An Inside Look at IoT Malware

Aohui Wang[1,3(✉)], Ruigang Liang[1,3], Xiaokang Liu[1,3], Yingjun Zhang[2,3], Kai Chen[1,3], and Jin Li[4]

[1] State Key Laboratory of Information Security, Institute of Information Engineering, CAS, Beijing, China
{wangaohui,liangruigang,liuxiaokang,chenkai}@iie.ac.cn
[2] Trusted Computing and Information Assurance Laboratory, Institute of Software, CAS, Beijing, China
yjzhang@tca.iscas.ac.cn
[3] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[4] Department of Computer Science, Guangzhou University, Guangzhou, China
jinli71@gmail.com

Abstract. It was reported that over 20 billion of Internet of Things (IoT) devices have connected to Internet. Moreover, the estimated number in 2020 will increase up to 50.1 billion. Different from traditional security-related areas in which researchers have made many efforts on them for many years, researches on IoT have just started to receive attentions in recent years. The IoT devices are exposing to many security problems, such as weak passwords, backdoors and various vulnerabilities including buffer overflow, authentication bypass and so on. In this paper, we systemically analyze multiple IoT malware which have appeared in the recent years and classify the IoT malware into two categories according to the way in which IoT malware infect devices: one is to infect IoT devices by brute force attacks through a dictionary of weak usernames and passwords; while the other one by exploiting unfixed or zero-day vulnerabilities found in IoT devices. We choose Mirai, Darlloz and BASHLITE as examples to illustrate the attacks. At the end, we present strategies to defend against IoT malware.

Keywords: Internet of Things · Malware · Botnet

1 Introduction

In recent years, the Internet of Things (short for IoT) which connect cyber devices embedded with software, electronics and sensors, have been developed prosperously. Traditional physical devices are offline, while IoT technologies push them online, making it possible to control these devices remotely by exchanging various data through Internet. This not only makes our life easier, but also increases the risk of malware infection on the IoT devices at the same time. According to the report from Symantec [1], IoT devices have been the target of lots of malware and have become one of the main sources of the distributed denial of service (DDoS) attacks. The dilemma is partially because of the design flaws in the IoT architecture [2], and also partially because of low quality of IoT software code. By taking the advantage of problems listed above,

F. Chen and Y. Luo (Eds.): Industrial IoT 2017, LNICST 202, pp. 176–186, 2017.
DOI: 10.1007/978-3-319-60753-5_19

malware is created aiming at IoT devices. The IoT malware could steal users' private information, build botnets and even break the whole network infrastructure.

Discovering and analyzing software vulnerabilities and malware in IoT plays an important role in current security researches [3]. According to a report by Businessinsider [4], over 45 vulnerabilities in IoT devices are found in Defcon 2016, and totally 21 companies were impacted. Types of vulnerabilities found range from bad software design such as the use of weak and hard-coded passwords to flaws of coding like buffer overflows and command injection. And according to recent researches, more than 10% apps in 33 Android market and 6.84% apps in Google Play may contain malicious code [5].

There was not much systematic work to analyze IoT malware before. Motivated by this, we choose some typical malware to analyze. According to how malware affects IoT devices, we find that there are two major categories of IoT malware as described in Abstract. Although IoT malware based on brute-force attack plays the major role nowadays, this problem is easy to fix by vendors. However, fixing the vulnerabilities such as buffer overflow is very hard. Thus, code injection attack by exploiting the vulnerabilities could be the first choice for IoT malware in the near future. In this paper, we analyze these two kinds of IoT malware and give the examples of attacks. We also summarize the ways to defend them.

Section 2 describes IoT malware which are based on brute-force attack. We choose the Mirai to show details about this kind of malware. Section 3 describes IoT malware which is based on exploiting vulnerabilities in devices. We make a summary of popular vulnerabilities in the IoT devices manufacturer, and choose Darlloz and BASHLITE as examples to reveal the details. Section 4 describes some defense strategies to prevent IoT malware from spreading. Section 5 introduces the related work on IoT security. Section 6 gives the conclusion.

2 IoT Malware Based on Brute-Force Attack

2.1 Background

Using weak passwords is a security issue that has been present since the born of computers. According to a report by ESET, about 15% of the tested routers use weak or default usernames and passwords. It was reported that "admin" is the most common username. It is also discovered that nearly 20% of the tested routers expose their Telnet port to the Internet, which is a serious security implication [6].

Dyn is a cloud-based Internet Performance Management company in charge of many companies' internet domain name system (DNS) infrastructure. In October 2016, Dyn encountered an attack by more than one hundred thousand infected end devices [7]. Many of these devices got infected with a notorious malware called Mirai. This attack made websites such as Twitter, GitHub and Airbnb inaccessible to nearly half of Americans. We find that Mirai spread by brute-force attack and there are also some other IoT malware such as Remaiten and Aidra which affect devices in similar ways. In Sect. 2.2, we will make a complete analysis of Mirai.

2.2 Mirai

Mirai is malware that can compromise IoT devices which run Linux operating system and have Telnet (port 23) or port 2323 open remotely by brute-force attack. Those compromised devices are used as part of a botnet for large-scale DDoS attacks. It primarily targets online IoT devices such as cameras and routers that have at least ten architectures including ARM, MIPS and X86. The Mirai botnet played an important role in the recent destructive attacks, such as DDoS attacks on security journalist Brian Krebs's website in September 2016, and an attack on Dyn in October 2016. The source code of Mirai was publicly released on September 30, 2016 by Anna-senpai in the hacking community Hackforums. According to a report by IT world, after the Mirai source code is released, more IoT botnets are created by hackers [8].

Operating Principle

Mirai botnet consists of a Command and Control (short for CNC) server, a receiver for scanning results and a distributor (CB server for short), and an http or ftp server to store bot programs for downloading. The CNC server is used for managing the botnets and distributing commands to bot devices. CB server is used for receiving devices information which are just compromised and guides the devices to download bot program. The whole working network could be shown as (Fig. 1).

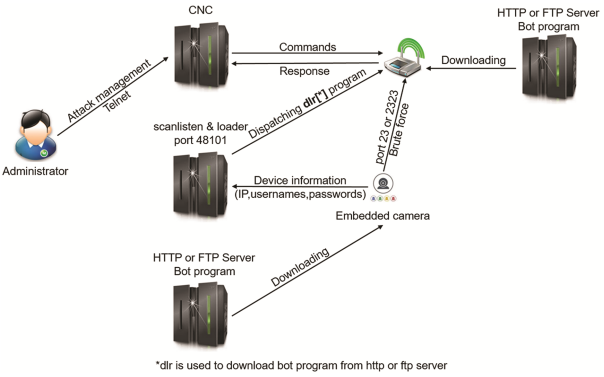

Fig. 1. The working principle of the Mirai botnet.

When a device is compromised, it will randomly select an IP address to scan. If the device being scanned has 23 port (or 2323 port) open, Mirai malware try to attack the device by brute-forcing through a dictionary of popular usernames and passwords, such as "admin", "password" and "root". If the username and password are right, the device's IP address, port, username and password will be transferred to the CB server. The CB server receives the compromised devices' details and tries to guide the device to download bot program. If the device finishes downloading the bot program, it becomes a bot device, continuing to scan other devices and waiting for commands from CNC server. An administrator can login on a CNC server to manage the bot devices and distribute commands to bot devices.

Distinguished features

There are some distinguished features which make Mirai powerful and different from precedent IoT malware. Mirai can disable devices' watchdog function to prevent them from restarting and kill competitor malware processes.

1. Monopolize devices. Mirai will disable watchdog function, and kill SSH, Telnet and HTTP daemons and occupy these ports to prevent others to access the device. Mirai will also kill other competitor malware such as Qbot [9], Zollard [10] and Remaiten [11].
2. Hide process name. Program name can be determined by the Linux command ps aux, or by reading the /proc/pid/cmdline. The running process's argument 0 is the process name, Mirai uses the random string to replace the argument 0 string. Also Mirai use the prctl system call with the PR_SET_NAME argument to make the process name to random string.
3. Unique infecting methods. Different from old methods to infect more devices directly through bot devices, a CB server is used for infecting devices specially. A CB server is used to receive feedback results from brute force attack, and distribute bot program to the compromised devices.
4. Advanced SYN scan technology. Bots brute working devices by scanning Telnet service using an advanced SYN scanner that is around 80 times faster than scanners in Qbot malware, and uses almost 20 times less resources [12].
5. Variety of attack methods. Mirai botnet can launch multiple attacks including straight UDP flood, DNS water torture, SYN flood, GRE IP flood and so on.

Detection and Defense

We made a study of the source code of Mirai that was released to public. From the analysis, we found there is an approach to locate the active infected devices and the attack infrastructure such as CNC servers. Also, we figured out a way to protect our devices from the infection of Mirai.

Detection. Internet Service Provider can locate the bot devices and CNC servers from the network traffic. To make a connection to a CNC server, bot devices will try to resolve the domain name of the CNC server by a DNS server 8.8.8.8. We can collect some features such as CNC domain name and look for the features in the network traffic. In this way, we were able to locate the infected devices. We also found that bot devices and CNC servers will send each other heartbeat packets every 60 s. Searching this pattern from the network traffic, we can locate the infected devices and the CNC server.

Defense. An infected device uses the port number 48101 to prevent multiple instances of bot program running together. The bot program will listen on the port when the device is first infected, as shown in Fig. 2. It will quit if another bot program connects this port which means two bot programs are running in the same device, as shown in Fig. 3. We create one defensive program which runs in the device forever and connects the port number 48101 every ten seconds. If Mirai infects the device, it suicides in 10 s, as shown in Fig. 5. We did experiments and the results shown that the malware exits right away after it infects the device (Fig. 4).

```
addr.sin_family = AF_INET;
addr.sin_addr.s_addr = local_bind ? (INET_ADDR(127,0,0,1)) : LOCAL_ADDR;
addr.sin_port = htons(SINGLE_INSTANCE_PORT);  //bind to port 48101 when run the first time
if (bind(fd_ctrl, (struct sockaddr *)&addr, sizeof (struct sockaddr_in)) == -1)
{  addr.sin_family = AF_INET;
   addr.sin_addr.s_addr = INADDR_ANY;
   addr.sin_port = htons(SINGLE_INSTANCE_PORT);
   //connect to port 48101 when the bot program run the second time
   if (connect(fd_ctrl, (struct sockaddr *)&addr, sizeof (struct sockaddr_in)) == -1){}
}else
   if (listen(fd_ctrl, 1) == -1){}   //listen to socket when bot program first run
```

Fig. 2. When the bot program runs the first time, it will bind and listen to the port number 48101. When the bot program runs the second time, it will connect to port 48101.

```
if (fd_ctrl != -1 && FD_ISSET(fd_ctrl, &fdsetrd))  another Mirai program connects to port 48101
{   struct sockaddr_in cli_addr;
   socklen_t cli_addr_len = sizeof (cli_addr);
   accept(fd_ctrl, (struct sockaddr *)&cli_addr, &cli_addr_len);
#ifdef DEBUG
   printf("[main] Detected newer instance running! Killing self\n");
#endif
#ifdef MIRAI_TELNET
   scanner_kill();
#endif
   killer_kill();  attack_kill_all();  kill(pgid * -1, 9);
   exit(0);   };
```

Fig. 3. When another Mirai instance connects to port 48101, the first Mirai instance will kill itself.

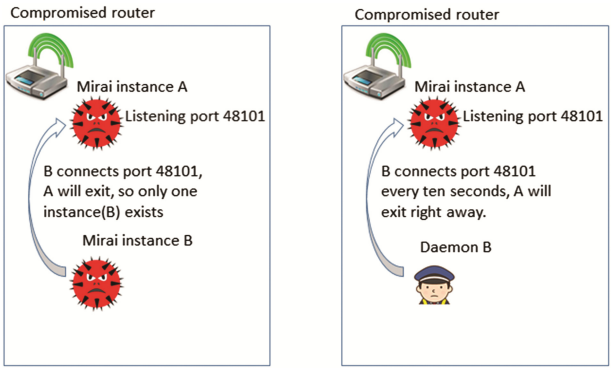

Fig. 4. Figure on the left shows how Mirai bot program prevents multiple instances running at the same time. Figure on the right shows how to protect device from Mirai infection.

3 IoT Malware Based on Exploiting Vulnerabilities

3.1 Background

More topics about the IoT security are shown on security conferences such as BlackHat, Defcon and GeekPwn in recent years with the increasing attention on IoT. Many topics about the IoT security are show on security conferences in recent years such as BlackHat, Defcon, Usenix, Pwn2Own and Geekpwn. For example, players are encouraged to crack IoT devices live in GeekPwn. In GeekPwn 2016, players from Chaitin exploit ten routers from Cisco, Huawei, Xiaomi, Asus and cameras from Xiaomi [13]. According to OWASP, the top IoT vulnerabilities include unencrypted services, poorly implemented encryption, buffer overflow, denial of service and so on. In Sect. 3.2, in order to learn the present situation and risks that IoT devices are facing, we make a statistics about the vulnerabilities in eight IoT manufacturers. In Sect. 3.3 we choose Darlloz and BASHLITE as examples to illustrate how IoT malware use vulnerabilities to spread.

3.2 Statistics

According to the Common Vulnerabilities and Exposures (short for CVE) [14] databases, we make a statistics to vulnerabilities of IoT devices in Cisco, Huawei and other six companies, as shown in Figs. 5 and 6. From Fig. 5 we can learn that the IoT devices from Cisco have the most vulnerabilities through 2012 to 2015, which are 60, 136, 142, 178 respectively. It's not surprised that there are so many vulnerabilities in Cisco because of the big market of Cisco products. The number of vulnerabilities found in EMC's products is not a large number but increase each year through 2012 to 2015. And the number of vulnerabilities found in the rest of the companies is in a relatively stable state, with subtle increase.

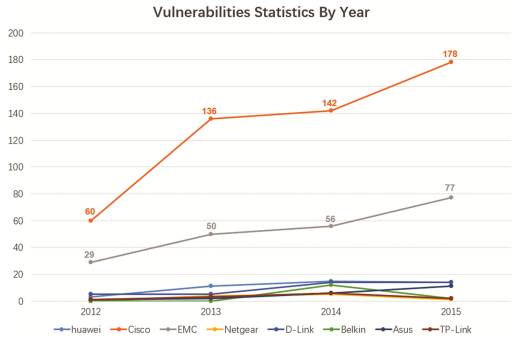

Fig. 5. Vulnerabilities grouped by company from 2012 to 2015

Figure 6 shows the number of vulnerabilities found in eight companies according to vulnerability type through 2012 to 2016(till November). From the figure, we can learn that DoS vulnerabilities take a large portion in most companies. Off all the vulnerability types, DoS takes 29.2%, while XSS and code execution take 15.2% and 11.5%, ranking

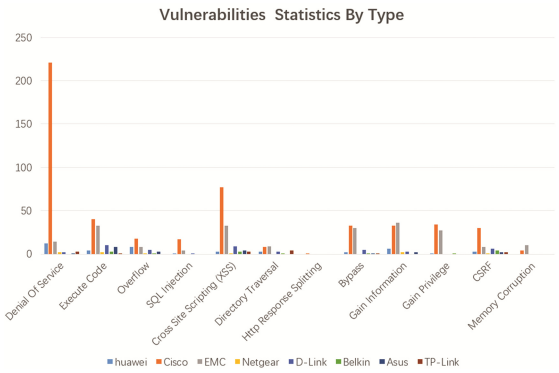

Fig. 6. Vulnerabilities grouped by type from 2012 to 2016 (till November).

the second and the third respectively. These vulnerabilities are the source of malware based on exploiting unfixed or zero-day vulnerabilities. On November 28, 2016, a variant of Mirai botnet is scanning IoT devices using a code execution vulnerability in TR069/TR064 that can hijack or crash the device. The attack caused about 900 thousands routers crash and affect over 20 million users in Germany [15]. IoT devices are facing great challenges because of the more vulnerabilities found in IoT. In the Sect. 3.3, in order to learn how malware based on exploiting vulnerabilities works, we select Darlloz and BASHLITE to illustrate.

3.3 Samples Analysis

Darlloz is a worm which targets at the IoT and infects cameras, routers and so on by exploiting a 'php-cgi' information disclosure vulnerability in PHP which is an old vulnerability that was patched in May 2012. The Darlloz was first discovered by Symantec in 2013 [16]. The whole working network can be shown as Fig. 7.

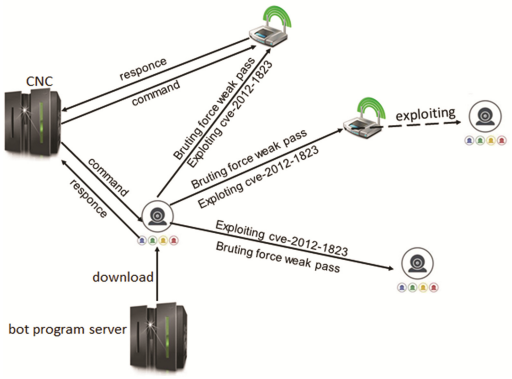

Fig. 7. The working principle of the Darlloz botnet.

When a device is compromised, it will randomly select an IP address to scan. If the Telnet port (port 23) in the device being scanned is open, it will be attacked by brute force. If the device is not vulnerable to weak passwords, the malware will try to exploit the target device using CVE-2012-1823. CVE-2012-1823 exists in PHP version before 5.3.12 and 5.4.x before 5.4.2, there is a fatal problem when cgi_main.c which is configured as a CGI script in PHP handles query strings that lack an equal sign character. This problem allows remote attackers to execute arbitrary commands in the query string [17]. If the device is exploited, it downloads the bot program from a malicious server. Then it connects to CNC server, waits for commands and tries to spread at the same time.

BASHLITE is another malware which affects IoT devices using ShellShock. ShellShock [18] is a family of severe security bugs in Unix Bash Shell. It was disclosed on 24 September 2014. The vulnerability exists in GNU Bash before version 4.3 and allows attackers to execute arbitrary commands. Vulnerable GNU Bash executes commands that are concatenated to the end of function definitions which are stored in various environment variables. IoT devices with busybox [19] installed have this vulnerability. When a device is compromised, it will download bot program from malicious server. The bot program is used to compromise other devices and waits for CNC server to launch DDoS attack.

4 Strategies for Defending

We find that most IoT malware attack devices by brute force methods or exploiting vulnerabilities in software or hardware in devices. So we come up some IoT malware defending strategies.

First, IoT devices producer should disable default or weak usernames and passwords. It's the reason that most of IoT malware exists.

Second, improving code quality in IoT devices' software. According to the statistics of the IoT vulnerabilities, we find that poor quality of code contributes to most of vulnerabilities.

Third, design secure IoT architecture that covers aspects from bottom up. Some aspects such as secure booting, access control, device authentication and updates & patches should be taken into consideration. Designing secure architecture can make the devices secure from the root level.

5 Related Work

IoT secure architecture, malware analysis, detection and prevention. A lot of efforts have been made to keep IoT devices secure. S. Chakrabarty et al. present a secure IoT architecture that contains four basic IoT architectural blocks to ensure a secure Smart City. The architecture can help mitigates cyber attacks at IoT nodes themselves [20]. A. Vimal Jerald et al. propose a novel security architecture that can help protect IoT devices from user and device authentication, sensor network, cloud and internet, applications and services [21]. Much of work propose detection and prevention methods for IoT malicious malware. Hao Sun et al. propose an anti-malware system called CloudEyes

that provides efficient and secure services for resource-constrained IoT devices [22]. Android and IOS devices take the big part in all IoT devices. Ham et al. use linear support vector machine to detect Android malware code to ensure the safety of Android devices [23]. Chen et al. design a novel homology analysis method to detect application clones [24] on Android markets and malware on Android [5] and IOS [25] platform. Pa, Y.M.P et al. design a practical IoT honeypot and sandbox, and catch at least 4 distinct IoT malware families that target at Telnet-enabled IoT devices [23]. Chun-Jung Wu et al. capture logs of 3 million telnet sessions of IoT malware and design a method based on text mining algorithm for IoT malware behavior analysis [24]. Since a number of various IoT malware that spread by exploiting vulnerabilities of PE file format have been catched. June Ho Yang et al. design a command-line tool for IoT malware detection [25]. Byungho Min et al. design various advanced attacks targeting at IoT aspect of smart home, and evaluate the impact via practical evaluations and propose offensive techniques [26]. Software-defined networking [27] (short for SDN) has been popular in the recent years, which is a novel approach that allow network administrators to manage network services easily. Vandana C.P. design a new security framework for IoT based on SDN-IoT architecture [28].

6 Conclusion

In this paper, we seek to analyze two major kinds of malware targeting IoT devices. In malware based on brute force attack, we choose Mirai as an example to analyze. Mirai has some unique advantages such as monopolizing devices, hiding process information, advanced scan technology which make it more powerful than former malware. We make a statistics about the vulnerabilities in IoT devices, and conclude that IoT malware will utilize vulnerabilities such as buffer overflow and command injection more and more. We take Darlloz and BASHLITE as an example to analyze. Darlloz uses the CVE-2012-1823 to exploit IoT devices, BASHLITE uses the ShellShock to do these things. There are many topics about IoT devices in security conferences around the world in recent years, including secure architecture design, vulnerabilities analysis. Designing secure architecture can protect the devices from the root level. At the end of paper, we present strategies for protect IoT devices.

Acknowledgement. The IIE authors were supported in part by NSFC U1536106, 61100226, Youth Innovation Promotion Association CAS, and strategic priority research program of CAS (XDA06010701). Yingjun Zhang was supported by National High Technology Research and Development Program of China (863 Program) (No. 2015AA016006) and NSFC 61303248.

References

1. Symantec blog. https://www.symantec.com/connect/blogs/iot-devices-being-increasingly-used-ddos-attacks
2. Jing, Q., Vasilakos, A.V., Wan, J., et al.: Security of the Internet of Things: perspectives and challenges. Wirel. Netw. **20**(8), 2481–2501 (2014)

3. Zhang, Z.K., Cho, M.C.Y., Wang, C.W., Hsu, C.W., Chen, C.K., Shieh, S.: IoT security: ongoing challenges and research opportunities. In: 2014 IEEE 7th International Conference on Service-Oriented Computing and Applications, pp. 230–234. IEEE (2014)
4. Inside the Internet of Things village at DefCon. http://www.businessinsider.com/iot-village-defcon-2016-8
5. Chen, K., Wang, P., Lee, Y., et al.: Finding unknown malice in 10 seconds: mass vetting for new threats at the Google-Play scale. In: USENIX Security, vol. 15 (2015)
6. Enjoy Safer Technology. https://www.eset.com/int/
7. Dyn blog. http://dyn.com/blog/dyn-statement-on-10212016-ddos-attack/
8. Hackers create more IoT botnets with Mirai source code. http://www.itworld.com/article/3132570/hackers-create-more-iot-botnets-with-mirai-source-code.html
9. The Qbot. https://sourceforge.net/p/theqbot/wiki/Home/
10. Linux.Darlloz. https://www.symantec.com/security_response/writeup.jsp?docid=2013-112710-1612-99
11. Remaiten. https://en.wikipedia.org/wiki/Remaiten
12. Mirai Source. https://github.com/jgamblin/Mirai-Source-Code
13. GeekPwn blog. https://blog.geekpwn.org/2016/05/19/security-geek-winners-awarded-one-million-yuan-prize/
14. Common Vulnerabilities and Exposures. https://cve.mitre.org
15. Mirai bots attack 1 m German routers. http://www.theregister.co.uk/2016/11/28/
16. Symantec blog. https://www.symantec.com/connect/blogs/linux-worm-targeting-hidden-devices
17. CVE-2012-1823. https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2012-1823
18. ShellShock. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6271
19. Busybox. https://en.wikipedia.org/wiki/BusyBox
20. Chakrabarty, S., Engels, D.W.: A secure IoT architecture for Smart Cities. In: 2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC). IEEE (2016)
21. Jerald, A.V., Rabara, S.A., Bai, D.P.: Secure IoT architecture for integrated smart services environment. In: 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), pp. 800–805. IEEE, October 2016
22. Sun, H., Wang, X., Buyya, R., et al.: CloudEyes: cloud-based malware detection with reversible sketch for resource-constrained Internet of Things (IoT) devices. Pract. Exp., Software (2016)
23. Ham, H.S., Kim, H.H., Kim, M.S., et al.: Linear SVM-based android malware detection for reliable IoT services. J. Appl. Math. (2014)
24. Chen, K., Liu, P., Zhang, Y.: Achieving accuracy and scalability simultaneously in detecting application clones on android markets. In: Proceedings of the 36th International Conference on Software Engineering, pp. 175–186. ACM (2014)
25. Chen, K., Wang, X., Chen, Y., et al.: Following devil's footprints: cross-platform analysis of potentially harmful libraries on android and iOS. In: 2016 IEEE Symposium on Security and Privacy (SP), pp. 357–376. IEEE (2016)
26. Pa, Y.M.P., Suzuki, S., Yoshioka, K., et al.: IoTPOT: analysing the rise of IoT compromises. EMU **9**, 1 (2015)
27. Wu, C.-J., et al.: IoT malware behavior analysis and classification using text mining algorithm (2016)
28. Yang, J.H., Ryu, Y.: Design and development of a command-line tool for portable executable file analysis and malware detection in IoT devices. Int. J. Secur. Appl. **9**(8), 127–136 (2015)

29. Min, B., Varadharajan, V.: Design and evaluation of feature distributed malware attacks against the Internet of Things (IoT). In: 2015 20th International Conference on Engineering of Complex Computer Systems (ICECCS). IEEE (2015)
30. SDN. https://en.wikipedia.org/wiki/Software-defined_networking
31. Vandana, C.P.: Security improvement in IoT based on Software Defined Networking (SDN)