DNS Rebinding Detection for Local Internet of Things Devices

Xudong He[1], Jian Wang[1](✉), Jiqiang Liu[1], Zhen Han[1], Zhuo Lv[2], and Wei Wang[1,3]

[1] Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University, Beijing, China
wangjian@bjtu.edu.cu

[2] State Grid Henan Electric Power Research Institute, Zhengzhou, China

[3] Division of Computer, Electrical and Mathematical Sciences and Engineering (CEMSE), King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia

Abstract. Smart home technology makes the living environment comfortable and safe. However, threats in the smart home environment bring more new challenges. As a typical attack method, DNS rebinding seriously threatens the data privacy and the security of smart home devices. Aiming at detecting this attack and minimizing its effect as much as possible, we use simulation experiments to model the DNS rebinding attack scenarios. Based on the analysis of the key factors of the experiments, a DNS rebinding attack detection model is proposed. When devices in a smart home environment meet the detection model, they may be vulnerable to DNS rebinding attacks. Our simulation experimental results show that the smart home devices in the detection model are vulnerable to DNS rebinding attacks. Finally, we put forward some defensive measures.

Keywords: Smart home · DNS rebinding · IoT · TTL · Attack detection

1 Introduction

In the smart home system environment, the technology of computer, network communication, and the sensor is adapted to combine the electrical equipment related to home life with the network master station which provides the service [1,2]. Communicate within the LAN, call the interface, can be completed by mature protocol standards such as FTP, HTTP, ZigBee, RFID, etc. [3]. For

This work was supported in part by the Natural Science Foundation of China under Grants 61672092, in part by the Fundamental Research Funds for the Central Universities of China under Grants 2018JBZ103, the Fundamental Research Funds for the Central Universities 2019YJS033, Major Scientific and Technological Innovation Projects of Shandong Province, China (No. 2019JZZY020128).

G. Xu et al. (Eds.): FCS 2020, CCIS 1286, pp. 19–29, 2020.
https://doi.org/10.1007/978-981-15-9739-8_2

the manufacturer, the open API interface gives the user some control of the local area network freedom. The API service interface implements the functions of the IoT device through protocol calls and parameter passing. It is a good choice for the linkage of devices implemented by different manufacturers. Because vendors open up API services interfaces that control freedom. The Internet of things devices and sensors devices include smart toys, smart home devices, smart cameras and more. We have recently witnessed many Internet of Things privacy and security issues [4,5]. Many security problems will have a serious impact on personal safety [6]. Therefore, these security issues have aroused great concern. Attacks on Internet of things, including children's security and privacy, and so on [7]. In addition, IoT devices are used to attack the Internet environment. For example, the 2016 Mirai botnet destroyed Internet of things devices and launched one of the most destructive DDoS attacks in the history of the Internet [8]. Armis, a cyber security company, recently issued another warning after discovering the Bluetooth protocol vulnerability "BlueBorne" in 2017, saying that about 500 million smart devices are still affected by old-fashioned attacks such as DNS rebinding [9]. In the smart home environment, the intrusion of sensitive devices such as cameras or constant temperature devices will have a serious impact on people's privacy and even personal safety. DNS rebinding is a form of computer attack [10]. In this attack, a malicious Web page causes the visitor to run a client script that attacks computers elsewhere on the network. The same-origin policy specifies that clientside scripts only allow access to resources on the same host that serve the script [11]. In theory, the same-origin policy can prevent this kind of attack from happening. But comparing domain names is an important part of implementing this policy, and DNS rebinding can bypass this protection by abusing the DNS. There is a browser device or a client that can access a network link in a smart home environment. When such a device inadvertently accesses a malicious site controlled by an attacker, the attacker loads the JS code for the malicious site. It is possible to control the malicious DNS server back and forth domain query, and send malicious request data to the device service interface in the environment, so as to control the devices in the LAN [12]. IoT devices are vulnerable to the threat of DNS rebinding. In this paper, the DNS query process and DNS rebinding attack process are simulated by simulation experiment. We will abstract the DNS rebinding attack detection model in the local Internet of things device scenario. The contribution of this paper is as follows:

- Through experiments, it is proved that this attack really exists, and the attack process is reproduced.
- The key factors of DNS rebinding attack in smart home environment are put forward for the first time, and the key factors are discussed, which provides a powerful basis for preventing such attacks.
- The DNS rebinding attack detection model in smart home environment is proposed for the first time. Target IoT device is under the detection model, it is proved by experiments that the TTL value and JS loading frequency in the IoT rebinding attack in the smart home environment are important factors. We can use them as a basis for detecting attacks.

The rest of the paper is organized as follows: In Sect. 2, we introduce related work on this topic. In Sect. 3, we describe the DNS rebinding detection model. In Sect. 4, Through experiments, it is proved that DNS rebinding attack exists in Smart home. The experimental validation of the model is presented. In Sect. 5, we summarize the contribution of the article and put forward the future work.

2 Related Work

This section mainly describes the related work to the key conditions involved in DNS rebinding attacks.

2.1 Domain Name System and DNS Rebinding Attack

Domain name system as a distributed database which maps domain name and IP address to each other, it can make it easier for people to access the Internet. The frequency of resource record updates in the DNS is determined by the TTL. They concluded that DNS scalability was not threatened by the TTL value of 0 resource records. TTL (Time-To-Live), which is the lifetime of a domain name resolution record in the DNS server [13,14].

There is a kind of attack called DNS rebinding in threatening the security of DNS system and users. In 1996, Princeton computer Science Laboratory first implemented DNS rebinding attack [16]. This attack is very harmful [17]. The attack process is shown in the Fig. 1.

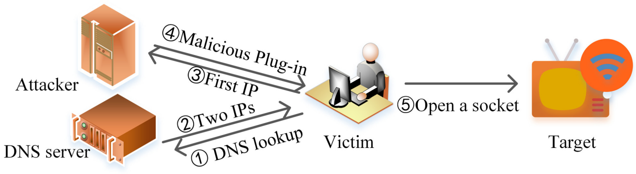

Fig. 1. DNS rebinding attack

The DNS rebinding attack subverts the browser's same-origin policy and converts it into an open network proxy. These attacks can be used to bypass firewalls and are highly effective for sending spam and spoofing payper click advertisers [15]. This attack can also be used to compromise a private network by giving the victim's Web browser access to the computer with a private IP address and returning the results to the attacker.

DNS rebinding uses the victim host as a proxy to transmit the content of the target server to the attacker. Because the local victim host is used to access the target server, it is difficult for the intranet firewall to provide effective protection. Xin Hongming et al. studied the influence of DNS rebinding on routers and the protection strategy [18]. Pandiaraja proposed a new technique by using a security proxy with a hash function. Rebinding attacks can be avoided by using this technique. It provides a secure environment for the DNS to communicate with other DNS. While the source DNS is receiving a response from any DNS it will authenticate all the receiving packets and then sends the data to the client [19].

2.2 API and JavaScript Security

Predecessors have also studied the security of JavaScript to a certain extent, Gupta, S. et al. [20] they proposed an injectionbased and clusterbased cleanup framework, JS-SAN (JavaScript SANitizer), to mitigate JavaScript code injection vulnerabilities. Because of the urgent need for the security and services of Internet of things devices, the problem of API security has attracted people's attention. Despite the development of security technology, hackers still seem to be able to find security vulnerabilities in software applications to make their attacks successful [21–23]. Lack of security and availability is one of the main factors that programmers often make mistakes in developing the application programming interface (API) for applications, which can easily lead to security vulnerabilities [24]. Wijayarathna et al. evaluated the availability of Java secure Sockets extension (JSSE) API. Their findings provide useful insights into how to design, develop, and improve TLS API [25]. Kai Mindermann et al. believe that many encryption software libraries are not easy to use, and that there are many ways to improve (encrypt) API, in such a way as to improve the robustness of existing API [26].

2.3 DNS Rebinding in IoT

DNS rebinding attacks are very harmful. Existing mechanisms cannot prevent all types of DNS rebinding attacks. Siva Brahmasani et al. proposed a twolevel solution [27], the first level is to use the IP address returned by the DNS response to reverse the comparison of the corresponding domain name with the original domain name, and the second level is to compare each IP address returned by the DNS response in the HTTP response content. The proposed solution can detect and prevent all subsequent DNS rebinding attacks. In the local Internet of things smart device environment, JavaScript security and API security are very important. In this environment, the Internet of things devices are vulnerable to DNS rebinding attacks even behind the firewall. Acar, G et al. described the attack scenario [28], which allows the victim to visit the attacker's Web site and communicates with an Internet of things device with an open HTTP server on the local network. However, they did not give a formal description of the attack model, and finally did not give a specific defense method. By using a Remote Code Execution vulnerability [29] or simple credentials, which are still common in many home devices, the attacker can gain control of the entire internal network. Based on the previous research on DNS rebinding attack, we reproduce DNS rebinding attack in simulated local Internet of things environment.

3 DNS Rebinding Attack Detection Model in Smart Home

This section describes the attack process and necessary parameters of the DNS rebinding attack. Through the analysis of the process and parameters, we abstract the attack detection model of the attack process.

3.1 DNS Rebinding Attack Process and Necessary Factors

This section describes the DNS rebinding attack process of smart home devices in a local Internet of things environment, as shown in the following Fig. 2. The devices include DNS server, Attacker web server and Target IoT device.

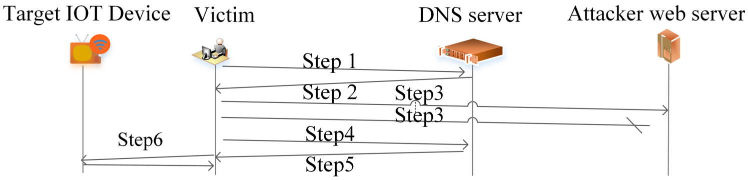

Fig. 2. DNS rebinding attack process

Step 1: The attacker controls a site to cause the victim to issue a DNS request.
Step 2: The attacker controlled DNS server responds to the victim's DNS request.
Step 3: JS fragments continue to make requests to malicious domain names.
Step 4: Do the DNS query again.
Step 5: Return the IP address of the target IoT device.
Step 6: The victim keeps making requests to the target device.

The victim's computer receives this malicious DNS response. The browser sends a POST request to API . At this point, the JavaScript code continuously sends POST or GET requests in JSON format to the target IoT to control the device.

3.2 Necessary Factors Analysis

In the course of this attack, there are three prerequisites: external malicious DNS server, internal springboard Web browser and internal IoT device address. After satisfying these three conditions, we can form a complete DNS rebinding attack in the smart home environment.

External Malicious DNS Server. The external malicious DNS server builds the attack model in the smart home device, and the malicious DNS server plays an essential role. It responds to the domain name query request and points the IP to the target device.

Internal Springboard Web Browser. Springboard Web browser is an important part of building attack model in smart home devices. In the process of DNS rebinding, it communicates with the external network as the initiator of the request to the external network.

Internal IoT Device Discovery. At present, there are many security problems of IoT devices in smart home environment. It easily contributes the device to be the target of DNS rebinding attack. In the constructed attack model, the discovery of the internal device IP address and the interface are the key issues [30], which are the core of detecting this attack.

3.3 DNS Rebinding Attack Detection Model

After satisfying the necessary conditions for DNS rebinding, we model the DNS rebinding for a single attack, as shown in the Fig. 3.

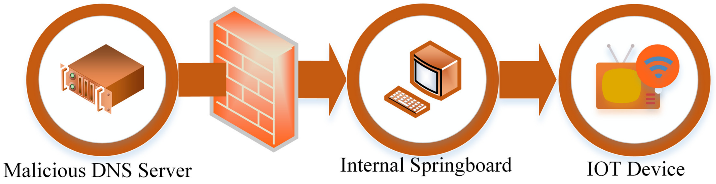

Fig. 3. DNS rebinding attack model

In the process of building the attack model, we found several key factors, including the DNS query response time (Affected by TTL), JS load frequency, and the response time of the victim page.

(1) The DNS query response time that between springboard and malicious DNS server affect the domain name of the internet and the request corresponding time of the IP returned by the malicious attack.

(2) JS load frequency between the springboard and the IoT device affects the discovery of the service interface of the IoT device. Malicious JS code loading frequency is too fast, increasing the pressure on local IoT device services, and easily blocking normal request links. The request frequency is too low, malicious JS code can not efficiently explore the effective port of IoT device service.

(3) The victim's page response time is related to the efficiency of DNS query, and the most intuitive feeling feedback to the attacker is that the loading speed of the page. The value of TTL and the JS loading frequency play an important role in DNS rebinding attacks.

4 Simulation Experiment

According to our attack detection model, the attack of DNS rebinding in the environment of the Internet of things is simulated.

4.1 Simulation Network Structure and Process

The environment built by the experiment is shown in the Fig. 4.

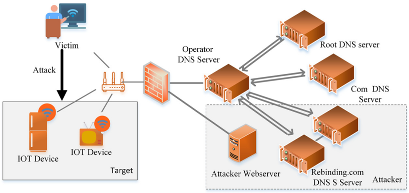

Fig. 4. Lab environment topology diagram

We used a total of eight machines to do simulation experiments. The information are shown in the Table 1.

Table 1. Resource of devices.

Device name	IP address	OS	Software	Service	Port
Victim	192.168.100.22	win7	Chrome	ssh	22 53
IoT Device	192.168.100.25	CentOS	Bind	ssh DNS	22 53
Attacker Webserver	172.17.10.11	CentOS	Apache	ssh apache	22 80
Rebinding.com DNS M Server	172.17.10.6	CentOS	Bind	ssh DNS	22 53
Rebinding.com DNS S Server	172.17.10.7	CentOS	Bind	ssh DNS	22 53
Operator DNS Server	172.17.10.8	CentOS	Bind	ssh DNS	22 53
Com DNS Server	172.17.10.5	CentOS	Bind	ssh DNS	22 53
Root DNS server	172.17.10.4	CentOS	Bind	ssh DNS	22 53

In order to better illustrate, we designed the experiment. The structure as shown in the Fig. 5.

Step 1: The victim accesses a malicious domain named www.rebinding.com built in Attacker Web Server. The ComDNS server and the RebindingDNS server to get the IP address (172.17.0.11) corresponding to the built service domain name on the webserver.

Step 2: The victim requests a web server with a IP address of 172.17.0.11 through the domain name, and the web server returns the result of the request, which contains a snippet of JavaScript that contains malicious behavior. The value of TTL in the Rebinding DNS server is set T seconds.

Step 3: The JS code snippet runs on the victim's machine and sends a request for JSON format data at intervals t seconds to the Attacker DNS server.

Step 4: The victim device fails the cache after T(the value of TTL) seconds and perform step 1 again. However, the IP address returned to the victim this

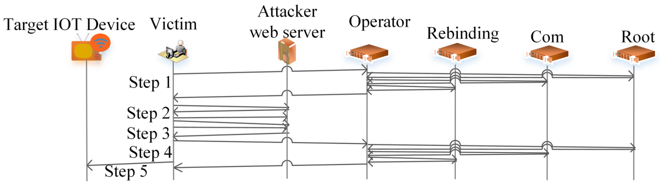

Fig. 5. Smart home DNS rebinding process

time is the address of the target IoT device in the same network segment as him, such as 192.168.100.25.

Step 5: JavaScript reloads the request www.rebinding.com/settemp interface, the victim makes repeated requests to the Internet of things device with IP address 192.168.100.25.

4.2 Analysis of Experimental Results

In the experiments, we analysis TTL value T and JavaScript loading interval t respectively. We also analyzed whether the IP address responds and the time it took for the victim's web page to load the response.

The respective values of the TTL and JavaScript variables are in seconds, and the result of a random look at whether IP binds successfully in the server is shown in the Table 2 (S is rebinding success, F is rebinding failure).

Table 2. Rebinding result

JS	TTL						
	1	15	30	60	900	1800	3600
1	S	S	S	S	F	S	S
15	S	F	S	S	F	F	S
30	S	S	F	F	S	S	S
60	S	F	F	F	F	F	F
900	S	F	F	F	F	F	F
1800	S	F	S	S	F	F	F
3600	S	F	F	S	F	F	F

To observe the response time of the victim's web page loaded, the values of the TTL and JS variables are set to the same values as in Table 2. We analyzed the load time of the victim web page as shown in the Fig. 6.

The Z-axis in the three-dimensional coordinate system of the figure is the time of the page response. Combining the results of Table 2 and Fig. 6, we can draw a conclusion that the smaller the TTL, the faster the JavaScript loading frequency, the higher the success rate of DNS rebinding attacks, and the slower the response speed of the page.

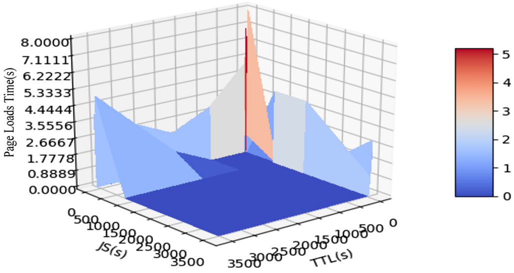

Fig. 6. Result of web page loads time

4.3 Precautionary Suggestions

Some scholars have proposed an automatic model to analyze attack behavior through mixed analysis of flow based and graph based network traffic behavior [31]. In order to protect the privacy and security of users in the vehicle network of the Internet of things, some scholars have proposed a solution based on block chain [32].

In this paper, the following suggestions can be given to detect DNS rebinding attacks in the Internet of things environment.

(1) Excluding the network speed, when the user uses a similar browser page, whether the page access speed is significantly reduced.

(2) The code on the local Internet of things device is detected whether there are frequent requests for services under a domain name by JavaScript fragments.

(3) Detect the service interface provided by the local Internet of things sensitive equipment, such as thermostat, camera and so on, whether there is abnormal traffic. Whether the device has a normal request that cannot respond.

5 Conclusion

In this paper, we found that local Internet of things devices are more vulnerable to DNS rebinding attacks because of their weak level of protection. We reproduce this kind of attack through the simulation experiment. We also quantitatively analyze the TTL value of DNS rebinding attack and the loading frequency of client JavaScript. Finally, it is found that the TTL value and the loading frequency of JS are the key factors of this attack. We propose a DNS rebinding attack detection model in smart home environment, including internal springboard, malicious DNS server and target Internet of things devices. Finally, We have put forward some preventive suggestions. The following work will focus on the defense method of multi-device attack threat model and the edge security in the local network.

References

1. Panwar, N., Sharma, S., Mehrotra, S., Krzywiecki, L., Venkatasubramanian, N.: Smart home survey on security and privacy. CoRR, abs/1904.05476 (2019)

2. Tong, X., Fang, B., He, Y., Zhang, Y.: Analysis on the development of internet of things smart home. Mobile Commun
3. Chirila, S., Lemnaru, C., Dînsoreanu, M.: Semantic-based IoT device discovery and recommendation mechanism. In: IEEE 12th International Conference on Intelligent Computer Communication and Processing, ICCP 2016, Cluj-Napoca, Romania, 8–10 September 2016, pp. 111–116. IEEE (2016)
4. Castro, R.R., López, J., Gritzalis, S.: Evolution and trends in IoT security. IEEE Comput. **51**(7), 16–25 (2018)
5. Wang, W., Wang, X., Feng, D., Liu, J., Han, Z., Zhang, X.: Exploring permission-induced risk in android applications for malicious application detection. IEEE Trans. Inf. Foren. Secur. **9**(11), 1869–1882 (2014)
6. Xu, T., Wendt, J.B., Potkonjak, M.: Security of IoT systems: design challenges and opportunities. In: Chang, Y.-W. (ed.) The IEEE/ACM International Conference on Computer-Aided Design, ICCAD 2014, San Jose, CA, USA, 3–6 November 2014, pp. 417–423. IEEE (2014)
7. McReynolds, E., Hubbard, S., Lau, T., Saraf, A., Cakmak, M., Roesner, F.: Toys that listen: a study of parents, children, and internet-connected toys. In: Mark, G., et al. (eds.) Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, Denver, CO, USA, 06–11 May 2017, pp. 5197–5207. ACM (2017)
8. Wikipedia. 2016 dyn cyberattack (2016). https://en.wikipedia.org/wiki/2016_Dyn_cyberattack
9. Armis. DNS rebinding exposes half a billion IoT devices (2018). https://armis.com/dns-rebinding-exposes-half-a-billion-iot-devices-in-the-enterprise/
10. Johns, M., Lekies, S., Stock, B.: Eradicating DNS rebinding with the extended same-origin policy. In: King, S.T. (ed.) Proceedings of the 22th USENIX Security Symposium, Washington, DC, USA, 14–16 August 2013, pp. 621–636. USENIX Association (2013)
11. Karlof, C., Shankar, U., Tygar, J.D., Wagner, D.A.: Dynamic pharming attacks and locked same-origin policies for web browsers. In: Ning, P., di Vimercati, S.D.C., Syverson, P.F. (eds.) Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, Alexandria, Virginia, USA, 28–31 October 2007, pp. 58–71. ACM (2007)
12. NPM. Whonow dns server. https://www.npmjs.com/package/whonow
13. Yi, W., Janne, T., Mikael, L.: An analytical model for DNS performance with TTL value 0 in mobile internet. In: Ning, P., di Vimercati, S.D.C., Syverson, P.F. (eds.) TENCON 2006 IEEE Region 10 Conference, pp. 1–4. IEEE (2006)
14. Cohen, E., Kaplan, H.: Proactive caching of DNS records: addressing a performance bottleneck. Comput. Netw. **41**(6), 707–726 (2003)
15. Jackson, C., Barth, A., Bortz, A., Shao, W., Boneh, D.: Protecting browsers from DNS rebinding attacks. ACM Trans. Web **3**(1), 2:1–2:26 (2009)
16. Princeton University. DNS attack scenario. http://sip.cs.princeton.edu/
17. Dean, D., Felten, E.W., Wallach, D.S.: Java security: from hotjava to netscape and beyond. In: 1996 IEEE Symposium on Security and Privacy, 6–8 May 1996, Oakland, CA, USA, pp. 190–200. IEEE Computer Society (1996)
18. Xin, H., Wang, Y., Zhao, R.: Research on the influence of DNS rebinding on router and its protection strategy. Chengdu Inst. Inf. Eng. **29**(6) (2014)
19. Pandiaraja, P., Parasuraman, S.: Applying secure authentication scheme to protect dns from rebinding attack using proxy. In: 2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015], pp. 1–6. IEEE (2015)

20. Gupta, S., Gupta, B.B.: JS-SAN: defense mechanism for html5-based web applications against javascript code injection vulnerabilities. Secur. Commun. Netw. **9**(11), 1477–1495 (2016)
21. Fahl, S., Harbach, M., Muders, T., Smith, M., Baumgärtner, L., Freisleben, B.: Why eve and mallory love android: an analysis of android SSL (in)security. In: Yu, T., Danezis, G., Gligor, V.D. (eds.) The ACM Conference on Computer and Communications Security, CCS 2012, Raleigh, NC, USA, 16–18 October 2012, pp. 50–61. ACM (2012)
22. Fahl, S., Harbach, M., Perl, H., Koetter, M., Smith, M.: Rethinking SSL development in an appified world. In: Sadeghi, A.-R., Gligor, V.D., Yung, M. (eds.) 2013 ACM SIGSAC Conference on Computer and Communications Security, CCS 2013, Berlin, Germany, 4–8 November 2013, pp. 49–60. ACM (2013)
23. Georgiev, M., Iyengar, S., Jana, S., Anubhai, R., Boneh, D., Shmatikov, V.: The most dangerous code in the world: validating SSL certificates in non-browser software. In: Yu, T., Danezis, G., Gligor, V.D. (eds.) The ACM Conference on Computer and Communications Security, CCS 2012, Raleigh, NC, USA, 16–18 October 2012, pp. 38–49. ACM (2012)
24. Wurster, G., van Oorschot, P.C.: The developer is the enemy. In: Bishop, M., Probst, C.W., Keromytis, A.D., Somayaji, A. (eds.) Proceedings of the 2008 Workshop on New Security Paradigms, Lake Tahoe, CA, USA, 22–25 September 2008, pp. 89–97. ACM (2008)
25. Wijayarathna, C., Arachchilage, N.A.G.: Why Johnny can't develop a secure application? A usability analysis of java secure socket extension API. Comput. Secur. **80**, 54–73 (2019)
26. Mindermann, K., Wagner, S.: Usability and security effects of code examples on crypto apis. In: McLaughlin, K., et al. (eds.) 16th Annual Conference on Privacy, Security and Trust, PST 2018, Belfast, Northern Ireland, Uk, 28–30 August 2018, pp. 1–2. IEEE Computer Society (2018)
27. Brahmasani, S., Sivasankar, E.: Two level verification for detection of DNS rebinding attacks. Int. J. Syst. Assur. Eng. Manag. **4**(2), 138–145 (2013)
28. Acar, G., Huang, D.Y., Li, F., Narayanan, A., Feamster, N.: Web-based attacks to discover and control local iot devices. In: Proceedings of the 2018 Workshop on IoT Security and Privacy, IoT S&P@SIGCOMM 2018, Budapest, Hungary, 20 August 2018, pp. 29–35. ACM (2018)
29. MSRC: Prevent a worm by updating remote desktop services (cve-2019-0708). https://msrc-blog.microsoft.com/2019/05/14/prevent-a-worm-by-updating-remote-desktop-services-cve-2019-0708/
30. Feng, X., Li, Q., Wang, H., Sun, L.: Acquisitional rule-based engine for discovering internet-of-thing devices. In: 27th USENIX Security Symposium USENIX Security 18), pp. 327–341 (2018)
31. Wang, W., Shang, Y., He, Y., Li, Y., Liu, J.: Botmark: automated botnet detection with hybrid analysis of flow-based and graph-based traffic behaviors. Inf. Sci. **511**, 284–296 (2020)
32. Li, L., et al.: Creditcoin: a privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles. IEEE Trans. Intell. Transp. Syst. **19**(7), 2204–2220 (2018)