Domain Name Server Spoofing Using Ettercap And Dnsmasq

Adithya Nair, Avighna Reddy Katipally, P Ananthapadmanabhan Nair
Department Of Computer Science And Engineering
Amrita School Of Computing, Bengaluru
Amrita Vishwa Vidyapeetham, India
bl.en.u4aid23002@bl.students.amrita.edu
bl.en.u4aid23005@bl.students.amrita.edu
bl.en.u4aid23036@bl.students.amrita.edu

***Abstract*—The Domain Name System (DNS) represents a foundational yet critically vulnerable component of global internet infrastructure, serving the essential function of translating human-readable domain names into machine-addressable IP addresses. Despite its ubiquity and fundamental role in facilitating internet communication, DNS remains susceptible to sophisticated exploitation techniques that can compromise network security and user safety. This comprehensive research paper presents an in-depth technical exploration of DNS spoofing, a sophisticated network attack methodology that allows malicious actors to manipulate DNS resolution processes. By establishing a controlled experimental environment using virtualized network resources, we demonstrate the technical intricacies of DNS spoofing attacks, focusing on their implementation, potential impact, and strategic mitigation approaches. Our experimental methodology leverages virtualization technologies, specifically Oracle VirtualBox [1], and employs specialized cybersecurity tools including Kali Linux, Metasploitable, dnsmasq, ettercap, and Wireshark. Through meticulously crafted experiments, we systematically capture, analyze, and deconstruct DNS spoofing attack vectors, providing researchers and cybersecurity professionals with granular insights into these network vulnerabilities. The research not only illustrates the technical mechanics of DNS spoofing but also critically evaluates multiple mitigation strategies. These include implementing static ARP entries, enforcing trusted DNS server configurations, and exploring encrypted DNS query mechanisms such as DNS-over-HTTPS. By presenting a holistic view of the attack landscape, this study aims to advance understanding of DNS infrastructure vulnerabilities and provide actionable strategies for enhancing network security defenses.**

***Index Terms*—DNS Spoofing, Domain Name System, Cybersecurity, Network Exploits, Virtual Machines, Pentesting**

I. Introduction

The Domain Name System (DNS) serves as a critical architectural backbone of internet communication, functioning as a distributed, hierarchical naming system that translates human-readable domain names into machine-processable IP addresses. This fundamental protocol enables seamless internet navigation, allowing users to access online resources through intuitive, memorable domain names rather than complex numerical IP addresses. Implemented through sophisticated software systems like BIND (Berkeley Internet Name Domain) [2], DNS operates through a complex network of interconnected servers that cache and propagate domain name information. However, this critical infrastructure is inherently vulnerable to various sophisticated exploitation techniques, with DNS spoofing emerging as one of the most prevalent and potentially devastating attack methodologies. DNS spoofing, alternatively termed DNS cache poisoning, represents a sophisticated network attack where malicious actors manipulate DNS record mappings to redirect network traffic toward unauthorized or malicious IP addresses. The potential consequences of such attacks are profound and multifaceted, including:

1) Compromising data integrity by intercepting and potentially modifying network communications
2) Facilitating advanced phishing campaigns by redirecting users to fraudulent websites
3) Enabling complex man-in-the-middle attacks that can intercept sensitive information
4) Potentially breaching organizational network security frameworks

This research paper presents a comprehensive, technically rigorous examination of DNS spoofing through a carefully constructed experimental framework. By utilizing virtualization technologies and specialized cybersecurity tools, we aim to provide an empirical demonstration of DNS spoofing mechanics, attack vectors, and potential mitigation strategies. The experimental setup employs Oracle VirtualBox [1] to create a controlled network environment, with a Kali Linux virtual machine serving as the attack platform and a Metasploitable virtual machine representing a vulnerable target system. Through the strategic deployment of tools like dnsmasq for rogue DNS server establishment and ettercap for network traffic manipulation, we simulate a realistic DNS spoofing scenario targeting the hypothetical domain grouptwo.com. By meticulously capturing and analyzing network traffic using Wireshark, our research offers unprecedented insights into the technical execution of DNS spoofing attacks. Moreover, we systematically evaluate and propose multiple mitigation techniques, including static ARP entry configuration, trusted DNS server enforcement, and the implementation of encrypted DNS query protocols like DNS-over-HTTPS [3] The primary objectives of this research are threefold:

1) Demonstrate the technical mechanics of DNS spoofing attacks
2) Analyze the potential network security implications of such exploits
3) Provide actionable strategies for preventing and mitigating DNS infrastructure vulnerabilities

Through this comprehensive exploration, we aspire to contribute to the ongoing dialogue surrounding network security, raise awareness about DNS vulnerabilities, and equip cybersecurity professionals with practical knowledge for defending against these sophisticated attack methodologies.

II. Implementation

A. Setup

We set up *dnsmasq* with the following configuration, where we write a local DNS record stating that grouptwo.com is situated at 10.0.2.15

We do this by updating the *dnsmasq.conf* file.

```
interface=eth0
address=/grouptwo.com/10.0.2.15
```

We then update the *etter.dns* file as well. This allows the DNS Spoofing software we are using to log when our DNS server has successfully spoofed a machine.

```
grouptwo.com A 10.0.2.15
```

We enable IP forwarding on our VM with the following command. IP Forwarding allows our VM to inspect all packets that come in, tamper with the packets and then send them to their destination address.

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

We update our iptables [4] to route both TCP and UDP traffic directed towards port 53 to be redirected towards our Kali VM. Port 53 is the default used by Domain Name System for communication.

```
sudo iptables -t nat -A PREROUTING -p udp --
    dport 53 -j DNAT --to-destination
    10.0.2.15:53

sudo iptables -t nat -A PREROUTING -p tcp --
    dport 53 -j DNAT --to-destination
    10.0.2.15:53
```

The last command is used to rewrite the packet such that the source address seems to come from a legitimate DNS server rather than our malicious local DNS server.

```
sudo iptables -t nat -A POSTROUTING -j
    MASQUERADE
```

B. Execution

Startup *dnsmasq*, this starts up our small DNS forwarding service at port 53 of the Kali VM's IP.

```
systemctl start dnsmasq.service
```

Startup *ettercap*. Ettercap [5] is a utility that enables our VM to perform ARP [6] and DNS Spoofing [7]. This software scans the network for hosts, and starts monitoring the network for traffic.

```
ettercap -G
```

C. Packet Capture

Packets of the attack, both the DNS request and the response from our malicious Kali VM are captured using Wireshark.

```
97    39.532672   10.0.2.4    8.8.8.8 DNS 70
        Standard query 0x8dd1 A grouptwo.com
```

2) Response: Here's the packet capture summary of

```
99    39.533556   8.8.8.8 10.0.2.4  DNS   86
        Standard query response 0x8dd1 A
    grouptwo.com A 10.0.2.15
```

III. Mitigation Strategies

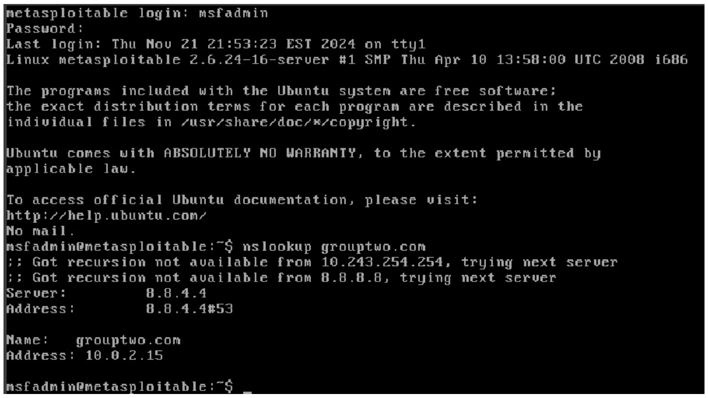
We have found four major strategies in helping to mitigate such attacks.

A. Static ARP Entries

We propose implementing static ARP entries, specifically for the router gateway, like so.

```
arp -s <gateway-ip> <gateway-mac-address>
```

Upon doing this, any DNS request that gets redirected will be marked as unexpected. The victim device no longer accepts forged ARP entries. DNS requests are sent only to the MAC address of the legitimate DNS server. Any mismatch between the attacker's spoofed IP and MAC causes DNS queries to fail, effectively blocking the attack. [8]



```
metasploitable login: msfadmin
Password:
Last login: Thu Nov 21 21:53:23 EST 2024 on tty1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ nslookup grouptwo.com
;; Got recursion not available from 10.243.254.254, trying next server
;; Got recursion not available from 8.8.8.8, trying next server
Server:         8.8.4.4
Address:        8.8.4.4#53

Name:   grouptwo.com
Address: 10.0.2.15

msfadmin@metasploitable:~$ _
```

Fig. 1. Output given by Metasploitable

This technique is effective because ARP spoofing operates at the local network level, and static ARP enforces strict control over IP-MAC mappings. [9]

B. Arp Detection Tooling

Tools such as *arpwatch* [10] can be used to monitor traffic and IP-MAC mappings, such tools can generate a log of any changes in IP-MAC addresses and send these logs via email to the administrator

C. Enforce Trusted DNS Servers

Using DHCP, we can ensure that all devices are configured to a trusted DNS server, such as 1.1.1.1, 8.8.8.8 or an organizational DNS server.

D. DNS Over HTTPS

Encrypts DNS queries by using HTTPS (Hyper Text Transfer Protocol Secure) [11] [3] to prevent attackers from intercepting or tampering with them, which can be done like so:

```
curl -H 'accept: application/dns-json' 'https
    ://1.1.1.1/dns-query?name=grouptwo.com&type
    =A'
```

Although this isn't perfect, as argued by [3]. It's a better solution than current methods of DNS Resolution.

IV. Conclusion

DNS Poisoning is a serious attack that can be performed on local networks which can lead to large amounts of personal data theft or compromise entire organizations. It is paramount to understand strategies for mitigation as well as understanding the attack vectors through which this takes place. Our work has shed light on the attack and the methodology for implementing it.

V. Future Work

This study focused on implementing and mitigating DNS spoofing within a controlled virtual network. While the experiment successfully demonstrated the attack and evaluated mitigation strategies, several avenues for future research and exploration remain:

1) Future work can expand the attack methodology to include advanced techniques, such as DNS cache poisoning [12] at a higher network level (e.g., ISP-level attacks) or combining DNS spoofing with other exploits like phishing or malware delivery to study their compounded effects. [13]
2) The experiment was conducted in a simple NAT-based virtual network. Future studies could examine DNS spoofing in more complex setups, involving more end devices, including corporate networks, cloud-based environments, or networks with active intrusion detection systems (IDS) and intrusion prevention systems (IPS).
3) Developing and testing automated systems that detect DNS spoofing attempts in real-time, using tools like machine learning models trained on data logs and labelled verified logs of attacks to identify anomalies in DNS traffic, could significantly enhance defensive capabilities.

VI. Acknowledgments

We would like to take this opportunity to thank our professor and mentor, Mr. Vishwas H.N. His guidance provided for the project allowed us to achieve and implement this attack. The insights that he provided through the lectures allowed us to have a solid foundation to explore and understand this avenue of cybersecurity and network vulnerabilities.

References

[1] Oracle, "VirtualBox - a general-purpose full virtualization software for x86_64 hardware."

[2] S. Ali, "DNS Using BIND and DHCP," in *Practical Linux Infrastructure*, S. Ali, Ed. Berkeley, CA: Apress, 2015, pp. 197–224.

[3] L. Csikor, H. Singh, M. S. Kang, and D. M. Divakaran, "Privacy of DNS-over-HTTPS: Requiem for a Dream?" in *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, Sep. 2021, pp. 252–271.

[4] G. N. Purdy, *Linux Iptables Pocket Reference: Firewalls, NAT & Accounting*. "O'Reilly Media, Inc.", Aug. 2004.

[5] A. Ornaghi and Marco Valleri, "Ettercap - a comprehensive suite for man in the middle attacks."

[6] K. M. Majidha Fathima and N. Santhiyakumari, "A Survey On Network Packet Inspection And ARP Poisoning Using Wireshark And Ettercap," in *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, Mar. 2021, pp. 1136–1141.

[7] P. Babu, L. Bhaskari, and CH.Satyanarayana, "A Comprehensive Analysis of Spoofing," *International Journal of Advanced Computer Sciences and Applications*, Jan. 2011.

[8] M. Data, "The Defense Against ARP Spoofing Attack Using Semi-Static ARP Cache Table," in *2018 International Conference on Sustainable Information Engineering and Technology (SIET)*, Nov. 2018, pp. 206–210.

[9] Z. Trabelsi and W. El-Hajj, "ARP spoofing: A comparative study for education purposes," in *2009 Information Security Curriculum Development Conference*, ser. InfoSecCD '09. New York, NY, USA: Association for Computing Machinery, Sep. 2009, pp. 60–66.

[10] Sudhakar and R. K. Aggarwal, "A survey on comparative analysis of tools for the detection of ARP poisoning," in *2017 2nd International Conference on Telecommunication and Networks (Tel-NET)*, Aug. 2017, pp. 1–6.

[11] A. A. Maksutov, I. A. Cherepanov, and M. S. Alekseev, "Detection and prevention of DNS spoofing attacks," in *2017 Siberian Symposium on Data Science and Engineering (SSDSE)*, Apr. 2017, pp. 84–87.

[12] J. Trostle, B. Van Besien, and A. Pujari, "Protecting against DNS cache poisoning attacks," in *2010 6th IEEE Workshop on Secure Network Protocols*, Oct. 2010, pp. 25–30.

[13] F. Alharbi, Y. Zhou, F. Qian, Z. Qian, and N. Abu-Ghazaleh, "DNS Poisoning of Operating System Caches: Attacks and Mitigations," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 4, pp. 2851–2863, Jul. 2022.