

Malware Threat on Edge/Fog Computing Environments From Internet of Things Devices Perspective

Publisher: IEEE

[Cite This](#)[PDF](#)Ibrahim Gultas ; H. Hakan Kilinc ; A. Halim Zaim; M. Ali Aydin [All Authors](#)19
Cites in
Papers2231
Full
Text Views[Open Access](#) [Comment\(s\)](#)Under a [Creative Commons License](#)

Abstract

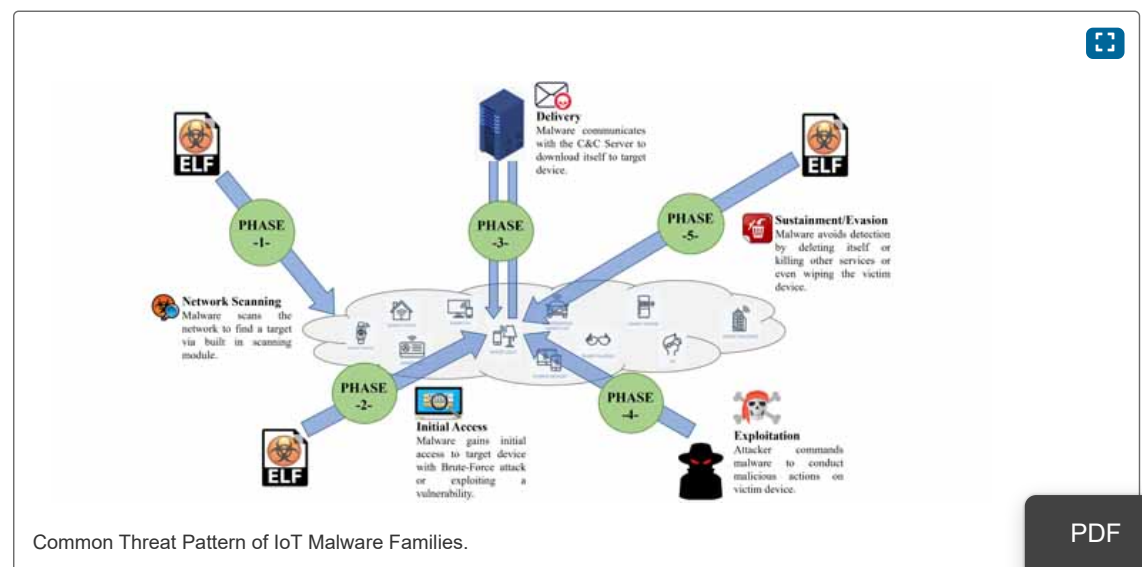
Document Sections

- I. Introduction
- II. Related Work
- III. Methodology
- IV. Malware Families Affecting IoT Devices
- V. Evolution of Malware Families

[Show Full Outline](#)[Authors](#)[Figures](#)[References](#)[Citations](#)[Keywords](#)[Metrics](#)[More Like This](#)

Abstract:

Developing a secure information processing environment highly depends on securing all the layers and devices in the environment. Edge/Fog computing environments are no exception in this case, and the security of these environments highly depends on securing Internet of Things (IoT) devices which are the most vulnerable devices throughout the environment. The adoption of Edge/Fog computing paradigms by new emerging technologies has stimulated malware development for IoT platforms. Recent attacks initiated by IoT malware show that these attacks have a tremendous impact on compromised systems in terms of the Quality of Service because of the number of infected IoT devices. In the light of these developments, there is an enormous need for efficient solutions. However, defense capabilities against these new malware types are highly constrained by the limited understanding of these new emerging paradigms and the lack of access to malware samples. This study mainly focuses on IoT malware to understand the behaviors of malware in the most vulnerable layer of the Edge/Fog computing environments. Mainly, 64 IoT malware families are identified from 2008 when the first known IoT malware emerged to October 2022. These malware families are systematically characterized by various aspects, including target architecture, target device, delivery methods, attack vectors, persistence techniques, and their evolution from existing malware. During this characterization process, two different investigation frameworks, "Cyber Kill Chain" and "Mitre ATT&CK for ICS," have been adopted in the different investigation layers. This paper aims to bring light to future researches with the presented features of the IoT malware.

Published in: [IEEE Access](#) (Volume: 11)

Page(s): 33584 - 33606

DOI: [10.1109/ACCESS.2023.3262614](https://doi.org/10.1109/ACCESS.2023.3262614)

SECTION I.

Introduction

The number of connected devices is increasing with colossal velocity. It is estimated that Internet of Things(IoT) devices will account for 50 percent (14.7 billion) of all networked devices by 2023 [1]. This increase in ubiquitous devices also brings new problem areas in terms of bandwidth consumption and latency between these devices and cloud servers. Edge and Fog Computing paradigms have emerged as a solution to this bottleneck [2]. The heterogeneous and distributed structures of these paradigms reveal some security leaks. As seen in previous newly emerging technologies, devices inside these environments are becoming a new target for malicious activities.

Moreover, Edge/Fog computing environments inherit all the security vulnerabilities of previous technologies, which enable these environments to evolve, such as wireless sensor networks (WSNs), distributed P2P (peer-to-peer) systems, and virtualization platforms. Moreover, these computing environments take advantage of their layered structures to provide a solution to the bottlenecks in the internet infrastructure; however, each layer of these structures has novel vulnerabilities. In this regard, one of the leading security flows is the attacks initiated by malware. Because of the heterogeneous architecture of the Edge/Fog Computing Environment, these systems are targets for malware developed for various architectures.

Edge devices are taking a big part with a huge number of areas of utilization in these new computing ecosystems. We believe that the constrained nature of edge devices makes them the weakest point in the security chain of the Edge/Fog Computing Environment. Only one exploit in one device may pose a threat to all the devices in the same environment. Recent reports have shown that IoT devices have become one of the most popular targets among malicious people in recent years. According to the Cyber Threat Report of Sonicwall, 57 million IoT malware attacks were detected in the first six months of 2022 which indicates a 77 percent increase from the first six months of the previous year [3]. This 6-months attack volume is higher than the yearly attacks recorded in each of 2018, 2019, and 2020. The number of IoT malware families may be limited for now compared to malware that infects conventional devices. However, the vast increase in IoT devices will also lead to an increase in the number of malicious software.

The recent attacks initiated by infected IoT devices have revealed the importance of the situation. The infamous Mirai attack is one of the most remarkable examples of IoT botnets [4]. IoT devices infected by Mirai malware are used in botnets to initiate one of the biggest Distributed Denial of Service (DDoS) attacks in the history. The volume of the generated data traffic is a new record in information technology. During the peak of the attack, 600 Gbps data traffic is generated by the least powerful devices [5].

Unfortunately, Mirai is not the last example of IoT botnets. Quite the contrary, it was just the beginning of the era of IoT botnets. In this research, we located 21 different malware families from 2008, when the first known IoT malware “Hydra” was firstly seen in the wild to the September 2016 “First Seen In The Wild” date of Mirai malware. On the other hand, 43 different malware families are located from Mirai to October 2022. In addition, 29 of these 43 malware families are variants of Mirai malware, which indicates that these malware families inherited some features of Mirai malware. This huge increase in the IoT malware types is mainly caused by the leakage of the Mirai source code on publicly available code repositories. For this reason, the IoT malware evolution process can be classified as “Before Mirai” and “After Mirai”.

Even though there are some invaluable studies on the IoT Malware domain, none of them reflect the entire attack scope of IoT Malware. This study aims to reflect the state-of-the-art IoT malware threat by utilizing research papers, technical reports of antivirus (AV) vendors, and blog posts of malware researchers. In the scope of this research, research papers, technical reports of AV vendors, and blog posts of malware researchers related to IoT malware are collected for the period between 2008 and October 2022. During our research, we assessed that the efforts of AV vendors are far more informative than academia. AV vendors and some malware researchers publish their analysis results in a timely manner to warn their subscribers about

recent threats in the IoT malware domain. For this reason, this research provides summarized information on all those reports in a framework that will be described hereinafter.

The main motivation behind this research is to introduce IoT Malware Families along with their behaviors and features to bring light for further research, which aims to contribute to the security of Edge/Fog Computing Environments by securing IoT Devices. The main contributions of this research are summarized in three stages.

- 64 different malware families that affect IoT devices are located. This comprehensive collection reflects the majority of IoT malware if it is not all. Besides, most of these malware families are becoming a topic in academia for the first time.
- A novel Phylogenetic Tree of IoT Malware is presented in the scope of this research to gain an understanding of the evolution of IoT Malware.
- Behavioral analysis of the located malware families is presented in a three-layered approach. In this study, The Cyber Kill Chain Framework and Mitre ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) for the ICS (Industrial Control Systems) matrix are used collaboratively to present the behaviors of IoT malware for the first time in the academia.

As previously stated, the main goals and contributions of this study can be classified into three stages. First, this study fulfills the need for presenting an extensive collection of IoT malware families. Within this research, 64 different malware families are located from 2008 to October 2022. We believe that this collection reflects the majority, if it is not all, of the known IoT malware families by the date of the published time of this paper. Most of these malware families are covered in an academic research for the first time. Botnet attacks are the most common attack type among the IoT malware families. Infected IoT devices are used in a botnet to initiate DDoS attacks. However, the attack types are not only limited to creating botnets but also mining cryptocurrencies, and DNS Poisoning attacks are in scope.

This research's second goal and contribution is to reveal the evolutionary progress of the IoT malware families. As a state-of-the-art, Phylogenetic Tree of IoT Malware is built up. Four different malware families Hydra, Tsunami, Gafgyt, and Mirai, are determined as parent malware families and highly inherited by other malware families.

Last but not least, the third goal and contribution of this research is presenting a behavioral analysis of the detected malware families. We analyzed the behaviors of the IoT malware in a three-layered approach. The Cyber Kill Chain framework is adopted as a high-level investigation framework. The Mitre ATT&CK framework which is becoming an industry standard knowledge base for presenting adversarial tactics and techniques for different platforms was adopted as a mid-level framework. The Mitre framework is not only becoming an industry standard but also academia started to apply this framework to researches. Gittins and Soltys utilize the Mitre framework to present only Persistence tactics and techniques used by malware [6]. Al-Shaer et al. developed a tool called as Cyber Threat Dictionary which offers solutions for the threats by mapping Mitre ATT&CK framework to the NIST Cyber Security Framework [7]. In this study, the Mitre Framework is used for presenting the behavioral analysis of IoT malware for the first time in academia. In addition, this research presents the most comprehensive utilization of the Mitre ATT&CK framework for IoT malware with covering all of the attack phases of the Cyber Kill Chain Framework. Finally, as a low-level investigation, the attack vectors, communication ports, and protocols are presented. The low-level behavioral analysis shows that brute force attacks are the most common attack vector for gaining initial access to the victim devices. The most common attack vectors to compromise target devices are flooding attacks used for DDoS. Except for the five P2P communicating malware, all the rest of the malware communicates with a C2 (Command and Control) server to obtain the attack commands.

The remainder of this paper is organized as follows. The next section is about previously conducted studies by other researchers. The Methodology of this research is explained in [Section III](#). Malware types affecting IoT devices are introduced in [Section IV](#). The evolution of IoT malware types is summarized in [Section V](#). Behavioral Analysis of Detected Malware Families is given in [Section VI](#). Finally, in [Section VII](#), open issues are highlighted, and new directions for the research community to help them develop more precise detection mechanisms have been proposed.

SECTION II.

Related Work

The previous studies conducted in the IoT malware domain are presented in three subcategories as surveys and general informative research papers, IoT malware datasets, and IoT malware analysis and detection researches.

A. IoT Malware Surveys

Wang et al. categorized IoT malware only according to infection techniques and analyzed only three families as an example [8]. Vignau et al. present features of the most commonly seen IoT malware families in the first 10 years of the IoT Malware History and they present a Phylogenic Graph which contains 16 malware families [9]. In that research, they focus on attack vectors and feature propagation between the malware families. Two years later, Vignau et al. presented another survey research that investigates IoT malware between 2008 and 2019 [10]. They again presented feature propagation between the malware families but this time their research was not only limited to attack vectors; they also added other features of the malware such as infection method, C2 communication, and persistence methods, etc. Their research contains 28 different IoT malware families. Even though, they state that their malware collection reflects all active botnets between 2008 and 2019, they only covered 28 malware families, while we are able to locate 42 different malware families for the same period. Alrawi et al. compared IoT malware with desktop and mobile malware. They presented general insights about some features of IoT malware such as infection methods, attack vectors, and persistence techniques, but their work did not contain specifically adopted techniques by each malware family [11].

During our research on previously conducted survey studies on IoT Malware, we found that these studies provide a general perspective of IoT malware behaviors. These studies cover only a limited number of malware families. Besides, they present only a few characteristics of the malware families, and there is no behavioral information regarding most IoT malware families. In our study, we aim to present detailed information on our malware collection, and to the best of our knowledge, our study covers all of the IoT malware families since 2008. A comparison of the survey studies on IoT malware is presented in Table 1.

TABLE 1 Comparison of IoT Malware Surveys

Research	Covered Number of Malware Families	Malware Lineage	Target Architecture	Target Devices	Infection Techniques	Exploited Vulnerabilities	C2 Communications	Attack Vectors	Persistence and Anti-analysis Techniques
Wang et al. (2017)	3	-	-	-	+	-	-	-	-
Vignau et al. (2019)	16	+	-	-	-	-	-	+	-
Vignau et al. (2021)	28	-	+	-	+	+	+	+	+
Alrawi et al. (2021)	16	-	-	-	+	+	+	+	+
This Research	64	+	+	+	+	+	+	+	+

B. IoT Malware Datasets

The rampant growth of IoT malware, as hereinbefore stated, shows that effective mitigation methods or defense mechanisms are urgently needed. The key to protecting systems effectively highly depends on understanding the attacks. However, the lack of a comprehensive IoT malware dataset makes it almost impossible for researchers to understand IoT malware attacks. There are only a few publicly available IoT malware datasets. One of the publicly available datasets created for IoT malware was presented by Pa et al. They deployed a honeypot called IoTPot to collect active IoT malware [12]. They have published their findings and collected malware binaries with a dataset that contains 86.496 malware binary files. Another dataset was presented by Azmoodeh et al. that contains only 128 malware samples of ARM-based IoT applications between February 2015, and January 2017 [13]. Also, capturing network traffic that belongs to compromised IoT devices is an applicable method for creating datasets. Meidan et al. presented the N-BaIoT dataset [14]. Their dataset was created by injecting Mirai and Bashlite malware into nine commercial IoT devices which are very few in number to present the general of the IoT malware cluster. The dataset contains 7062606 instances of network traffic data. Garcia et al. presented the IoT-23 dataset that includes network traffic from 20 malware families, which are again very few to present the general of the IoT malware cluster [15]. Pour et al. also present a highly comprehensive dataset that contains 3.6TB of network traffic created by 440.000 compromised IoT devices [16]. Trajanovski and Zhang presented one of the most recent and most comprehensive IoT malware dataset [17]. They proposed an IoT Botnet Detection and Analysis (IoT-BDA) framework which combines a honeypot and a sandbox. During 7 months period, their framework collected 4077 malware samples and they presented the analysis results with a dataset that covers static analysis features, dynamic analysis (System Calls and Network Traffic Capture), and malware sample files (ELF

Binaries). Nevertheless, their static analysis only covers binary analysis results such as symbols and linking information but lacks OPCODE sequences which is the main output of a static analysis. Another popular technique among researchers is creating their own datasets for their research. This method is highly effective in evaluating the proposed methods; however, the lack of a commonly adopted dataset prevents comparing the evaluation metrics of the proposed methods. Botacin et al. presented challenges and pitfalls in the malware research area and highlight the usage of non-publicly available datasets limits reproducibility and prevents comparison of evaluation metrics [18]. The comparison of IoT Malware datasets is presented in Table 2.

TABLE 2 Comparison of IoT Malware Datasets

Research	Records	Binary Files	OPCODE	Symbols	Network Traffic	System Calls
Pa et al.- IoTPoT (2015)	86.496	+	-	-	-	-
Azmoodeh et al. IoT Dataset (2018)	128	-	+	-	-	-
Meidan et al - N-BaloT (2018)	7.062.606	-	-	-	+	-
Garcia et al.- IoT-23 (2020)	23	-	-	-	+	-
Safai Pour et al. (2020)	440.000	-	-	-	+	-
Trajanovski and Ning (2021)	4.077	+	-	+	+	+

C. IoT Malware Detection

IoT malware detection is a relatively new research area compared to classical PCs and mobile devices. However, as it is mentioned in the previous sections, the impact of IoT malware should not be regarded as small, quite the contrary, the Mirai example shows that it can be more dangerous than classical malware. Even though there is still a long way to go to secure IoT devices against malware attacks, there have been successful efforts for detecting IoT malware. Most of the IoT malware detection studies are based on static analysis. Haddadpajouh et al. used their own OPCODE based dataset called as IoT Dataset with Recurrent Neural Network (RNN) and achieved an accuracy rate of 98,18% [19]. Su et al used the IoTPoT dataset and converted the malware binary files into gray-scale images [20]. They used Convolutional Neural Network (CNN) algorithm for classification and achieved an accuracy rate of 94%. Alasmay et al. created their own dataset and compared the characteristics of IoT malware and Android malware which are also mainly based on Linux by utilizing Control Flow Graphs [21]. Also, they proposed a detection mechanism based on CNN which achieves a very high accuracy rate of 99.66%. Dovom et al. used the IoT Dataset and achieved an average accuracy rate of 96.41% with Fuzzy Pattern Tree algorithm [22]. Darabian et al. used the IoT Dataset with Decision Tree, KNN, Random-Forest, Multi-Layer Perceptron (MLP), Support Vector Machine (SVM), and AdaBoost algorithms and they achieved the highest accuracy rate of 99,80% with the Decision Tree classifier [23]. Vasan et al. presented a Cross-Architecture IoT Malware Detection method called as MTHAEL [24]. They built their own dataset and their method is based on Neural Network Advanced Ensemble Learning combining RNN and CNN with a very high accuracy rate of 99.98%. Another research for securing the Edge Computing ecosystem against IoT malware is conducted by Haddadpajouh et al. [25]. They used their previous dataset called as the IoT Dataset and proposed a malware detection mechanism for a cloud-edge gateway layer based on SVM by utilizing Grey Wolf Optimization (GWO) for feature selection with a very high accuracy rate of 99.72%.

On the other hand, there have been some efforts based on dynamic analysis, and most of them are based on the investigation of network traffic. Meidan et al. used their own dataset N-BaIoT with Artificial Neural Network (ANN) and achieved TPR rate of 100%, and FPR rate of 1% [14]. Jeon et al. conducted a dynamic analysis for malware detection based on Convolutional Neural Network (CNN) [26]. They used their own dataset and convert the behavioral data of IoT malware into images to analyze and detect malware, and with this effort, they reach a high accuracy rate of 99.28%. Rey et al. used the N-BaIoT dataset to detect IoT Malware [27]. They applied Federated Learning (FL) for two different models as supervised and unsupervised models, for the supervised model they applied MLP which obtains an accuracy rate of 99.38% and for the unsupervised model, they applied ANN which obtains a TPR rate of 99.98%, and TNR rate of 91.78%. The comparison of IoT Malware Detection Research is presented in Table 3.

TABLE 3 Comparison of IoT Malware Detection Research

Research	Dataset	Analysis Technique	Feature	Algorithm	Performance
Haddadpajouh et al. (2018)	IoT Dataset	Static	OPCODE	RNN	Accuracy:98,18
Su et al. (2018)	IoTPoT	Static	Malware Binary	CNN	Accuracy:94
Meidan et al. (2018)	N-BalIoT	Dynamic	Network Traffic	ANN	TPR:100 FPR:0,01
Alasmay et al. (2019)	Own Dataset	Static	OPCODE	CNN	Accuracy:99,66
Dovom et al. (2019)	IoT Dataset	Static	OPCODE	Fuzzy Pattern Tree	Accuracy:96,41
Darabian et al. (2020)	IoT Dataset	Static	OPCODE	Decision Tree	Accuracy:99,80
Vasan et al. (2020)	Own Dataset	Static	OPCODE	RNN+CNN	Accuracy:99,98
Jeon et al. (2020)	Own Dataset	Dynamic	Network Traffic +System call	CNN	Accuracy:99,28
Haddadpajouh et al. (2021)	IoT Dataset	Static	OPCODE	SVM+GWO	Accuracy:99,72
Rey et al. (2022)	N-BalIoT	Dynamic	Network Traffic	FL (MLP and ANN)	MLP Accuracy: 99,38 ANN TPR:99,98 TNR:91,78

SECTION III. Methodology

This study follows a systematic methodology to provide a comprehensive approach for revealing malware threats on the edge devices layer of edge/fog computing environments. In this regard, firstly to identify the names of IoT malware families, exhaustive research has been conducted on academic papers, malware databases such as VirusTotal and MalwareBazaar, malware threat alarms of AV vendors, and OSINT(Open Source Intelligence) sources such as social media platforms and forums. Secondly, malware analysis reports of the identified malware families are collected from academic papers, technical reports of AV vendors, and blog posts of malware researchers. Besides, the source code of some of these malware families are publicly available on the code repositories such as GitHub. These source codes are also added to our collection. As a result of this step, we collected 470 academic studies and industrial reports which were published between 2008 and October 2022. Finally, these scientific papers and industrial reports are investigated manually. During this investigation, the scientific papers and industrial reports which does not contain information related to the contribution of this research are excluded. Moreover, the industrial reports may raise an issue regarding the validity of the provided information. For this reason, reports which are only generated by reverse engineering efforts and supported by screenshots of the conducted steps are taken into consideration while the others are omitted from the collection. As a result of these efforts, our collection contains 91 records containing eight source codes, 13 scientific papers, and 70 industrial reports. The general overview of our methodology is shown in Fig. 1.

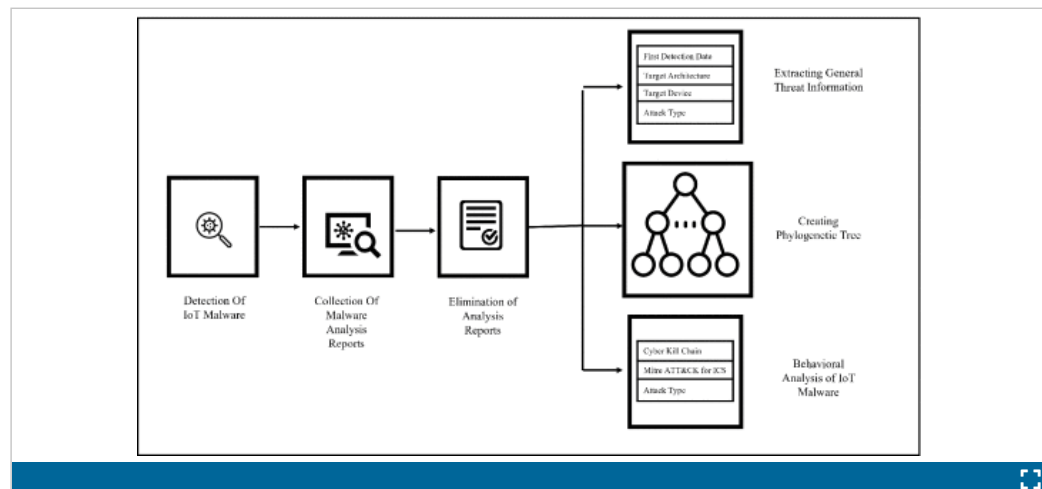


FIGURE 1.
General overview of methodology.

The scientific papers, industrial reports, and source codes within the collection are reviewed to achieve the goals of this research. Firstly, 64 IoT malware families are located and investigated in terms of “First Detection Date,” “Target Architecture,” “Target Device” and “Attack Type.” The “First Detection Date” is determined by comparing the “First Seen In The Wild” value of VirusTotal and the detection date mentioned in the relevant report. According to this comparison, the earliest date is assessed as the “First Detection Date” because sometimes the researchers may submit malware samples after their malware analysis process. The “Target Architecture” is determined by combining the findings of the relevant reports and the “ELF Header

Information” section of the VirusTotal. The “Target Device” and “Attack Type” information are extracted from the findings of the relevant reports. As a result, general information about the malware families which affects IoT devices is presented in [Section IV](#).

Secondly, the Phylogenetic Tree of IoT Malware is generated to provide an understanding about the evolvement of IoT Malware. Computer Virus Phylogenies was first introduced by Goldberg et al. in 1998 by inspiring biological species evolution. They define Malware Phylogenies as the “evolutionary history of computer viruses” [28]. Besides, Allix et al. showed that most of the malware families are generated by slight modifications in the source code of existing malware [29]. From this point of view, the inherited features between the malware families are collected from scientific papers, industrial reports, and source codes of the malware. The results of these efforts are presented in [Section V](#).

Lastly, the third goal of this research is to illustrate the behavioral analysis of malware families by adopting an investigation framework from the high level to the low level. The ICS Cyber Kill Chain framework of the SANS organization has been adopted for high-level investigation of IoT Malware [30]. This framework contains five main and seven sub-phases.

For the mid-level investigation, Mitre ATT&CK for ICS framework has been adopted [31]. This framework is a knowledge base that contains malicious TTP (tactics, techniques, and procedures). Mitre ATT&CK has three different investigation matrices for enterprise devices, mobile devices, and ICS systems. The enterprise and mobile device matrices are also divided into subcategories in terms of the target operating system. Unfortunately, there is no matrix for IoT devices and there are two different approaches for investigating IoT malware with Mitre’s frameworks. Since all of the located malware families for IoT devices based on Linux, enterprise framework for Linux is adopted by some AV vendors. However, we believe that this framework is more applicable to enterprise devices. In this research, ATT&CK for ICS has been adopted because ICS devices are more similar to target devices of located malware families. ATT&CK for ICS framework has a comprehensive matrix to point out adversarial behaviors on the target device. This matrix consists of 11 tactics and 96 techniques. On the matrix, tactics represent the tactical objective of malicious activity while techniques represent the performed action to reach that tactical objection. For example, the adversaries’ tactical objective is to manipulate, interrupt, or destroy the target system, defined as the “Impact” tactic in the ATT&CK matrix. To achieve this tactical objective, the attacker could benefit from one or more of the 11 techniques under this tactic.

For low-level investigation attack vectors, exploited vulnerabilities, communication ports, and services are presented in a systematic manner. As a result of this three-layered investigation, a novel behavioral analysis framework is generated. The first layer covers the ICS Cyber Kill Chain Framework. In the second layer, the tactics and techniques of each phase in the ICS Cyber Kill Chain Framework are presented. In the third layer, attack vectors are highlighted as a part of the technique. The overview of our behavioral analysis framework is shown in [Fig. 2](#) and the results of the behavioral analysis are presented in [Section VI](#).

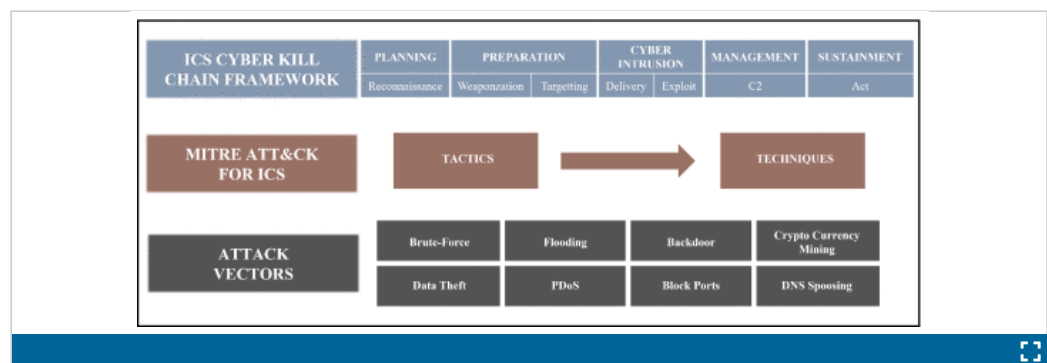


FIGURE 2.
Behavioral analysis framework.

SECTION IV. Malware Families Affecting IoT Devices

The Hydra malware, the first known malware affecting IoT systems, started a new era for malicious code developers. After the source code of this malware was released publicly on the code repositories [32], new malware families evolved based on this source code. Since 2008, the number of IoT malware families increased with colossal velocity. The first goal of this research is to present the majority, if it is not all, of the IoT malware families to provide a deep understanding of the behaviors of these malware families for the research community to develop more precise defense mechanisms. As a result of our efforts, 64 different malware families are identified. By the date of the published time of this research, we believe this collection reflects the state of the art of IoT malware. Brief information about these malware families is provided in Table 4.

TABLE 4 Summary of IoT Malware[illegible]

IoT malware is generally used for creating botnets to initiate DDoS attacks. The 51 (80%) malware families are used for creating botnets. However, the attack types are not only limited to creating botnets but also include cryptocurrency mining, DNS Poisoning, Permanent Denial of Service (PDOS), and Data Exfiltration attacks are in scope. Besides, 12 (19%) of the malware use more than one attack type to compromise the target. Another dramatic statistic is that 13 (20%) of malware families (Carna, Linux.Darloz, Linux.Wifatch, Brickerbot, PNScan, Moose, Linux.MulDrop, VPNFilter, LiquorBot, Silex, Sora, ZuoRAT, and Shikitega) which represents 20 percent of our collection, are not used for DDoS attacks. Moreover, there are two White-Hat Trojans in this collection (Carna and Linux.Wifatch). These White-Hat Trojans were developed by security professionals to emphasize the security flows of IoT devices.

Most of the malware families target more than one CPU architecture, and they have different binary files for each of the target architectures. This diversity in the target creates a new challenge for signature-based detection approaches. Because the binary executables of the malware change according to the target CPU architecture, there are different signatures for the same malware family. Besides, most of the IoT malware detection methods are based on machine learning models. As it is discussed in [Section II](#) most of these efforts are based on OPCODEs which are obtained by static analysis of malware. Since IoT devices have diversity in terms of CPU architecture, the OPCODEs also vary according to the executed CPU architecture. This situation maintains a significant challenge in building an effective malware detection method based on OPCODE. Developing an anti-malware tool based on machine learning techniques requires separate training processes for each of the target CPU architectures.

ARM and MIPS are the most commonly used target architectures for IoT malware. There are 49 malware families which are targeting ARM, while this number is 48 for MIPS-based devices. Intel architectures are also on target with 41 malware families. PowerPC, Motorola, and Sparc architectures are susceptible to malware as well. Besides, some malware families like LuaBot and BrickerBot target only specific architectures to attack. The distribution of the target architecture is shown in Fig. 3.

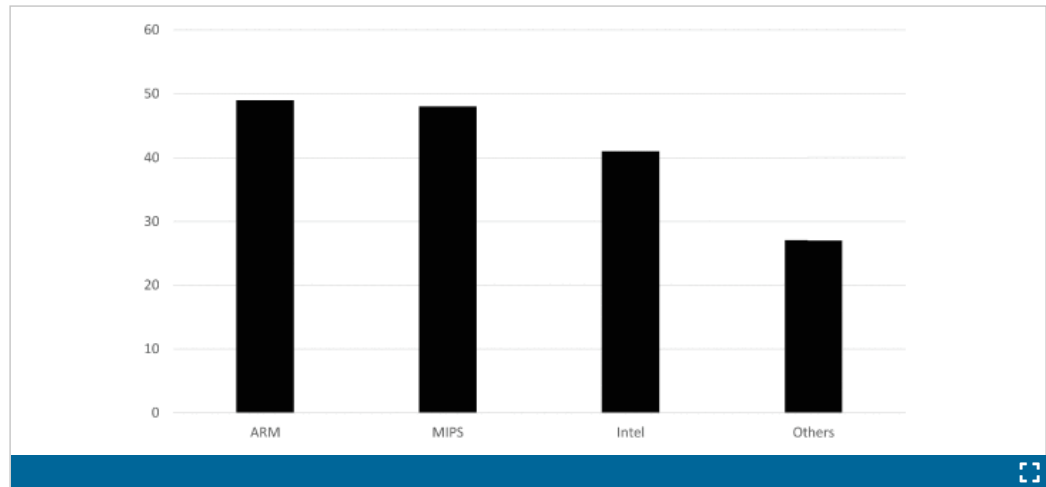


FIGURE 3.
Target architecture distribution.

Network devices such as routers are the leading target devices for IoT malware. The list of target devices is given in Table 4; however, we believe that the target devices are not limited to these devices. IP cameras, DVRs (Digital Video Recorders), toys, CCTV cameras, and TV receivers are infected by IoT malware as well. We assess that IoT malware can infect any device if they have relevant vulnerabilities.

SECTION V. Evolution of Malware Families

As it is mentioned hereinbefore, Hydra is the first known malware family for edge devices. It is not surprising to see the development of other malware families after the public release of the source code of the Hydra malware. After 2014, the rate of increase in IoT malware increased as IoT devices getting popular. After the infamous Mirai attack in 2016, the IoT malware threat takes the attention of all information security sector. Fig. 4 shows the yearly distribution of IoT Malware development to denote the IoT malware growth better.

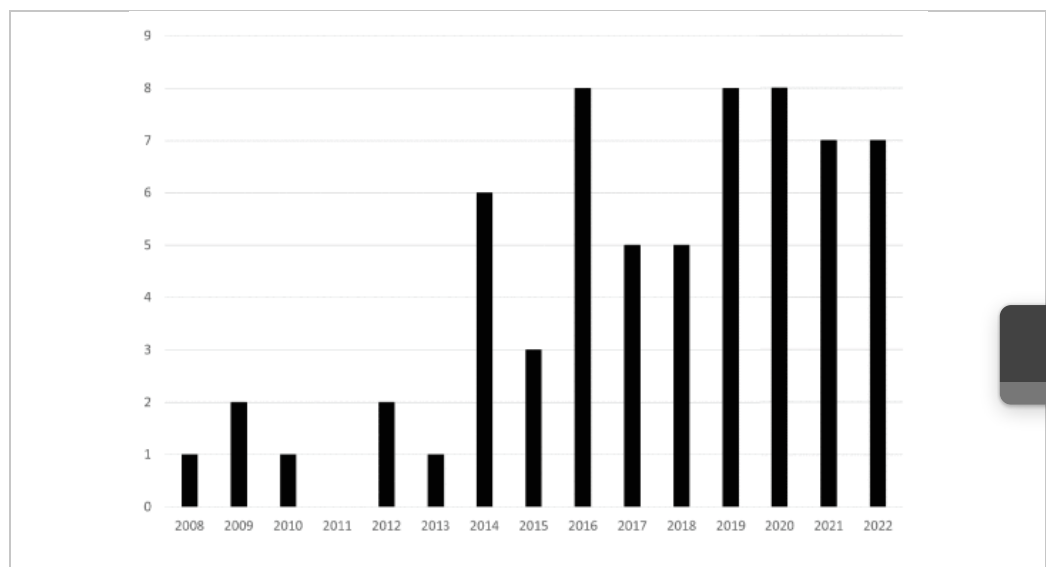


FIGURE 4.

Yearly distribution of IoT malware development.

Some of the previous studies provide insight into the evolution of IoT malware [9], [125], [126], [127], [128]. Angrishi presents 14 malware families from Hydra to IRCTelnet on a timeline [125]. On the other hand, Donno et al. also present 13 malware families from Hydra to Mirai on a timeline while showing the relationship between the malware families [126]. Vignau et al. present features of the most commonly seen IoT malware families on the first 10 years of the IoT Malware History and they present a Phylogenetic Graph which contains 16 malware families [9]. One of the latest research in this field performed by Ngo et al., the relationship between 12 malware families from Hydra to VPNFilter is presented on their research [127]. Researches [125] and [126] analyzed 13 and 12 malware families from 2008 to 2016 timeframe; in our research, we located 24 malware families for the same timeframe. We believe that the lack of resources causes this difference in the malware family numbers for the newly emerging malware families. Besides, Cozzi et al. presented a lineage of IoT malware families [128]. They benefit from machine learning techniques to find the similarities between the source codes of malware families. Their results were awe-inspiring; most of the malware families in their dataset share common functions. Kim et al. proposed a malware classification method based on “Recursive Feature Elimination (RFE)” [129]. Even though they applied their proposed method only to windows-based malware, we believe that their methodology could also be used to classify IoT malware.

When [Tables 4](#) and [7](#) are investigated deeply, it provides insight into the evolution of malware families. For example, from Hydra to Tsunami (2008-2010), mostly MIPS architecture was targeted. After Aidra, other architectures have started to become targets for malware families. Also, if the service used for communication with the C2 server under the Management (Mngt) phase of [Table 4](#) is investigated, it can be seen that the IRC service is used for C2 communication from Hydra to Tsunami. After Aidra, other services started to be used for communication with the C2 server. This finding indicates that Aidra contains some new features for other architectures, and it has a different module for communicating with C2 Server. In this context, the Phylogenetic Tree of the IoT Malware Families is built-up and shown in [Fig. 5](#). Our Phylogenetic Tree of IoT Malware is an acyclic graph. The nodes contain malware families, and their edges map ancestors. The numbers above the nodes show the inherited features that are defined in [Table 5](#).

TABLE 5 Inherited Feature Definitions

Feature Number	Feature
1.1	Scanning Module
2.1	Dictionary(Username and Password List)
3.1	Malware Download Server IP Address
3.2	Malware Delivery Technique
4.1	SYN Flood
4.2	ACK Flood
4.3	UDP Flood
4.4	HTTP Flood
4.5	PDoS
5.1	C2 Technique
5.2	Control Server
5.3	C2 Server IP Address
6.1	Obfuscation/Encryption Technique
6.2	Kill Other Botnets or Process

TABLE 6 The Defense-in-Depth Approach for IoT Malware

Phase		Attack Vector	Defence Mechanism
Planning		System Discovery	Firewall Configuration
		Service Scanning	
Preparation		Dictionary Attack	Changing Default Credentials
		Exploit Vulnerabilities	Patching System
Cyber Intrusion	Delivery	Download Malware From Server	Blocking The IP Address Of Malware Download Servers
			Anti-Malware Tools (Static Analysis)
	Exploitation	Attack Vectors	Firewall Configuration
			Anti-Malware Tools (Dynamic Analysis)
Management		C2 Server Communication	Blocking The IP Address Of C2 Servers
			Blocking Unnecessary Ports
Sustainment		Attack Vectors	Anti-Malware Tools (Dynamic Analysis)

TABLE 7 Behavioral Analysis of IoT Malware Families[illegible]



One of the critical aspects obtained from the Phylogenetic Tree is the parent malware families. Four different malware families, Hydra, Tsunami, Gafgyt, and Mirai, are determined as parent malware families where some features of these malware families are highly inherited by other malware families. Besides, there are some small branches on the tree. Carna and Wifatch create a small branch as they are classified as “White-Hat Trojans.” Also, VPNFilter inherited some features from BrickerBot. On the other hand, there are 12 malware families without any relationship and similarities with other malware families. The inherited features of the malware are discussed below.

Three malware families Psypot, Chuck Norris, and Tsunami inherited some features from Hydra. All of these malware families target the same CPU architecture; however, we do not see that as an inherited feature as it is only related to the compile process of malware. The most significant inherited feature is the C2 technique. These malware families used Internet Relay Chat (IRC) channels to send commands to malware [32], [33], [36], [37], [39], [41], [42]. For this reason, these malware families are also known as IRC Botnets. Besides, the “Readme” file of the Hydra malware contains some evidence about the nationality of the developer. The e-mail address of the developer most probably belongs to an Italian person or organization [33]. Moreover, the IP address of the C2 server of Chuck Norris and Tsunami are the same, and the location of this IP address is in Italy [36]. Even though the location of the developers gives us strong evidence of the inspiration from the previous malware, we did not take this feature as an inheritance. However, Tsunami inherited the C2 server and obfuscation and encryption keys from Chuck Norris [36].

Seven malware families are located with inherited features from Tsunami. We analyzed the source codes of Tsunami and Aidra to detect the similarities between these two malware [41], [45]. The Aidra has two attack vectors, SYN Flood and ACK Flood. During our source code analysis, we detected that Aidra uses exactly the same source codes with Tsunami for these attack vectors. Reverse engineering efforts of other researchers also showed that Setag, Dofloo, and XoRDDoS malware families inherited SYN and UDP flooding attack vectors of

Tsunami [62], [65], [73]. Besides, XoRDoS uses a similar control panel with Dofloo which is based on HTTP File Server (HFS) [74]. The successful attack vectors of Tsunami are also adopted by the Gafgyt malware, however more sophisticated initial access and C2 communication techniques differ these malware families from each other [69]. Amnesia and Radiation have different attack vectors than other Tsunami variants, but they have the same C2 technique with Tsunami [86], [91]. Moreover, Radiation has the same dictionary (contains credentials for gaining initial access) with the Tsunami to be used for gaining initial access to the victim machine.

C. Inherited Features From GAFGYT

Eight malware families are located with inherited features from Tsunami. Analysis of the Remaiten revealed the comments of the developer on the source code. According to the comments of the developer, Remaiten adopted the “Telnet Scanning”, “Malware Delivery”, and “SYN Flooding” modules from Gafgyt; and “ACK Flooding”, “UDP Flooding”, and “C2 Communication” from Tsunami [84]. New Aidra also inherited some features from three previous malware. New Aidra adopted the “C2 Communication” module from Tsunami; “Telnet Scanning” and “Malware Delivery”, modules from Gafgyt, and it has the same dictionary from Mirai [103]. Hajime also inherits some features from Gafgyt and Mirai. Hajime adopted the scanning module from Gafgyt and the dictionary from Mirai [100], [101]. The infamous Mirai malware was also derived from Gafgyt. The dictionary used in the Gafgyt contains six usernames and 14 passwords. On the other hand, Mirai has 62 credentials which also include the Gafgyt’s set [5]. Mozi, Vbot, and Sbidiot inherit the attack vectors [63], [64], [80], [98], [99]. Enemybot also inherits attack vectors along with C2 module. [110]

D. Inherited Features From MIRAI

As it is mentioned hereinbefore, we are able to locate 29 malware families inspired by Mirai. The key features behind the huge impact of the Mirai attacks can be listed as:

- 1) Scanning module for finding new victims,
- 2) Dictionary of credentials to be used for the brute-force attack to gain initial access,
- 3) Simple but effective attack vectors,
- 4) Killing other processes which use Telnet, SSH, and HTTP services to avoid any performance loss and getting detected.
- 5) Lightweight obfuscation with encryption and decryption with XoR operation of strings.

Mirai inspired most of the IoT malware families from various aspects. Some of the inspired malware families inherit all of the mentioned features while some of them use only one feature. Reaper only inherits the scanning module [112]. Persirai inherits the scanning module, and UDP flooding [107]. Brickerbot and Manga inherit the scanning module and the dictionary [105], [106]. Satori, FBot, Shinoa, Miori, EchoBot, Ttint, Dark Nexus, Hoho, Katana, ZHtrap, Scsihelper, Beastmode, and RapperBot inherit the DDoS attack vectors of Mirai [34], [38], [51], [52], [57], [59], [60], [75], [85], [89], [90], [102], [108], [116], [117], [123], [130], [131]. Along with DDoS attack vectors, Shinoa and Miori also inherit the scanning module, ZHtrap inherits the dictionary, and Echobot, Ttint, and Scsihelper inherit the obfuscation technique of Mirai. EchoBot also has the same C2 server IP address with Mirai. Dark Nexus also inherits additional features from Mirai, it has the same IP address of C2 and malware download servers and inherits the scanning module and the killing other botnets feature. LiquorBot inherits different features of Mirai. It has the same C2 server, download module, and string obfuscation technique with Mirai [40]. MooBot, Mukashi, Manga, Enemybot inherit the scanning module and dictionary of Mirai [46], [47], [48], [77], [78], [106], [110], [111]. Additionally, MooBot inherits the obfuscation technique, Mukashi inherits attack vectors, and Enemybot inherits killing other botnets features of the Mirai. Rhombus and Zuorat inherit the scanning module and obfuscation technique while AirdropBot and Vbot only inherit the obfuscation technique with the same XoR key of the Mirai [54], [72], [80], [119]. Sora, Unstable and Dark.IoT inherits the dictionary and obfuscation technique of Mirai [59], [60], [67], [132]. Besides, Sora and Unstable have the same malware download server IP address of Mirai.

PDF

Help

SECTION VI.

Behavioural Analysis of Malware Families

Most of the malware families located within the scope of this project follow a common threat pattern. The common threat pattern contains five phases to compromise the target machine. As the first step of the attack, malware scans the network to find a target via its built-in scanning module. In the second phase, the malware attempts to establish initial access to the target. There are two different techniques adopted by the malware to succeed in this phase. One of these techniques is trying default password and username combinations in a brute-force attack and the other technique is exploiting known vulnerabilities of a running service on the victim machine. After establishing successful initial access, the third phase of the attack begins. In this phase, the malware communicates with the C2 server or P2P host to download malware ELF files to the victim device. After downloading the malicious ELF file, the malware waits for the fourth phase of the attack. In this phase, malware listens to the dedicated port for commands from the C2 server to conduct malicious activities. The fifth and last phase does not apply to all the malware families in our collection. Some of the malware families try to maintain their presence on the target device or avoid getting detected or analyzed. For this reason, malware families apply different techniques such as killing other malware processes, blocking some ports, or even wiping the target device. The common threat pattern of the IoT malware is shown in Fig. 6.

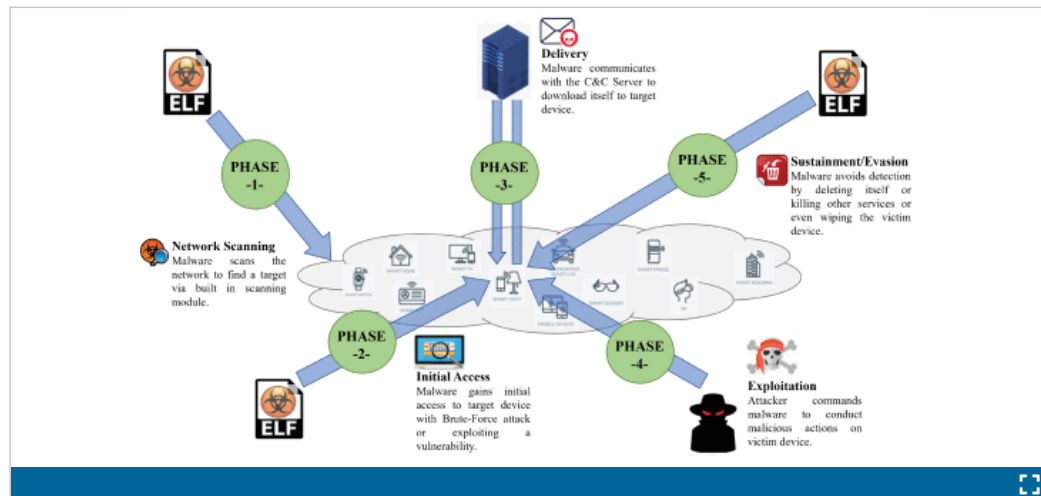


FIGURE 6.
Common threat pattern of IoT malware families.

As a result of our efforts, we present the behavioral analysis IoT malware families by adopting a layered investigation framework. By utilizing this framework, the tactics of the adversaries to exploit the target devices, and techniques (along with ID numbers) used for conducting the mentioned tactics are given below and in Table 7. Besides attack vectors, exploited vulnerabilities, communication ports, and services are presented. Our investigation framework of the Behavioral Analysis of IoT Malware Families is presented in Table 7 in Appendix A.

A. Planning

- **Common Tactic: Discovery:** The attacker tries to collect information about the target device.
- **Techniques:** There are two different techniques to achieve the “Discovery” tactic.
 - **Remote System Discovery (To846):** Detection of running hosts by IP address, hostname, or other logical identifiers on a network.
 - **Network Service Scanning (To841):** Conducting port scanning to locate running services on the victim host.

In the planning phase for reconnaissance, all 64 malware families are using the discovery tactic on Mitre ATT&CK for ICS. To conduct this tactic, they used two different techniques which are, “Remote System Discovery” and “Network Service Scanning,” regarding their attack vector in the Preparation phase. All the malware families used their built-in scanning modules to accomplish the reconnaissance phase. The distribution of the used techniques is shown in Fig. 7.

PDF

Help

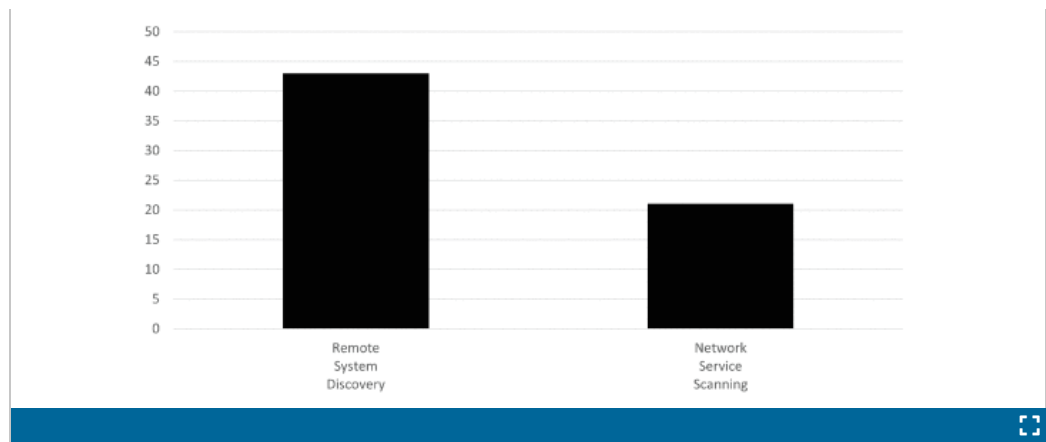


FIGURE 7.
Distribution of techniques on planning phase.

B. Preparation

- **Common Tactic: Initial Access:** The attacker tries to get access to the target device.
- **Common Technique: Internet Accessible Device (To883):** Attackers gain access to Edge/Fog computing environment through devices exposed directly to the internet.

In the preparation phase, malware is weaponized; and target hosts are set based on the results of the previous phase. The applied tactic in this phase is the same for all malware families. All the malware families attempt to establish an “Initial Access” to the victim host. To implement this tactic, all the malware families used the “Internet Accessible Device” technique to establish a first connection with the victim host. Malware families differ according to the attack vectors during this phase. In order to gain initial access to the victim machine, some malware families use publicly known built-in credentials for the Brute-Force attacks. For example, the infamous Mirai malware has a dictionary consisting of 61 credential pairs to brute force the victim device. Another attack vector is exploiting the vulnerabilities of a target device. All of the exploited vulnerabilities are shown in Table 7. While some of the malware families use only one attack vector, others use both attack vectors. The distribution of the used attack vectors is shown in Fig. 8.

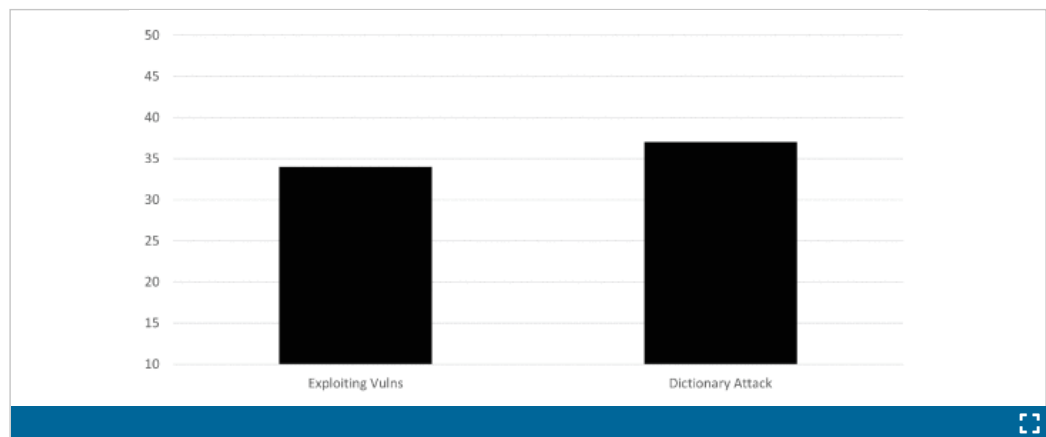


FIGURE 8.
Distribution of attack vectors on preparation phase.

C. Cyber Intrusion

The Cyber Intrusion phase contains two sub-phases as Delivery and Exploitation.

1) Delivery

- **Common Tactic: Persistence:** The attacker tries to get access to the target device.

PDF

Help

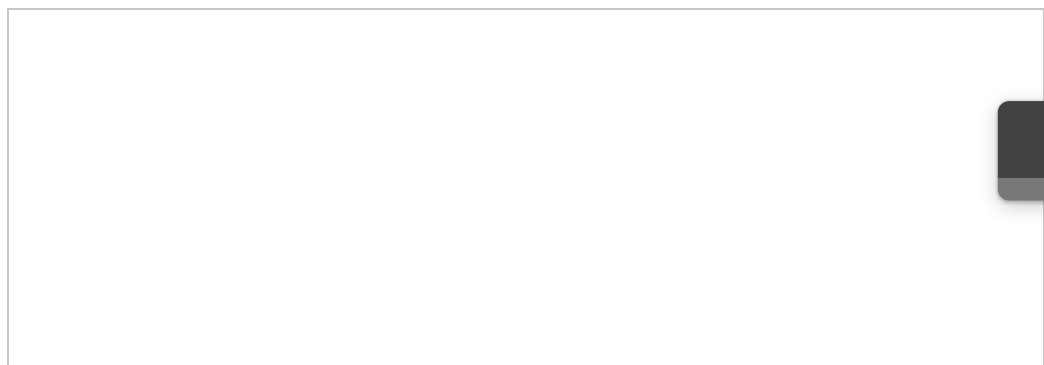
- **Common Techniques: Program Download (To843):** Performing a program download to transfer malicious software

All malware families follow a common tactic and technique in this sub-phase. One of the essential capabilities of malware is to be able to achieve persistence. The persistence tactic is used by attackers to maintain their presence in the victim device after gaining initial access in the previous phase. The “Program Download” technique is used to conduct this tactic. Attackers attempt to infect the victim device with a malicious code. There are two different attack vectors in this subphase, after the initial access to the victim device, most of the malware families attackers download their malicious code from their servers to ensure their presence on the system. On the other hand, instead using a centralised malware download server, five of the malware families (Wifatch, Hajime, Hide and Seek, Mozi, and HeH) deliver malware with P2P connection.

2) Exploitation

- **Tactics:** There are four different tactics in Exploitation Subphase
 - **Inhibit Response Function:** Preventing the target device’s safety, protection, quality assurance, and operator intervention functions from responding to a failure, hazard, or unsafe state.
 - **Impact:** Aims to manipulate, interrupt, or destroy target device.
 - **Execution:** Running malicious code on the victim host.
 - **Persistence:** Maintaining presence in the victim device.
- **Techniques:** There are eight different techniques to achieve relevant tactics.
 - **Denial of Service (To814):** Performing Denial-of-Service (DoS) attacks to disrupt expected device functionality.
 - **Manipulation of Control (To831):** Manipulation on communication or commands of victim host.
 - **Loss of View (To829):** Causing permanent loss of view where the target IoT device requires local, hands-on operator intervention.
 - **Theft of Operational Information (To882):** Stealing important data.
 - **Loss of Availability (To826):** Disrupting essential components or systems.
 - **Command Line Interface (To807):** Utilizing command line interfaces (CLIs) to interact with victim device and execute commands.
 - **Scripting (To853):** Using scripting languages to execute arbitrary code.
 - **Program Download (To843):** Performing a program download to transfer malicious software.

In this sub-phase, all malware families differ from each other in terms of Tactics, Techniques, and Attack Vectors. Malware families use four different tactics (Inhibit Response Function, Impact, Execution, and Persistence) to compromise the victim device. The distribution of used Tactics is shown in [Fig. 9](#).



PDF

Help

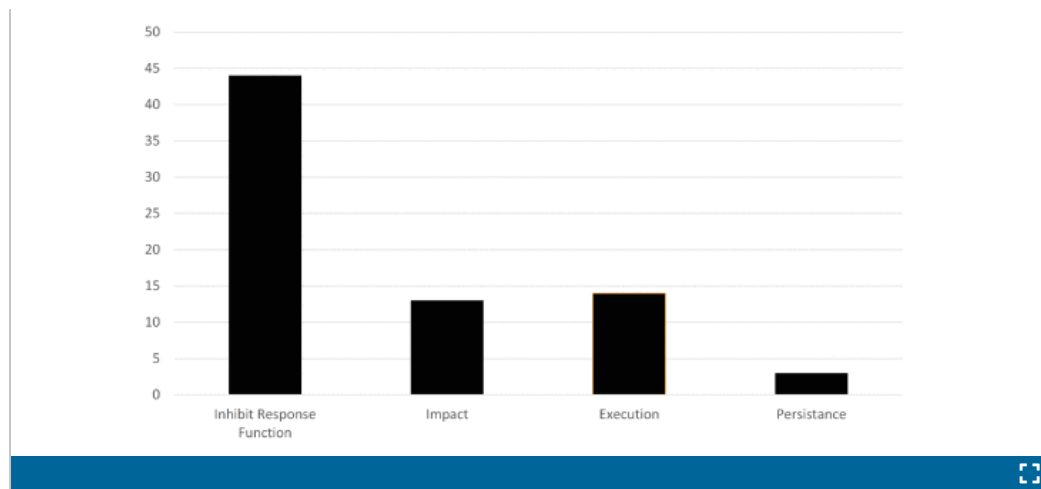


FIGURE 9.
Distribution of tactics on exploitation phase.

Eight different techniques are used to implement those tactics. The DDoS technique is used for the Inhibit Response Function tactic. For the Impact tactic, Manipulation of Control, Loss of View, Theft of Operational Information, and Loss of Availability techniques are used. For the Execution tactic, Command Line Interface and scripting techniques are used. Lastly, for the Persistence tactic, the Program Download technique is used. The distribution of the attack vectors by techniques is shown in Fig. 10.

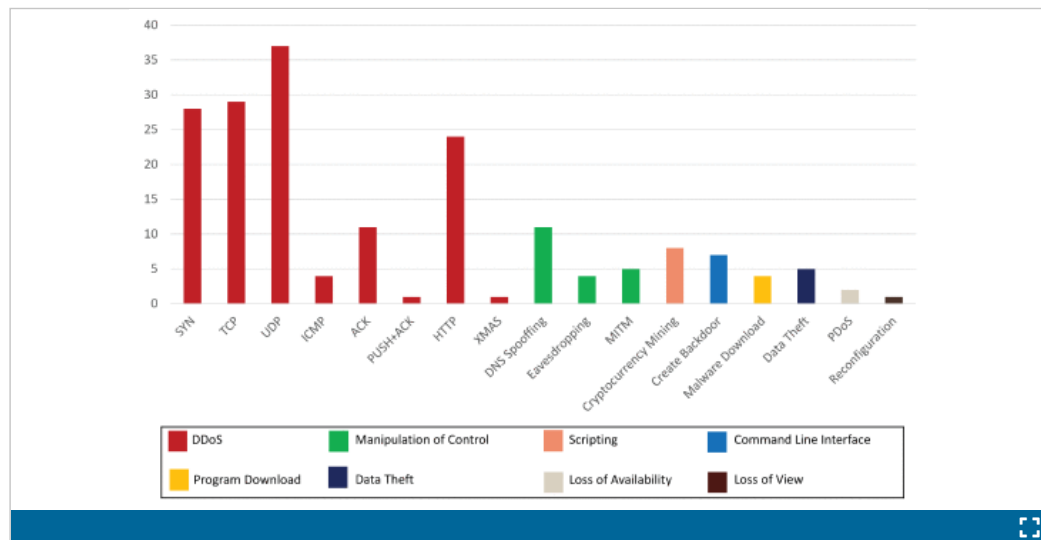


FIGURE 10.
The distribution of attack vectors by techniques on exploitation phase.

IoT malware families used 18 different attack vectors to exploit the victim device. Eight of these attacks are the “flooding attacks” for DDoS attacks. After the DDoS attacks, the most commonly used attack type is DNS Spoofing. Some malware families such as Tint and LuaBot were detected to be used for Socket Secure (SOCKS). With the newly emerging paradigm Malware-as-a-Service (MaaS), these botnets are rented to malicious people to route their malicious traffic over these victim devices as proxy servers. Another interesting attack vector is cryptocurrency mining. These devices are highly constrained in terms of computation power. For this reason, it is impossible to mine Bitcoin, which requires high computational power. However, there are a few other crypto coins like “Monero,” which do not require too much computation power for mining activities. Eight of the malware families (13%) have an attack vector of cryptocurrency mining. There are two malware families, “Chuck Norris” and “Brickerbot”, which make the victim devices unusable without operator intervention. Chuck Norris re-configures the device firmware and requires a user to reboot the device. On the other hand, “BrickerBot” wipes the partial or whole of the device memory and attack with the “Permanent Denial of Service (PDOS)” attack vector. After the “BrickerBot”

attack, more expertise intervention is required to make the device operable again. The other attack vectors are “Download Another Malware,” “Social Media Hijacking,” “Eavesdropping,” “Man in the Middle (MITM),” “Creating Backdoor,” and “Executable Installer.”

D. Management

- **Common Tactic: Command and Control(C2):** The attacker tries to communicate and control the victim machine to use it for malicious purposes.
- **Techniques:** There are two different techniques to achieve this tactic.
 - **Commonly Used Port (To885):** Communicating on a commonly used port to not get detected by firewalls or network detection systems and to hide behind normal network activity to avoid more detailed inspection.
 - **Connection Proxy (To884):** Using a connection proxy to direct network traffic between systems or act as an intermediary for network communications. Also, this definition of a proxy can be expanded to encompass trust relationships for P2P networks consisting of hosts or systems that regularly communicate with each other.

During the management phase, all the malware families adopt the same Tactic. Malware developers use “Command and Control Tactic” to conduct their malicious activities on the infected device. They need to establish communication to send their commands to malware and run this command in the victim device. Two different techniques are applied for achieving this tactic. Most malware families apply the “Commonly Used Ports” technique to hide behind normal network data packets.

On the other hand, five of the malware families apply the “Connection Proxy” technique. Malware families Wifatch, Hajime, Hide and Seek, Mozi, and HeH use P2P network communication to communicate with the infected device. This new technique makes it impossible to centrally disinfect these malware families because malware researchers and security companies generally disable malicious C2 servers. Since P2P botnet malware does not need such a central C2 server, it is much more difficult to disinfect them centrally [133].

All the malware families adopted the same approach in this phase and apply the “Commonly Used Port” technique. After the victim machine gets infected, the malware establishes communication with C2 servers to get commands and download extra malicious files if needed. However, malware families get differ on the attack vector side. Fourteen different ports and seven different services are used by malware to communicate with their server. IRC, Telnet, TCP, UDP, DHT, SSH, and WSS services are used for this purpose. The distribution of the used ports is shown in Fig. 11. On the other P2P IoT botnet malware families do not have any central C2 servers, and their communication ports and services may vary by the command. The details of the communication protocols of the P2P botnets could be found in the relevant analysis researches [63], [87], [118].

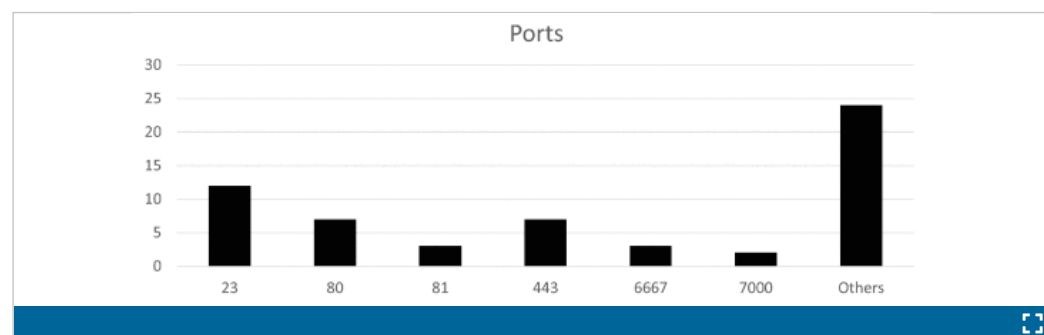


FIGURE 11.

Distribution of used ports on management phase.

E. Sustainment/Evasion

- **Tactics:** There are five different tactics for sustainment on the target or evasion of the target to avoid detection or removal from the device.

PDF

Help

- **Inhibit Response Function:** Preventing the target device's safety, protection, quality assurance, and operator intervention functions from responding to a failure, hazard, or unsafe state.
- **Impair Process Control:** Trying to manipulate, disable, or damage running processes.
- **Evasion:** Avoiding to get detected.
- **Persistence:** Maintaining presence in the victim device.
- **Impact:** Aims to manipulate, interrupt, or destroy target device.
- **Techniques:** There are five different techniques to achieve relevant tactics.
 - **Block Serial COM (To805):** Blocking access to serial communication ports to prevent instructions or configurations from reaching target hosts.
 - **Service Stop (To881):** Stopping or disabling services on a system to render those services unavailable to other users.
 - **Indicator Removal on Host (To872):** Attempting to remove artifacts of their presence on the target host to cover their tracks.
 - **System Firmware (To857):** Exploiting the firmware update feature on accessible devices to upload malicious or out-of-date firmware.
 - **Manipulation of View (To832):** Attempting to manipulate the information reported back to operators or controllers.

As in the Exploitation sub-phase, all malware families differ in Tactics, Techniques, and Attack Vectors in this phase. Five different tactics are used by malware families for sustainment or avoiding detection purposes. It can be observed from Table 7 that malware families before 2016 (before Remaiten) do not have any sustainment or evasion tactics generally. Only two of the malware families, Wifatch and Setag, used sustainment tactics. If we take into consideration that Wifatch is a "White-Hat-Trojan" and aims to protect IoT devices from getting infected by other malware, Setag is the first malware that uses sustainment tactics to maintain its presence on the target host. Even though some new malware types emerged between Setag and Remaiten, none of them applied any tactics for sustainment and evasion purposes.

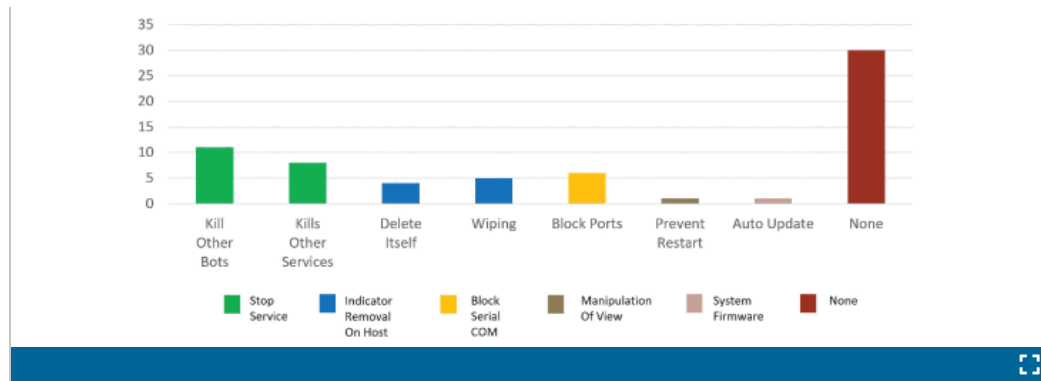
Also, five different techniques are used to conduct those tactics. For the Inhibit Response Function tactic, the "Block Serial COM" technique is used. Using this technique, malware blocks some specific ports that are generally used by other malware. As mentioned hereinbefore, IoT devices are highly constrained devices in terms of computation power, and running more than one malware may result in performance loss. This performance loss may be noticed by the user, and the user may take some prevention against it. With this understanding, this technique is generally used for the prevention of other IoT botnet malware infecting the device. Also, the "Service Stop" technique under the Impair process control is used for the same purpose. With this technique, if there is any other botnet malware on the victim host, it would be killed before beginning any other malicious activity.

Another technique used in this stage is "Indicator Removal on Host" to conduct the "Evasion" tactic. With this technique, the malware tries not to get detected on the victim device. To avoid getting detected, different attack vectors are used by malware families. Remaiten and Liquorbot delete themselves or some of the downloaded malware files from the C2 server. Amnesia has a very particular attack vector. If it detects that it is running in a Virtual Machine, it wipes the VM to avoid being exposed and getting analyzed. Birckerbot, VPNFilter, Silex, and HeH wipe some part or whole of the host device.

Only one malware family uses the System Firmware technique to implement the Persistence tactic. VBot has a feature to update itself automatically, like a firmware update. Another technique used by only one malware family is the Manipulation of View. This technique is applied only by Katana to conduct the "Impact" tactic. This malware prevents the infected device from restarting to maintain its existence. The distribution of attack vectors by techniques is shown in Fig. 12.

PDF

Help

**FIGURE 12.**

Distribution of attack vectors by techniques on sustainment/evasion phase.

F. Lessons Learned

Defense-in-depth approach needs to be applied to defend against IoT malware. An efficient defense mechanism should aim to defend the whole computing environment for every phase of the malware attack. In the first phase of the attack, two different attack vectors are applied by the malware. Both of these attack vectors are based on network scanning. One of these attack vectors is discovering the running hosts on the network and the other one is discovering running services on these hosts. Defending against both of these attack vectors is very easy by configuring a firewall properly.

The second phase also contains two different attack vectors. Defense against these attack vectors requires the awareness of the users. Most of the malware uses default username and password tuples to gain initial access. Changing default usernames and passwords can easily provide security against these malware families. The other attack vector is exploiting vulnerabilities to gain initial access. To defend against these malware families, the best action is to apply patches in a timely manner.

The third phase contains two sub-phases. The first sub-phase is delivery. Blocking the known IP addresses of known malware download servers may partially provide security. Another solution is deploying anti-malware tools based on static analysis. Nevertheless, there are no prevailing anti-malware tools for IoT devices. The second sub-phase is the exploitation sub-phase. There are two different defense solutions for this phase. One of these solutions is to apply proper firewall configuration. The other solution is again deploying an anti-malware tool, but in this sub-phase, the tool should be based on the dynamic analysis of malware and to the best of our knowledge there is no such anti-malware tool that runs on IoT devices for now.

In the fourth phase, the applicable defense mechanism is very similar to the delivery sub-phase. Blocking the IP address of the known C2 servers prevents communication between the malware and the C2 server. Additionally blocking unnecessary ports is an efficient solution.

The last phase contains the sustainment techniques of the malware. Similar to the exploitation sub-phase, a dynamic analysis based anti-malware tool should be used to detect sustainment techniques, such as killing other processes, blocking some ports, or deleting some files. The defense-in-depth approach for IoT malware is presented in Table 6.

SECTION VII. Conclusion

IoT malware poses a severe threat to all devices in Edge/Fog computing environment. These ecosystems are still far from being secure, and there is much progress that needs to be done for securing edge devices. Several steps have to be taken for developing efficient defense mechanisms against malware for edge devices. We believe that developing an effective anti-malware mechanism is possible only with a high understanding of malware behaviors. This paper aims to bring light for future researches with presented features of the IoT malware.

PDF

Help

This research fills three important research gaps for an improved understanding of defending IoT devices against malware attacks. As a first step, 64 different malware families have been introduced to familiarise IoT malware. These 64 different malware families cover the majority, if it is not all, of existing IoT malware ranging from its debut in February 2008 to recent ones by the published date of this paper. Then, the evolutionary development of these malware families are presented with a phylogenetic tree. Four different malware families Hydra, Tsunami, Gafgyt, and Mirai, are determined as parent malware families. Those malware families could be assessed as milestones in IoT Malware development and their features are highly inherited by other malware families. Lastly, a methodical characterization of existing IoT malware families is presented. The malware families are characterized by various aspects, such as target architecture, evolution, delivery techniques, attack vectors, and sustainment methods. Further detailed analysis of IoT malware shows that the velocity of newly emerging malware families, increased sophistication, and newly applied techniques such as P2P communication and obfuscation techniques pose significant challenges for IoT malware detection.

Considering the significant malware threat for IoT devices, it is surprising to see that the literature lacks a fundamental methodology to guide malware analysis and development of anti-malware tools in the IoT domain. We believe that the first step for securing IoT devices highly depends on the understandability of the malware behaviors. During the literature review, we surprisingly realize that there is only a limited number of studies on the IoT malware analysis domain. Most of the studies focused on malware detection based on machine learning. Developing efficient anti-malware tools requires combining the efforts of two different disciplines which are Malware Analysis and Machine Learning. These two disciplines require different skills. Specifically, malware analysis mostly requires reverse engineering skills along with knowledge of operating systems and network communication, while machine learning requires mostly statistics. In this regard, a malware analysis framework that combines static and dynamic analysis methods is highly needed for providing a dataset for the training process of machine learning-based approaches.

Despite the novel approach of this study, it is worth mentioning some remaining challenges and limitations. Securing edge devices against malware is insufficient for providing a secure ecosystem in Edge/ Fog computing. This research only presents the malware threat on the edge devices layer. Similar research also has to be conducted for the other layers of the Edge/Fog computing environment. Providing end-to-end security on all over the Edge/Fog Computing ecosystem is only possible by securing all the layers.

Appendix A

Malware BEHAVIORAL Analysis Framework for Edge Devices

See [Table 7](#).

Authors	▼
Figures	▼
References	▼
Citations	▼
Keywords	▼
Metrics	

PDF

Help

ALSO ON IEEE XPLORE

Empowering Network Security: BERT ...

4 months ago · 1 comment

Intrusion detection systems (IDS) stand as formidable guardians in network ...

Graph Neural Networks for Individual ...

6 months ago · 1 comment

Individual treatment effect (ITE) estimation is an important task for ...

Machine Learning-Enabled ...

5 months ago · 1 comment

Hypertension, referred to as the "silent killer" by the World Health ...

Statistical Insights Into Machine ...

9 months ago · 1 comment

Maternal mortality is a major public health concern worldwide. It is the ...

Early Detection of Ripeness for the ...

a year ago · 1 comment

The seed oil of Xanthox sorbifolium is a new kind vegetable oil which is ...

0 Comments

1 Login ▾

G

Start the discussion...

LOG IN WITH



OR SIGN UP WITH DISQUS ?

Name

Email

Password

By clicking submit, I authorize Disqus, Inc. and its affiliated companies to:

- Use, sell, and share my information to enable me to use its comment services and for marketing purposes, including cross-context behavioral advertising, as described in our [Terms of Service](#) and [Privacy Policy](#)
- Supplement the information that I provide with additional information lawfully obtained from other sources, like demographic data from public sources, interests inferred from web page views, or other data relevant to what might interest me, like past purchase or location data
- Contact me or enable others to contact me by email with offers for goods and services (from any category) at the email address provided
- Process any sensitive personal information that I submit in a comment for the purpose of displaying the comment
- Retain my information while I am engaging with marketing messages that I receive and for a reasonable amount of time thereafter. I understand I can opt out at any time through an email that I receive. Companies that we share data with are listed [here](#).



♡ Share

Best Newest Oldest

Be the first to comment.

Subscribe

Privacy

Do Not Sell My Data

PDF

Help

IEEE Personal Account

CHANGE
USERNAME/PASSWORD

Purchase Details

PAYMENT OPTIONS
VIEW PURCHASED

Profile Information

COMMUNICATIONS
PREFERENCES

Need Help?

US & CANADA: +1 800
678 4333

Follow




DOCUMENTS

PROFESSION AND
EDUCATION

TECHNICAL INTERESTS

WORLDWIDE: +1 732
981 0060

CONTACT & SUPPORT

About IEEE *Xplore* | Contact Us | Help | Accessibility | Terms of Use | Nondiscrimination Policy | IEEE Ethics Reporting  | Sitemap | IEEE Privacy Policy

A public charity, IEEE is the world's largest technical professional organization dedicated to advancing technology for the benefit of humanity.

© Copyright 2025 IEEE - All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies.

