Estimation of the Average Age of IoT Malware Exploits Using Static and Dynamic Analysis

Mr. MARLAPUDI MURALI[1],Dr. B. NAGESWARA RAO[2] ,Mr. D. RAHULKHANNA[3]
PROFESSOR[2] ,ASSISTANT PROFESSOR[1,3] DEPARTMENT OF ECE,
SWARNANDHRA COLLEGE OF ENGINEERING AND TECHNOLOGY, NARASAPUR

ABSTRACT

There are a lot of security holes in the many software and hardware components that make up the Internet of Things (IoT). According to earlier research, the first attempts at infection can occur as little as a few minutes after an Internet of Things device is connected to the Internet. Unfortunately, details on the evolution of attack vectors, such as the vulnerabilities being targeted, the changes in functionality, and the duration of exploit use, are still lacking. Building and launching IoT networks with more assurance may be possible with a deeper understanding of these challenges. We provide the first longitudinal study of malware assaults on the Internet of Things (IoT) by analyzing 17,720 samples collected from three separate sources between 2015 and 2020. After extracting exploits from these binaries using static and dynamic analysis, we analyze them along four dimensions: (1) the evolution of infection vectors, (2) the duration of an exploit's use, the age of the vulnerability, and the time required to exploit it, (3) the nature and functionality of the exploits, and (4) the manufacturers and types of IoT devices that have been compromised. Our descriptive analysis reveals many trends: Malware for the Internet of Things has evolved from using brute force attacks alone to including a suite of vulnerabilities tailored to individual devices. Once an exploit has been built, it is seldom forgotten. Modern binaries still take use of (very) old vulnerabilities. New exploits are being developed for vulnerabilities that have been known for a long time. We find that the average time to exploit once a vulnerability is disclosed is around 29 months, which is much longer than malware that targets other settings.

KEYWORDS

Internet of Things, malware, exploits, vulnerabilities, infection vectors, static analysis, and dynamic analysis

INTRODUCTION

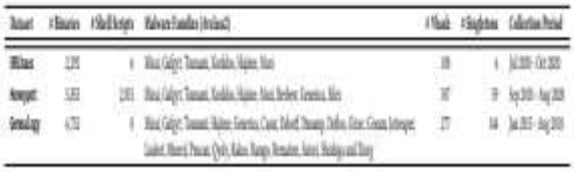
There are new opportunities for cybercriminals brought about by the proliferation of Internet of Things (IoT) devices like IP cameras and smart home appliances, which provide us with innovative services. There has been a concerning increase in the amount of hacked electronic devices [4]. Internet of Things (IoT) vulnerabilities continue to be the principal vector for infection, even if user involvement and social engineering have becoming more prevalent attack vectors for desktop and mobile devices [3]. While we have a better grasp of the capabilities and families of IoT malware [11, 61], we still don't know much about how attackers chose which vulnerabilities to exploit. The number of vulnerabilities related to the Internet of Things (IoT) is growing at the same pace as the overall number of vulnerabilities, going from a dozen or so in 2010 to more than 500 in 2019 [6]. Then which of these vulnerabilities is the target? Is it common for different types of malware to go for the same security holes? How much time elapses between the disclosure of a security hole and its subsequent exploitation? For what amount of time will they not move on from fixing one security hole? We have observed that, when targeting PCs and servers, attackers often target software versions that are only one version behind the most current patch version [1, 51, 58]. This tendency, however, is not expected to continue since patching becomes more complicated inside the IoT ecosystem [55]. We currently lack a solid systemic understanding of the overall vulnerability targeting strategy in the IoT malware ecosystem, even though prior research has examined the exploit code used by certain malware families at different times [4, 14, 24]. For the most comparable previous work, see Alrawi et al.[3]'s concurrent inquiry. A total of 25 vulnerabilities discovered in 2019 were examined in the study, which examined a sizable sample of IoT malware binaries collected that year. In the end, our study confirms some of the conclusions drawn from this paper. Beyond only monitoring the evolution of vulnerabilities, we also monitor the evolution of exploits among malware families over a five-year period. For some time now, we have been keeping tabs on 63 attacks and the 68 vulnerabilities they target. This allows us to demonstrate previously unseen patterns in the time-to-exploit, vulnerability-exposure, and exploit-lifespan metrics.

CONTENT THAT IS RELEVANT

Most research on the security of the Internet of Things has concentrated on developing solutions for devices with limited resources, but very little has looked at the vulnerabilities and security of already-deployed IoT devices. In their investigation of Internet of Things (IoT) vulnerabilities, Feng et al. [21] drew on a variety of publicly available resources, such as vulnerability and exploit databases, discussion forums, email lists, and blogs, to propose better mitigation strategies. By combining similar data sets with machine learning, Lebowski and Piotrowski [6] were able to classify IoT system vulnerabilities according to the CVE. By excluding publicly accessible data and instead concentrating on a subset of IoT installations in residential areas, Alawi et al. [2] performed the first empirical investigation of the security features and flaws of commercially available IoT devices. Recent studies have studied attacks by researching IoTmalware[3,5,10,14,16], while earlier research focused on protective measures. Virus Total [50] samples, publicly accessible threat intelligence data (e.g., Cyberbooks [15]), or honeypots (e.g., Hotpots [43]) are the mainstays of these studies' methods for detecting IoT malware. There has been prior research on IoT malware; for example, Hamulate and Razali [23] examined the most publicized CVEs. Their research demonstrated that malware specifically designed for the Internet of Things targets security holes that might be exploited to sneak into devices unnoticed by the user. Alawi et al. [3] recently examined a dataset of 166,000 IoT malware samples collected in 2019 to get a better understanding of the code duplication and evolution of different families of IoT malware. Just like us, the writers compared and contrasted the different malware strains using static and dynamic analytics. In order to characterize the evolution of different sorts of exploits over a longer period of time, our study expands upon their initial work in four ways. Compared to Alawi et al. [3], who only looked at 25 vulnerabilities, we analyze 68 vulnerabilities (not including hard-coded credentials) that are present in the binaries. This allows us to: (1) cover binaries from 2015 to 2020, which is a much larger time frame; (2) use a combination of static and dynamic analysis to extract exploits. The precise timing of the first industry reports on the vulnerabilities they discovered eluded them. They aren't seeing this in their own data, and they aren't keeping tabs on it over time, either (like the time binary attacks are utilized). Our findings provide the framework for understanding the motivations and mechanisms behind persistent exploit use, even in the face of patch notifications.

METHODOLOGY

We may learn more about the vulnerabilities and devices that are targeted over time by tracking the changes in the exploit code utilized by IoT malware. To find exploit code, binaries may be analyzed statically or dynamically. While automatic static analysis is more reliable and complete, manual static analysis, also known as reverse engineering, requires more labor but is more resistant to code obfuscation and packing. However, although auto-mated dynamic analysis is scalable and capable of handling packing, it suffers from a lack of comprehensive coverage of ex-plaits. Our method combines the best features of static assessments done by people with those of dynamic analyses done by computers. Once we have these first results, we will enhance them by searching a binary repository covering the three years before our binaries were collected for particular vulnerabilities. We provide a high-level overview of our method in Figure 1. Table 1: Malware families covered, number of samples obtained per dataset, and time period of collection.



Dataset	# Binaries	# Shell Scripts	Malware Families (AvClass)	# Vuln	# Singletons	Collection Period
Uraeus	2,298	6	Mirai, Gafgyt, Tsunami, [illegible]	108	4	Jul 2020 - Oct 2020
Honeypot	5,855	2,815	Mirai, Gafgyt, Tsunami, [illegible]	[illegible]	39	Sep 2018 - Aug 2020
Genealogy	[illegible]	8	Mirai, Gafgyt, Tsunami, [illegible]	[illegible]	14	Jan 2015 - Aug 2020

Data Collection

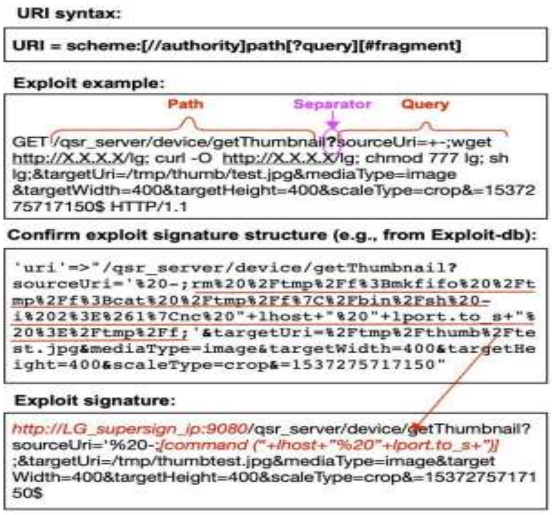
Weaponized Code for Internet-of-Things Devices

In order to create exploit signatures, we first collect samples from two separate sources: Uraeus and a honeypot. Then, we compare these samples to a longer-term dataset called Genealogy. One central database that gathers malware spreading connections is Uraeus [56]. Our dynamic and static testing now includes a library of newly compiled binaries that we retrieved from this repository. From July 2020 through October 2020, we were able to retrieve a daily file that included the URLs of all recorded binaries along with other pertinent details such as file type. We only gathered URLs for "Executable and Linkable Format" (ELF) files since our study is focused on malware that infects the Internet of Things. We routinely employed a script to obtain those data throughout the course of those four months. In all, 2,298 binaries were obtained for a wide range of platforms and CPU architectures, including Renesas SH, Motorola 68000, SPARC, Intel 80386, ARM, PowerPC, MIPS, ARC Cores Tangent-A5, and AMD x86-64.

Included in our investigation is x86-based malware, as previous research has shown to be common in IoT devices [37]. We looked over the 6,298 files thoroughly and found that they just downloaded binary files and did not use any propagation methods, even though they were shell scripts. Between September 2018 and August 2020, we collected 5,855 MIPS binaries using the Hotpots [43] honeypot. Hotpots are a hybrid of two types of honeypots: one with limited interaction and one with strong engagement. Several network services use the low-interaction honeypot as a proxy. These include Telnet, HTTP front-ends, CPE WAN Management Protocol (CWMP), a backdoor of Natis routers, and the remote access setup service of several IP cameras. A router, an IP camera, and two WIFI storage devices make up the high-interaction honeypot's quartet of bare-metal Internet of Things components. The honeypot is now associated with about 130 IP addresses from Japan. The 2,815 files captured by Hotpots but not in ELF binary format were also part of the dataset we were able to gather. When used as shell scripts in a protected setting, 2.608 included the capabilities of downloaders using wet, curl, etc. Out of the remaining 207 files, 10 were identified as Python scripts, 2 as Perl scripts, and the other 195 as plain ASCII texts, not scripts. Just one of the Python scripts contains vulnerabilities, according to our follow-up evaluation of these ten programs and two Perl scripts that we conducted in a sandbox. Because of this, we will only look at the binary samples going forward.

CONSUME THE LANDSCAPE

The results of our research on exploits and vulnerabilities in IoT malware are shown below. The results from all three datasets are summarised in Table 3. We discovered a total of 64 infection vectors, the majority of which involve brute-forcing hard-coded credentials, and 63 distinct exploits that aim to attack 68 vulnerabilities. Table 3 shows the frequency with which each vulnerability was found in each dataset in the last column. The vulnerabilities, exploits, and device makers are all identified. The table excludes two sets of Uraeus binaries because they did not include vulnerabilities. Twenty-seven of the 108 (or 25%) binaries only had brute-force credentials hard-coded. A



URI syntax:

```
URI = scheme:[//authority]path[?query][#fragment]
```

Exploit example:

```
GET /qsr_server/device/getThumbnail?sourceUri=+-;wget
http://X.X.X.X/lg; curl -O  http://X.X.X.X/lg; chmod 777 lg; sh
lg;&targetUri=/tmp/thumb/test.jpg&mediaType=image
&targetWidth=400&targetHeight=400&scaleType=crop&=15372
75717150$ HTTP/1.1
```

Confirm exploit signature structure (e.g., from Exploit-db):

```
'uri'=>"/qsr_server/device/getThumbnail?
sourceUri='%20-;rm%20%2Ftmp%2Ff%3Bmkfifo%20%2Ft
mp%2Ff%3Bcat%20%2Ftmp%2Ff%7C%2Fbin%2Fsh%20-
i%202%3E%261%7Cnc%20"+lhost+"%20"+lport.to_s+"%
20%3E%2Ftmp%2Ff;'&targetUri=%2Ftmp%2Fthumb%2Fte
st.jpg&mediaType=image&targetWidth=400&targetHe
ight=400&scaleType=crop&=1537275717150"
```

Exploit signature:

```
http://LG_supersign_ip:9080/qsr_server/device/getThumbnail?
sourceUri='%20-;{command ("+lhost+"%20"+lport.to_s+")}
;&targetUri=/tmp/thumbtest.jpg&mediaType=image&target
Width=400&targetHeight=400&scaleType=crop&=15372757171
50$
```

Figure 2: Example of a signature we generated for an exploit against CVE-2018-17173 [20, 40].

The second group of eleven binaries (about 10% of the total) included no infection vectors and solely routines for receiving commands from a command and control (C2) server and launching attacks. Among the many attack vectors described as being implemented using these commands are UDP floods, SYN floods, ACK floods, TCP floods, UDP floods, VSE floods, DNS floods, GRE IP floods, GRE Ethernet floods, and HTTP floods [29]. The individuals who had these binary files were Tsunami, Ordos, Hajime, and Singletons alike. There were 256 vulnerabilities in the remaining 65%, or 70 binaries, that targeted internet services, namely those that relied on HTTP GET and POST requests. Table 3 shows the six types of vulnerabilities that were identified based on the descriptions in NVD or Exploit-DB: RCE, backdoors, CIA, buffer overflow, WAF bypass, and brute force. At least 55.62 percent of the vulnerabilities were infected by Remote Code Execution (RCE). Similarly, the honeypot dataset has 53.65% of exploits and the Uraeus dataset has 55.9%, making Remote Code Execution (RCE) the most exploited vulnerability type overall. All three datasets that used the same vulnerabilities showed that CIA was the most common infection vector, accounting for 56.25 percent of the total infections.

Utilize Your Life Expectancy as Much as Possible

There has been a steady increase over time in both the amount and frequency of exploits for vulnerabilities in the Internet of Things (IoT). We searched the Genealogy dataset for matches to exploit signatures (2015–2018) using the method described in Section 2.3. Seventeen attack signatures (representing sixteen vulnerabilities) were found to be compatible with the Genealogy dataset out of a total of sixty-four. Figure 3 shows one potential rationale for the low prevalence of vulnerabilities in previous binaries. After the collection time for the Genealogy dataset concluded in August 2018, 32 vulnerabilities, or 47% of the total, were made public. Despite advancements, fifteen assaults targeting previously fixed vulnerabilities are absent from the Genealogy dataset. This suggests that developers of more recent malware are selecting vulnerabilities that were discovered a long time ago, supposing the Genealogy dataset is representative of the age in issue. We found matches in 5,421 samples (or 80%) out of 6,752 binaries. None of the samples match the remaining 20%, which may be because there are a lot of packed and end coded samples in this dataset, as the developers of the repository concede. In Section 6, we go into further detail about this limitation. The exploits' lives, or the duration between an exploit's first discovery and its last detection, were examined using the longer time periods available in the Genealogy dataset. The time it takes for a vulnerability to get from being published to an exploitable binary being visible in the wild is another metric we examine. All binaries' "first seen" dates were collected from Virus Total.

Table 2: Number of hits (occurrence), exploits, and vulneraryabilities per year

Year	# Occurrences	# Exploits	# Vulnerabilities
2017	46	10	8
2018	727	15	15
2019	376	26	27
2020	1,855	58	63

By comparing the dates of vulnerability disclosure (black X) and exploit code publication with the number of occurrences of the exploit in binaries (coloured dots) (red circle), Figure 2 shows the lives of exploits. A number of CVEs, including CVE-2013-7471, had their IDs collected much before the 2019-

06-11 official publishing date. Four other CVEs (CVE-2020-1956, CVE-2018-20841, CVE-2019-2725) had publishing dates that did not correspond with the CVE ID. The exploit's publishing date may occasionally precede the relevant vulnerability's publication date, and this might be the reason why. The official vulnerability disclosure date was followed by twenty exploits, albeit the dates were sometimes very close together, which might be due to errors in the underlying data rather than a true sequence of events. Development of malware families is also seen in Figure 2. All binaries released in 2015 and 2016 depend only on text file information for brute force. A total of 4,091 out of 5,421 were binaries. Perhaps this was further reinforced by the November 2016 [22] release of the Mirai code. Mirai was a huge deal for the Internet of Things malware community since it was the first botnet to successfully collect infected devices in the millions. It would seem that other versions and families adopted Mirai's practice of using hard-coded credentials based on brute force attacks. From that point on, binary brute force remained in use until the most recent data became accessible.

DISCUSSION

The number of vulnerabilities that Mirai might exploit rapidly increased once it started attempting to infect computers by brute-forcing default or weak passwords. Our findings are more alarming than those of the last research [3], which identified 25 vulnerabilities. With the discovery of 68 new targeted vulnerabilities and 68 new exploits, we demonstrated that the development of exploits is gaining pace. Since 2017, annual attacks on specific vulnerabilities and exploits have roughly doubled. The tactics and thought process of the criminals are also illuminated by our investigation. Around half of all vulnerabilities are used for at least two years, whereas the other half are used briefly and then left alone. The second one can mean that you often mess up while you're learning new things. If the exploit code successfully recruits bots, the attacks will go on. The likelihood of the exploit code being disseminated throughout other families and groups increases as an attack continues. After then, the flaw is relentlessly pursued by cybercriminals for a long time. It is fascinating to see that attackers intentionally target vulnerable areas. Malicious actors targeting Internet of Things devices are more likely to take advantage of old vulnerabilities than those targeting desktop operating systems or server software. Scientists have found that the second group exploits the most current vulnerabilities by targeting software versions that are only one patch release behind [58]. A single day may pass between a vulnerability's publication and the first detection of a binary exploiting that vulnerability (for example, "Exploit Wednesday" after Microsoft's "Patch Tuesday"), or several months may pass for a handful of high-profile attacks such as Wannacry and Not-Petya [18, 63]. Since most of the compromised computers are running the very latest version of software, this strategy makes sense.

CONCLUSION

To investigate the dynamics of the Internet of Things (IoT) malware ecosystem, we performed the first longitudinal measuring study to use several viewpoints. From 17,720 binaries, spanning 26 different types of IoT malware, we were able to extract 63 unique vulnerabilities using static analysis, dynamic analysis, and signature matching. Our research shows that the ecosystem has expanded its emphasis beyond brute-force attacks to include a wide range of device-specific vulnerabilities. When it comes to innovation and progress, the Mirai family has been at the forefront from its inception in 2016. This originally appeared in Mirai, where the majority of security flaws were discovered. Malware targeting the Internet of Things (IoT) became more sophisticated as the number of devices and protocols compromised increased. The pace of change is accelerating, with an annual rise in exploits and targeted vulnerabilities since 2017. Once an exploit has been built, it is seldom forgotten. Even in the most recent binaries, you may find a lot. The average duration of our expeditions is 38 months, but they might extend over five years. Attackers are free to target vulnerabilities of any age. The average time it takes for an exploit to make its initial appearance in a binary is 29 months, but this may vary greatly depending on the exploit's window of opportunity. That has nothing in common with the patterns of viruses that target servers and PCs. Assuming this novel method of targeting the IoT is legitimate, our data shows that the targeted devices are seldom, if ever, updated. Thus, the window of opportunity to attack a vulnerability gradually shrinks as time passes. The ease of constructing exploit vectors and the device's installation base are more important to attackers than the age of the vulnerability. They will have a lengthy lifespan after creation. It is clear from our study that the many vulnerabilities in the IoT ecosystem are being taken advantage of by attackers. These vulnerabilities include the ecosystem's lack of patching and the diverse range of devices and manufacturers, which is estimated to exceed 14,000 distinct firms [30]. Every single gadget has its own special ways of avoiding malware, and because of this, there are a lot of possible targets. Quite a few clients, internet service providers, and industrial enterprises will feel the effects of our findings.

REFERENCES

Allodi, Luca. 2017. Commercial Considerations of Exposure, Trafficking, and Exploitation. The ACM SIGSAC Conference on Computer and Communications Security is the source for this information.

[2] in A group including Omar Alrawi, Chaz Lever, Manos Antonakakis, and FabianMonrose. The year 2019. Proof of Concept: Assessing the Safety of Internet of Things (IoT) Installations in Residential Settings. S&P is the IEEE Symposium on Security and Privacy's proceedings. Accessed at https://doi.org/10.1109/SP.2019.00013.

the third Alrawi, Omar, Lever, Charles, Valakuzhy, Court, Kevin, Snow, Monrose, Fabian, and Antonakakis, Manos. The year 2021. The Life Cycle: A Comprehensive Analysis of the Internet of Things Malware Lifecycle. The USENIX Security 21 is the 30th annual USENIX Security Symposium. 3505–3522, USENIX Association.

[4] The authors of the study include Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, ElieBursztein, Jaime Cochran, ZakirDurumeric, J Alex Halderman, Luca Invernizzi, MichalisKallitsis, and others. year 2017. The Mirai Botnet: A Comprehensive Overview. Included in the USENIX Security Symposium proceedings, pages 1093–1100.

[5] In the group of Afsah Anwar, Jinchun Choi, AbdulrahmanAlabduljabbar, HishamAlasmary, Jeffrey Spaulding, An Wang, Songqing Chen, DaeHunNyang, AmroAwad, and David Mohaisen. The year 2021.Deciphering IoT Malware via Static Artifact Analysis at Endpoints 2. The preprint may be found at arXiv:2103.14217 (2021).

Grzegorz J. Blinowski and PawełPiotrowski are the authors of chapter 6. 2020. Cyber Vulnerability Evaluation for Internet of Things Systems.With references to pages 82–93 of the DeepCoS International Conference on Dependability and Complex Systems.

The author(s) listed are BrennenBouwmeester, Elsa Rodríguez, Carlos Gañán, Michel van Eeten, and Simon Parkin. The year 2021. "The Thing Doesn't Have a Name": Gaining Insight into Smart Home Security from New Interventions in the Real World. Section SOUPS, USENIX Symposium on Usable Security and Privacy.IT Professionals at USENIX Ltd.

OrtçunÇetin, Carlos Gañán, LisetteAltena, Takahiro Kasama, Daisuke Inoue, Kazuki Tamiya, Ying Tie, Katsunari Yoshioka, and Michel van Eeten are the authors of the cited work. The year 2019. Evidence from Real-World Efforts by ISPs and Consumers to Remove Mirai: Cleaning Up the Internet of Evil Things. This year (2019). DOI: 10.14722/ndss.2019.23438

[9] Michel Van Eeten, OrçunÇetin, Carlos Ganán, LisetteAltena, and SamanehTajalizadehkhoob.The year 2019. Evaluation of Vulnerability Notifications via Quarantine Networks: Tell Me You Fixed It. As part of the 2019 IEEE European Symposium on Security and Privacy (EuroS&P). SPIE, 326-339.

[10] Jinchun Choi, Afsah Anwar, Jeffrey Spaulding, DaeHunNyang, and Aziz Mohaisen are also involved. The year 2019. An Exploration of the Internet of Things Malware Ecosystem: Analysis and Exposures in the Wild. Sections 413–418 of the ACM/IEEE Symposium on Edge Computing Proceedings.

the eleventh Costin Andrei and Zaddach Jonas. present year. Internet of Things Malware: Full Survey, Analysis Methodology, and Real-World Examples. (2018) by BlackHat USA.

[12] In 2021, EmanueleCozzi. Internet of Things Malware Catalogue: A Complex Family Tree. Here is the dataset: https://github.com/eurecom-s3/tangled_iot/tree/master/dataset.

Paris, France's Sophia Antipolis, EmanueleCozzi Guillaume Claret France, France, Matteo Dell Matteo Dell, Pierre-Antoine Vervier, Yun Shen, Leyla Bilge, DavideBalzarotti, and Balzarotti. 2020. A Complex Family Tree of Internet of Things Malware. Included in the ACSAC Proceedings, the annual conference on computer security applications.

DavideBalzarotti, EmanueleCozzi, Mariano Graziano, and YanickFratantonio published their work in 2018. Understanding Linux Malware. Section S&P of the 2018 IEEE Symposium on Security and Privacy, pages 161–175, presented in May.

"CyberIOCs" (2020) [15].Malware bundle updated daily by CyberIOCs. At https://freeiocs.cyberiocs.pro, you can discover

Authors: Fan Dang, Yunhao Liu, Yan Chen, TianyinXu, EnnanZhai, Qi Alfred Chen, and Jingyu Yang [16]. The year 2019.Exploring Fileless Attacks on Linux-based Internet of Things Devices using Honeycloud.Pages 482-493 in MobiSys: Proceedings of the Annual International Conference on Mobile Systems, Applications, and Services.

[17] Carlos Gañán and Antoine d'Estalenx. 2021. The NURSE app is a smart home malware detector that is easy to use. The article is included in the IoT '21 proceedings and can be found on pages 1-8.

[18] Authors: Alan Said, Michel Edkrantz, and StaffanTruvé. 2015. Exploit Vulnerability Prediction in the Real World. Includes pages 513-514 from the IEEE International Conference on Cyber Security and Cloud Computing.

This information is sourced from Exploit-DB in 2009. A database of exploits for use by researchers, penetration testers, and ethical hackers. Results obtained on June 15, 2021, from the following URL: https://www.exploit-db.com/r

In [20], Last year, Alejandro Fanjul... CMS 2.5 from LG SuperSign EZ. Obtain this information on May 01, 2021 by visiting https://www.exploit-db.com/exploits/45448.

[21] This is a 2019 publication by XuanFeng, Xiao Liao, X Wang, Qiang Li, Kai Yang, Hong Zhu, and Limin Sun. Understanding and Securing Device Vulnerabilities via Automated Bug Report Analysis. Published in the USENIX Security Symposium Proceedings.

[22] It was 2016. Jerry Gamblin.The Code for Mirai. Here is the link to the Mirai source code: https://github.com/jgamblin/MiraiSource-Code.