

ADCS Attack Paths in BloodHound — Part 2

posts.specterops.io/adcs-attack-paths-in-bloodhound-part-2-ac7f925d1547

Jonas Bülow Knudsen

May 1, 2024

Posts from SpecterOps team members on various topics relating information security

In [Part 1](#) of this series, we explained how we incorporated Active Directory Certificate Services (ADCS) objects into BloodHound and demonstrated how to effectively use BloodHound to identify attack paths including the ESC1 abuse technique.

In this blog post, we will continue to explore more of the new edges we have introduced with ADCS support in BloodHound. More specifically, we will cover how we have incorporated the [Golden Certificate](#) and the [ESC3](#) abuse technique.

I have written this blog post on behalf of the BloodHound Enterprise team at SpecterOps, which has designed and implemented the BloodHound edges described in this blog post. Thanks to everyone on the team who helped out with this effort!

Hosts and Golden Certificate

The computer hosting a certificate authority (CA) service holds the private key of its CA certificate. The key must be there for the CA to sign and issue certificates. This makes CA hosts a very lucrative target. As [Will Schroeder](#) and [Lee Chagolla-Christensen](#) described under DPERSIST1 in the ADCS whitepaper [Certified Pre-Owned](#), it is possible to craft “golden certificates” with the private key of the CA certificate, which then allows you to authenticate as anyone just as ESC1.

The AD enterprise CA object holds the DNS name of its CA host in its `dnHostName` attribute. This enables us to look up the corresponding AD computer object. To represent the relationship between the AD computer object and the enterprise CA object in BloodHound, we create a non-traversable edge named *HostsCAService*, going from the Computer node to the EnterpriseCA node.

For an attacker to craft a golden certificate that works for domain authentication, the enterprise CA’s certificate must chain up to a trusted root CA for the domain, and the NTAuth store must include the enterprise CA certificate, just as with ESC1. If these conditions are met, we create a traversable *GoldenCert* edge from the CA Computer node to the domain:

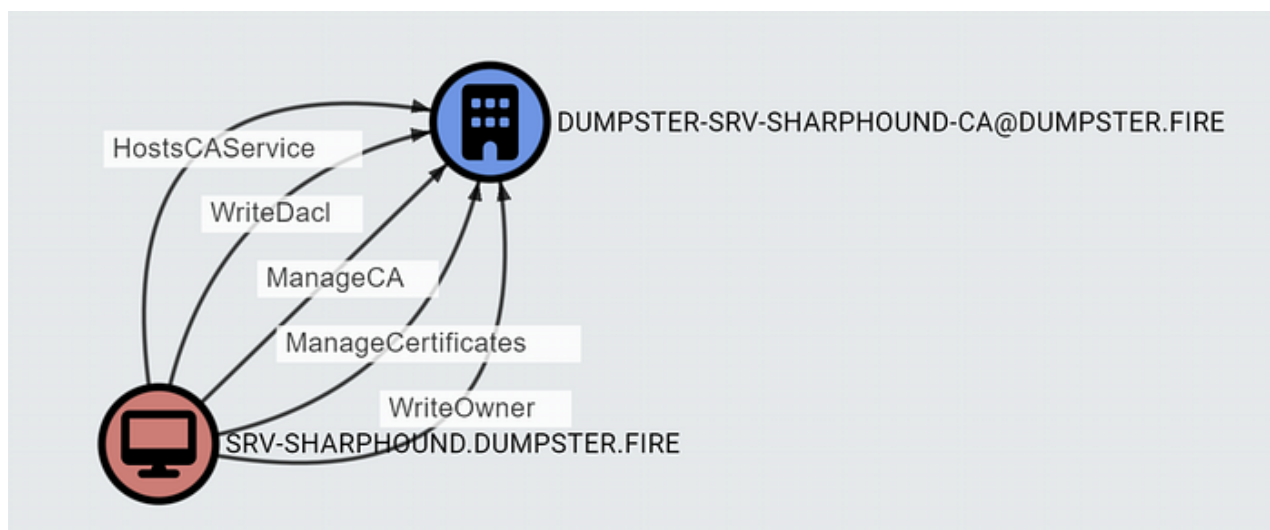
The GoldenCert edge makes it easy to identify attack paths leading to a domain compromise through a CA host.

Many organizations do not protect enterprise CA hosts as well as they should. It is a common misunderstanding that only root CAs are Tier Zero, and not issuing CAs and intermediate CAs. Both issuing CAs and intermediate CAs are enterprise CAs, and will by

default meet the requirements for the GoldenCert edge. We strongly recommend treating all CA hosts as Tier Zero.



There are exceptions to the statement in the above meme; for example, hardware protection on the CA host may prevent you from obtaining the CA private key. However, it is still possible to compromise the environment most likely, as an enterprise CA host can publish certificate templates, approve certificate requests the CA has denied, and more.



We will dive further into the edges in the above screenshot in a future blog post about ESC5 and ESC7.

There are even scenarios where an attacker can abuse a compromised CA host **not** trusted for NT authentication. An attacker may compromise users configured with an explicit certificate mapping of the type X509IssuerSubject, X509IssuerSerialNumber, X509SKI, or X509SHA1PublicKey as [Hans-Joachim Knobloch](#) called out and described in

this blog post: [Kleines Nilpferd trampelt über Microsofts PKI-Webdienste](#). The attacker can also compromise any group set up with Authentication Mechanism Assurance (AMA), as [Carl Sörqvist](#) explains in this blog post: [Forest Compromise Through AMA Abuse](#).

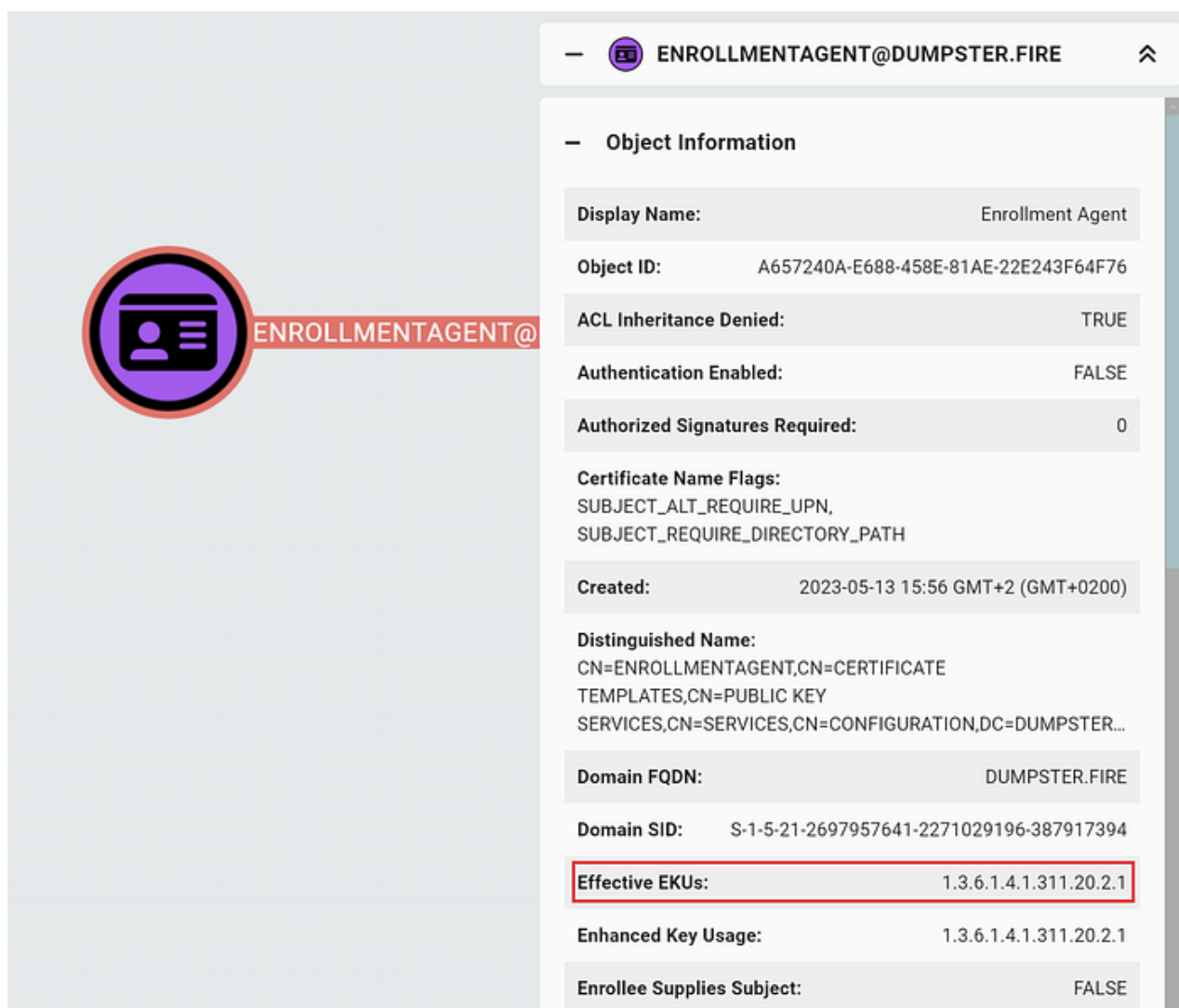
ESC3

ESC3 is similar to ESC1 in the sense that you as an attacker enroll a certificate as a targeted principal of your choice, which you then use to perform domain authentication. In ESC3, we abuse the ADCS concept of *enrollment agents*. Let us dive into the requirements.

Enrollment Agent Templates

An enrollment agent can enroll certificates on behalf of other principals. The most frequent use case for the enrollment agent concept is for an administrator who needs to issue smart cards to employees of the organization. The administrator will obtain an enrollment agent certificate and use that to enroll certificates on behalf of employees who need a smart card. This is a more secure solution than using an ESC1-type certificate template, as the enrollment agent setup enables you to restrict the targeted certificate templates and the targeted principals, as we will explore [later in this blog post](#).

To become an enrollment agent, you need an enrollment agent certificate containing the extended key usage (EKU): *Certificate Request Agent* (1.3.6.1.4.1.311.20.2.1). Such a certificate allows you to enroll on behalf of other principals in all certificate templates of schema version 1. Additionally, targeted templates of schema version 1 also allow the enroll-on-behalf-of functionality if you present a certificate with the Any Purpose EKU or no EKUs (ESC2 certificate). As explained in the [Part 1 blog post](#), you can view the effective EKUs of certificate templates on the CertTemplate node entity panel in BloodHound:



ENROLLMENTAGENT@

Object Information

Display Name:	Enrollment Agent
Object ID:	A657240A-E688-458E-81AE-22E243F64F76
ACL Inheritance Denied:	TRUE
Authentication Enabled:	FALSE
Authorized Signatures Required:	0
Certificate Name Flags:	SUBJECT_ALT_REQUIRE_UPN, SUBJECT_REQUIRE_DIRECTORY_PATH
Created:	2023-05-13 15:56 GMT+2 (GMT+0200)
Distinguished Name:	CN=ENROLLMENTAGENT,CN=CERTIFICATE TEMPLATES,CN=PUBLIC KEY SERVICES,CN=SERVICES,CN=CONFIGURATION,DC=DUMPSTER...
Domain FQDN:	DUMPSTER.FIRE
Domain SID:	S-1-5-21-2697957641-2271029196-387917394
Effective EKUs:	1.3.6.1.4.1.311.20.2.1
Enhanced Key Usage:	1.3.6.1.4.1.311.20.2.1
Enrollee Supplies Subject:	FALSE

If the targeted certificate template is of schema version 2 or above, then the targeted template must contain the Certificate Request Agent ECU in its **msPKI-RA-Application-Policies** attribute to allow enroll-on-behalf-of functionality. You can check this requirement under *Issuance Requirements* in the Windows built-in Certificate Templates Console (**certtmpl.msc**):

ESC3v2 Properties

Superseded Templates Extensions Security Server

General Compatibility Request Handling Cryptography Key Attestation

Subject Name Issuance Requirements

Require the following for enrollment:

CA certificate manager approval

✓ This number of authorized signatures: 1

If you require more than one signature, autoenrollment is not allowed.

Policy type required in signature:

Application policy

Application policy:

Certificate Request Agent

Issuance policies:

Add...

Remove

Require the following for reenrollment:

☑ Same criteria as for enrollment

☐ Valid existing certificate

Allow key based renewal (*)

Requires subject information to be provided within the certificate request.

* Control is disabled due to [compatibility settings](#).

OK Cancel Apply Help

Alternatively, you can check the *Application Policies* property in BloodHound:

ESC3V2@DUMPSTER.FIRE

Object Information

Display Name: ESC3v2

Object ID: 18EA3150-6461-4E1A-BFF3-2CF1DDC20121

ACL Inheritance Denied: TRUE

Application Policies: 1.3.6.1.4.1.311.20.2.1

Authentication Enabled: TRUE

Authorized Signatures Required: 1

Certificate Application Policies: 1.3.6.1.5.5.7.3.2

1.3.6.1.5.5.7.3.1

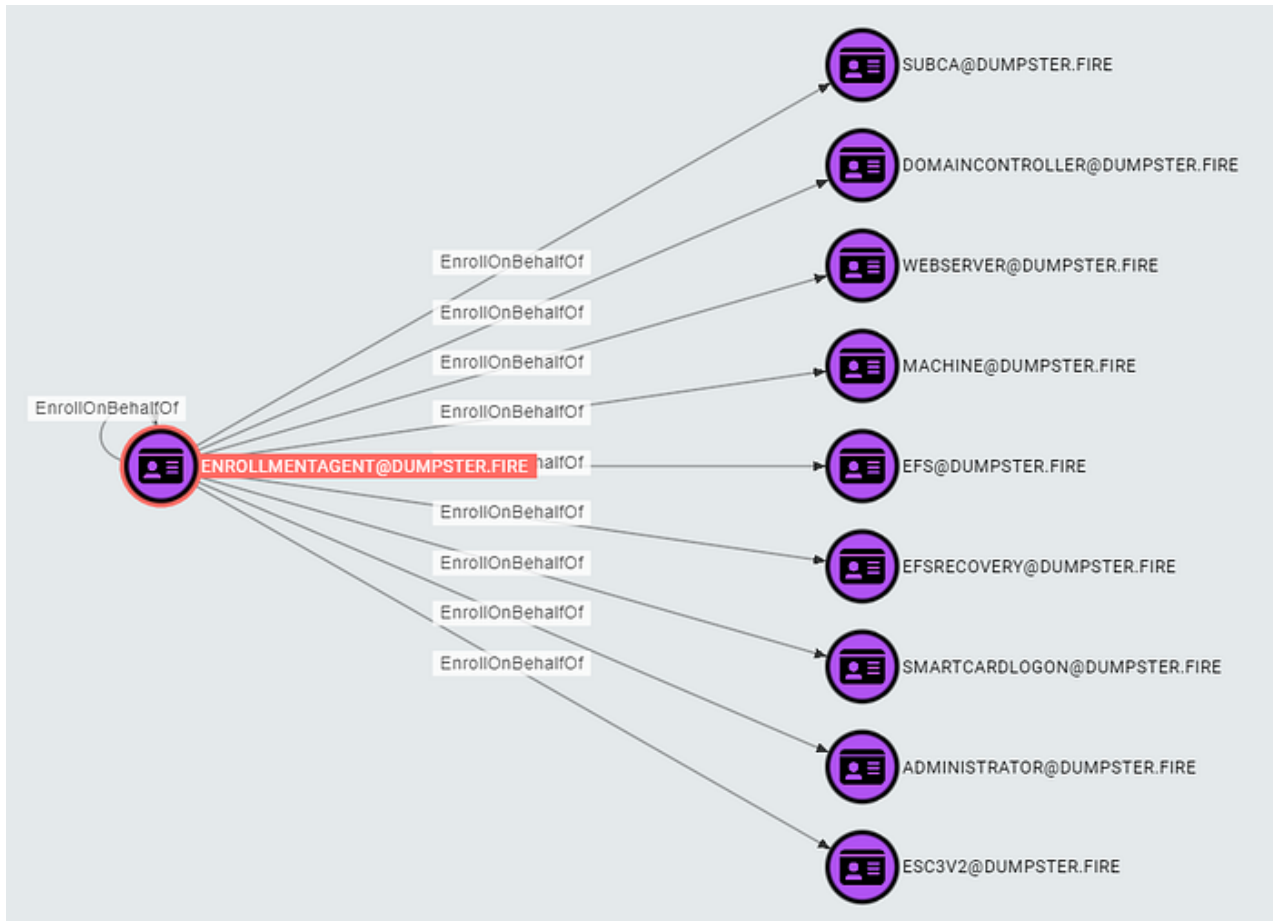
1.3.6.1.4.1.311.20.2.2

Certificate Name Flags:

SUBJECT_ALT_REQUIRE_UPN, SUBJECT_REQUIRE_EMAIL, SUBJECT_REQUIRE_DIRECTORY_PATH

At last, the enrollment agent certificate needs to chain up to a trusted root CA for the environment of the targeted certificate template.

To make it easy to find enrollment agent certificate templates and find the certificate templates that will accept the enrollment agent certificate, we have implemented a new non-traversable edge named *EnrollOnBehalfOf* to represent exactly that relationship:



To summarize, we create an *EnrollOnBehalfOf* edge between an enrollment agent certificate template and a targeted template if the following requirements are met.

If the targeted template is of schema version 1:

The enrollment agent template has one of the EKUs:- Certificate Request Agent- Any Purpose- Null (No EKUs)

If the targeted template is of schema version 2 or above:

- The enrollment agent template has the Certificate Request Agent EKU
- The targeted template contains the Certificate Request Agent EKU in its **msPKI-RA-Application-Policies** attribute.

Additionally, the enrollment agent certificate must be issued by an enterprise CA which is trusted by the NTAAuth store and it must chain up to a trusted root CA for the environment.

Requirements for Enrollment and Domain Authentication

To use (or abuse) an enrollment agent certificate template, you need enrollment rights on the enrollment agent certificate template, the targeted template, and an enterprise CA with the templates published. Note that it does not have to be the same enterprise CA that has both templates published. Both the enterprise CA for the enrollment agent template and the enterprise CA for the targeted template must chain up to a trusted root CA for the domain, and the NTAAuth store must include the enterprise CA certificate.

Issuance requirements, *manager approval* and *authorized signatures required*, can prevent enrollment on a certificate template. Both the enrollment agent certificate template and the targeted template must have manager approval disabled. The enrollment agent template must also have no authorized signatures required but the targeted template will have this set to one, as it requires the Certificate Request Agent ECU, unless it is of schema version 1 which does not support authorized signatures.

At last, the targeted template must enable domain authentication for the attacker to log in as the targeted principal.

To summarize, we add the following requirements for the ADCSESC3 edge:

- The principal has enrollment rights (potentially through group membership) on:- The enrollment agent template- The targeted template- One or more enterprise CAs where the templates are published
- The certificate chain of the enterprise CA with the enrollment agent template published is trusted
- The certificate chain of the enterprise CA with the targeted template published is trusted
- The enterprise CA of the targeted template is trusted for NT authentication
- The enrollment agent template has manager approval disabled
- The targeted template has manager approval disabled
- The enrollment agent template has no authorized signatures required
- The targeted template enables domain authentication

The [Part 1 blog.post](#) covers all of the above requirements in detail.

Subject Name and Subject Alternative Name Requirements

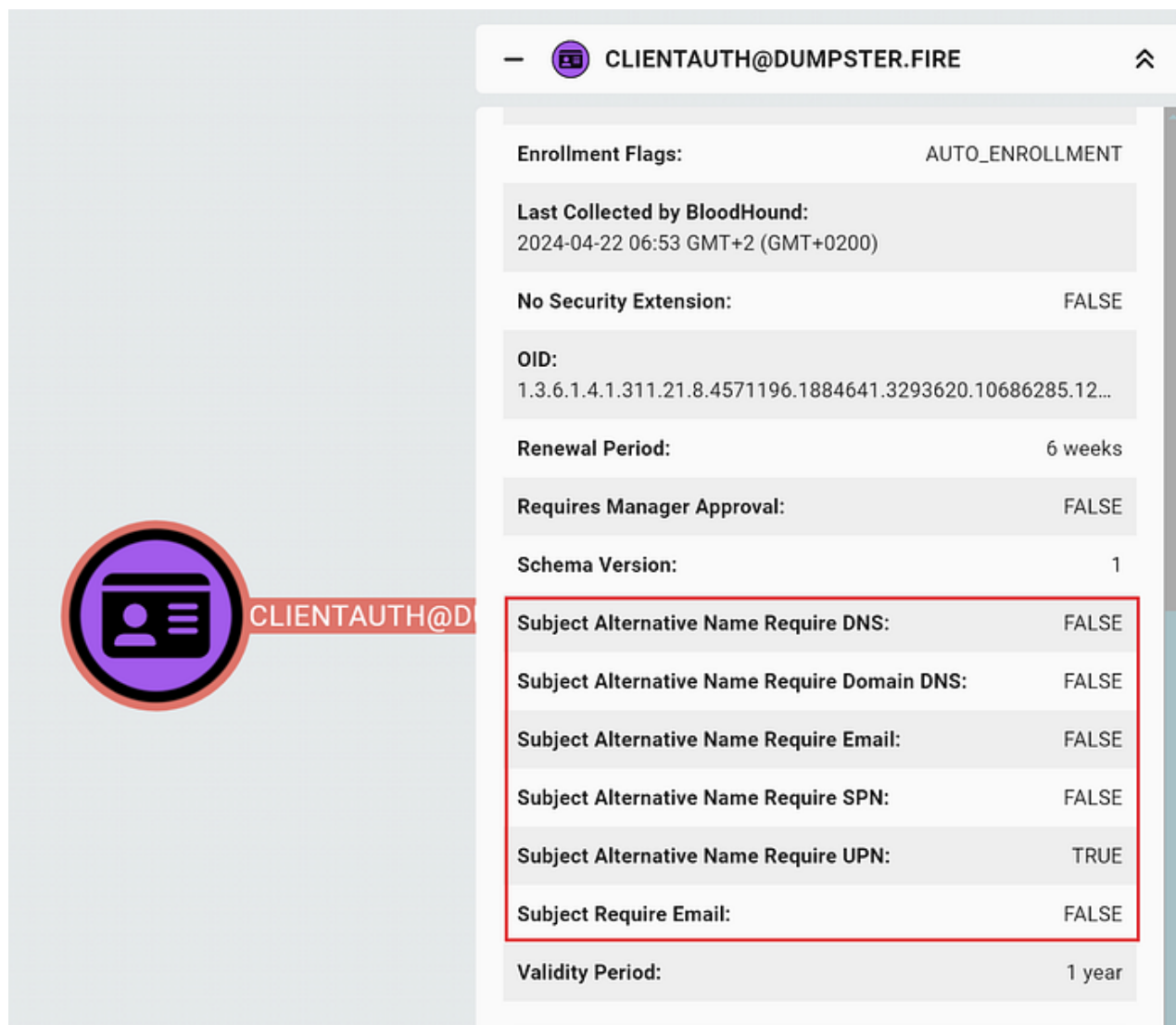
A principal must meet a certificate template's Subject Name and Subject Alternative Name (SAN) requirements to enroll in it. The certificate template has these requirements defined on the Subject Name tab of the Certificate Templates Console:

I documented my research of these requirements in the [Certificate Template Flags and Certificate Fields](#) section of the ADCS ESC14 blog post. Here are the key takeaways in terms of flags that prevent principals from enrolling:

- **CT_FLAG_SUBJECT_ALT_REQUIRE_DNS**CT_FLAG_SUBJECT_ALT_REQUIRE_DOMAIN_DNSdNSHostNameThe AD user class does not include the dNSHostName attribute, so users cannot enroll in certificate templates requiring dNSHostName. Computers will get their dNSHostName attribute set when you domain-join a computer, but the attribute is null if you simply create a computer object in AD. Computers have to their dNSHostName attribute meaning they can add a DNS name matching their computer name.
- **CT_FLAG_SUBJECT_ALT_REQUIRE_EMAIL**CT_FLAG_SUBJECT_REQUIRE_EMAILmailUsers and computers do not have their mail attribute set by default, and they cannot write to the attribute themselves. It is common to see users with the mail attribute set, but rare for computers.

When performing the enroll-on-behalf-of enrollment in a given target certificate template, it is then the target principal's attribute that goes into the certificate. Therefore, it is the target principal that must meet the target template's requirements for DNS and email and not the enrollment agent principal.

You can check what the certificate requires for Subject Name and SAN in the CertTemplate entity panel:



The screenshot shows the BloodHound interface for the CertTemplate entity panel. The entity is CLIENTAUTH@DUMPSTER.FIRE. The panel displays various enrollment flags and requirements. A red box highlights the Subject Alternative Name requirements.

Field	Value
Enrollment Flags:	AUTO_ENROLLMENT
Last Collected by BloodHound:	2024-04-22 06:53 GMT+2 (GMT+0200)
No Security Extension:	FALSE
OID:	1.3.6.1.4.1.311.21.8.4571196.1884641.3293620.10686285.12...
Renewal Period:	6 weeks
Requires Manager Approval:	FALSE
Schema Version:	1
Subject Alternative Name Require DNS:	FALSE
Subject Alternative Name Require Domain DNS:	FALSE
Subject Alternative Name Require Email:	FALSE
Subject Alternative Name Require SPN:	FALSE
Subject Alternative Name Require UPN:	TRUE
Subject Require Email:	FALSE
Validity Period:	1 year

In case you want to know what an entity panel field corresponds to in AD, or what the BloodHound database name is (for the Cypher-ninjas), then you can look up the node documentation at <https://support.bloodhoundenterprise.io>:

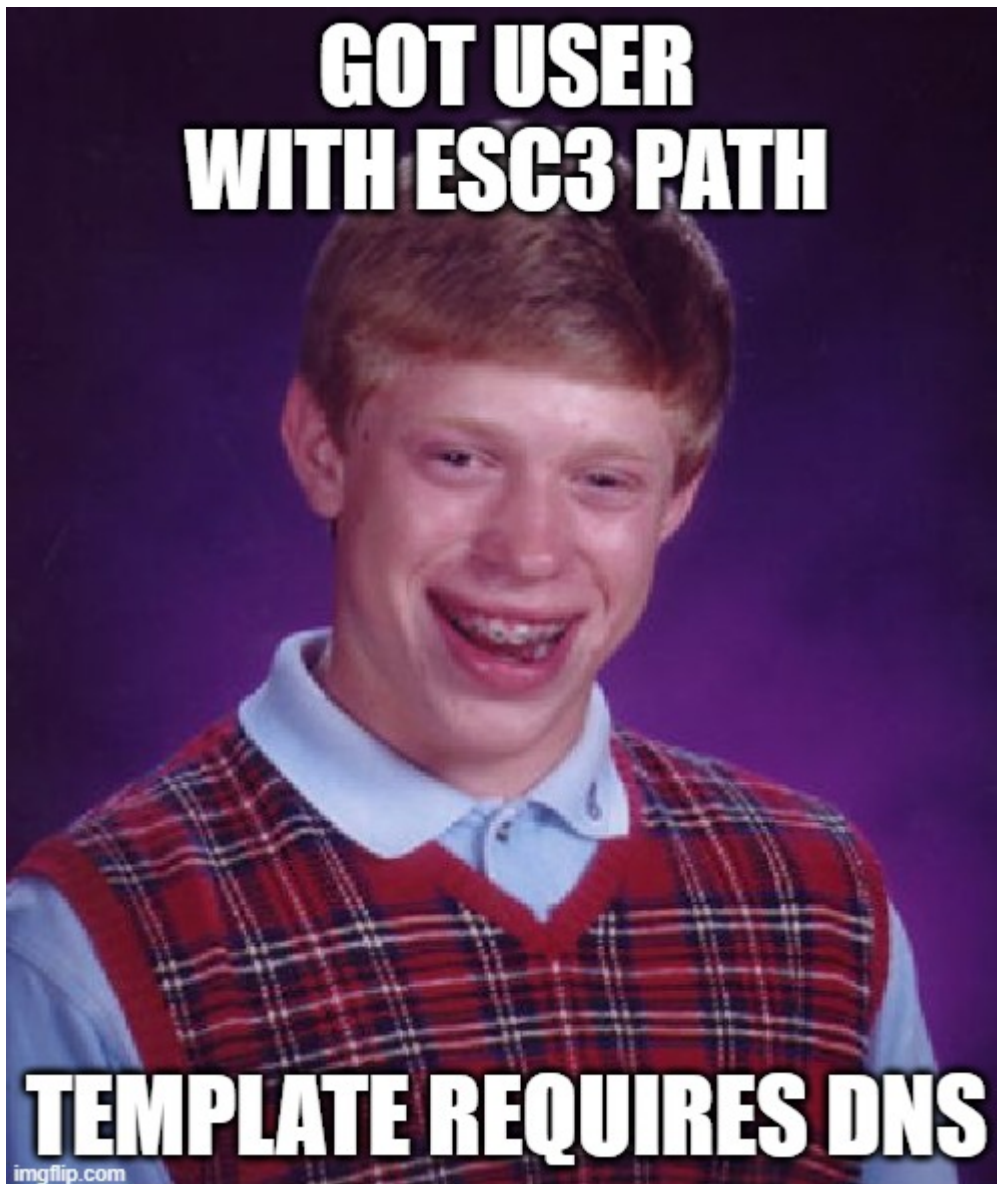
Node properties

The node supports the properties of the table. Three types of property names will be used, depending on where the property is found:

- **Entity Panel:** Name shown in the BloodHound UI.
- **Database:** Name stored in the BloodHound database and returned by the BloodHound API. This is to be used when running Cypher queries.
- **Directory:** Name collected from the directory the node is stored in, for example, the LDAP name for an Active Directory property.

Entity Panel	Database	Directory	Description
Display Name	displayname	displayName	The display name of the object.
Object ID	objectid	objectGUID	The object's unique identifier in the directory.
ACL Inheritance Denied	isaclprotected	nTSecurityDescriptor	Whether inherited permissions (ACEs) from containers are blocked on this object.
Application Policies	applicationpolicies	msPKI-RA-Application-Policies	The required RA application policy EKU in the counter signatures of certificate requests.
Authentication Enabled	authenticationenabled	-	Whether the certificate can be used for authentication. See this blog post for more

We want to avoid creating false positive BloodHound edges that make you feel like Bad Luck Brian here:



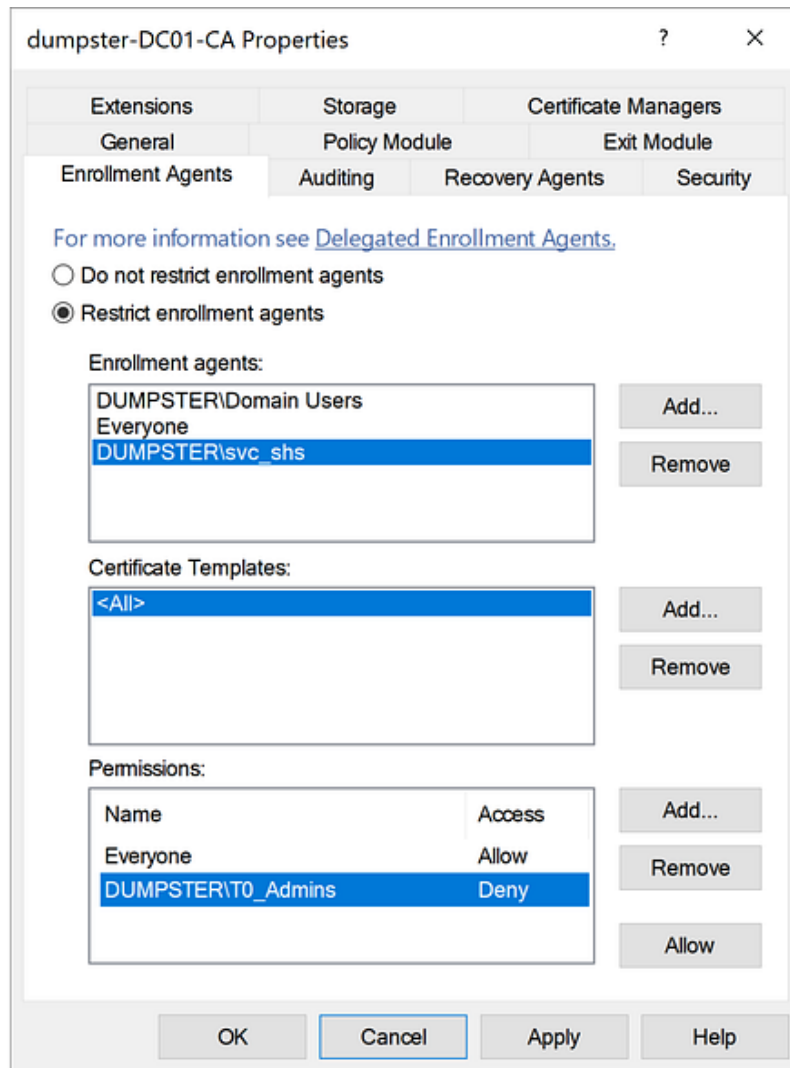
If a user meets all the requirements for an ESC3 abuse, but the enrollment agent certificate template requires DNS, then we can say with certainty that the user cannot execute the abuse. In other scenarios, it depends on your level of control over the principal as an attacker. For example, a template that requires the `mail` attribute set prevents the abuse for principals without `mail`, but an attacker with write access to the `mail` attribute of the principal can easily circumvent that.

To summarize, we add the following ADCSESC3 requirement:

If the attacker principal is a user:- The enrollment agent certificate template does not require DNS

Enrollment Agent Restrictions

You can configure super granular enrollment agent restrictions per each enterprise CA. You can specify exactly what principals the CA should allow enrolling as enrollment agents in what certificate templates and on behalf of which targeted principals:



This is super powerful from a defensive perspective, but challenging to model with a graph as each rule potentially involves more than two nodes. What we have done, though, is create a non-traversable edge named *DelegatedEnrollmentAgent* from the enrollment agent principals to the certificate templates specified in the restrictions, if it is an allow-rule.

The CA host stores the enrollment agent restrictions in registry. You can see whether SharpHound collected the enrollment agent restrictions and whether the CA has any in the EnterpriseCA node entity panel:



ESC3-ESC3-DC-CA@ESC3.LOCAL

Certificate Thumbprint:
485A76A37381CB8876E3DE33DEC228AC360489C4

Created: 2022-09-29 06:37 GMT+2 (GMT+0200)

Distinguished Name:
CN=ESC3-ESC3-DC-CA,CN=ENROLLMENT SERVICES,CN=PUBLIC KEY SERVICES,CN=SERVICES,CN=CONFIGURATION,DC=ESC3,DC=L...

DNS Hostname: ESC3-DC.ESC3.LOCAL

Domain FQDN: ESC3.LOCAL

Domain SID: S-1-5-21-3110909217-2995314319-4096102696

Enrollment Agent Restrictions Collected: TRUE

Flags:
SUPPORTS_NT_AUTHENTICATION,
CA_SERVERTYPE_ADVANCED

Has Basic Constraints: FALSE

Has Enrollment Agent Restrictions: TRUE

Is User Specifies San Enabled Collected: TRUE

Is User Specifies San Enabled: FALSE

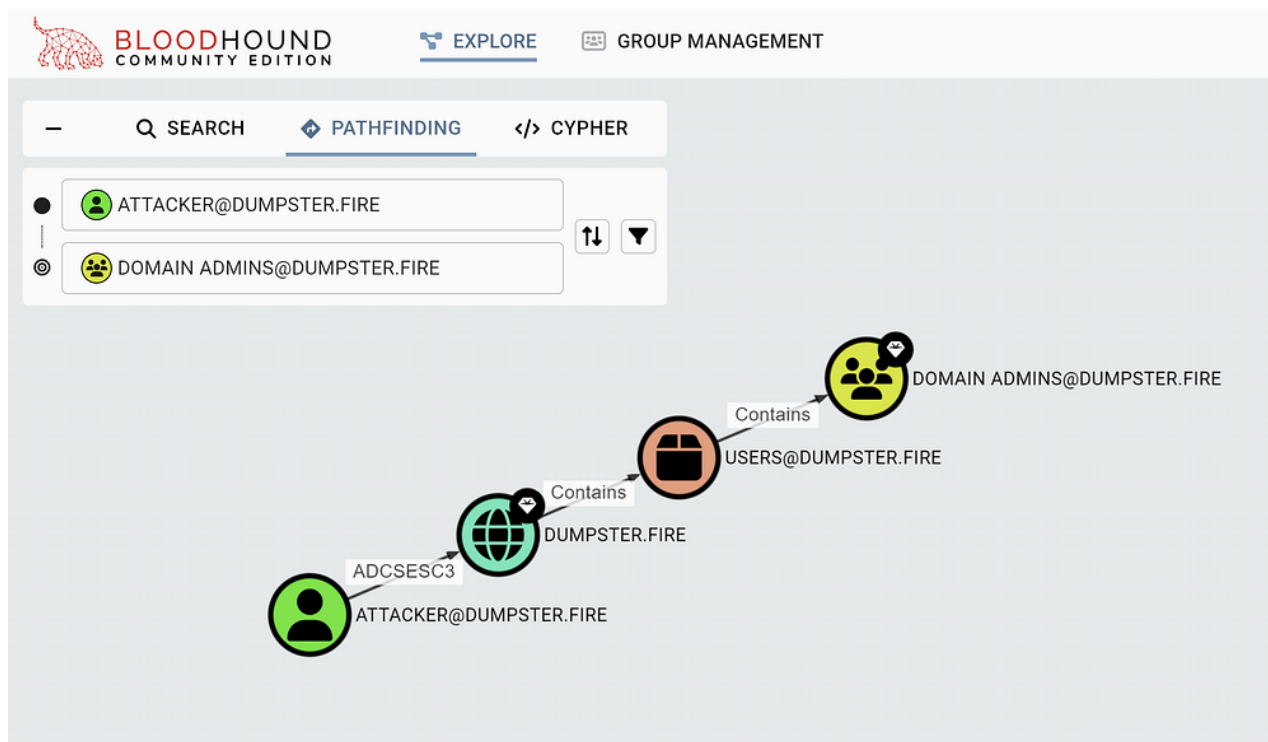
Last Collected by BloodHound:
2024-04-22 18:41 GMT+2 (GMT+0200)

We add a final requirement for the ADCSESC3 edge:

If the enterprise CA of the targeted certificate template has enrollment agent restrictions:- The principal has a DelegatedEnrollmentAgent edge to the targeted certificate template (potentially through group membership)

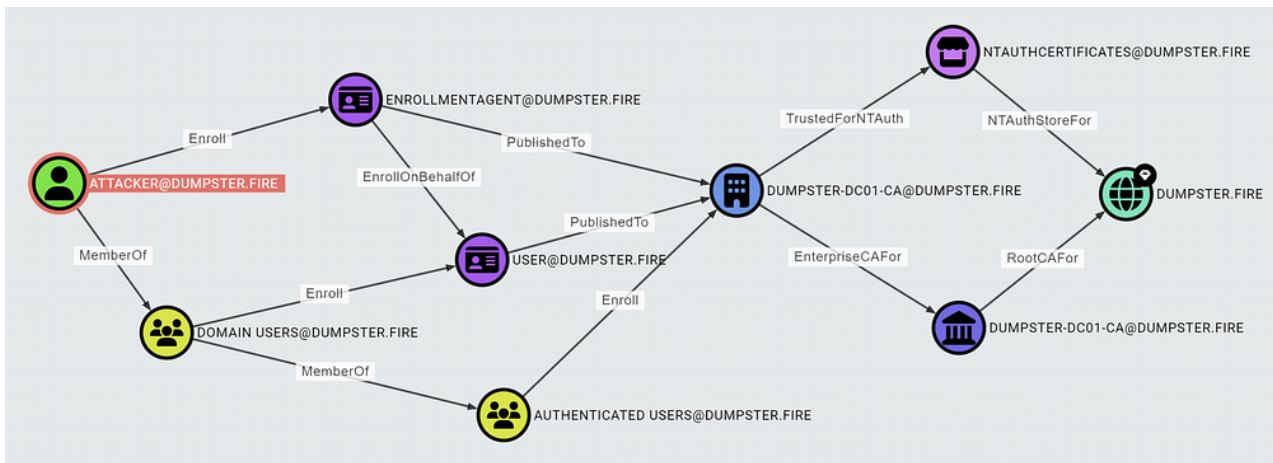
The ADCSESC3 Edge

For principals that meet all the requirements above and have the permissions required to perform an ESC3 abuse, BloodHound creates a traversable *ADCSESC3* edge to the forest root domain, similar to the *ADCSESC1* edge. So instead of checking all the requirements manually, you can easily identify attack paths that include the ESC3 abuse:



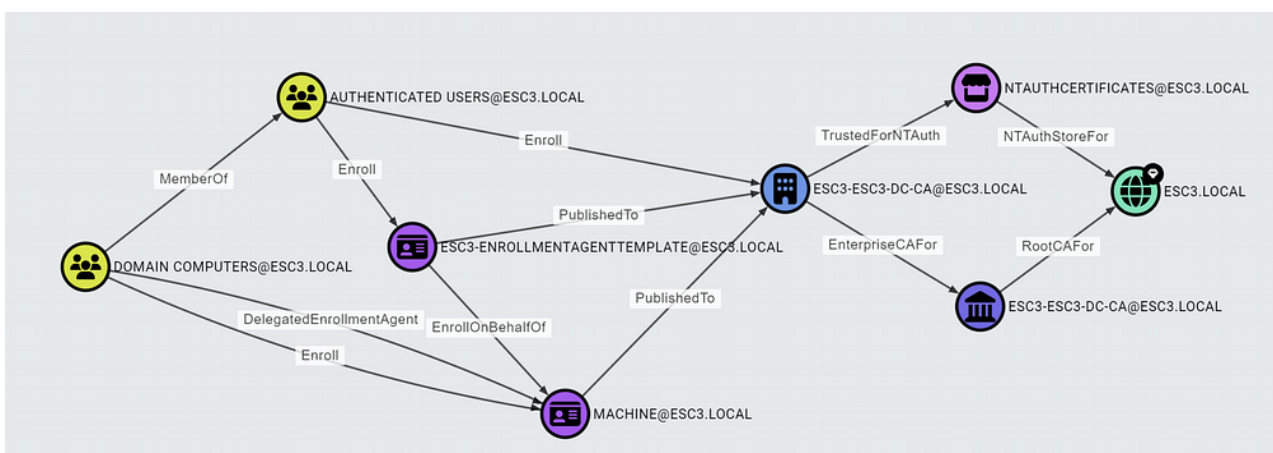
As with all other edges, you can click on it to view the related entity panel and learn more about the edge; including how to abuse it:

Clicking on “Composition” in the edge entity panel reveals the composition graph with the nodes and edges the ADCSESC3 edge is based on:



The graph shows you how the principal meets the requirements for the ESC3 abuse. The graph in the above screenshot is a simple example that only contains a single EnterpriseCA node and two CertTemplates. You may encounter graphs with many more nodes if the principal meets the requirement through several certificate templates and enterprise CAs.

The graph may also include a DelegatedEnrollmentAgent edge:

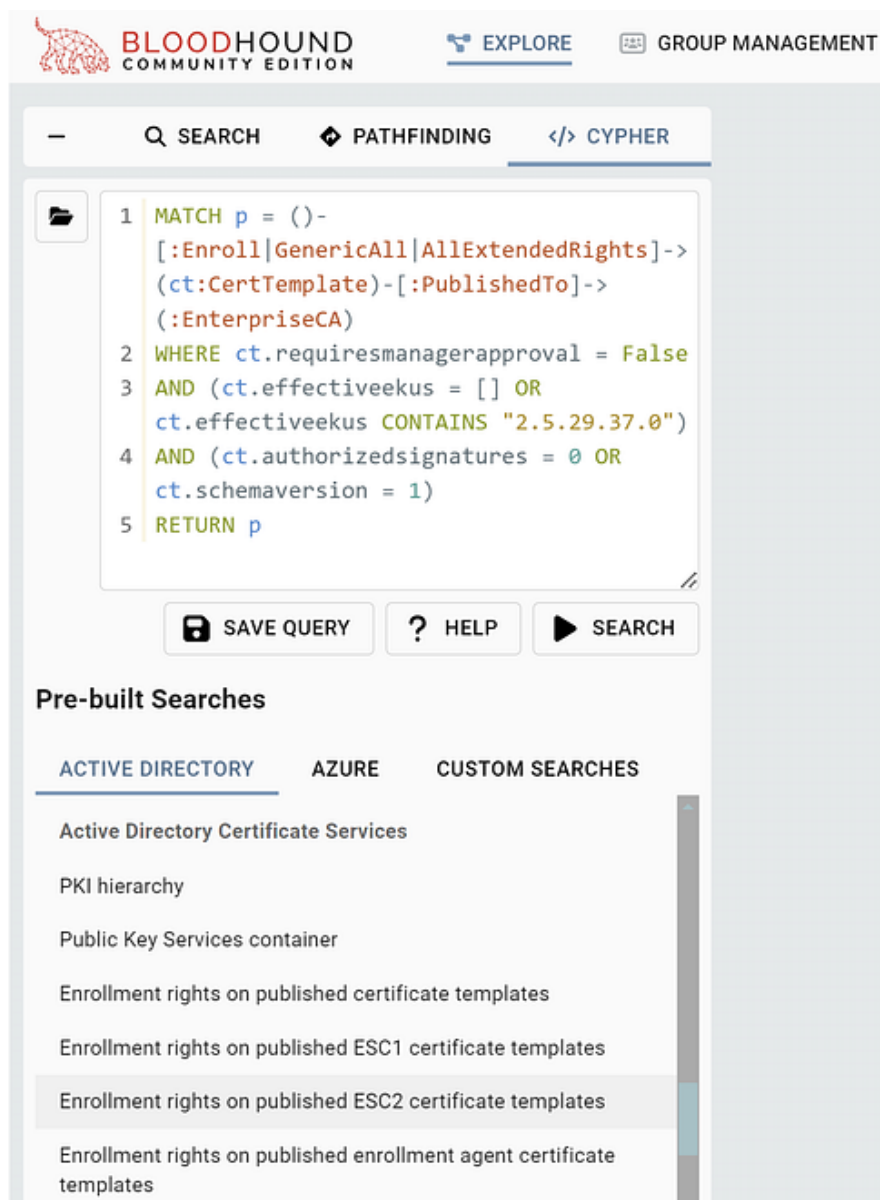


If you see one of these DelegatedEnrollmentAgent edges in the composition graph, check the scope of targeted principals in the enrollment agent restrictions and confirm that there is no deny rule overruling the permission.

No ESC2 in BloodHound?

ESC2 is where you enroll a certificate with the Any Purpose EKU or no EKUs (a.k.a., sub-CA certificate). The Any Purpose EKU means you can use the certificate for any purpose, but it does not enable impersonation on its own. A sub-CA certificate enables you to create certificates of any kind including certificates as other principals. However, you cannot perform domain authentication using these certificates, as the NTAuth store does not include the sub-CA certificate automatically. That leaves us with no end node we can draw a potential ADCSESC2 edge to.

ESC2 certificates are still powerful, though, and may enable an attacker to perform an attack outside of the scope of BloodHound. You can use this pre-built Cypher query to find principals with enrollment rights on ESC2 certificate templates:



The screenshot shows the BloodHound Community Edition web interface. At the top, there's a navigation bar with the BloodHound logo, 'COMMUNITY EDITION', and links for 'EXPLORE' and 'GROUP MANAGEMENT'. Below this is a search bar and tabs for 'SEARCH', 'PATHFINDING', and 'CYPHER'. The 'CYPHER' tab is active, displaying a Cypher query in a text editor. The query is as follows:

```
1 MATCH p = ()-
  [:Enroll|GenericAll|AllExtendedRights]->
  (ct:CertTemplate)-[:PublishedTo]->
  (:EnterpriseCA)
2 WHERE ct.requiresmanagerapproval = False
3 AND (ct.effectiveekus = [] OR
  ct.effectiveekus CONTAINS "2.5.29.37.0")
4 AND (ct.authorizedsignatures = 0 OR
  ct.schemaversion = 1)
5 RETURN p
```

Below the query editor are buttons for 'SAVE QUERY', '? HELP', and 'SEARCH'. Underneath is a section titled 'Pre-built Searches' with three tabs: 'ACTIVE DIRECTORY', 'AZURE', and 'CUSTOM SEARCHES'. The 'ACTIVE DIRECTORY' tab is selected, showing a list of pre-built queries. The query 'Enrollment rights on published ESC2 certificate templates' is highlighted.

We have also added a handful of other ADCS queries that you might find useful. Check them out, and feel free to submit a pull request if you feel like something is missing.

What is Next

We have now covered ESC1, Golden Certificate, and ESC3 with this blog post series. Stay tuned for future posts, as we will dive further into more advanced ADCS escalations and how you can identify them using BloodHound.

We are very eager to get your feedback. Please join us in the [BloodHound Slack](#) or report any issues on the [BloodHound GitHub repo](#).