Providing IoT host-based datasets for intrusion detection research *

Vitor Hugo Bezerra[1], Victor G. Turrisi da Costa[1],Ricardo Augusto Martins[1], Sylvio Barbon Junior[1], Rodrigo Sanches Miani[2], Bruno Bogaz Zarpelão[1]

[1]Computer Science Department – State University of Londrina (UEL)
Londrina – PR – Brazil

[2]School of Computer Science (FACOM) – Federal University of Uberlândia
Uberlândia – MG – Brazil

{vitorbezerra, victorturrisi, barbon, brunozarpelao}@uel.br,

martinsaugustoricardo@gmail.com, miani@ufu.br

Abstract. *The high number of vulnerabilities in Internet of Things devices has created malware-prone networks. A type of malware that imposes a serious threat to the Internet security is known as botnets. This malware exploits some vulnerabilities of IoT devices to infect them and perform large-scale Distributed Denial of Service attacks, affecting many users who depend on their services. This work presents the construction of an experimental environment to generate a dataset that contains data from a real IoT device that was infected by botnet malware in a laboratory. The dataset can be used to support the development of defence tools for IoT devices to identify botnets, as it contains network traffic and host-based features, such as, CPU and memory usage. The dataset and network environment files are available for the research community.*

1. Introduction

Internet of Things (IoT) has promoted great changes in our everyday life in many aspects, e.g., health care and traffic monitoring services. Among other things, this paradigm made machine-to-machine communication over the Internet possible, connecting more devices online and allowing them to actively participate in the network. IoT consists of many heterogeneous and low-cost devices with little or no security embedded into them, which generate a huge amount of private information, and may create many security problems. This open wound in IoT security is likely to prevail for years to come and must not be ignored given the broad range of applications of the IoT paradigm [Angrishi 2017, Atzori et al. 2010, Whitmore et al. 2015].

According to Gubbi et al. [2013] , in 2011, the number of interconnected devices overcame the number of people in the world, and forecasts indicate the number of devices will reach 24 billion in 2020. This high increase can be associated with the huge amount of devices that will be connected for IoT purposes. Likewise, this has attracted the attention of malicious users, who may target those devices to gather computation power to carry out illegal activities. Those users can, for example, perform Distributed Denial of Service (DDoS) attacks by creating a large-scale IoT-based botnet [Angrishi 2017]. Moreover, compromised devices may not demonstrate any clear symptoms of infection, being able to

*The authors would like to thank CAPES for financial support.

continue with the execution of their normal activities. Therefore, detecting compromised devices is a challenging subject and requires specialised tools [Kolias et al. 2017].

A botnet is a collection of compromised devices, referred to as bots, controlled by one or more malicious users, which communicate with the bots to perform malicious activities. In 2016, a botnet called Mirai infected surveillance cameras - a type of IoT device - by taking advantage of their default security settings and performed a large-scale DDoS attack against Dyn, a DNS service provider [Mansfield-Devine 2016]. To neutralise the threat imposed by botnets, systems capable of efficiently detecting them are needed [Costa et al. 2017]. However, developing this kind of tool requires a detailed study of the behaviour of IoT botnets. García et al. [2014] point out that one of the main problems when creating new defence approaches is the lack of datasets to aid in the understanding of botnets.

This problem occurs because security analysts do not share their information due to concerns about privacy, and also by the fact that there are a few botnets that are available for studying [García et al. 2014]. An alternative solution to mitigate the lack of real datasets is the creation of controlled environments to support experiments, e.g., simulating a botnet scenario. To create a scenario containing IoT botnets, for example, it is necessary to set up an isolated network, find botnet malware samples with active servers, be able to capture the communication of the bot with their controller and create tasks to mimic normal traffic. These requirements rely on very time-consuming tasks, and acquiring IoT devices to build these type of networks and environments can be expensive, due to taxes and charges in some places of the world.

The lack of IoT-based datasets for security research can be noted in some works that propose approaches to protect IoT devices from network attacks [Raza et al. 2013, Cervantes et al. 2015, Amaral et al. 2014]. These approaches were evaluated in simulated networks composed of nodes with Contiki and TinyOS operating systems (OS), without creating, or using, any datasets to evaluate and test them [Zarpelão et al. 2017].

This work presents the construction of an experimental environment to generate a dataset containing data collected from a real IoT device, a Raspberry Pi, which was infected by botnet malware. The goal of this work is to provide means to observe different perspectives of the operation of the device such as the use of CPU and memory, electric potential difference, and network traffic. The collected data includes legitimate and malicious behaviour. Legitimate data concerns the emulation of regular behaviour of domestic IoT devices, such as surveillance cameras and media centres. On the other hand, the malicious data is related to botnets' activities, including communication with the Command & Control (C&C) and some DDoS attacks. The dataset was labelled and the files were separated to organise the different situations faced during the experiments.

The contributions of this paper are the following:

1. Creating a controlled environment that has characteristics of a domestic IoT network and can be infected by IoT botnet malware;
2. Providing resources that emulate the behaviour and functions of domestic IoT devices;
3. Providing a labelled dataset that covers periods of legitimate and malicious activity of an IoT device. This dataset will be shared with the academic community;
4. Analysing and discussing the behaviour of IoT botnets, using traffic and host-based features, providing some guidelines to researchers on intrusion detection in

the IoT field.

This paper is organised as follows: Section 2 describes the prior works done in network dataset creation. In Section 3, a brief background about the botnet malware used is presented. Section 4 describes the experimental environment constructed. Section 5 presents the experiment done to create the dataset and the files available, and, in Section 6, a brief analysis of the behaviour of the device infected by botnets is presented. Lastly, Section 7 provides the conclusion of the work.

2. Prior Work

There are several examples of datasets containing network traffic, e.g., CAIDA[1] (Center for Applied Internet Data Analysis), KDD [Bay et al. 2000] (Knowledge Discovery and Data Mining), DARPA[2] (Defense Advanced Research Projects Agency), and CIC (Canadian Institute for Cybersecurity) [Shiravi et al. 2012], which have been used by researchers for many purposes, such as, to test the effectiveness of different IDS (Intrusion Detection Systems) and provide benchmark resources for the comparison of proposed solutions.

The CAIDA repository consists of datasets with different types of network data collected from a diverse range of situations. Some packet fields are anonymised or completely removed for security reasons. Also, some packets are not labelled. Nonetheless, these processes can have a negative impact on the evaluation of an IDS [Shiravi et al. 2012, Garcıa 2014]. To prevent this, we did not remove any payload data from the packets in our dataset.

Another public repository is the DARPA dataset, which was built by the Lincoln Laboratory at the Massachusetts Institute of Technology (MIT) with the objective of evaluating IDS. The datasets in this repository are in the *tcpdump* format and include all the packets payloads, which were collected from real and emulated machines, and where attacks were only performed on real machines. However, the experimental environment used to build the dataset was not made public and the attack methods used are considered outdated [Thomas et al. 2008].

The KDD repository has a network dataset as well, which is mainly used for benchmark tests of IDS. It contains data collected from six machines, where three were real machines and the others were simulated. Each simulated machine had a different OS. The three real machines were used to generate background traffic. In addition, a sniffer captured and stored the traffic in *tcpdump* format. As DARPA datasets, the KDD dataset is quite old, so the attacks and the background traffic do not match to current networks reality.

CIC provides a dataset and scripts that can be used to generate legitimate and malicious traffic on-demand [Shiravi et al. 2012]. Their dataset is composed of network traffic collected from several machines, which performed regular activities and were also infected by different malicious applications. Although it meets the criteria of a good dataset described in [Shiravi et al. 2012], it may not represent a real botnet behaviour [Garcıa 2014], as they used a botnet malware developed by them. In this sense,

[1] http://www.caida.org/home/

[2] https://www.ll.mit.edu/ideval/data/

to have a realistic scenario, we use only samples of real and active botnet malware for IoT devices.

Overall, it can be observed that CAIDA, KDD and DARPA repositories focus on benchmark tests, but they use obsolete attack types and tools. In addition, they do not have all the necessary network traffic information, e.g., destination IP addresses, payloads, and labels. The CIC dataset is newer than the other ones mentioned, hence it includes more recent attacks such as botnets and HTTP Denial of Service (DoS). However, as CAIDA, KDD, and DARPA datasets, the CIC dataset does not provide data related to IoT. Considering these previous works, we built an IoT dataset that fills the pointed gaps, such as using real samples of botnet malware, including legitimate traffic and providing all packet payloads.

3. IoT Botnets

IoT networks usually rely on low-cost devices with limited resources, e.g., surveillance cameras, temperature sensors, and baby monitors, which perform very specific tasks, generally requiring little computational power [Angrishi 2017]. Due to the simplicity of those devices, they normally have little or none embedded security.

Therefore, they can be compromised without much effort by a malicious user, and be used to perform several illegal activities, e.g., a DDoS attack [Zargar et al. 2013]. DDoS attacks consist of compromising a very large group of devices, usually scattered around the globe, which perform coordinated DoS attacks. This type of attack is characterised by performing precise attempts to compromise a service [Angrishi 2017]. Generally, it achieves this goal by sending a large volume of packets that occupy a significant proportion of the available bandwidth and/or computational capacity of the target [Peng et al. 2007] [Kolias et al. 2017].

Usually, DDoS attacks are controlled by a botnet, a network formed by infected devices remotely controlled. Botnets are composed of three main parts: the bots, which are the infected devices; the botmaster, which is the malicious user that controls the bots; and the C&C infrastructure, which is used to establish a communication channel between the botmaster and the bots [Silva et al. 2013, Costa et al. 2017]. One point to note is that bots can be any device ranging from traditional desktop computers to mobile and IoT devices. By using a large collection of bots, the botmaster is able to carry out a diverse set of coordinated malicious actions such as DDoS attacks, generating spam and stealing sensitive information.

There are several types of botnets spread over the Internet. This work was focused on botnets designed for IoT devices that run Linux or BusyBox[3], i.e., programs found in some domestic IoT devices and exploited by botnets. Using the VirusShare[4] platform, we searched for malware samples for IoT architectures, such as MIPS and ARM, and by names of IoT botnets found in [Angrishi 2017]. Seven botnet families were found with working samples as well as the source code of the Mirai botnet [5]. The botnets used in this work are:

- **Mirai**: this botnet focuses on compromising IP cameras by performing a brute-force attack to Telnet. Its main malicious activity is to perform a range of DDoS

[3] https://busybox.net/

[4] https://virusshare.com/

[5] https://github.com/jgamblin/Mirai-Source-Code

attacks. Also, it uses a centralised C&C infrastructure via HTTP protocol. In its peak of activity, it had a network traffic of 1.1 Tbps [Angrishi 2017];

- **Hajime**: a botnet that focuses on the same IoT devices as Mirai and uses the same type of infection strategy. The purpose of Hajime is unknown, as it has only closed some vulnerabilities in IoT devices, i.e., it has not behaved maliciously so far. Its C&C infrastructure is fully decentralised by using the BitTorrent Distributed Hash Table (DHT) protocol, while also encrypting its messages with RC4. When a device is infected by this botnet, the terminal prints a message informing the infection to the user[Stavrou et al. 2017, Kolias et al. 2017];
- **Aidra**: this botnet compromises devices with the following architectures: MIPS, MIPSEL, ARM, PPC, x86/86-64 and SuperH. Its infection process works in the same way as Mirai. This botnet malware opens a port on Linux-based devices or computers to wait for commands of the C&C server. The main malicious activity performed is DDoS attacks. A version of the botnet source code was released on the Internet [6] [Stavrou et al. 2017];
- **Bashlite**: the main targets of this botnet are Linux-based IoT devices. The infection of the device occurs through brute-force attack on the Telnet port using default usernames and passwords. After the infection, the device performs DDoS attacks on C&C command. The C&C's IP addresses for communication are included in the source code, facilitating the task of monitoring the botnet communication. In its peak of activity, it had a network traffic of 400 Gbps [Angrishi 2017];
- **Dofloo**: this botnet is found on regular OS for desktop computers, such as Windows and Linux, and IoT devices with MIPS and ARM architectures. This botnet is used to launch several DDoS attacks following the instructions of the centralised C&C, using AES-encrypted messages. In its peak of activity, it had a network traffic of 215 Gbps. Also, it collects memory, CPU and network traffic data and sends to the attacker [Angrishi 2017];
- **Tsunami**: also targeting IoT devices, the infection occurs by downloading and executing infected files. The malicious activity of this botnet is to perform DDoS attacks. It uses an IRC channel and HTTP requests to communicate with the malicious servers controlled by the attackers. When it infects a device, it changes its DNS server settings [Angrishi 2017];
- **Wroba**: having been found initially targeting Android devices [Abdul Kadir et al. 2015, Nigam 2015], Wroba migrated to IoT devices with ARM architecture. Its main purpose is to intercept or attack web banking activities, while infecting other devices. It connects with its C&C infrastructure using HTTP, but on Android devices it can also use SMS messages.

4. Experimental Environment

This section presents the environment developed to accomplish the main objective of this work, which is to build a dataset with host-based data collected from an IoT device infected by botnet malware. This work was aimed to provide a dataset that could be useful for researchers that are interested in IoT protection tools such as IDS but do not find a public IoT dataset to develop their work.

To build the experimental environment and, consequently, the dataset, works that proposed IDS for IoT were reviewed, analysing the IoT devices they used in their tests.

[6] https://github.com/eurialo/lightaidra

As a result, we decided to use a Raspberry Pi in our network. Next, profiles of operation were defined for the Raspberry Pi, emulating real devices such as multimedia centres and surveillance cameras. Lastly, a network scenario was created to support the dataset generation.

4.1. Device Selection

Works in the literature that address intrusion detection for IoT do not focus on a unique device or type of device to test their approaches on. Thus, there is no type of device that is considered a standard for IoT environments. This fact can be explained by the non-standard definition of IoT itself. IoT has a long list of definitions, which usually share only a few characteristics such as the use of devices with low-cost, limited resources and wireless communication capabilities. Table 1 presents some examples of devices used in previous works that addressed intrusion detection for IoT, since our focus is to contribute to this research field.

Table 1. IoT devices found in previous work.

Reference	Device(s) Used
[Amaral et al. 2014]	Cooja Motes
[Habibi et al. 2017]	Amazon Dash Button, Arlo Home Security System, August Smart Lock, Lifx Smart Bulb, Nest Thermostat, Raspberry Pi
[Summerville et al. 2015]	Ambient Weather, Network Camera
[Nobakht et al. 2016]	Philips Hue
[Oh et al. 2014]	Raspberry Pi
[Sforzin et al. 2016]	

As presented in Table 1, the devices range from very simple simulated boards (e.g., Cooja Motes) to daily objects transformed into IoT systems (e.g., Phillips Hue lamps and Amazon dash buttons), which are made by different companies and use different types of software to control them. Among the devices used in previous work, the Raspberry Pi was chosen because it is a customisable general purpose device with good computational power, but still limited to the IoT constraints. This device, which was used in [Habibi et al. 2017, Oh et al. 2014, Sforzin et al. 2016], has Wi-Fi and Bluetooth connectivity, is cheap, and supports a wide variety of OS, e.g., Linux and Windows. As commented by [Habibi et al. 2017, Nayyar and Puri 2015], the Raspberry Pi is a device that can be applied in several fields and regarded as a small, powerful and compact device. Also, it supports additional hardware, e.g., camera, and is energy efficient. The specifications of the Raspberry Pi used in this work will be presented in Section 4.3.

4.2. Device Profiles

Raspberry Pi is a general purpose device and can be used to develop multiple IoT devices. Therefore, we defined three typical profiles of IoT devices that could be targeted by the IoT botnets listed on Section 3, and implemented them in the Raspberry Pi. The profiles created are:

- **Multimedia Centre (MC)**: found in today's living rooms, a multimedia centre is a device that consumes streams of video such as movies or TV shows, to transform regular TVs in smart TVs. In addition to the use of video content, it also accesses an application store for updates and installation of other programs. The task of rendering video combined with the heavy traffic generated by video streams make this device to use a significant portion of its resources during its operation;

- **Surveillance Camera with Additional Traffic (ST)**: used both on the inside and outside parts of houses and companies, the operation of surveillance cameras is mostly characterised by the transmission of a video stream to a computer or station. However, this kind of device can also present traffic from other protocols. Telnet or SSH connections can be established to check if the device is working, its temperature, and the running processes, for example. These cameras may also include configuration web pages, which users access from their browsers to set up the parameters of cameras operation. As the device do not have to render video before transmitting, it consumes fewer resources than the multimedia centre;
- **Surveillance Camera (SC)**: this profile is similar to the ST profile, but it does not include as many interactions with users through Telnet, SSH, and configuration web pages as the ST profile. Having the video transmission as their single task, devices of this profile consume fewer resources than those of ST profile.

During all the experiments with the proposed profiles, the following programs are executed: *tcpdump*[7] software to capture network packets, *top* task manager program to capture CPU and memory usage and the number of processes running, and the function *vcgencmd*[8] of the Raspbian OS to capture the electric potential difference and CPU temperature. The video streams of ST and SC profiles were made using a VLC player[9]. In the MC profile, VLC was also used to consume video streams, and a Chromium Browser was employed to consume video and access the Chrome Web Store.

All the three profiles have Telnet and SSH ports open and a web server installed. This web server hosts the DVWA software[10], which, in these devices, emulates a vulnerable web page used as an interface for device configuration. However, only devices of ST profile have legitimate traffic related to these three services. These three profiles were created to emulate an environment with devices that have different levels of resource consumption. SC devices show the lowest levels under normal conditions, while MC devices show the highest ones. Resource consumption in ST devices is higher than in SC devices and lower than in MC devices. The different profiles also allow the generation of more diverse traffic. Consequently, more possibilities of analysis are provided, such as the possibility of observing the impact of botnets in devices with different levels of resource consumption and multiple types of legitimate traffic.

4.3. Network Scenario

After defining the device profiles that were going to be analysed, a network environment was created to support them. This environment provided Internet access and wireless communication for the IoT device. The network components present in the environment consist of a switch with Ethernet and Wi-Fi interfaces, four computers (*Machine 1*, *Machine 2*, *Machine 3*, and *Gateway*) connected to the switch through Ethernet, and one

[7] www.tcpdump.org/

[8] www.elinux.org/RPI_vcgencmd_usage

[9] www.videolan.org/

[10] www.dvwa.co.uk/

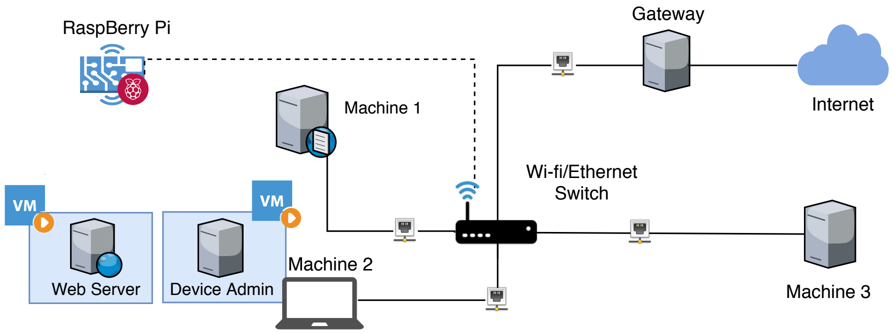

Figure 1. Network topology of the experimental environment.

Raspberry Pi model 3B connected to the switch via Wi-Fi. *Machine 2* provided two virtual machines, with the first one hosting a Web Server and the second being a client, referred to as *Device Admin*. Detailed specifications of the environment are presented in Table 2.

Table 2. Description of the experimental environment components.

Component	Function	Specs	Operating System	Virtual/Real
Raspberry Pi	Mimic different types of IoT devices	Ram: 1 GB CPU: ARM 1.2 GHz	Rasbian Stretch	Real
Machine 1	DNS server and video consumer	Ram: 4 GB CPU: Intel i5 3.2 Ghz	Ubuntu 17.04	
Gateway	Provide Internet access and routing			
Machine 3	Deliver and control malware	Ram: 8 GB CPU: Intel Xeon 3.1 Ghz		
Device Admin	Makes SSH and Telnet connections with the Raspberry Pi and updates	Ram: 1 GB CPU: 1 Ghz		Virtual
Web Server	Host a static web page used for updates in the device			

Figure 1 details the network environment used to generate the dataset. The *Gateway* was used to provide Internet access and DHCP functionality to the network. *Machine 1* hosted a DNS server that was used by the Mirai botnet. The same machine also hosted a VLC client that consumed the video stream generated by SC/ST profiles and a VLC server that generated a video stream for the MC profile. *Machine 2* hosted the *Web Server* and the *Device Admin*. The *Web Server* was responsible for providing a static Apache Web page, which was accessed by the *Raspberry Pi* using a script in all the profiles to emulate the consumption of web services. The *Device Admin* emulated the transactions of an administration software used to control and set up the emulated camera in the ST profile. To do so, it interacted with the *Raspberry Pi* through Telnet, SSH, and the Web configuration page. The *Raspberry Pi* was responsible for running the profiles presented in Section 4.2. Lastly, *Machine 3* was responsible for emulating an attacker on the network, which infected IoT devices with botnet samples. Particularly for Mirai, this machine also hosted a C&C server, which could be used to make the *Raspberry Pi* launch DoS attacks against selected targets.

5. Dataset

This section details the steps carried out to generate the dataset as well as the resulting files, which are available to the research community.

5.1. Dataset Construction

This section presents the experiment performed to create the dataset, using the environment described in Section 4. Our goal is to provide two types of logs: one related exclusively to legitimate activities, and the other containing data about both legitimate and malicious activities.

First, each profile defined in Section 4.2 was executed for one hour without any infection. The objective was to collect data exclusively related to legitimate activities. In ST and SC profiles, the *Raspberry Pi* transmitted a video stream to the *Machine 1* using VLC over HTTP through the port 8080. The video resolution was 80x40 pixels, which was chosen to simulate the minimum amount of data that a camera could send, and the transmission lasted for 20 minutes, with a total size of 4 MB. In the MC profile, the *Raspberry Pi* consumed videos from YouTube and Twitch [11], and a high definition video from *Machine 1*. It also accessed the Chrome Web Store, emulating searches for new applications. In all the profiles, the *Raspberry Pi* accessed a web page hosted by the *Web Server* at 1-minute intervals, emulating situations in which the IoT device consumes web services. Particularly in the ST profile, the *Device Admin* interacted with the *Raspberry Pi* at time intervals of approximately 5 minutes through Telnet, SSH, and the web configuration page. During the execution of each profile, the following data was collected in the *Raspberry Pi* every 5 seconds: CPU and memory consumption, electric potential difference, CPU temperature, and the number of tasks running. All the network packets transmitted and received by the *Raspberry Pi* were also captured.

After the acquisition of data exclusively related to legitimate activity, data was captured on situations that combined legitimate and malicious behaviour. To generate this part of the dataset, the infections was organised into three categories: *Type 1*, *Type 2*, and *Type 3*. We varied the number of samples and botnet families running on the device to have more diversified data. Table 3 presents the botnet families and the number of samples running in each profile and method of infection.

In the *Type 1* infection, three botnets were selected, and they were installed one at a time in the *Raspberry Pi*. The objective was to observe their behaviour without the interference of any other malware. The first botnet selected was Hajime, because its main purpose is unknown and then it would be better to analyse its behaviour without the presence of other botnets. Botnets Aidra and Bashlite were selected because they have the highest number of samples. Thus, combining them with other malware might degrade the *Raspberry Pi* performance, hindering the tests. To infect the *Raspberry Pi* with these malware, the *Machine 3* connected to the device via SSH using Rasbian default credentials (Raspberry:Pi). Then, malware samples were transferred to the device through the established connection.

The *Type 2* infection was focused on Mirai botnet. The Mirai source code is available on the Internet, making it possible to create a complete scenario with the malware and the C&C server in our environment. Having access to a C&C server instance, we could control all the life cycle of the botnet, since the infection until the DoS attacks against the victims. To infect the *Raspberry Pi* with the Mirai malware, we accessed the device from the *Machine 3* via SSH using the default credentials. Next, the Mirai C&C server was deployed at *Machine 3*. After having infected the device, we used the C&C command-line

[11] http://twitch.tv/

interface to make the *Raspberry Pi* launch DoS attacks against the *Gateway*, the *Machine 1*, and the *Web Server* at *Machine 2*. The duration of the attacks was short to simulate the malicious user testing the functions of the botnet.

In the *Type 3* infection, the objective was to observe the behaviour of the device when it is infected by multiple botnets. Therefore, in this infection type, we selected all botnets that were not used in previous ones. We also included the Mirai botnet to check whether the most popular IoT botnet would dominate or be dominated by its competitors. The infection process was carried out as in previous infections, i.e., via SSH connection.

Each type of infection was executed in each profile for one hour. After the end of each execution, we erased all data and the OS from the *Raspberry Pi* to avoid any contamination among the multiple types of infection.

Table 3. List of infections made in the Raspberry Pi profiles.

Infection Type	Profile	Botnet(s)	Method of Infection	Number of Samples
1	MC	Hajime	SSH	4
	SC	Aidra		11
	ST	BashLite		
2	MC	Mirai	SSH	1
	SC			
	ST			
3	MC	Mirai, Doflo, Tsunami,Wroba		28
	SC			
	ST			

Lastly, all the network traffic collected at the *Raspberry Pi* was labelled. Using the *Nfdump* software, NetFlow files were created from all the captured network packets. After that, all flows composed of packets that were captured during the one hour of execution without infection were labelled as legitimate.

To label the malicious traffic, firstly, all the flows with packets that were collected when the *Raspberry Pi* was infected were separated. Then, flows with IP addresses that did not match addresses of legitimate components of our network or known services such as YouTube, Twitch and the Chrome Web Store were labelled as malicious. Also, flows of IP addresses that are legitimate components of the network using different ports than usual were labelled as malicious. The remaining flows were labelled as legitimate.

5.2. Dataset Files Description

Table 4 presents the details about all files in the dataset. File names with suffix “L” refer to the data collected when the *Raspberry Pi* was not infected. On the other hand, those marked with “I” contain data collected during infected periods, including legitimate and malicious activities. For each file name presented in Table 4, there are actually three files. The first one is a CSV file with the host data collected (CPU and memory consumption, electric potential difference, CPU temperature, and the number of tasks). The second file is a CSV file that contains IP flow data labelled as malicious or legitimate. The third file is a PCAP file with network packets captured at the *Raspberry Pi*.

Files regarding the experimental environment are provided as well[12]. They consist

[12] The PCAP files and Netflow/Host information files in CSV format are available on http://www.uel.br/grupo-pesquisa/secmq/dataset-iot-security.html as well as the *Raspberry Pi* system image in ISO format

of the image (ISO file) of the *Raspberry Pi* system used in the experiment, with the scripts to capture its host data and tools to create network traffic, the malware samples used in this environment, and other network modules.

Table 4. Files that compose the dataset.

File Name	Description	Profile	Infected by	Malicious activity?
MC_L	One hour of legitimate Netflow/Host Info from profile	MC	No infection	No
SC_L		SC		
ST_L		ST		
MC_I1	One hour of legitimate and malicious Netflow/Host Data from profile	MC	Type 1	Yes
SC_I1		SC		
ST_I1		ST		
MC_I2		MC	Type 2	
SC_I2		SC		
ST_I2		ST		
MC_I3		MC	Type 3	
SC_I3		SC		
ST_I3		ST		

6. Collected Results

This section discusses some aspects of the *Raspberry Pi* behaviour during the dataset generation. In Table 5, the behaviour analyses are presented for different types of infections. The analyses were made by comparing the average of data collected when there were no infections with the average of data collected when the *Raspberry Pi* was infected. For example, we computed the average of CPU usage during the one hour without infection with the *Raspberry Pi* executing the MC profile. Then, we computed the average of CPU usage for the one hour when the *Raspberry Pi* was executing the MC profile and was infected by Hajime. Finally, we compared the computed averages. By doing so, we report whether the average values obtained when the device was infected were higher (↑), lower (↓) or similar (–) to the legitimate data average. To be considered similar, the averages must present a difference lower than 5%. The following features were analysed: CPU usage, memory usage, electric potential difference, number of tasks running, CPU temperature, and packets per second.

Table 5. Analyses performed by comparing legitimate and infected behaviours on multiple scenarios.

Botnet	Profile	Infection Type	CPU Usage	Memory Usage	Potential Difference	Number of Tasks	CPU Temp.	Packets per Second
Hajime	MC	1	↑	↑	↑	↑	↑	↑
Aidra	SC		↑	↓	↑	↑	↑	–
BashLite	ST		↑	↓	↑	↑	↑	–
Mirai	MC	2	–	–	↑	↑	–	↑
	SC		↓	–	↑	↑	–	–
	ST		↓	–	↑	↑	–	–
Multiple Botnets	MC	3	↑	↑	↑	↑	↑	↑
	SC		↑	↑	↑	↑	↑	↑
	ST		↑	↑	↑	↑	↑	↑

For *Type 1* infections, which consist of the combinations Hajime/MC-profile, Aidra/SC-profile and BashLite/ST-profile, the following behaviours were found. In the first combination, the averages computed for the infected period were higher than those of the period without infections. It shows that Hajime's actions caused a variation in multiple features, which could facilitate its detection based on host data analysis. For the infection based on Aidra, the device presented a lower memory usage when compared to the period without infections. On the other hand, all the other features presented higher average values. The decrease in memory usage is a consequence of the way Aidra works since it stops other processes running on the device to gather more resources for itself. The behaviour of Bashlite was similar to Aidra. It is important to observe that each botnet can cause different changes in the features behaviour. Intuitively, the presence of malware on a device was suppose to result in an increase in all observed features. However, it is possible to see that Aidra and Bashlite caused a decrease in memory usage, contradicting this intuition.

For *Type 2* infections, which consist of combinations Mirai/MC-profile, Mirai/SC-profile, and Mirai/ST-profile, the botnet controlled the device to perform DoS attacks targeting different machines on the network. When considering the MC profile, the levels for CPU and memory usage and CPU temperature were similar for infected and non-infected periods. Whereas, the other features presented higher averages for infected periods. The main difference between the SC and ST profiles and the MC profile relied on lower averages for CPU usage during the infected period. This may have occurred because Mirai stops pre-defined processes to free more resources, also searching for specific botnet competitors running in the device. As SC and ST profiles have a lower CPU usage than MC profile (refer to Section 4.2), killing a process may have more impact on the total CPU usage for SC and ST profiles than for the MC profile.

Lastly, in *Type 3* infections, the *Raspberry Pi* was compromised with Mirai, Doflo, Tsunami, and Wroba malware simultaneously. This was done to observe the behaviour of botnets when competing for resources. As expected, all profiles showed higher averages for all features, due to the high amount of malware processes running on the device. An interesting situation observed was that Tsunami, having been installed after Mirai, uninstalled Mirai. So, to avoid this, Mirai was re-installed lastly.

Overall, when the device was infected by some sort of botnet malware, it presented a higher average for most of its features. The electric potential difference and the number of active tasks had higher averages during infected periods for all scenarios. For the other features, there were some variations depending on the scenario. In some cases, CPU and memory usage presented a lower average values during the infected period. For example, memory usage in the Aidra/SC-profile case. This occurs because these malware kill other processes to free resources. Therefore, to detect these botnets, it might be necessary to monitor both the increase and the decrease in average values.

7. Conclusion

The vulnerabilities found in IoT devices have been leading to an increase in research and development of new approaches to protect those devices and their networks. To do that, researchers need access to IoT networks or datasets to evaluate the performance of their approaches.

This paper presented an IoT experimental environment using a Raspberry Pi to generate a labelled dataset that may be helpful for researchers interested in IoT devices protection. The dataset includes data related to the network traffic on the Raspberry Pi and the usage of multiple resources such as CPU and memory. The dataset covers a period when the device was not infected, and other ones when multiple botnet malware infected the device. Three profiles of operation were defined for the Raspberry Pi, aiming to emulate typical domestic IoT devices such as surveillance cameras and multimedia centres. Furthermore, it was provided a first analysis of the changes in the device behaviour when malware compromised it. The dataset and the experimental environment provided here can benefit researchers by giving them data to develop and compare their solutions.

As future work, we intend to create a dataset that includes data collected from all the machines in the network simultaneously, while also including more types of botnets and other IoT devices. Lastly, we intend to use the dataset created here to support the validation of IDSs developed for IoT environments like the one explored in this work.

References

Abdul Kadir, A. F., Stakhanova, N., and Ghorbani, A. A. (2015). Android botnets: What URLs are telling us. In Qiu, M., Xu, S., Yung, M., and Zhang, H., editors, *Network and System Security*, pages 78–91, Cham. Springer International Publishing.

Amaral, J. P., Oliveira, L. M., Rodrigues, J. J., Han, G., and Shu, L. (2014). Policy and network-based intrusion detection system for IPv6-enabled wireless sensor networks. In *Communications (ICC), 2014 IEEE International Conference on*, pages 1796–1801. IEEE.

Angrishi, K. (2017). Turning Internet of Things(IoT) into Internet of Vulnerabilities (IoV) : IoT Botnets. *arXiv preprint arXiv:1702.03681*, pages 1–17.

Atzori, L., Iera, A., and Morabito, G. (2010). The Internet of Things: A survey. *Computer networks*, 54(15):2787–2805.

Bay, S. D., Kibler, D., Pazzani, M. J., and Smyth, P. (2000). The UCI KDD Archive of Large Data Sets for Data Mining Research and Experimentation. *SIGKDD Explor. Newsl.*, 2(2):81–85.

Cervantes, C., Poplade, D., Nogueira, M., and Santos, A. (2015). Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for internet of things. In *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*, pages 606–611. IEEE.

Costa, V. G. T., Barbon Jr, S., Miani, R. S., Rodrigues, J. J. P. C., and Zarpelão, B. B. (2017). Detecting Mobile Botnets Through Machine Learning and System Calls Analysis. *Proceedings of the 2017 IEEE International Conference on Communications (ICC)*, pages 917–922.

Garcıa, S. (2014). *Identifying, Modeling and Detecting Botnet Behaviors in the Network*. PhD thesis, Universidad Nacional del Centro de la Provincia de Buenos Aires.

García, S., Zunino, A., and Campo, M. (2014). Survey on network-based botnet detection methods. *Security and Communication Networks*, 7(5):878–903.

Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660.

Habibi, J., Midi, D., Mudgerikar, A., and Bertino, E. (2017). Heimdall: Mitigating the Internet of insecure things. *IEEE Internet of Things Journal*, 4(4):968–978.

Kolias, C., Kambourakis, G., Stavrou, A., and Voas, J. (2017). DDoS in the IoT: Mirai and Other Botnets. *Computer*, 50(7):80–84.

Mansfield-Devine, S. (2016). DDoS goes mainstream: how headline-grabbing attacks could make this threat an organisation's biggest nightmare. *Network Security*, 2016(11):7 – 13.

Nayyar, A. and Puri, V. (2015). Raspberry Pi-A Small, Powerful, Cost Effective and Efficient Form Factor Computer: A Review. *International Journal of Advanced Research in Computer Science and Software Engineering*, 5(12):720–737.

Nigam, R. (2015). A timeline of mobile botnets. *Virus Bulletin, March*.

Nobakht, M., Sivaraman, V., and Boreli, R. (2016). A Host-Based Intrusion Detection and Mitigation Framework for Smart Home IoT Using OpenFlow. In *Availability, Reliability and Security (ARES), 2016 11th International Conference on*, pages 147–156. IEEE.

Oh, D., Kim, D., and Ro, W. W. (2014). A malicious pattern detection engine for embedded security systems in the Internet of Things. *Sensors*, 14(12):24188–24211.

Peng, T., Leckie, C., and Ramamohanarao, K. (2007). Survey of network-based defense mechanisms countering the DoS and DDoS problems. *ACM Computing Surveys*, 39(1):3–es.

Raza, S., Wallgren, L., and Voigt, T. (2013). SVELTE: Real-time intrusion detection in the Internet of Things. *Ad hoc networks*, 11(8):2661–2674.

Sforzin, A., Mármol, F. G., Conti, M., and Bohli, J.-M. (2016). RPiDS: Raspberry Pi IDS—A Fruitful Intrusion Detection System for IoT. In *UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld, 2016 Intl IEEE Conferences*, pages 440–448. IEEE.

Shiravi, A., Shiravi, H., Tavallaee, M., and Ghorbani, A. A. (2012). Toward Developing a Systematic Approach to Generate Benchmark Datasets for Intrusion Detection. *Comput. Secur.*, 31(3):357–374.

Silva, S. S. C., Silva, R. M. P., Pinto, R. C. G., and Salles, R. M. (2013). Botnets: A survey. *Computer Networks*, 57(2):378–403.

Stavrou, A., Voas, J., and Fellow, I. (2017). DDoS in the IoT. *Computer*, 50:80–84.

Summerville, D. H., Zach, K. M., and Chen, Y. (2015). Ultra-lightweight deep packet anomaly detection for Internet of Things devices. In *Computing and Communications Conference (IPCCC), 2015 IEEE 34th International Performance*, pages 1–8. IEEE.

Thomas, C., Sharma, V., and Balakrishnan, N. (2008). Usefulness of DARPA dataset for Intrusion Detection System Evaluation. *SPIE 6973, Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security*, 6973:69730G–69730G–8.

Whitmore, A., Agarwal, A., and Da Xu, L. (2015). The Internet of Things—A survey of topics and trends. *Information Systems Frontiers*, 17(2):261–274.

Zargar, S. T., Joshi, J., and Tipper, D. (2013). A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. *IEEE Communications Surveys and Tutorials*, 15(4):2046–2069.

Zarpelão, B. B., Miani, R. S., Kawakani, C. T., and de Alvarenga, S. C. (2017). A survey of intrusion detection in Internet of Things. *Journal of Network and Computer Applications*, 84(September 2016):25–37.