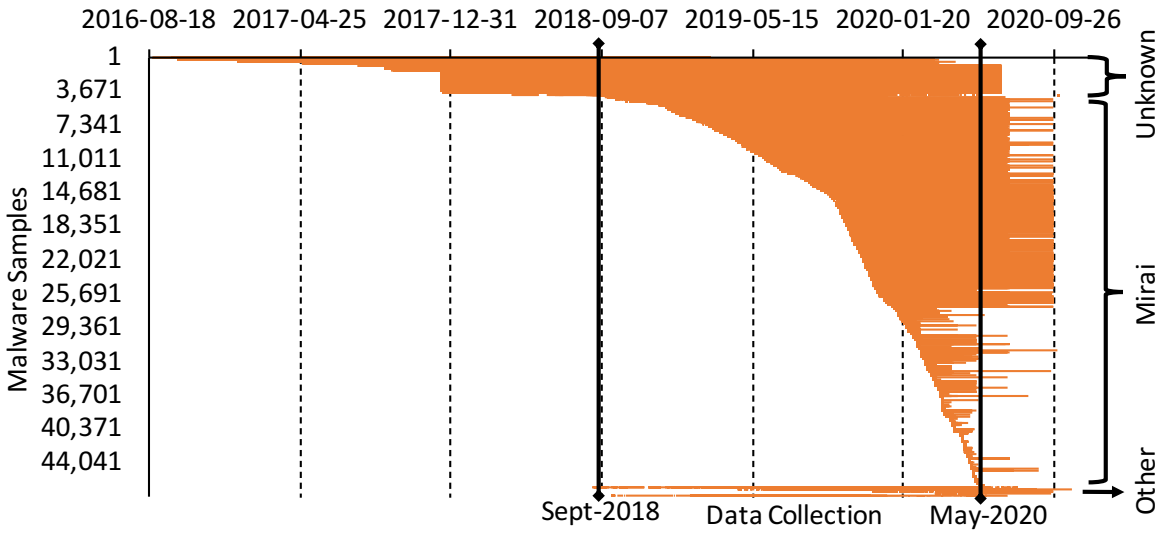


Figure 6.3: Detection timeline and duration as seen from `VirusTotal` reports.

detected at least 5 months before the start of the data collection period (around Dec-2017). Furthermore, about 89.7% of all malware samples (43,957) were detected by `VirusTotal` during our data collection period, adding up to about 97.2% of all samples, which have been already seen on `VirusTotal`. Finally, 56 samples were first seen on `VirusTotal` after the end of our data collection period, which indicates early detection of such samples by the specialized IoT honeypot (IoTPOT [6]).

Unseen Samples. Note that the IoT malware samples in this study were detected by IoTPOT [6] between Sep-2018 and May-2020. While the majority of the obtained IoT malware samples were already seen on `VirusTotal` by the end of the data collection period, it is interesting to see that 1,312 IoT malware samples (about 2.8%) were never seen on `VirusTotal` reports even after 5 months from being detected by `IoTPOT` (last checked on October 30, 2020). This gives us a strong evidence on the effectiveness of specialized IoT honeypots towards timely detection of various IoT-tailored malware. In addition, by analyzing such malware samples and attributing them to known malware families (whenever possible), we can contribute towards IoT malware detection and threat mitigation.

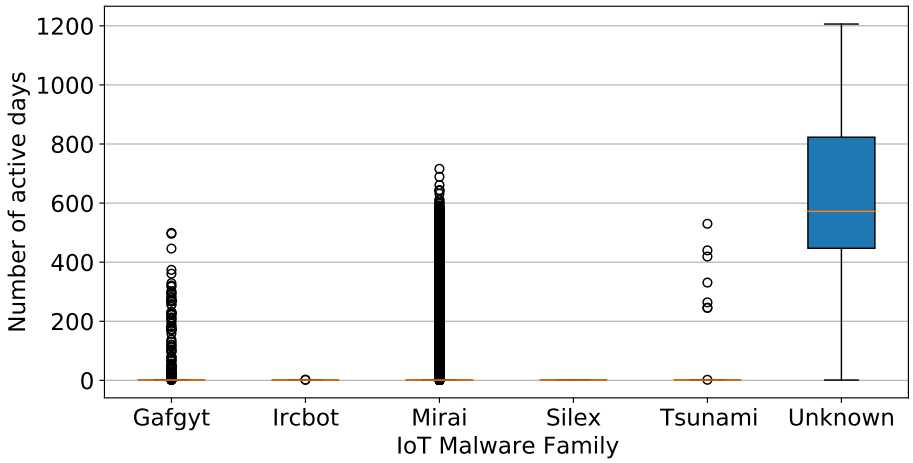


Figure 6.4: Total number of active days for the analyzed malware samples within every family.

6.4.3 IoT Malware Activity Duration (Threat Persistence)

To investigate the persistence of the analyzed malware samples, we looked at the distribution of IoT malware sample activity duration as perceived from the dates/times they were first and last seen in `VirusTotal` reports. As illustrated in Figure 6.3, it is interesting to see that while the majority (about 92%) of the `Unknown` malware samples were detected before the data collection period, they appeared to be active online for a long period of time, as compared to other malware families. This difference is clearly illustrated in Figure 6.4, where the `Unknown` samples tend to produce the largest activity duration, respectively ($min = 1$, $max = 1,206$, $mean = 601.5$, $median = 572$, $\sigma = 256.8$).

On the other hand, samples from the `Mirai` and `Gafgyt` families, which correspond to a significantly larger portion of the malware samples in our data, were associated with a relatively shorter activity duration (e.g., `Mirai`: $min = 1$, $max = 716$, $mean = 22.2$, $median = 1$, $\sigma = 74.8$). In general, these `Mirai`/`Gafgyt` samples were relatively newer in terms of their detection times. Moreover, we notice that samples that were detected later in time have relatively shorter activity duration and lifetimes. While the actual reason is not known, this could be due to a better detection and mitigation countermeasures, which blocked such malware just after they appeared online. From a different perspective, the shorter lifetime of the newly detected malware samples might be due to the adversarial behavioral changes, where they tend to leverage IoT malware as disposable resources that are discarded/recycled after every executed malicious task (e.g., Internet scanning).

6.5 Static Malware Analysis Results

In this work, we perform a series of reverse-engineering techniques and use static malware analysis to explore characteristic of the obtained IoT malware executables. More specifically, to answer our first research question (RQ1 in Section 6.3), we extract meaningful strings from the IoT malware binaries and utilize them to investigate the similarities and correlations of the analyzed malware samples. In what follows, we present details of the strings extraction and similarity analysis methodologies and corresponding results.

6.5.1 Adversarial IP Addresses

In general, about 77% of the analyzed malware binaries (37,904 out of 49,004) contained one or more IP addresses associated with adversaries (Table 6.1). These addresses are mainly used for communications with hosts that are controlled by the adversary (e.g., Listing 6.1). Interestingly, these adversarial IP addresses correspond to 7,340 unique IP addresses, which are distributed across 54 countries, with about half of them (50.46%) located in the U.S., as illustrated in Figure 6.5. Furthermore, about 2% of the identified malware samples (786 samples) contained masked IP addresses. More specifically, the analysis revealed a total of 2,083 unique IP address and/or subnet masks. These IP ranges can be used by the malware to determine specific targets for possible scanning or DDoS attacks. On the other hand, to evade detection, some malware implementation such as the `Mirai`, may include a list of "do not scan" IP addresses that are associated with high profile targets (e.g., known trap-based monitoring systems, the U.S. NSA, etc.). Confirming this however, is beyond the scope of this work and might be considered for future work.

6.5.2 Adversarial Similarity Analysis (IP-Based)

Given a set of IP addresses associated with each IoT malware binary, we performed our similarity analysis by identifying the pair-wise similarity coefficient using Jaccard index. Overall, 95.7% (36,281) of the analyzed IoT malware samples were found to be connected to one or more samples via shared IP addresses. Only 1,623 samples (4.28%) were found to be isolated, with no correlation to other samples according to the identified IP addresses.

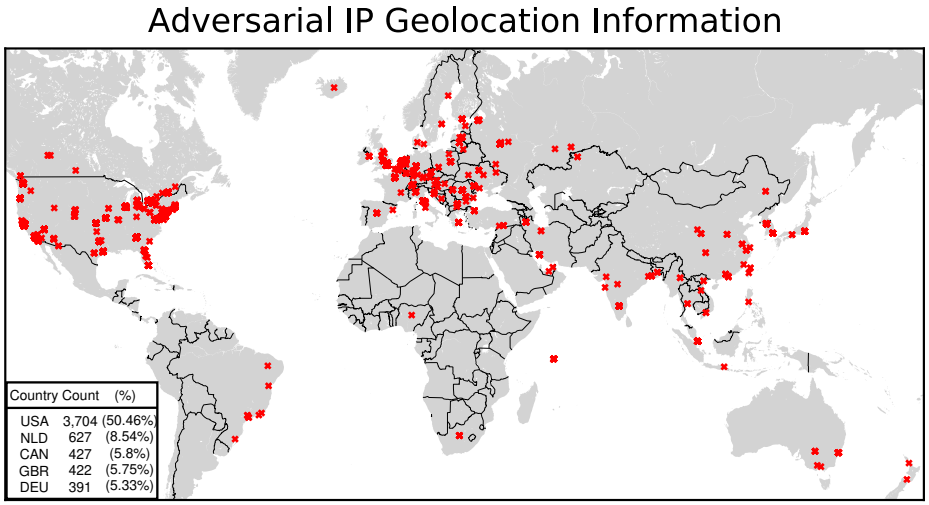


Figure 6.5: The distribution of the adversarial IP addresses across 54 countries.

Identified Connected Components. As illustrated in Figure 6.6, our analysis resulted in a graph of interrelated IoT malware binaries with common IP addresses. It is clearly observed that the analyzed IoT malware samples are grouped into various connected components (CCs). Typically, a connected component resembles a subgraph with connected vertices that are reachable through paths, while disjoint from other vertices in the graph. We identified 4,594 CCs in the graph, which are likely to correspond to similar adversaries. The cumulative distribution of the size of the connected components shows that about half of the identified CC consist of 2 or 3 IoT malware samples. Moreover, while about 99% of the identified CC contained less than 45 IoT malware samples, we identified 12 CCs that consist of more than 100 interrelated malware samples each.

IoT Malware Label within CCs. We categorized the identified CCs according to the number of malware families within them. As shown in Table 6.2, about 63.5% of the identified CCs consist of samples from a single IoT malware family, respectively. On the other hand, while a significantly fewer number of CCs contained between 2–4 malware families, the largest CC in our data, which corresponds to 13.7% of malware samples, contains samples from 5 different families, including `Unknown/Unseen` samples.

In fact, as clearly observed in Figure 6.6, some of the identified CCs contain samples from multiple IoT malware families. Furthermore, while the majority of the malware samples within the

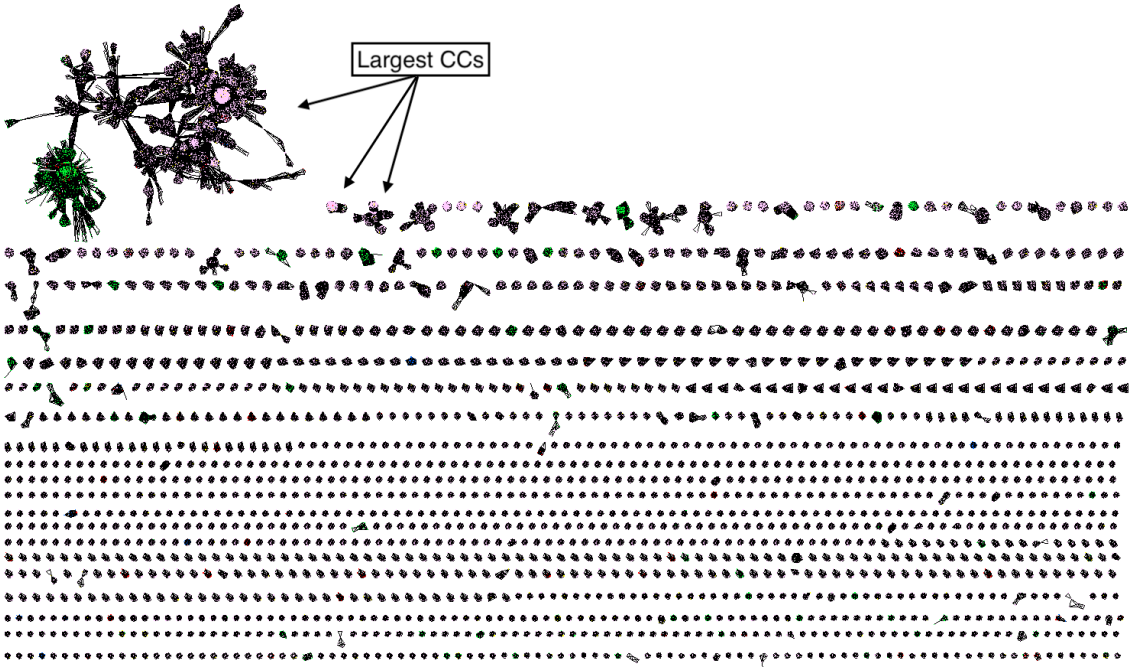


Figure 6.6: Identified connected components (CCs) based on similar adversarial IP addresses.

largest CC belong to `Mirai` (Figure 6.7), we note that many `Unknown` malware samples are also correlated with those `Mirai`-related samples. It is worth noting that the common IP addresses among malware samples do not necessarily indicate association to the same malware family. Instead, this can give us an indication about common adversarial resources, which are used to operate such malicious activities. In addition, it can be used as a starting point for further investigations to determine the actual (predicted) labels for such `Unknown` malware samples.

6.5.3 Evolution of the Adversarial Infrastructure

As illustrated in Figure 6.7, the analysis of interrelated adversarial addresses uncovered some very large components. These CCs consist of IoT malware samples that belong to multiple families, as summarized by the top 5 largest CCs in Table 6.3. To investigate the evolution of the adversarial IP-based infrastructure, we explored the number of detected IoT malware samples within the three largest CCs as specified by their detection times on `VirusTotal`. As illustrated in Figure 6.8, the number of newly detected IoT malware samples from CC#1 follows an overall increasing pattern, even before the data collection time period (highlighted area in Figure 6.8). Moreover, we notice

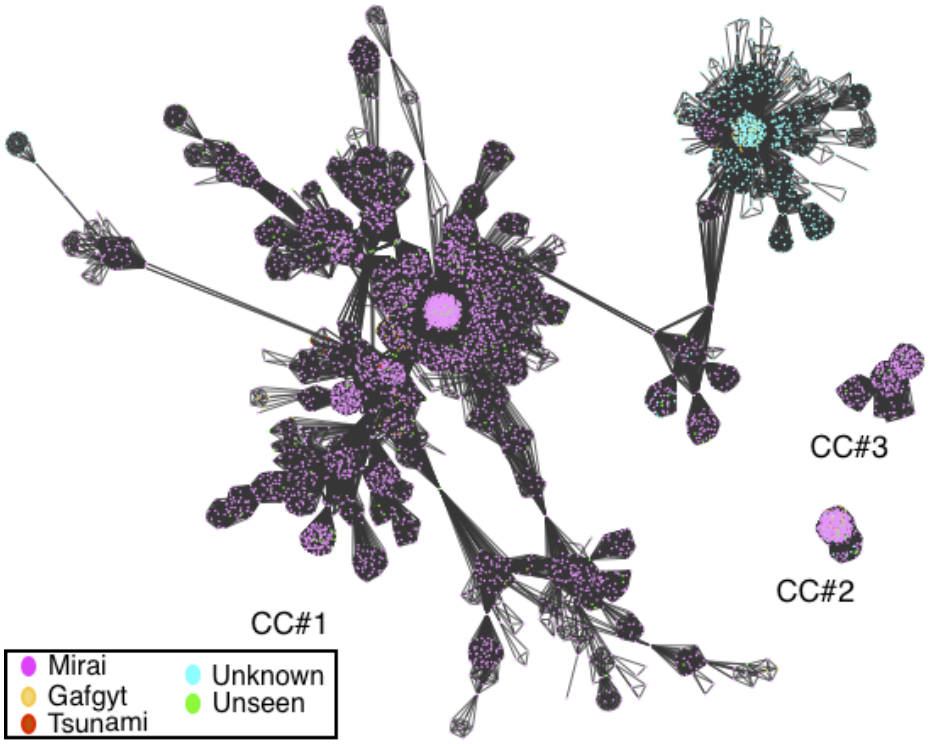


Figure 6.7: The three largest IP-based connected components (CCs).

an abrupt increase in the detected IoT malware samples around December 2017, with an increase of about 300 samples, respectively. Despite the smaller magnitude, we also notice similar abrupt changes for CC#2 and CC#3 around November-2019 and March-2020, respectively. These abrupt increases highlight a short period of intensive adversarial activities, which resulted in the detection of a relatively large samples of correlated IoT malware. Furthermore, we observe an increasing pattern in the number of detected IoT malware samples within CC#1 during the data collection period, which clearly highlights the trending activities of the adversaries within this period.

Evolution of the Largest CCs. In addition, we investigate the evolution of the largest CCs by following the relationships of the detected malware samples over time. More specifically, we looked at the formation of sub-components within CC#1, which represent the convergence of the underlying relationship among the adversarial IP addresses over time. As shown in Figure 6.8, the number of detected sub-components within CC#1 were increasing in correlation with the number of detected IoT malware samples until November, 2017. This increasing pattern indicates that the detected IoT malware samples were not highly associated in terms of the underlying adversarial

Table 6.2: Identified IP-based CCs with underlying malware samples from unique malware families.

# of Families	# of CCs	Samples (%)	Min	Max	Mean
1	**4,034**	**23,021 (63.5)**	2	161	5.7
2	522	6,520 (18.0)	2	289	12.5
3	34	1,012 (2.8)	6	132	29.8
4	3	743 (2.0)	23	646	247.7
5	1	**4,985 (13.7)**	4,985	4,985	4,985

IP addresses until we reached the intensive adversarial activities around December 2017, which resulted in detecting a relatively larger number of IoT malware samples. After that point, we observe a sudden drop in the sub-components count, which is resulted from uncovering further correlated malware samples. Following the sudden drop, we observe a period of inactivity (between December, 2017 and September, 2018), where almost no new malware were detected, and thus, no change in the number of detected sub-components.

Following the same logic, while we observe a relative increase in the number of detected IoT malware after September 2018 and towards the end of the data collection period, we notice a decreasing trend in the number of uncovered sub-components within CC#1. Indeed, a Pearson's correlation test shows a significant negative correlation ($r_{Pearson} = -0.989$) between the number of sub-components and the number of detected IoT malware samples within CC#1 between September 2018 till the end of the data collection period. This correlation clearly highlights the underlying relationship among the detected malware samples that are controlled/operated by the same adversaries, who tend to utilize their limited resources to perform large-scale distributed malicious activities. Therefore, by monitoring adversarial IP-based relationships over time, we can reveal further information about the adversarial infrastructure and controlled resources, which can be used to detect and mitigate possible orchestrated attacks.

Note that we followed a similar procedure to investigate the number of the identified sub-components within CC#2 and CC#3, where we found less than 3 sub-comments that converged quickly towards the final CC during the illustrated intensive adversarial activities around November 2019 and March 2020, respectively. We excluded these results from Figure 6.8 due to the relatively small observed measurements.

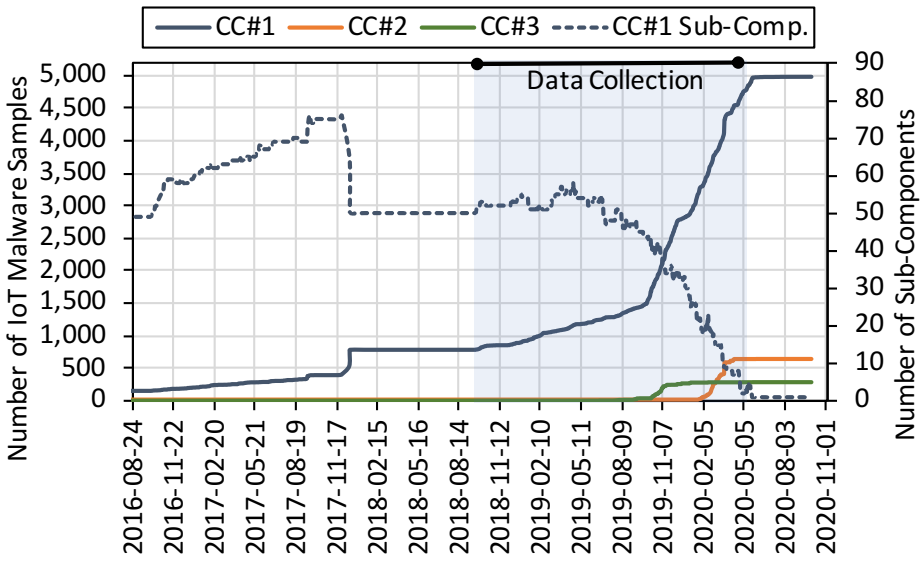


Figure 6.8: Evolution of the largest connected components as perceived from their detection times on `VirusTotal`.

Table 6.3: The largest IP-based CCs and their underlying malware family distribution.

#	CC Size	Mirai	Unknown/Unseen	Gafgyt	Tsunami
1	4,985	4,120	650/153	57	5
2	646	576	29/21	20	-
3	289	284	-/5	-	-
4	200	197	-/3	-	-
5	161	161	-/-	-	-

6.6 Unknown/Unseen IoT Malware Labeling

Given the identified adversarial IP-based similarities, and the fact that the correlated IoT malware samples belong to a variety of families including `Unknown/Unseen` samples, in what follows, we try to answer our second research question (RQ2) by performing a functional similarity analysis approach, which relies on extracting meaningful command strings from the IoT malware binaries. To achieve this, we utilize NLP techniques and similarity measures to extract features from the identified strings (e.g., commands) and correlate malware samples into groups based on their structural/functional similarities.

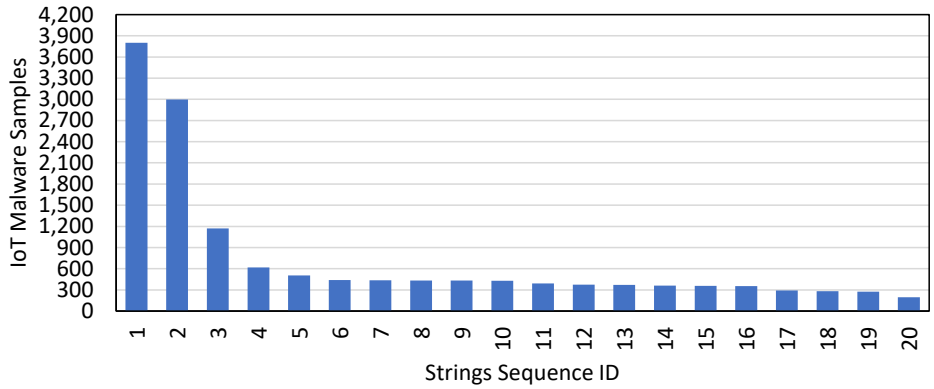


Figure 6.9: Top 20 identified string sequences and their frequencies across the IoT malware samples.

6.6.1 Functional Similarity Analysis

It is important to know that malicious code/implementation reuse is a common practice among adversaries and malware writers, especially in the context of IoT malware, where some publicly available source code such as the `Mirai` have shown to be effective towards exploiting IoT devices in the wild. In addition, it is assumed that adversaries create multiple versions/copies of their malicious executables to evade detection by distributing their malicious activities while increasing their adversarial impact by having disposable resources at hand. Thus, we hypothesize that IoT malware binaries belonging to the same family, or those created by the adversaries to serve similar objectives are likely to include sequences of commands/strings that reflect the underlying functionalities of the malicious code.

To perform functional similarity, we explored syntactic and semantic similarities by extracting strings that represent common commands/instructions within the malware binaries. Initially, we leverage a list of 33 keywords (e.g., `wget`, `tftp`, etc.) to obtain 13,970 unique strings sequences that were common across 45,165 IoT malware samples. It is interesting to see that the identified strings sequences were shared across many IoT malware samples, which may indicate similar sequence of strings/commands. For instance, the first two unique string sequence in Figure 6.9 (ID # 1 and # 2) were shared across 3,803 and 2,997 malware samples, respectively. More importantly, many of these string sequences were found to be very similar and therefore, contributing to a smaller number of common strings sequences across IoT malware.

Table 6.4: Summary of the identified sub-connected based on the strings-based similarity analysis of the top 3 largest CCs.

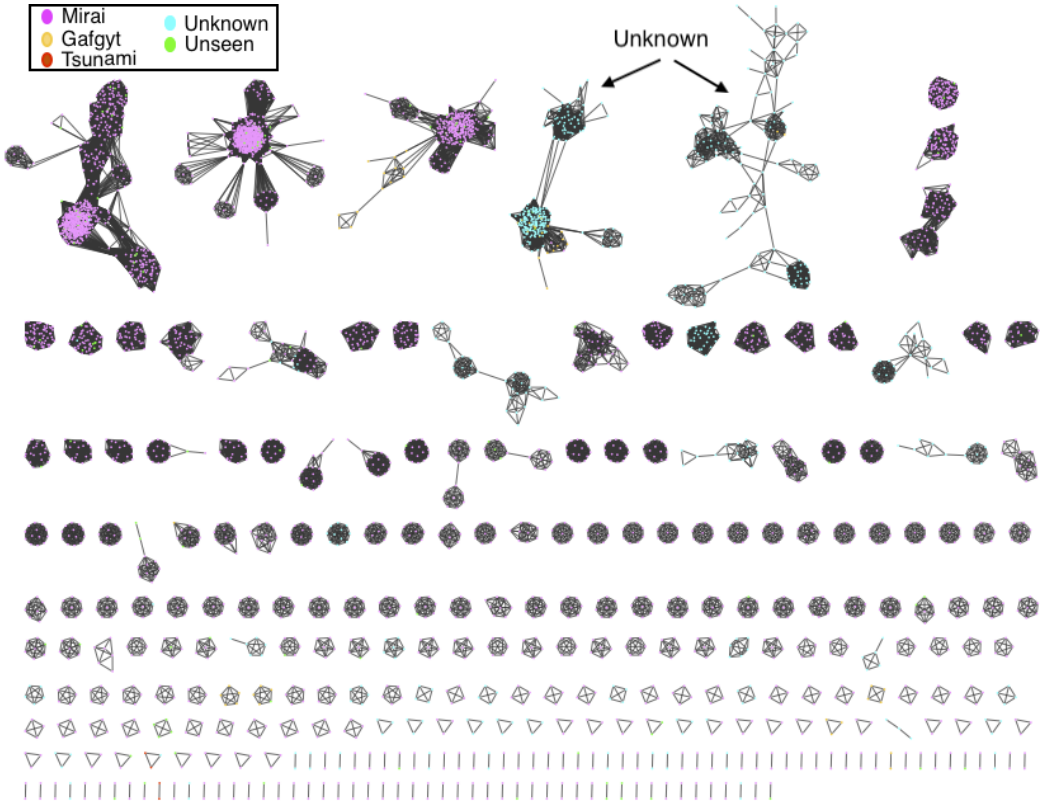
Description	**CC#1**	**CC#2**	**CC#3**
Total Nodes	4,631	638	282
Total Edges	386,711	28,167	4,462
Total Sub-components	305	24	12
Fully Connected Sub-components (%)	85.3	79.2	58.3
Average Clustering Coefficient	0.87	0.74	0.87

Correlated Sub-Components. We perform the similarity analysis by leveraging the calculated pair-wise cosine distances for the members of the top three largest CCs. As illustrated in Figures 6.10(a)–6.10(c), we use graph representations to present the identified correlated sub-components within the top three CCs. Note that an edge in the graph is added following a predefined threshold of 80%, which indicates a very high degree of similarity among the identified strings.

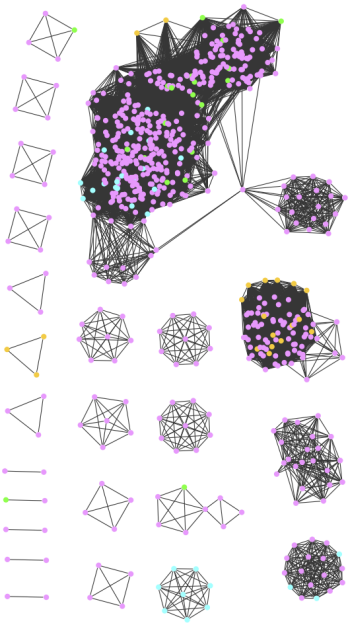
A summary of the correlated sub-components within the top three CCs is presented in Table 6.4. It is important to realize that a very small portion of the analyzed samples did not contain useful strings or contained some strings that were not correlated to other malware samples (isolated nodes). Despite that, the majority of the analyzed malware samples within the top three CCs were forming highly correlated sub-components, with high average clustering coefficient, as shown in Table 6.4. In addition, while CC#1 consists of the largest number of IoT malware samples, it was also found to be corresponding to the largest number of sub-components and malware families (305), as compared to CC#2 (24) and #3 (12).

Network Centrality Measures. As illustrated in Figure 6.10, the majority of the identified sub-components represent clusters of densely connected IoT malware samples with high functional/structural similarity (similar command strings). To investigate further, we leveraged notions rooted in graph theory to explore a number of centrality measures [114], which can help in identifying the importance of the nodes (IoT malware) and their centrality in the underlying adversarial network. Specifically, we calculated the degree centrality, closeness, and betweenness centrality measures with respect to the identified sub-components within C#1–C#3.

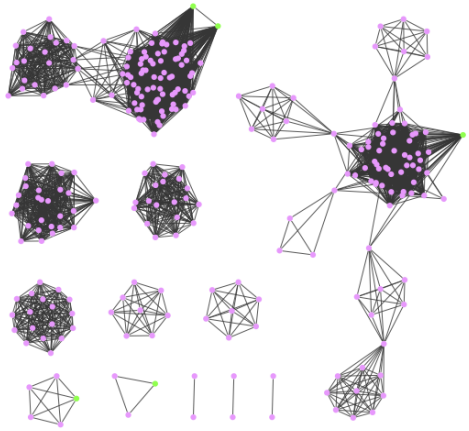
In general, the analysis of the degree and closeness centrality indicates high centrality measures



(a) CC#1 sub-components.



(b) CC#2 sub-components.



(c) CC#2 sub-components.

Figure 6.10: Similarity analysis results in terms of the correlated sub-components within C#1–C#3.

for the majority of the sub-components within C#1–C#3, which indicates that the nodes are highly connected and thus, forming dense clusters. Note that about 84% of the identified sub-components were fully connected (i.e., all nodes are linked directly), which result in the maximum degree and closeness centrality values ($value = 1$ for all nodes).

We present the centrality measures for the remaining 55 sub-components in Figure 6.11. In general, while the average degree and closeness centrality measures (square markers) for the remaining sub-components indicate relatively high values (Figure 6.11), they also illustrate the existence of nodes with smaller degree and closeness centralities measures within certain sub-components (e.g., #5, 15, and 46). In fact, such variations are clearly reflected in the shape of the sub-components (Figure 6.10), which consist of multiple dense clusters that are loosely connected to each other through few nodes. Consequently, the high betweenness centrality values within a number of these sub-components (e.g., # 45, 19, and 50) indicates the existence of bridging nodes, which connect different clusters of nodes to form the final sub-components.

It is worth mentioning that the betweenness centrality can be used to highlight important nodes that have authority over disparate clusters in a network. In contrary, it can also indicate if some nodes are on the periphery (on the edge or outside the boundary) of multiple clusters, and thus, having less importance. To investigate this, we select the nodes with the highest betweenness centrality and compared their identified command strings to find further information. Indeed, we found that they share marginal information in terms of their underlying command strings, which resulted in placing them along the boarders of dense clusters. Therefore, we can conclude that these nodes can either be removed or are duplicated to create new homogeneous sub-components with tighter relationships (similarities). On the other hand, such nodes may indicate a temporal evolution and lineage of IoT malware samples within the adversarial network of resources [63]. This is mainly due to the fact that adversaries tend to create new instances of IoT malware by reusing and modifying existing source codes and malicious executables.

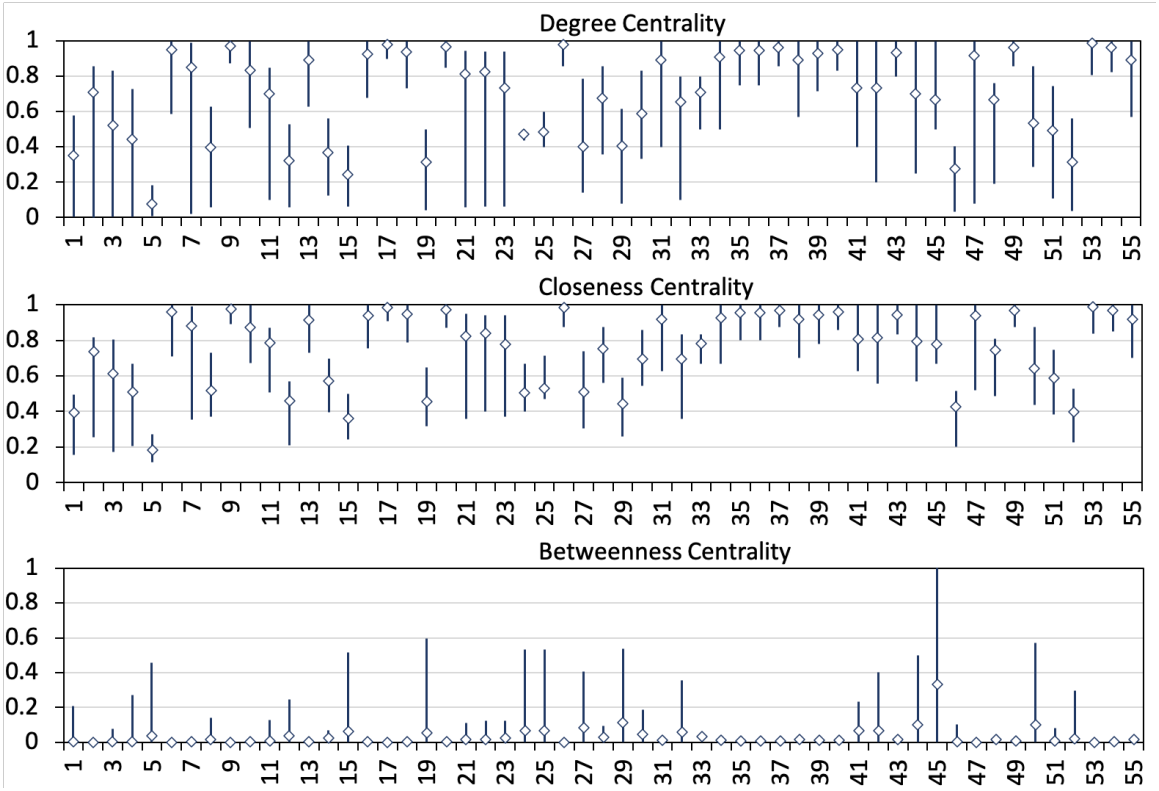


Figure 6.11: Minimum, maximum, and average centrality measures for nodes within the remaining 55 sub-components withing CC#1–CC#3.

6.6.2 Malware Family Labeling

To investigate malware family labels, we closely examined the known labels (Section 6.4.1) for all IoT malware samples within the identified sub-components. Given that the majority of the sub-components are fully connected or contain clusters of fully connected nodes, we applied a majority voting system to specify the proportion of the known family labels within each sub-components. For sub-components with four or more IoT malware samples, we specify the majority label by 75% of the population (i.e., majority vote $\geq$ 75%). For sub-components with three nodes, we specify the majority by two thirds of the population (67%).

In general, about 96.5% of all the sub-components (329 out of the 341) produced a majority vote. Furthermore, the remaining 12 sub-components were all associated with `Mirai` as almost all of them consisted of two samples, among which one was labeled as `Mirai`. Consequently, we identified 235 mislabeled malware samples, with the majority of them to fall under `Unseen` (61.7%)

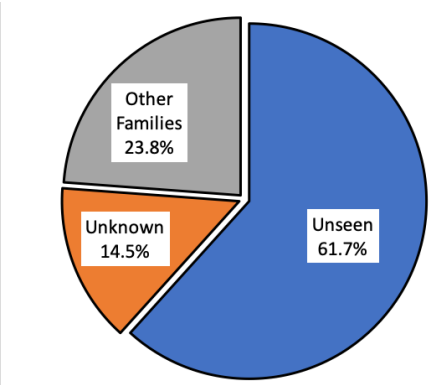


Figure 6.12: Distribution of the mislabeled IoT malware samples ($n = 235$).

and `Unknown` (14.47%) malware categories, as shown in Figure 6.12. Therefore, following our approach, we were able to assign labels for such malware samples, which were either never seen on `VirusTotal` or were not associated to any known IoT malware family (`Unknown`).

In addition, we identified 46 sub-components with a majority labels corresponding to `Unknown`/`Unseen` categories. It is interesting to see that only three of these sub-components (found within CC#1) contained a minority of malware samples with known families. For instance, we found a three nodes sub-component with two `Unseen` samples and one `Mirai`. The other two large sub-components, which are marked as `Unknown` in Figure 6.10(a), contained 25 (`Mirai` and `Gafgyt`) and 2 (`Gafgyt`) known malware samples, respectively. While having a small number of known labels cannot provide definite answers in terms of labeling `Unknown` malware sample within these sub-components, we believe that our analysis can provide a starting point for further investigation towards labeling those samples. On the other hand, the remaining 43 sub-components were purely associated with `Unknown` (40) and `Unseen` (3) labels, respectively. This however, can give us a clear indication about the evolution of IoT malware, and the effectiveness of IoT-specific honeypots towards detecting possibly new IoT malware variants, which require further investigation to identify their behaviors and possible family names.

6.7 Experimental Results: Dynamic Analysis

In addition to strings-based analysis, to answer our third research question (RQ3), we leveraged a virtual sandboxing environment [7] to perform dynamic malware analysis by executing IoT malware binaries and collecting their generated network traffic (Figure 6.13). The objective is twofold: (i) validate the extracted adversarial IP addresses from the strings-based analysis by collecting malware-generated connection attempts towards such destinations, and (ii) explore the behavioral characteristics of IoT malware by analyzing their generated scanning traffic (if any) in terms of targeted ports/services and associated IoT-specific vulnerabilities.

6.7.1 IoT Malware Sandboxing Environment

As shown in Figure 6.13, we emulate a multi-architecture environment for common CPU architectures using a virtual sandboxing environment (Chapter) on `Qemu` systems [91]. The created testing environments was used to execute malware binaries for a period of 30 minutes to capture the generated network packets at the gateway using `TShark`. We stop when reaching the end of the collection time period or when capturing 2,000 packets. We performed dynamic analysis on all the IoT malware samples within the top CCs (45,165 samples). To this end, while a considerable number of these samples were executed in our dynamic sandboxing environment, we were only able to obtain useful traffic information from 245 samples. This however, reflects the significant challenges and limitations when performing dynamic malware analysis in the IoT realm, where extra levels of customization is required to ensure proper execution of device-specific IoT malware in a controlled environment. Moreover, despite previous efforts towards addressing the limitations of large-scale dynamic analysis of IoT malware [115], addressing this open problem is beyond the scope of current work.

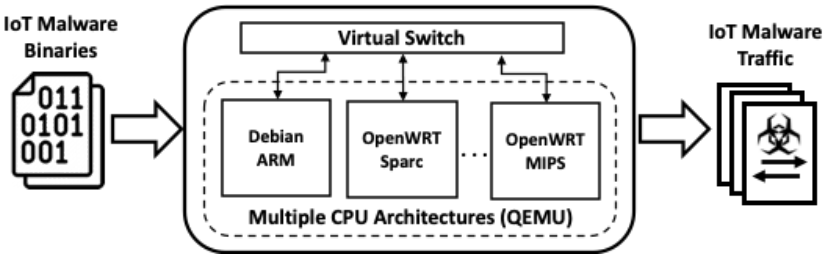


Figure 6.13: The implemented dynamic IoT malware analysis environment.

6.7.2 Obtained Network Traffic

In general, the obtained network traffic packets consists of connection attempts to adversarial addresses (e.g., C&C servers) and/or Internet scanning activities (e.g., TCP-SYN requests) targeting a set of destination ports on different IP addresses. Indeed, we were able to match the IP addresses obtained from the connection attempts with the list of adversarial IP addresses found from the strings-based IoT malware analysis, which validates our results and assumptions regarding the adversarial IP addresses. We also process the collected scanning packets in the form of `n-tuple` flows with information about the destination ports and IP addresses. Our results show that the executed malware samples targeted a total of 22 ports, with each malware generating scanning packets (mainly TCPSYN) towards a set of ports/services on various destination IP addresses. More specifically, we identified 24 unique port sets that were scanned by the analyzed IoT malware. As shown in Table 6.5, TCP ports 23, 5555, and 23/37215 were targeted by the largest number of IoT malware samples, respectively.

Functional and Behavioral Similarities. As shown in Table 6.5, the analyzed IoT malware samples generated scanning traffic towards a short list of targeted destination ports/services (P1–P10). Moreover, these scanned port sets were common across different IoT malware samples, resulting in groups of IoT malware with behavioral similarities. For instance, 149 IoT malware scanned P1 (Telnet/23) while P2 and P3 were scanned by 22 and 12 IoT malware samples, respectively.

It is important to realize that having behavioral similarities in terms of scanning the same destination port sets does not necessarily indicate that the underlying malware samples are the same. In contrary, in this work (Section 6.6.1), we hypothesized that the functional similarity based on the

Table 6.5: Top 10 scanned destination port sets ($n = 24$).

#	TCP Port Sets	# Samples
P1	23	149
P2	5555	22
P3	23, 37215	12
P4	23, 2323	8
P5	40	6
P6	37215	4
P7	8080, 37215	2
P8	443, 8080, 37215, 52869	2
P9	443, 8080	2
P10	23, 80, 5500, 8080, 37215	2

identified command strings from the malware binaries can indicate common implementation and functionalities, which result in similar behaviors. To test this hypothesis, we explored the underlying functional similarities across the IoT malware samples that scanned the top 5 port sets (P1–P5). As illustrated in Figure 6.14, the IoT malware samples that targeted P1–P5 were grouped into a total of 16 connected components based on their functional similarities. More importantly, the majority of the correlated IoT malware (13 out of 16 connected components) were associated with unique scanned port sets, which is a clear indication of common behavioral characteristics as a results of having strong functional similarity.

Moreover, we note that despite their differences in terms of the identified scanning behaviors, some malware samples showed strong functional similarities (Figure 6.14). For instance, all malware samples that targeted P5 were strongly correlated with a group of malware samples from P1. Similarly, two malware samples that targeted P4 (TCP ports 23/2323) were part of a fully connected components from P1 (TCP port 23). This however, gives us a clear indication about the adversarial behaviors in terms of reusing malicious implementation/code for the purpose of creating/operating new malware instances that serve different objectives, as specified by their scanning behaviors.

Associated Vulnerabilities. It is worth mentioning that almost all of the TCP port sets in Table 6.5 (e.g., Telnet 23/2323, HTTP(s) 80/8080/443, 5555, and 5500) are associated with known vulnerabilities that are widely scanned by the `Mirai` IoT malware variants [3, 5, 7]. Moreover, we

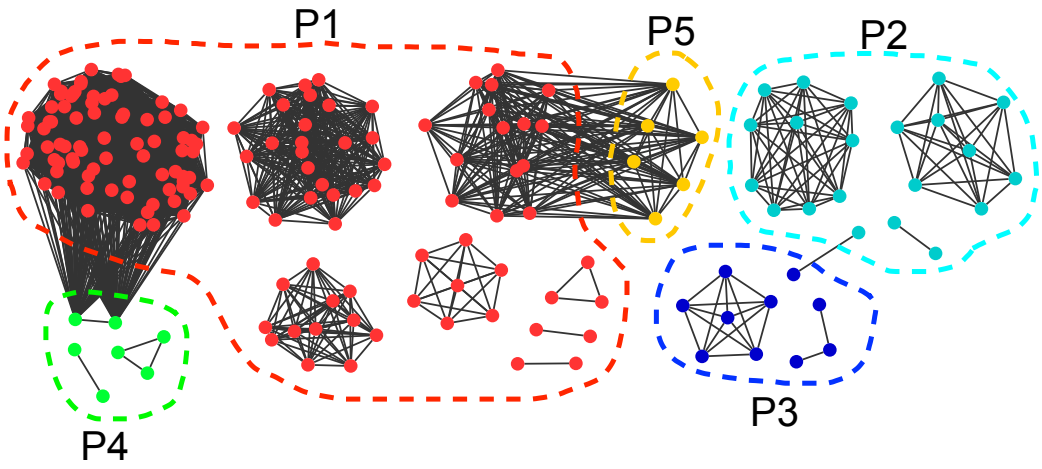


Figure 6.14: Functional similarity analysis results for the IoT malware samples targeting top 5 scanned port sets.

found 6 samples that were scanning TCP port 40, which is not commonly scanned by IoT malware. Interestingly, this port (TCP/40) is associated with a threat caused by `Midnight Commander` [22], a visual file manager which sometimes access FTP servers running at this port. In addition, we found TCP port 37215 among several scanned port sets. This port was used by newer `Mirai` variants such as the `Satori` botnet [23, 24], to remotely exploit specific models of Huawei routers by invoking a firmware upgrade action through the vulnerable Universal Plug and Play (UPnP). Furthermore, TCP port 52869 was also found to be associated with a similar vulnerability on UPnP, which can be reached via SOAP requests [25] via TCP port 52869 [26]. To investigate these two vulnerabilities, we examined the extracted strings from the IoT malware binaries that scanned ports 37215/52869 and found clear indications of attack instructions that confirm such attacks towards `Huawei` routers.

6.8 Summary and Concluding Remarks

In this work, we utilized reverse-engineering and static malware analysis techniques to characterize a representative sample of IoT malware and create a better understanding about their interrelations. Our findings indicate that despite the rising number of IoT malware, the majority of them fall under a handful of known families (e.g., Mirai and Gafgyt). Furthermore, while we explore

malware detection timeline and their persistence, we associate a relatively shorter life time to recently detected malware samples, which may indicate that adversaries are in fact using IoT malware as disposable resources. Moreover, while we leverage strings-based analysis to extract useful information from IoT malware binaries, our devised IP-based malware correlation uncovered adversarial infrastructure and resources, which are used to orchestrate malicious campaigns. Consequently, our functional similarity analysis based on the extracted command strings from those malware samples indicate that adversaries are likely to recycle and reuse a limited number of malicious executables to obtain new IoT malware samples whenever necessary.

From a different perspectives, we demonstrate the effectiveness of IoT honeypots towards identifying new IoT malware samples that were undetected by major antivirus vendors. Indeed, we utilize our functional similarity analysis approach towards IoT malware labeling and family attribution, while highlighting clusters of IoT malware samples that are related to new, unknown malware families/variants. Additionally, we leverage an implemented dynamic analysis environment to execute IoT malware binaries to obtain behavioral characteristics and scanning traffic (if any). Indeed, we obtain connection attempts to adversarial IP addresses, which corroborate our findings from the strings-based malware analysis. Moreover, while we investigate malware-generated scanning behaviors in terms of targeted ports and associated vulnerabilities, our analysis in contrast with the underlying functional similarities demonstrate that correlated malware implementations are likely to scan similar destination port sets. Nevertheless, we also highlight cases of possible implementation reuse, where similar commands are found in different malware executables that have variable scanning objectives (targeted ports).

Chapter 7

Conclusion and Future Work

7.1 Conclusion

Internet of Things (IoT) devices has been integrated in different aspects of our daily lives. Indeed, such benefits have been a major driver behind the introduction of new technologies such as the 5G networks, which will support "smart" technologies by enabling the integration of a massive number of IoT devices withing future networks. Despite their benefits, the emergence of IoT-driven cyber attacks, which utilize malware infected IoT devices to orchestrate large-scale distributed malicious activities, has been considered as a major threat to the Internet ecosystem. Indeed, the rising number of such attacks in recent years shed light on the insecurity of the IoT paradigm at scale. More importantly, the projected increase in the number of IoT devices within 5G networks will amplify such threats, which cause a major concern for the security and operations of telecommunication networks.

This thesis was dedicated to tackle the security concerns associated with the operation of malware-infected IoT devices and their unsolicited Internet activities in light of the rapidly emerging IoT-driven cyber attacks. Considering the various challenges specific to the IoT paradigm, we successfully achieved the latter by employing data-driven approaches and leveraging passive data and Internet measurements. In particular, we addressed the lack of empirical data and knowledge about IoT devices and their unsolicited behaviors by proposing a data-driven approach, which utilizes publicly available IoT devices information and passive Internet measurement data collected at a

large network telescope (darknet) to empirically characterize the magnitude of Internet-scale IoT exploitations in both, consumer and critical CPS realms.

Consequently, while we execute a first-of-a-kind, large-scale empirical characterization of IoT-generated activities in the wild, we demonstrate the feasibility of our approach to identify compromised, malware-infected IoT devices that participate in Internet-scale malicious scanning/probing activities. After inferring such malicious scanning activities, we attempted to uncover scanning campaigns initiated by well-coordinated IoT botnets by correlating/clustering IoT devices with similar behavioral characteristics and scanning objectives. Additionally, we tackle the operational challenges associated with the detection of exploited IoT devices and the analysis of their large-scale malicious activities. More specifically, we developed a scalable system for cyber-threat intelligence reporting and analysis that can trigger informed decisions for in-depth forensic investigations in near real-time.

In addition to leveraging passive Internet measurements to identify malware-infected IoT devices and analyze their activities, in this thesis, we focus on building a better understanding about the root cause of the emerging IoT threats by analyzing a large corpus of recently detected IoT malware binaries. Specifically, we characterize known IoT malware samples in terms of their family distribution, detection timeline, and activity duration (threat persistent). Furthermore, we utilize reverse-engineering and static malware analysis techniques to investigate adversarial infrastructure and shared resources used to operate malware-driven malicious activities. Subsequently, we leverage functional similarity analysis to address the problems of unknown malware family labeling and attribution by correlating IoT malware samples into groups with common malicious implementations.

Finally, we implement a dynamic malware analysis testbed to extend our findings by executing malware binaries and extracting behavioral characteristics in terms of connection attempts to adversarial IP addresses and scanning traffic, whenever available. Despite the challenges in performing large-scale dynamic analysis of IoT malware, our findings corroborate and extend knowledge about the behavioral characteristics of IoT malware in terms of the scanned destination ports/services, which shed light on the associated vulnerabilities and IoT-specific threats.

7.2 Future Work

As for future work, we aim at addressing the problems associated with IoT threat detection and mitigation within 5G networks. More specifically, we focus on network slicing as a fundamental part of 5G network architecture, which enables the implementation of flexible and scalable service-oriented network slices on top of a common network infrastructure network by leveraging the benefits of Software-Defined Networking (SDN) for separating control and data plane functions, while utilizing resources virtualization through Network Function Virtualization (NFV).

From a network operator perspective, each network slice can be administrated and managed independently to serve the needs associated with certain applications and/or supporting technologies within the 5G network such as IoT. Moreover, while network slicing can provide some intra-slice security guarantees by separating the resources/operations of different slices, the inter-slice security remains a major concern, especially with the massive deployment of vulnerable IoT devices within the 5G networks. For instance, exploited user-controlled devices (e.g., IoT devices) can be utilized to perform different attacks within a given slice such as unauthorized resource allocation/consumption and Denial of Service (DoS) attacks, unsolicited scanning and reconnaissance activities, malware execution/distribution, and unauthorized access to intra-slices shared resources, to name a few.

To mitigate such threats, there is an utmost need to develop effective Mobile Edge Computing (MEC) tools and techniques for prompt detection and characterization of IoT threats by inferring infected IoT devices and characterizing their unsolicited activities. This is an extremely challenging task due to: (i) the highly dynamic environment with a large number of insecure IoT devices, (ii) the lack of information about exploited IoT devices within the user space, and (iii) the lack of control over user-owned IoT device operations.

To address these challenges, future work will focus on achieving the following main research objectives:

- Given the information collected about IoT devices and their activities, we aim at extending our data-driven methodologies to analyze, integrate, and synthesize such information to identify compromised devices and characterize their unsolicited behaviors within the 5G network.

- Given the distributed nature of MEC, and the extracted behavioral characteristics/features associated with compromised IoT devices, we will leverage federated/collaborative learning approaches to build robust classifiers for enabling effective and efficient threat detection and attribution across the network.

- Given the rapid evolution of IoT-driven cyber attacks, we will leverage information about existing IoT-driven cyber attacks along with the advances in Adversarial Machine Learning (AML) techniques to assure the resilience of the proposed detection/mitigation approach against evasion techniques employed by adversaries.

Bibliography

[1] J. Manuel, R. Joven, and D. Durando. (2018, February) OMG: Mirai-based Bot Turns IoT Devices into Proxy Servers. Retrieved from https://www.fortinet.com/blog/threat-research/omg--mirai-based-bot-turns-iot-devices-into-proxy-servers.

[2] S. Torabi, E. Bou-Harb, C. Assi, M. Galluscio, A. Boukhtouta, and M. Debbabi, "Inferring, Characterizing, and Investigating Internet-Scale Malicious IoT Device Activities: A Network Telescope Perspective," in *Proc. of the 48th Annual IEEE/IFIP Int. Conf. on Dependable Systems and Networks (DSN)*, June 2018, pp. 562–573.

[3] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis *et al.*, "Understanding the Mirai Botnet," in *Proc. of the 26th USENIX Security Symp.*, Vancouver, BC, 2017, pp. 1093–1110.

[4] C. Cimpanu, "Hajime Botnet Makes a Comeback With Massive Scan for MikroTik Routers," Retrieved from https://www.bleepingcomputer.com/news/security/hajime-botnet-makes-a-comeback-with-massive-scan-for-mikrotik-routers/, March 2018.

[5] P.-A. Vervier and Y. Shen, "Before Toasters Rise Up: A View into the Emerging IoT Threat Landscape," in *Int. Symp. on Research in Attacks, Intrusions, and Defenses*. Springer, 2018, pp. 556–576.

[6] Y. M. P. Pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama, and C. Rossow, "IoTPOT: Analysing the Rise of IoT Compromises," in *9th USENIX Workshop on Offensive Technologies (WOOT 15)*, Washington, D.C., 2015.

[7] S. Torabi, E. Bou-Harb, C. Assi, E. B. Karbab, A. Boukhtouta, and M. Debbabi, "Investigating Internet-Scale Reconnaissance Activities by Compromised IoT Devices Through The Lens of A Large-Scale Network Telescope," *IEEE Transactions on Dependable and Secure Computing TDSC (DOI: 10.1109/TDSC.2020.2979183)*, 2020.

[8] M. S. Pour, S. Torabi, E. Bou-Harb, C. Assi, and M. Debbabi, "Stochastic Modeling, Analysis and Investigation of IoT-Generated Internet Scanning Activities," *IEEE Networking Letters*, pp. 1–1, 2020.

[9] S. Torabi, E. Bou-Harb, C. Assi, and M. Debbabi, "A Scalable Platform for Investigating Exploited IoT Devices and Fingerprinting Unsolicited Activities," *Forensic Science International: Digital Investigation (ISNN: 2666-2817)*, 2020.

[10] S. Torabi, E. B. Karbab, E. Bou-Harb, C. Assi, and M. Debbabi, "A Strings-Based Similarity Analysis Approach for Large-Scale IoT Malware Analysis, Characterization, and Family Attribution," *IEEE Transactions on Dependable and Secure Computing TDSC (Submission ID: TDSC-2021-01-0038)*, 2021.

[11] S. Torabi, M. Dib, E. Bou-Harb, C. Assi, and M. Debbabi, "A Strings-Based Similarity Analysis Approach for Characterizing IoT Malware and Inferring Their Underlying Relationships," *IEEE Networking Letters*, 2021.

[12] E. Glatz and X. Dimitropoulos, "Classifying Internet One-way Traffic," in *Proceedings of the 2012 Internet Measurement Conference*, ser. IMC '12, Boston, MA, USA, 2012, pp. 37–50.

[13] C. Fachkha and M. Debbabi, "Darknet as a Source of Cyber Intelligence: Survey, Taxonomy, and Characterization," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1197–1227, 2016.

[14] E. Bou-Harb, C. Assi, and M. Debbabi, "CSC-Detector: A System to Infer Large-Scale Probing Campaigns," *IEEE Transactions on Dependable and Secure Computing*, 2016.

[15] N. Furutani, T. Ban, J. Nakazato, J. Shimamura, J. Kitazono, and S. Ozawa, "Detection of DDoS Backscatter Based on Traffic Features of Darknet TCP Packets," in *Ninth Asia Joint Conference on Information Security (ASIA JCIS)*. IEEE, 2014, pp. 39–43.

[16] E. Bou-Harb, M. Debbabi, and C. Assi, "On Fingerprinting Probing Activities," *Computers & Security*, vol. 43, pp. 35–48, 2014.

[17] M. H. Bhuyan, D. Bhattacharyya, and J. Kalita, "Surveying Port Scans and Their Detection Methodologies," *The Computer Journal*, vol. 54, no. 10, pp. 1565–1581, Oct. 2011. [Online]. Available: http://dx.doi.org/10.1093/comjnl/bxr035

[18] N. Blenn, V. Ghiëtte, and C. Doerr, "Quantifying the Spectrum of Denial-of-Service Attacks Through Internet Backscatter," in *Proc. of the 12th Int. Conf. on Availability, Reliability and Security*, ser. ARES '17, Reggio Calabria, Italy, 2017, pp. 21:1–21:10.

[19] 360Netlab. (2018, February) ADB.Miner: More Information [Blog post]. Retrieved from https://blog.netlab.360.com/adb-miner-more-information-en/.

[20] W. Andy Greenberg, "The reaper IoT botnet has already infected a million networks," Online: https://www.wired.com/story/reaper-iot-botnet-infected-million-networks/, 2017.

[21] FBI, "Foreign Cyber Actors Target Home and Office Routers and Networked Devices Worldwide," https://www.ic3.gov/media/2018/180525.aspx, May 2018, public Service Announcement (Alert Number: I-052518-PSA).

[22] "Midnight Commander," Retrieved from https://midnight-commander.org/, 2021.

[23] SISSDEN. (2018, February) Darknet - Satori strikes again. [Blog post]. Retrieved from https://sissden.eu/blog/darknet-satori-dasan.

[24] T. Spring. (2017, December) Code Used in Zero Day Huawei Router Attack Made Public. [Blog post]. Retrieved from https://threatpost.com/code-used-in-zero-day-huawei-router-attack-made-public/129260/.

[25] D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. F. Nielsen, S. Thatte, and D. Winer, "Soap: Simple object access protocol," Retrieved from https://tools.ietf.org/html/draft-box-http-soap-01., 1999.

[26] J. Ullrich. (2018, August) When Cameras and Routers attack Phones. Spike in CVE-2014-8361 Exploits Against Port 52869. Retrieved from https://isc.sans.edu/diary/rss/23942.

[27] A. Moser, C. Kruegel, and E. Kirda, "Limits of Static Analysis for Malware Detection," in 23^{rd} *Annual Computer Security Applications Conference (ACSAC 2007)*, 2007, pp. 421–430.

[28] O. Or-Meir, N. Nissim, Y. Elovici, and L. Rokach, "Dynamic Malware Analysis in the Modern Era–A State of the Art Survey," *ACM Computing Surveys (CSUR)*, vol. 52, no. 5, pp. 1–48, 2019.

[29] A. Afianian, S. Niksefat, B. Sadeghiyan, and D. Baptiste, "Malware dynamic analysis evasion techniques: A survey," *ACM Computing Surveys (CSUR)*, vol. 52, no. 6, pp. 1–28, 2019.

[30] D. Gibert, C. Mateu, and J. Planes, "HYDRA: A Multimodal Deep Learning Framework for Malware Classification," *Computers & Security*, p. 101873, 2020.

[31] E. B. Karbab and M. Debbabi, "Maldy: Portable, data-driven malware detection using natural language processing and machine learning techniques on behavioral analysis reports," *Digital Investigation*, vol. 28, pp. S77–S87, 2019.

[32] C. Jindal, C. Salls, H. Aghakhani, K. Long, C. Kruegel, and G. Vigna, "Neurlux: Dynamic Malware Analysis Without Feature Engineering," in *Proceedings of the 35th Annual Computer Security Applications Conference*, San Juan, PR, USA, 2019, pp. 444–455.

[33] A. Cui and S. J. Stolfo, "A quantitative analysis of the insecurity of embedded network devices: results of a wide-area scan," in *Proc. of the 26th Annual Comput. Security Applicat. Conf.* ACM, 2010, pp. 97–106.

[34] V. Sachidananda, S. Siboni, A. Shabtai, J. Toh, S. Bhairav, and Y. Elovici, "Let the Cat Out of the Bag: A Holistic Approach Towards Security Analysis of the Internet of Things," in

Proc. of the 3rd ACM Int. Workshop on IoT Privacy, Trust, and Security, ser. IoTPTS '17, 2017, pp. 3–10.

[35] A. Costin, J. Zaddach, A. é. Francillon, D. Balzarotti, and S. Antipolis, "A large-scale analysis of the security of embedded firmwares," in *In 23rd USENIX Security Symp.*, 2014, pp. 95–110.

[36] D. D. Chen, M. Woo, D. Brumley, and M. Egele, "Towards Automated Dynamic Analysis for Linux-based Embedded Firmware," in *Proc. of the Network and Distributed Syst. Security Symp. (NDSS)*, 2016.

[37] T. Yu, V. Sekar, S. Seshan, Y. Agarwal, and C. Xu, "Handling a trillion (unfixable) flaws on a billion devices: Rethinking network security for the Internet-of-Things," in *Proc. of the 14th ACM Workshop on Hot Topics in Networks*. ACM, 2015, p. 5.

[38] Y. J. Jia, Q. A. Chen, S. Wang, A. Rahmati, E. Fernandes, Z. M. Mao, A. Prakash, and S. J. Unviersity, "ContexIoT: Towards Providing Contextual Integrity to Appified IoT Platforms," in *Proc. of the Network and Distributed Syst. Security Symp. (NDSS'17)*, 2017.

[39] E. Fernandes, J. Paupore, A. Rahmati, D. Simionato, M. Conti, and A. Prakash, "FlowFence: Practical Data Protection for Emerging IoT Application Frameworks," in *25th USENIX Security Symp.*, 2016.

[40] B. Ur, J. Jung, and S. Schechter, "The Current State of Access Control for Smart Devices in Homes," in *Workshop on Home Usable Privacy and Security (HUPS)*, 2013.

[41] E. Ronen and A. Shamir, "Extended Functionality Attacks on IoT Devices: The Case of Smart Lights," in *IEEE European Symp. on Security and Privacy (EuroS&P)*. IEEE, 2016, pp. 3–12.

[42] J. D. Guarnizo, A. Tambe, S. S. Bhunia, M. Ochoa, N. O. Tippenhauer, A. Shabtai, and Y. Elovici, "Siphon: Towards Scalable High-Interaction Physical Honeypots," in *Proc. of the 3rd ACM Workshop on Cyber-Physical Syst. Security*. ACM, 2017, pp. 57–68.

[43] T. Luo, Z. Xu, X. Jin, Y. Jia, and X. Ouyang, "IoTCandyJar: Towards an Intelligent-Interaction Honeypot for IoT Devices," in *Blackhat*, 2017.

[44] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, and J. A. Halderman, "A Search Engine Backed by Internet-Wide Scanning," in *22nd ACM Conf. on Computer and Commun. Security*, Oct. 2015.

[45] "Shodan," Retrieved from https://www.shodan.io/, 2019.

[46] X. Feng, Q. Li, H. Wang, and L. Sun, "Acquisitional Rule-based Engine for Discovering Internet-of-Things Devices," in *27th USENIX Security Symp.*, 2018, pp. 327–341.

[47] S. Torabi, A. Boukhtouta, C. Assi, and M. Debbabi, "Detecting Internet Abuse by Analyzing Passive DNS Traffic: A Survey of Implemented Systems," *IEEE Commun. Surveys & Tutorials*, 2018.

[48] M. A. Hakim, H. Aksu, A. S. Uluagac, and K. Akkaya, "U-pot: A honeypot framework for upnp-based iot devices," in 37^{th} *IEEE International Performance Computing and Communications Conference (IPCCC)*. IEEE, 2018, pp. 1–8.

[49] C. Labovitz, A. Ahuja, and M. Bailey, *Shining Light on Dark Address Space*. Arbor Networks Inc., 2001.

[50] C. Fachkha, E. Bou-Harb, A. Keliris, N. Memon, and M. Ahamad, "Internet-scale Probing of CPS: Inference, Characterization and Orchestration Analysis," in *Proc. of the Network and Distributed Syst. Security Symp. (NDSS'17)*, San Diego, California, 2017.

[51] S. Bellovin, "There Be Dragons," in *USENIX Summer*, 1992.

[52] S. M. Bellovin, "Packets Found on an Internet," *ACM SIGCOMM Computer Communication Review*, vol. 23, no. 3, pp. 26–31, 1993.

[53] D. Moore, C. Shannon, D. J. Brown, G. M. Voelker, and S. Savage, "Inferring Internet Denial-of-Service Activity," *ACM Transactions on Computer Systems (TOCS)*, vol. 2, no. 2, pp. 115–139, 2006.

[54] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver, "The Spread of the Sapphire/Slammer Worm," https://www.caida.org/publications/papers/2003/sapphire/sapphire.html, 2003.

[55] M. Bailey, E. Cooke, F. Jahanian, D. Watson, and J. Nazario, "The Blaster Worm: Then and Now," *IEEE Security and Privacy*, vol. 3, no. 4, pp. 26–31, Jul. 2005. [Online]. Available: http://dx.doi.org/10.1109/MSP.2005.106

[56] K. Limthong, F. Kensuke, and P. Watanapongse, "Wavelet-based unwanted traffic time series analysis," in *International Conference on Computer and Electrical Engineering (ICCEE)*. IEEE, 2008, pp. 445–449.

[57] A. Dainotti, A. King, k. Claffy, F. Papale, and A. Pescapè, "Analysis of a "/0" Stealth Scan from a Botnet," in *Proceedings of the 2012 Internet Measurement Conference*, ser. IMC '12, 2012, pp. 1–14. [Online]. Available: http://doi.acm.org/10.1145/2398776.2398778

[58] C. Fachkha, E. Bou-Harb, and M. Debbabi, "Fingerprinting Internet DNS Amplification DDoS Activities," in *the 6th International Conference on New Technologies, Mobility and Security (NTMS)*. IEEE, 2014, pp. 1–5.

[59] C. Rossow, "Amplification Hell: Revisiting Network Protocols for DDoS Abuse," in *Symp. on Network and Distributed System Security (NDSS)*, 2014.

[60] M. S. Pour, A. Mangino, K. Friday, M. Rathbun, E. Bou-Harb, F. Iqbal, S. Samtani, J. Crichigno, and N. Ghani, "On Data-driven Curation, Learning, and Analysis for Inferring Evolving Internet-of-Things (IoT) Botnets in the Wild," *Computers & Security*, p. 101707, 2019.

[61] B. Wang, Y. Dou, Y. Sang, Y. Zhang, and J. Huang, "IoTCMal: Towards A Hybrid IoT Honeypot for Capturing and Analyzing Malware," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–7.

[62] Q.-D. Ngo, H.-T. Nguyen, L.-C. Nguyen, and D.-H. Nguyen, "A survey of iot malware and detection methods based on static features," *ICT Express*, 2020.

[63] E. Cozzi, P.-A. Vervier, M. DellâĂŹAmico, Y. Shen, L. Bilge, and D. Balzarotti, "The Tangled Genealogy of IoT Malware," in *The Annual Computer Security Applications Conference (ACSAC)*, 2020.

[64] H. Alasmary, A. Khormali, A. Anwar, J. Park, J. Choi, A. Abusnaina, A. Awad, D. Nyang, and A. Mohaisen, "Analyzing and detecting emerging internet of things malware: A graph-based approach," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8977–8988, 2019.

[65] M. Alhanahnah, Q. Lin, Q. Yan, N. Zhang, and Z. Chen, "Efficient Signature Generation for Classifying Cross-Architecture IoT Malware," in *IEEE Conference on Communications and Network Security (CNS)*. IEEE, 2018, pp. 1–9.

[66] I. U. Haq and J. Caballero, "A Survey of Binary Code Similarity," *arXiv preprint arXiv:1909.11424*, 2019.

[67] "The CAIDA UCSD Real-time Network Telescope Data," UCSD - Center for Applied Internet Data Analysis. Retrieved from https://www.impactcybertrust.org/, 2018.

[68] K.-D. Kim and P. R. Kumar, "Cyber–Physical Systems: A Perspective at the Centennial," *Proceedings of the IEEE 100*, no. Special Centennial Issue, pp. 1287–1308, 2012.

[69] J. Wurm, K. Hoang, O. Arias, A.-R. Sadeghi, and Y. Jin, "Security Analysis on Consumer and Industrial IoT Devices," in *21st Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 2016, pp. 519–524.

[70] M. McKeay, J. Arteaga, A. Fakhreddine, D. Lewis, L. Cashdollar, C. Seaman, J. Thompson, R. Barnett, and E. Caltum, "The Q4 2016 State of the Internet / Security Report," Akamai Technologies Inc. (Technical Report), 2017.

[71] K. Zetter, "Inside the Cunning, Unprecedented Hack of Ukraine's Power Grid," Online https://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/, Wired, March 2016.

[72] E. MacAskill, S. Thielman, and P. Oltermann. (2017, March) WikiLeaks publishes 'biggest ever leak of secret CIA documents'. Retrieved from https://www.theguardian.com/media/2017/mar/07/wikileaks-publishes-biggest-ever-leak-of-secret-cia-documents-hacking-surveillance. The Guardian.

[73] Z. Durumeric, M. Bailey, and J. A. Halderman, "An Internet-Wide View of Internet-Wide Scanning," in *Proc. of the 23rd USENIX Security Symp.*, San Diego, CA, 2014, pp. 65–78.

[74] E. Balkanli and A. N. Zincir-Heywood, "On the Analysis of Backscatter Traffic," in *Local Computer Networks Workshops (LCN Workshops), 2014 IEEE 39th Conference on*. IEEE, 2014, pp. 671–678.

[75] J. Liu and K. Fukuda, "Towards a Taxonomy of Darknet Traffic," in *International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 2014, pp. 37–43.

[76] T. Yeh, "Netis Routers Leave Wide Open Backdoor," http://blog.trendmicro.com/trendlabs-security-intelligence/netis-routers-leave-wide-open-backdoor/, August 2014.

[77] "Cymon Open Threat Intelligence," https://cymon.io/.

[78] J. Blackford and M. Digdon, "TR-069: CPE WAN Management Protocol," https://www.broadband-forum.org/technical/download/TR-069_Amendment-5.pdf, November 2013.

[79] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2702–2733, 2019.

[80] "Corsaro," Center for Applied Internet Data Analysis (CAIDA), https://www.caida.org/tools/measurement/corsaro/.

[81] "The ELK Stack," Retrieved from https://www.elastic.co/elk-stack, 2019.

[82] K. Thomas, R. Amira, A. Ben-Yoash, O. Folger, A. Hardon, A. Berger, E. Bursztein, and M. Bailey, "The abuse sharing economy: Understanding the limits of threat exchanges," in *Int. Symp. on Research in Attacks, Intrusions, and Defenses*. Springer, 2016, pp. 143–164.

[83] A. Dainotti, A. King, K. Claffy, F. Papale, and A. Pescapé, "Analysis of a/0 Stealth Scan From a Botnet," *IEEE/ACM Transactions on Networking (TON)*, vol. 23, no. 2, pp. 341–354, 2015.

[84] I. A. N. A. (IANA), "Service Name and Transport Protocol Port Number Registry," Retrieved March 1, 2019 from https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml, 2019.

[85] "SX-Virtual Link Software Frequently Asked Questions," Retrieved from https://www.silextechnology.com/sx-virtual-link-faq, 2019.

[86] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in *ACM SIGMOD record*, vol. 22, no. 2. ACM, 1993, pp. 207–216.

[87] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *Proc. of KDD*, vol. 96, no. 34, 1996, pp. 226–231.

[88] G. H. Shah, C. Bhensdadia, and A. P. Ganatra, "An Empirical Evaluation of Density-Based Clustering Techniques," *Int. J. of Soft Comput. and Eng. (IJSCE)*, vol. 22312307, pp. 216–223, 2012.

[89] J. Mazel, P. Casas, R. Fontugne, K. Fukuda, and P. Owezarski, "Hunting attacks in the dark: clustering and correlation analysis for unsupervised anomaly detection," *Int. J. of Network Management*, vol. 25, no. 5, pp. 283–305, 2015.

[90] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "BotMiner: Clustering Analysis of Network Traffic for Protocol-and Structure-Independent Botnet Detection," in *Proc. of the 17th USENIX Security Symp.*, 2008, pp. 139–154.

[91] "Qemu," Retrieved from https://qemu.org/, 2019.

[92] "Rasberry Pi," Retrieved from https://www.raspberrypi.org/, 2019.

[93] O. Kubovic. (2019, May) EternalBlue Reaching New Heights Since WannaCryptor Outbreak. [Blog post]. Retreived from https://www.welivesecurity.com/2019/05/17/eternalblue-new-heights-wannacryptor/. WeLiveSecurity.

[94] L. Fengpei. (2017, April) New Threat Report: A new IoT Botnet is Spreading over HTTP 81 on a Large Scale. Retrieved from http://blog.netlab.360.com/a-new-threat-an-iot-botnet-scanning-internet-on-port-81-en/.

[95] (2011, December) Identifying and Mitigating Exploitation of the Cisco Unified Communications Manager Express and Cisco IOS Software H.323 and SIP DoS Vulnerabilities. Retrieved from https://www.cisco.com/c/en/us/support/docs/cmb/cisco-amb-20100324-voice.html. Cisco.

[96] G. Escueta. (2017, February) Mirai Widens Distribution with New Trojan that Scans More Ports. Retrieved from https://blog.trendmicro.com/trendlabs-security-intelligence/mirai-widens-distribution-new-trojan-scans-ports/. Trend Micro.

[97] A. K. Marnerides and A. U. Mauthe, "Analysis and Characterisation of Botnet Scan Traffic," in *Computing, Networking and Communications (ICNC), 2016 International Conference on*. IEEE, 2016, pp. 1–7.

[98] Z. Li, A. Goyal, Y. Chen, and V. Paxson, "Automating Analysis of Large-scale Botnet Probing Events," in *Proc. of the 4th Int. Symp. on Information, Comput., and Commun. Security*, ser. ASIACCS 09, 2009, pp. 11–22.

[99] A. Dvoretzky, J. Kiefer, J. Wolfowitz *et al.*, "Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator," *The Annals of Mathematical Statistics*, vol. 27, no. 3, pp. 642–669, 1956.

[100] R. Verde, A. Irpino, and A. Balzanella, "Dimension Reduction Techniques for Distributional Symbolic Data," *IEEE Transactions on Cybernetics*, vol. 46, no. 2, pp. 344–355, Feb 2016.

[101] A. Irpino, *HistDAWass: Histogram-Valued Data Analysis*, 2020, R package version 1.0.4.

[102] L. McInnes, J. Healy, and S. Astels, "HDBSCAN: Hierarchical density based clustering," *J. Open Source Software*, vol. 2, no. 11, p. 205, 2017.

[103] M. S. P. et al., "Data-driven Curation, Learning and Analysis for Inferring Evolving IoT Botnets in the Wild," in *Proc. of the 14th Int. Conf. on Availability, Reliability and Security (ARES 2019)*, August 2019, pp. 1–10.

[104] S. Herwig, K. Harvey, G. Hughey, R. Roberts, and D. Levin, "Measurement and Analysis of Hajime, a Peer-to-peer IoT Botnet," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2019.

[105] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver, "Inside the Slammer Worm," *IEEE Security & Privacy*, no. 4, pp. 33–39, 2003.

[106] (2019) Spark DataFrame API. Retrieved from https://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark.sql.DataFrame.

[107] C. Seaman. (2018, November) UPNPROXY: ETERNALSILENCE. Retrieved from https://blogs.akamai.com/sitr/2018/11/upnproxy-eternalsilence.html.

[108] M. Sebastián, R. Rivera, P. Kotzias, and J. Caballero, "AVClass: A Tool for Massive Malware Labeling," in *International Symposium on Research in Attacks, Intrusions, and Defenses*. Springer, 2016, pp. 230–253.

[109] UPX Team, "UPX." [Online]. Available: https://upx.github.io/

[110] Hex-Rays, "IDA Pro." [Online]. Available: https://www.hex-rays.com/products/ida/

[111] T. Joachims, "A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization," Carnegie-Mellon University, Pittsburgh, PA, Dept. of Computer Science, Tech. Rep., 1996.

[112] C. D. Manning, H. Schütze, and P. Raghavan, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[113] H. Griffioen and C. Doerr, "Examining Mirai's Battle over the Internet of Things," in *Proc. of the 27'th ACM SIGSAC Conf. on Computer and Communications Security*, ser. CCS '20, 2020, pp. 743–756.

[114] S. P. Borgatti, M. G. Everett, and J. C. Johnson, *Analyzing Social Networks*. Sage, 2018.

[115] A. Darki and M. Faloutsos, "RIoTMAN: A Systematic Analysis of IoT Malware Behavior," in *Proc. of the 16th Int. Conf. on emerging Networking EXperiments and Technologies (CoNEXT)*, 2020, pp. 169–182.