300143 Network Security Practical 8

Introduction

This report will examine the processes of UDP port scanning, service detection and operating system (OS) detection using the network scanner Nmap. After an explanation of the core concepts this report will show how Nmap is used to scan UDP port and detect the running OS and other services on another computer.

Introduction to Nmap

Nmap is a piece of software used for network discovery and examination and is also useful in the area of cyber security for determining how secure a network is as well as testing for potential vulnerabilities (Lyon, 2020). For this report, we will be focused on Nmap's ability to scan ports to determine:

- which ones are being used or not;
- what services are using those ports; and
- what OS the target computer is using.

Nmap can be used in a CLI form or through their GUI application called 'Zenmap'.

UDP Port Scanning, Service and OS Detection

UDP Port Scanning

Port scanning, in general, is the process of probing individual ports to determine if that port is open or closed (paloalto NETWORKS, n.d.). An open port is one that is currently being used by a service to send and receive information over a network, whereas a closed port isn't currently being used (Borges, 2019).

There are a variety of different methods in determining the status of a port and the use of a technique is determined by which transport layer protocol (TCP or UDP) the service using the port is running (avast, n.d.).

Service Detection

Once a port has been identified the process of service detection can begin. Port scanners will normally have a database of services and the ports they are commonly run on (Lyon, 2020a).

However, simply matching the port number with the common service is not a guarantee that is the service listening on that port. For more accurate data the scanner can send more packets to the service listening on that port to extract more information such as the application name, version, what device the application is running on and OS family (Lyon, 2020a).

Nmap carries out service detection by using the wide variety of probe tests stored in its 'nmap-service-probes' database which gives instructions on what information to send and what to look for in the response. If information in the response matches with the expected response for that probe then Nmap identifies that service as running on the port.

OS Detection

OS Detection is the process of identifying the current OS being used by the target machine. This is achieved by sending TCP and UDP packets to the target with each packet being used to perform a specific test (Lyon, 2020b). Responses from each test are examined and the data is compared to known responses for each OS in a database. If the captured response data matches with an entry in the OS database then it can be said that the target is running the matched OS.

Techniques/Methods for UDP Port Scanning

Due to the fact that UDP is a connection-less protocol, methods such as SYN scan, ACK scan or XMAS scan are not going to be effective as they rely on the fact that the TCP protocol establishes a connection using a three-way handshake before data is sent.

There are two methods for determining the status of a port that is using the UDP protocol. Each method uses the presence of a response to determine the status of the port (Lyon, 2020c).

The first method is simply sending a UDP packet to every port that is being targeted. The packet sent will be empty. When the target receives this empty packet it is common for the service listening on the port to discard the packet (Lyon, 2020c). Therefore, if no response is received from that port then it can assumed that the port is open (Lyon, 2020c). However, it is also a possibility that a firewall or other device that filters network traffic drops the packet before it ever reaches its destination. This causes no response to be sent even if the port is closed and therefore, the port will be listed as open. Meaning, there is the chance that some ports listed as open are false positives and actually closed.

The second method is used on ports that are running common UDP protocols. These ports are sent a packet with a payload targeted specifically for the service running on that port and is meant to generate an application layer response (Lyon, 2020c). An example of a port where this method can be used is on Port 53. The DNS protocol is the service listening on Port 53 and will generate a response if a DNS server is present on the network (Petters, 2020).

Results

In order to scan the target computer we first needed to know its IP address. To do this we opened a command prompt on the target computer and used the command ‘ipconfig’. From this we could see the target’s IP address was 192.168.0.207.

Once we ascertained that, we could open Zenmap on the attacker computer and begin the process of UDP port scanning, service detection and OS detection.

UDP Scanning

Top 1000 UDP Port Scan: In order to determine what the state of all UDP ports on the target computer the following command was used:

```
nmap -sU -v 192.168.0.207
```

This command would scan the top 1000 commonly used UDP ports. The results from the completed scan (Figure 1) show that out of the 1000 scanned ports:

- one (1) is open;
- six (6) are possibly open; and
- 993 are closed.

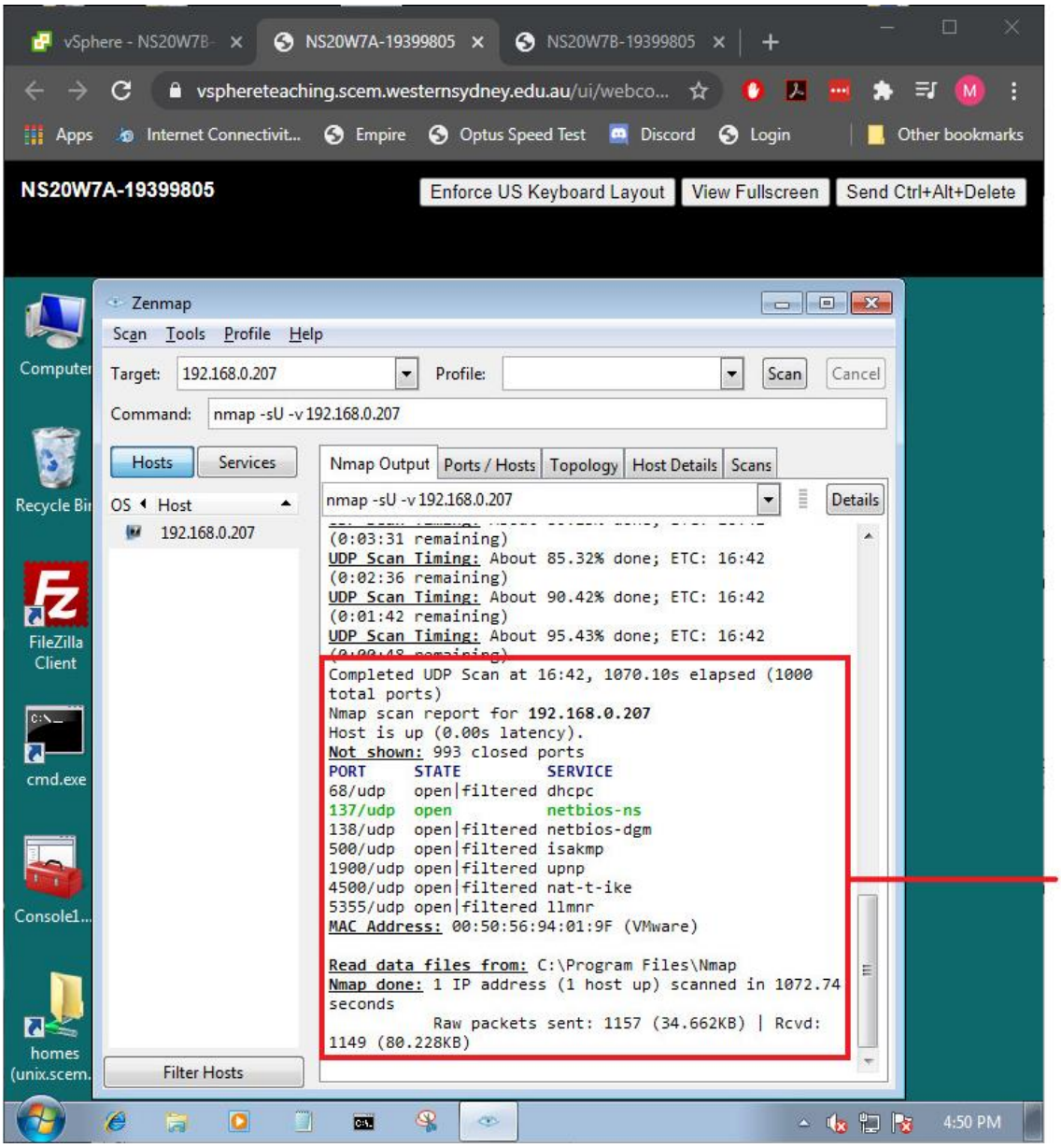

Figure 1: UDP Port Scan Results

Port 53 & 137 Scan with Traffic Capture: In order to see the difference between how Nmap scans an open port compared to a closed port the closed Port 13 (Figure 2) and the open Port 137 (Figure 3) was scanned and traffic was captured with Wireshark.

In order to scan those specific port the following commands were used:

`nmap -p 13 192.168.0.207` and
`nmap -p 137 192.168.0.207`

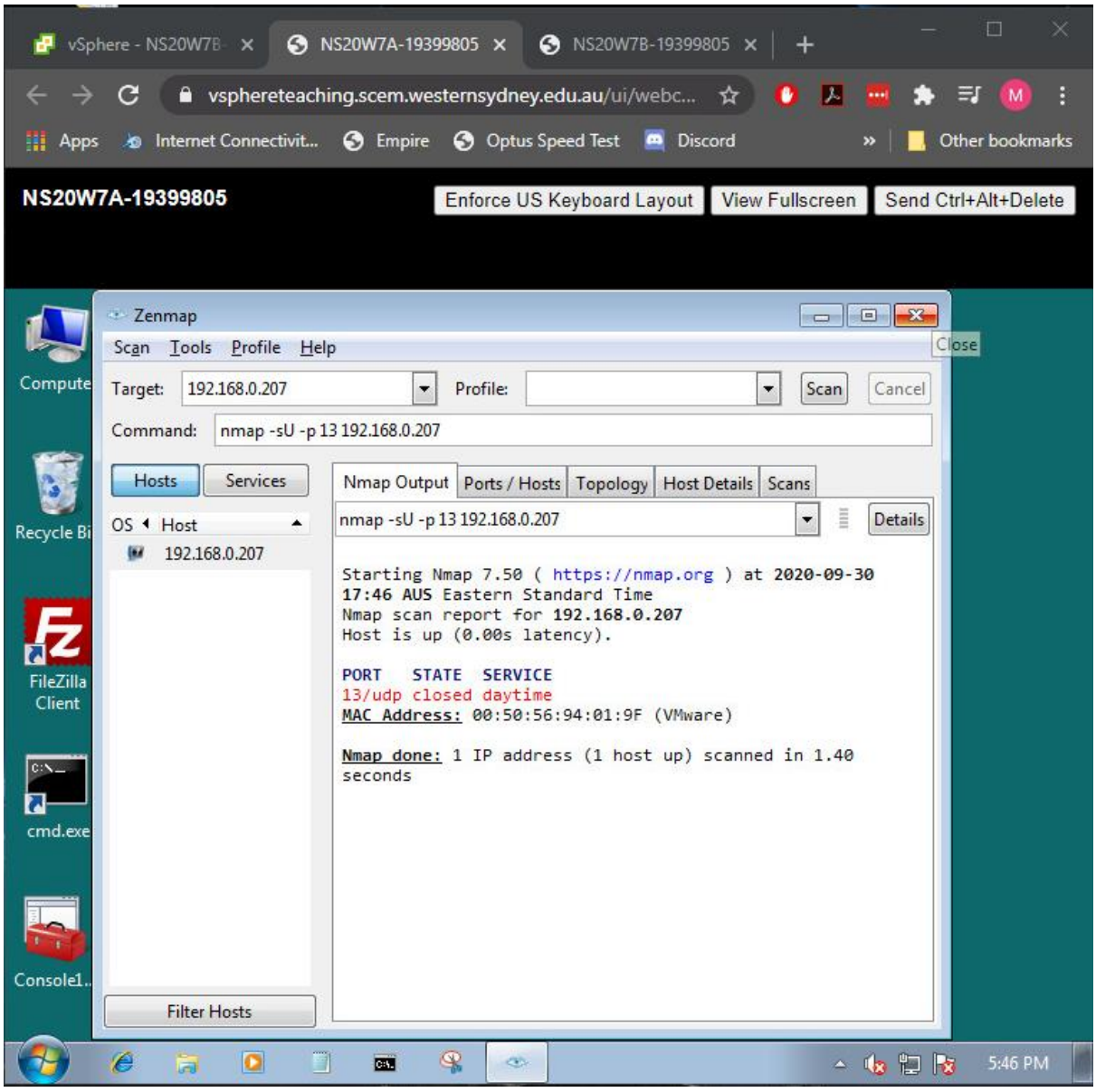

Figure 2: Port 13 Scan Results

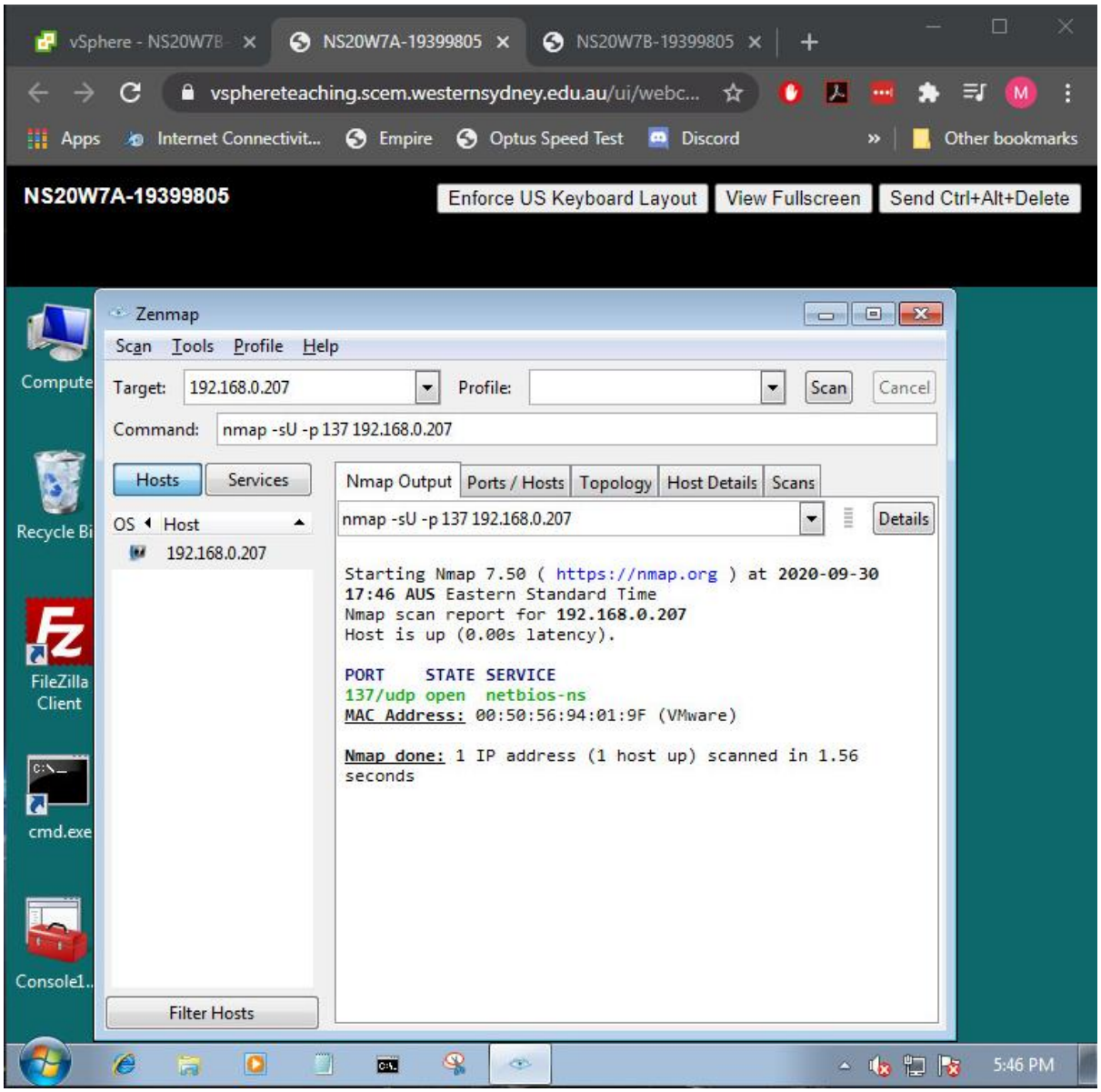

Figure 3: Port 137 Scan Results

Traffic Analysis: With the two port scans completed we can filter the packets sent to those ports in Wireshark using the filter:

```
udp.port == 13 || udp.port == 137
```

Using this filter we see the following results.

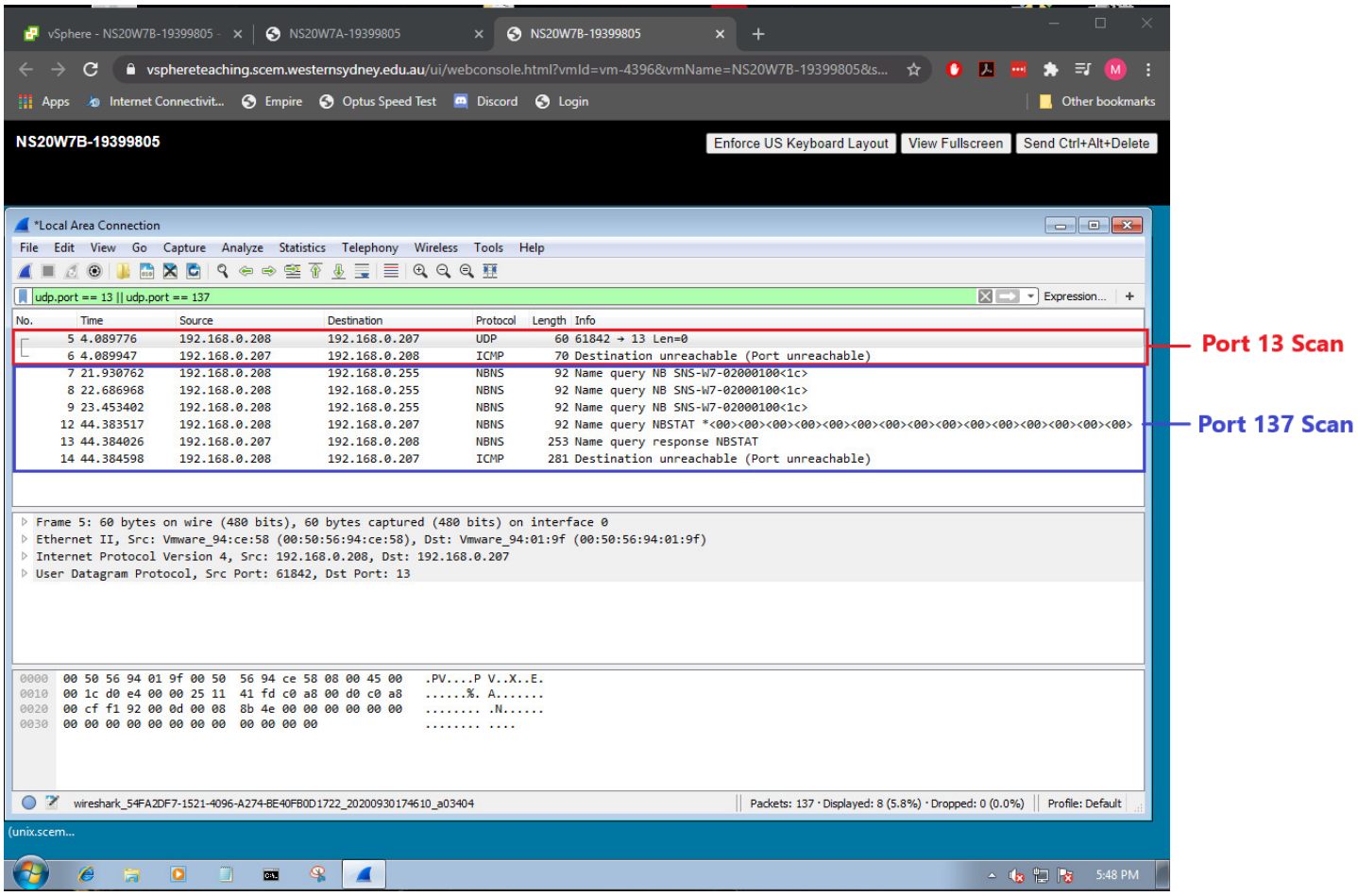

Figure 4: UDP Ports 13 and 137 Scan Traffic

From the screenshot (Figure 4) we see that in order to scan Port 13 a single UDP packet is sent and as the port is closed then a reply is sent informing Nmap that the port is unreachable. If no response was sent, then Nmap would display the port as open|filtered. Port 137 is a port where Nmap can send a UDP packet with a custom payload unlike Port 13, where the payload was empty. Using the custom payload and the response from the target computer Nmap is able to definitively say that Port 113 is open instead of simply open|filtered.

Service Detection

Scan TCP ports: In order to determine what TCP ports were open the following command was used:

```
nmap -sT 192.168.0.207
```

With the following results (Figure 5).

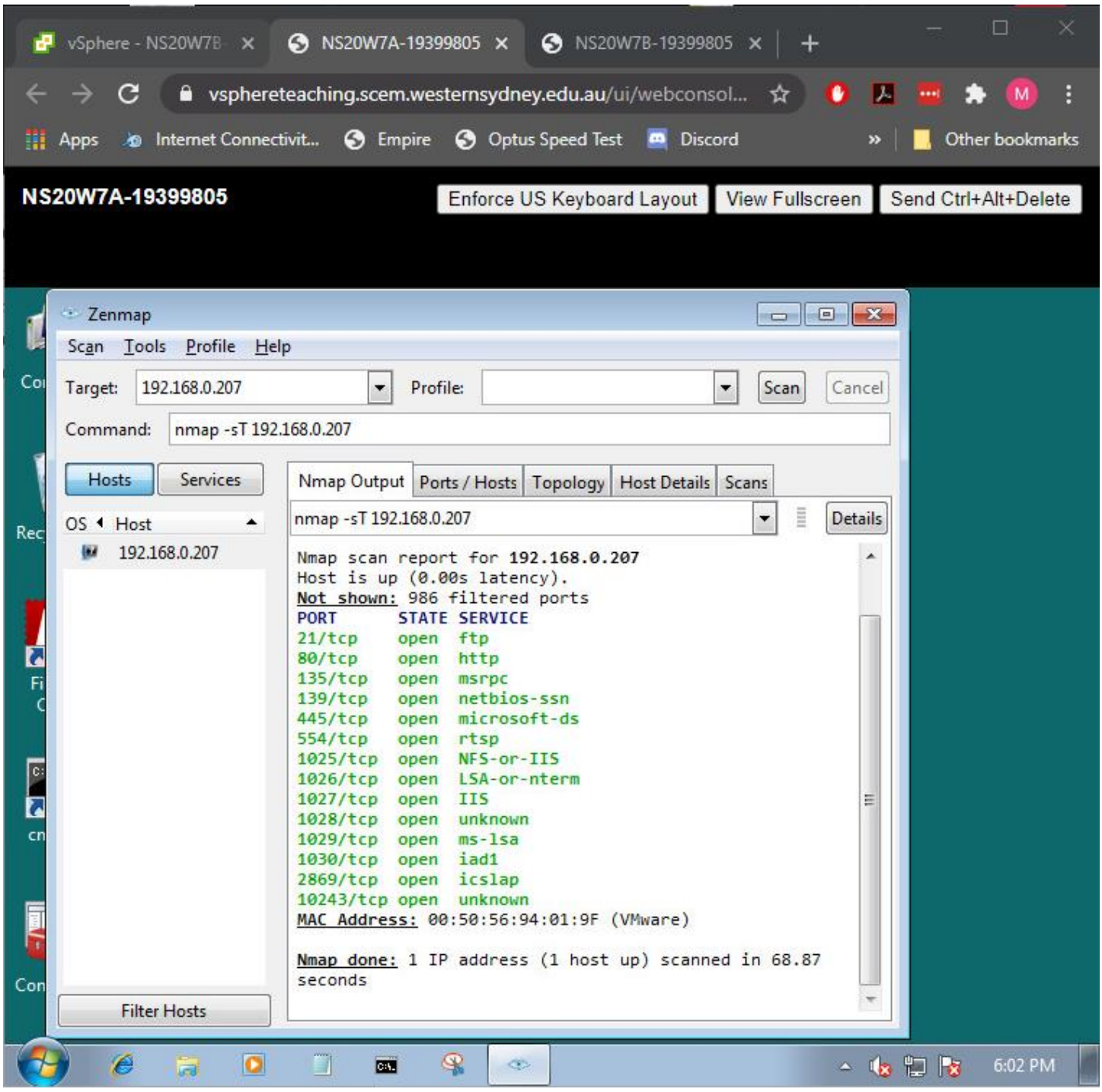

Figure 5: TCP Port Scan Results

Service Detection: Once open ports were known service detection could be performed. For this exercise, Port 21 was chosen to have service detection done. The following command was used:

```
nmap -sT -sV -p 21 192.168.0.207
```

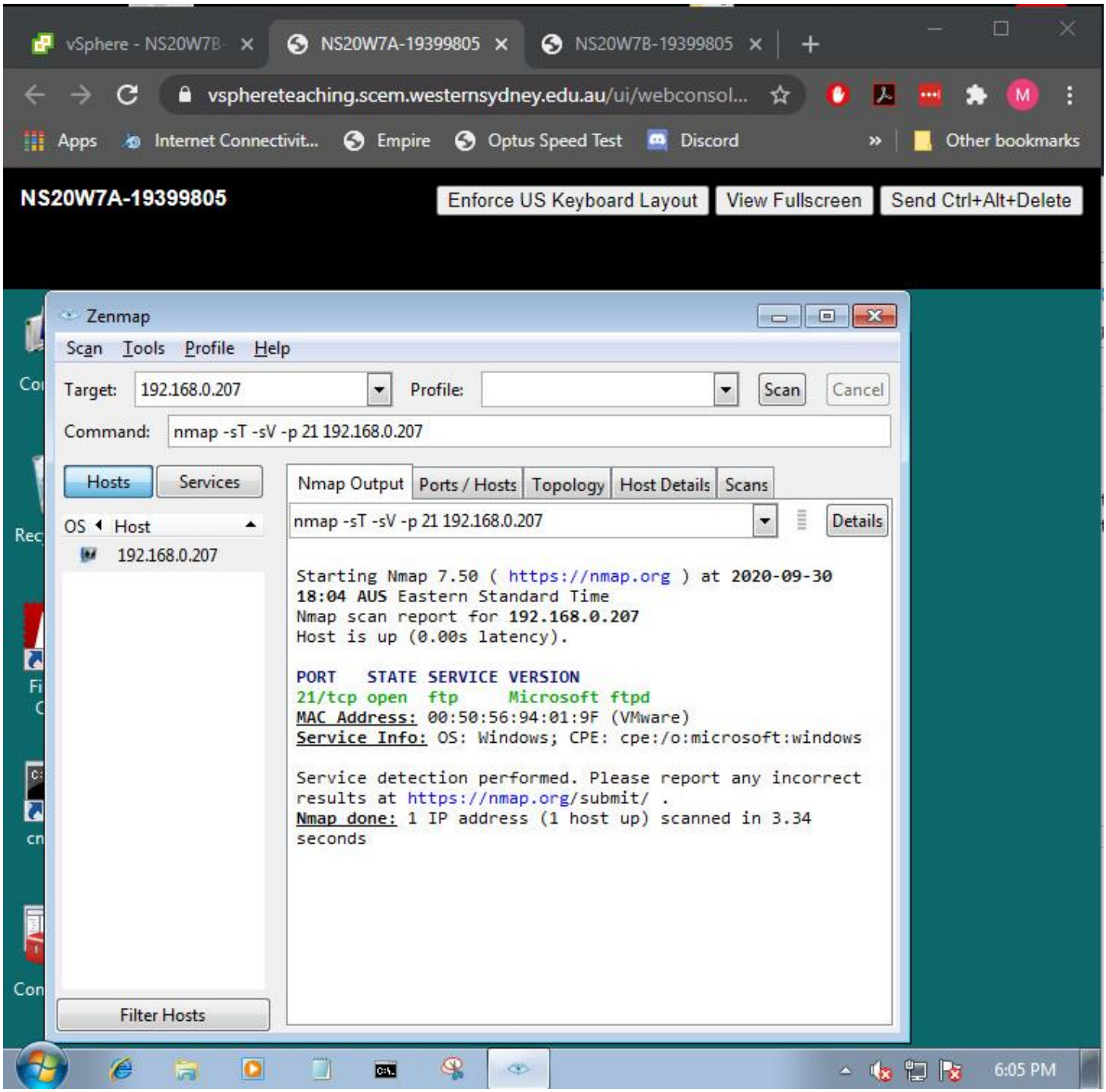

Figure 6: Port 21 Service Detection Results

From the screenshot (Figure 6) it can be seen that the File Transfer Protocol (FTP) is being run on Port 21 using the Microsoft 'ftpd' version of the protocol.

Traffic Analysis: The following traffic (Figure 7) was captured by Wireshark during the service detection of Port 21.

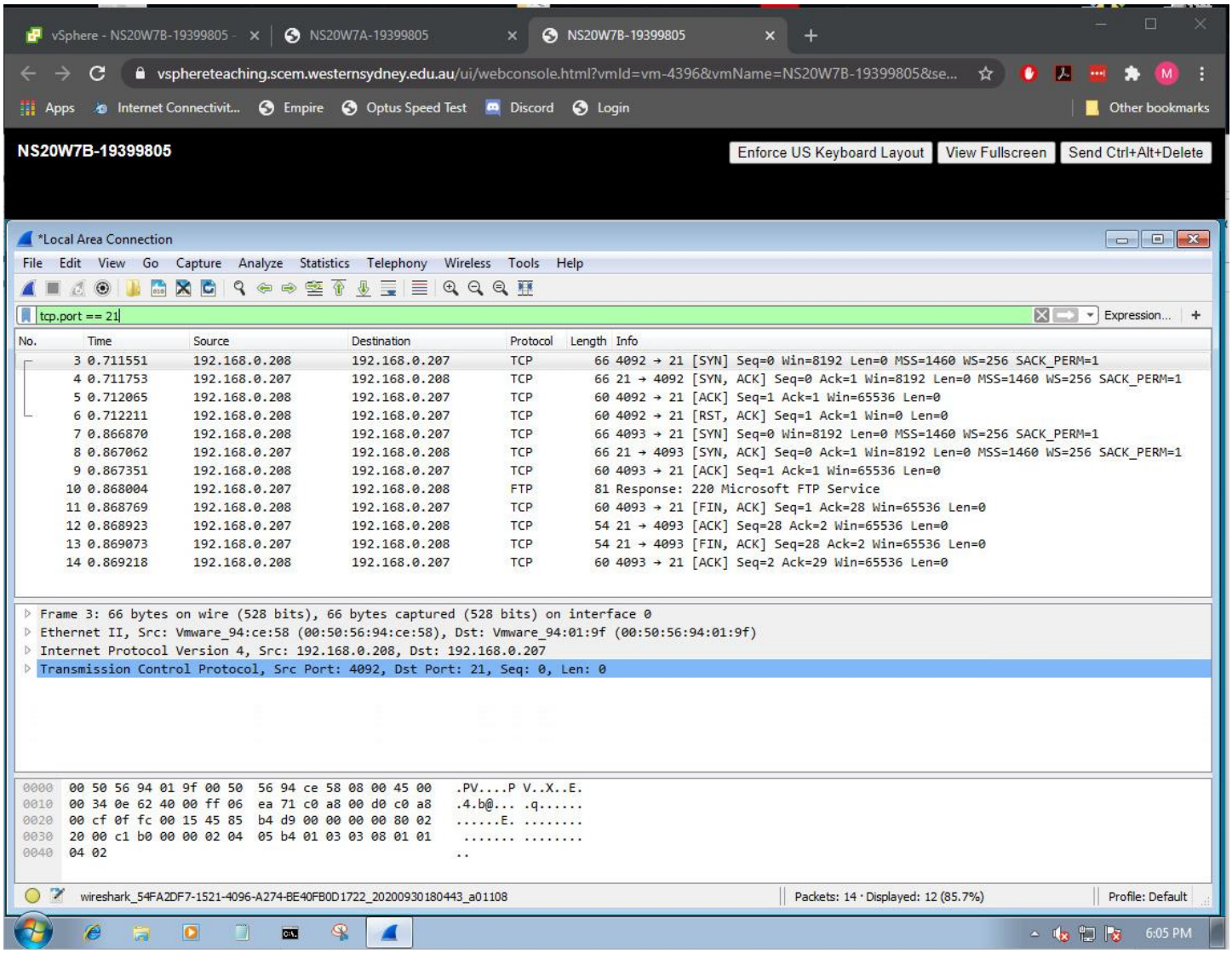

Figure 7: Port 21 Service Detection Traffic

Multiple TCP connections were made to the port with the goal of retrieving a piece of information from the service. Nmap then compared the information gathered with its database and found a match to the Microsoft 'ftpd' version of the service.

OS Detection

For increased efficiency detecting what OS the target computer is running, Nmap requires an open and closed port for scanning. Although it can also be done by only providing an open port. To detect the OS running on the target computer and analyse the traffic captured on Wireshark both methods were used. Initially Ports 21 and 22 were specified to be used by Nmap.

To detect the OS using Ports 21 and 22 the following command was run (Figure 8):

```
nmap -O -p 21,22 192.168.0.207
```

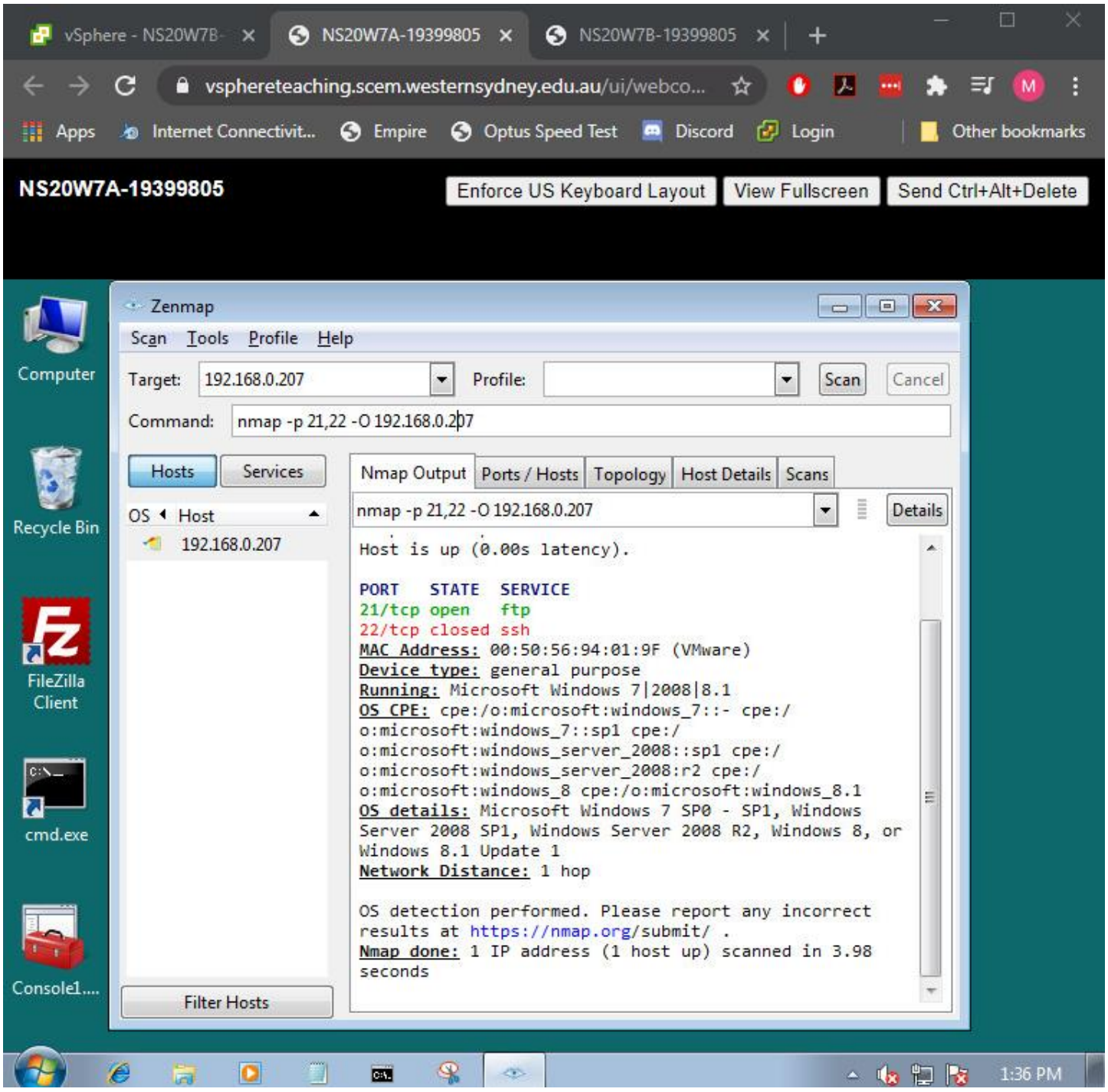

Figure 8: OS Detection Using Ports 21 and 22

The following traffic was captured (Figure 9).

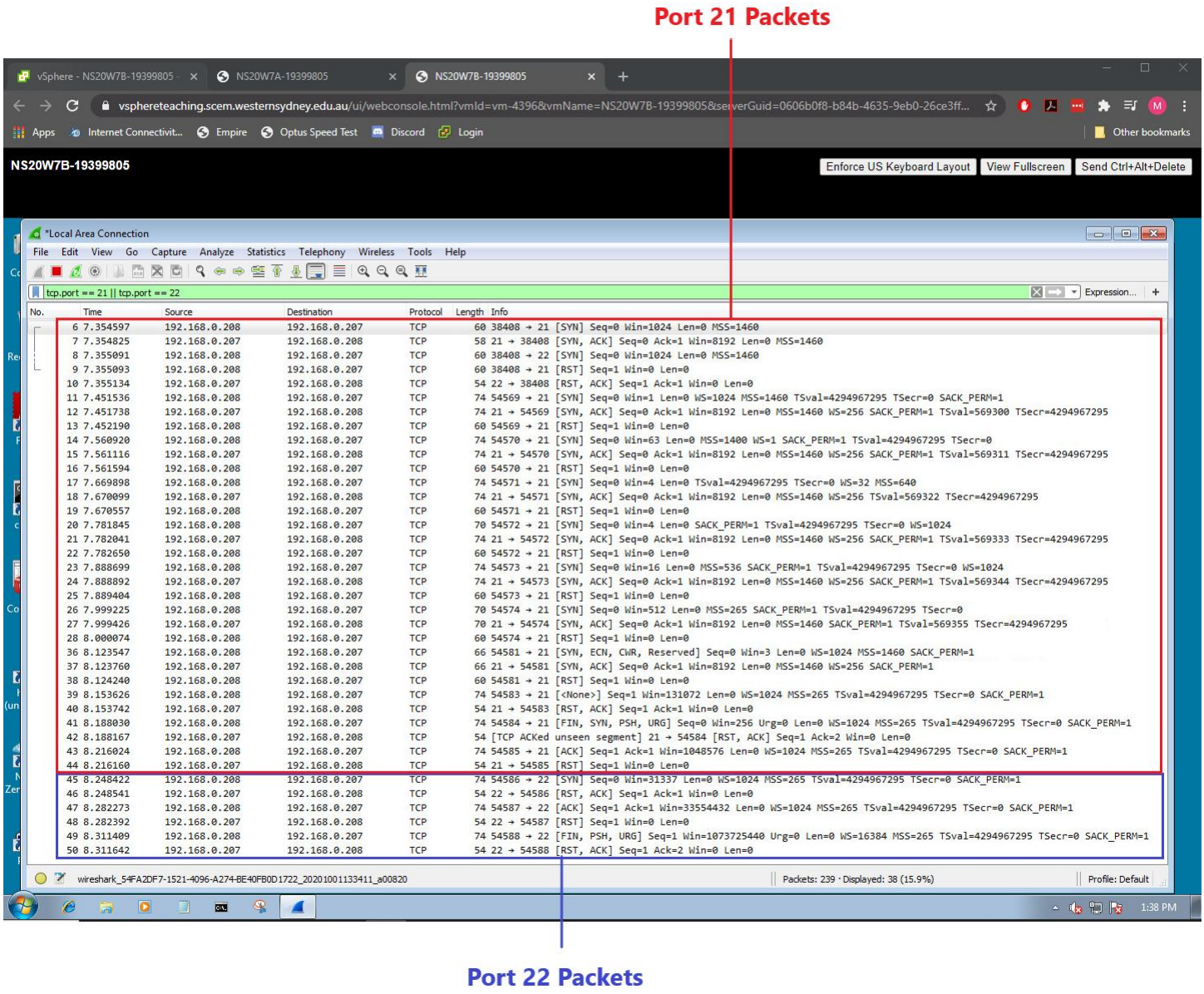

Figure 9: OS Detection Using Ports 21 and 22 Traffic

For the second test only Port 21 (Figure 10) was used to detect the OS using the following command:

```
nmap -O -p 21 192.168.0.207
```

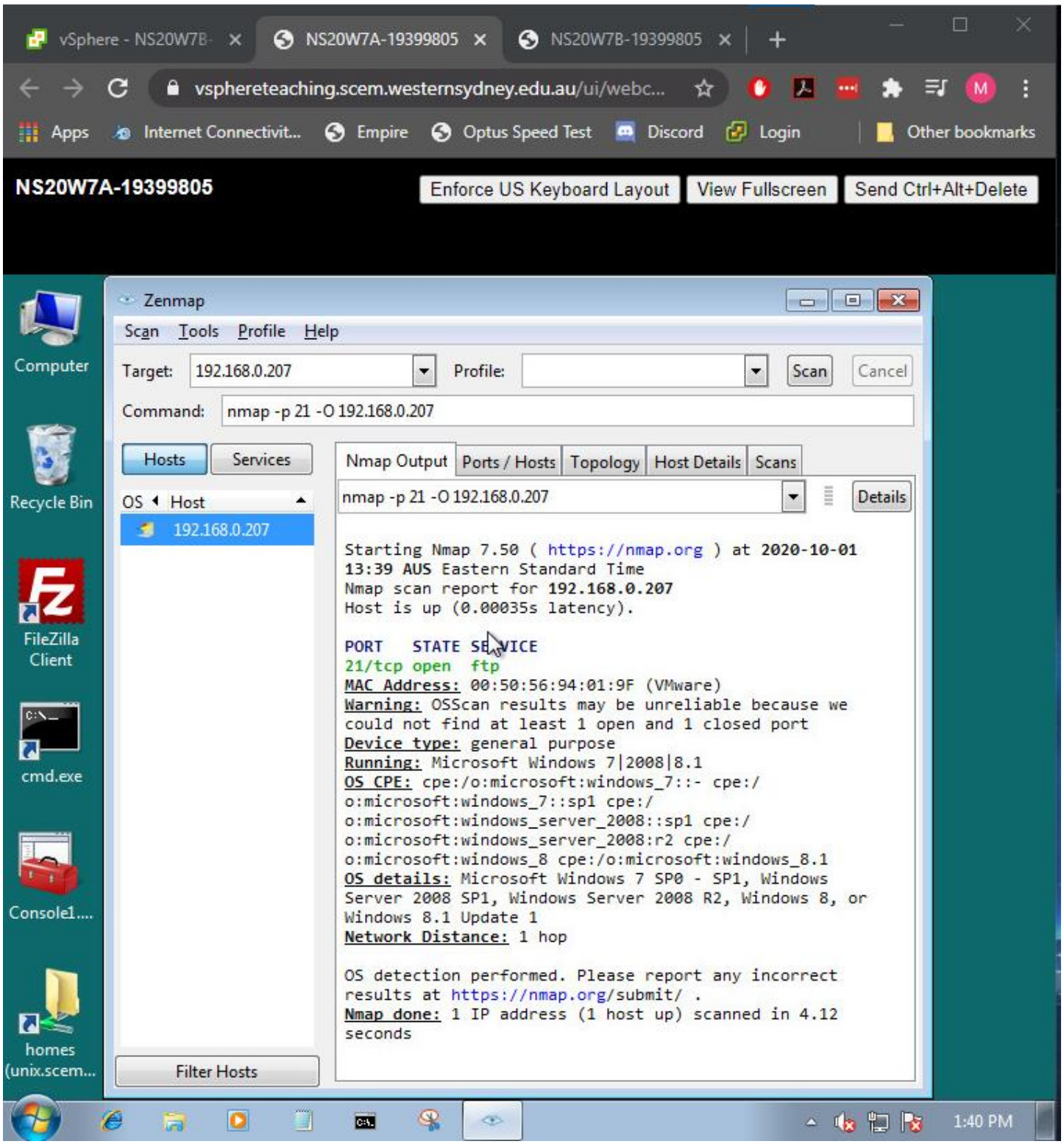

Figure 10: OS Detection Using Port 21

The following traffic was captured (Figure 11).

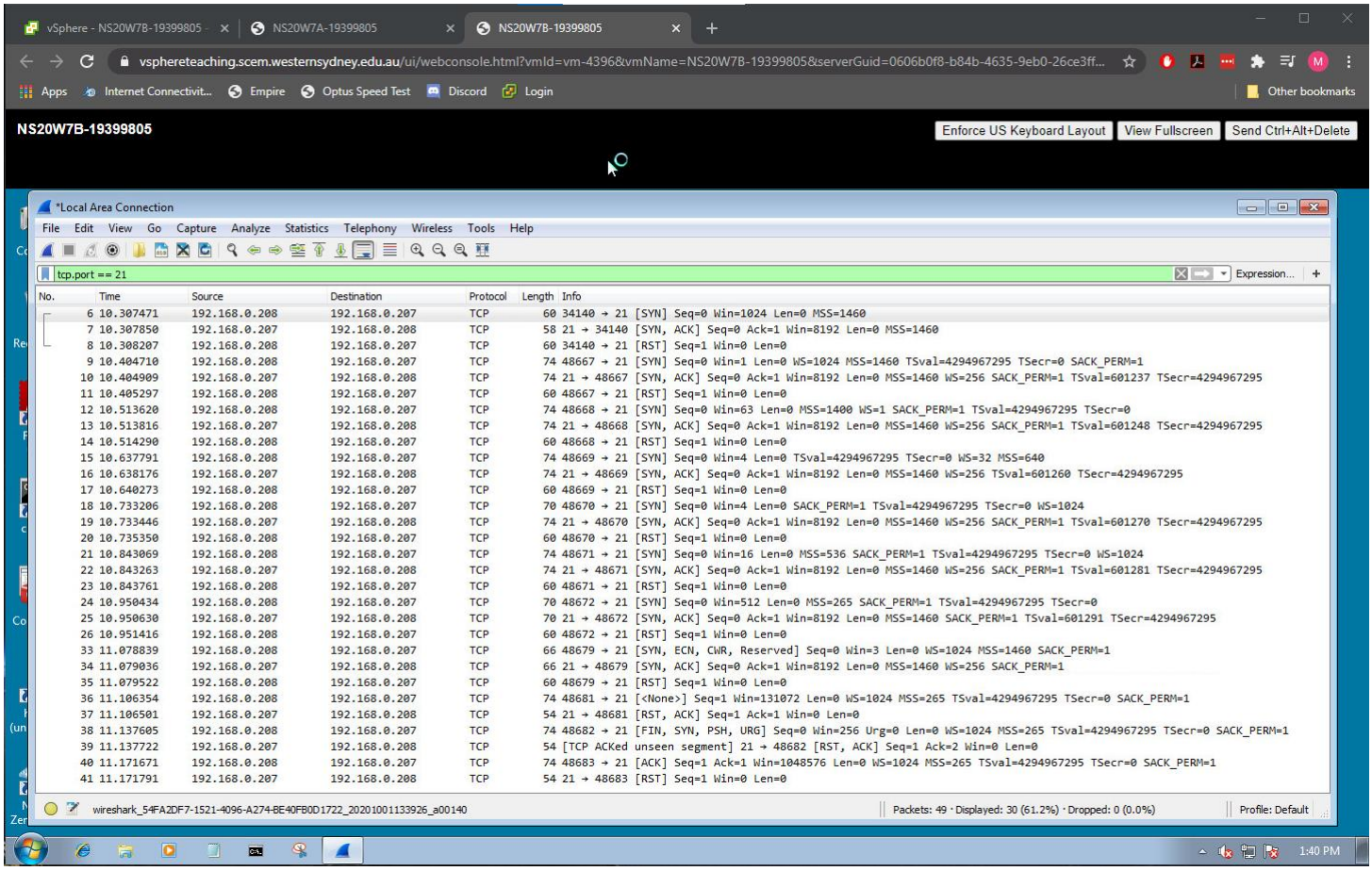

Figure 11: OS Detection Using Port 21 Traffic

Analysis: From the screenshots it is evident that the majority of information when trying to detect an OS comes from the open port. Similar to service detection, multiple TCP connections are established, and ended, trying to gather the specific information needed by Nmap to try and match it with an entry in their OS fingerprint database. From the information Nmap

Conclusions

Looking at the traffic captured by Wireshark, it does confirm the processes outlined in the introduction to this report. To detect UDP ports Nmap will send empty UDP packets to a port and await a response. If one is received then the port is closed and if no response is received then the port is listed as open|filtered because it is still possible that the packet was dropped before reaching the destination. On specific UDP ports, a custom payload can be used to give a more definitive answer as to whether the port is open or not. For service detection, Nmap probes the target port using multiple TCP connections to retrieve information about the service. It then tries to match the information collected to a database that has records of how each version reacts to a TCP probe. Finally, OS detection works in a similar way to service detection in that Nmap will probe the target port with TCP connections and try to match response and information to a database entry.

Overall, the findings recorded in the practical exercise do support the information found in the literature review.

References

Avast. (n.d.). What is port scanning and how does it work? Retrieved October 6, 2020, from https://www.avast.com/en-au/business/resources/what-is-port-scanning

Borges, E. (2019, December 5). SecurityTrails: What are Open Ports? Retrieved October 6, 2020, from https://securitytrails.com/blog/open-ports

Lyon, G. (2020). Nmap. Retrieved October 6, 2020, from https://nmap.org/

Lyon, G. (2020a). Service and Version Detection: Nmap Network Scanning. Retrieved October 6, 2020, from https://nmap.org/book/man-version-detection.html

Lyon, G. (2020b). OS Detection: Nmap Network Scanning. Retrieved October 6, 2020, from https://nmap.org/book/man-os-detection.html

Lyon, G. (2020c). UDP Scan: Nmap Network Scanning. Retrieved October 6, 2020, from https://nmap.org/book/scan-methods-udp-scan.html

Paloalto NETWORKS. (n.d.). What is a Port Scan? Retrieved October 6, 2020, from https://paloaltonetworks.com/cyberpedia/what-is-a-port-scan

Petters, J. U. (2020, March 29). 5 Basic Port Scanning Techniques. Retrieved October 6, 2020, from https://www.varonis.com/blog/port-scanning-techniques/