ATTACKING AND DEFENDING ACTIVE DIRECTORY WITH OPENSSL

OPENSSL AND CERTIFCATE CHAOS

C:\>WHOAMI

Darryl G Baker

Principal Solutions Architect @ Netwrix

Creator of AD Hacking Village

Army -> IT -> AD/IAM -> ITDR

@dfirdeferred

dbaker-cissp-ceh

AGENDA

- AD CS & PKI overview
- OpenSSL in Windows security
- ESC1–ESC15 vulnerabilities
- Red vs. Blue OpenSSL techniques

ADCS—THE FOUNDATION OF WINDOWS PKI

- **Core purpose:** Issues and manages digital certificates that provide identity, authentication, encryption, and secure access (e.g. smart cards, Wi-Fi, VPN, SSO, EFS).
- **Trust anchor:** Built on a hierarchical PKI with Root and Subordinate Certification Authorities (CAs) that define the trust chain.
- **Integration:** Tightly integrated with Active Directory for auto-enrollment, Group Policy, and certificate mapping to users, computers, and services.
- **Attack surface:** Vulnerable certificate templates, misconfigured CA permissions, web enrollment endpoints, key protection, and weak certificate-to-account mappings.

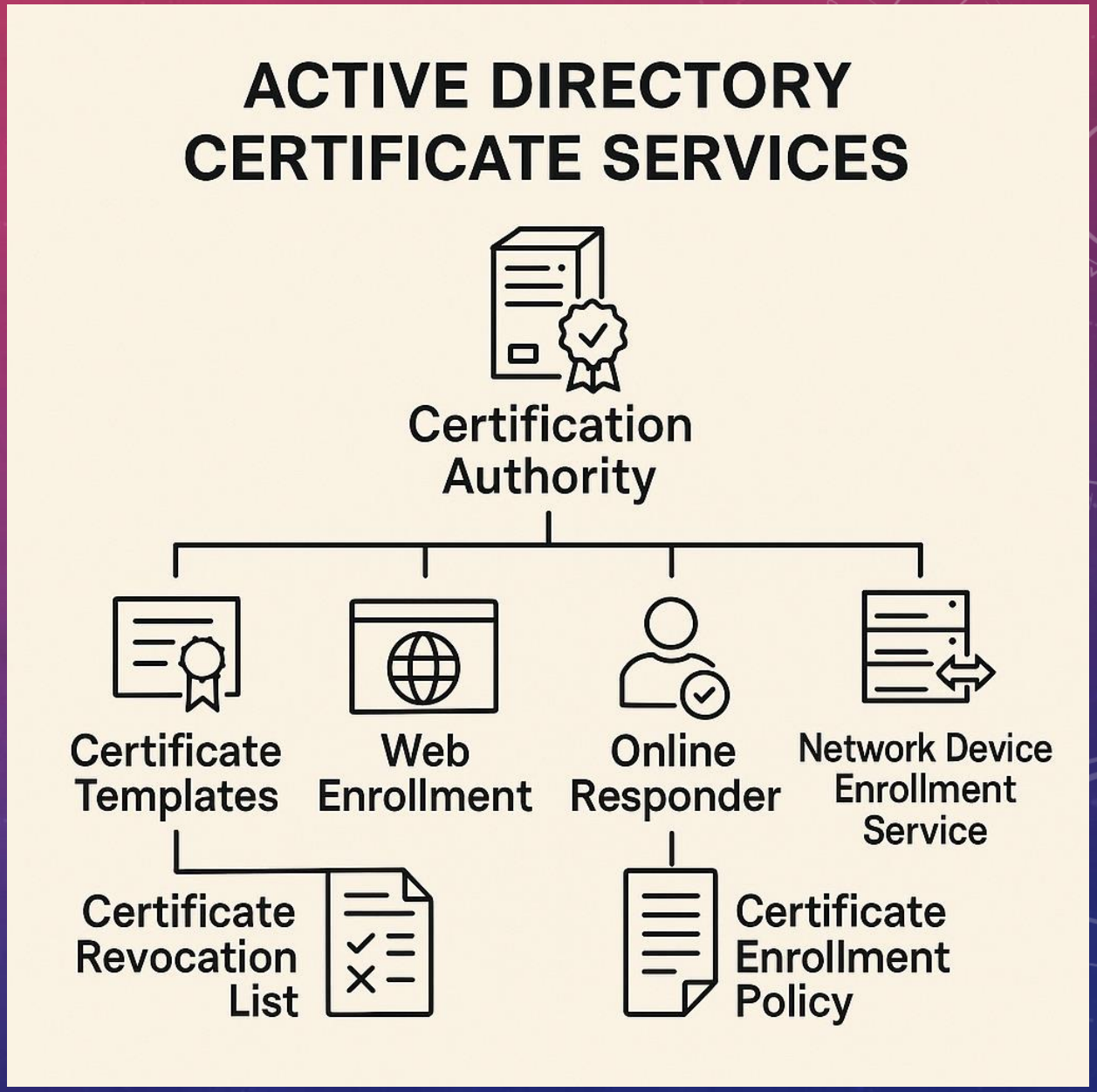
ADCS- Configurations

Enterprise Mode:

- CA is in a Trust relationship with ADDS
- Manages certificates via group membership
- Supports auto-enrollment to simplify the process

Standalone Mode:

- CA is not integrated with ADDS
- Stores user data in the application directory
- Enrollment requests are submitted manually



OPENSSL—SWISS ARMY KNIFE FOR PKI

- Open-source cryptographic toolkit
- Generate, parse, audit, convert certificates and keys
- Available cross-platform
- Key for forensic review

THE CERTIFIED PRE-OWNED FRAMEWORK (ESC1–ESC15)

- SpecterOps originally documented ESC1 through ESC8 in Certified Pre-Owned.
- Later community / offensive research has extended this set (e.g. ESC9 to ESC14) and more recent “ESC15” (aka EKUwu) addressing a legacy template bug
- Series of template, CA, and policy misconfigurations
- Enables privilege escalation, lateral movement, persistence
- Each ESC = a specific exploit path

ESCALATION PATHS 1 - 15

ESC1	Pass-the-Certificate / Golden Cert lite
ESC2	Swiss Army Knife Cert (Any Purpose EKU)
ESC3	Agent Smith (Request Agent chaining)
ESC4	Template Jacking
ESC5	Golden Certificate (CA takeover)
ESC6	Forged SAN (AltName abuse)
ESC7	Rubber Stamp (Manager Approval abuse)
ESC8	PetitPotam → CertSrv (NTLM relay)
ESC9	Naked Certificate (no security ext)
ESC10	UPN Spoof (weak mapping)
ESC11	DCOM Relay to CA
ESC12	Shell-to-Sign (HSM abuse)
ESC13	OID → DA Membership
ESC14	Wildcard Mapping Attack
ESC15	EKUwu (Application Policy override)

ESC1—ENROLLEE SUPPLIES SUBJECT (IDENTITY IMPERSONATION VIA SAN INJECTION)

The certificate template has the “Enrollee supplies subject” flag (object’s *mspki-certificate-name-flag property* has CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT) enabled

The template includes an EKU (Extended Key Usage) that allows authentication (Client Authentication, Smart Card Logon, or “Any Purpose”)

Low/medium privilege (e.g. Domain Users) has Enroll permissions on that template

No mitigating constraints (e.g. requiring manager approval or authorized signature) blocking arbitrary SANs

ESC1—OFFENSIVE WORKFLOW

Attack mechanism:

- The attacker requests a certificate using that template, injecting a SAN / UPN that belongs to a privileged account (e.g. Administrator@domain.local).
- The CA (due to weak template) issues a cert with that SAN.
- The attacker can then use that certificate to authenticate (via Kerberos PKINIT or Schannel) as the target account.

Mitigations:

- Disable the “Supply subject in request” feature unless absolutely necessary.
- Require manager approval or authorized signatures for certificate issuance.
- Restrict who has Enroll permission on templates (avoid broad groups like Domain Users).
- Monitor certificate issuance events (Event IDs 4886 / 4887) and look for unusual SANs / forged UPNs.

ESC1—USING OPENSSL

Red Team- Craft CSR with -subj or custom SAN:

```
openssl req -new -newkey rsa:2048 -keyout attacker.key -out esc1.csr \
  -subj "/CN=Administrator" \
  -addext
“subjectAltName=otherName:1.3.6.1.4.1.311.20.2.3;UTF8:Administrator@domain.local"
```

Blue team- Inspect issued certs for suspicious SANs/subjects:

```
openssl x509 -in suspicious.crt -text -noout | grep -i "Subject Alternative Name"
```

DEMO

ESC2—"ANY PURPOSE" EKU

CERTIFICATE IS VALID FOR ANY SCENARIO (AUTHENTICATION, ENCRYPTION, CODE SIGNING, ETC.)

- Template’s EKU “any purpose” (2.5.29.37.0) used or No EKU’s (a subordinate CA certificate)

- Certificate is valid for any scenario (authentication, encryption, code signing, etc.)
- Attacker has Enroll permission
- No constraints that restrict the usage of the certificate

ESC2—OFFENSIVE WORKFLOW

Attack mechanism:
The certificate can be used for purposes beyond what the template was intended for (e.g. authentication or privilege escalation). It effectively becomes a “Swiss army knife” certificate.

Mitigations:
- Use strict EKU settings on templates, only permit the specific usages required.
- Do not include “Any Purpose” unless absolutely needed (and then only under tight control).
- Audit templates for overly permissive EKUs.

ESC2—USING OPENSSL

Red Team- Inspect cert template any purpose EKUs or No EKUs. If so, enroll the cert:

```
if openssl x509 -in swiss.crt -text -noout | grep -qEi "Any
Purpose|2\.5\.29\.37\.0"; then
    echo "⚠ Vulnerable: Any Purpose EKU present"
elif openssl x509 -in swiss.crt -text -noout | grep -q "Extended Key Usage";
then
    echo "EKU present but not Any Purpose"
else
    echo "⚠ Vulnerable: No EKU present (implicitly Any Purpose)"
fi
```

Blue team- Inspect issued certs for suspicious SANs/subjects:

- Enumerate templates with Any Purpose EKU.
- Validate EKU in issued certs with OpenSSL to confirm policy adherence.

ESC3—CHAINING VIA CERTIFICATE REQUEST AGENT

CROSS-TEMPLATE EXPLOITATION

- Template configured with Certificate Request Agent rights (OID 1.3.6.1.4.1.311.20.2.1). Allows for requesting certificates on behalf of another principal
- The attacker has Enroll on that agent template
- Manager approval is disabled.
- No authorized signatures are required.
- The attacker uses that agent cert to enroll for more powerful templates (e.g. ones used for authentication)


YOU GET A CERT! YOU GET A CERT!
YOU GET A CERT!
imgflip.com

ESC3—OFFENSIVE WORKFLOW

Attack mechanism:

- The attacker first obtains a Certificate with the Request Agent EKU (OID 1.3.6.1.4.1.311.20.2.1).
- Using that, they can enroll for other templates (even ones they couldn't previously access), eventually obtaining a certificate that lets them impersonate a privileged account.

Mitigations:

- Avoid delegating request-agent privileges lightly.
- Restrict which templates have agent rights.
- Monitor issuance of agent certificates and chaining behavior.

ESC3—USING OPENSSL

Red Team- After a certificate with the Certificate Request Agent EKU is obtained, sign another CSR with it.

```
openssl ca -cert agent.pem -keyfile agent.key -in victim.csr -out victim.crt
```

Blue team:

- Inspect cert chain to see “Certificate Request Agent” EKU
- Audit who has agent enrollment rights.

ESC6—SAN ATTRIBUTE ABUSE

ABUSE OF EDITF_ATTRIBUTESUBJECTALTNAME2 FLAG

- Requires the CA to allow user-specified alternative names via the **EDITF_ATTRIBUTESUBJECTALTNAME2** flag. If this flag is set on the CA, any request (including when the subject is built from Active Directory®) can have user defined values in the subject alternative name.

- Attacker has Enroll on template

ESC6- OFFENSIVE WORKFLOW

Attack mechanism:

Attacker crafts CSR / request to exploit how the template handles SAN extension (e.g. injecting disallowed SANs). They might bypass restrictions or validation logic and get issued a cert with an escalated SAN (e.g. impersonating accounts).

Mitigations:

Ensure templates do not allow dangerous SAN behavior unless carefully controlled.

Where possible, disable or restrict EDITF_ATTRIBUTESUBJECTALTNAME2 usage.

Use newer Windows versions / ADCS versions that have stricter behavior.

ESC6—USING OPENSSL

Red Team- Craft CSR with SAN extension (bypassing intended policy)

```
openssl req -new -key attacker.key -out esc6.csr \
  -subj "/CN=NormalUser" \
  -addext "subjectAltName=otherName:1.3.6.1.4.1.311.20.2.3;UTF8:Administrator@corp.local"
```

Blue team:

- Use OpenSSL to decode SANs, confirm they match account identity

DEMO

ESC9—NO SECURITY EXTENSION

MISSING TEMPLATE SECURITY ENFORCEMENTS

- Template configured without requiring security extensions (e.g. missing msPKI-Enrollment-Flag bits like CT_FLAG_NO_SECURITY_EXTENSION
- Attacker has Enroll rights

Template omits security extension or lacks proper extensions that enforce validation, allowing bypass of usual checks.

ESC9- OFFENSIVE WORKFLOW

Attack mechanism:

- Because the template doesn’t enforce certain security checks (like verifying name or subject constraints), the attacker can bypass restrictions on subject name, key usage, etc., thereby escalating.

Mitigations:

- Use templates that enforce security extensions.
- Enable CT_FLAG_NO_SECURITY_EXTENSION when appropriate to force inclusion of security extension logic.
- Audit templates for those missing essential extensions.

ESC9—USING OPENSSL

Red Team

- Check Certficate for missing expected security extensions. Enroll Certificate is extension is missing.

```
openssl x509 -in naked.crt -text -noout | grep -A3 "X509v3 extensions"
```

Blue team:

- Detect certs without Subject Key Identifier / constraints

ESC13—ISSUANCE POLICY MAPPED TO A PRIVILEGED AD GROUP (OID → GROUP LINKING)

- The template specifies an Issuance Policy
- This policy is linked to a privileged group via msDS-OIDToGroupLink
- Attacker has Enroll on the template

ESC13—OFFENSIVE WORKFLOW

Attack mechanism:
When the attacker enrolls for the certificate, the certificate contains the linked Issuance Policy OID. Upon authentication (PKINIT), the KDC sees the OID, looks up the linked group, and maps the user's Kerberos ticket with the privileges of that AD group. Thus, the attacker gains those privileges without needing to impersonate that account explicitly.

Mitigations:
- Avoid mapping Issuance Policy OIDs to privileged groups unless absolutely necessary and well controlled.
- Audit the OID → group links in the AD forest.
- Restrict templates that use issuance policy OIDs to trusted accounts.

ESC13—USING OPENSSL

Red Team

- CSR specifying issuance policy OID

```
openssl req -new -newkey rsa:2048 -nodes -keyout esc13.key -out esc13.csr \
  -subj "/CN=NormalUser" \
  -addext "certificatePolicies=1.2.3.4.5.6"
```

Blue team:

- Parse cert policies with OpenSSL

```
openssl x509 -in esc13.crt -text -noout | grep -A2 "Certificate Policies"
```

- Look for OID objects that have a group linked via powershell:

```
PS> Get-ADObject -LDAPFilter '(msDS-OIDToGroupLink=*)' -SearchBase "CN=OID,CN=Public Key
Services,CN=Services,CN=Configuration,DC=corp,DC=local" -Properties msDS-OIDToGroupLink |
    Select-Object Name, DistinguishedName, msDS-OIDToGroupLink
```

DEMO

ESC15—ARBITRARY APPLICATION POLICY

"EKUWU" BUG (APPLICATION POLICIES VS EKU CONFLICT)

- The template is a version 1 template
- Authorizes the SAN
- The attacker has Enroll permission on that template

- The attacker uses the Application Policy extension in the CSR to insert Policies similar to EKUs (e.g. Client Authentication) which take precedence over the template EKUs

ESC15—OFFENSIVE WORKFLOW

Attack mechanism:
When a vulnerable certificate template is used to issues a certificate, it may include the Application Policy OIDs supplied by the requester.

In Windows, these Application Policies take precedence over the template's defined EKUs during certificate validation. As a result, even if the template was only meant for something like Web Server Authentication, the attacker's injected Application Policy (for example, Client Authentication) is honored instead. This allows the attacker to use the certificate for authentication, impersonating privileged accounts or gaining logon capabilities that were never intended.

Mitigations:
- Avoid using **version 1** templates for any purpose that allows enrollment by non-trusted principals.
- Always clone default templates (which upgrades to version 2) rather than using original version 1 templates.
- Audit templates for those that are version 1 and have "Supply subject" flags.
- After patching (CVE-2024-49019), apply vendor fixes and configuration recommendations

ESC15—USING OPENSSL

Red Team

- Create CSR with Application Policies for (clientAuth)

```
openssl genrsa -out esc15.key 2048
openssl req -new -key esc15.key -out esc15.csr \
  -subj "/CN=NormalUser" \
  -addext "1.3.6.1.4.1.311.21.10=ASN1:SEQUENCE:OID:1.3.6.1.5.5.7.3.2"
```

- Forces Client Authentication EKU even if template didn’t intend it.

Blue team:

- Detect Application Policies in the issued certificate

```
openssl x509 -in suspicious.crt -text -noout | grep -A2 "1.3.6.1.4.1.311.21.10"
```

DEMO

SUMMARY: OPENSSL ROLES IN ADCS OPERATIONS

Red team:

•Craft malicious CSRs with forged SANs, EKUs, OIDs.
•Inspect issued certs for exploitable EKUs/mappings.
•Chain exploitation (ESC1, ESC2, ESC3).

Blue team:

•Review issued certs for policy misconfigurations.
•Validate EKUs, SANs, issuance policies, mapping fields.
•Compare cert contents against intended template settings.
•Audit logs + OpenSSL parsing to spot anomalies.

THANK YOU!

SOMEONE@EXAMPLE.COM