DNS Spoofing in Local Networks Made Easy

Nikhil Tripathi, Mayank Swarnkar and Neminath Hubballi
Discipline of Computer Science and Engineering, School of Engineering
Indian Institute of Technology Indore
{phd1401101002, phd1401101001, neminath}@iiti.ac.in

***Abstract*—Domain Name System (DNS) is a central protocol of the internet and provides a way to resolve domain names to their corresponding IP addresses. It is one of the most critical protocols being used in the internet. However, DNS is known to be vulnerable to a popular attack called DNS poisoning. Fortunately, DNS poisoning has become difficult to launch due to introduction of techniques like source port and query identification value randomization aftermath of Kaminsky attack. In this paper, we propose a targeted DNS spoofing attack that exploits a vulnerability present in DHCP server-side IP address conflict detection technique to prevent a genuine DHCP server from offering network parameters; while sending a fake offer on its own. We discuss how proposed attack can target even a single victim client also without affecting other clients. We test the effectiveness of proposed attack in a real network setup and report the results. Further, we discuss how known detection and mitigation techniques are unable to detect the attack.**

I. Introduction

Domain Name System (DNS) [18] is one of the critical protocols for efficient working of internet applications. It provides a way to match a domain name to its corresponding IP address, thus, mitigating the need of remembering IP address of a web server. DNS is also said to be a database of resource records as it maintains not only the host name to IP mapping but also other records such as nameserver, mail exchanger, etc. However, this protocol is vulnerable to attacks like DNS cache poisoning [16], DNS amplification attacks [17] and DNS query flood [25]. These attacks either compromise the confidentiality and integrity of end users by altering their traffic or cause Denial-of-Service (DoS) by targeting availability of resources. Various tools and techniques [5], [7] are available in the wild to launch such attacks. However, with the release of security fixes [6], [1], it has become quite difficult for a malicious entity to poison DNS cache of nameservers. Moreover, DNS amplification and query flood attacks require a large amount of malicious client's bandwidth and thus can be detected easily. Also, various third party vendors such as Cloudflare [2] provide DoS/DDoS mitigating solutions for smooth business running of enterprises.

In this paper, we propose a targeted DNS spoofing attack that aims to deceive end clients instead of poisoning DNS server's cache. The proposed attack exploits a loophole present in DHCP server-side IP address conflict detection scheme [12], [14], [21]. RFC 2131 [12] mentions that before offering an IP, a DHCP server should probe it to make sure no other client is using it. A malicious client can exploit this vulnerability by sending fake replies to such probes due to which the DHCP server will not be able to offer IP address and other network configuration parameters to victim client. At the same time, malicious client offers the desired network configuration parameters and thus, victim client ultimately starts using these parameters for further communication. This finally allows malicious client to alter or redirect the victim client's traffic as and when required. We show later in Section III-B how proposed attack is easier to launch and more effective as compared to previously known DNS attacks.

Rest of the paper is organized as follows. In Section II, we describe the working of DHCP and DNS protocols and some popular attacks against DNS protocol. The proposed targeted DNS spoofing attack and its comparison with previously known attacks is discussed in Section III. We present experimental details and results in Section IV. In Section V, we discuss some known detection and mitigation techniques. Finally, the paper is concluded in Section VI.

II. Background

In this section, we briefly discuss how an end client uses DHCP and DNS protocols in order to communicate with other entities, e.g. a web server. We also discuss previously known attacks against DNS protocol in this section.

A. DHCP and DNS Protocols

As soon as a client joins a network, it obtains IP address and other network configuration parameters from DHCP server(s) equipped within the network. The messages exchanged between the client and DHCP server while obtaining IP address and other network parameters are shown in Figure 1. These network parameters include local DNS server's IP address also. In case client requires to resolve a domain name into corresponding IP address, it sends a DNS query to the DNS server. On client's behalf, this server communicates with one of the thirteen root nameservers and other authoritative nameservers in order to resolve the required domain name. This process of domain name resolution is shown in Figure 2.

B. Attacks Against DNS Protocol

DNS is vulnerable to three popular types of attacks. In this subsection, we briefly describe each of these attacks.

1) Kaminsky DNS Cache Poisoning: Dan Kaminsky proposed an approach to hijack the authority nameserver records [16] by first forcing the victim nameserver to trigger a DNS resolution query for a target domain and then sending forged DNS responses so that victim nameserver accepts the response assuming it is sent by

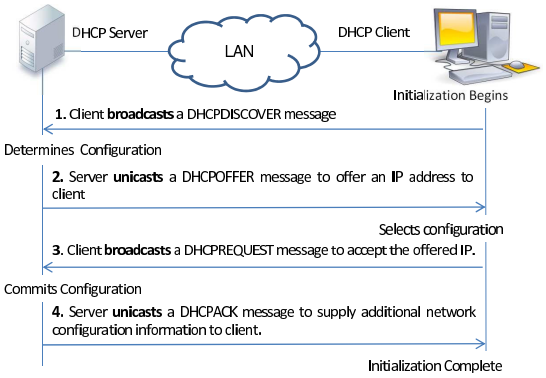

Fig. 1: IP Address Allocation Process

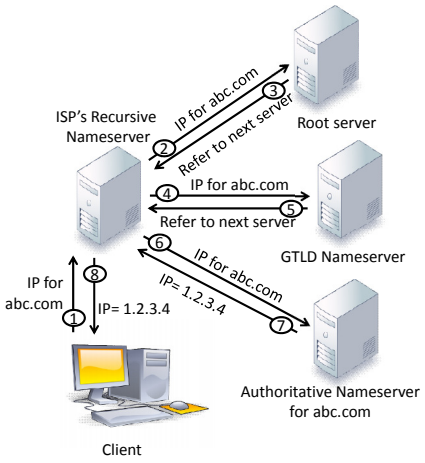

Fig. 2: Domain Name Resolution

authoritative nameserver for the target domain. However, to launch the attack, malicious client needs prior knowledge of which port number and query identification value is currently in use by nameserver. Thus, if both 16 bit port number and 16 bit query identification values are randomized, it becomes extremely difficult for malicious client to guess correct combination before the victim nameserver reaches the genuine authoritative nameserver.

2) DNS Amplification Attack: DNS amplification [17] is typically a DDoS flooding attack where malicious client makes use of several zombies to generate small DNS queries with forged source IP addresses. These queries are meant to generate large volume of network traffic as DNS response messages are comparatively much larger than the DNS query messages. This large volume of network traffic is directed towards the victim in order to consume its resources. This attack makes use of open DNS resolvers in order to reflect and amplify the network traffic. Since this attack generates a large amount of DNS response traffic on the victim host, various cloud based techniques [2] are able to detect and mitigate this attack as soon as DNS traffic crosses a predefined threshold.
3) DNS Query Flooding: This attack [25] is somewhat similar to the DNS amplification attack as it is also a type of DDoS attack. However, to launch this attack, malicious client floods the victim nameserver using a large number of DNS queries so that it is not able to resolve genuine clients' DNS queries. Since this attack also generates large volume of DNS traffic, it can easily be detected if DNS traffic crosses a predefined threshold.

III. Proposed Targeted DNS Spoofing

In this section, we propose the targeted DNS spoofing attack and discuss its working. We also discuss how the proposed attack is different and more effective from the previously known attacks against DNS protocol. We use the notations shown in Table I to describe the attack.

TABLE I: Notations

IP_OFFERED_GENUINE	IP address selected by genuine DHCP server to be offered to client
IP_OFFERED_MALICIOUS	IP address offered by malicious client to victim client
MAC_MALICIOUS	MAC address of malicious client
IP_SERVER	DHCP server's IP address
MAC_SERVER	DHCP server's MAC address
ARP_REQ	ARP Request
ARP_REPLY	ARP Reply
ICMP_REQ	ICMP Ping Request
ICMP_REPLY	ICMP Ping Reply
D_N	Domain Name being queried
FAKE_WEB_SERVER_IP	IP address of web server hosting fake web page
REAL_WEB_SERVER_IP	IP address of web server hosting D_N

A. Attack Description

We consider a network topology similar to the one shown in Figure 3. In this network, there are seven entities namely a DHCP server, a malicious client, a victim client, a switch, genuine DNS server, genuine web server and a fake web server. Various messages exchanged between these entities during attack are elaborated below:

1) DHCPDISCOVER Message by Victim Client: As soon as victim client joins the network, it broadcasts a DHCPDISCOVER message in order to locate DHCP server(s) within the network.
2) ARP_REQ/ICMP_REQ Probe by DHCP Server: Once a DHCP server receives DHCPDISCOVER message sent in previous step, it broadcasts either an ARP probe or ICMP probe depending on its implementation to check if IP address, IP_OFFERED_GENUINE, chosen to offer is already in use. If DHCP server is designed to send ARP_REQ, source MAC, source IP, destination MAC and target IP are set to MAC_SERVER, IP_SERVER, "ff:ff:ff:ff:ff:ff" and IP_OFFERED_GENUINE respectively. In case DHCP server is designed to send ICMP_REQ, it sends an ICMP_REQ with source

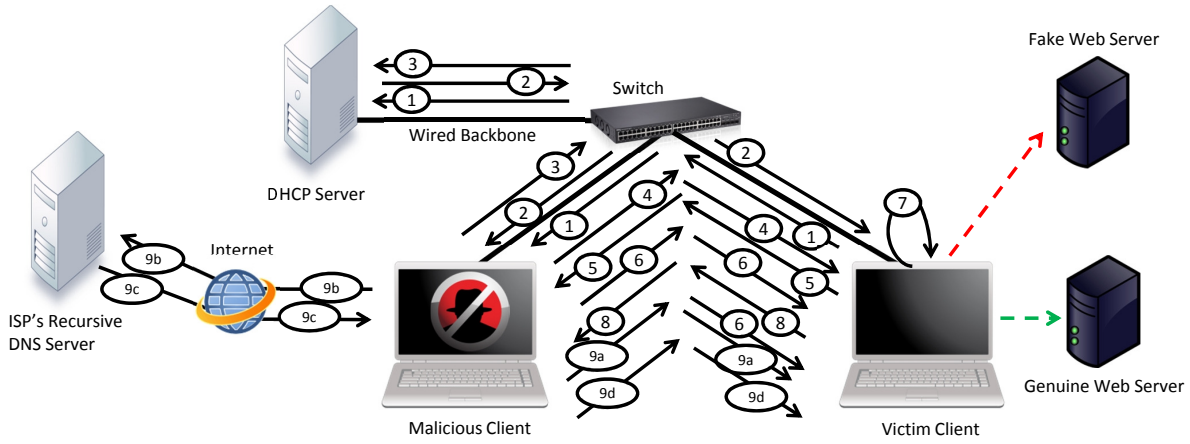

Fig. 3: Attack Description

and destination IP addresses as IP_SERVER and IP_OFFERED_GENUINE respectively.

3) ARP_REP/ICMP_REP Reply by Malicious Client: If DHCP server sends ARP_REQ, malicious client responds back with a fake ARP_REP having its source MAC, source IP, destination MAC and target IP addresses as MAC_MALICIOUS, IP_OFFERED_GENUINE, MAC_SERVER and IP_SERVER respectively. If DHCP server sends ICMP_REQ, malicious client responds back with a fake ICMP_REP having its source IP and destination IP as IP_OFFERED_GENUINE and IP_SERVER respectively. Due to this reply, DHCP server is not able to offer IP_OFFERED_GENUINE and thus, it selects next available IP address from the pool to offer it to victim client. DHCP server continuously tries to offer an IP address unless it receives the DHCPREQUEST message[1] broadcasted by the victim client in 5^{th} step.
4) DHCPOFFER Message by Malicious Client: After sending fake probe reply in previous step, malicious client immediately offers an IP address, IP_OFFERED_MALICIOUS, to victim client by sending a DHCPOFFER message. IP_OFFERED_MALICIOUS belongs to the same IP address pool range as configured on genuine DHCP server.
5) DHCPREQUEST Message by Victim Client: As soon as victim client receives DHCPOFFER message, it sends a DHCPREQUEST message to malicious client in order to request to use IP_OFFERED_MALICIOUS. Since DHCPREQUEST message is a broadcast message, genuine DHCP server also receives it. This message informs genuine DHCP server to stop trying to offer an IP address to victim client as it has already been offered an IP address from malicious client.
6) DHCPACK Message by Malicious Client: In response to the DHCPREQUEST message sent by victim client in previous step, malicious client sends a DHCPACK message confirming the allocation of IP_OFFERED_MALICIOUS to victim client. Other network configuration parameters like default gateway's and DNS server's IP address is also sent in this DHCPACK message. In order to receive traffic (including DNS queries) coming from and going to victim client, malicious client claims its own IP address as DNS server's IP address and/or default gateway address.
7) IP Address and Other Network Parameters Configuration by Victim Client: As soon as victim client receives DHCPACK message sent by malicious client in previous step, it configures its interface with IP_OFFERED_MALICIOUS and other network parameters.
8) DNS Query by Victim Client: To resolve a domain name, D_N, to corresponding IP address, victim client sends a DNS query with the destination IP address as that of malicious client.
9) DNS Response by Malicious Client: Malicious client checks the category to which D_N belongs.

   a) If D_N is of interest to malicious client like financial or commercial website, malicious client sends forged DNS response so as to redirect victim client to FAKE_WEB_SERVER_IP hosting website exactly similar to D_N.
   b) If it is not of interest to it like a simple search engine or educational institution website, malicious client may forward the DNS query to genuine ISP's DNS server for resolution. This is because there is no incentive for malicious client to send forged DNS responses for domains belonging to these categories.
   c) If the DNS query is forwarded to ISP server,

[1]DHCPREQUEST message is usually broadcasted by a client in order to inform other DHCP servers within the network that it has been offered an IP address from one of the DHCP servers and other servers must now stop offering IP address to it.

it resolves the query by communicating with other DNS servers[2] and returns the resolved IP address, GENUINE_WEB_SERVER_IP, to malicious client.

d) Malicious Client finally sends the DNS response back to victim client so as to redirect it to GENUINE_WEB_SERVER_IP hosting genuine website.

B. Comparing Targeted DNS Attack with Other DNS based Attacks

Our proposed attack is effective and easier to launch as compared to other DNS based attacks in following way:

- The proposed targeted DNS spoofing attack does not involve usage of several thousands of computers (zombies) as required in case of DNS amplification attack. Moreover, proposed attack is much more stealthier than DNS amplification attack and DNS query flooding attack as the proposed attack sends just few fake probe responses to DHCP server. This traffic is almost negligible to the traffic generated in case of DNS amplification attack and DNS query flooding attack. Kaminsky DNS cache poisoning also generates comparatively large amount of DNS traffic while guessing the correct query identification value and source port number that victim nameserver is using.
- Kaminsky DNS cache poisoning targets a nameserver due to which all those clients which are using the targeted nameserver are affected due to wrong domain name resolution. However, attack proposed in this paper can target even a specific client also instead of targeting all the clients within the network. In Section III-D, we discuss how malicious client can target a specific client instead of targeting all the clients at the same time.
- DNS servers, these days, use source port number and query identification value randomization to communicate with root servers and other nameservers. Thus using Kaminsky DNS cache poisoning, it is extremely difficult to guess the correct source port and query identification value before victim nameserver reaches the genuine authoritative nameserver. However, there is no such limitation in case of proposed attack.

C. Comparing Targeted DNS Spoofing Attack with Rogue DHCP Server based DNS Spoofing

DNS spoofing can also be performed by configuring a rogue DHCP server [8] within the network and offering IP addresses and other network configuration parameters to victim clients from this server. In this method, since genuine DHCP server is not stopped from offering an IP address, it also sends a DHCPOFFER message to victim client. This leads to a race condition between malicious client's and genuine DHCP server's DHCPOFFER message. As a result, DNS spoofing is possible only if rogue DHCP server's offer reaches to victim client earlier than genuine DHCP server's offer. Since it is not always possible, victim client is able to configure genuine network configuration parameters. This leads to failure of DNS spoofing attack. Moreover, various detection techniques like [8], [20] raise an alarm in case more than one DHCPOFFER messages are received for a DHCPDISCOVER message. Targeted DNS spoofing attack, on the other hand, prevents the genuine DHCP server from offering IP address due to which no race condition occurs. Thus, DNS spoofing can be launched effectively without any issue using the proposed method. As a result, the victim client is easily redirected to the desired web server.

D. Targeting a Specific Client

Using targeted DNS spoofing, malicious client can target a specific client also without affecting other clients. To do so, malicious client first captures the DHCPDISCOVER message sent by the client to be targeted. Since this message is the broadcast one, malicious client also receives this message. On receiving this message, malicious client sniffs probe request to capture the probe sent by DHCP server for precautionary checking of IP address usage. As soon as malicious client receives the probe, it immediately sends back a probe response to prevent DHCP server from offering IP address to target client. Malicious client further offers IP address and other network parameters to victim client, thereby, completing the attack procedure. In this way, malicious client can easily target a specific client as well.

IV. Experiments and Discussion

In order to demonstrate the execution of proposed attack, we created a network setup similar to the one shown in Figure 4. There were few entities within the network setup, a malicious client, few victim clients and a D-Link DIR-600M router having built-in DHCP server. The server was configured with a pool of 14 IP addresses ranging from 192.168.0.1 to 192.168.0.14. The server used ARP requests[3] for probing in order to detect IP address conflict. Depending on the DHCP software implementation, ICMP requests can also be used for probing purpose. Table II shows different DHCP servers and the probe types they use to detect IP address conflicts. The malicious and victim clients were running Ubuntu 16.04 and Windows 7 Service Pack 1 operating systems respectively. We configured an ISC DHCP server [4] on malicious client to offer IP address and other network configuration parameters to victim client. The address pool on this server was also ranging from 192.168.0.1 to 192.168.0.14 so that victim client could communicate with other on-link devices without any issue. However, this DHCP server was configured to allot malicious client's IP address as DNS server's IP address to victim client. We wrote two C programs which were running on malicious client to launch the proposed attack. First program was used to sniff ARP requests coming from DHCP server's source IP address and also to send spoofed ARP replies. These

[2]For sake of simplicity, we do not show ISP's DNS server's interaction with other nameservers for domain resolution.

[3]The vendors of router implemented the DHCP software to use ARP requests instead of ICMP for probing purpose.

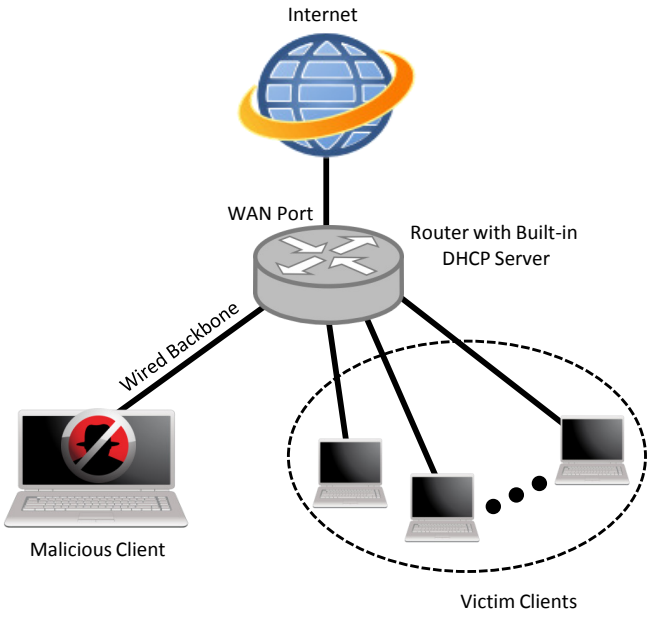

Fig. 4: Network Setup

TABLE II: Different DHCP Servers and the Probe Types

Vendor	Probe Type
Netgear N150 Router (inbuilt DHCP)	ARP Request
D-Link DIR-600M N150 Router (inbuilt DHCP)	ARP Request
ISC DHCP server	ICMP Ping Request
Microsoft Windows Server 2008	ICMP Ping Request

two modules were implemented as two threads. One thread executed the sniffing module while other thread generated spoofed ARP replies. Second program was used to send either fake or correct DNS response depending on the domain type. To do so, we implemented three threads in this program. First and second threads were used to sniff DNS queries from victim client and send fake DNS responses respectively while third thread was used to communicate with ISP's DNS server for resolving the victim client's DNS query and also to send genuine DNS response back to victim client.

Using this setup, we launched the proposed attack by targeting upto 5 victim clients at a time. We created 5 scenarios such that only one victim client is targeted in first scenario, two victim clients in second scenario and so on. We should notice that malicious client requires at least three messages to target a victim client. These messages are DHCPOFFER, DHCPACK and a fake DNS reply. Along with these messages, malicious client also needs to send fake probe replies to prevent genuine DHCP server from offering an IP address to victim client. From our experiments, we observed that DHCP server could perform maximum of three trials to offer IP address to victim client before it receives DHCPREQUEST message broadcasted by victim client. Thus to prevent DHCP server from offering the IP address, malicious client requires to send upto three fake probe replies. Figure 5 shows the

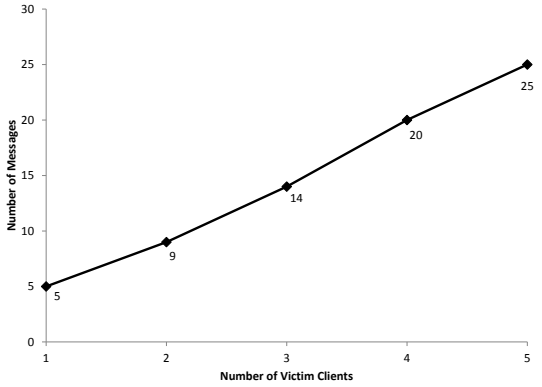

Fig. 5: Number of Messages required to Target Different Number of Clients

Fig. 6: Victim Host redirected to Another Web Server

number of messages to be sent by malicious client in order to target different number of victim clients. Figure 6 shows the screen capture of one of the victim clients that tried to access the domain $www.iiti.ac.in$. However, it was redirected to another sample web server hosted by us on internet. Thus, we can notice that the proposed attack can successfully deceive a victim client by launching targeted DNS spoofing attack.

V. Detection and Prevention

In this section, we discuss some popular detection and mitigation techniques which are relevant to DNS poisoning attack. We also name few known techniques to detect unauthenticated DHCP messages and IP conflicts in a local network.

A. Mitigating DNS Cache Poisoning Attack

To detect forged DNS response, DNSSEC [9] was proposed that involves digitally signing the DNS responses to authenticate and protect their integrity. However, DNSSEC does not provide end-to-end encryption due to which techniques like man-in-the-middle can still make victim clients connect to malicious hosts [10]. Moreover, DNSSEC significantly increases

the computational burden, space usage and huge consumption of bandwidth due to large DNS responses. This increased bandwidth consumption due to large responses also opens door for DNS amplification attack [10]. Due to these reasons, DNSSEC is still not deployed widely on the internet. Various other cryptographic techniques [15] are proposed to prevent DNS poisoning attacks, however, the limitations like involvement of computationally expensive tasks still exist which hinder their wider adoption. In [19], authors proposed an entropy-based detection scheme that can detect the poisoning attack only if a single DNS cache server is attacked. Another technique [23] can detect the distributed DNS poisoning as well. These approaches are based on the assumption that the IP entropy sequences are stationary under normal cases. However, this assumption does not hold true for different time periods of a day [24]. In another work [24], authors proposed a detection method for the case that the sequence of entropy is non-stationary and follows a dynamic behaviour. In particular, authors first modeled the entropy sequence by a state space equation and then used kalman filter for the detection purpose. Few other schemes [26], [13] are proposed which monitors the DNS traffic on DNS resolvers. All these techniques are proposed in order to detect attacks that involve poisoning DNS cache at server side. However, we target the client itself where the DNS query originates. As a result, the resultant DNS traffic received at server side possesses similar characteristic as that of normal DNS traffic. Thus, it is difficult to detect the proposed attack using these techniques.

B. Detecting Unauthenticated DHCP messages and IP Conflicts

Proposed DNS spoofing attack can be prevented with any method that mitigates DHCP starvation attack. There are techniques like cryptographic methods [11] which can prevent these attacks but they are rarely implemented due to implementation complexity. DHCP Snooping [3] filters DHCPOFFER and DHCPACK coming from an interface of a switch which is not trusted, however, they are ineffective in wireless networks.

Authors in [14], [22] proposed schemes to detect IP address conflicts by comparing normal DHCP traffic profile with the profiles generated in different time windows. Since the proposed attack does not result into high DHCP traffic, it is difficult to detect the attack using these techniques.

VI. Conclusion

DNS has been a critical protocol of the internet architecture since last three decades. Vulnerabilities in the protocol are extensively researched in order to make the protocol more robust. Due to this, attacks like DNS poisoning and DNS amplification has become difficult to launch and can be detected easily. In this paper, we proposed a targeted DNS spoofing attack that exploits vulnerability present in DHCP server-side IP address conflict detection technique. We showed that the proposed attack is easier to launch and much stealthier as compared to previously known attacks. We discussed how known detection and mitigation strategies are ineffective to counter the attack. We hope that this work will motivate researchers in the security community to develop robust detection and mitigation techniques in order to detect the proposed attack and overall make DNS a secure protocol.

References

[1] An Illustrated Guide to the Kaminsky DNS Vulnerability. http://unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html.

[2] Cloudflare, Inc. https://www.cloudflare.com/.

[3] DHCP Snooping. http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst6500/ios/12-2SX/configuration/guide/book/snoodhcp.html.

[4] ISC DHCP. https://www.isc.org/downloads/dhcp/.

[5] Saddam. https://github.com/OffensivePython/Saddam.

[6] Secure the Server Cache Against Names Pollution. https://technet.microsoft.com/en-us/library/cc772349(v=ws.11).aspx.

[7] Tsunami - DNS Amplification Attack Tool. https://www.infosec-ninjas.com/tsunami.

[8] M. Agarwal, S. Biswas, and S. Nandi. Discrete Event System Framework for Fault Diagnosis with Measurement Inconsistency: Case Study of Rogue DHCP Attack. *IEEE/CAA Journal of Automatica Sinica*, PP(99):1–18, 2017.

[9] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Protocol Modifications for the DNS Security Extensions. RFC 4035, 2005.

[10] A. Cowperthwaite and A. Somayaji. The Futility of DNSSec. In *Annual Symposium Information Assurance (ASIA)*, 2010.

[11] K. de Graaf, J. Liddy, P. Raison, J. Scano, and S. Wadhwa. Dynamic Host Configuration Protocol (DHCP) Authentication using Challenge Handshake Authentication Protocol (CHAP) Challenge, 2013. US Patent 8,555,347.

[12] R. Droms. Dynamic Host Configuration Protocol. RFC2131, 1997.

[13] A. Herzberg and H. Shulman. *Unilateral Antidotes to DNS Poisoning*, pages 319–336. 2012.

[14] N. Hubballi and N. Tripathi. A Closer Look into DHCP Starvation Attack in Wireless Networks. *Computers & Security*, 65:387 – 404, 2017.

[15] M. A. Hussain, H. Jin, Z. A. Hussien, Z. A. Abduljabbar, S. H. Abbdal, and A. Ibrahim. Enc-DNS-HTTP: Utilising DNS Infrastructure to Secure Web Browsing. *Security and Communication Networks*, 2017, 2017.

[16] D. Kaminsky. Its the End of the Cache As We Know It. In *Black Hat Conference*, 2008.

[17] D. C. MacFarland, C. A. Shue, and A. J. Kalafut. The Best Bang for the Byte: Characterizing the Potential of DNS Amplification Attacks. *Computer Networks*, 116:12 – 21, 2017.

[18] P. Mockapetris. Domain Names - Implementation and Specification. RFC 1035, 1987.

[19] Y. Musashi, M. Kumagai, S. Kubota, and K. Sugitani. Detection of Kaminsky DNS Cache Poisoning Attack. In *2011 4th International Conference on Intelligent Networks and Intelligent Systems*, pages 121–124, 2011.

[20] R. Spacil, J. Ikonen, and J. Porras. Forcing Usage Rules in Public Wireless LANs. In *27th Annual IEEE Conference on Local Computer Networks, 2002. Proceedings. LCN 2002.*, pages 415–420, 2002.

[21] N. Tripathi and N. Hubballi. Exploiting DHCP Server-side IP Address Conflict Detection: A DHCP Starvation Attack. In *2015 IEEE International Conference on Advanced Networks and Telecommuncations Systems (ANTS)*, pages 1–3, 2015.

[22] N. Tripathi and N. Hubballi. A Probabilistic Anomaly Detection Scheme to Detect DHCP Starvation Attacks. In *2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pages 1–6, 2016.

[23] H. Wu, X. Dang, L. Wang, and L. He. Information Fusion-based Method for Distributed Domain Name System Cache Poisoning Attack Detection and Identification. *IET Information Security*, 10:37–44, 2016.

[24] H. Wu, X. Dang, L. Zhang, and L. Wang. Kalman Filter based DNS Cache Poisoning Attack Detection. In *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 1594–1600, 2015.

[25] S. T. Zargar, J. Joshi, and D. Tipper. A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks. *IEEE Communications Surveys & Tutorials*, 15(4):2046–2069, 2013.

[26] B. Zdrnja, N. Brownlee, and D. Wessels. Passive Monitoring of DNS Anomalies. In *Proceedings of the 4th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, DIMVA '07, pages 129–139. Springer-Verlag, 2007.