



Analyzing and Executing ADCS Attack Paths with BloodHound

Andy Robbins / Jonas Bülow Knudsen

SpecterOps



Andy Robbins

- Principal Product Architect @ SpecterOps
- Co-creator of BloodHound



Jonas Bülow Knudsen

- Product Architect @ SpecterOps



Agenda

- What is ADCS
- ADCS Components in BloodHound
- Demo Time!
- ADCS in BloodHound Enterprise



Acknowledgements

- Oliver Lyak – Offensive Expert @ Institute for Cyber Risk
- Jean Marsault – Manager @ Wavestone
- Benjamin Delpy – Creator of Mimikatz, Chef de Service d'ARCOS @ Banque de France
- Christoph Falta
- Maciej Kosz - IT Security Officer @ Vattenfalland
- Mike Jankowski-Lorek – Cyber Security Architect @ CQUIRE
- Elke Stangl – Engineer @ punktwissen Proyer & Stangl OG
- Carl Sörqvist - Senior Consultant @ Bitoba
- Ceri Coburn – Red Team Operator & Offensive Security Dev @ Pen Test Partners



Acknowledgements

- Brad Hill – Software Engineer @ Meta
- Keyfactor Technical Team
- Mark Gamache – Principal Cryptography Engineer @ Salesforce
- Daniel Scheidt – Pentester @ Vorwerk Gruppe
- Vadims Podāns - PKI Consultant @ PKI Solutions Inc.
- Andrea Pierini – Senior Incident Response Consultant @ Semperis
- Charlie Clark – Senior Security Consultant @ MDSec
- Will Schroeder – Researcher @ SpecterOps
- Lee Christensen – Researcher @ SpecterOps
- BloodHound Enterprise Team @ SpecterOps

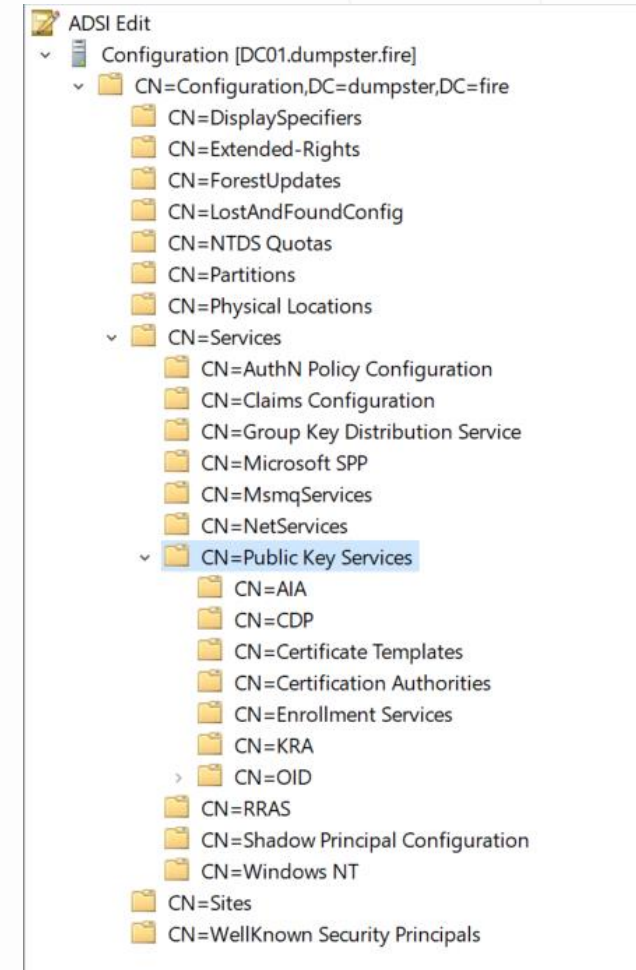


What is ADCS

Active Directory Certificate Services (ADCS)

What is ADCS

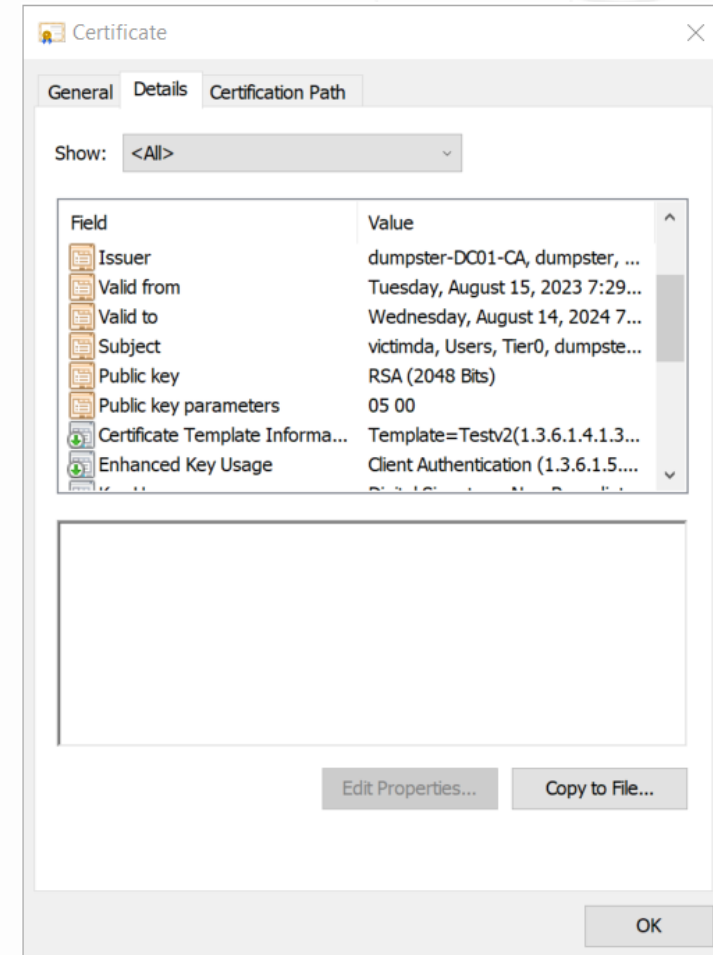
- Provides scalable Public Key Infrastructure (PKI)
- Used for issuing and managing digital certificates
- Located in the Public Key Services container



Digital certificate

What is ADCS

- Asymmetric cryptography (public and private key pair)
- Bound to a “Subject”
- Used for encryption, signing, and authentication
- Holds a certificate chain



ADCS components – RootCA

What is ADCS



- Root Certificate Authority
- Self-signed certificate (no issuer)
- Trusted by all computers in the forest
- Issues Enterprise CA certificates

The screenshot displays two windows from a Windows environment. The top window is 'ADSI Edit', showing the 'Configuration [DC01.dumpster.fire]' tree. Under 'CN=Services', 'CN=Public Key Services' is expanded, and 'CN=Certificate Authorities' is selected. The right pane shows a list of certificate authorities:

| Name | Class |
|---------------------|------------------------|
| CN=MYFAKECA3 | certificationAuthority |
| CN=MYFAKECA2 | certificationAuthority |
| CN=MYFAKECA | certificationAuthority |
| CN=dumpster-DC01-CA | certificationAuthority |

The bottom window is 'certlm - [Certificates - Local Computer\Trusted Root Certification Authorities\Certificates]'. It shows a list of certificates issued by the 'dumpster-DC01-CA' authority:

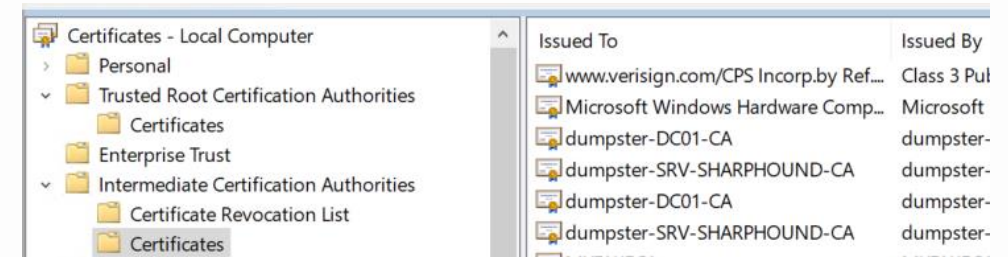
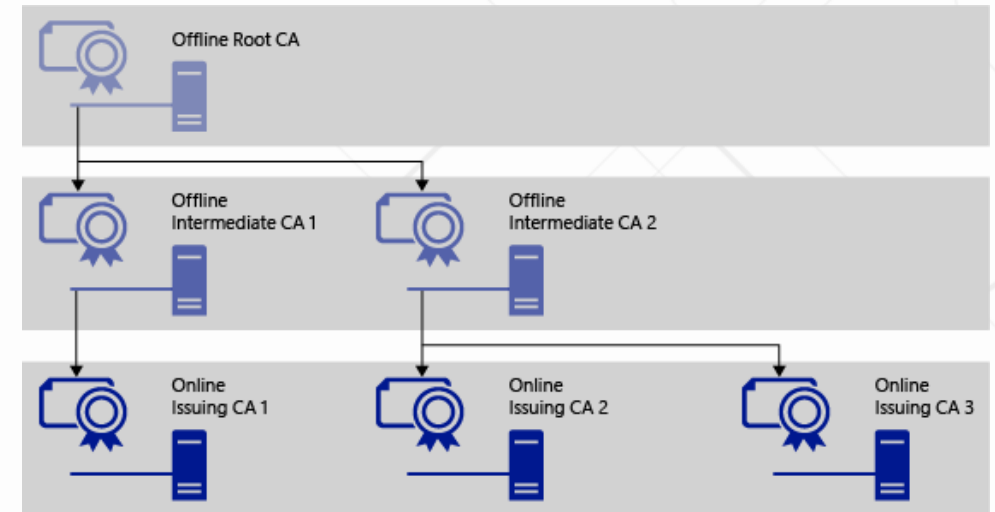
| Issued To | Issued By |
|----------------------------------|--------------------------------------|
| MYFAKECA | dumpster-DC01-CA |
| MYFAKECA2 | MYFAKECA |
| MYFAKECA3 | MYFAKECA2 |
| Symantec Enterprise Mobile Ro... | MYFAKECA3 |
| Certum Trusted Network CA | Symantec Enterprise Mobile Root f... |
| DigiCert Assured ID Root CA | Certum Trusted Network CA |
| DigiCert Global Root CA | DigiCert Assured ID Root CA |

ADCS components – EnterpriseCA

What is ADCS



- Aka enrollment service
- Certificate chains up to a RootCA
- Intermediate CAs and Issuing CAs = EnterpriseCAs
- Located in the “Enrollment Services” container
- Trusted by all computers in the forest



ADCS components – NTAAuthStore

What is ADCS



- EnterpriseCA must be trusted for *NT authentication*
- NTAAuthCertificates object (aka *NTAuth store*)
- Replicated to the local NTAuth store on DCs

The screenshot shows the ADSI Edit console with the tree view expanded to 'CN=Public Key Services'. The 'CN=NTAuthCertificates' object is selected. The right pane shows the 'Name' and 'Class' of various objects in the hierarchy. Below this, the 'CN=NTAuthCertificates Properties' dialog is open, showing the 'Security' tab. The 'Attributes' list shows 'cACertificate' with a value of '\30\82\05\58\30\82\04\40\A0\03\02\01\02\02\10\30\130\82\05\58\30\82\04\40\A0\03\02\01\02\02\13\7B\'. A 'Multi-valued Octet String Editor' dialog is also open, showing the 'cACertificate' attribute and its values, with the first value selected and highlighted.

| Name | Class |
|------------------------------|------------------------|
| CN=AIA | container |
| CN=CDP | container |
| CN=Certificate Templates | container |
| CN=Certification Authorities | container |
| CN=Enrollment Services | container |
| CN=KRA | container |
| CN=NTAuthCertificates | certificationAuthority |
| CN=OID | msPKI-Enterprise-Oid |

Multi-valued Octet String Editor

Attribute: cACertificate

Values:

- \30\82\03\71\30\82\02\59\A0\03\02\01\02\02\10\30\130\82\05\58\30\82\04\40\A0\03\02\01\02\02\13\7B\

Buttons: Add, Remove, Edit

ADCS components – CertTemplate

What is ADCS



- Used for certificate enrollment requests
- Holds characteristics of a certificate
 - Certificate usage
 - Validity period
 - And more..
- Published by EnterpriseCAs

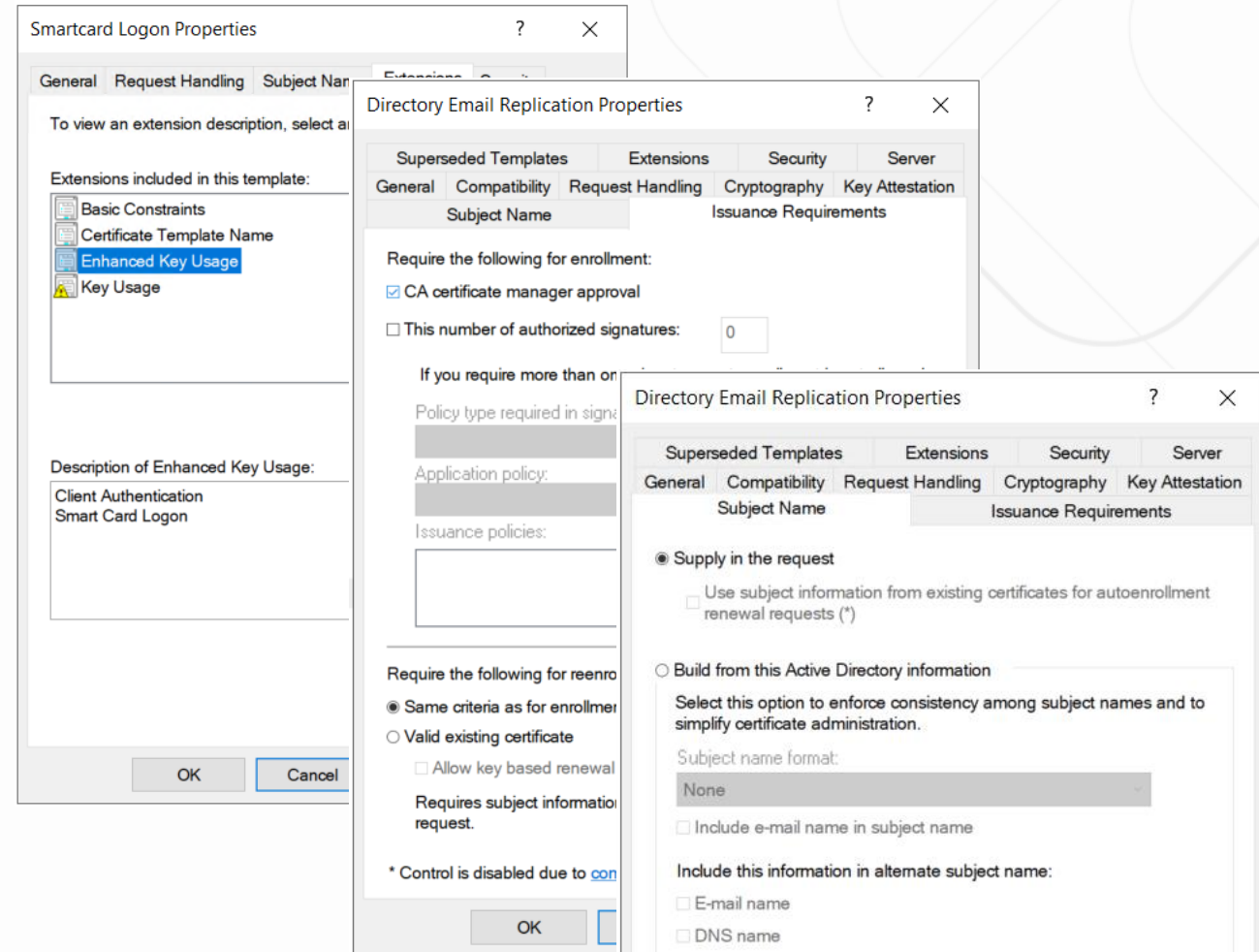
| ADSI Edit | | Name | Class |
|--------------------------------------|--|--|------------------------|
| Configuration [DC01.dumpster.fire] | | | |
| CN=Configuration,DC=dumpster,DC=fire | | | |
| CN=DisplaySpecifiers | | CN=Administrator | pKICertificateTemplate |
| CN=Extended-Rights | | CN=Authentication Mechanism Medium Le... | pKICertificateTemplate |
| CN=ForestUpdates | | CN=CA | pKICertificateTemplate |
| CN=LostAndFoundConfig | | CN=CAExchange | pKICertificateTemplate |
| CN=NTDS Quotas | | CN=CEPEncryption | pKICertificateTemplate |
| CN=Partitions | | CN=ClientAuth | pKICertificateTemplate |
| CN=Physical Locations | | CN=CodeSigning | pKICertificateTemplate |
| CN=Services | | CN=CrossCA | pKICertificateTemplate |
| CN=AuthN Policy Configuration | | CN=CTLSigning | pKICertificateTemplate |
| CN=Claims Configuration | | CN=DirectoryEmailReplication | pKICertificateTemplate |
| CN=Group Key Distribution Service | | CN=DomainController | pKICertificateTemplate |
| CN=Microsoft SPP | | CN=DomainControllerAuthentication | pKICertificateTemplate |
| CN=MsmqServices | | CN=EFS | pKICertificateTemplate |
| CN=NetServices | | CN=EFSRecovery | pKICertificateTemplate |
| CN=Public Key Services | | CN=EnrollmentAgent | pKICertificateTemplate |
| CN=AIA | | CN=EnrollmentAgentOffline | pKICertificateTemplate |
| CN=CDP | | CN=ESC1 | pKICertificateTemplate |
| CN=Certificate Templates | | CN=ESC1SmartCard | pKICertificateTemplate |
| CN=Certification Authorities | | CN=ESC3-self | pKICertificateTemplate |
| CN=Enrollment Services | | CN=ESC3v2 | pKICertificateTemplate |
| CN=KRA | | CN=ESC9 | pKICertificateTemplate |
| CN=OID | | CN=ExchangeUser | pKICertificateTemplate |
| CN=RRAS | | CN=ExchangeUserSignature | pKICertificateTemplate |
| CN=Shadow Principal Configuration | | CN=IPSECIntermediateOffline | pKICertificateTemplate |
| CN=Windows NT | | CN=IPSECIntermediateOnline | pKICertificateTemplate |
| CN=Sites | | CN=KerberosAuthentication | pKICertificateTemplate |
| CN=WellKnown Security Principals | | CN=KeyRecoveryAgent | pKICertificateTemplate |

ADCS components – CertTemplate

What is ADCS

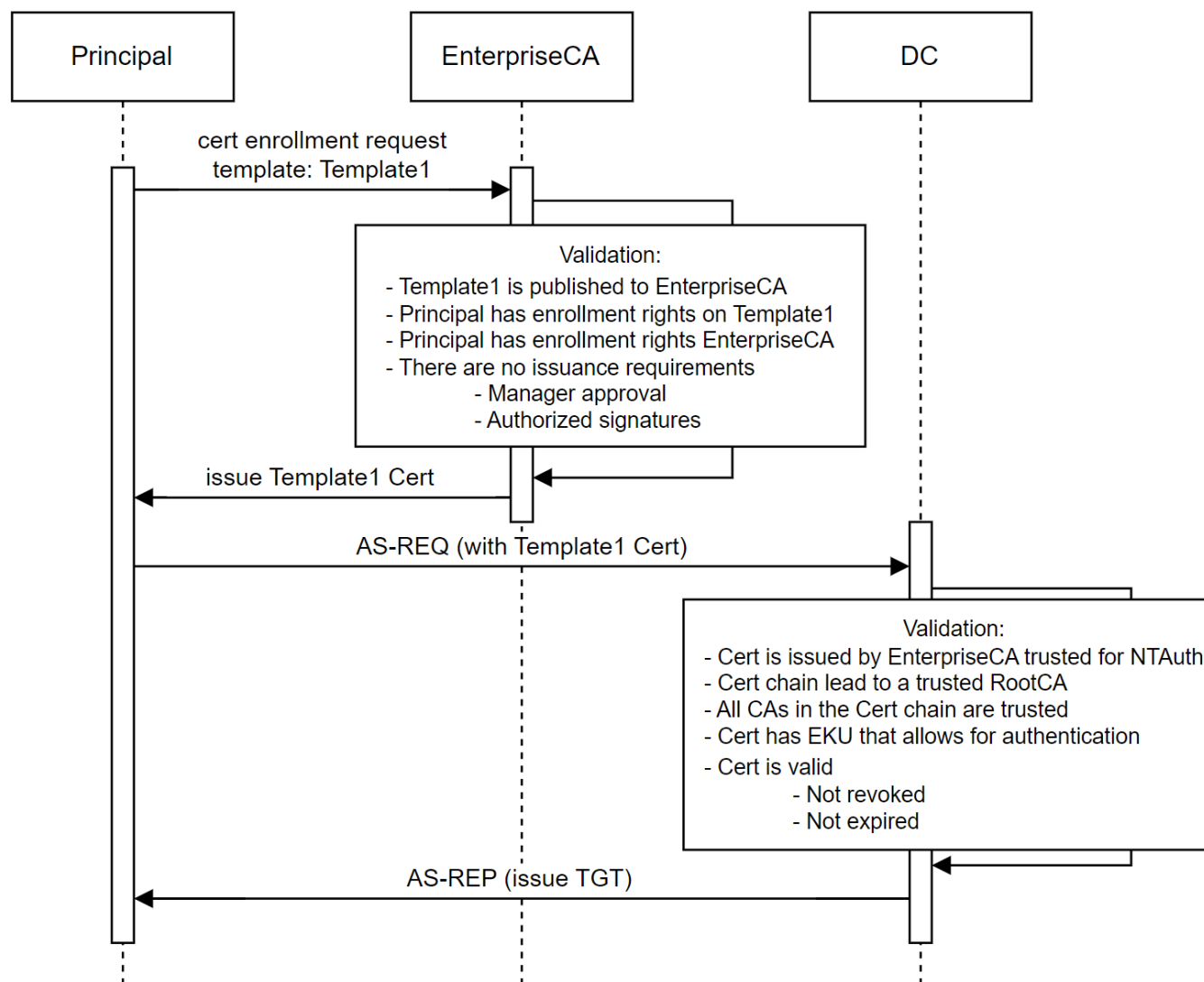


- Enhanced Key Usage (EKU)
 - Client Authentication (1.3.6.1.5.5.7.3.2)
 - PKINIT Client Authentication (1.3.6.1.5.2.3.4)
 - Smart Card Logon (1.3.6.1.4.1.311.20.2.2)
 - Any Purpose (2.5.29.37.0)
 - SubCA (no EKUs)
- Issuance requirements
 - Manager approval
 - Authorized signatures
- ENROLLEE_SUPPLIES_SUBJECT flag
 - Enroll as anyone 🔥



Enrollment and authentication process (simplified)

What is ADCS




ADCS Components in BloodHound

New node types

ADCS components in BloodHound

 AIACAs

 RootCAs

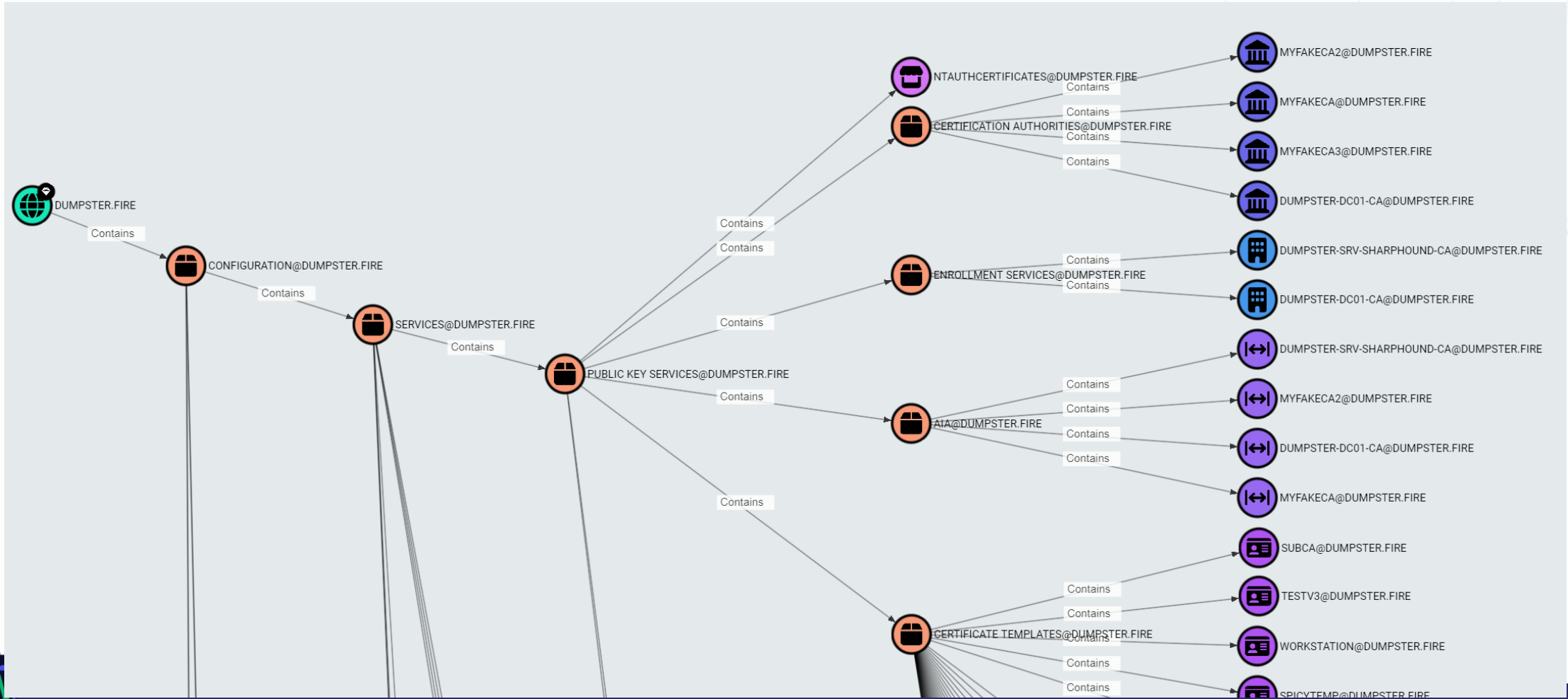
 EnterpriseCAs

 NTAuthStores

 CertTemplates

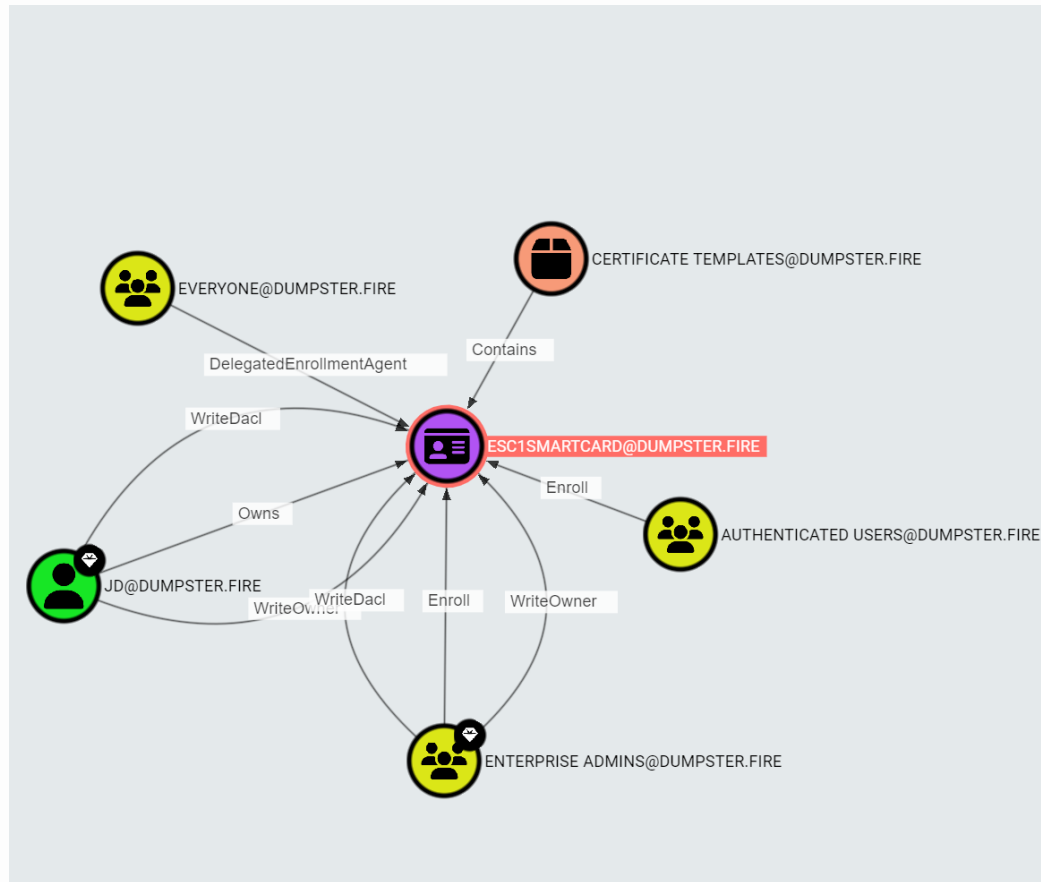
New node types

ADCS components in BloodHound



New node types

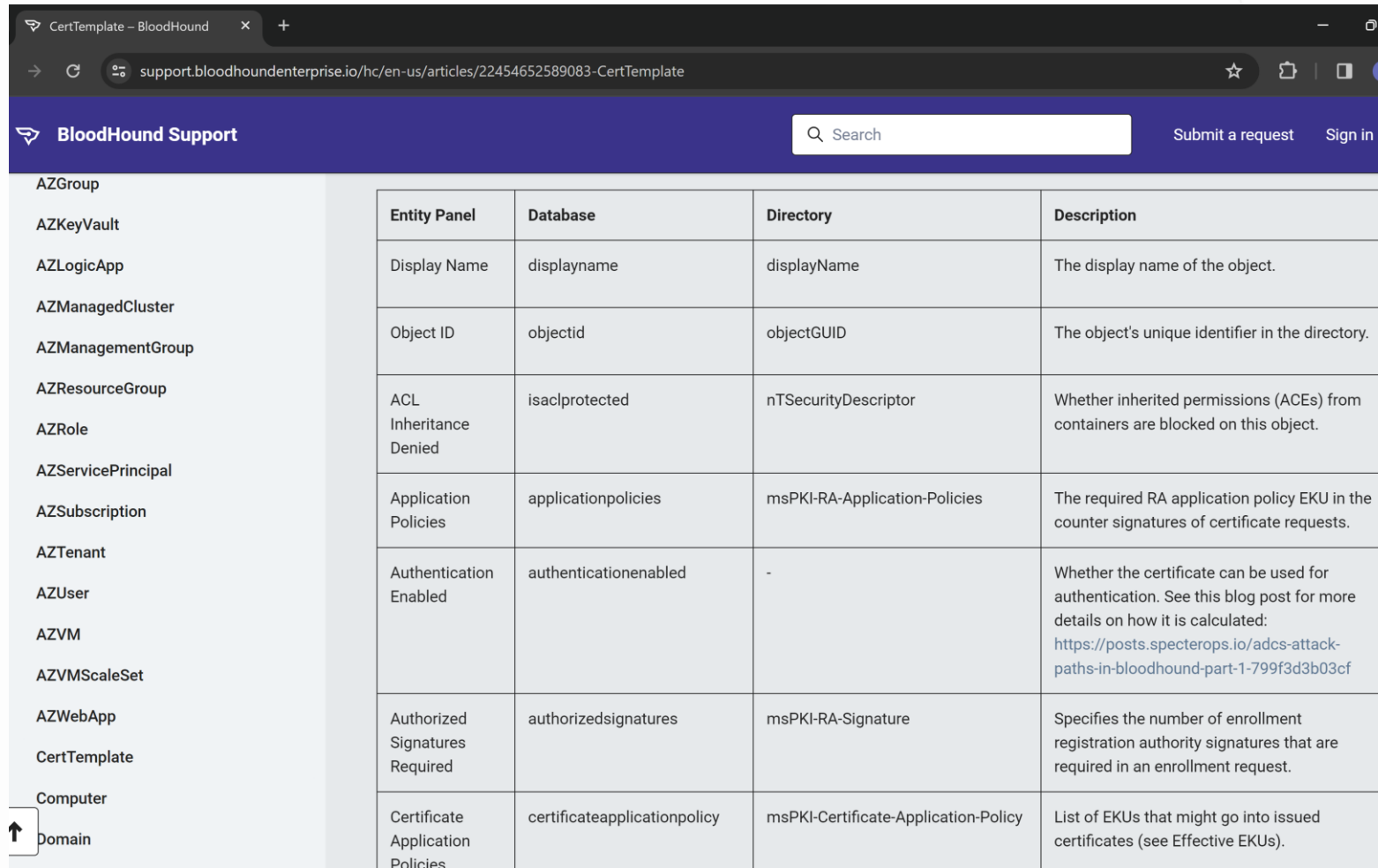
ADCS components in BloodHound



| | |
|---------------------------------------|---|
| ESC1SMARTCARD@DUMPSTER.FIRE | |
| Display Name: | ESC1SmartCard |
| Object ID: | 8482B82C-4103-43BA-B530-409A2C763965 |
| Authentication Enabled: | TRUE |
| Authorized Signatures Required: | 0 |
| Certificate Application Policies: | 1.3.6.1.4.1.311.20.2.2 |
| Certificate Name Flags: | ENROLLEE_SUPPLIES_SUBJECT, ENROLLEE_SUPPLIES_SUBJECT_ALT_NAME |
| Created: | 2023-10-19 07:19 GMT+2 (GMT+0200) |
| Distinguished Name: | CN=ESC1SMARTCARD,CN=CERTIFICATE TEMPLATES,CN=PUBLIC KEY SERVICES,CN=SERVICES,CN=CONFIGURATION,DC=DUMPSTER,DC=FIRE |
| Domain FQDN: | DUMPSTER.FIRE |
| Domain SID: | S-1-5-21-2697957641-2271029196-387917394 |
| Effective EKUs: | 1.3.6.1.4.1.311.20.2.2 |
| Enhanced Key Usage: | 1.3.6.1.4.1.311.20.2.2 |
| Enrollee Supplies Subject: | TRUE |
| Enrollment Flags: | PUBLISH_TO_DS |
| Last Collected by BloodHound: | 2023-11-27 18:06 GMT+1 (GMT+0100) |
| No Security Extension: | FALSE |
| OID: | 1.3.6.1.4.1.311.21.8.4571196.1884641.3293620.10686285.12068043.134.1116501.10660743 |
| Renewal Period: | 6 weeks |
| Requires Manager Approval: | FALSE |
| Schema Version: | 2 |
| Subject Alternative Name Require UPN: | FALSE |
| Validity Period: | 1 year |

New node types

ADCS components in BloodHound



The screenshot shows a web browser window with the URL `support.bloodhoundenterprise.io/hc/en-us/articles/22454652589083-CertTemplate`. The page header is "BloodHound Support" with a search bar and links for "Submit a request" and "Sign in". On the left, a sidebar lists various node types: AZGroup, AZKeyVault, AZLogicApp, AZManagedCluster, AZManagementGroup, AZResourceGroup, AZRole, AZServicePrincipal, AZSubscription, AZTenant, AZUser, AZVM, AZVMScaleSet, AZWebApp, CertTemplate, Computer, and Domain. The "Domain" node is selected and highlighted with an upward arrow. The main content area displays a table with four columns: Entity Panel, Database, Directory, and Description. The table lists several ADCS components and their properties.

| 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 details on how it is calculated: https://posts.specterops.io/adcs-attack-paths-in-bloodhound-part-1-799f3d3b03cf |
| Authorized Signatures Required | authorizedsignatures | msPKI-RA-Signature | Specifies the number of enrollment registration authority signatures that are required in an enrollment request. |
| Certificate Application Policies | certificateapplicationpolicy | msPKI-Certificate-Application-Policy | List of EKUs that might go into issued certificates (see Effective EKUs). |

New non-traversable edges

ADCS components in BloodHound

- RootCAFor
- EnterpriseCAFor
- NTAUTHStoreFor
- PublishedTo
- ManageCertificates
- ManageCA
- DCFor
- CanAbuseUPNCertMapping
- CanAbuseWeakCertBinding
- Enroll
- HostsCAService
- WritePKIEnrollmentFlag
- WritePKINameFlag
- IssuedSignedBy
- EnrollOnBehalfOf
- DelegatedEnrollmentAgent
- TrustedForNTAuth



What is a non-traversable edge?

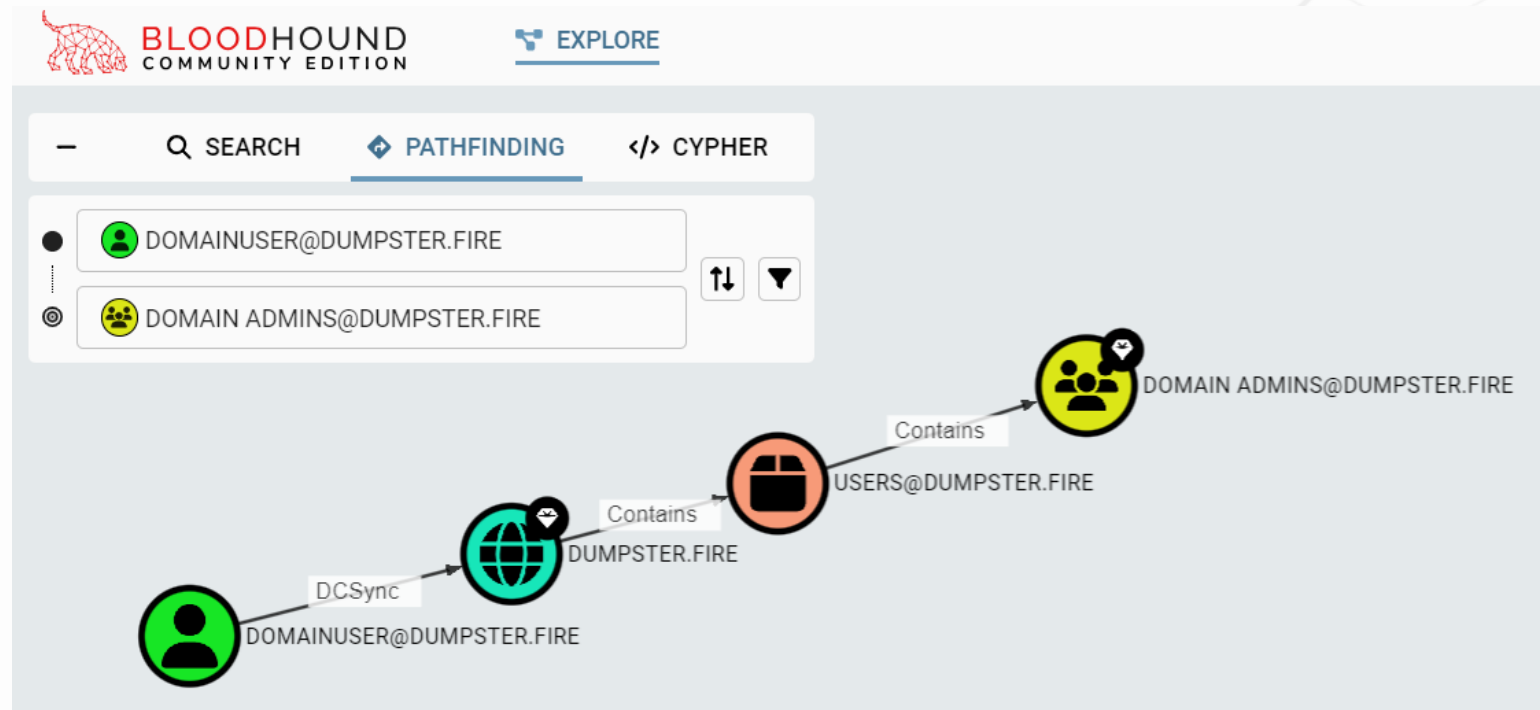
ADCS components in BloodHound

- Privileges and relationships that are not abusable on their own
- Excluded from path-finding
- Used to construct abusable (traversable) edges
- Example: GetChanges + GetChangesAll = DCSync

What is a non-traversable edge?

ADCS components in BloodHound

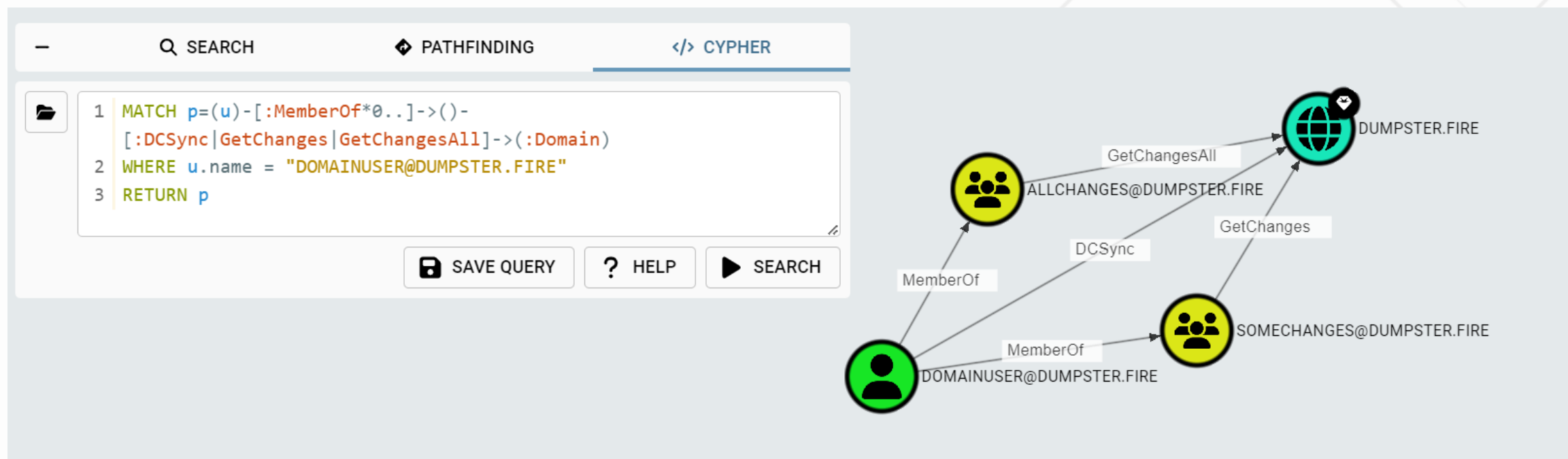
- Example: GetChanges + GetChangesAll = DCSync



What is a non-traversable edge?

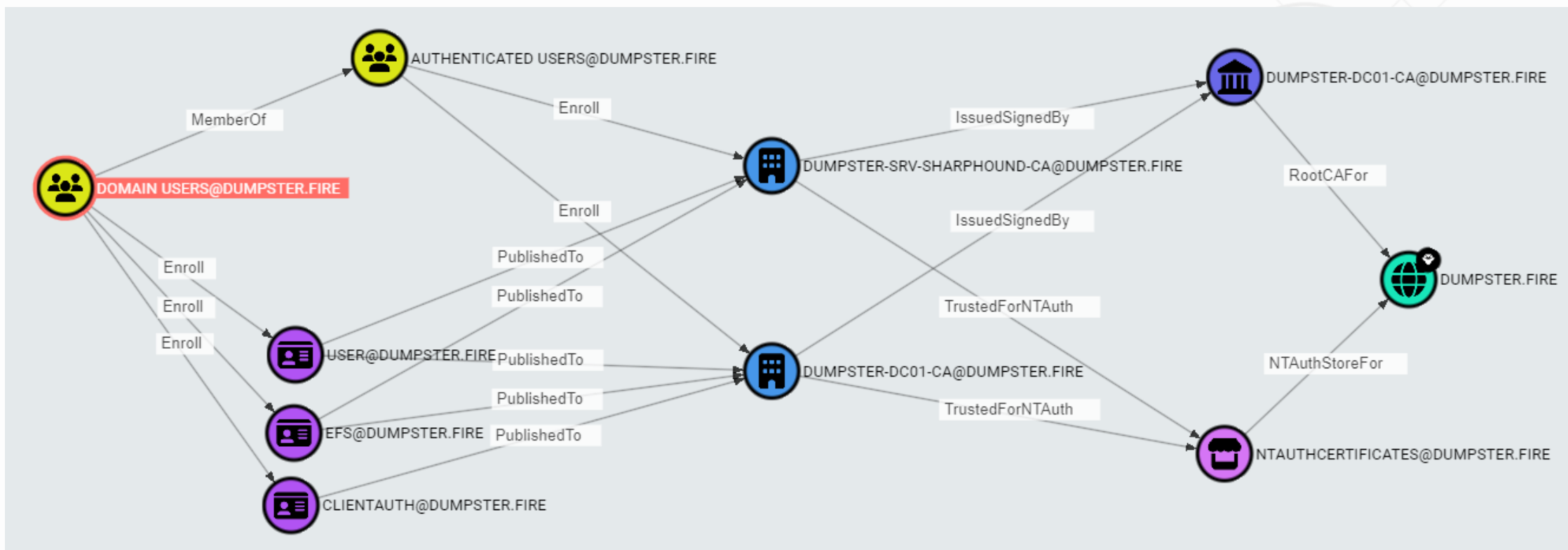
ADCS components in BloodHound

- Example: GetChanges + GetChangesAll = DCSync



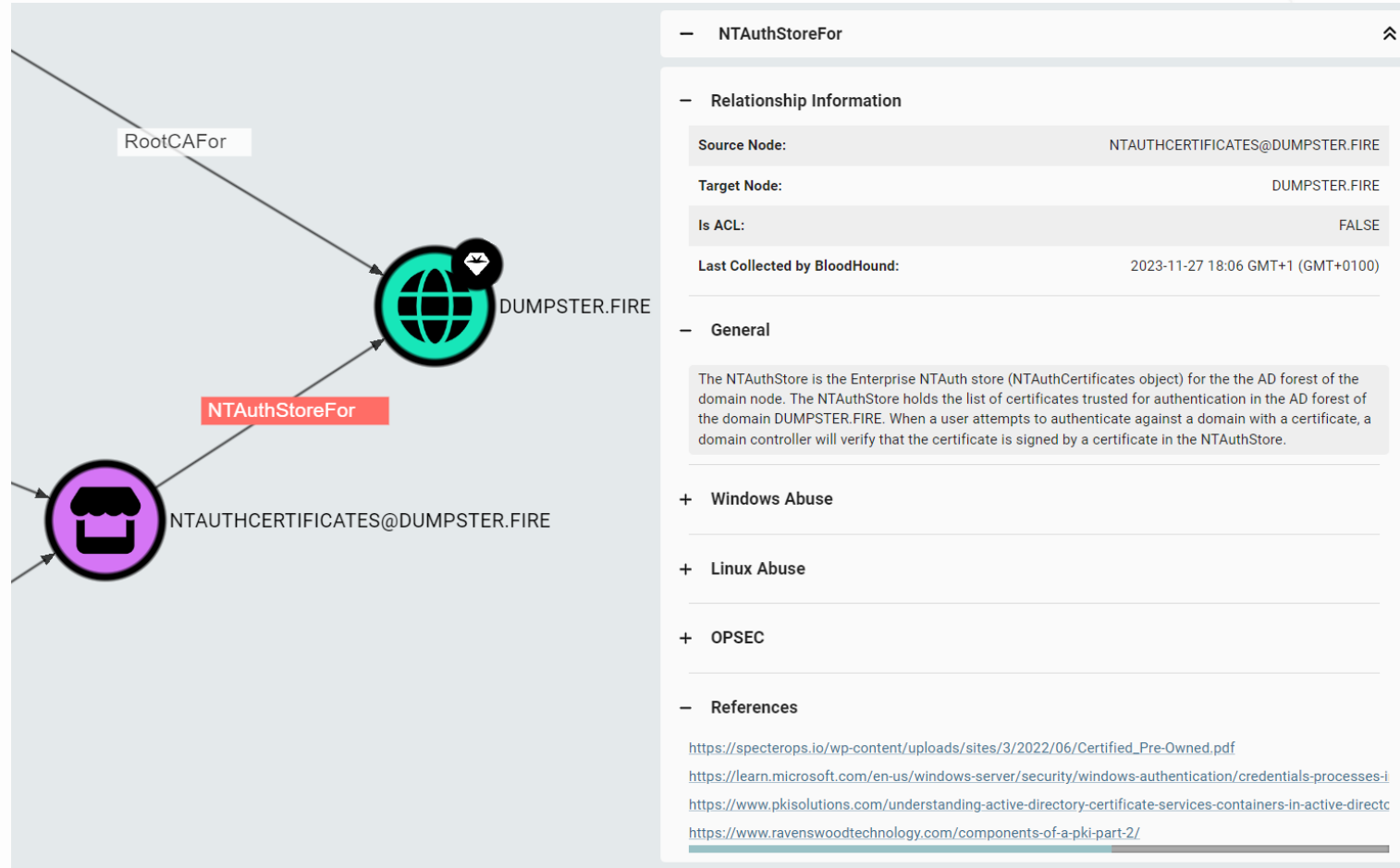
New non-traversable edges

ADCS components in BloodHound



New non-traversable edges

ADCS attack paths in BloodHound



Demo Time!

ADCS in BloodHound Enterprise

New findings

ADCS attack paths in BloodHound

BLOODHOUND ENTERPRISE ATTACK PATHS EXPLORE POSTURE

DUMPSTER.FIRE

DUMPSTER.FIRE ATTACK PATHS

Status: ● Idle
Last Analysis: 2023-11-23 17:18 GMT+1 (GMT+0100)

Non Tier Zero Principals with ADCS ESC1 Privileges HIGH

DESCRIPTION

The principal can perform the ADCS ESC1 attack against the target domain. This attack allows the principal to impersonate any other principal in the forest.

2 PRINCIPALS **TIMELINE**

☐ Show Muted

| Non Tier Zero Principal | Domain |
|--------------------------------------|----------|
| ⋮ AUTHENTICATED USERS@DUMPSTER.FIRE | DUMPSTER |
| ⋮ DOMAIN USERS@DUMPSTER.FIRE | DUMPSTER |

REMEDIATION

Remediating this finding can be approached in several ways, such as unpublishing the affected template(s), restricting enrollment permissions, modifying subject alternate name settings, or prohibiting the use of the certificate for authentication.

The finding can also be remediated by revoking permissions on CAs or modifying NT authentication trust for the enterprise CA(s). Important: these actions, while effective at reducing risk, carry substantial risk of disrupting legitimate usage of enterprise CA(s).

[VIEW / EXPORT FULL REMEDIATION PLAN](#)

New remediations

ADCS attack paths in BloodHound

Non Tier Zero Principals with ADCS ESC1 Privileges

Recommended Remediation

We advise gaining a clear understanding of the intended use of the certificate templates to determine the most suitable remediation approach. This can be achieved through an evaluation of existing certificates and authentication logs, as outlined in the [Certified Pre-Owned ADCS whitepaper](#) sections:

- Monitor User/Machine Certificate Enrollments - DETECT1
- Monitor Certificate Authentication Events - DETECT2

Collaborate with the individual responsible for ADCS within the organization to address the following questions pertaining to the identified certificate templates. This process will help in considering the appropriate checks and remedial actions described below:

1. Is the certificate template in use?

Check: Latest issued certificates and expiration dates.

Remediation: *Unpublish (disable) certificate template*

2. Which principals are enrolling in this template?

Check: Requester principals of issued certificates.

Remediation: *Remove Enroll permission (restrict to Tier Zero)*

3. Is the Subject Alternative Name (SAN) flag required?

Check: If the requester name and the SAN refer to the same principal in issued certificates.

Remediation: *Remove SAN flag*

4. Could the current setup be replaced with an enrollment agent setup?

Check: If it is feasible that a service account or group of employees in the IT department (potentially non-Tier Zero principals) performs the enrollment on behalf of the users that need the certificate.

Remediation: *Implement enrollment agent*

5. Does the certificate template need to allow for authentication?

Check: Login events using certificates created with the certificate template.

Remediation: *Remove EKU that enables authentication*

6. Could future certificate requests wait for a manual approval?

Check: If it is feasible that the certificate request has to wait for a Tier Zero principal to manually approve the request.

Remediation: *Enable manager approval*

Unpublish (disable) certificate template

For every Enterprise CA in the finding:





Thank you



Join us in the BloodHoundGang Slack: <https://ghst.ly/BHSlack>