IoT-Botnet Detection and Isolation by Access Routers

Christian Dietz*†, Raphael Labaca Castro*, Jessica Steinberger†,
Cezary Wilczak*, Marcel Antzek*, Anna Sperotto† and Aiko Pras†

* Research Institute CODE
Bundeswehr University Munich
Neubiberg, Germany
Email:{Christian.Dietz, Raphael.Labaca,
Cezary.Wilczak, Marcel.Antzek}@unibw.de

†Design and Analysis of Communication Systems
University of Twente
Enschede, The Netherlands
Email:{C.Dietz, J.Steinberger,
A.Sperotto, A.Pras}@utwente.nl

***Abstract*—In recent years, emerging technologies such as the Internet of Things gain increasing interest in various communities. However, the majority of IoT devices have little or no protection at software and infrastructure levels and thus are also opening up new vulnerabilities that might be misused by cybercriminals to perform large-scale cyber attacks by means of IoT botnets. These kind of attacks lead to infrastructure and service outages and cause enormous financial loss, image and reputation damage. One approach to proactively block the spreading of such IoT botnets is to automatically scan for vulnerable IoT devices and isolate them from the Internet before they are compromised and also become part of the IoT botnet. The goal of this paper is to present an IoT botnet detection and isolation approach at the level of access routers that makes IoT devices more attack resilient. We show that our IoT botnet detection and isolation approach helps to prevent the compromise of IoT devices without the need to have in-depth technical administration knowledge, and hence make it viable for customers and end users.**

I. Introduction

In recent years, the world and our daily life became increasingly connected [1]. These interconnections built a network of billions of connected general-purpose devices such as smartphones, PCs, wearable tech or everyday household objects that are referred to as the Internet of Things (IoT). The amount of IoT devices is still increasing and thus Gartner. Inc [1] forecasts an amount of 20.4 billion IoT devices in use worldwide by 2020.

Despite the steady year-over-year growth in the amount of IoT devices, the security of these has been criticized over the past few years [2], [3] as they have little or no protection at software and infrastructure levels [4].

The lack of protection of IoT devices along with poor security update management result in serious security flaws that gain increasing interest and are abused by cybercriminals. In 2016, they compromised approximately 600 000 IoT devices to set up an IoT botnet [5] and launch a large scale cyber attack, namely a Distributed Denial of Service (DDoS), with a traffic peak of 1.1 Tbps [6]. This DDoS attack led to infrastructure and service outages affecting companies such as AirBnB, Amazon, GitHub, PayPal, Netflix, Spotify and Twitter [5], [6].

The main attack techniques used by IoT botnets exploit security vulnerabilities and make use of sophisticated, complex and multi-vector large-scale cyber attacks based on *flooding* and *Water Torture* techniques whereas traditional Botnets make use of *Reflection* and *Amplification*. In particular, the IoT botnet Mirai used 10 predefined attack vectors [7] including generic routing encapsulation (GRE) flood, TCP STOMP and DNS Water Torture technique and mainly performed volumetric, application-layer, and TCP state-exhaustion attacks [6].

Given the quantity of IoT devices, the sophisticated complex attack vectors and the trend of reached attack intensities, DDoS attacks of IoT botnets could lead to enormous financial loss, image and reputation damage [8]. One approach to proactively block the spreading of such botnets is to automatically scan for vulnerable IoT devices and isolate them from the Internet before they are compromised and also become part of the IoT botnet to perform malicious actions.

To overcome critical security issues in IoT devices and make them more attack resilient, we propose an IoT botnet detection and isolation approach. Thus this research focuses to answer the following three research questions: i) How and where can vulnerable IoT devices be protected against IoT botnet infections, in particular Mirai? ii) How can detection and isolation approaches adapt to new derivatives of the Mirai family? iii) How can current barriers (such as costs, technical knowledge, hidden install routines of available detection solution) be eliminated?

Therefore the main contributions of this paper are: i) we provide a structured and detailed literature research resulting in an overview of existing automated scanning and isolations solutions in the context of IoT botnets. ii) we provide an open-source and resource efficient detection and isolation approach at the level of access routers to detect and mitigate the effects of large-scale DDoS attacks launched by IoT botnets. iii.) we provided a reference implementation of our solution that is extensible for other IoT botnets and portable to other router platforms.

978-1-5386-8503-7/18/$31.00 ©2018 IEEE

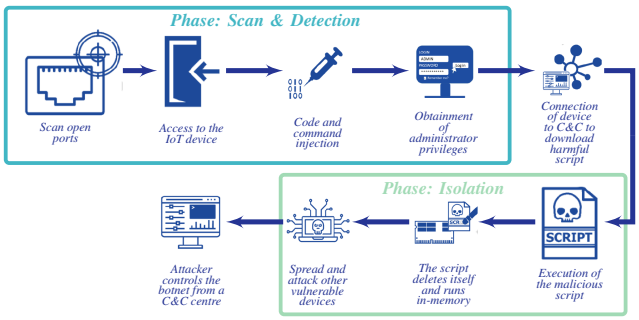

Fig. 1. Life cycle of an IoT botnet partially based on [9]

II. Terminology

In this section, we introduce the terms used throughout the paper and thus support better understanding of our work. First, we define SmartHome and IoT. Next, we generally describe the term botnet and provide a main introduction of a general botnet life cycle. Furthermore, we clarify the specific characteristics of IoT botnets.

A. IoT and SmartHome

The term *IoT* refers to an application domain that integrates different technological and social fields [10]. According to IETF [11], "The Internet of Things is the network of physical objects or "things" embedded with electronics, software, sensors, and connectivity to enable objects to exchange data with the manufacturer, operator and/or other connected devices." We adhere to the definition of IoT presented in [11]. Within this general definition of IoT devices SmartHomes represent an application domain described as follows: "*SmartHomes* are homes equipped with technology that provides the occupants with comprehensive information about the state of their home and allows them to control all connected devices, including remotely" [12]. Examples of such smart devices are cameras, TVs or fridges. Furthermore, ENISA [12] describes SmartHome as "a point of intense contact between networked information technology and physical space. This will create new yet unknown threat and vulnerability models that are the result of bringing together both the virtual and physical context". We also adhere to the definition of SmartHome presented in [12].

B. Botnet vs IoT Botnet

Botnets are networks of devices which have been infected with malware allowing a malicious actor to remotely control them. Although botnets are not new, IoT botnets disrupted the way some attacks are performed over the last few years. That is because unlike regular botnets the goal of the malicious IoT network varies substantially from its counterpart. While botnets were primarily responsible for spam advertised pharmaceuticals, robbing bank credentials and advertisement click fraud [8], IoT botnets have been reported to perform DDoS attacks [13]. That is based on the fact that such IoT devices are constantly online and their combined bandwidth is powerful enough to perform Denials-of-Services while lacking maintenance from security perspective, which allows criminals to leverage powerful attacks by combining a large number of compromised devices. One example of such an IoT botnet is Mirai. However since the source code has been leaked on the Internet many variants evolved including Satori, Okiru, Persirai, Masuta and Puremasuta [14].

In general botnets have a similar behavior regardless of their variant or the malware family they belong to. More specifically, IoT and regular botnets share similar botnet life cycles when compromising new devices.

In this paper, we refer to a bot as a compromised device remotely controlled by an attacker or botmaster.

The life cycle of a typical IoT botnet is visualized in Figure 1 and can be generally summarized into the following 7 stages: i) Scan the (inter)network for open ports on connected devices. ii) Brute-force the discovered ports to gain access to victims. iii) Kill potential competitors on the infected hosts. iv) Create a command and control (C&C) channel with the botmaster. v) Execute and sometimes delete malicious script (runs in memory). vi) Spread through the network by searching for new instances. vii) Launch attacks or perform other malicious actions.

This life cycle was observed in Mirai [15] and in some of its variants like Satori or Persirai [13]. Although the general operation remains the same, some steps might differ such as the initial compromise. For example, instead of scanning the network for open ports on devices, Satori attempts [16] to connect on ports 37 215 and 52 869 using an exploit while Persirai exploits [13] a reported zero-day vulnerability that provides access directly to the password data. Nevertheless, once the IoT-devices are infected, all of them can be used to execute large-scale DDoS attacks (e.g., attack against the French cloud service provider OVH where peaks of 1.1 Tbps were registered using a Mirai botnet [6]).

C. Detection and Defense

Throughout this paper the term *detection* refers to identifying vulnerabilities of connected IoT devices (e.g. open port and default credentials) that can be used to compromise a specific IoT device. We refer to *detection* as a proactively performed detection, which is done without triggering events or user interaction and targets the first stage of the botnet lifecylce (see Figure 1). We consider *defense* as reactive, as it is triggered by the detection output. Further *defense* refers to the automated creation of firewall rules that isolate vulnerable IoT devices from the Internet and thus prevent the device from becoming part of a botnet.

III. Scenario, Requirements & Assumptions

In this Section, we describe the main focus of this work. First, we define the networks in which we are going to place the detection and isolation approach. Second, we define requirements that the detection and isolation approach should fulfill, as they emerged from the scenario described in Section III-A. In the following, we will use these requirements to

978-1-5386-8503-7/18/$31.00 ©2018 IEEE

evaluate the detection and isolation approach in the context of large-scale DDoS attacks and at the level of access routers.

A. Scenario

The focus of this work are SmartHome IoT devices that are connected to the Internet via multiple ISP networks. According to [2]–[4], we assume that the IoT devices have little or no protection at software and infrastructure levels and are thus vulnerable to brute-force attacks against remote services like Telnet or SSH. We only focus on devices that are directly connected to the Internet via an access router and on SmartHomes that have a diverse set of devices of which some appear and disappear (e.g. SmartPhones and Watches) in the network while others are setup once and run for a couple of years (IP-cameras, Printers, SmartLights, TVs, SmartLocks) [12]. Further, we focus on IoT devices that do not check automatically for software updates and vulnerabilities on a regular basis and require a user interaction to implement security patches to fix known vulnerabilities. Besides, the heterogeneous set of IoT devices used in such a scenario, we also focus on a diverse set of manufacturers of which some care about security and will provide updates and patches while others do not offer updates or may even disappear from the market. As described in Section I, a botnet composed of such devices already have been used in large-scale DDoS attacks. Consequently, proactive countermeasures that are independent of the kind of IoT device and its manufacturer are required. To stay independent of device manufacturers, but handle the effects caused by an IoT botnet attack as close as possible to the source, we propose an access router based approach.

B. Requirements

In this Section, we introduce nine requirements that an access router based IoT botnet defense solution should fulfill. The requirements are derived from [17], [18].

Resource efficiency: Today, most access routers can be considered to have low computational power and memory capacities. Therefore an access router based botnet defense solution has to be thrifty with its resource consumption. Furthermore, scanning for vulnerabilities requires the scanned devices to be powered up and connected to the local network. This can be critical e.g. in case of battery powered devices such as wireless security cameras the scanning frequency scope and duration can have a crucial impact on the scanned device. Consequently, the solution has to take care of its impact on the scanned device.

Scalability: The continuous increase of IoT devices per house hold [1] confronts access router based scanning and isolation solutions with numerous potentially vulnerable devices in the next years. As a consequence, the proposed IoT solution must ensure scalability to be able to handle the increasing amount of devices.

Platform independence: As various device manufacturers and product lines are available on the access router market that offer different hardware configurations (e.g. memory capacity and CPU power) a diverse set of access routers with different operating systems (Unix-like) and firewall solutions are in use. Therefore, the proposed solution must ensure platform independence to be used on similar access router platforms.

Extendibility: The IoT defense solution should be able to handle different types of vulnerabilities and should be extensible to be used against newly appearing botnet families. Therefore, the solution should be modular and ensure extensibility due to the use of plugins.

Dynamic device discovery: In SmartHome environments usually different types of devices exist. Some devices are stationary such as security cameras or TVs while others are mobile like Smartphones or Tablets. Furthermore, guests that bring their own device might bring the risk to infect other IoT devices with malware or face a compromise of their own IoT device. Therefore, those IoT devices should be scanned immediately after joining and regularly during the connected time.

Ease of deployment: Usually, access routers are bought by end users with little or no technical knowledge. The deployment and use of an IoT detection and isolation solution must ensure as little user interaction as possible.

Timeliness: To effectively mitigate the effects caused by an IoT botnet, an access router based IoT detection and isolation solution requires frequently updated firmware and security patches. Further, the solution should provide the ability to detect changes on the local networks in short periods of time (e.g. minutes). These changes might be caused by new devices joining the network or known devices leaving and reconnecting to the local network.

Cost-consciousness: To reach a high number of deployed scanning and isolation instances of the IoT botnet detection and isolation approach and thus secured IoT devices, the cost of deploying such solution can be a barrier for some home users as well as for ISP providing remotely managed routers. To increase acceptance by the end-users, the envisioned solution should be usable on popular existing hardware and should be open source to avoid license costs.

Open source: To prevent security by obscurity, increase the acceptance of end users, professionals as well as the access router manufactures, the use of the IoT detection and isolation approach should be open source.

C. Assumptions

The use of an access router based botnet defense solutions depends on the end users willingness and the technical ability to deploy our solution. Today, many ISPs provide remotely managed access routers and hand them to their customers. Therefore, we assume that ISPs are motivated to keep their networks ”clean” and also want to mitigate large-scale attacks before attack traffic threatens their own network. However, we consider incentivizing ISPs and end-users to be beyond the scope of this work. For our research, we always assume that end users and ISPs are motivated to use our solution. Furthermore, we assume that the access router is either based on OpenWRT or a similar Unix-like operating system that is on an up-to-date patch-level and provides enough resources to

run our solution. Finally, we assume that the vulnerable IoT devices are directly connected to the access router via WiFi or standard Ethernet. IoT devices connected via Bluetooth, ZigBee or Z-Wave are considered out of the scope of this work.

IV. Related Work

In this Section, we present works that have been published in the area of IoT botnets, their detection and isolation, and the mitigation of large-scale DDoS attacks.

Zang and Green [19] proposed an IoT defense algorithm to prevent DDoS attacks by making IoT devices in same way intelligent as bots, while preserving a lightweight and inexpensive solution. To understand the difference between a benign and a malicious request, a node analyzes the consistency of the packet content. Although results showed that this approach helps to prevent attacks, it depends on the limited resources of every bot. However, a monitoring node to deal with the extra demand in storage is missing.

A host-based intrusion detection and mitigation (IoT-IDM) was presented by Nobakht et al. [20]. This IoT-IDM addresses malicious activity and blocks intruders from accessing the devices using Software-Defined Networking (SDN) with the OpenFlow protocol. Once an attack in the SmartHome environment is identified at network-level, the IoT-IDM creates policies to block and move infected hosts into quarantine. The IoT-IDM evaluation was only performed using an attack script against smart hues bulbs. In addition, the IoT-IDM is limited to have IoT devices be added manually.

In contrast to IoT-IDM and its analysis of packets, Jerkins [21] modified the leaked code from Mirai and deployed a benign botnet. This botnet uses the same compromise technique to scan and create a list of vulnerable devices and alert law enforcement agencies and IoT device owners about identified vulnerabilities. The source code designed to kill telnet, SSH and HTTP servers remains available on the IoT device with all scanning abilities to avoid further infections and to prevent further propagation. Even though, all attack functions have been removed and given the technical feasibility of such a strategy, the author recognizes the approach infringes laws in many jurisdictions given the lack of geographical boundaries of such botnets.

Further approaches related to our work can be categorized in the three categories: i) vulnerability scanning ii) access router based security monitoring and iii) compromise detection.

First, i) vulnerability scanning approaches are grouped into (a) external scanning and (b) internal scanning approaches. Well-known examples of external vulnerability scanning approaches are the Internet search engine *Shodan* and the web service *Censys.io*. Shodan allows to discover Internet connected devices, their location and their users [22]. In contrast to Shodan, Censys offers Internet wide scanning for connected devices and focuses on deriving and selling long term monitoring data on Internet connected services and devices [23]. Even though, both Shodan and Censys.io can be used for detecting vulnerable devices that are connected to the Internet, they do not cover automated isolation capabilities and their scanning is not optimized for timeliness. Usually, the scanning process of Shodan and Censys.io leave a larger time frame before the vulnerability of a device is reported and closed. Finally, both are commercial services.

Besides external scanning solutions, (b) internal scanning solutions exist. Nmap performs local host based network scanning [24], is free, open source and used for network discovery and security auditing. Nmap uses raw IP packets to determine available hosts on the network, supports all major operating systems and is suitable for large-scale and small networks. Besides Nmap, some commercial antivirus vendors started to sell smartphone or cloud-based IoT security scanners. Examples for such scanners are the internal smartphone scanner *Bullguard IoT Scanner* [25] and *Retina IoT Scanner* [26], which are both commercial. In addition, Kaspersky offers the internal cloud-based *Kaspersky IoT Scanner App* [27] which scans the local network for vulnerable devices using a smartphone. Even though the Kaspersky IoT Scanner comes free of charge, it is not able to automatically defend the detected vulnerable devices.

Second, ii) an access router based security monitoring solution has been proposed by the SPIN project [28]. SPIN focuses on access router level security and monitoring. The SPIN system protects the DNS infrastructure operators and other service providers on the Internet from DDoS attacks and protects users security and privacy in their homes. SPIN focuses on home networks as they are often not as well-managed as corporate ones [28]. Even though, SPIN provides an open source solution which can be used by end-users free of charge, it mainly focuses on detecting and stopping ongoing DDoS attacks. Further, SPIN uses IP flow-based analysis methods and thus focuses on the infection and spreading phases of the botnet life cycle after a compromise happened. Moreover, SPIN is constantly analysing flows or packets and thus constantly consumes available hardware and software resources on the access router.

Third, iii) flow-based detection systems such as SSHCure [29] are tailored specifically to identify SSH brute force attacks and compromise. SSHCure requires to have a flow-based monitoring setup in place and is only designed to detect HTTP or Telnet based attacks. However, SSHCure does not implement any automated defense mechanism.

The aforementioned approaches focus on analyzing the incoming traffic or emulating a benign botnet to prevent DDoS attacks, while further protections of the nodes are still limited. Therefore, an automated approach to detect and isolate vulnerable devices proactively is still missing. Thus, we introduce our novel access router based detection and isolation approach in Section V that combines detection and isolation in one fully automated solution.

V. Access Router based Detection and Isolation

In this Section, we describe the main components of our proposed detection and isolation approach at access level

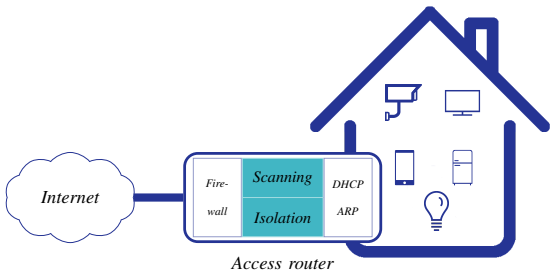

Fig. 2. Components of the IoT botnet detection and isolation approach

routers and how these components interact with each other.

A. Components of the IoT Botnet Defense and Isolation:

Our IoT botnet detection and isolation approach consists of multiple SmartHome networks. These SmartHome networks are connected via an Internet connection, use an access router and various IoT devices as shown in Figure 2. Further, our approach is deployed within the access router and consists of the two main components i) *scanning* and ii) *isolation*.

B. Interactions between main Components:

The IoT devices and the access router interact with each other using DHCP and ARP. The IoT devices request their IP addresses using DHCP that is provided by the access router. Further, the IoT devices and the access router use ARP to resolve IP address to a physical machine address.

C. Detection and Isolation approach

Our IoT botnet detection and isolation approach is multi-phased according to the Figure 1 and consists of the following phases: i) automated detection of vulnerable devices, ii) automated isolation based on the access routers internal firewall, iii) automated update mechanism based on the Common Vulnerability Enumeration (CVE online-service, which is used for an iv) self-optimizing scanning approach.

1) Automated detection of vulnerable devices: In this phase, our approach identifies the subnet address of the local network and scans for connected IoT devices. Therefore, our approach reads the DHCP lease table and the ARP cache to reduce the resource consumption on the access router and the load generated on the network interface. For each IoT device that was found a scan for common vulnerabilities is performed. This vulnerability check is done in two sequential steps. First, the IoT device is scanned for open ports or services. Second, an authentication check is performed based on a predefined list of commonly known credentials for IoT devices. This scanning process is either triggered by changes in the DCHP table or ARP cache or by an updated CVE information that matches discovered devices.

2) Automated isolation based on the access routers internal firewall: In this phase, automated isolation is performed by writing firewall rules to the access routers internal UCI (Unified Configuration Interface) firewall. These firewall rules block any communication to vulnerable services of the IoT devices. At the same time, all blocking actions are reported to the user via an aggregated email and additionally displayed in the solutions own web interface. The web interface provides further guidance to the user on how to deal with the vulnerability. The user can also whitelist devices that should not be isolated by our approach.

3) Automated update mechanism based on a CVE online-service: In this phase, the CVE online-service is queried in regular time intervals (default is hourly) to identify potentially vulnerable services and IoT devices. These queries are filtered by MAC address prefixes of the discovered local IoT devices. Potentially vulnerable services are automatically parsed from the CVE data and mapped to port numbers.

4) Self-Optimizing scanning: The derived port numbers of the previous phase are used for a self-optimizing scanning approach, as a scan is initiated for all potentially applicable devices from the same vendor and specifically adapted to only those ports that are targeted by the newly reported vulnerability information. Therefore, the approach is able to reduce its resource consumption on the access router as well as reducing the network load generated by the re-scanning of devices.

VI. Evaluation

In this Section, we describe the qualitative and quantitative evaluation of the detection and isolation of IoT botnets at access level routers to limit the effects of large-scale DDoS attacks. First, we describe the characteristics of the evaluation criteria. Second, we introduce nine evaluation criteria for the detection and isolation approach. Finally, we present and summarize the results of the evaluation.

A. Qualitative evaluation

In this Section, we perform a qualitative evaluation of the IoT botnet detection and isolation approach. First, we describe the characteristics of the evaluation criteria. Second, we introduce three evaluation criteria for our approach.

1) Evaluation methodology: The IoT botnet detection and isolation approach is evaluated based on the following nine criteria: i) Resource efficiency, ii) Scalability, iii) Platform independence, iv) Extendability, v) Dynamic device discovery, vi) Ease of Deployment, vii) Timeliness, viii) Cost-consciousness and ix) Open source. These criteria derived from the requirements described in Section III-B.

The criterion 'Resource efficiency' describes the ability of the IoT detection and isolation approach to run with a low resource (e.g. CPU or memory) consumption profile and avoid interference with the core services running on the access router and the scanned devices. The criterion 'Scalability' refers to the ability to support an increasing number of IoT devices that appear in the SmartHome scenario within a couple of years. The criterion 'Platform independence' describes the ability of the IoT botnet detection and isolation approach to be deployed on multiple access router platforms. The criterion 'Extendibility' describes the ability to add new algorithms to detect and isolate novel IoT botnet families. The 'dynamic device

discovery' criterion describes the ability to adapt the scanning and detection strategy according to dynamic changes in the network, e.g. devices joining or leaving the local network. The criterion 'Ease of Deployment' describes the ability to install and continuously run the detection and isolation solution with a minimum amount of user interaction and configuration overhead. 'Timeliness' refers to the ability by discovering currently used vulnerabilities and vulnerable IoT devices on the local network as well as isolating them to mitigate IoT botnet deployment and spreading in an appropriate amount of time. 'Cost-consciousness' refers to the criterion to avoid cost of using the IoT botnet detection and isolation approach to maximize the number of potential users and consequently to protect a larger number of vulnerable devices. The criterion 'OpenSource' refers to the public availability of source code, enabling end users to review and reconfigure the actions performed by the IoT botnet detection and isolation approach.

2) Qualitative Evaluation Results: In this paragraph, we present and discuss the results of the qualitative evaluation of our IoT botnet detection and isolation approach.

Scalability: The IoT botnet detection and isolation approach is scalable by design as it is based on Nmap and the Open WRT UCI firewall. Nmap is designed for performance and scanning large networks and is portable to most operating systems [24]. Further, the routers internal UCI firewall abstracts from the IPTables firewall that is designed for performance and is commonly used within linux operating systems [30]. Besides, Nmap and UCI/IPTables, we optimized the discovery of devices within the local network and therefore left enough spare resources. As a result, our IoT botnet detection and isolation approach is able to handle an increased number of devices.

Platform independence: The heterogeneity of access routers used by private end users in their home networks and the different types of operating systems used on these routers requires a platform independent IoT botnet detection and isolation approach that easily integrates within the existing infrastructure. Therefore, the implementation of our approach is based on OpenWRT, Nmap and Python. OpenWRT is a Linux derivate that was designed with a minimum size and resource consumption and is deployable on different hardware platforms as a third-party firmware on access routers of multiple vendors.

Extendibility: Our solution is extensible due to its modular and object oriented structure. The scanning component can be extended using a Plug-In to detect new IoT botnet families. Furthermore, the list of default credentials is also extensible and ensures the use of a different set of credentials of new IoT bot variants.

Dynamic device discovery: As the host discovery process uses the internal ARP table of the access route, the IoT botnet detection and isolation approach can detect changes in the network quickly and scan newly appearing devices.

Ease of deployment: To reach a maximum number of deployments, our approach uses an install process with minimal dependencies. A manual installation only requires Nmap and a Python environment to be installed. Further, OpenWRT offers public packet management services, which make the install of the IoT botnet detection and isolation approach using opkg install in the OpenWRT command line. Next, the user configures the email setting to receive notifications in case vulnerable devices were found and isolated. Optionally, the user can whitelist IoT devices to not be isolated by our approach. Our approach is preconfigured and operates automatically. To ensure the ease of deployment, our IoT botnet detection and isolation approach only requires low technical knowledge and interaction by the user.

Cost-consciousness: To reach a maximum number of end users and ISPs to deploy our approach, it is important that an IoT botnet detection and isolation approach is cost-conscious. Our IoT botnet detection and isolation approach usually does not require additional hardware and licenses. Further, our approach is free of charge and thus is cost-conscious.

Open source: Our IoT botnet detection and isolation approach is publicly available in a GIT repository to allow extension of its functionality and make its functionality transparent for privacy concerned end users.

B. Quantitative Evaluation

In this Section, we perform a quantitative evaluation of the IoT botnet detection and isolation approach. First, we describe the setup of the testbeds. Second, we present the test scenario.

1) Setup of the testbeds: We performed the quantitative evaluation of our approach by using two different testbeds. One testbed focused on Testbed 1 consists of a real hardware. In particular, an access router (TP-Link ArcherC7AC1750v4) running a standard OpenWRT implementation as well as our router based IoT botnet defense solution, one Dahua IP camera (HDBW1320E-W), two emulated TinyCore systems and a regular personal computer representing a common home network setup. Testbed 2 consists of a virtualized network of emulated vulnerable IoT devices. We emulated multiple connected home networks based on VirtualBox, TinyCore, OpenWRT and Vagrant. The setup emulates three different home networks that are connected to an ISP. Testbed 2 consists of a Mirai C&C server, an emulated router representing the ISP network and three TinyCore systems (representing vulnerable IP cameras) that are connected via virtualized OpenWRT based access routers to this ISP network. All three TinyCore systems were emulated with 1 CPU core and 128 MB RAM. The OpenWRT access routers were emulated with 1 CPU core and 256 MB RAM. During our experiments testbed 2 was fully separated from the Internet to prevent unintended spreading of the Mirai malware.

2) Test scenario: The objective of the experiments is to show that a target network with constrained resources benefits from the IoT botnet detection and isolation approach. We show that our IoT botnet detection and isolation approach is usable on constrained resources and able to detect and isolate an IoT device that has been compromised by an IoT botnet. As a result, the IoT device is no longer part of the botnet and does not participate in an ongoing attack. Furthermore, we show

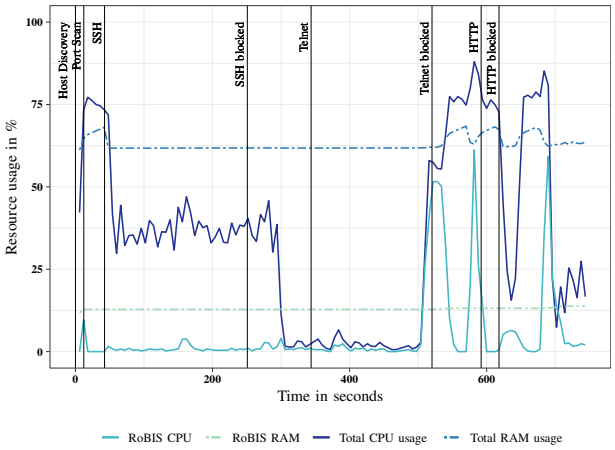

Fig. 3. Resource usage

that the IoT botnet detection and isolation approach prevents the spreading of an IoT botnet infection.

In our *testbed 1* we performed a three step experiment. First, we performed a scanning process to identify all connected IoT devices using the DHCP lease table and the internal ARP table. The benefit of using the DHCP lease table and the internal ARP table is to determine the MAC address and IP address of each connected IoT device and prevent Ping Sweeps. Second, the connected IoT devices are scanned for common vulnerabilities as described in the Subsubsection V-C1. The scanning process includes the protocols SSH, Telnet and HTTP. In the third step, we performed an automated isolation of the vulnerable IoT devices by writing firewall rules. This three step experiment was performed five times to prevent a biased result. All experiments performed scanning and brute-force attacks on real life hardware that is commonly used in SmartHome scenarios. In this testbed, we simulated how a real end user uses the system. Further, we test the installation and initialization process of our IoT botnet defense and isolation approach and evaluate its resource consumption under real-life conditions.

In our *testbed 2* we emulated an ISP network that interconnects three home networks to evaluate our IoT botnet detection and isolation approach against real Mirai infections. Therefore, we set up our testbed 2 with three IoT devices according to Table I. For example, in our first experiment all IoT devices did not make use of our IoT botnet detection and isolation approach whereas in our third experiment the IoT device 1 and the IoT device 3 used our IoT botnet detection and isolation approach. Similar to the experiments of our testbed 1, all experiments within testbed 2 were performed five times to prevent a biased result.

3) Quantitative evaluation results: In this paragraph, we present and discuss the results of the quantitative evaluation of the IoT botnet detection and isolation approach.

Figure 3 summarizes the results that were derived from testbed 1. The different phases of our IoT botnet detection an isolation approach are visually represented by the vertical lines in Figure 3, e.g. the first line represents the initial host discovery, the second vertical line represents the start of the scanning for open ports, the third line represents the start of the bruteforce attack on the SSH service on port 22 while the fourth line represents the time when the vulnerable SSH service was discovered and blocked using Mirai's list of default credentials and the access routers internal firewall. The subsequent vertical lines represent the same processes for the telnet and the http service. Regarding the requirements from Section III-B we derived the following results:

Resource efficiency: To evaluate the efficiency of our approach, we run our experiments using the Linux real-time process monitor program *top* that is pre-installed on many linux/unix operating systems. *Top* provides an overview of used system resources and running processes. This evaluation was performed on testbed 1. As shown in Figure 3 our IoT botnet detection and isolation approach uses around 12.5% RAM while running where as in total 63% of the available RAM on the access router are in use. As a result, our IoT botnet detection and isolation approach requires around 16 MB RAM on average.

Further, our IoT botnet detection and isolation approach consumes between 1% and 5% of CPU usage on average while running as shown in Figure 3. However, the CPU peaks at the beginning of each host's scan throughout the five experiments. Our experiments reported that the authentication of SSH is one of the reasons why the CPU load is high, whereas Telnet and HTTP checks turn the CPU usage back to normal.

Timeliness: To support timeliness in detection and isolation of vulnerable IoT devices, our IoT botnet detection and isolation approach implements a regular and self-optimized scanning approach based on external CVE data. Using our *testbed 1*, we found that the automated detection of vulnerable devices took 12.17 seconds, the port scan took 30.30 seconds on average. As the scan of the SSH, Telnet and HTTP protocol is performed sequentially and each found vulnerability is blocked immediately, the scanning and blocking process of SSH, Telnet and HTTP took 582.17 seconds on average. Timely reaction to newly discovered vulnerabilities is implemented on an hourly basis. Furthermore, using the internal ARP cache allows timely detection of devices joining the network.

In addition, Table I summarized the results of our experiments using *testbed 2*. The results show that the use of our IoT botnet detection and isolation approach systematically protects SmartHome networks and vulnerable IoT devices are effectively protected against Mirai infections. For example, the IoT devices within the first experiments did not use our IoT botnet detection and isolation approach and were able to be compromised by the Mirai botnet. In contrast, the third experiment showed a partial use of our IoT botnet detection and isolation approach and as a consequence only the unprotected IoT device was infected with the Mirai malware.

TABLE I
INFECTIONS WITH AND WITHOUT OUR SOLUTION DEPLOYED

	Test setup			Result		
Test	**Device 1**	**Device 2**	**Device 3**	**Device 1**	**Device 2**	**Device 3**
1	not protected	not protected	not protected	compromised	compromised	compromised
2	protected	protected	protected	no compromise	no compromise	no compromise
3	protected	not protected	protected	no compromise	compromised	no compromise
4	not protected	protected	not protected	compromised	not compromised	compromised

VII. CONCLUSIONS & FUTURE WORK

IoT botnets pose a serious threat to the Internet infrastructure and services. One approach to detect and isolate IoT botnets focuses on access routers. In this paper, we introduce an IoT botnet detection and isolation approach on access level routers that facilitates the automated detection of vulnerable IoT devices, the isolation based on the access routers internal firewall, the update mechanism based on a CVE online-service and provides a self-optimizing scanning. We have shown that our approach located at access level router proactively protects against IoT botnet infections and prevents the IoT botnet Mirai from further propagation. The main advantage of our IoT botnet detection and isolation approach over existing approaches is that it easily integrates with the existing infrastructure and is easy to deploy. Based on our qualitative and quantitative evaluation, our IoT botnet detection and isolation approach constitutes a viable solution to be used by end users with different level of technical knowledge.

As future work we plan to support and check routers for vulnerabilities to create an additional layer. This layer is intended to prevent router exploitation as a propagation vector. Furthermore, we plan to detect possible MAC or IP address changes and other attacks in future versions of our approach.

ACKNOWLEDGMENT

We thank the chair for Communication Systems and Network Security at the Bundeswehr University Munichen, Prof. Gabi Dreo Rodosek, and the research institute CODE for providing feedback and hardware facilities for our research. Further, this work was partly supported by the NWO D3 Project.

REFERENCES

[1] R. van der Meulen, "Gartner says 8.4 billion connected "things" will be in use in 2017, up 31 percent from 2016." [Online]. Available: https://www.gartner.com/newsroom/id/3598917

[2] H. Zhang, "How to disinfect and secure the internet of things," *Network Security*, vol. 2016, no. 9, pp. 18 – 20, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1353485816300903

[3] S. Mansfield-Devine, "Securing the internet of things," *Computer Fraud & Security*, vol. 2016, no. 4, pp. 15 – 20, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1361372316300380

[4] M. Hung, "Leading to iot." [Online]. Available: https://www.gartner.com/imagesrv/books/iot/iotEbook_digital.pdf

[5] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou, "Understanding the mirai botnet," in *26th USENIX Security Symposium (USENIX Security 17)*. Vancouver, BC: USENIX Association, 2017, pp. 1093–1110. [Online]. Available: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis

[6] E. Bursztein, "Inside the infamous mirai iot botnet: A retrospective analysis." [Online]. Available: https://blog.cloudflare.com/inside-mirai-the-infamous-iot-botnet-a-retrospective-analysis/

[7] C. Herberger, M. Groskop, B. Zilbermann, S. Shitrit, E. Levy, Y. Ben-Ezra, N. Ilani, D. Smith, P. Geenens, N. Pariente, and S. Trimble, "Global application & network security report 2017-2018." [Online]. Available: https://www.radware.com/pleaseregister.aspx?returnurl=ea128fa5-9f3f-43aa-b09c-cea5baba03ad

[8] C. G. J. Putman, Abhishta, and L. J. M. Nieuwenhuis, "Business model of a botnet," in *Proceedings of the 26th Euromicro International conference on Parallel, Distributed, and Network-Based Processing*, ser. PDP '18, vol. abs/1804.10848. IEEE Press, 2018. [Online]. Available: http://arxiv.org/abs/1804.10848

[9] European Union Agency for Network and Information Security, "Baseline security recommendations for iot." [Online]. Available: https://www.enisa.europa.eu/publications/baseline-security-recommendations-for-iot

[10] Roberto Minerva, Abyi Biru, Domenico Rotondi, "Towards a definition of the internet of things (iot)." [Online]. Available: https://iot.ieee.org/images/files/pdf/IEEE_IoT_Towards_Definition_Internet_of_Things_Revision1_27MAY15.pdf

[11] IETF, "The internet of things." [Online]. Available: http://ietf.org/topics/iot/

[12] European Union Agency for Network and Information Security, "Threat landscape and good practice guide for smart home and converged media." [Online]. Available: https://www.enisa.europa.eu/publications/threat-landscape-for-smart-home-and-media-convergence

[13] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "Ddos in the iot: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[14] L. G. A. Rodriguez, J. S. Trazzi, V. Fossaluza, R. Campiolo, and D. M. Batista, "Analysis of vulnerability disclosure delays from the national vulnerability database," in *Workshop de Segurança Cibernética em Dispositivos Conectados (WSCDC_SBRC)*, vol. 1, 2018.

[15] G. Kambourakis, C. Kolias, and A. Stavrou, "The mirai botnet and the iot zombie armies," in *Military Communications Conference (MILCOM), MILCOM 2017-2017 IEEE*. IEEE, 2017, pp. 267–272.

[16] Netlab, "Warning: Satori, a mirai branch is spreading in worm style on port 37215 and 52869," *http://blog.netlab.360.com/warning-satori-a-new-mirai-variant-is-spreading-in-worm-style-on-port-37215-and-52869-en/*, 2017.

[17] European Union Agency for Network and Information Security, "Security and resilience of smart home environments." [Online]. Available: https://www.enisa.europa.eu/publications/security-resilience-good-practices

[18] S. Jaiswal and D. Gupta, "Security requirements for internet of things (iot)," in *Proceedings of International Conference on Communication and Networks*. Springer, 2017, pp. 419–427.

[19] C. Zhang and R. Green, "Communication security in internet of thing: preventive measure and avoid ddos attack over iot network," in *Proceedings of the 18th Symposium on Communications & Networking*. Society for Computer Simulation International, 2015, pp. 8–15.

[20] M. Nobakht, V. Sivaraman, and R. Boreli, "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*. IEEE, 2016, pp. 147–156.

[21] J. A. Jerkins, "Motivating a market or regulatory solution to iot insecurity with the mirai botnet code," in *Computing and Communication Workshop and Conference (CCWC), 2017 IEEE 7th Annual*. IEEE, 2017, pp. 1–5.

[22] Shodan.io, "Shodan." [Online]. Available: http://www.shodan.io

[23] Censys.io, "Censys." [Online]. Available: https://censys.io

[24] NMAP, "Nmap." [Online]. Available: https://www.nmap.org

[25] Bullgurad, "Bullgurad iot scanner." [Online]. Available: https://iotscanner.bullguard.com

[26] BeyondTrust, inc, "Retina iot riot scanner." [Online]. Available: https://www.beyondtrust.com/resources/data-sheet/retina-iot-riot-scanner/

[27] Kaspersky IoT Scanner App, "Kaspersky iot scanner app." [Online]. Available: https://www.kaspersky.com/blog/kaspersky-iot-scanner/18449/

[28] Christian Hesselman, "Spin:a user-centric security extension for in-home networks." [Online]. Available: https://www.sidnlabs.nl/a/weblog/spin-a-user-centric-security-extension-for-in-home-networks

[29] R. Hofstede and L. Hendriks, "Unveiling sshcure 3.0: Flow-based ssh compromise detection," in *International Conference on Networked Systems (NetSys 2015), Demo Session, Cottbus, Germany*, 2015.

[30] OpenWRT, "Openwrt-firewall configuration." [Online]. Available: https://wiki.openwrt.org/doc/uci/firewall