Study of DNS Rebinding Attacks on Smart Home Devices

Dennis Tatang, Tim Suurland, and Thorsten Holz

Ruhr University Bochum, Germany
firstname.lastname@rub.de

Abstract. *DNS rebinding* is an attack technique know for more than 20 years, which is experiencing a revival caused by the ever-increasing networking of Internet of Things (IoT) devices. Thus, the potential attack surface is growing rapidly, and this paper shows that DNS rebinding attacks on many smart home devices are still successful. Nevertheless, various conditions must be fulfilled for this type of attack. This leads to the fact that such attacks rarely occur in practice since router vendors often provide DNS rebinding protection. Nevertheless, we believe that it is valuable to investigate whether individual devices are theoretically vulnerable and to create a certain awareness so that the existing countermeasures are used correctly.
As part of this paper, we conducted a study analyzing five devices, four smart home devices and one router as a smart-home gateway connected with the IoT products. Three out of four of the smart home devices are vulnerable, and the router is partially vulnerable because queries reach localhost despite activated DNS rebinding protection; thus, services on localhost are vulnerable. This indicates that the manufacturers of smart home devices rely on the countermeasures of the routers in the first place, but it might even improve the security of the devices if they already implement their own additional countermeasures.

Keywords: DNS · IoT · DNS Rebinding

1 Introduction

The spread of smart appliances leads to increase networking between the devices themselves and thus to smart homes. For attackers, this development represents an increased attack surface. In particular, devices accessible via the Internet are attractive targets. Consumers may assume that if smart home devices are reachable locally only, they pose no risk to the home network. However, with DNS rebinding attacks, it is possible to communicate with only internally accessible devices. DNS rebinding allows unauthorized access to private networks.

DNS rebinding attacks are known since 1996 [5,6]. As a result, various attack methods, as well as countermeasures, were already published [4,9]. However, a recent study from 2018 demonstrates that DNS rebinding attacks are still feasible today [1]. Even the assigned CVEs (a total of 25) indicate that DNS rebinding, since 2017 (11 out of 25 CVEs), experiences a revival [3]. This observation

correlates with the increasing number of Internet of Things (IoT) devices. Worldwide, approximately half a billion devices are estimated to be vulnerable to DNS rebinding attacks in 2018 [2].

In this paper, we investigate DNS rebinding attacks on smart household appliances, a subset of the IoT. We show that sensitive data can be extracted and remote control from the Internet is possible. We analyze the execution of the attacks systematically and summarize them. In addition, we investigate which prerequisites must be fulfilled in order to carry out the attacks. In this way, we discuss how serious the risk is for smart home device owners to become victims. Finally, we analyze the top 100 Alexa web pages that communicate over HTTP to investigate whether DNS rebinding is performed unnoticed on one of these pages. It was demonstrated that protection mechanisms of dnsmasq do not detect attacks on localhost (127.0.0.1), 4 out of 5 investigated devices are vulnerable, and none of the top 100 sites performs DNS rebinding attacks.

To summarize, we make the following contributions:

- We systematically analyze DNS rebinding attacks on four smart home devices and summarize our results in an overview.
- We investigate requirements for successfully DNS rebinding attacks and discuss the risk of becoming a victim of such an attack.
- We present a brief measurement study on the execution of DNS rebinding on popular websites.

In the remainder of the paper, we first introduce basic knowledge and identify requirements for successfully DNS rebinding attacks in Section 2. Afterwards, we describe our conducted experiments in Section 3, followed by presenting the results in Section 4. In Section 5, we discuss the results and limitations. Section 6 presents some related work and we conclude our work in Section 7.

2 DNS rebinding attack

During a DNS rebinding attack, an attacker bypasses the security mechanism of the firewall in the router and communicates interactively with devices in its local network by using the browser of the victim. This is achieved by manipulating the hostname and IP address mapping, which makes the attacker's browser become a proxy into the victim's private network.

2.1 High-level concept

The attacker bypasses the router firewall in a DNS rebinding attack by abusing the browser within the internal network as a proxy to communicate with the devices inside the local network. Figure 1 visualizes the concept of the attack.

To establish a connection to an internal local device of the victim, the attacker must assign the DNS hostname of his web server to the internal IP address of the target device. This works by the attacker running a DNS name server next to his web server on his attack server. If the domain of his website is to be resolved into

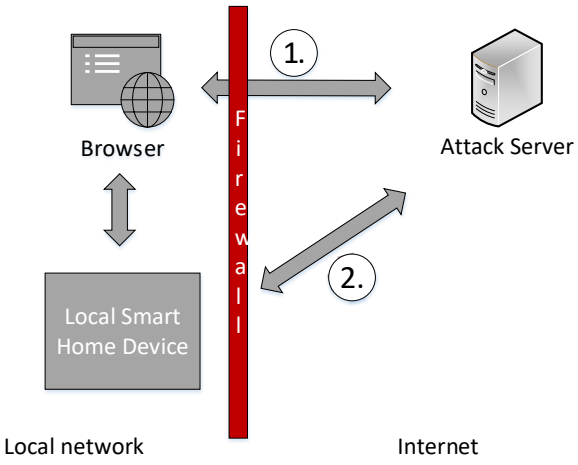

Fig. 1. High-level concept of DNS Rebinding attacks. The browser becomes by manipulating the mapping of hostnames and IP addresses to the proxy into the internal network (see ①). Direct access is blocked by the router firewall (see ②).

an IP address, he also receives the corresponding DNS request, which can then be manipulated. The browser trusts the DNS response, and thus the connection to a local network device can be established by the manipulated domain name and IP address assignment. An attacker does not have to compromise a DNS server; it is sufficient to generate valid DNS replies for requests to resolve his/her domain. Note, DNSSec is not able to prevent this attack scenario because the attacker only generates valid DNS responses to queries for his/her domain.

2.2 Attack methods

To perform a DNS rebinding attack, different methods can be exploited. In the following, we describe two examples. First, multiple A records were historically exploited to perform DNS rebinding attacks. A second vulnerability is time-varying DNS, which can be used to perform DNS rebinding attacks. Our experiments conducted subsequently are related to the second type of DNS rebinding attack.

Multiple A records. The mapping of a domain to an IP address is implemented using A record requests. The DNS allows mapping multiple IP addresses to one domain. These multiple A records are used to realize a load distribution in the DNS. All IP addresses are summarized as Resource Record set.

The primary attack of Princeton University is based on multiple A records [6]. They used Java applets for this. Once a victim accesses the domain of the attacker, the Java applet is loaded in the browser of the victim. The applet then

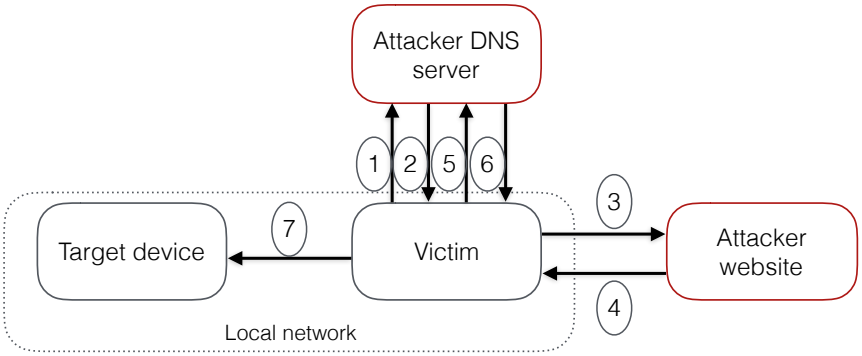

Fig. 2. Steps of a time varying DNS rebinding attack. The DNS request of the victim ① is answered by the attacker DNS server with his IP address and a short TTL ②. The victim's browser downloads malicious code from the attacker's website ③ and ④. When the malicious code is executed, the TTL has expired so that a new DNS request must be sent to the attacker's server ⑤. The response contains the IP address of the target device in the victim's local network ⑥. Thus, the request is redirected to the target device ⑦.

requests a subdomain of the attack domain. For this request, the attacker server provides a resource record set with IP addresses. The first entry in this record set must contain the internal IP address of the target device in the local network of the victim. The second IP address is identical to the IP of the attacker server. In this way, an external attacker can use a Java applet to implement interactive access to devices in the local network of the victim. It is exploited that a connection request is allowed by the Java system as soon as the IP address from which the Java applet was loaded appears in the resource record.

However, this attack is no longer feasible as DNS pinning has been introduced and the security policies for Java applets have been changed. Nowadays, an applet can only establish connections to the IP address from which it was originally loaded (Same Origin Policy) [8].

Time varying DNS In 2011 Roskind demonstrated that the Time-To-Live (TTL) of a DNS response is not trustworthy and that the mapping between a domain and IP address should be saved independently of that time (DNS Pinning). He introduced time-varying DNS rebinding attacks [14]. Figure 2 visualizes the steps during this attack.

In this attack, the DNS name server of the attacker responds to the request of the victim with a very short TTL, e.g., one second. The browser of the victim now uses this IP address to access the website of the attacker and downloads the HTML document together with malicious code. When the victim executes the malware code, an asynchronous connection request is made to a resource of the attacker server. At this time, the entry in the DNS cache of the browser with

the domain and IP address of the attacker is already deleted due to the very short TTL. Thus, to resolve the domain, a new DNS request must be triggered that the DNS server of the attacker responds with the private IP address of the target device in the network of the victim. In this way, the domain of the attacker in the DNS cache of the browser is assigned to the private IP address of the device by the victim. As a result, the asynchronous connection request is not sent to the Web server of the attacker, but the local network component of the victim. Thus, the attacker succeeds in establishing an interactive session to a device in the private network of the victim. The attack detection of routers can easily detect this attack method when a private IP address is resolved according to RFC1918 [13]. DNS pinning also prevents the attack in modern browsers. As a result, this simple attack is no longer exploitable.

The simple time-varying attack described above can be blocked by DNS pinning in the browser. The most straightforward strategy to bypass DNS pinning is to make the malicious script wait with the asynchronous connection request until the DNS entry expires in the cache of the browser. This trivial approach is called anti-DNS pinning [7]. In 2013, Dai and Resig showed that it is possible to significantly speed up the waiting period by flooding the DNS cache [4].

2.3 Countermeasures

DNS rebinding attacks have been known for a long time and so there are functioning countermeasures. On the server side, every web server in the local network can have its own authentication methods. Furthermore, communication with the web server should be secured by TLS; thus, no DNS rebinding attack is possible. The firewall settings should be such that requests from external host names must not be resolved with internal IP addresses. The DNS settings should also be configured so that external hostnames cannot be resolved with internal IP addresses. This adjustment is straightforward to do by using DNS rebinding protection mechanisms on many routers, e.g., dnsmasq uses this protection in the default settings. On the client side, browser extensions such as NoScript can be used when visiting web pages.

2.4 Requirements

In order to successfully perform a DNS rebinding attack today, several requirements must be fulfilled. We identified a total of six requirements and describe them in the following.

1. No transport layer security (TLS): If TLS is used, no DNS rebinding attack can be performed. This is because a TLS certificate is issued to a full hostname or a unique IP address. When a new connection request is made to the target device to perform a DNS rebinding attack, the TLS certificate verifies that the domain of the new connection matches the information of the certificate. Since the domain of the local target device differs from the information in the TLS certificate the TLS handshake fails and the connection request is rejected.

2. No authentication: If authentication is used on the application layer based on well-selected credentials, the attacker must first break them in order to perform a DNS rebinding attack successfully.
3. Visit the website of the attacker: The victim must visit the web page of the attacker to run malicious JavaScript in the background.
4. Dwell time: In addition to the fact that the victim must surf on the attacker's website itself, the victim must also stay on the website until the attack is successful.
5. IP address and port must be known: The attack targets are local network components with an open web server. An attacker must, therefore, know the IP address of the target device in the victim's private network and the port of the web server.
6. No specific countermeasures: The countermeasures presented in Section 2.3 may not be used.

3 Descriptions of experiments

In the following, we describe the experiments we conducted as part of our study. We start with the description of the Attacker Model, continue with the setup, and finally our measurement.

3.1 Attacker model

Primarily, DNS rebinding attacks aimed at classic network components such as routers, printers, or internal servers. We look at smart home devices. We investigate whether they are vulnerable and what possibilities a potential attacker has. We assume the following scenario for the following investigation: The victim stays on the website of the attacker long enough until the attack is completed, the attacker knows the internal IP in the local network, and the API endpoints of the device to be attacked.

The attacker model is reasonably realistic, as an attacker can use interesting content to trick the victim into spending the necessary time on a website, and the internal IP address of the target device may, e.g., be discovered by misconfigured information-leaking DNS servers [17].

3.2 Experimental setup

For the conducted experiments, we use a private network consisting of the smart home devices to be examined and the computer of the victim user. All devices use standard configurations and are connected to the Internet via a router. Figure 3 illustrates the experimental setup.

The used router is a mobile router RUT500 from Teltonika. The web interface can be reached under 192.168.1.1 from the local network. The computer of the victim has the local IP 192.168.1.181 and installed the Chrome browser version 63 for the investigations. Furthermore, there are four smart home devices to be

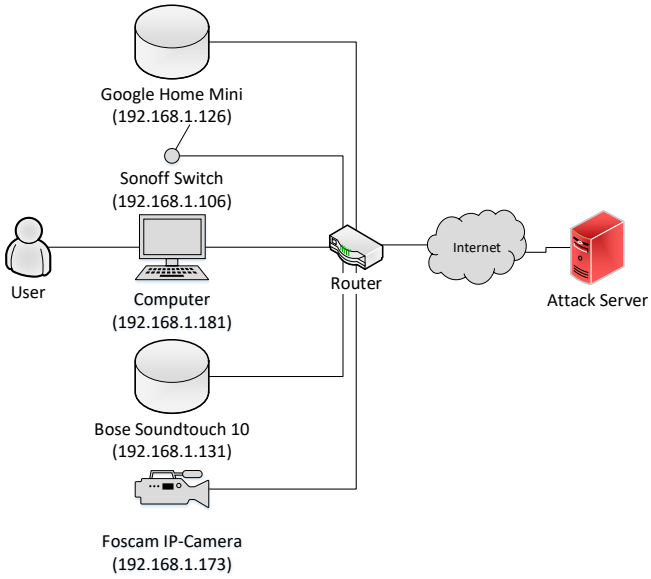

Fig. 3. Overview of experimental setup.

examined in the local network. We are examining a Google Home Mini, a Sonoff Basic, a Foscam IP Camera, and a Bose Soundtouch.

The Google Home Mini is a voice-controlled speaker, i.e., a smart assistant who is representative for other voice assistants in our study, such as the Alexa speakers. The Sonoff Basic is a smart switch that can be controlled via WLAN. The switch is flashed with the Tasmota firmware [16] and therefore representative for all devices with the Tasmota firmware. The Foscam IP camera is an IP camera with the exact designation FI9900p, and the Bose Soundtouch 10 is a WLAN controllable speaker.

We used the DNS rebinding attack framework Singularity of Origin of the NCC Group for the execution of the attacks [15]. This framework performs a time-varying DNS rebinding attack as already introduced in Section 2.2. As part of the work, we have accelerated an attack method of the framework by applying the DNS cache flooding technique presented by Dai and Resig [4] to the method used in the framework. In this way, the duration of the attack could be reduced from 60 seconds to 5 seconds.

3.3 DNS rebinding on websites in the wild

To complement our study, we examine websites and check whether they perform DNS rebinding attacks in the background. We performed this measurement by setting up a Ubuntu system and installing the DNS server dnsmasq. With DNS Rebinding Attack Protection enabled by default, dnsmasq reliably detects

Table 1. Overview of the results of the conducted DNS rebinding attacks against four different smart home devices. Three out of four devices were attacked successfully.

	Google Home	Sonoff Basic	Bose Soundtouch	Foscam
Vulnerable	✓	✓	✓	✗
Control of HW functions	✓	✓	✓	✗
Personal data	✓	✓	✓	✗
MAC	✓	✓	✓	✗
Location data	✓	✓	✓	✗
User name	✓	✗	✓	✗
Wifi information	✓	✓	✗	✗

whether private IP addresses are contained in DNS packets. Subsequently, we implemented a script that automatically visits websites and evaluates the log file from dnsmasq to detect DNS rebinding attempts.

4 Results

In this section, we present the results of our investigations. First, we describe the results of our conducted attacks on smart home devices. Second, we present the results of our brief measurement of DNS rebinding on popular websites.

4.1 Smart home devices

During the conducted tests it turned out that the router RUT500 is not vulnerable to DNS rebinding attacks due to the activation of the DNS rebinding protection of dnsmasq and therefore none of the devices behind the router. The manufacturer activates the protection that all incoming DNS packets with private IP addresses are directly blocked from dnsmasq by default. However, we noticed that requests reach localhost (127.0.0.1) and therefore services running on localhost might be vulnerable.

To enable the testing of the other smart home devices, we deactivated the DNS rebinding protection in the following. We summarized the results in Table 1 and detailed descriptions of the result are in the following paragraphs. The first row (Vulnerable) of the table indicates whether the device is potentially vulnerable or not. The following rows indicate what a potential attacker can achieve with a DNS rebinding attack on the particular device. We differentiate between the control of hardware functions, the extraction of personal data, finding out the MAC, the extraction of location data, finding out the user name, and getting further Wifi information.

Google Home Mini is potentially vulnerable to DNS rebinding attacks. An undocumented HTTP server was found on port 8008 and a Web API interface without authentication mechanisms. During a DNS rebinding attack, sensitive data can be extracted via the Web API. In addition, hardware functions of the device can be managed over the Internet.

Sonoff Basic with the Tasmota firmware and default configurations is also vulnerable to DNS rebinding attacks. The firmware uses a Web API interface. Therefore all Sonoff devices with the Tasmota firmware are potentially vulnerable. The active control of hardware functions is limited to toggling the relay of the switch. However, sensitive data such as usage habits and power consumption are readable.

Bose Soundtouch also has a Web API interface without authentication mechanisms and no TLS support. Thus, this device is also potentially vulnerable to DNS rebinding attacks. It provides access to hardware functions such as volume, as well as stored data such as MAC addresses of paired devices. Other Bose devices also use the same firmware. Therefore, we suspect that these devices are vulnerable as well.

Foscam FI9900p is the only tested device not potentially easily vulnerable to DNS rebinding attacks. When setting up the device, the user is forced by the vendor to set a username and password. When connecting to the web service of the camera, the user has to authenticate with his credentials. For this reason, a DNS rebinding attack is only possible if the attacker knows the credentials or is able to break the authentication.

4.2 Measuring DNS rebinding attempts on popular websites

This measurement led to the result that none of the top 100 Alexa sites that use HTTP execute DNS rebinding and is thus not surprising. However, the approach can be used to check a more significant number of websites.

5 Discussion

The results demonstrate that four out of five examined devices are vulnerable to DNS rebinding attacks. However, it should be noted that for the selection of the test devices, devices with open Web services were explicitly selected. Accordingly, this selection of test devices cannot be used to make a statement about all smart home devices. However, since the total number of smart home devices is very high, the number of potentially vulnerable devices should still be a non-negligible amount. In addition, our insights also confirm the results of Acar et al. [1] that many IoT devices are vulnerable to DNS rebinding attacks.

Furthermore, we have seen that for a successful DNS rebinding attack, many requirements have to fit, which limits it as a real-world threat. However, as soon as the attack is feasible, it can have serious consequences. For this reason, it is important to check which conditions have to be fulfilled and to evaluate the applied countermeasures. The results of the study indicate that manufacturers do not focus sufficiently on the security of their products when developing them. The potential vulnerability of smart home devices highlights the lack of security of IoT devices, which has repeatedly attracted media attention in recent years. In

many cases, inadequate authentication was the cause of attacks. The well-known Mirai botnet [12], for example, targeted IoT devices that were operated with standard credentials and accessible to the public. Manufacturers of smart home devices could protect their customers by introducing mandatory authentication at the application layer. A mandatory change to the authentication credentials was implemented for the Foscam IP camera, for example, so that no unauthorized access to web services can be made.

6 Related work

Since the attack technique is long-established, there is much work in this area. The first publication on DNS rebinding attacks was published in 1996 [5], after which further papers were presenting new variants of the attack [4, 9, 11]. Corresponding work with countermeasures also exists [8, 10]. One of the most recent papers deals with DNS rebinding attacks on IoT devices [1]. Especially the analysis of new as well as already known attacks on IoT devices is important, as the Mirai Botnet demonstrates [12].

We follow the course of the history of work on DNS rebinding and rely in particular on the current paper in the field of IoT devices. We perform the attack on smart home devices and give a rough overview of vulnerable devices. In addition, we discuss the attack surface and conclude that due to various countermeasures, despite the vulnerability of the devices themselves to DNS rebinding attacks, it is not likely to become a victim.

7 Conclusion

We showed that DNS rebinding attacks in the world of IoT are reviving and four out of five devices tested are vulnerable (three out of four smart home devices and a router). Nevertheless, possible attack targets are limited, as many countermeasures exist and several conditions must be fulfilled and as routers often already contain DNS rebinding attack detection. The manufacturers of smart home devices, therefore, rely on the router firewalls to provide security against this attack. In many cases, this will also be the case, but ideally, the smart home devices themselves should also be protected. In summary, this work suggests that when connecting all things, one needs to keep in mind known weaknesses and issues in order not to become vulnerable to attacks that have been known for years.

Future work in the DNS rebinding attacks IoT area can extend the results of our study by testing further smart home devices and also check a more significant number of websites to see whether a DNS rebinding attack is being carried out in the wild.

Acknowledgment

We would like to thank the anonymous reviewers for their valuable feedback.

References

1. G. Acar, D. Y. Huang, F. Li, A. Narayanan, and N. Feamster. Web-based attacks to discover and control local iot devices. In *Proceedings of the 2018 Workshop on IoT Security and Privacy*, 2018.
2. DNS Rebinding Exposes Half a Billion Devices in the Enterprise. `https://armis.com/dns-rebinding-exposes-half-a-billion-iot-devices-in-the-enterprise/`. Accessed: 2019-06-06.
3. CVE - Common Vulnerabilities and Exposures. `https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=DNS+Rebinding`. Accessed: 2019-06-06.
4. Y. Dai and R. Resig. Firedrill: Interactive {DNS} rebinding. In *7th {USENIX} Workshop on Offensive Technologies*, 2013.
5. D. Dean, E. W. Felten, and D. S. Wallach. Java security: From hotjava to netscape and beyond. In *IEEE Symposium on Security and Privacy*, 1996.
6. DNS Attack Scenario (February 1996). `http://sip.cs.princeton.edu/news/dns-scenario.html`. Accessed: 2019-06-06.
7. J. Grossman, S. Fogie, R. Hansen, A. Rager, and P. D. Petkov. *XSS attacks: cross site scripting exploits and defense*. Syngress, 2007.
8. C. Jackson, A. Barth, A. Bortz, W. Shao, and D. Boneh. Protecting browsers from dns rebinding attacks. In *ACM Conference on Computer and Communications Security (CCS)*, 2007.
9. M. Johns, S. Lekies, and B. Stock. Eradicating DNS rebinding with the extended same-origin policy. In *USENIX Security Symposium*, 2013.
10. M. Johns and J. Winter. Protecting the intranet against javascript malware and related attacks. In *Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA)*, 2007.
11. C. Karlof, U. Shankar, J. D. Tygar, and D. Wagner. Dynamic pharming attacks and locked same-origin policies for web browsers. In *ACM Conference on Computer and Communications Security (CCS)*, 2007.
12. C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas. Ddos in the iot: Mirai and other botnets. *Computer*, 50(7):80–84, 2017.
13. Y. Rekhter, B. Moskowitz, D. Karrenberg, G. de Groot, and E. Lear. Address Allocation for Private Internets. RFC 1918, RFC Editor, February 1996.
14. J. Roskind. Attacks Against the Netscape Browser, 2001. Talk at the RSA Conference.
15. Singularity of Origin. `https://github.com/nccgroup/singularity`. Accessed: 2019-06-06.
16. Fonoff-Tasmota. `https://github.com/arendst/Sonoff-Tasmota`. Accessed: 2019-06-06.
17. D. Tatang, C. Schneider, and T. Holz. Large-scale Analysis of Infrastructure-leaking DNS Servers. In *Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA)*, 2019.