

Privilege Escalation Techniques in Active Directory Certificate Services

ESC1a, ESC1b, EAB1, EAB2: Extension Injection, Identity Spoofing, and Enrollment Agent Abuse

Windows Server 2025

Active Directory Certificate Services (AD CS)

Table of Contents

1. [Introduction](#)
 2. [Background](#)
 - 2.1 Enrollee Supplies Subject (ESS)
 - 2.2 EnableEnrolleeRequestExtensionList
 - 2.3 StrongCertificateBindingEnforcement (SCBE)
 - 2.4 Authentication Mechanism Assurance (AMA)
 - 2.5 Enroll On Behalf Of (EOBO)
 - 2.6 Template Schema Versions
 3. [ESC1a: CertPolicies Passthrough via CSR Extension Injection](#)
 4. [ESC1b: Security Extension Omission - SAN URL SID Fallback](#)
 5. [EAB1: V1 EOBO Bypass - Missing RA Signature Enforcement](#)
 6. [EAB2: Enrollment Agent Restriction Bypass](#)
 7. [Cross-Version Compatibility Matrix](#)
 8. [Consolidated Detection Guidance](#)
 9. [Consolidated Mitigation Guidance](#)
-

1. Introduction

This paper presents four privilege escalation techniques (ESC1a, ESC1b, EAB1, and EAB2) identified in Active Directory Certificate Services (AD CS) on Windows Server 2025. These techniques exploit

design weaknesses in certificate enrollment, extension passthrough, identity binding, and enrollment agent delegation mechanisms.

All techniques were validated on a fully patched Windows Server 2025 domain controller with an AD CS Enterprise CA. Testing covered all four template schema versions (V1 through V4) and all three StrongCertificateBindingEnforcement (SCBE) levels: 0 (Disabled), 1 (Compatibility), and 2 (Full Enforcement).

Summary of findings:

- **ESC1a:** Self-escalation via Certificate Policy OID injection into Authentication Mechanism Assurance (AMA). Exploitable on all template versions. SCBE-independent.
- **ESC1b:** Identity spoofing via Security Extension omission, forcing the KDC to fall back to SAN URL SID. Exploitable on all template versions. Bypasses SCBE at all enforcement levels.
- **EAB1:** Enroll On Behalf Of (EOBO) bypass on V1 templates due to missing msPKI-RA-Signature enforcement. V1 templates only. SCBE-independent.
- **EAB2:** Unrestricted EOBO when Enrollment Agent Restrictions are not configured. V1 and V2 templates only. SCBE-independent.

Each technique is independently exploitable from a standard low-privileged domain user account. EAB1 and EAB2 require a Certificate Request Agent (CRA) certificate, which can be obtained via ECU injection on an ESS+NoECU template.

2. Background

The following AD CS components and mechanisms are referenced throughout this paper.

2.1 Enrollee Supplies Subject (ESS)

When a certificate template has the CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT flag set in msPKI-Certificate-Name-Flag (value 0x00000001), the enrollee controls the Subject and Subject Alternative Name (SAN) fields of the certificate. This allows the requestor to specify arbitrary UPN, DNS, and SID values in the certificate request. ESS is a prerequisite for ESC1a and ESC1b.

2.2 EnableEnrolleeRequestExtensionList

This CA registry value (located at HKLM\SYSTEM\CurrentControlSet\Services\CertSvc\Configuration\PolicyModules\CertificateAuthority_MicrosoftDefault.Policy) controls which X.509 extensions from the Certificate Signing Request (CSR) are passed through into the issued certificate. When an OID is listed, the CA copies that extension from the CSR without validation. This mechanism is the core enabler for ESC1a (CertPolicies passthrough).

This registry key resides on the CA server and cannot be queried remotely by a low-privileged domain user. From an attacker's perspective, the EnableEnrolleeRequestExtensionList configuration is a

blackbox: its contents can only be inferred by attempting an enrollment and observing whether the CA passes through the requested extension. Detection tools operating with low-privileged credentials face the same limitation and cannot directly audit this setting.

2.3 StrongCertificateBindingEnforcement (SCBE)

SCBE is a KDC registry value (HKLM\SYSTEM\CurrentControlSet\Services\Kdc\StrongCertificateBindingEnforcement) that controls how strictly the KDC validates certificate identity during PKINIT authentication:

- **Value 0 (Disabled):** No SID validation from the Security Extension.
- **Value 1 (Compatibility):** SID is checked if present; certificates without the extension are accepted.
- **Value 2 (Full Enforcement):** SID in the Security Extension must match the authenticating account.

A critical finding of this research is that SCBE has zero impact on any of the four techniques presented. ESC1a is self-escalation with no SID mismatch. ESC1b omits the extension entirely, triggering a fallback. EAB1 and EAB2 produce certificates with legitimate target identities populated by the CA.

2.4 Authentication Mechanism Assurance (AMA)

AMA maps issuance policy OIDs to security groups. When a certificate's CertPolicies extension (OID 2.5.29.32) contains an OID that has an msDS-OIDToGroupLink attribute pointing to a universal security group, the KDC adds that group's SID to the user's Kerberos PAC during PKINIT. This is the escalation mechanism exploited by ESC1a. AMA affects only Kerberos authentication, not NTLM.

2.5 Enroll On Behalf Of (EOBO)

EOBO allows an enrollment agent holding a certificate with the Certificate Request Agent EKU (OID 1.3.6.1.4.1.311.20.2.1) to request certificates on behalf of other users. On V2 and later templates, EOBO requires msPKI-RA-Signature ≥ 1 and the CRA OID in msPKI-RA-Application-Policies. The CA validates the co-signature before issuance. EAB1 exploits V1 templates where this validation is absent. EAB2 exploits the lack of EnrollmentAgentRights restrictions on the CA.

2.6 Template Schema Versions

AD CS templates have evolved through four schema versions:

- **V1:** Original format (Windows 2000 / Windows Server 2003). Lacks msPKI-RA-Signature, msPKI-RA-Application-Policies, and CNG algorithm attributes.
- **V2:** Introduced msPKI-RA-Signature for EOBO enforcement.
- **V3:** Added CNG algorithm support via msPKI-Asymmetric-Algorithm and related attributes stored in msPKI-RA-Application-Policies using a backtick-delimited format.
- **V4:** Extends V3 with Key Attestation support.

On V3 and V4 templates, msPKI-RA-Application-Policies serves a dual purpose: it stores both CNG algorithm configuration (backtick-delimited format) and RA policy OIDs (bare OID format). These formats are mutually incompatible, creating an architectural conflict that prevents EOBO from being configured on V3/V4 templates. This is relevant to EAB2.

3. ESC1a: CertPolicies Passthrough via CSR Extension Injection

3.1 Description

ESC1a abuses the EnableEnrolleeRequestExtensionList to inject arbitrary Certificate Policies (OID 2.5.29.32) into issued certificates via the CSR. When the injected OID matches an AMA-linked issuance policy, the KDC adds the linked privileged group's SID to the attacker's Kerberos PAC during PKINIT authentication.

ESC1a is a self-escalation attack: the attacker authenticates as themselves but with elevated group membership. No identity spoofing is involved.

Relationship to ESC1

ESC1a is classified under the ESC1 family because both techniques share the same root prerequisite and attack surface: ESS templates (CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT) with enrollment rights granted to low-privileged users. The naming convention groups techniques by the template condition that enables them, not by the escalation outcome. This has a practical reason: detection tools like Certipy evaluate templates once and flag all applicable vulnerabilities. Grouping ESC1a and ESC1b under ESC1 means a single ESS+NoEKU template produces ESC1a and ESC1b findings rather than overlapping ESC1/ESC13 flags on the same object.

ESC1 as documented in the Certipy wiki is a template-level misconfiguration: ESS enabled, authentication EKU present, enrollment rights too broad. The attacker controls the SAN fields (UPN, DNS) via ESS, and with the `-sid` flag, Certipy also injects the target's SID into the certificate. ESC1 does not reference EnableEnrolleeRequestExtensionList; the exploitation relies on the CA accepting attacker-supplied SAN values because the template has ESS set.

ESC1a goes further. It exploits the EnableEnrolleeRequestExtensionList to inject CertPolicies (OID 2.5.29.32) into the issued certificate via the CSR. This extension is not a SAN field; it is a separate X.509 extension that the CA copies from the CSR because the OID is in the allowlist by default. The escalation outcome is also different: ESC1 achieves identity spoofing, while ESC1a achieves self-escalation via AMA group membership injection. The attacker authenticates as themselves but with elevated group memberships in the Kerberos PAC.

	ESC1	ESC1a
Root prerequisite	ESS (CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT)	ESS (CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT)
CA mechanism exploited	SAN control via ESS template flag	CertPolicies passthrough via EnableEnrolleeRequestExtensionList
What the attacker controls	SAN fields (UPN, DNS, SID URL)	CertPolicies extension (OID 2.5.29.32)
EKU requirement	Client Authentication or equivalent	None (NoEKU)
Escalation type	Identity spoofing (impersonation)	Self-escalation (AMA group membership)
KDC behavior exploited	PKINIT identity resolution from SAN/Security Extension	PKINIT PAC injection from CertPolicies OID
Additional AD requirement	None	AMA OID-to-group link must exist
Bypasses SCBE=2	Yes (with -sid)	N/A (no SID mismatch, SCBE irrelevant)

ESC1 gives the attacker a certificate that authenticates as someone else. ESC1a gives the attacker a certificate that authenticates as themselves but with group memberships they do not hold. The shared foundation is ESS: without CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT, neither technique is possible.

Relationship to ESC13

ESC1a exploits the same AMA mechanism as ESC13 but differs in how the issuance policy OID reaches the certificate. In ESC13, an administrator has already configured the template with an issuance policy that happens to have an msDS-OIDToGroupLink, so the attacker simply enrolls in a template that was set up poorly. In ESC1a, the template carries no issuance policy at all. The attacker injects the OID directly into the CSR via the EnableEnrolleeRequestExtensionList passthrough, targeting any AMA-linked OID in the forest regardless of what any template references.

	ESC13	ESC1a
Extension abused	CertPolicies (2.5.29.32)	CertPolicies (2.5.29.32)
OID source	Template issuance policy (admin-configured)	Attacker-injected via CSR extension
ESS required	No	Yes (for CSR extension passthrough)
Template must reference AMA OID	Yes	No; any AMA OID in the forest works
Attacker chooses which OID	No, locked to template config	Yes, arbitrary OID in the request
Admin misconfiguration needed	Template linked to AMA OID	Only the AMA OID-to-group link itself

The practical difference: ESC13 is a template-level misconfiguration that an admin can fix by delinking the issuance policy. ESC1a does not depend on any template referencing a specific policy. If AMA exists anywhere in the forest and the CA passes through CertPolicies from CSRs, every ESS+NoEKU template is a viable attack surface.

Relationship to ESC15

ESC15 (CVE-2024-49019, "EKUwu") and ESC1a both inject policy extensions through the CSR on ESS templates, but they target different extensions, exploit different root causes, and produce different outcomes.

ESC15 injects Application Policies (szOID_APPLICATION_CERT_POLICIES) into the CSR. On unpatched CAs, this extension is copied into the issued certificate without validation, granting ECU capabilities not intended by the template. For example, injecting Client Authentication into a WebServer V1 template enables PKINIT or Schannel authentication, and injecting Certificate Request Agent enables EOBO enrollment. The November 2024 patch for CVE-2024-49019 blocks this injection. ESC15 is a vulnerability that has been fixed.

ESC1a injects Certificate Policies (OID 2.5.29.32) into the CSR. This is a different extension entirely. The CA copies it because 2.5.29.32 is listed in EnableEnrolleeRequestExtensionList by default. This is not a vulnerability or a missing patch; it is the intended behavior of the extension allowlist. The injected OID does not grant ECU capabilities. Instead, the KDC evaluates the OID during PKINIT and, if it matches an AMA-linked issuance policy, adds the linked group's SID to the attacker's PAC. ESC1a works on fully patched Server 2025.

ESC15		ESC1a
Extension injected	Application Policies (szOID_APPLICATION_CERT_POLICIES)	Certificate Policies (OID 2.5.29.32)
Root cause	CA bug (copies Application Policies from CSR on V1)	Design behavior (EnableEnrolleeRequestExtensionList allowlist)
Patched	Yes (CVE-2024-49019, November 2024)	No, works on fully patched CAs
Template version	V1 only	All versions (V1 through V4)
Escalation type	EKU capability injection (Client Auth, CRA)	AMA group membership injection
Outcome	Identity spoofing or EOBO capability	Self-escalation (elevated PAC groups)
Requires ESS	Yes	Yes
Additional AD requirement	None	AMA OID-to-group link must exist

ESC15 gives the certificate capabilities it should not have. ESC1a gives the authenticating user group memberships they should not have. The two techniques share the attack surface (ESS templates) and the delivery mechanism (CSR extension injection) but diverge in every other dimension: the extension, the root cause, the patch status, and the escalation path.

3.2 Prerequisites

1. ESS-enabled template (CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT).
2. No Extended Key Usage defined (pKIExtendedKeyUsage is empty).
3. EnableEnrolleeRequestExtensionList includes OID 2.5.29.32 (Certificate Policies).
4. An AMA configuration exists: an msPKI-Enterprise-Oid object with msDS-OIDToGroupLink pointing to a privileged security group.
5. Low-privileged users have enrollment rights on the template.
6. No manager approval required. No authorized signatures required (msPKI-RA-Signature = 0).

3.3 Technical Analysis

The Certificate Policies extension (OID 2.5.29.32) carries one or more policy OIDs indicating the policies under which a certificate was issued. In AD CS, these OIDs can correspond to issuance policies defined as msPKI-Enterprise-Oid objects in the Configuration partition.

When OID 2.5.29.32 is listed in EnableEnrolleeRequestExtensionList, the CA copies this extension from the CSR into the issued certificate without validation. The attacker can therefore specify any issuance policy OID in the request.

During PKINIT, the KDC evaluates the CertPolicies extension. If an OID has an msDS-OIDToGroupLink attribute, the linked group's SID is added to the PAC. The CA does not validate whether the enrollee is entitled to the claimed issuance policy; it passes through whatever the CSR contains.

3.4 Attack Flow

1. Identify a vulnerable template: ESS enabled, no EKU, enrollment rights granted.
2. Identify the AMA-linked issuance policy OID by querying msPKI-Enterprise-Oid objects with msDS-OIDToGroupLink set.
3. Construct a CSR with the CertPolicies extension (2.5.29.32) containing the target OID.
4. Submit the request. The CA passes through the CertPolicies extension from the CSR.
5. Authenticate via PKINIT. The KDC evaluates the CertPolicies extension, finds the AMA-linked OID, and adds the linked group's SID to the PAC.
6. The attacker obtains a TGT with effective membership in the privileged group.

3.5 Exploitation

```
certipy req -u 'user.alice@lab.local' -p 'D3m0P@ss2026!' \  
-dc-ip 192.168.163.185 -ca 'lab-CA' \  
-template 'ESC1a-V2' \  
-upn 'user.alice@lab.local' \  
-sid 'S-1-5-21-...-1105' \  
-certificate-policies '<AMA-OID>'  
  
certipy auth -pfx esc1a.pfx -dc-ip 192.168.163.185 -domain lab.local
```

3.6 Affected Versions

Schema Version	SCBE Value	Result
V1	0	Exploitable
V1	1	Exploitable
V1	2	Exploitable
V2	0	Exploitable
V2	1	Exploitable
V2	2	Exploitable
V3	0	Exploitable
V3	1	Exploitable
V3	2	Exploitable
V4	0	Exploitable
V4	1	Exploitable
V4	2	Exploitable

ESC1a is version-independent. SCBE has no impact because the certificate SID matches the authenticating user. There is no SID mismatch for SCBE to detect.

3.7 Impact

Privilege escalation from any domain user to any AMA-linked security group. The escalation is session-scoped (lifetime of the Kerberos TGT) and affects only Kerberos, not NTLM. Detection is difficult because the certificate is legitimately issued to the attacker's own identity.

3.8 Detection

- Monitor CA audit logs (Event ID 4887) for certificate requests containing the CertPolicies extension (2.5.29.32) with OIDs matching AMA-linked issuance policies.
- Alert on certificates issued from ESS+NoEKU templates that contain CertPolicies extensions.
- Monitor Kerberos TGT requests (Event ID 4768) where the PAC contains group memberships not present in the user's actual AD group membership.

3.9 Mitigation

- Remove OID 2.5.29.32 from the CA's EnableEnrolleeRequestExtensionList.
- Remove the ESS flag from templates that do not require it.

- Add EKU constraints to all templates.
 - Audit and minimize msDS-OIDToGroupLink mappings.
 - Restrict enrollment rights to the minimum necessary security principals.
-

4. ESC1b: Security Extension Omission - SAN URL SID Fallback

4.1 Description

ESC1b exploits the KDC's fallback behavior when the `szOID_NTDS_CA_SECURITY_EXT` extension (1.3.6.1.4.1.311.25.2) is absent from a certificate. When this extension is missing, the KDC falls back to reading the SID from the Subject Alternative Name (SAN) URL field for identity resolution during PKINIT.

On ESS-enabled templates, the attacker controls the SAN contents. By crafting a CSR that suppresses the Security Extension, the attacker forces the KDC to use the SAN URL SID, which contains the target victim's SID.

Relationship to ESC9 and ESC16

ESC9, ESC16, and ESC1b all produce the same outcome: a certificate without the Security Extension, forcing the KDC to fall back to weaker identity resolution. They differ in how the extension gets suppressed, what prerequisites the attacker needs, and how far the attack reaches.

ESC9 requires the template to have `CT_FLAG_NO_SECURITY_EXTENSION` set, an explicit misconfiguration that an admin chose or inherited. ESC16 requires the CA itself to have added OID 1.3.6.1.4.1.311.25.2 to its `DisableExtensionList` registry key, suppressing the extension globally across all templates. Both are configuration errors that a defender can audit and fix.

ESC1b requires neither. On an ESS template, the attacker simply does not include the Security Extension in the CSR, and the CA does not add one on its own. No template flag, no CA registry change. The extension is absent by default behavior when the CSR omits it on an ESS template.

The second difference is what happens after the extension is gone. ESC9 and ESC16 leave the attacker without direct SAN control. To exploit them, the attacker needs either `GenericWrite` over a victim account (to swap UPNs) with `SCBE < 2`, or `ESC6` on the CA (`EDITF_ATTRIBUTESUBJECTALTNAME2`) to inject a SAN URL SID. Both are additional prerequisites that may or may not be present.

ESC1b has neither constraint. ESS gives the attacker full SAN control as part of the same template that enables the extension omission. The attacker places the target's SID in the SAN URL field directly, with no UPN manipulation, no `ESC6`, and no `SCBE` restriction.

	ESC9	ESC16	ESC1b
Why the Security Extension is absent	Template flag (CT_FLAG_NO_SECURITY_EXTENSION)	CA registry (DisableExtensionList)	ESS: CA defers to CSR, CSR omits it
Misconfiguration required	Yes (template)	Yes (CA)	None
How attacker controls SAN SID	Needs ESC6 or UPN manipulation	Needs ESC6 or UPN manipulation	ESS: direct SAN control
Works at SCBE=2	Only with ESC6	Only with ESC6	Yes, natively
Additional prerequisites	GenericWrite or ESC6	GenericWrite or ESC6	Enrollment rights only
Scope	Specific template	All templates on the CA	Any ESS+NoEKU template
Defender can audit via	Template enrollment flags	CA registry	Requires checking ESS templates for missing extensions post-issuance

ESC1b is the zero-prerequisites version of this attack class. The primitive is the same (absent Security Extension triggers SAN URL SID fallback), but ESC1b reaches it without any misconfiguration and exploits it without any secondary vulnerability.

4.2 Prerequisites

1. ESS-enabled template with Client Authentication, Any Purpose, or no EKU.
2. The Security Extension can be suppressed via CSR manipulation.
3. Low-privileged users have enrollment rights on the template.
4. No manager approval or authorized signatures required.

4.3 Technical Analysis

The Windows KDC resolves certificate identity during PKINIT in the following priority order:

1. Security Extension SID (OID 1.3.6.1.4.1.311.25.2) - highest priority.
2. SAN URL SID (encoded as tag:microsoft.com,2022-09-14:sid:) - fallback.
3. SAN UPN - lowest priority for identity resolution.

The SAN URL SID format was introduced by Microsoft as part of KB5014754 as a legitimate strong mapping option for scenarios where the Security Extension cannot be used, such as offline certificate requests or manual mapping. Microsoft documented this format in the "Preview of SAN URI for Certificate Strong Mapping for KB5014754" blog post. The format uses a tag-based URI in the SAN: `URL=tag:microsoft.com,2022-09-14:sid:<value>`, where the value is the user or device SID.

When the Security Extension is present, the KDC validates its SID against the user account. When absent, the KDC falls back to the SAN URL SID. ESC1b exploits this fallback by ensuring the Security Extension is not present, forcing the KDC to use the attacker-controlled SAN URL SID. The SAN URI was designed as a mapping mechanism for legitimate certificate issuance; ESC1b abuses it as a fallback identity source on ESS templates where the attacker controls both the SAN contents and the absence of the Security Extension.

The Security Extension can be suppressed through several mechanisms:

1. The CSR explicitly requests suppression (Certipy's `-no-security-ext` flag).
2. The template has the `CT_FLAG_NO_SECURITY_EXTENSION` enrollment flag set (0x00080000).
3. The CA's policy module is configured not to add the extension.

ESC1b uses the first mechanism.

4.4 Attack Flow

1. Identify an ESS template with enrollment rights.
2. Construct a CSR with the admin's UPN in the SAN, the admin's SID in the SAN URL field, and the flag to suppress the Security Extension.
3. Submit the request. The CA issues a certificate without the Security Extension.
4. During PKINIT, the KDC finds no Security Extension, falls back to the SAN URL SID (admin SID), and issues a TGT for the admin account.

4.5 Exploitation

```
certipy req -u 'user.alice@lab.local' -p 'D3m0P@ss2026!' \
  -dc-ip 192.168.163.185 -ca 'lab-CA' \
  -template 'ESC1b-Demo' \
  -upn 'administrador@lab.local' \
  -sid 'S-1-5-21-4057504432-3103255123-3127012112-500' \
  -no-security-ext

certipy auth -pfx esc1b.pfx -dc-ip 192.168.163.185 -domain lab.local
```

4.6 Affected Versions

Schema Version	SCBE Value	Result
V1	0	Exploitable
V1	1	Exploitable
V1	2	Exploitable
V2	0	Exploitable
V2	1	Exploitable
V2	2	Exploitable
V3	0	Exploitable
V3	1	Exploitable
V3	2	Exploitable
V4	0	Exploitable
V4	1	Exploitable
V4	2	Exploitable

Version-independent. Bypasses SCBE at all levels because SCBE validates the Security Extension SID. When the extension is absent, the fallback to SAN URL SID occurs outside the SCBE validation path.

4.7 Impact

Full domain compromise. This technique exploits a design assumption in the KDC's certificate processing: that the Security Extension will always be present when strong binding is required. By violating this assumption, SCBE is entirely circumvented.

4.8 Detection

- Monitor for certificates issued from ESS templates that lack the Security Extension.
- Alert on PKINIT authentications (Event ID 4768) where the certificate does not contain the Security Extension.
- Audit CA logs for enrollment requests where the CSR requests suppression of the Security Extension.

4.9 Mitigation

- Enforce that the CA always adds the Security Extension regardless of CSR flags.

- Configure the KDC to reject certificates missing the Security Extension when SCBE >= 1.
 - Remove the ESS flag from templates unless required.
 - Remove the CT_FLAG_NO_SECURITY_EXTENSION enrollment flag from all templates.
-

5. EAB1: V1 EOBO Bypass - Missing RA Signature Enforcement

5.1 Description

EAB1 exploits a fundamental architectural difference between V1 (schema version 1) templates and later versions in how they handle Enroll On Behalf Of (EOBO) requests.

On V2 and later templates, EOBO requires msPKI-RA-Signature >= 1, which enforces CSR co-signature by a certificate with the Certificate Request Agent (CRA) EKU. V1 templates predate this attribute and do not enforce co-signature requirements. An attacker with a CRA certificate can submit EOBO requests against V1 templates without any RA signature validation, obtaining certificates for any domain principal.

The most impactful target is the Domain Controller machine account, enabling DCSync for full domain credential extraction.

Relationship to ESC3

ESC3 documents the baseline EOBO abuse path and already identifies V1 templates (User, Machine) as valid EOBO targets. The Certipy wiki notes that "Schema Version 1 templates... inherently permit [EOBO] without requiring specific enrollment agent restrictions." The ESC3 exploitation example uses the default User template (V1) as the EOBO target.

EAB1 does not change how the CRA certificate is obtained; that step is identical. What EAB1 adds are two findings that ESC3 does not cover.

First, the false hardening problem. An administrator who reads ESC3 guidance and tries to block EOBO on V1 templates by setting msPKI-RA-Signature via LDAP will see the write succeed (the attribute is stored in AD), but the CA ignores it entirely on V1 templates. On V2+ templates, msPKI-RA-Signature = 0 causes the CA to actively reject EOBO with CERTSRV_E_BAD_RENEWAL_SUBJECT. On V1 templates, the CA's code path never reads the attribute, so the value has no effect.

Second, the asymmetry in CA behavior. ESC3 treats V1 and V2+ templates similarly as EOBO targets without documenting the difference in how the CA handles each. EAB1 identifies the behavioral split: V2+ templates have an explicit rejection mechanism (RA-Sig = 0 blocks EOBO), while V1 templates have none. This means V1 templates cannot be hardened through attribute configuration. The only effective mitigation is upgrading the schema version.

	ESC3	EAB1
CRA certificate acquisition	Template with CRA ECU + enrollment rights	Same
V1 templates as EOBO targets	Documented (User, Machine mentioned)	Same finding
V2+ with RA-Sig=0 blocks EOBO	Not documented	Yes, CERTSRV_E_BAD_RENEWAL_SUBJECT
V1 RA-Sig hardening effective	Not addressed	No: LDAP write succeeds, CA ignores
Default templates at risk	User and Machine (V1) mentioned	All seven default V1 templates enumerated
Mitigation	Restrict enrollment rights, manager approval	Upgrade V1 to V2+; RA-Sig setting alone is insufficient

ESC3 identifies the attack surface. EAB1 documents why the obvious hardening step (set msPKI-RA-Signature) does not work on V1 templates, and why schema version upgrade is the only reliable fix.

5.2 Prerequisites

1. A V1 template (msPKI-Template-Schema-Version = 1) with Client Authentication ECU.
2. Low-privileged users have enrollment rights on the template.
3. The attacker possesses a CRA certificate (obtainable via ECU injection on an ESS+NoECU template).
4. No EnrollmentAgentRights restrictions configured on the CA.

5.3 Technical Analysis

V1 templates (Windows 2000 / Windows Server 2003 era) lack the msPKI-RA-Signature and msPKI-RA-Application-Policies attributes. When a V2 or later template has msPKI-RA-Signature = 0, the CA rejects EOBO requests with CERTSRV_E_BAD_RENEWAL_SUBJECT (0x80094806). However, this validation is skipped entirely for V1 templates because the attribute does not exist in the V1 schema.

The CA still validates the CRA ECU on the co-signing certificate, but does not check whether the template was designed to support EOBO. This allows any V1 template with Client Authentication to be used as an EOBO target.

By enrolling on behalf of the Domain Controller's machine account (e.g., DC01\$), the attacker obtains a certificate that authenticates as the DC. This certificate can then be used for DCSync (via MS-DRSR), extracting all domain credential hashes.

5.4 Attack Flow

1. Obtain a CRA certificate (e.g., via ECU injection on an ESS+NoEKU template).
2. Identify a V1 template with Client Authentication ECU and enrollment rights.
3. Submit an EOBO request specifying the DC machine account as the target, co-signed with the CRA cert.
4. The CA skips RA signature validation (V1 template). Certificate issued for the DC account.
5. Authenticate via PKINIT as the DC machine account.
6. Perform DCSync to extract all domain credentials.

5.5 Exploitation

```
# Obtain CRA cert via ECU injection on ESS+NoEKU template
certipy req -u 'user.alice@lab.local' -p 'D3m0P@ss2026!' \
  -dc-ip 192.168.163.185 -ca 'lab-CA' \
  -template 'ESS-NoEKU-V2' \
  -inject-eku 'Certificate Request Agent'

# EOBO as DC machine account on V1 template
certipy req -u 'user.alice@lab.local' -p 'D3m0P@ss2026!' \
  -dc-ip 192.168.163.185 -ca 'lab-CA' \
  -template 'User' \
  -on-behalf-of 'LAB\DC01$' \
  -pfx cra.pfx

# Authenticate as DC and DCSync
certipy auth -pfx eab1.pfx -dc-ip 192.168.163.185 -domain lab.local
```

5.6 Affected Versions

Schema Version	SCBE Value	Result
V1	0	Exploitable (DC\$ hash obtained)
V1	1	Exploitable (DC\$ hash obtained)
V1	2	Exploitable (DC\$ hash obtained)
V2	0	Blocked (CERTSRV_E_BAD_RENEWAL_SUBJECT)
V2	1	Blocked (CERTSRV_E_BAD_RENEWAL_SUBJECT)
V2	2	Blocked (CERTSRV_E_BAD_RENEWAL_SUBJECT)
V3	0	Blocked (CERTSRV_E_BAD_RENEWAL_SUBJECT)
V3	1	Blocked (CERTSRV_E_BAD_RENEWAL_SUBJECT)
V3	2	Blocked (CERTSRV_E_BAD_RENEWAL_SUBJECT)
V4	0	Blocked (CERTSRV_E_BAD_RENEWAL_SUBJECT)
V4	1	Blocked (CERTSRV_E_BAD_RENEWAL_SUBJECT)
V4	2	Blocked (CERTSRV_E_BAD_RENEWAL_SUBJECT)

V1 templates only. V2 and later templates enforce msPKI-RA-Signature validation. SCBE has no impact because the EOBO certificate contains the target's legitimate identity as populated by the CA.

5.7 Impact

Full domain compromise via any V1 template with Client Authentication. The attack chain (EKU injection to EAB1 to DCSync) requires only standard user credentials and enrollment rights on two templates. V1 templates are common in environments upgraded from Windows Server 2003. Default templates such as User, Machine, and DomainController are V1.

5.8 Detection

- Monitor CA logs for EOBO enrollment requests (RequesterName differs from the authenticated caller) against V1 templates.
- Alert on certificates issued for DC machine accounts via enrollment agent requests.
- Monitor for DCSync activity (DRS replication from non-DC sources).
- Audit which V1 templates exist and whether they grant enrollment rights to low-privileged users.

5.9 Mitigation

- Upgrade all V1 templates to V2 or higher. Set msPKI-Template-Schema-Version ≥ 2 and configure msPKI-RA-Signature appropriately.
 - If V1 templates must be retained, restrict enrollment rights to trusted administrative accounts only.
 - Enable EDITF_ENABLEAGENTRESTRICTION on the CA and configure EnrollmentAgentRights rules.
 - Monitor for CRA certificates issued to unexpected accounts.
-

6. EAB2: Enrollment Agent Restriction Bypass

6.1 Description

EAB2 exploits the absence of Enrollment Agent Restrictions on the Certificate Authority. When a template has msPKI-RA-Signature ≥ 1 with the CRA ECU as the required co-signing policy, it enables EOBO. However, if EDITF_ENABLEAGENTRESTRICTION is not configured with explicit EnrollmentAgentRights rules, any CRA certificate holder can enroll on behalf of any user for any EOBO-enabled template.

EAB2 is exploitable on V1 and V2 templates. V3 and V4 templates are architecturally immune due to the dual-purpose conflict in the msPKI-RA-Application-Policies attribute.

Relationship to ESC3

ESC3 focuses on the supply side: if a CRA certificate can be obtained, EOBO is possible. The implicit assumption is that EnrollmentAgentRights on the CA limits who can use EOBO, against which templates, and for which target users. ESC3 does not examine what happens when those restrictions are absent.

EAB2 fills that gap. On a default Windows Server 2025 installation, EDITF_ENABLEAGENTRESTRICTION is not set, and the certutil command to enable it (`certutil -setreg policy\EditFlags +EDITF_ENABLEAGENTRESTRICTION`) fails with ERROR_INVALID_DATA. No enrollment agent restrictions exist, and the documented method to create them does not work out of the box. Any CRA certificate holder has unrestricted EOBO access to every V1 and V2 template on the CA.

	ESC3	EAB2
CRA certificate acquisition	Template with CRA ECU + enrollment rights	Same
EOBO restrictions assumed	Present but misconfigured or too broad	Absent, not configured by default
Can admin restrict EOBO	Yes, configure EnrollmentAgentRights	certutil fails with ERROR_INVALID_DATA on Server 2025
Scope of EOBO access	Limited by restrictions	Unrestricted: any user, any V1/V2 template
V3/V4 templates	Subject to RA-Policy attribute	Architecturally immune (CNG attribute conflict)
Root cause	Overly permissive delegation config	No delegation controls exist out of the box

Where ESC3 is "the delegation was set up wrong," EAB2 is "the delegation was never locked down, and the mechanism to lock it down is broken on the current platform."

6.2 Prerequisites

1. A V1 or V2 template with msPKI-RA-Signature ≥ 1 and the CRA OID (1.3.6.1.4.1.311.20.2.1) in msPKI-RA-Application-Policies.
2. The template has Client Authentication or equivalent ECU.
3. Low-privileged users have enrollment rights on the template.
4. The attacker possesses a CRA certificate (obtainable via ECU injection on an ESS+NoECU template).
5. EnrollmentAgentRights not configured on the CA (this is the default).

6.3 Technical Analysis

Windows CA supports Enrollment Agent Restrictions (EDITF_ENABLEAGENTRESTRICTION) that can limit EOBO operations by specifying which enrollment agent certificates are authorized, which templates they can use for EOBO, and which target users they can enroll for. When not configured (the default), any CRA certificate holder has unrestricted EOBO access.

Many organizations deploy EOBO templates for legitimate delegated enrollment workflows (e.g., helpdesk provisioning smart cards) but fail to configure the corresponding restrictions.

V3/V4 Architectural Protection

V3 and V4 templates are immune due to an architectural conflict in msPKI-RA-Application-Policies. On these templates, this attribute serves two purposes:

1. Required ECU for co-signing certificates (bare OID format: '1.3.6.1.4.1.311.20.2.1').

2. CNG algorithm configuration string (backtick-delimited format).

These formats are mutually incompatible. With the CNG config present (required for V3/V4 recognition), the CA cannot parse the CRA OID. With the CRA OID alone, the CA does not recognize the template as V3/V4 and rejects enrollment with CERTSRV_E_UNSUPPORTED_CERT_TYPE. With both as multi-valued entries, the CA rejects with CERTSRV_E_TEMPLATE_POLICY_REQUIRED.

This dual-purpose attribute conflict is an unintentional security benefit in V3/V4 templates.

6.4 Attack Flow

1. Obtain a CRA certificate (via ECU injection on an ESS+NoEQU template).
2. Identify a V1 or V2 EOBO template (msPKI-RA-Signature >= 1, CRA policy).
3. Verify that EnrollmentAgentRights are not configured on the CA.
4. Submit an EOBO request for the target user, co-signed with the CRA certificate.
5. The CA validates the CRA co-signature, finds no restrictions, and issues the certificate.
6. Authenticate as the target user via PKINIT.

6.5 Exploitation

```
# Obtain CRA cert via ECU injection on ESS+NoEQU template
certipy req -u 'user.alice@lab.local' -p 'D3m0P@ss2026!' \
  -dc-ip 192.168.163.185 -ca 'lab-CA' \
  -template 'ESS-NoEQU-V2' \
  -inject-eku 'Certificate Request Agent'

# EOBO as administrator on V2 template
certipy req -u 'user.alice@lab.local' -p 'D3m0P@ss2026!' \
  -dc-ip 192.168.163.185 -ca 'lab-CA' \
  -template 'EAB2-V2' \
  -on-behalf-of 'LAB\administrator' \
  -pfx cra.pfx

# Authenticate as administrator
certipy auth -pfx eab2.pfx -dc-ip 192.168.163.185 -domain lab.local
```

6.6 Affected Versions

Schema Version	SCBE Value	Result
V1	0	Exploitable (admin hash obtained)
V1	1	Exploitable (admin hash obtained)
V1	2	Exploitable (admin hash obtained)
V2	0	Exploitable (admin hash obtained)
V2	1	Exploitable (admin hash obtained)
V2	2	Exploitable (admin hash obtained)
V3	0	Blocked (CERTSRV_E_TEMPLATE_POLICY_REQUIRED)
V3	1	Blocked (CERTSRV_E_TEMPLATE_POLICY_REQUIRED)
V3	2	Blocked (CERTSRV_E_TEMPLATE_POLICY_REQUIRED)
V4	0	Blocked (CERTSRV_E_TEMPLATE_POLICY_REQUIRED)
V4	1	Blocked (CERTSRV_E_TEMPLATE_POLICY_REQUIRED)
V4	2	Blocked (CERTSRV_E_TEMPLATE_POLICY_REQUIRED)

V1 and V2 only. V3/V4 are architecturally protected by the CNG/RA-policy attribute conflict. SCBE has no impact because the EOBO certificate contains the legitimate target identity.

6.7 Impact

Full domain compromise via any unrestricted EOBO template on V1/V2. The attacker can impersonate any user, including Domain Administrator. The risk is amplified by the fact that EOBO templates are commonly deployed but EnrollmentAgentRights restrictions are frequently absent.

6.8 Detection

- Monitor CA logs for EOBO requests from unauthorized enrollment agents.
- Alert on EOBO requests from CRA certificates not issued through legitimate provisioning.
- Audit the CA's EnrollmentAgentRights configuration. If empty, all EOBO templates are unrestricted.
- Monitor for EOBO-issued certificates targeting privileged accounts.

6.9 Mitigation

- Configure EDITF_ENABLEAGENTRESTRICTION with explicit EnrollmentAgentRights rules.
- Limit CRA certificate issuance to specific trusted enrollment agent accounts.

- Restrict EOBO templates to necessary target user groups.
- Upgrade EOBO templates to V3/V4 where architecturally possible (this disables EOBO entirely).

7. Cross-Version Compatibility Matrix

The following table summarizes exploitation results across all template schema versions and SCBE enforcement levels:

ESC	V1	V2	V3	V4	SCBE Impact	Attack Type
ESC1a	Yes	Yes	Yes	Yes	None	Self-Escalation
ESC1b	Yes	Yes	Yes	Yes	None (bypassed)	Impersonation
EAB1	Yes	No	No	No	None	EOBO + DCSync
EAB2	Yes	Yes	No	No	None	EOBO

ESC1a and ESC1b work across all template schema versions because they exploit CA-level extension passthrough behavior, which is version-agnostic.

EAB1 is restricted to V1 templates. V2 and later templates enforce msPKI-RA-Signature = 0 as an explicit block on EOBO.

EAB2 is restricted to V1 and V2 templates. V3 and V4 are architecturally immune due to the msPKI-RA-Application-Policies dual-purpose conflict between CNG algorithm configuration and RA policy OIDs.

StrongCertificateBindingEnforcement has zero impact on all four techniques. The primary certificate binding mitigation does not protect against any of these escalation paths.

8. Consolidated Detection Guidance

8.1 CA-Level Monitoring

- Monitor CA audit logs (Event ID 4887) for certificate requests containing CertPolicies (2.5.29.32) with AMA-linked OIDs. [ESC1a]
- Monitor for enrollment requests where the CSR requests suppression of the Security Extension. [ESC1b]
- Monitor for EOBO requests (RequesterName differs from authenticated caller) against V1 templates. [EAB1]
- Monitor for EOBO requests from unauthorized or unexpected CRA certificates. [EAB2]

8.2 KDC-Level Monitoring

- Monitor Kerberos TGT requests (Event ID 4768) where PAC group memberships exceed the user's actual AD group membership. [ESC1a]
- Alert on PKINIT authentications where the certificate does not contain the Security Extension. [ESC1b]

8.3 Certificate-Level Monitoring

- Alert on certificates from ESS templates containing CertPolicies extensions with AMA-linked OIDs.
 - Alert on certificates where the Security Extension SID does not match the requester's SID.
 - Alert on certificates from ESS templates that lack the Security Extension.
 - Alert on EOBO-issued certificates targeting privileged accounts (Domain Admins, DC machine accounts).
-

9. Consolidated Mitigation Guidance

9.1 Extension Passthrough Hardening

- Remove OID 2.5.29.32 (CertPolicies) from EnableEnrolleeRequestExtensionList unless specifically required.
- Remove OID 1.3.6.1.4.1.311.25.2 (Security Extension) from EnableEnrolleeRequestExtensionList.
- Remove OID 2.5.29.37 (EKU) from EnableEnrolleeRequestExtensionList to prevent EKU injection.
- Implement custom CA policy module validation to reject CSRs containing sensitive extensions.

9.2 Template Hardening

- Remove CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT from templates that do not require it.
- Add EKU constraints to all templates to prevent any-purpose certificate abuse.
- Upgrade V1 templates to V2 or higher where possible.
- Remove CT_FLAG_NO_SECURITY_EXTENSION from all templates.
- Restrict enrollment rights to minimum necessary security principals.

9.3 EOBO Hardening

- Configure EDITF_ENABLEAGENTRESTRICTION on all CAs with explicit EnrollmentAgentRights rules.
- Restrict CRA certificate issuance to specific, audited enrollment agent accounts.
- Consider upgrading EOBO templates to V3/V4 to architecturally disable EOBO.
- Monitor for unauthorized CRA certificate requests.

9.4 Vendor Recommendations

- Microsoft should update the CA to reject CSR-supplied Security Extensions on all templates.
 - Microsoft should update the KDC to require the Security Extension when SCBE ≥ 1 .
 - Microsoft should update the KDC to reject certificates with duplicate extensions.
 - Microsoft should provide clearer documentation on the security implications of EnableEnrolleeRequestExtensionList.
 - Microsoft should resolve the V3/V4 msPKI-RA-Application-Policies conflict to allow secure EOBO configuration.
-

Document Version: 2.1

Last Updated: October 2026

Validated on: Windows Server 2025 (Build 26100)