We pledge our honor that we have abided by the Stevens Honor System.

Catherine Javadian

Kaitlynn Prescott

Brianne Trollo

Operating System Fingerprinting

Operating systems (OS) has distinct characteristics that can identify it on a network. An OS's communication implementation can be as unique as a fingerprint. OS Fingerprinting is the process of learning what operating system is running on a particular device. This is achieved by analyzing certain protocol flags, options, and data in the packets that the device is sending onto the network.

To determine the operating system there are many different signatures that can be used. Four of the signatures that can be analyzed to determine the OS include the Time to Live (TTL), Window Size, the Don't Fragment bit (DF), and the Type of Service (TOS). The TTL is what the operating system sets the Time To Live on the outbound packet. The Window Size is what the operating system sets the TCP Window Size, which is the option to receive window size allowed in Transmission Control Protocol (Gibb). The analysis tests to determine if the remote host echoes back the IP DF bit in a response packet. An attacker then sends a UDP datagram with the DF bit set to a closed port on the remote host to observe whether the DF bit is set in the response packet. Some operating systems will echo the bit in the ICMP error message while others will zero out the bit in response packet ("IP (DF) 'Don't Fragment Bit' Echoing Probe."). The fingerprinting probe would also look to determine if the OS has set the TOS and if so, to what. By analyzing all these signatures of a packet, one is able to determine, with some degree of accuracy, the remote OS. There is no one signature that can reliably determine the remote

OS. It is only by looking at several signatures and piecing together the information gathered by them that the accuracy of the OS identification is increased (Spitzer).

OS Fingerprinting has two main uses, hacker attacks and network maintenance. Fingerprinting an OS of a target machine is a good reconnaissance technique for hackers. By pinpointing the exact OS of a host, a hacker can learn the vulnerabilities to exploit and launch a precise attack on a target machine (“What You Must Know About OS Fingerprinting.”). In addition, from the information fingerprinting can be used to make educated guesses concerning the importance of and the role the devices play on a network. OS Fingerprinting can also be used for maintaining a network. Regular scanning of a network can notify a user when a new device has been installed and help keep a network inventory clean and up to date.

OS fingerprinting techniques can be generalized into two categories: active and passive. Both rely on the principle that every operating system’s IP stack has its own idiosyncrasies, which enable an attacker to infer information. Active fingerprinting is more likely to return the information that will benefit an attacker. However, the main reason for that many attackers choose the passive approach is because it reduces the risk of being caught by an IDS, IPS, or a firewall. Active fingerprinting works by sending specially crafted packets to the target host and then analyzing the packets that are sent back. This enables attackers to obtain more accurate results in a shorter amount of time by using active rather than passive. In order to active fingerprint a target, it is almost always done using nmap, which is usually used by networking administrators. Administrators monitor the security of their networks and can make sure that all of the firewalls in their network are properly configured. They can also make sure that all of the TCP/IP stacks they maintain are functioning properly. By using nmap, an attacker can figure out the version of OpenSSH, the port the host is running on, and the server the host is running on. For example, using the command “nmap -A ip_adress_or_domain_naime_of_target”, can infer that target host is running a Debian-based Linux distro running Kubuntu 14.04. This sends a

number of TCP, UDP, and ICMP probes to the local machine and analyzes what it returns. Using this information, it is possible for an attacker to try to exploit vulnerabilities specific to the Linux kernel version in Kubuntu 14.04 or all current Debian-based OSes. Active fingerprinting will give the attacker a place to start. However, it is possible to sometimes get inaccurate results.

On the other hand, passive fingerprinting sniffs TCP/IP ports as opposed to generating network traffic by sending packets to them. As a result, it is a more effective way of avoiding detection or being stopped by a firewall. Based on the sniffer traces of these packets, an attacker can determine the operating system of the remote host. Passive fingerprinting uses a packet capture API and determines a target machine's OS by analyzing the initial Time to Live (TTL) in packet IP headers, and the TCP window size in the first packet of a TCP session. This is usually either a synchronize or a synchronize and acknowledge packet. Unlike active fingerprinting, there is no question to the legal nature of a passive scan because it is simply analyzing traffic that has already been sent. Passive fingerprinting can also be accomplished offline by examining packets that were previously captured. Since passive scanners are generally and inherently more inaccurate, they have less control over the data they analyze. However, it is almost completely undetectable. The most frequently used tools for passive fingerprinting are NetworkMiner and Satori.

P0f is a passive OS detection engine that reads packets from the network and analyzes them without generating any traffic of its own. It can operate in one of four different modes: SYN, SYN+ACK, RST, and stray ACK. The default is to listen for incoming connections and only fingerprints those clients that are making a connection to the host running p0f, making it the most comprehensive and accurate database of the four modes. SYN+ACK mode performs outgoing connection fingerprinting that can be used when an attacker wants to know what kind of server the user is connecting to. RST mode can be used to figure out when it is not possible

to establish a full connection to the target device. Through this case, a SYN packet is sent to a closed port and the resulting RST packet is analyzed. For stray ACK, it can be used to fingerprint an existing connection. A p0f fingerprinting file contains a very thorough explanation of all TCP traits that are examined: overall packet size, selective ACT (SACK) support, NOP option, EOL option, the sequence of TCP options, etc. P0f is able to determine network distance, link type, the probable presence of a NAT device between the remote and local hosts, and sometimes an uptime estimate.

Nmap and Xprobe2 are active scanners. Nmap is the most popular network scanner in use today. When at least one port is open and at least one port is closed, nmap will rapidly scan large segments of a network, looking for any active devices. With nmap, OS detection can only be performed after a port scan has been completed. A number of flags help determine how nmap is set up and run. The -sV flag determines the service running on a port. It is useful to note that simply using non standard ports is not a dependable form of security. The -T flag determines how fast nmap scans a host, 1 being the slowest and 5 being the fastest. A slower scan speed can reduce the possibility of being detected, and minimize potential flooding of the target. The -P0 flag tells nmap not to send an ICMP Echo before the port scan. Finally, the -O flag signifies that nmap will perform a remote OS detection.

Xprobe2 is an ICMP scanner, and is one of the quietest active scanners today. The Xprobe2 scanner is a matrix based OS fingerprint detector, which will match the fingerprint to the OS based on probability and statistical calculations. It yields extremely quick results with little packet traffic; it does not need to send or receive as many packets as nmap. One feature Xprobe2 has is that you can customize the modules to best fit your needs. Each module that is run is completely independent of the others, so the results of one module will not affect the rest of the modules. Once the scan is complete, the scores are compared, and statistical analysis will determine which OS is the most probable.

With all of these methods of OS fingerprinting, is there a way to prevent a successful fingerprint? The short answer is no, there is currently no method that completely protects a network from attack. While these concerns are only necessary when malicious reconnaissance is a concern, there are ways of detecting whether or not someone is already inside your network and trying to blend in. The first thing you can do is ensure that no external host can scan an internal host. Another method is changing the TCP/IP settings. This will change how the network traffic of packets appears, but can have an impact on the device's network performance. The last method is simply to perform scans against your own network, and document your findings. This will allow you to see if there is any suspicious activities on your network. While this does not protect your network from an outside party accessing your network, it can alert you to what is occurring on your network, and help you determine the best course of action for remedying it.

Works Cited

Allen, Jon Mark. “OS and Application Fingerprinting Techniques.” SANS Institute InfoSec Reading Room, 22 Sept. 2007.

Gibb, Taylor. “OS Fingerprinting With TTL and TCP Window Sizes.” *How-To Geek*, How-To Geek LLC, 1 Feb. 2012, www.howtogeek.com/104337/hacker-geek-os-fingerprinting-with-ttl-and-tcp-window-sizes/.

“IP (DF) 'Don't Fragment Bit' Echoing Probe.” *Common Attack Pattern Enumeration and Classification*, MITRE, 4 Aug. 2017, capec.mitre.org/data/definitions/319.html.

Spitzner, Lance. “Passive Fingerprinting.” *Symantec Connect*, Symantec Corporation, 2 May 2000, www.symantec.com/connect/articles/passive-fingerprinting.

“What You Must Know About OS Fingerprinting.” *InfoSec Resources*, InfoSec Institute, 11 Mar. 2015, resources.infosecinstitute.com/must-know-os-fingerprinting/.