

```
~/adcs $ ./escalate --from esc18 --to esc22 █
```

[DEF CON 34]

Escalating all the way up via AD CS

Five privilege-escalation paths through Active Directory Certificate Services: from ESC18 to ESC22

Eric Labrador @elabrador

Daniel Monzón @stark0de





Eric Labrador

security researcher

red teamer

- > Offensive-security researcher, 6+ years breaking into networks and applications for a living.
- > Red & Purple Team, web/API audits, Android & iOS mobile assessments, internal/external pentesting, physical intrusions and large-scale phishing.
- > Author of [ImagePanick](#), AFW + RCE chaining weak ImageMagick policies with Ghostscript SAFER bypasses, fired from a malicious SVG.

certs: OSCP · CRTE · CRT0 · BSCP · eWPTXv2 · OSWP

[elabrador.com](#)



Daniel Monzón

security researcher

red teamer

- > Offensive Security Engineer at Halborn, 6+ years across web hacking, code review, Active Directory (on-prem & hybrid), mobile pentests, thick-client, red teaming and OT environments.
- > Credited with multiple CVEs across open-source and commercial products. Prior speaker at RootedCON Málaga, DragonJARCON, hack0n and SecAdmin.
- > Currently focused on Web3 & financial-sector security, AI security and OS internals.



certs: OSCP · CRTP · OSWP · CREST CPSA · eMAPT · PACSP · CARTP

stark0de.com



Agenda

- 01 ESC18: CertPolicies Passthrough & AMA Escalation
- 02 ESC19: SID Injection via Security Extension
- 03 ESC20: KDC fallback
- 04 ESC21 / ESC22 - co-signature validation skip, EA restrictions
- 05 Defender playbook and wrap-up



Microsoft hardened the front door. We found five windows.

ESC18

CA CSR pass-through

vector
AMA OID injection

target
KDC PAC

Template Version
V1, V2, V3 & V4

ESC19

SID Injection via Security Extension

vector
SID Passthrough

target
KDC binding

Template Version
V1, V2, V3 & V4

ESC20

KDC SAN URL fallback

vector
Omit security ext

target
KDC fallback

Template Version
V1, V2, V3 & V4

ESC21

Co-Signature Validation Skip

vector
Default template

target
DCSync

Template Version
V1

ESC22

Phantom Enrollment Agent Restrictions

vector
Missing enforcement

target
EOBO unrestricted

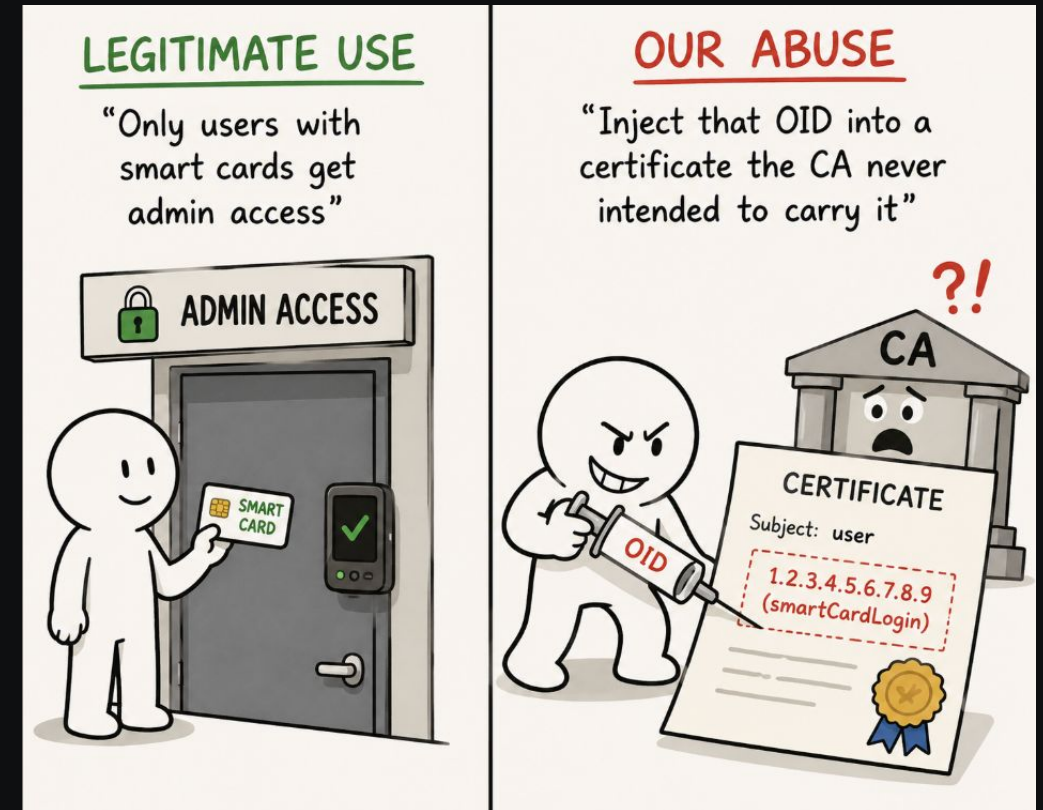
Template Version
V1 & V2



CA CSR Pass-Through AMA

Authentication Mechanism Assurance · Windows Server 2008 R2+

- > Admin creates an OID object in AD
`msPKI-Enterprise-Oid`
- > Sets `msDS-OIDToGroupLink` → points to a Universal Security group
- > Group is nested in Administrators / Domain Admins
- > When a certificate carries that OID, the KDC adds the group SID to the Kerberos PAC



What is different between ESC18 and ESC13?

ESC13

OID is pre-configured in the template.

```
msPKI-Certificate-Policy = [AMA_OID]
```

The admin placed the OID there intentionally.

ESC18

OID is injected at request time via the CSR.

The template has
`no certificate policy` configured at all.



PreRequisite: ESS

ENROLLEE_SUPPLIES_SUBJECT · msPKI-Certificate-Name-Flag & 0x1

Without ESS

CA strips CSR extensions even with an empty EKU.

ISSUANCE_POLICIES_FROM_REQUEST (0x20000)
conflicts with CSR-injected Certificate Policies.

With ESS

CA copies Certificate Policies from the CSR into the cert.

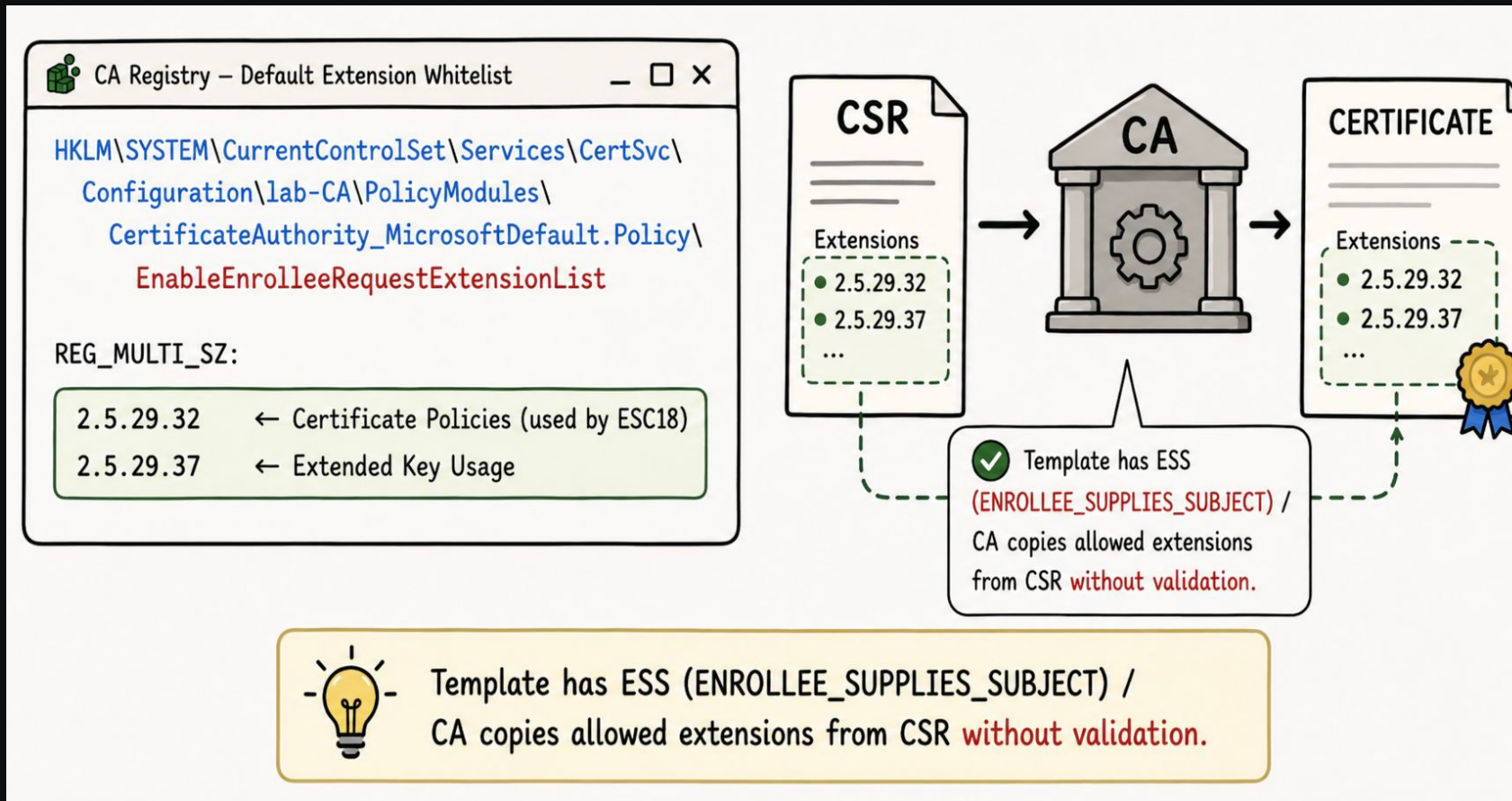
The AMA OID now appears in the issued certificate.

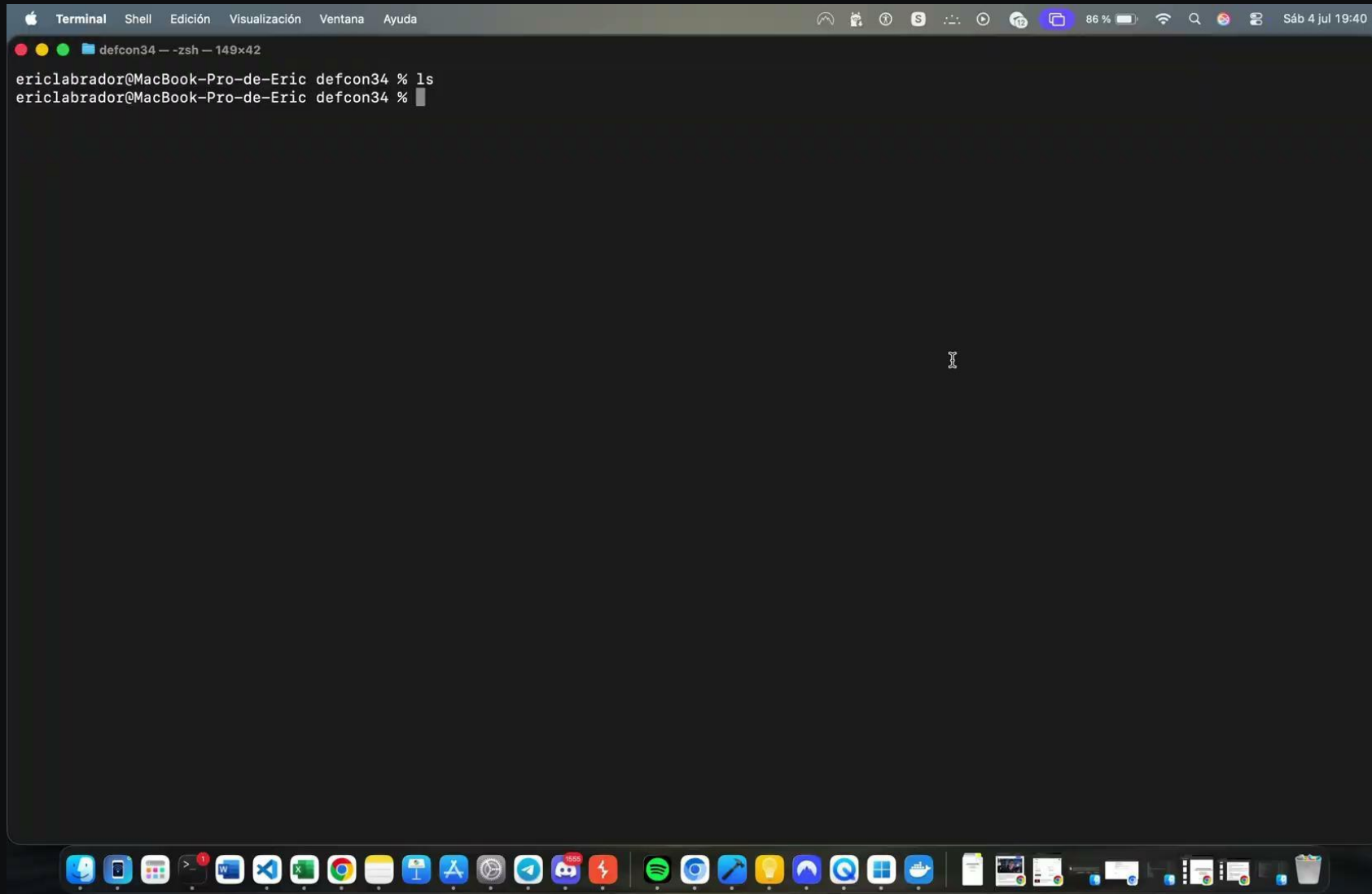
no ESS = no pass-through



Why this actually works?

`EnableEnrolleeRequestExtensionList`; the CA's default extension whitelist `Certificate Policies`





How we tracked down ESC18

- > Started from [KB5014754](#): Security Extension blocks SID impersonation. Asked: what other cert properties influence KDC authorization?
- > Found [AMA](#): a Certificate Policies OID linked to a Universal Security group → KDC adds the group SID to the Kerberos PAC. No impersonation needed.
- > Checked whether `Certificate Policies` can be `injected` via the CSR. Found `EnableEnrolleeRequestExtensionList` ; the CA's default whitelist includes `2.5.29.32`.
- > Tested on [ESS](#) template: CSR with `AMA` `OID` → CA copies it without validation → issued cert triggers PAC group injection → Domain Admin.



ESC18 Impact

- > Bypasses SCBE=2 without impersonation
- > The AMA OID looks legitimate, a valid AD object, no signature rule fires.
- > Exploits a legitimate Windows feature (AMA)
- > CA extension whitelist includes 2.5.29.32 by default
- > Requires: ESS template + AMA OID linked to privileged group
- > Self-Auth → SCBE validation passes → Group Injection via PAC → DCSync

**No impersonation. No SID mismatch.
Just privilege injection.**



How to defend from ESC18

- › Log the full CSR extension list, flag `2.5.29.32` in requests.
- › Restrict `EnableEnrolleeRequestExtensionList` which drop `2.5.29.32` from the default whitelist unless explicitly needed.
- › Enforce ECU on all templates, treat templates with no ECU as misconfigured and remediate.
- › Disable ESS on sensitive templates, no `ENROLLEE_SUPPLIES_SUBJECT` means no pass-through.



ESC19: SID Injection via Security Extension Passthrough

KB5014754

Maybe Microsoft



AD CS
is fixed.



How KB5014754 is supposed to work

- > SID embedded in the cert must match the authenticated account
- > ESC1 / ESC6 should be dead
- > Security Extension on all certs , `OID 1.3.6.1.4.1.311.25.2`

Normal ESS issuance

1. `user.alice` submits a CSR
2. CA looks up `alice` → SID -1105
3. CA writes Security Extension SID -1105
4. KDC: `cert SID == AD SID`
5. `PASS TGT issued as alice`

ESC1 attempt (-upn Administrador)

1. `alice` submits CSR, `UPN=Administrador`
2. CA still looks up `alice` → SID -1105
3. CA writes Security Extension SID -1105
4. KDC: `-1105 ≠ -500 (Administrador)`
5. `FAIL authentication rejected`



PreRequisites

- > ESS `ENROLLEE_SUPPLIES_SUBJECT`: lets the requester supply a UPN in the CSR
- > A template with a `PKINIT-Compatible` EKU (e.g., Client Authentication, Smart Card Logon) or `no` EKU
- > `EnableEnrolleeRequestExtensionList` allowing the `Security Extension` OID.



The SID Passthrough Attack

ESS + no-EKU template

1 · Attacker's CSR

UPN = Administrador@lab.local
SAN URL SID = -1105 (alice)
INJECTED Security Extension
(OID ...311.25.2) SID =
-500

2 · CA processes it (ESS template)

ESS template → CA skips SID lookup
CSR Security Extension passes
through as-is
No independent AD SID validation

Certificate gets attacker's SID →
-500 , TGT issued as Administrator

3 · Certificate has attacker's Security Extension

Security Ext SID =
S-1-5-...-500 (admin)
CA trusts CSR on ESS templates
– no SID lookup, no validation.

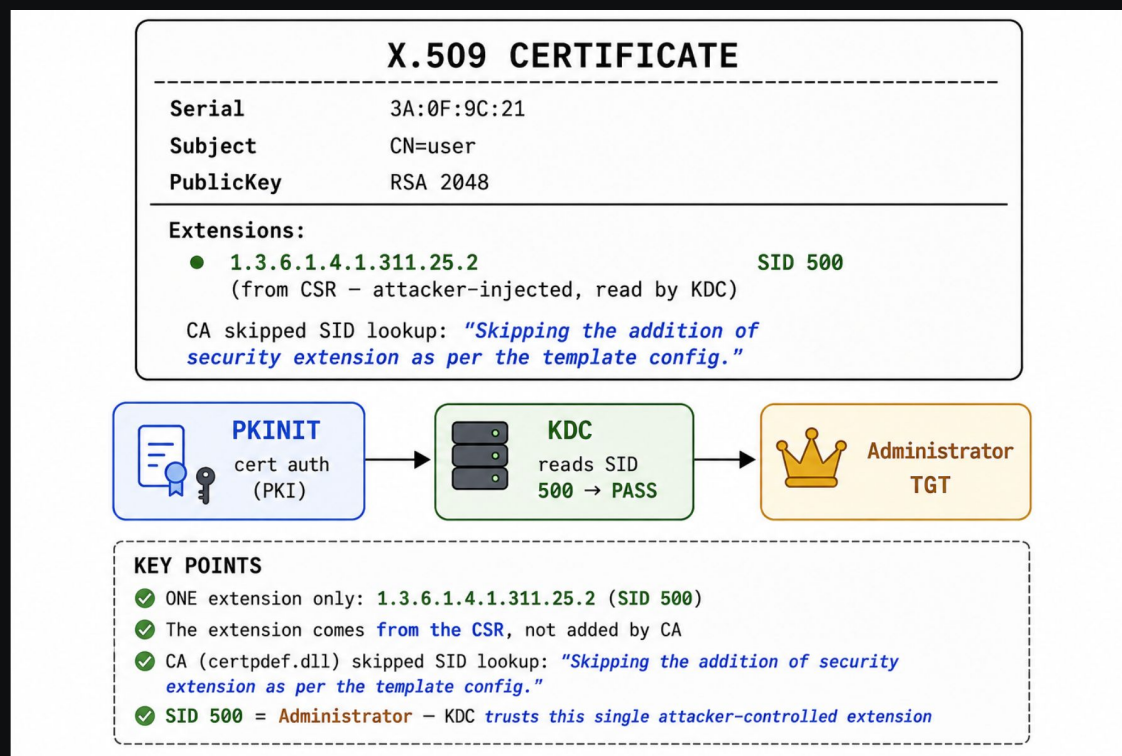
4 · KDC reads Security
Extension → SID -500 →
validates → PASS

5 · TGT issued as
Administrator



The CA never validates the SID

- > On ESS templates, the CA does not perform independent AD SID lookup.
- > The CSR's Security Extension passes through as-is into the issued cert.
- > The attacker duplicates the CA's SID binding role, one extension, attacker-controlled.



Impact

- > Direct bypass of KB5014754 on ESS templates
- > Exploits CA's lack of SID validation on ESS templates
- > Pure impersonation. No AMA, no EOBO, no chain

One Request → Domain Admin.



How to defend from ESC19

- > Remove `ESS` from templates with no `EKU`
- > Enforce explicit `EKU` on all templates, `no EKU = no trust boundary`.
- > Monitor `Event 4887` for `CSRs` that include the `Security Extension OID`, only the CA should populate it.
- > Remove the `Security Extension OID` from the allowlist
- > Check out 0xmaz's research (he found a very cool way to do this without needing the value in the allowlist)



Undocumented KDC Fallback to SAN URL SID

ESS template · KDC fallback · Full impersonation

- > On templates with `ESS`, the CA trusts the CSR for the Security Extension.
- > If the CSR omits the Security Extension, the CA does not add its own.
- > The certificate is issued with `no Security Extension at all`.
- > The KDC falls back to reading the SID from the `SAN URL` field , attacker-controlled.
- > This fallback is `undocumented`. SCBE=2 does not prevent it.

KB5014754's other blind spot.



How the CA handles the Security Extension

CA adds Security Extension (safe)

```
user.alice submits CSR
CA looks up alice in AD → SID -1105
CA writes Security Extension
with real objectSid

CSR contents IGNORED
for Security Extension
```

CA defers to CSR (vulnerable)

```
CSR has Security Ext? → CA copies it
CSR omits Security Ext?
→ CA does NOT add its own

Certificate has NO Security
Extension.

The CA trusts the CSR entirely.
```

ESS = the CA defers to the requester.



What happens with no Security Extension?

`StrongCertificateBindingEnforcement = 2`

Expected behavior

No Security Extension in certificate

- `reject certificate`
- `authentication fails`

The whole point of KB5014754.

Actual behavior

No Security Extension in certificate

- KDC falls back to SAN URL SID
`tag:microsoft.com,...:sid:<SID>`

Attacker controls the SAN URL SID.
Validation passes. TGT issued.

This fallback is UNDOCUMENTED.



Two independent bypasses of the same control

ESC19 , targets the CA

Supply the Security Extension via the CSR.

CA passes through CSR's ext on ESS.
KDC reads attacker-controlled SID.

Bug: CA skips SID validation on ESS.

ESC20 , targets the KDC

OMIT the Security Extension entirely from the CSR.

CA doesn't add its own (ESS).
KDC falls back to SAN URL SID.

Bug: undocumented fallback.

Patching one does NOT fix the other. Independent bugs.

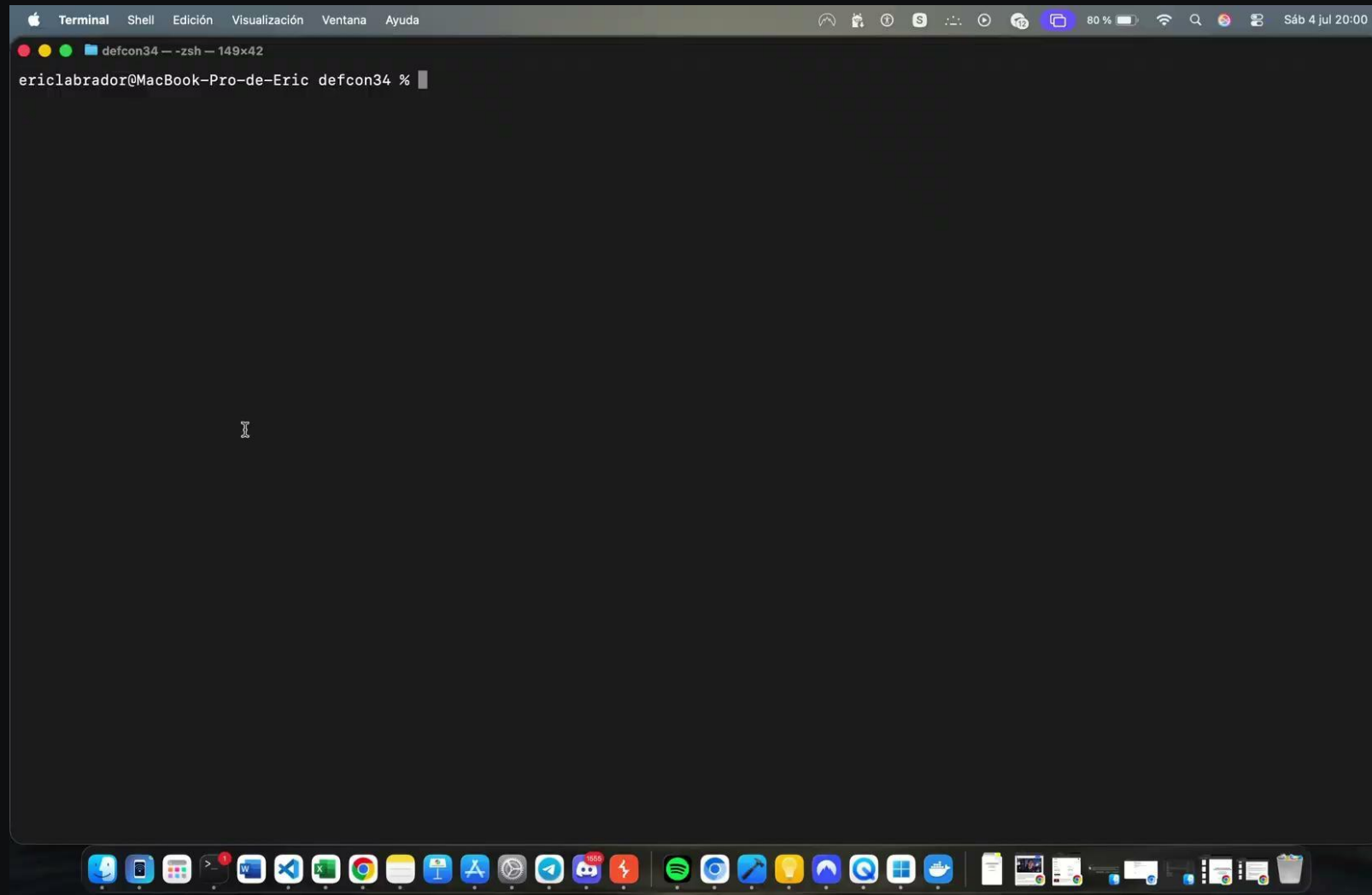


Impact

- > Undocumented KDC fallback: `SAN URL SID` used when Security Extension is absent.
- > ESS templates: CA does not add `Security Extension` if the `CSR` omits it.
- > Independent from `ESC19`, different component, different bug.
- > Requires: `ESS` template.
- > `SCBE=2` does NOT prevent this, the fallback ignores the enforcement setting.

Certificate has ZERO Security Extensions. KDC still authenticates.





How to defend from ESC20

- Remove `ESS` from templates that do not require it.
- Review all templates, custom templates may have ESS enabled.
- Microsoft should remove the undocumented SAN URL SID fallback from the KDC.
- Monitor Event `4768` for PKINIT authentication with certificates missing the Security Extension.

Removing ESS from templates kills ESC20 in one change.



Three primitives, three bypass mechanisms

Common thread: ENROLLEE_SUPPLIES_SUBJECT = the CA trusts the CSR.

- > ESC18 , AMA OID injection via CSR → privilege injection through the PAC.
- > ESC19 , SID Passthrough via Security Extension → KDC reads attacker-controlled SID.
- > ESC20 , Omit Security Extension → KDC falls back to SAN URL SID.

Now let's look at the EOBO path.



Enrollment On Behalf Of (EOBO)

The Enrollment Agent pattern , legitimate smart-card provisioning

- > A helpdesk operator with an `Enrollment Agent (EA)` certificate can request certs for other users.
- > Legitimate use case: provisioning smart cards without the employee touching the CA directly.
- > If an attacker obtains an EA cert, they can request certificates `as anyone`.
- > Enrollment agent restrictions are supposed to limit who can be targeted.

Enroll On Behalf Of (EOBO): A proxy enrollment process in which one user, typically an administrator, enrolls for a `certificate` for a second user by using the administrator credentials.

2.16 msPKI-Template-Schema-Version Attribute

✦ Summarize this article for me

The msPKI-Template-Schema-Version attribute specifies the schema version of the templates. The allowed values are 1, 2, 3, and 4.<17> For schema details of this `attribute`, see [MS-ADA2] section 2.625.

`EA cert + permissive template = impersonation.`



V1 templates , the CA's blind spot for EOBO

- > V1 Templates (Windows 2000 era) lack the msPKI-RA-Signature attribute entirely.
- > On V2+ RA-Signature controls whether EOBO is allowed and which co-signer cert is required.
- > V1 Templates skip this check, the CA's EOBO code path never reads the attribute.
- > Setting RA-Signature on V1 via LDAP succeeds, but the CA silently ignores it. False hardening.

```
Administrator: Windows Powe
PS C:\WINDOWS\system32> Get-ADObject "CN=Demo-ESC21,CN=Certificate Templates,CN=Public Key Services,CN=Services,CN=Configuration,DC=lab,DC=local" -Properties msPKI-RA-Signature | Select-Object Name, msPKI-RA-Signature

Name          msPKI-RA-Signature
----          -
Demo-ESC21    1

PS C:\WINDOWS\system32>
```

The Asymmetric Behavior

Co-signer: CRA cert

EOBO on V1 template

Template: Demo-ESC21 (V1)

Co-signer: CRA cert

CA skips RA-Sig attribute – EOBO
allowed without config ✓

Certificate issued ✓

Authentication SUCCEEDS ✓

EOBO on V2 template

Template: V2 equivalent

Co-signer: same cert

CERTSRV_E_BAD_RENEWAL_SUBJECT

Request REJECTED ✗

"Must require at least one
signature"

Same co-signer cert. Same EOBO. Different template version.
Different result.



Why V1 Is Different

- > V1 schema predates msPKI-RA-Signature, the attribute does not exist in V1.
- > The CA has separate code paths for V1 and V2+. V1 path skips RA-Sig entirely.
- > On V1, the CA never reads msPKI-RA-Signature. EOBO is implicitly allowed.
- > Unlike ESC3: no template EOBO configuration needed. CRA cert still required.
- > Admins cannot harden V1 against EOBO, the only fix is upgrading to V2+.

V1 = no RA-Sig = no EOBO enforcement.
On V1, EOBO is always open to any CRA holder.



The Attack Chain

- > Step 1: Obtain a CRA cert (Enrollment Agent certificate).
- > Step 2: EOBO as administrator on a V1 template, using the CRA cert as co-signer.
- > Step 3: Authenticate with PKINIT → TGT as administrator → admin hash.

Requires CRA cert + V1 template. Two templates, two requests.

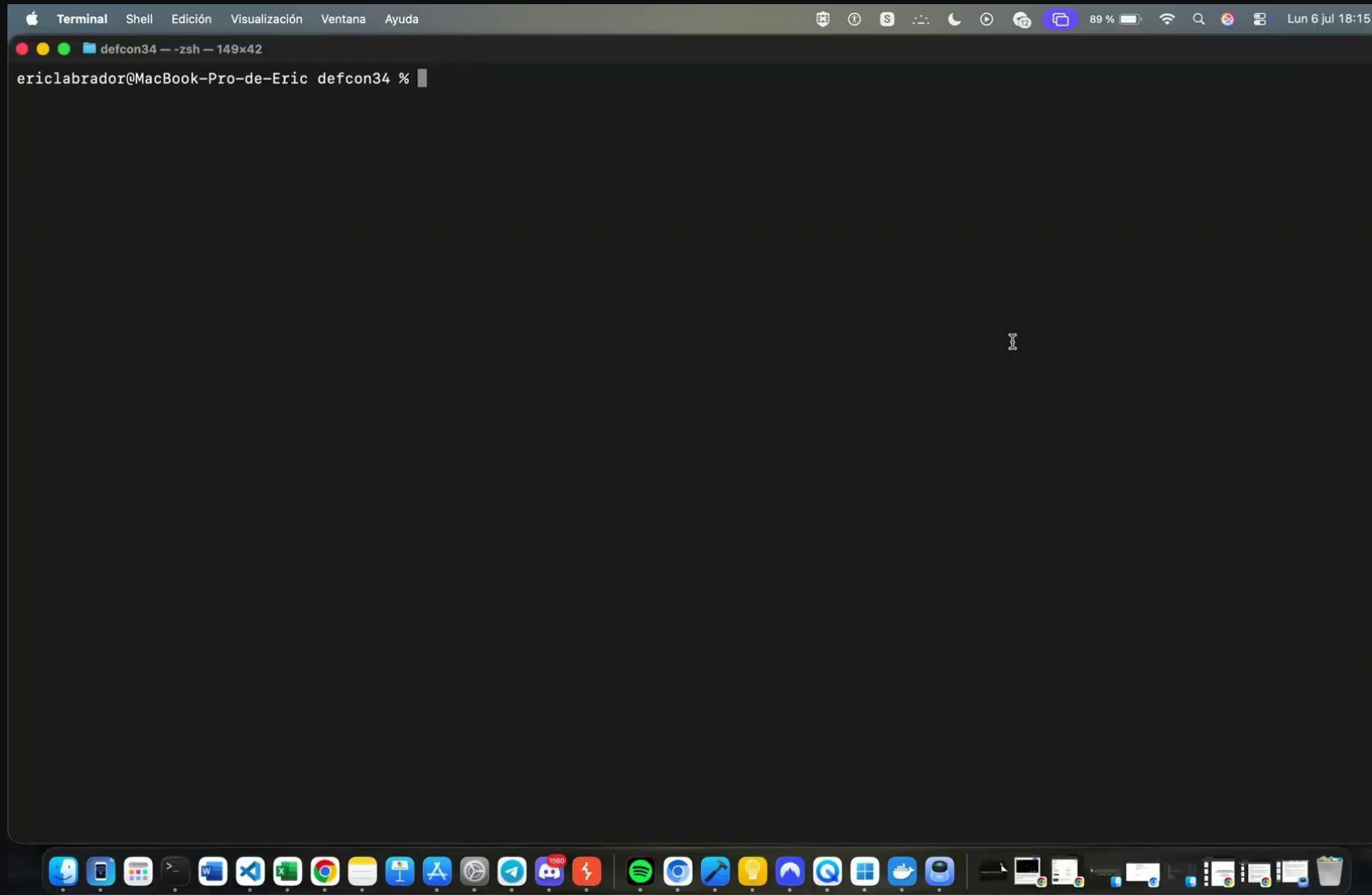
CRA cert → EOBO on V1 → Domain Admin.



ESC21 Impact

- > V1 templates implicitly allow EOBO — no RA-Sig configuration needed.
- > Unlike ESC3, no template needs to be configured for EOBO. Any V1 with enrollment rights is vulnerable.
- > False hardening: setting RA-Sig=1 on V1 via LDAP succeeds but the CA ignores it.
- > V1 templates with user enrollment rights are common in enterprise environments.
- > V1 templates exist by default on every Enterprise CA (User, Machine, DomainController).





Phantom Enrollment Agent Restrictions

CA configuration · Missing enforcement

- > Then configure restrictions via the `certsrv.msc` GUI (which EA certs, which targets, which templates)
- > Most admins run the `certutil` command, see it succeed, and stop there. They never open the GUI.
- > On Server 2025 the flag name is `not recognized` because the command fails outright.

```
Administrator: Windows Powe
PS C:\WINDOWS\system32> certutil -setreg policy\EditFlags +EDITF_ENABLEAGENTRESTRICTION
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CertSvc\Configuration\lab-CA\PolicyModules\Certific
ity_MicrosoftDefault.Policy\EditFlags:

Old Value:
EditFlags REG_DWORD = 11194e (1120590)
EDITF_REQUESTEXTENSIONLIST -- 2
EDITF_DISABLEEXTENSIONLIST -- 4
EDITF_ADDOLDKEYUSAGE -- 8
EDITF_BASICCONSTRAINTSCRITICAL -- 40 (64)
EDITF_ENABLEAKIKEYID -- 100 (256)
EDITF_ENABLEAKIISSUENAME -- 800 (2048)
EDITF_ENABLEAKIISSUERSERIAL -- 1000 (4096)
EDITF_ENABLEDEFAULTSMIME -- 10000 (65536)
EDITF_ENABLECHASECLIENTDC -- 100000 (1048576)
CertUtil: -setreg command FAILED: 0x8007000d (WIN32: 13 ERROR_INVALID_DATA)
CertUtil: The data is invalid.
PS C:\WINDOWS\system32>
```

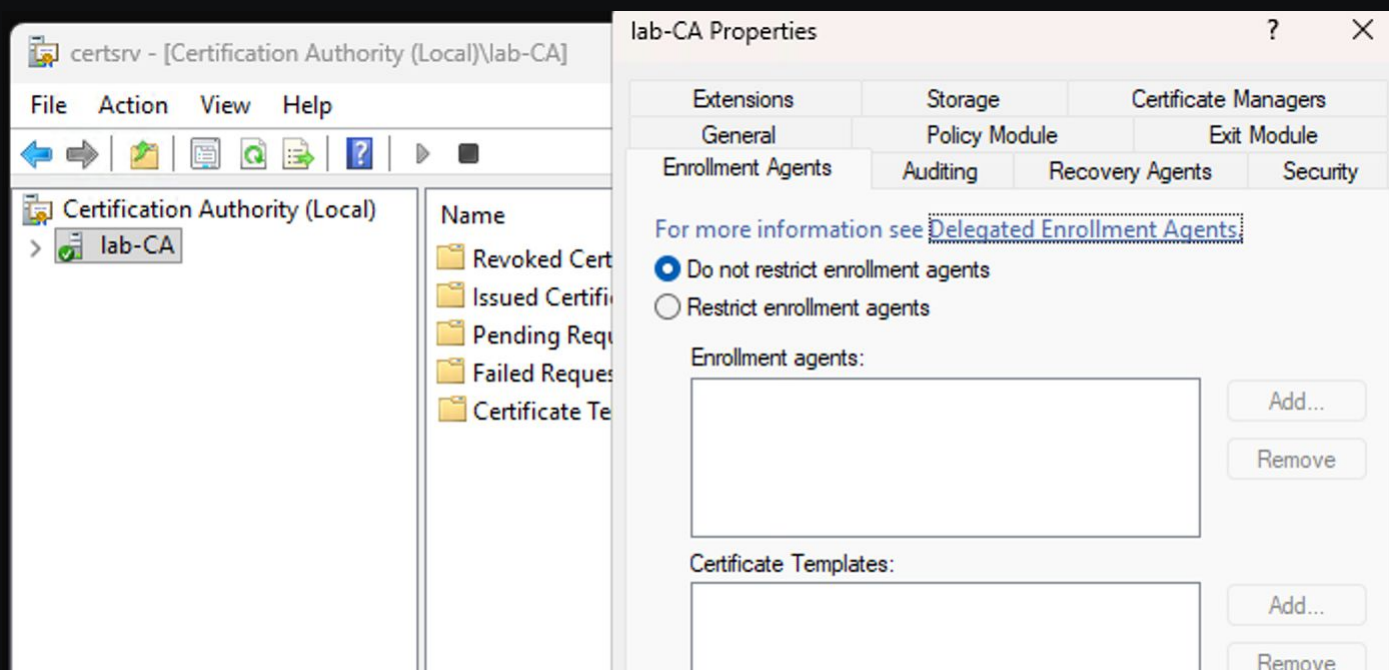
- Use Enrollment Agent restrictions on the Certificate Authority level. For example, you might want to restrict which users are allowed to act as an Enrollment Agent, and which templates can be requested.

The Flag That Never Was

Windows Server 2025, Build 26100:

- > `certutil -setreg policy\EditFlags +EDITF_ENABLEAGENTRESTRICTION` → `ERROR_INVALID_DATA`
- > The flag name `EDITF_ENABLEAGENTRESTRICTION` is not recognized by Server 2025's certutil.

CA Properties →
Enrollment Agents tab
(Default Config)



We Tested It

Validation on Server 2025:

- > Step 1: Verified `EnrollmentAgentRights` is absent in registry (default state: never configured).
- > Step 2: EOBO as Administrator; certificate issued, authentication succeeded.
- > Step 3: Extracted admin NT hash. Full impersonation with zero enforcement.

No registry value = zero enforcement. EOBO
unrestricted.

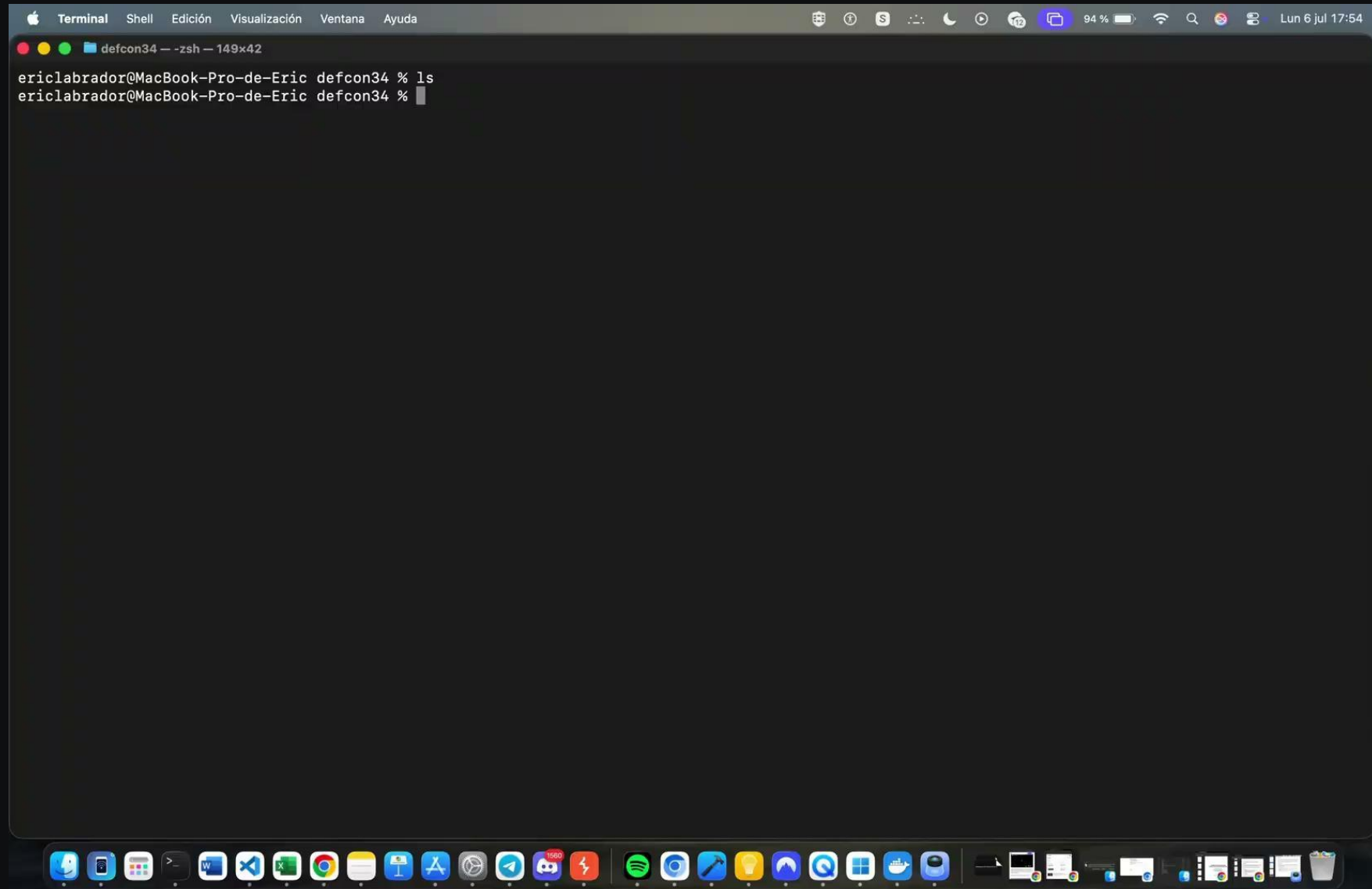


The Combination Is Devastating

- > `ESC21`: V1 skips RA-Sig → EOBO allowed without configuration. CRA cert required.
- > `ESC22`: Enrollment agent restrictions are not configured by default.
- > `Combined`: V1 templates + no restrictions = unrestricted EOBO = Domain Admin

"The door has a lock. The lock has no pins."





How to defend from ESC21 & ESC22

- > Upgrade all `V1` templates to `V2+` . Set `msPKI-Template-Schema-Version` ≥ 2
- > Add explicit `EKU` to all templates. `NoEKU` certs should not exist in production.
- > Do NOT rely on setting `msPKI-RA-Signature` on `V1` since the CA ignores it.
- > Configure `Enrollment Agent Restrictions` via `Restrictions` `certsrv.msc` → `Enrollment Agents` tab. Verify `EnrollmentAgentRights` exists in registry.

Audit: any V1 + low-priv enrollment = ESC21 vulnerable.



Detection & defense

- > ESC18 , Event 4887: Certificate Policies with AMA OID not configured on the template.
- > ESC19 , Event 4887: CSR includes Security Extension OID (should be CA-only).
- > ESC20 , Event 4768: PKINIT auth with certificate missing the Security Extension.
- > ESC21 , Event 4887: EOBO against V1 template where msPKI-RA-Signature is not configured.
- > ESC22 , Registry audit: EnrollmentAgentRights key empty or absent.



The ESS Kill Switch

ENROLLEE_SUPPLIES_SUBJECT is the root cause for ESC18, ESC19, and ESC20.

- Remove `ESS (0x1)` from `msPKI-Certificate-Name-Flag` → blocks ESC18, ESC19, ESC20 simultaneously.
- If ESS is required: add explicit EKU (e.g., Client Auth only) and remove 2.5.29.32 from `EnableEnrolleeRequestExtensionList` to prevent `CertPolicies Injection`.
- Restrict enrollment to specific groups , not `Authenticated Users`.
- Remove `2.5.29.32` from `EnableEnrolleeRequestExtensionList` if AMA is not used.

Removing ESS kills ESC18-20 in one change per template.



Defender Quick Reference

- > **ESC18** Remove ESS or remove 2.5.29.32 from CA whitelist. Remove AMA OID links.
- > **ESC19** Remove ESS from NoEKU templates. MS should enforce independent SID validation.
- > **ESC20** ESC20 Remove ESS from templates that don't require it. MS should remove SAN URL SID fallback from KDC.
- > **ESC21** Upgrade V1 to V2+. Add EKU constraints. Do not trust RA-Sig on V1.
- > **ESC22** Server 2025: compensating controls only. Older: verify rules in registry.



Certified Re-Pwned

All validated: Windows Server 2025 | SCBE=2 | Security Extension ON

- > ESC18 AMA OID injection via CSR → PAC group injection
- > ESC19 Security Extension SID passthrough → SID impersonation
- > ESC20 KDC SAN URL SID fallback → SID impersonation
- > ESC21 ESC21 V1 co-signature skip → EOBO → Domain Admin
- > ESC22 EA restrictions don't enforce → EOBO unrestricted

Microsoft hardened the front door. We found five windows.



Thank You, DEF CON.



Eric Labrador @elabrador

Daniel Monzón @stark0de

