REMOTE OPERATING SYSTEMS

Ethan Johnsrud
Operating Systems 4222
University of Minnesota Duluth

Detecting Remote Operating Systems

I. Reasons for Detection

The ability to detect a operating system remotely includes many implications. Knowing an operating system provides a wealth of knowledge of capabilities to vulnerabilities. Hackers are probably the most well known for attempts to detect Operating Systems. Operating system detection can reduce false positives, and narrow probing efforts. If a hacker can detect the client's operating system, whey can direct their attack to specific vulnerabilities, drastically increasing their chances of acquiring valuable information and control of their client's machine.

On the flip side, network administrators may have more positive intentions for acquiring a remote operating system. Informational Technology Department may use operating system scanning for inventory purposes. This may be to ensure all companies machines have been upgraded and identify ones that need attention. Another example is before they renew that IRIX support contract for another year. The network can be scanned to see if anyone still uses such machines; and use the money in the budget for better purposes. An inventory can also be useful for IT budgeting and ensuring that all company equipment is accounted [1].

II. Fingerprinting

Operating Systems are detected by acquiring fingerprints or footprints that lead back to a specific operating system and version. Fingerprinting uses information to correlate data sets in order to identify with high probability, everything from network services, operating system number and version, software applications, databases, configurations and more. A variety of information is collected and compiled to make these distinctions. Fingerprint techniques often analyze different types of packets and information such as TCP Window size, TCP Options in TCP SYN and SYN+ACK packets, ICMP requests, HTTP packets, DHCP requests, IP TTL values as well as IP ID values. Additionally, domain names, DNS services, as well as IP addresses and SSL certificates can often leave unseen trails and clues along their paths. Once enough information has been gathered, this fingerprinting data can be used as part of an exploit strategy against the target [4].

III. Active Fingerprinting

Active fingerprinting is the most popular type of fingerprinting, due to its speed and immediate results. It consists of sending packets to a victim and waiting for the victim's reply to analyze the results. This is often the easiest way to detect remote OS, network and services. It's also the riskiest as it can be easily detected by intrusion detection systems (IDS) and packet filtering firewalls [4]. Active operating system fingerprinting requires the use of a set of specialized probes that are sent to the system in question. System responses from this active probing give insight into what type of operating system might be installed [3].

Port scanning is one of the most traditional forms of fingerprinting. This can be accomplished by sending TCP or ICMP packets. The TCP fingerprinting process involves setting flags in the header that different Operating Systems and versions respond to differently. Usually several different TCP packets are sent, and the responses are compared to known fingerprints to determine the remote operating system. Typically, ICMP-based methods use fewer packets than TCP-based methods, so in an environment where you need to be stealthier and can afford a less specific fingerprint, ICMP may be the best direction [3].

IV. NMAP

One of the most popular applications used to launch active searches is Nmap. This handy tool can help detect specific Operating Systems and network service applications when you launch TCP, UDP or ICMP packets against any given target. By using internal scripting rules, Nmap analyzes the results from the victim replies, then displays the results and historically very accurate [4].

The active process that Nmap applies in order to conduct its fingerprinting scan involves a set of fifteen probes. These powerful probes are designed to utilize TCP, UDP, and ICMP protocols. Then, NMAP receives several Acknowledgment TCP packets and checks the headers for a timestamp option. Many Operating Systems use a simple counter for this which starts at zero at boot time then increments at a constant rate such as twice per second. By looking at several responses, Nmap can determine the current values and rate of increase. Simple linear

REMOTE OPERATING SYSTEMS

Ethan Johnsrud
Operating Systems 4222
University of Minnesota Duluth

extrapolation determines boot time. The timestamp algorithm is used for OS detection too; since the increment rate on different systems varies from 2 Hz to 1,000 Hz [3].

In addition, NMAP uses an uptime guess. Some Operating Systems do not start the timestamp counter at zero, but initialize it with a random value, making extrapolation to zero meaningless. Even on systems using a simple counter starting at zero, the counter eventually overflows and wraps around. With a 1,000 Hz counter increment rate, the counter resets to zero roughly every 50 days [1].

IV. NMAP Testing Trials

I executed serval NMAP trials to identify Operating Systems on my home network of a variety of devices. The results are found in the table below. Interestingly, NMAP only correctly identified Google Home as a Linux Operating System. The most common operating system in our home are Windows 10 PC's. However, NMAP identified both the Oth desktop and Microsoft Surface Laptops as Windows XP, which is a little concerning. When scanning phones, I attempted OnePlus 6t and Motorola Moto; and retuning results reported all ports were closed. In addition to detecting Operating Systems, NMAP was able to determine the manufacture on every device from the MAC address.

V. Passive Fingerprinting

Passive fingerprinting is an alternative approach to avoid detection while performing your reconnaissance activities. It uses similar techniques to the active fingerprinting performed by Nmap. The difference is that a passive system simply sniffs the network, opportunistically classifying hosts as it observes their traffic. A major downside is it depends whatever communication happens rather than designing your own custom probes. It is a valuable technique that's key features analyze network other traffic, allowing the process to go unnoticed to the client. But takes time and doesn't deliver instant results as active scanning and NMAP provide [1].

Device	NMAP Result
nmap -O -v 192.168.1.7 Windows 10 Desktop	AVtech embedded (87%), Microsoft Windows XP (87%), FreeBSD 6.X|10.X (86%)
nmap -O 192.168.1.22 Windows 10 Surface Laptop	AVtech embedded (87%), Microsoft Windows XP (87%), FreeBSD 6.X|10.X (86%)
nmap -O -v 192.168.1.30 OnePlus Phone	All Ports are Closed. Too many fingerprints match.
nmap -O 192.168.1.10 Motorolla Phone	All Ports are Closed. Too many fingerprints match.
nmap -v -O 192.168.1.14 Epson Printer	NO OS Detected: 993/1000 Ports Closed. PORT STATE SERVICE 80/tcp open http 139/tcp open netbios-ssn 443/tcp open https 445/tcp open microsoft-ds 515/tcp open printer 631/tcp open ipp 9100/tcp open jetdirect
nmap -v -O 192.168.1.6 Google Home	linux_kernel:2.6 994/1000 Ports Closed PORT STATE SERVICE 7778/tcp open interwise 8008/tcp open http 8009/tcp open ajp13 8443/tcp open https-alt 9000/tcp open cslistener 10001/tcp open scp-config

The most popular passive scanner for detecting Operating Systems remotely is p0f. Created by Michal Zalewski , he devised a clever technique to provide similar results to that of NMAP [1]. P0f is a great alternative to Nmap, it analyzes network traffic and identify patterns behind TCP/IP based communications that are often blocked for Nmap active fingerprinting techniques. It includes powerful network-level fingerprinting features, as well as one that analyzes application-level payloads such as HTTP. It's also useful for detecting NAT, proxy and load balancing setups [4].

The main difference between active and passive fingerprinting is that passive fingerprinting does not actively

REMOTE OPERATING SYSTEMS

Ethan Johnsrud
Operating Systems 4222
University of Minnesota Duluth

send packets to the target system. Instead, it acts as a network scanner in the form of a sniffer, merely watching the traffic data on a network without performing network alteration [4]. Passive operating system fingerprinting monitors network traffic at any given collection point and matching known patterns that pass to a table of pre-established operating system fingerprints. While this type of technique may bypass common network intrusion detection techniques, it's not guaranteed to hide your network presence while sniffing traffic. Also, does not always provide enough information to create a unique enough fingerprint to detect the operating system targeted [3].

Masking Operating System

I. Reasons for Masking

The most obvious is if someone can identify your operating system, makes things easier to find and successfully run an exploit against any of your devices. Additionally, an attacker is able to make data inferences from guessing which applications are you running in that operating system. An example of this is a computer running Microsoft Windows and running a database, it's highly likely that they are running Microsoft SQL. Also, is convenient for other software companies, to market a new operating system environment. And finally, privacy; nobody needs to know the Operating Systems you're using [1].

II. Preventing Operating System Detection

The best tool for detecting active port scanning is by using an Intrusion Detection System on a network. The common methods are host detection and network intrusion detection tools. Host intrusion detection tools monitor the activities of a specific host. And network intrusion detection tools monitor network traffic to recognize noteworthy or alarming network activity. The most popular application that provides these services is Snort [3].

As this paper has clearly laid out, applications such as NMAP use a variety of methods to sniff out an operating system. So, the only true way to totally mask an operating system is to fail all possible tests attacked. While this is for the most part impossible, providing enough false answers specifically in key areas should be enough to fool attackers. For instance, using IP Personality, changing every packet's window size, and faking responses is enough to fool a passive search from p0f. Another example is in a Linux environment, just changing the TCP/IP stack behavior will fool an NMAP attack. Other things to change to ward off operating system detection is TCP Initial Sequence Number (ISN), TCP initial window size, TCP options including their types, values and order in the packet structure [1].

Conclusion

Operating System detection is very valuable to attackers, network administrators, and something to guarded by possible subjects. Detecting an operating system remotely is not a straightforward task, and neither is fighting against the probing. There are a great number of variations in Operating Systems and versions, that identifying a unique enough fingerprint is a tricky challenge. Tools such as NMAP enable active tracking that sends traffic directly to a target in hopes of acquiring key information. This is risky behavior, as it easily detected by the target. Alternatively, passive tracking quietly monitors other traffic to develop a unique fingerprint, this takes time and is not always efficient to provide reliable results.

References

[1] " Nmap Network Scanning," Nmap Network Scanning. [Online]. Available: https://nmap.org/. [Accessed: 31-Jan-2020].

[2] B. Berr 31322 silver badges99 bronze badges, "Masking Operating System," Information Security Stack Exchange, 13-Nov-2017. [Online]. Available: https://security.stackexchange.com/. [Accessed: 31-Jan-2020].

[3] J. Faircloth, "Passive Fingerprinting," Enumeration and Scanning with Netcat and Nmap. [Online]. Available: https://www.sciencedirect.com/topics/computer-science/passive-fingerprinting. [Accessed: 31-Jan-2020].

[4] S. T. Team, "SecurityTrails: Cybersecurity Fingerprinting Techniques and OS-Network Fingerprint Tools," Cybersecurity Fingerprinting, 21-May-2019. [Online]. Available: https://securitytrails.com/cybersecurity-fingerprinting. [Accessed: 31-Jan-2020].