



# Global Information Assurance Certification Paper

Copyright SANS Institute  
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

## Interested in learning more?

Check out the list of upcoming events offering  
"Network Monitoring and Threat Detection In-Depth (Security 503)"  
at <http://www.giac.org/registration/gcia>

## **Fingerprinting Windows 10 Technical Preview**

*GIAC (GCIA) Gold Certification*

Author: Jake Haaksma, jlhaaksma@live.com

Advisor: Manuel Santander

Accepted: August X<sup>th</sup> 2015

### **Abstract**

Understanding the intricacies of a network is powerful information for security professionals and malicious attackers alike. Operating system (OS) fingerprinting is the process of determining the OS of a remote computer. This can be primarily accomplished by passively sniffing network packets between hosts or actively sending crafted packets to the ports of a target host in order to analyze its response. This paper attempts to fingerprint Windows 10 Technical Preview for the purpose of OS identification and to improve Nmap's OS detection database.

## 1. Introduction

There are five phases in the ethical hacking methodology, and they begin with reconnaissance, scanning, gaining access, maintaining access, and covering tracks (EC-Council, 2015). Reconnaissance is the preliminary phase where a malicious attacker attempts to gather as much information as possible about the target. The use of this information will assist in determining if any vulnerability exists on any of the live hosts. The next phase will attempt to exploit the vulnerable host in order to gain access. Once a malicious attacker gains access to the target host then she/he will attempt to install a backdoor or rootkit to maintain access for future entry. Finally, the malicious attacker will begin covering their tracks by destroying evidence of their presence.

In relation to this paper, the tools and methods used will primarily fall under the reconnaissance and scanning phases. “For attackers, network scanning is an initial reconnaissance to locate hosts running vulnerable network services” (Alsaleh, Oorschot, 2012). The malicious attacker must know the target host’s OS in order to have a high success rate of gaining access. As such, a malicious attacker will be meticulous when performing OS fingerprinting because this information will greatly affect each phase of their operation. “For example, the dbot worm invokes GetUserName() to check if the user is in its blacklist of user names. Windows operation system provides several API functions for gaining system-level information, which can be misused for identifying the running environment” (Xie, Lu, Wang, Su, 2013). OS fingerprinting isn’t limited to only conducting scans against a remote host. A malicious attacker might attempt multiple avenues to acquire this type of information, like reviewing publicly available information, social engineering, dumpster diving, or viruses.

Yet there also multiple reasons to perform OS fingerprinting that would be for the benefit of an IT security professional. OS fingerprinting can assist in determining if a remote host is susceptible to a particular vulnerability. Through OS detection scanning, an IT security professional can focus their efforts on patching a particular set of hosts that would be vulnerable to an OS-specific exploit. Yet, OS fingerprinting can also assist in tailoring OS-specific exploits. “Buffer overflows, format-string exploits, and many other vulnerabilities often require custom-tailored shellcode with offsets and assembly payloads generated to match the target OS and hardware architecture” (Lyon, 1997). There are also administrative reasons where OS fingerprinting scans can provide additional information such as network

inventory. These scans will be able to locate what is running on a particular network to keep an inventory of a company's assets. In addition, these scans will be able to assist in detecting unauthorized and potential dangerous devices for investigation and containment. "With the ubiquity of mobile devices and cheap commodity networking equipment, companies are increasingly finding that employees are extending their networks in undesirable ways" (Lyon, 1997). A company's corporate network may be opened up to potential attackers who may be in a close proximity if an employee installs an unauthorized wireless access point.

## 2. Fingerprinting Windows 10 Technical Preview

For the purpose of this exercise, all scanning activity was conducted in an isolated virtual environment. A virtual machine (VM) running OS Kali Linux version 3.18.0 with Nmap version 6.47 was utilized to probe another virtual machine running Windows 10 Technical Preview version 10.0.10041. "All the scanning tools fall into one of two major categories: Active or Passive" (Budhrani, Sridaran, 2015). Nmap is an active scanning tool as it will send out packets to a target host rather than passively sniff traffic to be analyzed. The configuration for each virtual machine's network adapter was set to host only and with no port forwarding in order to create an isolated, clean environment between source and destinations addresses. The IP address of the Kali Linux virtual machine is 192.168.56.102 and the IP address for the Windows 10 Technical Preview virtual machine is 192.168.56.101. These versions and system information can be seen in Figure 1-6.

Each type of scan was performed twice. The first set of scans probed the default configuration of Windows 10 Technical Preview. While the second set of scans probed Windows 10 Technical Preview with its host-based firewall turned off for both public and private networks. The host-based firewall disrupted all of the first set of scans. This rendered the outcome to be inadequate for analysis. As such, all forthcoming analysis will be derived from the second set of scans.

Figure 1 – Network Diagram

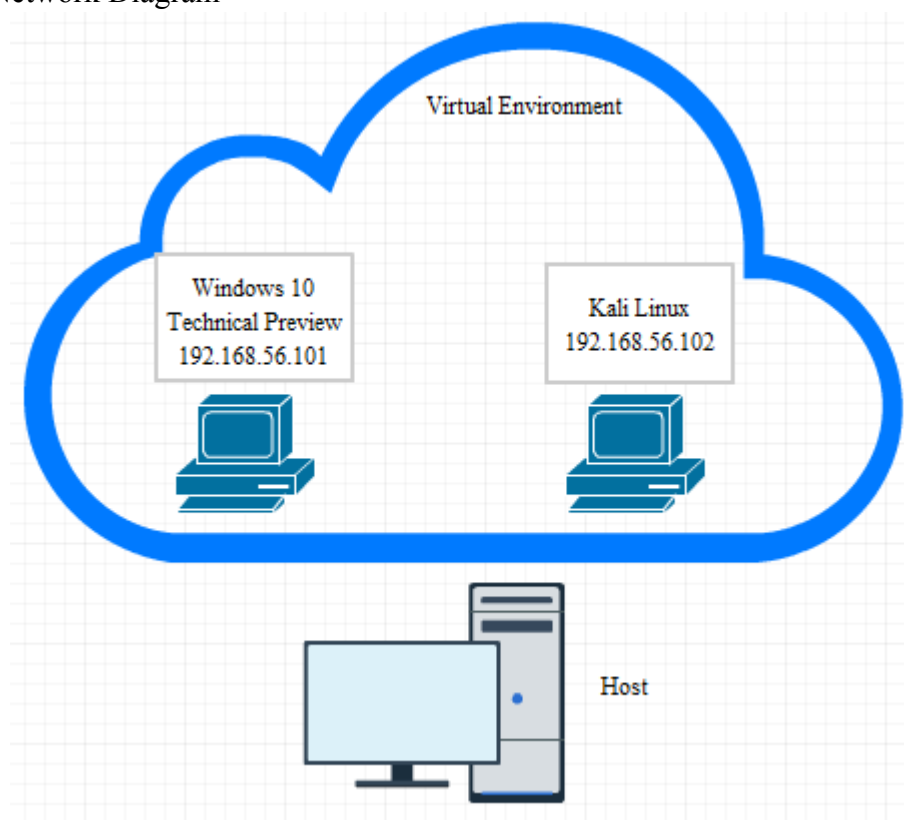


Figure 2 – Kali Linux Virtual Machine Overview

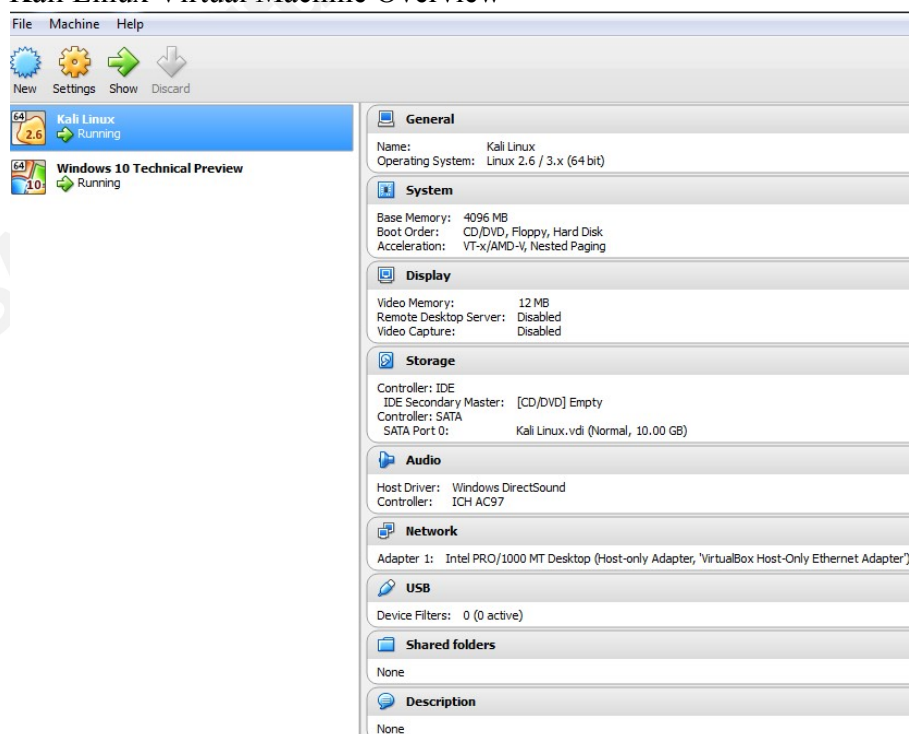


Figure 3 – Windows 10 Technical Preview Virtual Machine Overview

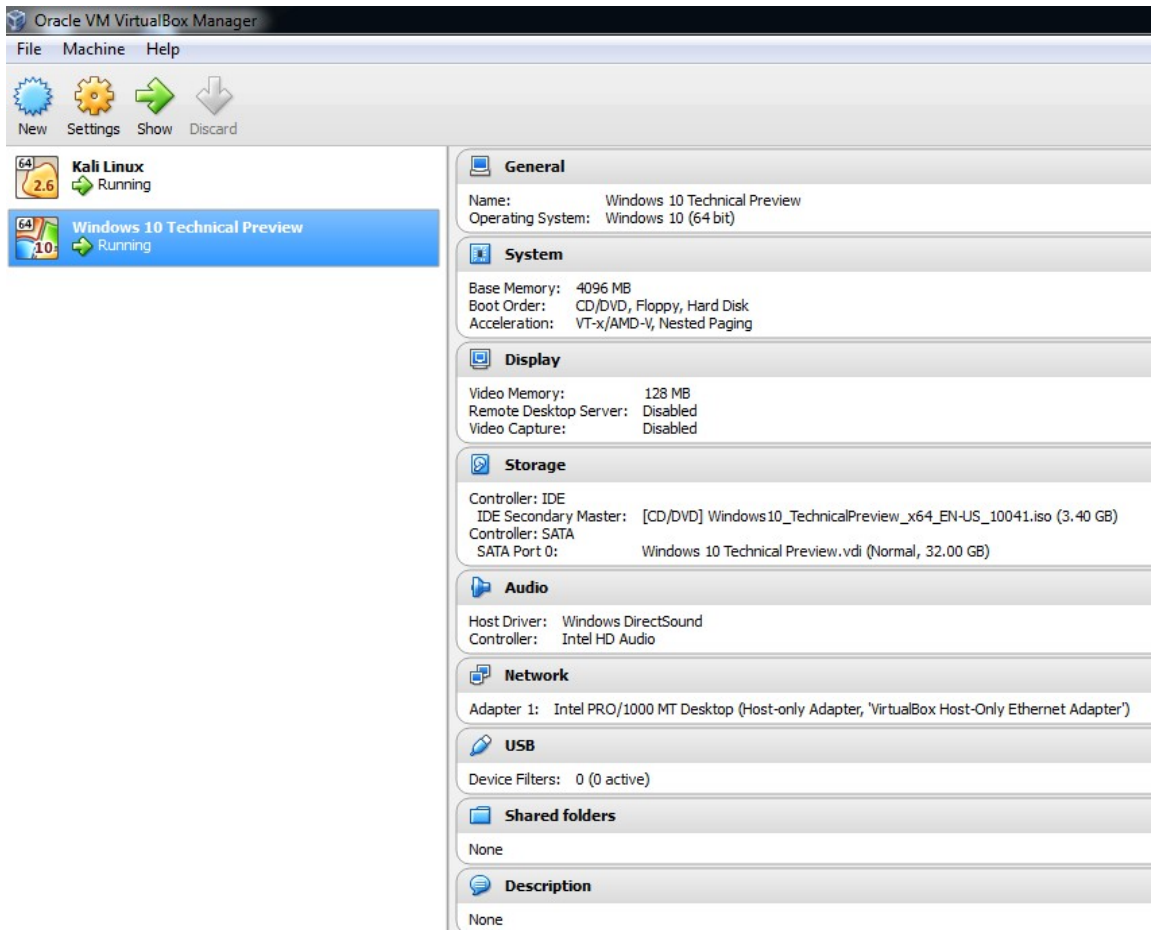


Figure 4 – Kali Linux version and IP address

```

root@kali:~# cat /proc/version
Linux version 3.18.0-kali3-amd64 (debian-kernel@lists.debian.org) (gcc version 4
.7.2 (Debian 4.7.2-5) ) #1 SMP Debian 3.18.6-1-kali2 (2015-03-02)
root@kali:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:38:45:95
          inet addr:192.168.56.102  Bcast:192.168.56.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe38:4595/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:207 errors:0 dropped:0 overruns:0 frame:0
          TX packets:16287 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:24381 (23.8 KiB)  TX bytes:1001570 (978.0 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128  Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:44 errors:0 dropped:0 overruns:0 frame:0
          TX packets:44 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:2640 (2.5 KiB)  TX bytes:2640 (2.5 KiB)

```

Figure 5 – Nmap version

```

root@Kali:~# nmap -V
Nmap version 6.47 ( http://nmap.org )
Platform: x86_64-unknown-linux-gnu
Compiled with: nmap-liblua-5.2.3 openssl-1.0.1e libpcap-1.3.0 nmap-
libdnet-1.12 ipv6
Compiled without:
Available nsock engines: epoll poll select

```

Figure 6 – Windows 10 Technical Preview system information

```

C:\Windows\system32>systeminfo

Host Name:                               WIN-9RS99NUCGHI
OS Name:                                 Microsoft Windows 10 Pro Technical Preview
OS Version:                             10.0.10041 N/A Build 10041
OS Manufacturer:                       Microsoft Corporation
OS Configuration:                      Standalone Workstation
OS Build Type:                           Multiprocessor Free
Registered Owner:                       Jake Haaksma
Registered Organization:
Product ID:                              00137-10010-52743-AA923
Original Install Date:                   6/25/2015, 5:14:03 PM
System Boot Time:                        6/25/2015, 5:07:01 PM
System Manufacturer:                    innotek GmbH
System Model:                            VirtualBox
System Type:                             x64-based PC
Processor(s):                            1 Processor(s) Installed.
                                           [01]: Intel64 Family 6 Model 60 Stepping 3 GenuineIntel ~3498 Mhz
BIOS Version:                            innotek GmbH VirtualBox, 12/1/2006
Windows Directory:                      C:\Windows
System Directory:                       C:\Windows\system32
Boot Device:                             \Device\HarddiskVolume1
System Locale:                           en-us;English (United States)
Input Locale:                            en-us;English (United States)
Time Zone:                               (UTC-08:00) Pacific Time (US & Canada)
Total Physical Memory:                   4,096 MB
Available Physical Memory:                3,165 MB
Virtual Memory: Max Size:                5,504 MB
Virtual Memory: Available:               4,541 MB
Virtual Memory: In Use:                  963 MB
Page File Location(s):                  C:\pagefile.sys
Domain:                                 WORKGROUP
Logon Server:                            \\WIN-N6DB63303HG
Hotfix(s):                               N/A
Network Card(s):                        1 NIC(s) Installed.
                                           [01]: Intel(R) PRO/1000 MT Desktop Adapter
                                           Connection Name: Ethernet
                                           DHCP Enabled:   Yes
                                           DHCP Server:    192.168.56.100
                                           IP address(es)
                                           [01]: 192.168.56.101
                                           [02]: fe80::542f:323b:afc1:8e5b
Hyper-V Requirements:                    UM Monitor Mode Extensions: No
                                           Virtualization Enabled In Firmware: No
                                           Second Level Address Translation: No
                                           Data Execution Prevention Available: Yes

```

## 2.1 Scanning for Ports/Services/Protocols

The reconnaissance phase begins with target definition. This is where an attacker specifies a target IP address, hostname, or a range of IP addresses of interest. The attacker's objective is to reduce a range of IP addresses into a list of active hosts, which is known as host discovery. Nmap contains a variety of options to perform this step such as TCP Connect option (-sT). This option will send a TCP packet with the SYN flag set. If the destination



host's port is open, a SYN/ACK packet is sent back as the second step of a TCP three-way handshake. However, if the destination host's port is closed, a RST packet is sent back. If a SYN/ACK packet is received, Nmap will complete the TCP three-way handshake with a final ACK packet. In Figure 7 and 8, tcpdump has captured both a successful and unsuccessful TCP connection. However, in this situation, the only target of interest is 192.168.56.101.

Figure 7 - SYN and RST/ACK

```

19:35:00.759682 IP 192.168.56.102.35207 > 192.168.56.101.32775: Flags [S], seq 1360142389, win 1024, options [mss 1460], length 0
19:35:00.759704 IP 192.168.56.101.3546 > 192.168.56.102.35207: Flags [R.], seq 0, ack 1360142390, win 0, length 0
19:35:00.759721 IP 192.168.56.102.35207 > 192.168.56.101.1217: Flags [S], seq 1360142389, win 1024, options [mss 1460], length 0
19:35:00.759740 IP 192.168.56.101.32775 > 192.168.56.102.35207: Flags [R.], seq 0, ack 1360142390, win 0, length 0

```

Figure 8 - SYN, SYN/ACK, and ACK

```

19:43:55.111162 IP 192.168.56.102.52124 > 192.168.56.101.submission: Flags [S], seq 1109792729, win 29200, options [mss 1460,sackOK,TS val 101767 ecr 0,nop,wscale 7], length 0
19:43:55.111250 IP 192.168.56.102.41303 > 192.168.56.101.netbios-ssn: Flags [S], seq 98288432, win 29200, options [mss 1460,sackOK,TS val 101767 ecr 0,nop,wscale 7], length 0
19:43:55.111375 IP 192.168.56.101.submission > 192.168.56.102.52124: Flags [R.], seq 0, ack 1109792730, win 0, length 0
19:43:55.111458 IP 192.168.56.101.netbios-ssn > 192.168.56.102.41303: Flags [S.], seq 1889179381, ack 98288433, win 8192, options [mss 1460,nop,wscale 8,sackOK,TS val 707746 ecr 101767], length 0
19:43:55.111466 IP 192.168.56.102.41303 > 192.168.56.101.netbios-ssn: Flags [.], ack 1, win 229, options [nop,nop,TS val 101767 ecr 707746], length 0

```

After the attacker has compiled a list of active hosts, then the next course of action is to determine what ports, protocols, services, and the version of those services are open on each host. This information can potentially become the avenue for the attacker's exploit if a vulnerability is present. "Various ports and IP addresses may lead to the same type of attack" (Sultana, Charles, Govardhan, 2013). Nmap allows a user to be quite flexible because of the variety of scans to choose from. Each Nmap's scan option will probe 1,000 ports of the destination host. The response will be recognized into six port states, which are open, closed, filtered, unfiltered, open|filtered, and closed|filtered.

1. An open port denotes that an application is actively accepting TCP connections and UDP datagrams (Lyon, 1997).
2. A closed port is accessible but no application is listening on it (Lyon, 1997).



3. A filtered port state indicates that Nmap cannot determine whether the port is open because packet filtering prevents the probe from reaching the port (Lyon, 1997).
4. An unfiltered port state represents the port is accessible, but Nmap is unable to determine whether it is open or closed. “Only the ACK scan, which is used to map firewall rulesets, classifies ports into this state. Scanning unfiltered ports with other scan types such as Window scan, SYN scan, or FIN scan, may help resolve whether the port is open” (Lyon, 1997).
5. The open|filtered port state signifies that the port could be open or filtered because no response from the port was received. “The lack of response could also mean that a packet filter dropped the probe or any response it elicited. So Nmap does not know for sure whether the port is open or being filtered. The UDP, IP protocol, FIN, NULL, and Xmas scans classify ports this way” (Lyon, 1997).
6. Lastly, the closed|filtered port state implies that the port could be closed or filtered. “It is only used for the IP ID idle scan” (Lyon, 1997).

An attacker can continue to study a particular host by querying for an accurate service version number. This piece of information can determine which exploits a host might be vulnerable to. “Nmap’s service and version detection scans determine the service protocol (e.g. FTP, SSH, Telnet, HTTP), the application name (e.g. ISC BIND, Apache httpd, Solaris telnetd), the version number, hostname, device type (e.g. printer, router), the OS family (e.g. Windows, Linux)” (Lyon, 1997). When a malicious attacker knows their target host’s open and closed ports, supported protocols, running services and the version of those services, then she or he can begin to anticipate a high chance of compromising the host.

As seen in Figure 9, the initial scan against 192.168.56.101 was to analyze the state of each port. Nmap has revealed that 192.168.56.101 has 10 open TCP ports but was only able to specify 3 services: Microsoft Remote Procedure Call (MSRPC), Netbios-ssn, and Microsoft- directory services (ds). This information could indicate that the target resembles a host running some type of Windows OS. MSRPC is an interprocess communication mechanism that enables data exchange and functionality residing in a different process (What is RPC?, 2003). Netbios provides services to allow applications on separate computers to communicate over a LAN. These services include a name service, datagram distribution service, and a session service. Netbios-ssn is a session service for connection-oriented

communication (NetBIOS Over TCP/IP, 2005). Microsoft-ds allow Server Message Block (SMB) to run over TCP for session-oriented communication (NetBIOS Over TCP/IP, 2005). “Port scanning can be conducted using special TCP scanning tools, such as NMAP, where the attacker usually attempts to connect to the target computer via various TCP ports” (Anbar, Manasrah, Manickam, 2012). If the host’s firewall were up, then the results would have been quite limited. All possible ports were probed with a TCP SYN packet, but the host’s firewall prevented the packet from reaching the port. Therefore, the host could not respond.

Figure 9 – Nmap Port Scan against firewall turned off

```
root@kali:~# nmap -p 0-65535 192.168.56.101

Starting Nmap 6.47 ( http://nmap.org ) at 2015-07-01 13:26 CDT
Nmap scan report for 192.168.56.101
Host is up (0.00029s latency).
Not shown: 65526 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
49408/tcp  open  unknown
49409/tcp  open  unknown
49410/tcp  open  unknown
49411/tcp  open  unknown
49412/tcp  open  unknown
49413/tcp  open  unknown
49414/tcp  open  unknown
MAC Address: 08:00:27:07:07:00 (Cadmus Computer Systems)

Nmap done: 1 IP address (1 host up) scanned in 16.74 seconds
```

The next type of scan is Nmap’s SYN scan, which can be executed by the `-sS` switch in one’s Nmap command. This popular scan can be performed quickly and is relatively unobtrusive and stealthy. The SYN scan will start the three-way handshake of a TCP connection to only see if the target host will reply with a SYN/ACK or a RST/ACK. Nmap will record the response and will not attempt to complete the three-way handshake, which can be seen in Figure 10. Figure 11 showcases the results of the scan. The second set of scans concluded that 3 registered TCP ports were open along with the same listening services from the previous scan.

Figure 10 - SYN and RST/ACK

```

19:35:00.759682 IP 192.168.56.102.35207 > 192.168.56.101.32775: Flags [S], seq 1360142389, win 1024, options [mss 1460], length 0
19:35:00.759704 IP 192.168.56.101.3546 > 192.168.56.102.35207: Flags [R.], seq 0, ack 1360142390, win 0, length 0
19:35:00.759721 IP 192.168.56.102.35207 > 192.168.56.101.1217: Flags [S], seq 1360142389, win 1024, options [mss 1460], length 0
19:35:00.759740 IP 192.168.56.101.32775 > 192.168.56.102.35207: Flags [R.], seq 0, ack 1360142390, win 0, length 0

```

Figure 11 – Nmap SYN Scan against firewall turned off

```

root@kali:~# nmap -sS 192.168.56.101

Starting Nmap 6.47 ( http://nmap.org ) at 2015-06-29 13:29 CDT
Nmap scan report for 192.168.56.101
Host is up (0.00032s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
MAC Address: 08:00:27:07:07:00 (Cadmus Computer Systems)

Nmap done: 1 IP address (1 host up) scanned in 14.22 seconds

```

A generally slower, more difficult scan such as the UDP scan tend to be overlooked, but there are exploitable UDP services. “UDP scanning is not commonly used compared to TCP scanning, but it is still used and gives accurate results for scanners” (Elejla, Jantan, Ahmed, 2014). A UDP scan (-sU) can be quite customized due to the number of options available. The conclusions shown in Figure 12 reveal new information about

192.168.56.101. Nmap is able to determine that UDP port 137 is open, but the other ports could be open or filtered.

1. UDP port 123 could be running the Network Time Protocol (NTP), which is for clock synchronization between computer systems.
2. UDP port 137 is running Netbios-ns. This service is another one of the Netbios distinct functions specifically for name registration and resolution.
3. UDP port 138 could be running Netbios-dgm, which is the final service of Netbios used for datagram distribution service for connectionless communication.

4. UDP port 500 could be running the Internet Security Association and Key Management Protocol (ISAKMP). “Port 500 is used by the Internet key exchange (IKE) that occurs during the establishment of secure VPN tunnels. Users of VPN servers and clients may encounter this port” (Gibson, 2008).
5. UDP Port 1900 is Simple Service Discovery Protocol. “This UDP port is opened and used by Universal Plug N' Play (UPnP) devices to receive broadcasted messages from other UPnP devices. UPnP devices broadcast subnet-wide messages to simultaneously reach all other UPnP devices” (Gibson, 2008).
6. UDP port 4500 could be running Network Address Translation – Traversal – IKE (NAT-T-IKE).
7. The zeroconf service on UDP port 5353 is associated with Mac OS X Bonjour/Zeroconf, which on Windows, allows for communication between Windows and Mac OS.
8. Finally, UDP port 5355 could be running Link-Local Multicast Name Resolution (LLMNR). This service provides name resolution services for hosts on an ad-hoc network (Posey, 2006).
9. UDP port 137, 5355, and 138 are another set of services typically seen on Windows machines. It appears that this host uses the Internet Key Exchange (IKE) protocol in the IPsec protocol suite when establishing a VPN tunnel or performing NAT traversal” (Gibson, 2008).

Figure 12 – Nmap UDP Scan against firewall turned off

```
root@kali:~# nmap -sU 192.168.56.101
Starting Nmap 6.47 ( http://nmap.org ) at 2015-06-29 13:31 CDT
Nmap scan report for 192.168.56.101
Host is up (0.00067s latency).
Not shown: 992 closed ports
PORT      STATE      SERVICE
123/udp    open|filtered ntp
137/udp    open       netbios-ns
138/udp    open|filtered netbios-dgm
500/udp    open|filtered isakmp
1900/udp   open|filtered upnp
4500/udp   open|filtered nat-t-ike
5353/udp   open|filtered zeroconf
5355/udp   open|filtered llmnr
MAC Address: 08:00:27:07:07:00 (Cadmus Computer Systems)
Nmap done: 1 IP address (1 host up) scanned in 1085.67 seconds
```

This next Nmap scan is not technically a port scan. An IP protocol scan (-sO) allows a user to determine which IP protocols are supported by the target host. “Rather than cycling through the target’s ports it tries all 256 possible IP protocol numbers in the IP header” (Lyon, 1997). Figure 13 showcases that 192.168.56.101 supports fairly common IP protocols and even two members of the IPsec protocol suite, Encapsulating Security Payload (ESP) and Authentication Header (AH).

Figure 13 – Nmap IP Protocol Scan against firewall turned off

```
root@kali:~# nmap -sO 192.168.56.101

Starting Nmap 6.47 ( http://nmap.org ) at 2015-06-29 17:55 CDT
Nmap scan report for 192.168.56.101
Host is up (0.00027s latency).
Not shown: 248 closed protocols

```

PROTOCOL	STATE	SERVICE
1	open	icmp
2	open filtered	igmp
4	open filtered	ip
6	open	tcp
17	open	udp
41	open filtered	ipv6
50	open filtered	esp
51	open filtered	ah

```
MAC Address: 08:00:27:07:07:00 (Cadmus Computer Systems)

Nmap done: 1 IP address (1 host up) scanned in 315.07 seconds
```

Nmap’s Service/Version Detection Scan (-sV) can be enhanced through the following options: --allports and --version-all. The option --allports informs Nmap to not exclude a ports from this scan and the option --version-all increases the intensity of the scan to the maximum level of 9 (Lyon, 1997). The intensity level changes the number of probes Nmap will send to accurately identify running services. Although the results in Figure 14 are similar to several of the previous port and SYN scans, it differs in that Netbios-ssn is listening on both TCP port 139 and 445. The port/SYN/TCP-connect scans concluded that Microsoft-ds is listening on TCP port 445 rather than Netbios-ssn. These results provide indication that the target host’s OS is some version of Windows and not



UNIX because of the MSRPC and Netbios services as these services were developed by Microsoft.

Figure 14 – Nmap Service Version Detection Scan against firewall turned off

```
root@kali:~# nmap -sV --allports --version-all 192.168.56.101
Starting Nmap 6.47 ( http://nmap.org ) at 2015-06-29 18:01 CDT
Nmap scan report for 192.168.56.101
Host is up (0.00027s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE      VERSION
135/tcp    open  msrpc        Microsoft Windows RPC
139/tcp    open  netbios-ssn  Microsoft Windows RPC
445/tcp    open  netbios-ssn  Microsoft Windows RPC
MAC Address: 08:00:27:07:07:00 (Cadmus Computer Systems)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 20.25 seconds
```

The default IP version for Nmap's scan is version 4; however, Nmap can conduct an IPv6 scan (-6) against a target host. The IPv6 scan is identical to the SYN scan where instead IPv6 SYN packets are sent to the target host. Nmap will look for a SYN/ACK or RST/ACK to judge whether the port is open or not. In Figure 15, Nmap was able to detect through the IPv6 scan that TCP port 135 and 445 are open. Again, the same services can be seen before in the previous scans and reinforce the suspicion that the target host's OS is some version of Windows.

Figure 15 – Nmap IPv6 Scan against firewall turned off

```
root@kali:~# nmap -6 fe80::542f:323b:afc1:8e5b
Starting Nmap 6.47 ( http://nmap.org ) at 2015-06-30 06:43 CDT
Nmap scan report for fe80::542f:323b:afc1:8e5b
Host is up (0.000081s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
445/tcp    open  microsoft-ds
MAC Address: 08:00:27:07:07:00 (Cadmus Computer Systems)

Nmap done: 1 IP address (1 host up) scanned in 0.09 seconds
```

## 2.2 Windows 10 Technical Preview Ports/Protocols/Services

The results, particularly from the second set of Nmap scans, provided a great deal of information regarding the host's port and service activity. The next series of

screenshots (Figure 16-19) were taken from the Windows 10 Technical Preview VM to display the actual open ports, protocols, processes, and running services. This information will also act as the benchmark to verify how accurate the scan results were.

Figure 16 shows that a default installation of Windows 10 Technical Preview has 10 open TCP ports and 12 open UDP ports. Nmap scans were able to correctly detect all 10 TCP ports, however was only able to identify 8 UDP ports. The enumerate UDP ports of 57455, 57456, 57457, 57458, and 546 were not revealed through the Nmap scans conducted in this experiment. Nmap may not be at fault for not determining the other 4 UDP ports because of the nature of the protocol. UDP provides no guarantee of delivery or anticipate a response from the given port.

Figure 17 correlates each port's PID to a more detailed view of the process and Figure 18 lists all of the started services for Windows 10 Technical Preview. There are some tactics available to the IT security professional to thwart such reconnaissance attempts such as port hopping. "In this way, port hopping disrupts the static associations between port numbers and services, which can be used to mitigate reconnaissance attacks because the attacker often identifies services of a given system by network probes, and then breaks in the target system by exploiting a known vulnerability of the service identified" (Luo, Wang, Cai, 2015). This concept of port hopping is a proactive cyber defense technology to mislead a potential attacker.

Figure 16 – Windows 10 Technical Preview open ports

```
C:\Windows\system32>netstat -aon | more
Active Connections

```

Proto	Local Address	Foreign Address	State	PID
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING	652
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING	4
TCP	0.0.0.0:49408	0.0.0.0:0	LISTENING	436
TCP	0.0.0.0:49409	0.0.0.0:0	LISTENING	912
TCP	0.0.0.0:49410	0.0.0.0:0	LISTENING	1112
TCP	0.0.0.0:49411	0.0.0.0:0	LISTENING	816
TCP	0.0.0.0:49412	0.0.0.0:0	LISTENING	516
TCP	0.0.0.0:49413	0.0.0.0:0	LISTENING	524
TCP	0.0.0.0:49414	0.0.0.0:0	LISTENING	1268
TCP	192.168.56.101:139	0.0.0.0:0	LISTENING	4
TCP	[::]:135	:::0	LISTENING	652
TCP	[::]:445	:::0	LISTENING	4
TCP	[::]:49408	:::0	LISTENING	436
TCP	[::]:49409	:::0	LISTENING	912
TCP	[::]:49410	:::0	LISTENING	1112
TCP	[::]:49411	:::0	LISTENING	816
TCP	[::]:49412	:::0	LISTENING	516
TCP	[::]:49413	:::0	LISTENING	524
TCP	[::]:49414	:::0	LISTENING	1268
UDP	0.0.0.0:5000	::*		816
UDP	0.0.0.0:4500	::*		816
UDP	0.0.0.0:5353	::*		636
UDP	0.0.0.0:5355	::*		636
UDP	127.0.0.1:1900	::*		920
UDP	127.0.0.1:57458	::*		920
UDP	192.168.56.101:137	::*		4
UDP	192.168.56.101:138	::*		4
UDP	192.168.56.101:1900	::*		920
UDP	192.168.56.101:57457	::*		920
UDP	[::]:5000	::*		816
UDP	[::]:4500	::*		816
UDP	[::]:5353	::*		636
UDP	[::]:5355	::*		636
UDP	[::]:1900	::*		920
UDP	[::]:57456	::*		920
UDP	[fe80::542f:323b:afc1:8e5b::3]:546	::*		912
UDP	[fe80::542f:323b:afc1:8e5b::3]:1900	::*		920
UDP	[fe80::542f:323b:afc1:8e5b::3]:57455	::*		920



Figure 17 Windows 10 Technical Preview Task Manager

Task Manager						
File Options View						
Processes Performance App history Startup Users Details Services						
Name	PID	Status	User name	CPU	Memory (private worki...	Description
System interrupts	-	Running	SYSTEM	02	0 K	Deferred procedure calls and interrupt service routines
System Idle Process	0	Running	SYSTEM	94	4 K	Percentage of time the processor is idle
System	4	Running	SYSTEM	00	28 K	NT Kernel & System
smss.exe	252	Running	SYSTEM	00	196 K	Windows Session Manager
csrss.exe	328	Running	SYSTEM	00	492 K	Client Server Runtime Process
csrss.exe	404	Running	SYSTEM	00	548 K	Client Server Runtime Process
wininit.exe	436	Running	SYSTEM	00	452 K	Windows Start-Up Application
winlogon.exe	444	Running	SYSTEM	00	668 K	Windows Logon Application
services.exe	516	Running	SYSTEM	00	1,740 K	Services and Controller app
lsass.exe	524	Running	SYSTEM	00	2,928 K	Local Security Authority Process
svchost.exe	604	Running	SYSTEM	00	2,672 K	Host Process for Windows Services
svchost.exe	636	Running	NETWORK SERVICE	00	3,428 K	Host Process for Windows Services
svchost.exe	652	Running	NETWORK SERVICE	00	2,060 K	Host Process for Windows Services
dwm.exe	744	Running	DWM-1	00	17,424 K	Desktop Window Manager
svchost.exe	816	Running	SYSTEM	00	13,092 K	Host Process for Windows Services
svchost.exe	868	Running	SYSTEM	03	21,320 K	Host Process for Windows Services
svchost.exe	912	Running	LOCAL SERVICE	00	9,660 K	Host Process for Windows Services
svchost.exe	920	Running	LOCAL SERVICE	00	836 K	Host Process for Windows Services
svchost.exe	992	Running	LOCAL SERVICE	00	3,772 K	Host Process for Windows Services
spoolsv.exe	1112	Running	SYSTEM	00	2,452 K	Spooler SubSystem App
svchost.exe	1136	Running	LOCAL SERVICE	00	5,616 K	Host Process for Windows Services
svchost.exe	1268	Running	NETWORK SERVICE	00	576 K	Host Process for Windows Services
svchost.exe	1376	Running	SYSTEM	00	5,668 K	Host Process for Windows Services
MsMpEng.exe	1472	Running	SYSTEM	00	29,604 K	Antimalware Service Executable
RuntimeBroker.exe	1840	Running	Jake Haaksma	00	4,588 K	Runtime Broker
searchui.exe	1856	Running	Jake Haaksma	00	21,608 K	searchui
svchost.exe	1968	Running	SYSTEM	00	2,240 K	Host Process for Windows Services
sihost.exe	2076	Running	Jake Haaksma	00	2,388 K	Shell Infrastructure Host
taskhost.exe	2092	Running	Jake Haaksma	00	1,536 K	Host Process for Windows Tasks
ApplicationFrameHo...	2192	Running	Jake Haaksma	00	1,704 K	Application Frame Host
explorer.exe	2280	Running	Jake Haaksma	00	9,128 K	Windows Explorer
svchost.exe	2356	Running	SYSTEM	00	2,280 K	Host Process for Windows Services
WmiPrvSE.exe	2420	Running	SYSTEM	00	920 K	WMI Provider Host
Taskmgr.exe	2492	Running	Jake Haaksma	02	6,992 K	Task Manager
ShellExperienceHost...	2872	Running	Jake Haaksma	00	11,536 K	Windows Shell Experience Host
WShost.exe	2960	Running	Jake Haaksma	00	1,912 K	Store Broker
SearchIndexer.exe	3068	Running	SYSTEM	00	5,060 K	Microsoft Windows Search Indexer
cmd.exe	3276	Running	Jake Haaksma	00	328 K	Windows Command Processor
conhost.exe	3284	Running	Jake Haaksma	00	1,264 K	Console Window Host
fontdrvhost.exe	3344	Running	Jake Haaksma	00	400 K	Usermode Font Driver Host
MSASCui.exe	3580	Running	Jake Haaksma	00	2,580 K	Windows Defender User Interface
OneDrive.exe	3692	Running	Jake Haaksma	00	3,144 K	Microsoft OneDrive

Figure 18 – Windows 10 Technical Preview default started services

```

C:\Windows\system32>net start
These Windows services are started:

Application Information
Background Intelligent Transfer Service
Background Tasks Infrastructure Service
Base Filtering Engine
Client License Service (ClipSUC)
COM+ Event System
CoreMessaging
Cryptographic Services
DCOM Server Process Launcher
Device Setup Manager
DHCP Client
Diagnostic Policy Service
Diagnostic Service Host
Diagnostics Tracking Service
Distributed Link Tracking Client
Distributed Transaction Coordinator
DNS Client
Geolocation Service
Group Policy Client
IP Helper
Local Session Manager
Microsoft Account Sign-in Assistant
Network Connection Broker
Network List Service
Network Location Awareness
Network Store Interface Service
Plug and Play
Power
Print Spooler
Program Compatibility Assistant Service
Remote Procedure Call (RPC)
RPC Endpoint Mapper
Security Accounts Manager
Security Center
Sensor Monitoring Service
Sensor Service
Server
Shell Hardware Detection
SSDP Discovery
Superfetch
System Event Notification Service
System Events Broker
Task Scheduler
TCP/IP NetBIOS Helper
Themes
tiledatamodelsvc
Time Broker
TP AutoConnect Service
User Manager
User Profile Service
VMware Tools
Windows Audio
Windows Audio Endpoint Builder
Windows Connection Manager
Windows Defender Service
Windows Driver Foundation - User-mode Driver Framework
Windows Event Log
Windows Firewall
Windows Font Cache Service
Windows Management Instrumentation
Windows Search
Windows Time
WinHTTP Web Proxy Auto-Discovery Service
Workstation

```

## 2.2 OS Detection Scanning

When Nmap begins to perform OS fingerprinting, it does this through a series of probes that are sent to the target. “These TCP, UDP, and ICMP probes are crafted to exploit various ambiguities for Nmap to analyze each response or lack thereof to generate a fingerprint for a particular OS. The supported methods for TCP/IP fingerprint by Nmap are split into two categories, probes sent and response tests” (Lyon, 1997). Below are the probes that are sent:

1. Sequence generation (SEQ, OPS, WIN, and T1). This test involves sending a series of 6 crafted TCP SYN packets to analyze the target’s sequence algorithms, which include initial sequence numbers (ISN), IP IDs, and TCP timestamps. The results of each response are contained into 4 category lines. The SEQ line will showcase the sequence analysis of the packets. The OPS line contains the TCP options received for each of the packets. The WIN line covers the window sizes for the packet responses. The T1 line will contain various test values based on the response of the first TCP SYN packet sent (Lyon, 1997).
2. ICMP echo (IE). This test will send 2 crafted ICMP echo request packets to the target host. The IE line will contain the results of each ICMP echo reply (Lyon, 1997).
3. TCP explicit congestion notification (ECN). This tests the target’s TCP/IP stack for ECN support. If a response is received, then the ECN line will record its response (Lyon, 1997).
4. TCP (T2-T7). This test will send 6 crafted TCP packets with various characteristics then each response is recorded in the T2-T7 lines.
5. UDP (U1). This final probe test will send a UDP packet to a closed port to verify if the port is truly closed and there is no packet-filtering device in place. This result will be displayed in the U1 line (Lyon, 1997).

The response tests are along with the Windows 10 Technical Preview responses:

1. TCP ISN greatest common divisor (GCD). This response test is to determine the smallest number by which the target host increments their ISN,  $GCD=1$  (Lyon, 1997).

Jake Haaksma, jlhaaksma@live.com

2. TCP ISN counter rate (ISR). This response test reports the average rate of increase for the target host's ISN,  $ISR=1$  (Lyon, 1997).
3. TCP ISN sequence predictability index (SP). This response test, combined with GCD and ISR, attempts to estimate how difficult it would be to predict the target host's next ISN,  $SP=100$  (Lyon, 1997).
4. IP ID sequence generation algorithm (TI, CI, II). These tests will examine the responses in the IP header ID field. These tests come from the SEQ, T5-T7, and IE probes,  $TI=I$ ,  $CI=I$ ,  $II=I$  (Lyon, 1997).
5. Shared IP ID sequence Boolean (SS). This response test will record whether the target host shares its IP ID sequence between the TCP and ICMP protocols,  $SS=S$  (Lyon, 1997).
6. TCP timestamp option algorithm (TS). This response test will attempt to determine OS characteristics by the TCP timestamp options in the SEQ responses,  $TS=A$  (Lyon, 1997).
7. TCP options (O, O1-O6). This response test will record what TCP options are set in the response packets,  $O1=M5B4NW8ST11$ ,  $O2=M5B4NW8ST11$ ,  $O3=M5B4NW8ST11$ ,  $O4=M5B4NW8ST11$ ,  $O5=M5B4NW8ST11$ ,  $O6=M5B4ST1$  (Lyon, 1997).
8. TCP initial window size (W, W1-W6). This response test will record the 16-bit TCP window size of the received packets,  $W1=2000$ ,  $W2=2000$ ,  $W3=2000$ ,  $W4=2000$ ,  $W5=2000$ ,  $W6=2000$  (Lyon, 1997).
9. Responsiveness (R). This response test will simply record whether or not the target host has responded to a given packet,  $R=Y$  (Lyon, 1997).
10. IP don't fragment bit (DF). This response test will record if the don't fragment bit is set or not,  $DF=Y$  (Lyon, 1997).
11. Don't fragment ICMP (DFI). Similar to the DF test, it will look for the don't fragment bit from the IE packets,  $DFI=N$  (Lyon, 1997).
12. IP initial time-to-live (T). This response test will record how many hops away it is from the target host,  $T=1$  (Lyon, 1997).
13. IP initial time-to-live guess (TG). This response test will attempt to estimate the target host's initial TTL value, nothing set (Lyon, 1997).

Jake Haaksma, jlhaaksma@live.com

14. Explicit congestion notification (CC). This response test will record if ECN is supported or not,  $CC=N$  (Lyon, 1997).
15. TCP miscellaneous quirks (Q). This response test will examine and record oddities in the target host's TCP stack such as if the reserved field in the TCP header is nonzero,  $Q=(\text{nothing set})$  (Lyon, 1997).
16. TCP sequence number (S). This response test will examine the 32-bit sequence number field in the TCP header and compare it to the 32-bit acknowledge number field, for each TCP probe sent  $S=0|Z|Z|A|Z|A|Z$  (Lyon, 1997).
17. TCP acknowledge number (A). This particular test is just the opposite of the TCP sequence number response test, for each TCP probe sent  $A=S|S|O|O|S|S|S$  (Lyon, 1997).
18. TCP flags (F). This response test will records what TCP flags were set in the response packets from the target host, for each TCP probe sent  $F=A|AR|AR|R|AR|R|AR$  (Lyon, 1997).
19. TSP RST data checksum (RD). If the target host returns ASCII data in RST packets then that data is recorded, for each TCP probe sent  $RD=0|0|0|0|0|0|0$  (Lyon, 1997).
20. IP total length (IPL). This response test will record in the total length of an IP packet,  $IPL=164$  (Lyon, 1997).
21. Unused port unreachable field nonzero (UN). This response test will view the last 4 bytes of an ICMP port unreachable message header and record its value,  $UN=0$  (Lyon, 1997).
22. Returned probe IP total length value (RIPL). This response test will examine if the IP header is returned as it was received in an ICMP port unreachable message,  $RIPL=G$  (Lyon, 1997).
23. Returned probe IP ID value (RID). This response test will examine if the IP ID value is returned as it was received in an ICMP port unreachable message,  $RID=G$  (Lyon, 1997).
24. Integrity of returned probe IP checksum value (RIPCK). This response test will verify if the IP packet's checksum is valid when it is received,  $RIPCK=G$  (Lyon, 1997).

Jake Haaksma, jlhaaksma@live.com

25. Integrity of returned probe UDP checksum (RUCK). This response test will verify if the UDP packet's checksum is valid when it is received, RUCK=G (Lyon, 1997).
26. Integrity of returned UDP data (RUD). This response test will check the integrity of the UDP's payload, RUD=G (Lyon, 1997).
27. ICMP response code (CD). This final response test will examine the ICMP response code from the target host, CD=Z (Lyon, 1997).

This combination of probing and testing the target's responses are the hidden workings for Nmap OS detection (Lyon, 1997). There are two types of OS fingerprints that Nmap creates. "The fingerprints of known operating systems that Nmap reads in are called *reference fingerprints*, while the fingerprint Nmap displays after scanning a system is a *subject fingerprint*" (Gibson, 2008). The following OS scans have produce subject fingerprints. The TCP/IP fingerprint is prefixed with 'OS:' and for a subject reference is comprised of two sections, the scan line and then the values from the probes sent/response test. The scan line is a series of conditions to describe the environment for the Nmap scans. These conditions are:

1. the Nmap version number (V),
2. Date of scan (D),
3. Open and closed TCP ports (OT and CT),
4. Closed UDP port (CU),
5. Private IP space (PV),
6. Network distance (DS),
7. Distance calculation method (DC),
8. Good results (G),
9. Target MAC prefix (M),
10. OS scan time (TM),
11. Platform Nmap was compiled in (P).

Nmap's OS detection scans (-O) can be augmented by 2 options which are -  
 osscan-limit and --osscan-guess. The --osscan-guess option will offer up near matches as possibilities and provide a percentage to display its assurance for that guess. However the

Jake Haaksma, jlhaaksma@live.com

second set of scans came back with several guesses as seen in Figure 19. It first makes an initial guess of 97% that the OS is either Microsoft Windows

7|2008|Vista|Longhorn|2012|8.1. None of these initial guesses are correct but all of the guesses were in the Microsoft Windows family. Secondly, Nmap begins to aggressively print its assurance level on each OS guess. Thirdly, Nmap states that no exact OS match can be derived from this particular OS detection scan. Finally, Nmap prints its own TCP/IP fingerprint for the OS.

Figure 19 – Nmap OS Detection Guess Scan against firewall turned off

```
root@kali:~# nmap -O --osscan-guess 192.168.56.101

Starting Nmap 6.47 ( http://nmap.org ) at 2015-06-29 18:05 CDT
Nmap scan report for 192.168.56.101
Host is up (0.00031s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
MAC Address: 08:00:27:07:07:00 (Cadmus Computer Systems)
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows 7|2008|Vista|Longhorn|2012|8.1 (97%)
OS CPE: cpe:/o:microsoft:windows_7 cpe:/o:microsoft:windows_server_2008::sp1 cpe:/o:microsoft:windows_8 cpe:/o:microsoft:windows_vista::sp1 cpe:/o:microsoft:windows_vista::sp2 cpe:/o:microsoft:windows cpe:/o:microsoft:windows_2012 cpe:/o:microsoft:windows_8.1
Aggressive OS guesses: Microsoft Windows 7 Professional (97%), Microsoft Windows 7 SP0 - SP1, Windows Server 2008 SP1, or Windows 8 (97%), Microsoft Windows 7 Ultimate (97%), Microsoft Windows 7 or Windows Server 2008 (97%), Microsoft Windows Vista SP1 - SP2, Windows Server 2008 SP2, or Windows 7 (96%), Microsoft Windows 7 or Windows Server 2008 R2 (96%), Microsoft Windows Longhorn (96%), Microsoft Windows Server 2008 SP2 (95%), Version 6.1 (Build 7601; Service Pack 1) (94%), Microsoft Windows 7, Windows Server 2012, or Windows 8.1 Update 1 (94%)
No exact OS matches for host (If you know what OS is running on it, see http://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=6.47%E=4%D=6/29%OT=135%CT=1%CU=30143%PV=Y%DS=1%DC=D%G=Y%M=080027%
OS:TM=5591CF70%P=x86_64-unknown-linux-gnu)SEQ(SP=100%GCD=1%ISR=100%TI=I%CI=
OS:I%II=I%SS=S%TS=A)OPS(O1=M5B4NW8ST11%O2=M5B4NW8ST11%O3=M5B4NW8NNT11%O4=M5
OS:B4NW8ST11%O5=M5B4NW8ST11%O6=M5B4ST11)WIN(w1=2000%w2=2000%w3=2000%w4=2000
OS:%w5=2000%w6=2000)ECN(R=Y%DF=Y%T=80%W=2000%O=M5B4NW8NNS%CC=N%Q=)T1(R=Y%DF
OS:=Y%T=80%S=0%A=S+F=AS%RD=0%Q=)T2(R=Y%DF=Y%T=80%W=0%S=Z%A=S+F=AR%O=0%RD=0%
OS:Q=)T3(R=Y%DF=Y%T=80%W=0%S=Z%A=0%F=AR%O=0%RD=0%Q=)T4(R=Y%DF=Y%T=80%W=0%S=A
OS:%A=0%F=AR%O=0%RD=0%Q=)T5(R=Y%DF=Y%T=80%W=0%S=Z%A=S+F=AR%O=0%RD=0%Q=)T6(R=Y
OS:%DF=Y%T=80%W=0%S=A=0%F=AR%O=0%RD=0%Q=)T7(R=Y%DF=Y%T=80%W=0%S=Z%A=S+F=AR
OS:%O=0%RD=0%Q=)U1(R=Y%DF=N%T=80%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RU
OS:D=G)IE(R=Y%DFI=N%T=80%CD=Z)

Network Distance: 1 hop
OS detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 24.17 seconds
```

Nmap's second OS detection option `--osscan-limit` is a more effective guess because of the required condition a target host must met. The target host must have at least one and one closed TCP port otherwise Nmap will move on to the next host if one has been specified. When 192.168.56.101's firewall is turned off then the `--osscan-limit` scan provides nearly the same information as the `--osscan-guess` option. Such as an identically TCP/IP fingerprint, the statement that says no exact OS matches has been found, but didn't make any sort of OS guesses. Figure 20 will display these results.

Jake Haaksma, jlhaaksma@live.com



Figure 20 – Nmap OS Detection Limit Scan against firewall turned off

```

root@kali:~# nmap -O --osscan-limit 192.168.56.101

Starting Nmap 6.47 ( http://nmap.org ) at 2015-06-29 18:04 CDT
Nmap scan report for 192.168.56.101
Host is up (0.00033s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
MAC Address: 08:00:27:07:07:00 (Cadmus Computer Systems)
No exact OS matches for host (If you know what OS is running on it, see http://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=6.47%E=4%D=6/29%OT=135%CT=1%CU=30273%PV=Y%DS=1%DC=D%G=Y%M=080027%
OS:TM=5591CF22%P=x86_64-unknown-linux-gnu)SEQ(SP=FE%GCD=1%ISR=10E%TI=I%CI=I
OS:%II=I%SS=S%TS=A)OPS(O1=M5B4NW8ST11%02=M5B4NW8ST11%03=M5B4NW8NNT11%04=M5B
OS:4NW8ST11%05=M5B4NW8ST11%06=M5B4ST11)WIN(W1=2000%W2=2000%W3=2000%W4=2000%
OS:W5=2000%W6=2000)ECN(R=Y%DF=Y%T=80%W=2000%O=M5B4NW8NNS%CC=N%Q=)T1(R=Y%DF=
OS:Y%T=80%S=0%A=S+F=AS%RD=0%Q=)T2(R=Y%DF=Y%T=80%W=0%S=Z%A=S+F=AR%0=%RD=0%Q
OS:)=)T3(R=Y%DF=Y%T=80%W=0%S=Z%A=0%F=AR%0=%RD=0%Q=)T4(R=Y%DF=Y%T=80%W=0%S=A
OS:A=0%F=R%0=%RD=0%Q=)T5(R=Y%DF=Y%T=80%W=0%S=Z%A=S+F=AR%0=%RD=0%Q=)T6(R=Y
OS:DF=Y%T=80%W=0%S=A%0%F=R%0=%RD=0%Q=)T7(R=Y%DF=Y%T=80%W=0%S=Z%A=S+F=AR%
OS:0=%RD=0%Q=)U1(R=Y%DF=N%T=80%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD
OS:)=G)IE(R=Y%DFI=N%T=80%CD=Z)

Network Distance: 1 hop

OS detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 24.15 seconds

```

Thus, both OS detection scans were unable to correctly guess that the target's OS is running Windows 10 Technical Preview. There was not one exact fingerprint match in Nmap's OS database, nmap-os-db. The next screenshot is an attempt to generate a suitable fingerprint for Windows 10 Technical Preview. In order to create a fingerprint, Nmap requires the command to be 'nmap -O -sV -T4 -d <target>'. This particular command starts nmap, enables OS detection, probes open ports to determine service/version information, set the timing template to 4 seconds, and increases the debugging level. The scan was able to successfully create a TCP/IP fingerprint for the target host's OS. In order to further improve Nmap's OS database, the fingerprint in Figure 21 has been sent to Nmap's fingerprint submitter webpage.

Figure 21 – Nmap generate own fingerprint against firewall turned off

```

root@kali:~# nmap -O -sV -T4 -d 192.168.56.101

Starting Nmap 6.47 ( http://nmap.org ) at 2015-06-29 18:43 CDT
PORTS: Using top 1000 ports found open (TCP:1000, UDP:0, SCTP:0)
----- Timing report -----
  hostgroups: min 1, max 100000
  rtt-timeouts: init 500, min 100, max 1250
  max-scan-delay: TCP 10, UDP 1000, SCTP 10
  parallelism: min 0, max 0
  max-retries: 6, host-timeout: 0
  min-rate: 0, max-rate: 0
-----
NSE: Using Lua 5.2.
NSE: Script Arguments seen from CLI:
NSE: Loaded 29 scripts for scanning.
Initiating ARP Ping Scan at 18:43
Scanning 192.168.56.101 [1 port]
Packet capture filter (device eth0): arp and arp[18:4] = 0x08002738 and arp[22:2] = 0x4595
Completed ARP Ping Scan at 18:43, 0.00s elapsed (1 total hosts)
Overall sending rates: 1092.90 packets / s, 45901.64 bytes / s.
mass_rdns: Using DNS server 75.75.76.76
mass_rdns: Using DNS server 75.75.75.75
Initiating Parallel DNS resolution of 1 host. at 18:43
mass_rdns: 13.01s 0/1 [#: 2, OK: 0, NX: 0, DR: 0, SF: 0, TR: 4]
Completed Parallel DNS resolution of 1 host. at 18:44, 13.01s elapsed
DNS resolution of 1 IPs took 13.01s. Mode: Async [#: 2, OK: 0, NX: 0, DR: 1, SF: 0, TR: 4, CN: 0]
Initiating SYN Stealth Scan at 18:44
Scanning 192.168.56.101 [1000 ports]
Packet capture filter (device eth0): dst host 192.168.56.102 and (icmp or icmp6 or ((tcp or udp or sctp) and (src host 192.168.56.101)))
Discovered open port 139/tcp on 192.168.56.101
Increased max_successful_tryno for 192.168.56.101 to 1 (packet drop)
Discovered open port 445/tcp on 192.168.56.101
Discovered open port 135/tcp on 192.168.56.101
Completed SYN Stealth Scan at 18:44, 1.10s elapsed (1000 total ports)
Overall sending rates: 914.92 packets / s, 40256.39 bytes / s.
Initiating Service scan at 18:44
Scanning 3 services on 192.168.56.101
Completed Service scan at 18:44, 6.01s elapsed (3 services on 1 host)
Packet capture filter (device eth0): dst host 192.168.56.102 and (icmp or (tcp and (src host 192.168.56.101)))
Initiating OS detection (try #1) against 192.168.56.101
OS detection timingRatio() == (1435621450.677 - 1435621450.171) * 1000 / 500 == 1.010
Retrying OS detection (try #2) against 192.168.56.101
OS detection timingRatio() == (1435621452.504 - 1435621452.001) * 1000 / 500 == 1.004
Retrying OS detection (try #3) against 192.168.56.101
OS detection timingRatio() == (1435621454.335 - 1435621453.831) * 1000 / 500 == 1.008
Retrying OS detection (try #4) against 192.168.56.101
OS detection timingRatio() == (1435621457.660 - 1435621457.158) * 1000 / 500 == 1.004
Retrying OS detection (try #5) against 192.168.56.101
OS detection timingRatio() == (1435621459.484 - 1435621458.981) * 1000 / 500 == 1.006
NSE: Script scanning 192.168.56.101.
NSE: Starting runlevel 1 (of 1) scan.
Nmap scan report for 192.168.56.101
Host is up, received arp-response (0.00032s latency).
Scanned at 2015-06-29 18:43:49 CDT for 30s
Not shown: 997 closed ports
Reason: 997 resets
PORT      STATE SERVICE      REASON  VERSION
135/tcp   open  msrpc        syn-ack Microsoft Windows RPC
139/tcp   open  netbios-ssn syn-ack
445/tcp   open  netbios-ssn syn-ack
MAC Address: 08:00:27:07:07:00 (Cadmus Computer Systems)
No exact OS matches for host (If you know what OS is running on it, see http://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=6.47%E=4%D=6/29%OT=135%CT=1%CU=37116%PV=Y%DS=1%DC=D%G=Y%M=080027%
OS:TM=5591D853%P=x86_64-unknown-linux-gnu)SEQ(SP=103%GCD=1%ISR=109%TI=I%CI=
OS:I%II=I%SS=S%TS=A)OPS(O1=M5B4NW8ST11%O2=M5B4NW8ST11%O3=M5B4NW8NNT11%O4=M5
OS:B4NW8ST11%O5=M5B4NW8ST11%O6=M5B4ST11)WIN(W1=2000%W2=2000%W3=2000%W4=2000
OS:%W5=2000%W6=2000)ECN(R=Y%DF=Y%T=80%W=2000%O=M5B4NW8NNS%CC=N%Q=)T1(R=Y%DF
OS:=Y%T=80%S=0%A=S+%F=AS%RD=0%Q=)T2(R=Y%DF=Y%T=80%W=0%S=Z%A=S%F=AR%O=%RD=0%
OS:Q=)T3(R=Y%DF=Y%T=80%W=0%S=Z%A=0%F=AR%O=%RD=0%Q=)T4(R=Y%DF=Y%T=80%W=0%S=A

```

```

OS:%A=0%F=R%0=%RD=0%Q=) T5(R=Y%DF=Y%T=80%W=0%S=Z%A=S+%F=AR%0=%RD=0%Q=) T6(R=Y
OS:DF=Y%T=80%W=0%S=A%A=0%F=R%0=%RD=0%Q=) T7(R=Y%DF=Y%T=80%W=0%S=Z%A=S+%F=AR
OS:%0=%RD=0%Q=) U1(R=Y%DF=N%T=80%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RU
OS:D=G) IE(R=Y%DFI=N%T=80%CD=Z)

Uptime guess: 4.068 days (since Thu Jun 25 17:07:00 2015)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=259 (Good luck!)
IP ID Sequence Generation: Incremental
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
Final times for host: srtt: 315 rttvar: 67 to: 100000

Read from /usr/bin/./share/nmap: nmap-mac-prefixes nmap-os-db nmap-payloads nmap-service-probes nmap-services.
OS and Service detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 30.32 seconds
Raw packets sent: 1090 (51.514KB) | Rcvd: 1081 (46.690KB)

```

### 3.0 Future Work

This experiment allows for future work to extend the knowledge of the Windows 10 OS fingerprint. A proposal of conducting this experiment with the official Windows 10 OS could produce additional information. Windows 10 for customer PCs and tablets is set to be officially released on July 29, 2015. It would be interesting to discover if the default open ports, services, and the OS fingerprint would differ from Windows 10 Technical Preview; also if having the OS runs on a virtual machine as opposed on a standard workstation.

### 4.0 Conclusion

The process of conducting network exploration and security auditing will yield a vast amount of information about a particular network. This information is currently what a malicious attacker will attempt to acquire. “A host that runs a sniffer can easily embezzle private and confidential information of network users. Hence detection of a sniffer is an essential task to maintain network security” (Khan, Qureshi, Khan, 2012). A security professional must understand their environment in order to properly defend their network(s) and a large part of the picture is to know what operating systems are running within a network. While Nmap was unable to precisely identify the target’s OS it was able to determine the target’s open/closed ports, running services, and protocols. The fingerprint for Windows 10 Technical Preview was created to improve the accuracy of Nmap’s OS database and for the benefit of the security community.

Jake Haaksma, jlhaaksma@live.com

## 5.0 Appendix A – Additional Scans against Host's firewall OFF

Nmap TCP NULL Scan against firewall turned off

```
root@kali:~# nmap -sN 192.168.56.101

Starting Nmap 6.47 ( http://nmap.org ) at 2015-06-29 17:48 CDT
Nmap scan report for 192.168.56.101
Host is up (0.00023s latency).
All 1000 scanned ports on 192.168.56.101 are closed
MAC Address: 08:00:27:07:07:00 (Cadmus Computer Systems)

Nmap done: 1 IP address (1 host up) scanned in 14.14 seconds
```

Nmap FIN Scan against firewall turned off

```
root@kali:~# nmap -sF 192.168.56.101

Starting Nmap 6.47 ( http://nmap.org ) at 2015-06-29 17:50 CDT
Nmap scan report for 192.168.56.101
Host is up (0.00025s latency).
All 1000 scanned ports on 192.168.56.101 are closed
MAC Address: 08:00:27:07:07:00 (Cadmus Computer Systems)

Nmap done: 1 IP address (1 host up) scanned in 14.14 seconds
```

Nmap Xmas Scan against firewall turned off

```
root@kali:~# nmap -sX 192.168.56.101

Starting Nmap 6.47 ( http://nmap.org ) at 2015-06-29 17:51 CDT
Nmap scan report for 192.168.56.101
Host is up (0.00015s latency).
All 1000 scanned ports on 192.168.56.101 are closed
MAC Address: 08:00:27:07:07:00 (Cadmus Computer Systems)

Nmap done: 1 IP address (1 host up) scanned in 14.14 seconds
```



Nmap ACK Scan against firewall turned off

```
root@kali:~# nmap -sA 192.168.56.101

Starting Nmap 6.47 ( http://nmap.org ) at 2015-06-29 17:52 CDT
Nmap scan report for 192.168.56.101
Host is up (0.00020s latency).
All 1000 scanned ports on 192.168.56.101 are unfiltered
MAC Address: 08:00:27:07:07:00 (Cadmus Computer Systems)

Nmap done: 1 IP address (1 host up) scanned in 14.14 seconds
```

Nmap Window Scan against firewall turned off

```
root@kali:~# nmap -sW 192.168.56.101

Starting Nmap 6.47 ( http://nmap.org ) at 2015-06-29 18:24 CDT
Nmap scan report for 192.168.56.101
Host is up (0.00030s latency).
All 1000 scanned ports on 192.168.56.101 are closed
MAC Address: 08:00:27:07:07:00 (Cadmus Computer Systems)

Nmap done: 1 IP address (1 host up) scanned in 14.14 seconds
```

Nmap Maimon Scan against firewall turned off

```
root@kali:~# nmap -sM 192.168.56.101

Starting Nmap 6.47 ( http://nmap.org ) at 2015-06-29 17:53 CDT
Nmap scan report for 192.168.56.101
Host is up (0.00018s latency).
All 1000 scanned ports on 192.168.56.101 are closed
MAC Address: 08:00:27:07:07:00 (Cadmus Computer Systems)

Nmap done: 1 IP address (1 host up) scanned in 13.09 seconds
```

## 6.0 References

- Alsaleh, M., & Oorschot, P. (2013). Evaluation in the absence of absolute ground truth: toward reliable evaluation methodology for scan detectors. *International Journal Of Information Security*, 12(2), 97-110. doi:10.1007/s10207-012-0178-1
- Anbar, M., Manasrah, A., & Manickam, S. (2012). Statistical cross-relation approach for detecting TCP and UDP random and sequential network scanning (SCANS). *International Journal Of Computer Mathematics*, 89(15), 1952-1969. doi:10.1080/00207160.2012.696621
- Budhrani, R., & Sridaran, R. (2015). Wireless Local Area Networks: Threats and Their Discovery Using WLANs Scanning Tools. *International Journal Of Advanced Networking & Applications*, 137-150.
- Council, EC. (2015). The 5 Phases Every Hacker Must Follow. 1-8.
- Chang-Su, M., & Sun-Hyung, K. (2014). A Study on the Integrated Security System based Real-time Network Packet Deep Inspection. *International Journal Of Security & Its Applications*, 8(1), 113-122. doi:10.14257/ijisa.2014.8.1.11
- Doria, A., Ed., Hadi Salim, J., Ed., Haas, R., Ed., Khosravi, H., Ed., Wang, W., Ed., Dong, L., Gopal, R., and J. Halpern, "Forwarding and Control Element Separation (ForCES) Protocol Specification", RFC 5810, DOI 10.17487/RFC5810, March 2010, <<http://www.rfc-editor.org/info/rfc5810>>.
- ELEJLA, O. E., JANTAN, A. B., & AHMED, A. A. (2014). THREE LAYERS APPROACH FOR NETWORK SCANNING DETECTION. *Journal Of Theoretical & Applied Information Technology*, 70(2), 251-264.
- Gibson, S. (2008). GRC | Port Authority, for Internet Port 500. Retrieved July 7, 2015.
- Gibson, S. (2008). GRC | Port Authority, for Internet Port 1900. Retrieved July 7, 2015.
- Khan, A. N., Qureshi, K., & Khan, S. (2012). An Intelligent Approach of Sniffer Detection. *International Arab Journal Of Information Technology (IAJIT)*, 9(1), 9-15.
- Lyon, G. (1997). Nmap - Free Security Scanner For Network Exploration & Security Audits. Retrieved July 6, 2015.
- NetBIOS Over TCP/IP. (2005). Retrieved July 16, 2015.

Jake Haaksma, jlhaaksma@live.com

- Peidai, X., Xicheng, L., & Yongjun, W. (2013). Eliminate Evading Analysis Tricks in Malware using Dynamic Slicing. *International Journal Of Security & Its Applications*, 7(3), 357-365.
- Posey, B. (2006, November 29). An Overview of Link Local Multicast Name Resolution. Retrieved July 7, 2015.
- Sultana, F., Charles, S., & Govardhan, A. (2013). A Real Time Intrusion Aggregation And Prevention Technique. *International Journal Of Advanced Networking & Applications*, 4(5), 1719-1724.
- Yue-Bin, L., Bao-Sheng, W., & Gui-Lin, C. (2015). Analysis of Port Hopping for Proactive Cyber Defense. *International Journal Of Security & Its Applications*, 9(2), 123-134. doi:10.14257/ijisia.2015.9.2.12
- What Is RPC? (2003, March 28). Retrieved July 16, 2015.