OS Fingerprinting Analysis

Nathan Hirata

Cal Poly Pomona

Dr. Tamer Omar, Ph.D.

May 5, 2023

Table of Contents

OS Fingerprinting 3
Introduction 3
Experimental Methodology 5
Results and Challenges 6
Conclusion 10
References 11

OS Fingerprinting

Introduction

One major subject of network packet analysis involves operating system fingerprinting. Operating system fingerprinting is a process of determining the operating system (OS) of a remote target computer system, without accessing the system directly. The process consists of collecting various pieces of traffic information about a system. The data collected is compared to a database containing operating system characteristics and determines the most accurate match. Several techniques can be performed to gather the necessary data.

The first strategy takes into consideration the operating system's response when the target's IDS (Intrusion Detection System) or IPS (Intrusion Protection System) is activated. Active or passive OS fingerprinting can be performed depending on the environment that is available to the attacker. Active OS Fingerprinting involves sending packets to the target system and analyzing the responses. The responses can contain information about the type and version of the operating system being used by the target system. This is an aggressive tactic because the attacker is actively sending packets to a network, which can trigger the target's network intrusion system. If the attacker cannot access the network directly, active OS fingerprinting could be the only way to gather relevant information. The other approach is known as passive OS fingerprinting, which involves analyzing the traffic between the target system and other systems on the network. This process is less detectable because the attacker does not send packets to the target's system, instead, it collects and analyzes any traffic passing by. Passive OS fingerprinting requires the attacker to be connected to the network, which is often not possible. In Figure 4 shown below, the OS fingerprinting map showcases the levels of hierarchy when conducting OS fingerprinting. The right-most category displays the various tools that can be used to perform OS fingerprinting. The Nmap networking tool will be used for the lab experiment in this research report.

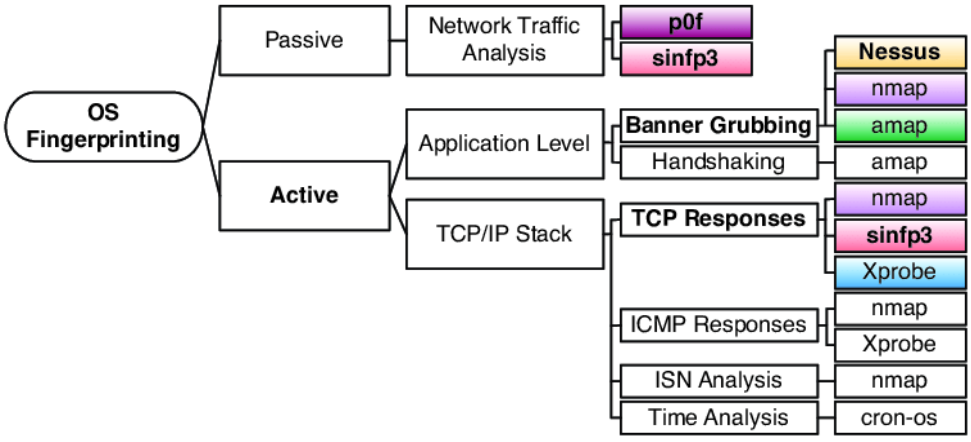

Figure 4: Taxonomy of OS fingerprinting

Data that can lead to the prediction of a remote operating system consists of Time-to-Live values, protocol flags, window and packet sizes, and timestamps. Time-to-Live (TTL) analysis collects the TTL value in the IP packets sent by the target system. The TTL value can provide clues about the type of operating system being used. Some well-known TTL values are 64, 128, and 255. Linux and macOS are associated with 64, Windows OS is associated with 128, and Cisco devices are associated with 255 [5]. One thing to note is that these TTL values are not definitive. They can be changed by network administrators or by the specific implementation of the operating system. Therefore, TTL values should not be used as the only identifying characteristic for an operating system.

Protocol analysis can also be used to identify an operating system. Protocol analysis involves studying behaviors in networking protocols such as TCP and ICMP. For example, the way a system responds to TCP SYN packets can provide information about the operating system's network stack. Another protocol that can help identify an OS is ICMP. It is possible to identify the underlying operating system of a device on the network by analyzing the Type and Code values of ICMP responses. This is because different operating systems may generate different ICMP response messages and use different Type and Code values. For example, the TTL value in an IP header of an ICMP Echo Reply (Type 0, Code 0) can be used to determine the operating system. Some operating systems have default TTL values that are different from others. This allows for the identification of the OS based on the TTL value in the ICMP response. When the data is collected by using any of the techniques mentioned, it is compared to a database of collected operating system characteristics. The accuracy of the OS fingerprinting process depends on the completeness and accuracy of the database used.

OS fingerprinting is helpful for network administrators because they can use the knowledge for network hardening, applying proper policies, and recognizing potential threats [1]. Knowing the operating system of a target system is also valuable information for an attacker, as it can help them launch specific malware. Older operating systems may have vulnerabilities that were never patched. This can be exploited if an attacker knows the specific version of the targeted OS. Websites such as CVE.org record all known vulnerabilities for every version of most operating systems [4]. It is also possible to use the knowledge of the operating system to craft social engineering attacks. For example, an attacker can send out emails to victims offering patches for known vulnerabilities in their operating system. Another threat involves an operating system's default configurations. For instance, default passwords or open ports can be used to gain unauthorized access to the system. Lastly, network enumeration can also be performed with the knowledge of the targeted operating system. This involves identifying other systems on the network that are running the same operating system. The attacker can identify additional targets or launch coordinated attacks. To sum up, knowing the operating system of a target system can be valuable information for an attacker. It can help them find potential vulnerabilities and launch

targeted attacks. It is important for organizations to be aware of these vulnerabilities. They can harden their systems or protect their data.

Experimental Methodology

The purpose of this experiment is to conduct OS fingerprinting against different operating systems using the Nmap tool. The end results will tell which operating systems are deployed with fingerprint security and what configurations can leave a system susceptible to OS fingerprinting. Virtual environments will be used to simulate different operating systems, including Ubuntu 20.04, Windows 10, macOS Ventura 13, Kali Linux 2023, Fedora 38, FreeBSD 13.2, and Oracle Solaris 11. The experiment will aim to determine the accuracy and effectiveness of OS fingerprinting using the Nmap tool.

Software and Hardware:

- Nmap networking tool
- VirtualBox VM software
- ISO images of Ubuntu 20.04, Windows 10, macOS Ventura 13, Kali Linux 2023, Fedora 38, FreeBSD 13.2, and Oracle Solaris 11
- Host computer with at least 8GB of RAM and a quad-core CPU

Experimental setup:

1. Install VirtualBox or other virtualization software on the host computer.
2. Download the ISO images of Ubuntu 20.04, Windows 10, macOS Ventura 13, Kali Linux 2023, Fedora 38, FreeBSD 13.2, and Oracle Solaris 11.
3. Create virtual machines for each operating system using VirtualBox. Allocate at least 2GB of RAM and 2 CPUs to each virtual machine.
4. Configure network settings for each virtual machine to use the Bridged Adapter in VirtualBox. This will allow the virtual machines to communicate with each other and the host computer. Additional configurations such as opening ports and firewall rules will be used to find further results
5. Install the Nmap tool on the host computer.

OS Fingerprinting:

1. Start a virtual machine and ensure they are connected to the same network.
2. Run the following command in the terminal of the testing computer to perform OS fingerprinting: $ sudo nmap -A -Pn <IP address>
   For <IP address>, use the IP address of the target machine. Get the IP address using ifconfig (Linux/Unix systems) or ipconfig (Windows systems). The -A command enables OS detection, version detection, script scanning, and traceroute [3]. The -Pn command disables host discovery and performs port scan only [3].
3. Record the results of the OS fingerprinting for each virtual machine.

Results and Challenges

After completing multiple fingerprinting tests, the Nmap tool was very successful in identifying most of the simulated operating systems. Table 2 shown below is split up into two column headers to better display results.

Operating System	OS fingerprint with base OS	Successful OS fingerprint after port configuration
Ubuntu 20.04	Failed to verify	Successful
Fedora 38	Failed to verify	Successful
Kali Linux 2023.1	Failed to verify	Successful
FreeBSD 13.2	Successful	Successful
Oracle Solaris 11	Successful	Successful
macOS Ventura 13	Failed to verify	Failed to verify
Windows 10	Failed to verify	Successful

Table 2: OS fingerprinting results

In the second column, Nmap's OS fingerprinting tool was performed on the base operating system freshly installed. FreeBSD and Solaris were able to be identified immediately due to the lack of firewall and port security. The rest of the tested operating systems gave incorrect or inaccurate results. The third column displays results after modifications were made to each of the operating systems. The only changes configured to each operating system were opening an SSH port and allowing port 22 communications to pass through the firewall. By opening this port, the Nmap tool was able to accurately identify all but macOS Ventura 13.

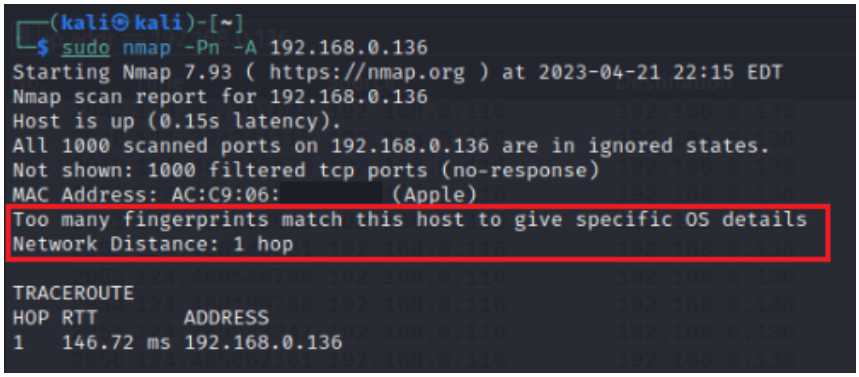

Figure 5: macOS Ventura 13 fingerprinting output

Figure 5 above displays the output after executing the Nmap OS fingerprinting tool. macOS Ventura 13 did not display sufficient results even after the firewall was disabled and an SSH port opened. There are a few reasons for this outcome. The first is the Nmap database. Nmap's database may not be updated or accurate when targeting macOS systems. Another reason could be the security of macOS Ventura 13. This was suspected after using Wireshark to analyze the packets sent and received when performing OS fingerprinting on this system. It was noticed that the Nmap tool was sending multiple TCP and ICMP packets to the device, but the macOS system was not responding back. Responses are crucial when trying to identify an operating system through packet analysis. It is possible that the macOS detected an aggressive scan and decided not to respond to a potential threat.

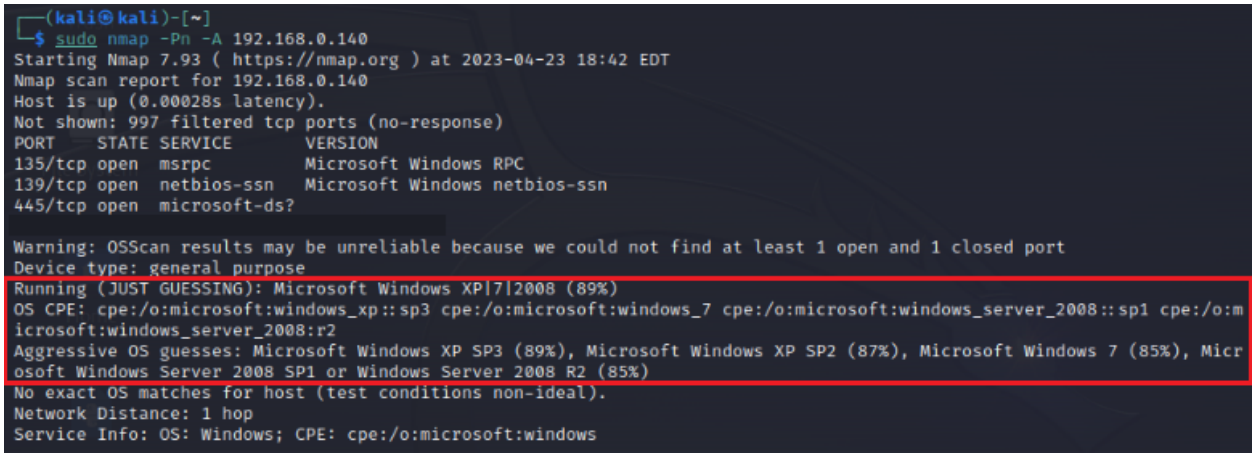

Figure 6: Windows 10 fingerprinting output with firewall enabled

In Figure 6, the output of the Nmap scan was inconclusive. It was able to identify that the system was Microsoft Windows, but not the correct version. The Microsoft firewall provides some protection against OS fingerprinting.

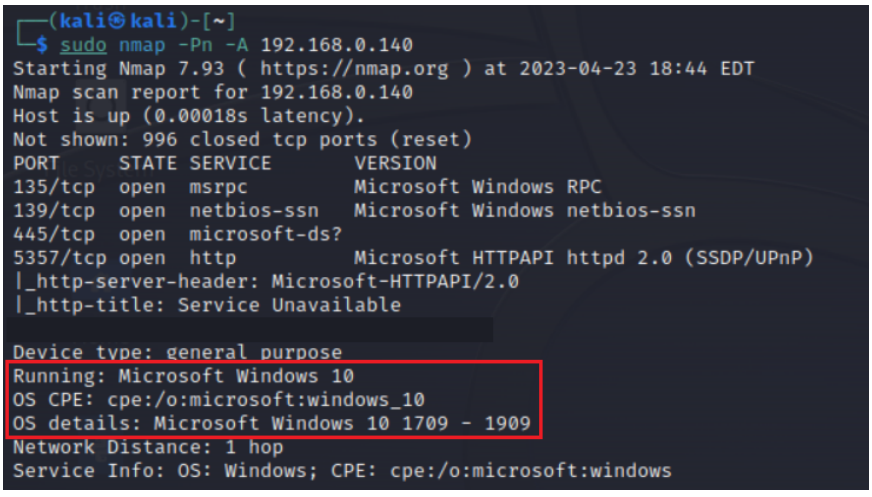

Figure 7: Windows 10 fingerprinting output with firewall disabled

After disabling the Windows Defender Firewall, the Nmap tool successfully identified the correct operating system. The output can be seen in Figure 7 showcasing the version number as well. For this specific test, an open SSH port was not configured. Instead, the firewall was disabled and the Nmap tool was able to capture a fingerprint through an open HTTP port.

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -Pn -A 192.168.0.144
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-23 17:44 EDT
Nmap scan report for 192.168.0.144
Host is up (0.00033s latency).
Not shown: 984 filtered tcp ports (no-response), 15 filtered tcp ports (host-prohibited)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 18a589be7c0cce6f                (RSA)
|   256 1b0834d2fdbe2183b               (ECDSA)
|_  256 2080b6279f69fdfb2               (ED25519)
MAC Address: 08:00:27:20:F0:BB (Oracle VirtualBox virtual NIC)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.33 ms 192.168.0.144
```

Figure 8: Ubuntu 20.04 with open SSH port configured

The output in Figure 8 displays an example of successfully fingerprinting the Linux-based operating system Ubuntu 20.04. It can be examined that the tool did not identify the exact version of Linux, but it outputted a precise range of possibilities. Linux version 5.4 was the correct version of the operating system. It is still accurate enough to gather important information about the system.

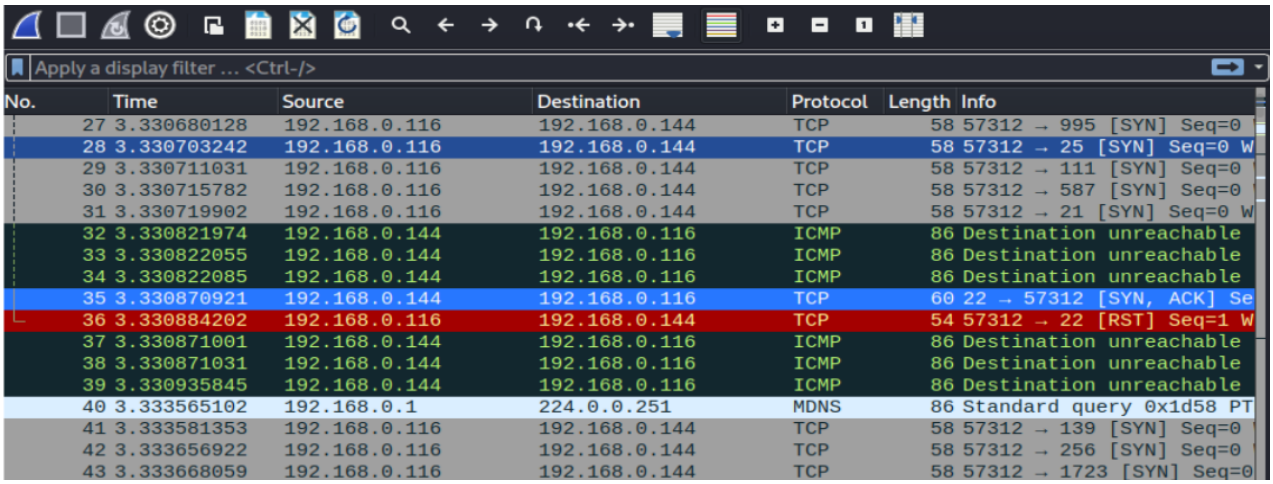

Figure 9: Nmap fingerprinting on Ubuntu via Wireshark traffic

Figure 9 illustrates the Nmap OS fingerprinting tool in action. Wireshark was used to capture communication while Nmap was scanning the Ubuntu OS. The target (Ubuntu OS) IP is identified as 192.168.0.144 and the attacker (Kali Linux) IP is identified as 192.168.0.116. It can

be examined through the grey-colored rows that the Nmap tool was sending groups of modified TCP packets in an attempt to retrieve unique responses. The green highlighted rows are the responses captured from the targeted machine.

One obstacle that was difficult to get through in this experiment was finding a way to make the secured operating systems vulnerable to Nmap's OS fingerprinting tool. This involved testing many of the different argument options with Nmap commands. Also, configuring the ports to be open and assuring that the specific port was allowed through the firewall while actively listening. This experiment shows the importance of firewalls and port security. Even if a firewall is active, an open port can be vulnerable to OS fingerprinting. One way to ensure a device is secure from OS fingerprinting is by using IPsec (Internet Protocol Security). IPsec is a protocol designed to provide secure communication over the internet. It operates at the network level and encrypts IP packets, which is the critical information needed to fingerprint an OS.

Conclusion

Fingerprinting an operating system is an analysis technique for identifying a remote computer's OS. The process consists of collecting unique traffic information such as Time To Live values, IP headers, TCP flags, ICMP codes, and window size. The data collected is then compared to a database of known OS characteristics to determine the most precise match. Understanding OS fingerprinting can help with using best practices for securing a system. It is also useful for network administrators as they can use the OS information for recognizing potential vulnerabilities, network hardening, and applying proper policies. In the experiment performed, OS fingerprinting was conducted against different operating systems using the Nmap tool. Virtual environments were used to simulate different operating systems. In each different operating system, the accuracy of Nmap's OS fingerprinting tool and database was analyzed. The results of this experiment helped in understanding how vulnerable each operating system is to OS fingerprinting and emphasized the importance of securing firewall ports and network data.

References

[1] S. Salah, M. Abu Alhawa and R. Zaghal. “Desktop and mobile operating system fingerprinting based on IPv6 protocol using machine learning algorithms,” *International Journal of Security and Networks*, 2022. https://www.inderscienceonline.com/doi/abs/10.1504/IJSN.2022.122543

[2] M. Lastovicka, M. Husák, Petr Velan, T. Jirsík, and P. Čeleda, “Passive Operating System Fingerprinting Revisited: Evaluation and Current Challenges,” *Social Science Research Network*, Jan. 2022, https://doi.org/10.2139/ssrn.4292623.

[3] “Nmap: the Network Mapper” Nmap.org, 2017. https://nmap.org/docs.html

[4] “cve-website,” Cve.org, 1999. https://www.cve.org/

[5] Y.-C. Chen, Y. Liao, M. Baldi, S.-J. Lee, and L. Qiu, “OS Fingerprinting and Tethering Detection in Mobile Networks,” *Proceedings of the 2014 Conference on Internet Measurement Conference*, Nov. 2014, https://doi.org/10.1145/2663716.2663745.