

Active Directory Certificate Service

Realisatiedocument

Daan Van Winkel
Student Bachelor in de Elektronica-ICT – Cloud & Cyber Security

Inhoudsopgave

1. TERMINOLOGY & ABBREVIATIONS	4
2. INTRODUCTION	5
3. REFRACTED	6
4. INTERNSHIP	7
4.1. Background	7
4.2. Objectives	7
4.3. Assignment	8
4.4. Personal development	8
4.5. Provided Documents	8
5. ASSIGNMENT OVERVIEW	9
5.1. Technologies	9
5.2. Setup	10
5.2.1. Lab Environment (VirtualBox)	10
5.2.2. Manual Infrastructure Building	10
5.2.3. Monitoring & Telemetry Integration	11
5.2.4. Development Workflow	11
5.3. Scripting & Logic	11
5.4. Monitoring & Detection Analysis	11
6. TECHNICAL RESEARCH: AD CS VULNERABILITY CATALOG	12
6.1. Understanding the Attack Surface	12
6.2. Vulnerability Summaries	13
6.2.1. ESC1: Subject Alternative Name (SAN) Misconfiguration	13
6.2.2. ESC2: Overly Permissive Extended Key Usage (EKU)	14
6.2.3. ESC3: Certificate Request Agent Abuse	14
6.2.4. ESC4: Weak Template Access Control	15
6.2.5. ESC5: PKI Infrastructure Access Control Abuse	16
6.2.6. ESC6: CA-Level SAN Injection	17
6.2.7. ESC7: Misconfigured CA Access Control	17
6.2.8. ESC8: NTLM Relay to AD CS Web Enrollment	18
6.2.9. ESC9: No Security Extension (SID Mapping Bypass)	19
6.2.10. ESC10: Weak Certificate-to-Account Mapping	19
6.2.11. ESC11: NTLM Relay Vulnerabilities in AD CS RPC Enrollment	20
6.2.12. ESC12: CA Private Key Provider Abuse	21
6.2.13. ESC13: Policy OID Group Linkage	22
6.2.14. ESC14: Explicit Certificate Mapping Abuse	22
6.2.15. ESC15: Application Policy Injection (EKUwu)	24
6.2.16. ESC16: CA-Wide SID Extension Suppression	25
7. ADVANCED IDENTITY RESEARCH: TRUSTS & PERSISTENCE	26
7.1. Cross-Domain Trust Attacks	26

7.2. Golden gMSA (Group Managed Service Accounts)	26
7.3. Cross-Domain & Cross-Forest Trust Exploitation	27
7.4. Case Study: Cross-Domain Exploitation (XYZ.COM → ABC.COM)	28
8. MONITORING & DETECTION ANALYSIS (WAZUH)	29
8.1. Monitoring Architecture	29
8.2. LDAP Enumeration Telemetry Analysis	29
8.3. Critical Telemetry & Key Event Ids	30
8.4. Wazuh Rule Customization	30
8.5. Preliminary Findings (The Detection Gap)	31
9. CONCLUSION	32
9.1. Acknowledgements	32
10. REFERENCES	33

1. Terminology & Abbreviations

In this document there will be abbreviations and terms which may be hard to understand. This list will note all of them and give brief explanation if necessary.

Abbreviation / Terms	Explanation
AD	Active Directory
AD CS	Active Directory Certificate Services
ACL	Access Control List
AIA	Authority Information Access
CA	Certificate Authority
CRL	Certificate Revocation List
DC	Domain Controller
EKU	Extended Key Usage
EPA	Extended Protection for Authentication
gMSA	Group Managed Service Account
HSM	Hardware Security Module
KDC	Key Distribution Center
KSP	Key Storage Provider
LDAP	Lightweight Directory Access Protocol
NTLM	NT LAN Manager authentication protocol
OID	Object Identifier
OSINT	Open-Source Intelligence
PAC	Privilege Attribute Certificate
PKI	Public Key Infrastructure
PKINIT	Public Key Cryptography for Initial Authentication in Kerberos
RPC	Remote Procedure Call
SAN	Subject Alternative Name
SID	Security Identifier
SIEM	Security Information and Event Management
SOC	Security Operations Center
TGT	Ticket Granting Ticket
UPN	User Principal Name
Wazuh	Open-source SIEM and monitoring platform
HTB	Hack The Box, an online cybersecurity training platform containing intentionally vulnerable systems for penetration testing practice

Table 1. Terminology and abbreviations used throughout the document

2. Introduction

In modern enterprise environments, identity infrastructure has become one of the most critical security boundaries. As organizations increasingly rely on centralized authentication systems, Active Directory (AD) remains a primary target for adversaries seeking privilege escalation, persistence, and lateral movement. Within this infrastructure, Active Directory Certificate Services (AD CS) introduces additional complexity due to its role in certificate-based authentication and trust management. Misconfigurations within AD CS can allow attackers to impersonate privileged users, bypass traditional authentication controls, and maintain long-term persistence within a domain environment.

Recent research by SpecterOps demonstrated that many organizations unknowingly expose critical attack paths through improperly configured certificate templates and weak access control mechanisms (Schroeder & Christensen, 2021). These vulnerabilities are commonly categorized as ESC1 through ESC16 and represent multiple methods of abusing AD CS for domain compromise.

This internship focused on researching these vulnerabilities and developing custom automation scripts to identify vulnerable certificate templates more discreetly than existing public tooling. The project combined offensive security research with defensive monitoring by comparing the telemetry generated by targeted enumeration techniques against standard tools commonly used during Active Directory assessments. In addition to vulnerability research, the project explored trust relationships between Active Directory domains and forests, including the abuse of inter-realm trust accounts and Group Managed Service Accounts (gMSAs). A monitored laboratory environment containing Active Directory, AD CS, Wazuh, and Sysmon was constructed to evaluate how identity-based attacks appear within a Security Operations Center (SOC).

This document first introduces the internship environment and project objectives. It then explains the technical architecture, scripting logic, and monitoring infrastructure used throughout the research. The main section of the document provides a detailed analysis of AD CS vulnerabilities from ESC1 through ESC16, followed by research into trust exploitation, persistence mechanisms, and detection engineering. Finally, the document evaluates the effectiveness of defensive monitoring within the created lab environment and summarizes the key findings of the project.

3. Refracted

Refracted is a dynamic, young cybersecurity firm specializing in the deep technical aspects of the digital landscape. Their team consists of highly skilled and certified professionals with extensive expertise in Red Teaming, Open-Source Intelligence (OSINT), penetration testing, vulnerability management, and incident response.

The mission of Refracted is to empower organizations to fortify their digital defences through technical excellence, transparent guidance, and a steadfast commitment to real-world results. Rather than following trends, they focus on identifying what is truly at risk and providing tangible fixes.

I chose to pursue my internship at Refracted because its smaller size fosters direct communication and a collaborative atmosphere. The team is highly motivating and consistently willing to share their deep knowledge of complex cybersecurity topics. Furthermore, Refracted is one of the few organizations offering specialized Red Teaming internships. This was a natural next step for me, coming from the new Ethical Hacking trajectory within the Cloud & Cybersecurity program at Thomas More.

What appealed to me most was the level of autonomy, I was given the opportunity to select my assignment from a curated list of projects. While I knew I wanted to dive into Active Directory, choosing AD CS was a very intentional decision. This project allowed me to bring my own ideas to the table regarding the scope and technical execution, giving me the freedom to shape the assignment as it evolved.

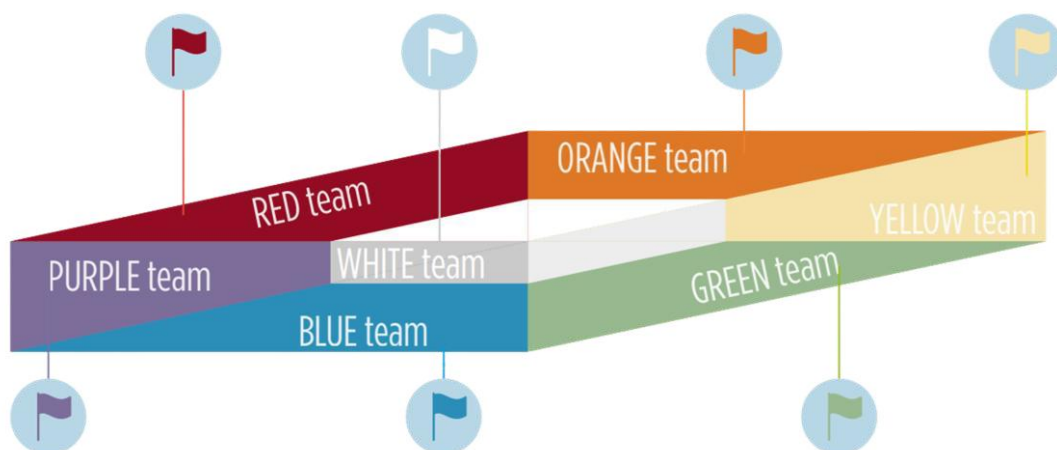


Figure 1 Refracted Security color spectrum

Cybersecurity can be compared to light: it is composed of various color components that form a whole. The name "Refracted" represents how security can be broken down into distinct, specialized elements while remaining a unified defense. They integrate different "color teams," each focusing on a specific security domain, such as Blue Teaming for defensive improvements or Yellow Teaming for securing software development.

While Refracted primarily focuses on the Red Teaming aspect, they collaborate closely with partners to provide a comprehensive solution for any cybersecurity challenge.

4. Internship

From February to May, I conducted my internship at Refracted. My goal was to develop a project centered around Active Directory security, a topic that deeply interests me and aligns with my specialization. This chapter explains the project's background, objectives, and the specific assignment I undertook.

4.1. Background

In a modern enterprise environment, Active Directory (AD) is the heart of the network. While many organizations focus on perimeter defense, internal identity structures often remain vulnerable. One of the most critical yet misunderstood components is Active Directory Certificate Services (AD CS). When misconfigured, AD CS can provide an attacker with a direct path to full Domain Admin privileges.

A significant challenge for security teams is the manual effort required to audit these complex certificate templates and trust relationships. Furthermore, even when these vulnerabilities are exploited, organizations often lack the visibility to detect the "noise" created by such attacks. There is a clear need for automation to identify these flaws and for research into how these techniques appear within monitoring systems.

4.2. Objectives

In collaboration with my mentor, I defined a project that provides value to Refracted's Red Teaming operations while allowing me to deepen my technical expertise.

The project has three main goals:

- **Infrastructure Research:** To gain hands-on experience with complex AD environments, including the study of Trust relationships and simulated infrastructures.
- **Stealth-Focused Automation:** To develop custom PowerShell and Python scripts that identify AD CS vulnerabilities more discreetly than existing public tools. By targeting specific vulnerable templates rather than performing broad, noisy queries, the goal is to reduce the footprint left within the network.
- **Detection & Visibility Analysis:** To evaluate the effectiveness of these scripts against a basic SOC and monitoring setup, determining which specific actions trigger alerts and which remain under the radar.

4.3. Assignment

The core of this assignment is the development of a **Security Assessment Framework** specifically for AD CS. A key differentiator of this project is the focus on **stealth and efficiency**.

Most existing scanning tools are "noisy" because they request and analyze every available certificate template, which is easily flagged by modern monitoring systems. My assignment involved writing scripts that specifically query for known misconfigurations. While this targeted approach requires more precise logic and acknowledges that no script is 100% perfect in every edge case, it significantly increases the "silence" of the operation.

The assignment was executed in three phases:

1. **Scripting & Logic:** Developing PowerShell and Python tools to perform targeted queries for vulnerable certificate templates (such as ESC1) without triggering mass-enumeration alerts.
2. **Lab Simulation:** Setting up a controlled environment with a Domain Controller and a Certificate Authority (CA) to refine the script logic and test cross-domain Trust scenarios.
3. **Monitoring Confrontation:** Running the developed scripts against a monitored AD structure to compare their telemetry against common, louder offensive tools. This provides a direct insight into the trade-off between script "perfection" and operational stealth.

4.4. Personal development

During the internship period, additional practical experience was gained through participation in weekly Hack The Box (HTB) challenges. Although only three machines were fully completed to both user and system level, the platform provided valuable hands-on experience in areas such as enumeration, privilege escalation, Active Directory exploitation, and post-exploitation techniques. This practical exposure complemented the research conducted throughout the internship.

4.5. Provided Documents

Since the developed tools and scripts are intended for internal use by Refracted, the source code and detailed manuals will remain confidential a few will be given as an example. This document provides a high-level overview of the realization.

- Projectplan
- Realization document (this document)
- Documentation about the attacks
- Source code repository (Internal for Refracted)
- Personal Reflection

5. Assignment Overview

This chapter provides an overview of the technical architecture of the project and the technologies used to develop the scripts and the research environment. The project focuses on a combination of custom-developed offensive scripts and a defensive monitoring setup to evaluate attack visibility.

5.1. Technologies

To achieve the project goals, a specific stack of technologies was chosen. These tools were selected for their effectiveness in Active Directory environments and their ability to provide detailed telemetry for the monitoring phase.

PowerShell

As the native management language for Windows, PowerShell was the primary choice for developing the AD CS discovery scripts. It allows for direct interaction with Active Directory objects and Certificate Templates.

- **Purpose:** Extracting certificate template metadata and identifying misconfigured permissions (e.g., ESC1).
- **Key Advantage:** Using native AD modules makes the scripts "quieter" than external binaries, as they leverage legitimate administrative functions.

Python

Python was used to complement the PowerShell scripts, specifically for data processing and interacting with more complex exploit logic.

- **Purpose:** Automating the analysis of exported AD data and scripting specific exploitation paths that are more efficiently handled in a cross-platform language.
- **Library usage:** Leveraging libraries like *Impacket* for interacting with network protocols when testing from a non-domain-joined perspective.

Active Directory Certificate Services (AD CS)

The core subject of the research. A Windows Server-based PKI (Public Key Infrastructure) was implemented to serve as the target for the scripts.

- **Purpose:** Hosting the Certificate Authority (CA) and various "intentionally misconfigured" templates to test the accuracy of the detection scripts.

Wazuh (SIEM)

Wazuh was implemented as the central Security Information and Event Management (SIEM) solution. It is a widely used, open-source platform that is essential for the "Blue Team" aspect of this project.

- **Purpose:** Aggregating telemetry from the Domain Controller and Certificate Authority.
- **Analysis:** Custom rules and dashboards in Wazuh were used to monitor specific Event IDs related to certificate requests and template modifications. This allowed for a direct comparison of the "noise" generated by standard offensive tools versus my custom, targeted scripts.

Lab Infrastructure (Virtualization)

The project was built within a virtualized environment to simulate a corporate network.

- **Components:** Multiple Windows Server instances (Domain Controllers and CAs) and a workstation for the attacker's perspective.
- **Trusts:** The environment was expanded to include AD Trusts, allowing for research into how certificate-based attacks can cross-domain boundaries.

5.2. Setup

The setup for this project was a deliberate balance between automated scripting and manual infrastructure building. Unlike pre-configured labs, I chose a manual approach to gain deep, hands-on experience with technologies like AD CS, which were not extensively covered in my curriculum at Thomas More.

5.2.1. Lab Environment (VirtualBox)

The lab was hosted locally on a high-performance workstation using **Oracle VM VirtualBox**. This provided a controlled, isolated environment to perform "offensive" actions without risking any external infrastructure. The environment evolved during the internship based on the technical requirements of the vulnerabilities being researched:

- **Phase 1 (Consolidated):** Initially, I deployed a single Windows Server 2022 instance acting as both the Domain Controller (DC) and the Certificate Authority (CA).
- **Phase 2 (Distributed):** During the testing of **ESC8 (AD CS HTTP Relay)**, I discovered that performing NTLM relaying back to the same machine is restricted by default security hardening. To bypass this and create a realistic attack scenario, I transitioned to a distributed architecture with a separate **Domain Controller** and a dedicated **Issuing CA server**.
- **Monitoring Hub:** The prebuilt Wazuh VM was added to the lab to host the **Wazuh Manager**, ensuring that both Windows servers and clients could stream their logs to a central location.

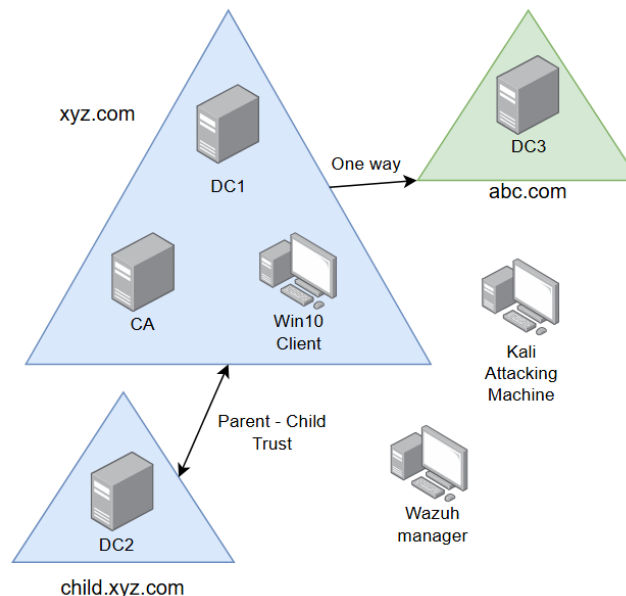


Figure 2 Final lab setup

5.2.2. Manual Infrastructure Building

I chose to build the Active Directory and PKI (Public Key Infrastructure) from scratch rather than using automation scripts like *BadBlood*. This manual process was essential for:

- **Hands-on Learning:** Installing and configuring the AD CS role, setting up the Root CA, and managing the Certificate Revocation List (CRL).
- **Intentional Misconfiguration:** Manually editing certificate templates to introduce flaws like ENROLLEE_SUPPLIES_SUBJECT (ESC1) or enabling insecure Web Enrollment (ESC8). This ensured I knew exactly which settings my scripts were scanning for.

5.2.3. Monitoring & Telemetry Integration

To bridge the gap between attacking and monitoring, I manually implemented a high-visibility logging stack:

- **Wazuh Agents:** Installed on both the DC and the CA to forward events to the Wazuh Manager.
- **Sysmon:** Deployed with a specialized configuration to track process executions and network connections, which are critical for detecting "silent" PowerShell activity.
- **Audit Policies:** Manually configured "Advanced Audit Policies" on the Windows servers to capture Event ID 4898 (Certificate Services template access), which I thought was the primary indicator of AD CS enumeration. But this event only loads upon start of the Certificate Service when loading each template.

5.2.4. Development Workflow

The development of the PowerShell and Python scripts took place on the host machine using **Visual Studio Code**, connected to the lab environment. This allowed for an iterative process: writing a "silent" query in the script, executing it against the CA, and immediately checking the **Wazuh dashboard** to see if the activity was flagged or if it remained undetected.

5.3. Scripting & Logic

The core of the assignment involved developing a specialized toolkit for AD CS enumeration. The project was split into two distinct perspectives to simulate different stages of an attack: an authenticated internal user and an external attacker with compromised credentials.

PowerShell (Internal Perspective)

This script is designed to be executed from a **domain-joined host**.

- **Targeted Enumeration:** It leverages the native AD modules to query the Certificate Authority. Instead of broad enumeration, the script specifically filters for attributes that indicate vulnerabilities like ESC1 or ESC3.
- **Stealth:** By performing targeted LDAP queries rather than a "dump all" approach, the script minimizes the number of Directory Service objects accessed, reducing the likelihood of triggering broad enumeration alerts in a SOC.

Python (External Perspective)

This script is designed for use from a **non-domain-joined machine** (e.g., a Kali Linux instance).

- **Remote Enumeration:** Using only a compromised username and password, this script interacts with the domain via RPC/LDAP protocols.
- **Functionality:** It replicates the logic of the PowerShell module but is built to function without the benefit of native Windows modules or domain-joined status, providing a realistic "entry point" scenario for an attacker who has just gained credential access.

5.4. Monitoring & Detection Analysis

The final part of the assignment involved using the **Blue Team** perspective to measure the effectiveness of the custom scripts.

- **Visibility Comparison:** The primary goal was to see if the "targeted" approach of my scripts was more silent than industry-standard tools. In the **Wazuh dashboard**, I analyzed the logs to see if my scripts generated fewer "Object Access" and "Directory Service" events than tools like Certipy.
- **Telemetry Gap:** I researched which Event IDs (such as 4898 or 4899) were triggered during enumeration. This helped identify the "detection gap", where an attacker could potentially find a path to Domain Admin without ever triggering a high-severity alert in a basic SOC setup.

6. Technical Research: AD CS Vulnerability Catalog

To develop effective enumeration scripts, it was essential to first conduct a deep-dive analysis of known AD CS misconfigurations and attack paths. This chapter serves as a technical catalog of the research performed throughout the internship. For each vulnerability, the core logic, exploitation path, detection considerations, and mitigation strategies were analyzed.

The research in this section is primarily based on technical documentation and research published by SpecterOps, Certipy, The Hacker Recipes, and Microsoft Learn. Because these sources were used extensively across all ESC-related subsections, individual citations are not repeated within every paragraph in order to preserve readability and maintain a clear document structure.

Only a summarized version of the research is included within this realization document. The complete technical documentation, including detailed exploitation steps, commands, screenshots, and operational notes, can be found at <http://www.daanvanwinkel.be/internship/> under the ZIP file download, which is intentionally not a zip bomb.

6.1. Understanding the Attack Surface

The vulnerabilities, labeled ESC1 through ESC16, generally fall into three categories:

1. **Template Misconfigurations:** Flaws in how certificate templates are defined (e.g., ESC1, ESC3).
2. **Access Control Issues:** Weak permissions on the CA or templates (e.g., ESC4, ESC7).
3. **Protocol & Mapping Flaws:** Exploiting how Windows maps certificates to accounts or how protocols like NTLM interact with web enrollment (e.g., ESC8, ESC9).

6.2. Vulnerability Summaries

6.2.1. ESC1: Subject Alternative Name (SAN) Misconfiguration

ESC1 is one of the most critical AD CS privilege escalation paths because it allows a low-privileged user to request a certificate for another identity. The vulnerability occurs when a certificate template permits requester-defined Subject Alternative Names (SANs) while simultaneously allowing authentication-related Extended Key Usages (EKUs).

Attack Logic

The attacker abuses the template by embedding the UPN and SID of a privileged account into the certificate request. Because the CA signs the certificate as trusted, the Domain Controller accepts the resulting certificate during PKINIT authentication.

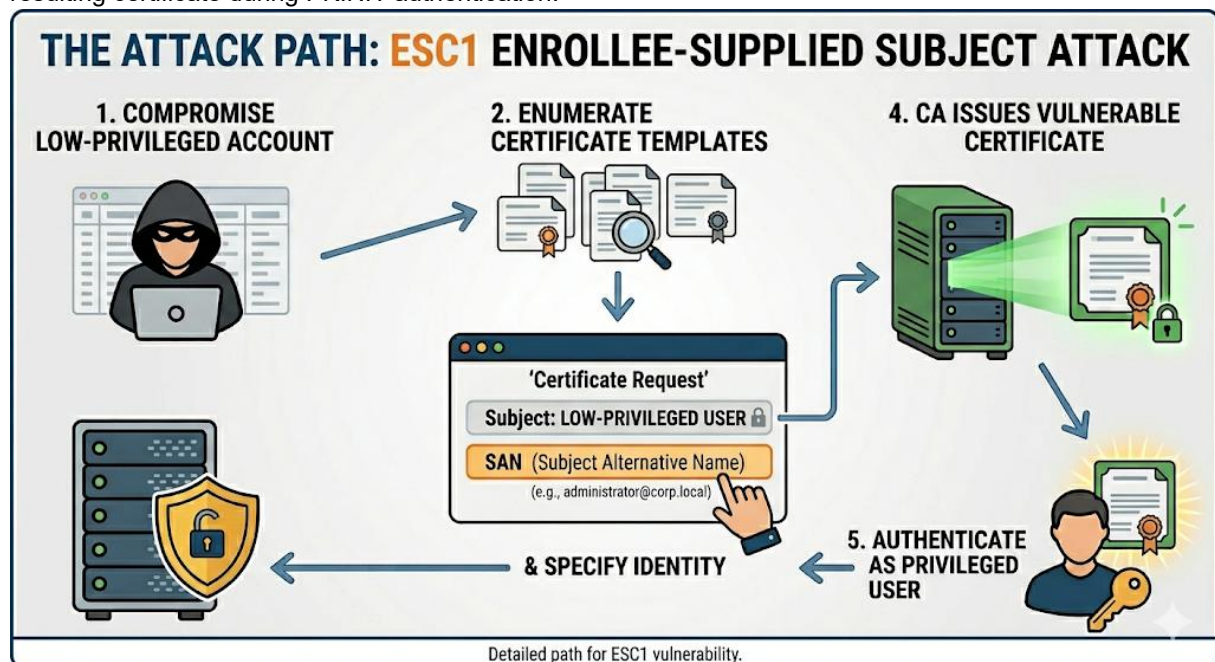


Figure 3 ESC1 Attack Flow

Detection Considerations

From a monitoring perspective, ESC1 abuse can resemble legitimate certificate enrollment activity. Detection requires inspection of SAN mismatches between the requester and issued certificate. Certificate Services events and CA audit logs provide the most reliable telemetry.

Operational Impact

Successful exploitation results in impersonation of privileged users, including Domain Admins. Certificate-based authentication also enables long-term persistence independent of password rotation.

Mitigation

- Disable requester-defined SAN values
- Restrict enrollment permissions
- Require manager approval
- Enforce strong certificate mapping

6.2.2. ESC2: Overly Permissive Extended Key Usage (EKU)

ESC2 occurs when a certificate template allows unrestricted Extended Key Usage through the “Any Purpose” EKU or through the absence of EKU restrictions entirely. This causes certificates to become valid for unintended authentication scenarios.

Attack Logic

The attacker enrolls in a template that was not intended for authentication but still permits authentication implicitly because of overly broad EKU configuration. The issued certificate can then be used for PKINIT authentication.

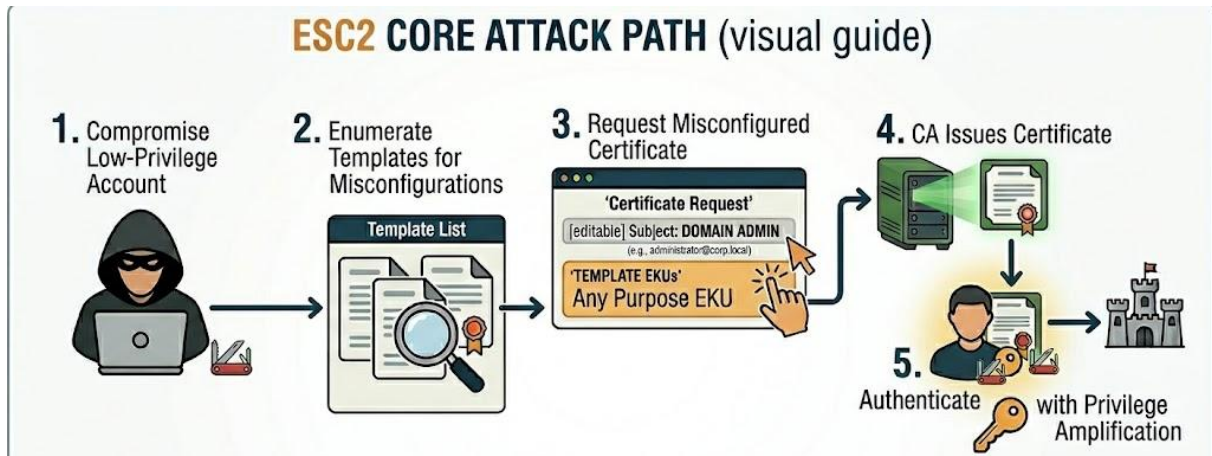


Figure 4 ESC2 Attack Flow

Detection Considerations

ESC2 abuse often appears as legitimate enrollment activity because no impersonation occurs during issuance. Detection therefore depends on identifying templates with dangerous EKU configurations rather than monitoring exploitation itself.

Operational Impact

The vulnerability enables attackers to transform generic certificate issuance into valid domain authentication credentials, potentially allowing persistence and privilege amplification.

Mitigation

- Remove “Any Purpose” EKU
- Explicitly define allowed EKUs
- Restrict enrollment permissions
- Disable unused templates

6.2.3. ESC3: Certificate Request Agent Abuse

ESC3 occurs when low-privileged users can obtain Enrollment Agent certificates that allow requesting certificates on behalf of other users.

Attack Logic

The attacker first requests an Enrollment Agent certificate. Using this certificate, they submit a second request on behalf of a privileged user. The CA trusts the Enrollment Agent and issues a valid authentication certificate for the target identity.

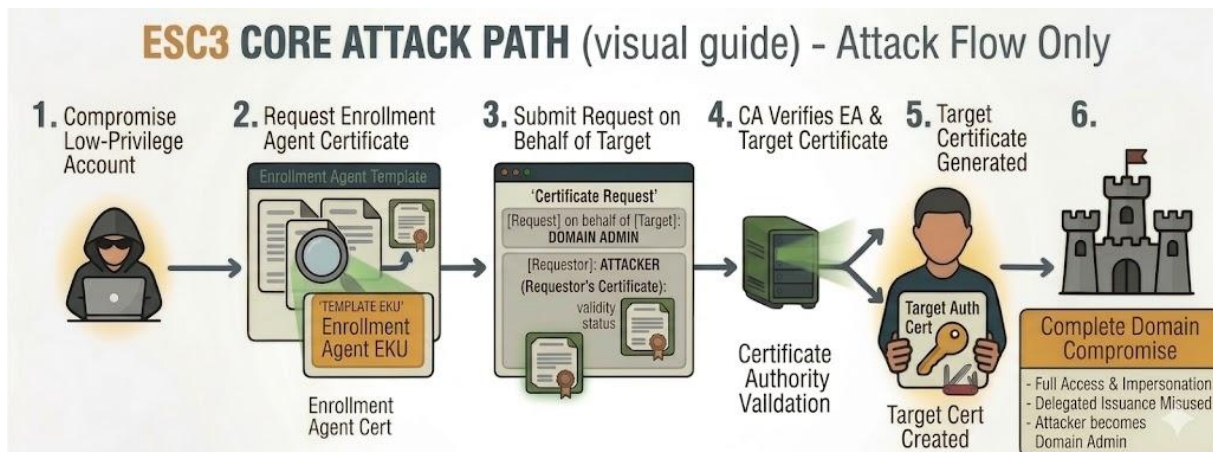


Figure 5 ESC3 Attack Flow

Detection Considerations

Enrollment Agent activity generates more detectable telemetry than standard enrollment. Certificate requests made "on behalf of" another user should be considered highly suspicious outside controlled administrative workflows.

Operational Impact

ESC3 effectively delegates certificate issuance authority to attackers, allowing direct impersonation of privileged users and complete domain compromise.

Mitigation

- Restrict Enrollment Agent templates
- Replace legacy V1 templates
- Require manager approval
- Disable unused enrollment templates

6.2.4. ESC4: Weak Template Access Control

ESC4 occurs when attackers possess dangerous permissions over certificate templates, allowing them to modify otherwise secure templates into vulnerable configurations.

Attack Logic

The attacker modifies template settings to enable authentication-related EKUs, SAN supply, or weak issuance requirements. The modified template then becomes exploitable through ESC1-style abuse.

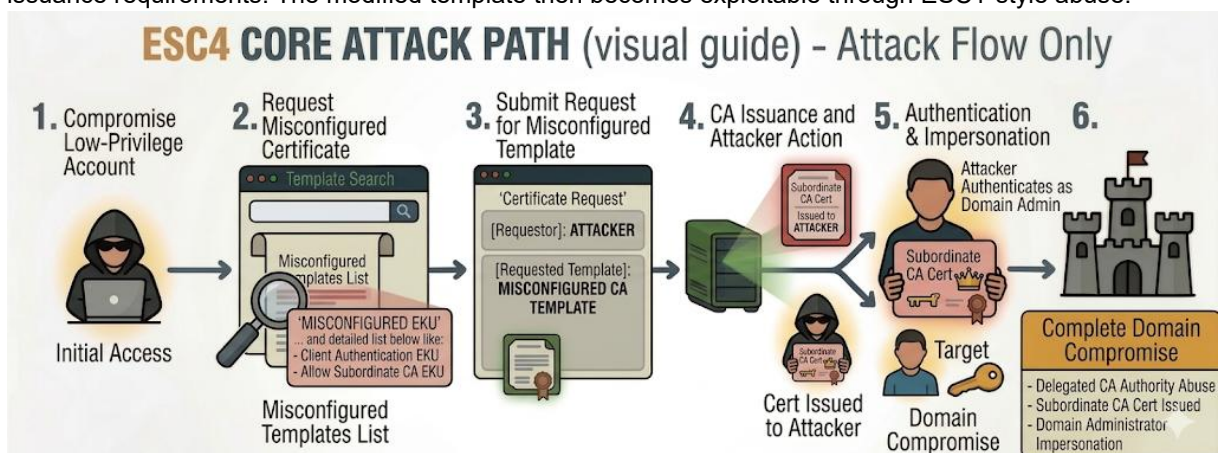


Figure 6 ESC4 Attack Flow

Detection Considerations

Template modifications generate more detectable telemetry than normal enrollment activity. LDAP object modifications and template configuration changes should be monitored closely.

Operational Impact

ESC4 demonstrates that AD CS security depends heavily on Active Directory ACL hygiene. Even secure templates become dangerous if modification permissions are delegated incorrectly.

Mitigation

- Restrict template ACLs
- Audit inherited permissions
- Monitor template changes
- Remove unnecessary write access

6.2.5. ESC5: PKI Infrastructure Access Control Abuse

ESC5 targets excessive permissions over core AD CS infrastructure objects rather than certificate templates themselves.

Attack Logic

Attackers abuse write access over critical PKI objects such as NTAuthCertificates, Enrollment Services, or AIA/CDP containers. This allows modification of domain trust anchors and certificate validation mechanisms.

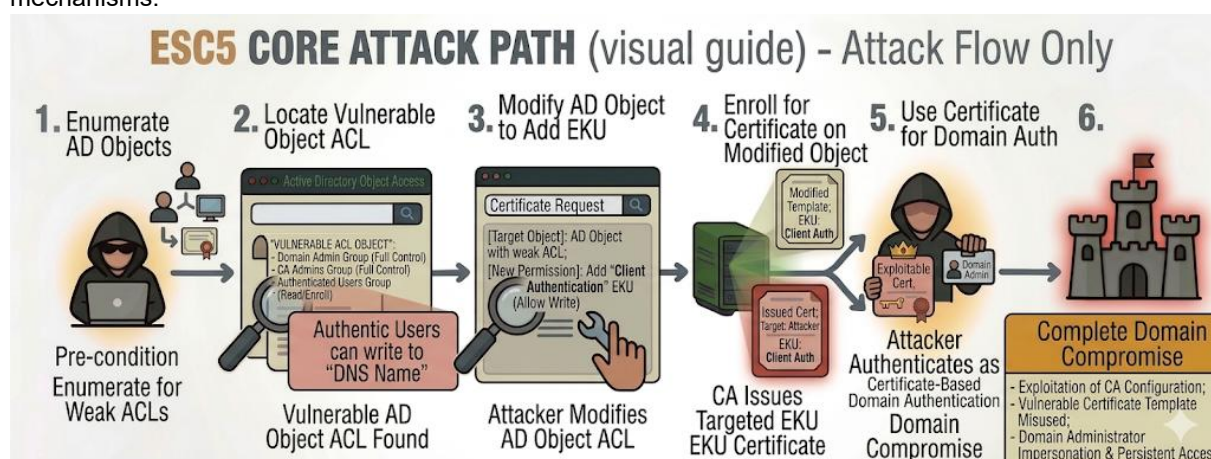


Figure 7 ESC5 Attack Flow

Detection Considerations

Infrastructure-level PKI changes are rare in normal operations. Monitoring modifications to Public Key Services containers provides strong detection opportunities.

Operational Impact

ESC5 compromises the trust foundation of Active Directory itself. Successful exploitation can allow rogue CA trust insertion and complete identity compromise.

Mitigation

- Restrict PKI ACLs
- Audit PKI containers regularly
- Monitor trust anchor changes
- Harden CA administration

6.2.6. ESC6: CA-Level SAN Injection

ESC6 occurs when the Certification Authority globally allows requester-defined Subject Alternative Names through the EDITF_ATTRIBUTESUBJECTALTNAME2 flag.

Attack Logic

Even secure templates become vulnerable because the CA overrides template restrictions and accepts arbitrary SAN values supplied during certificate requests.

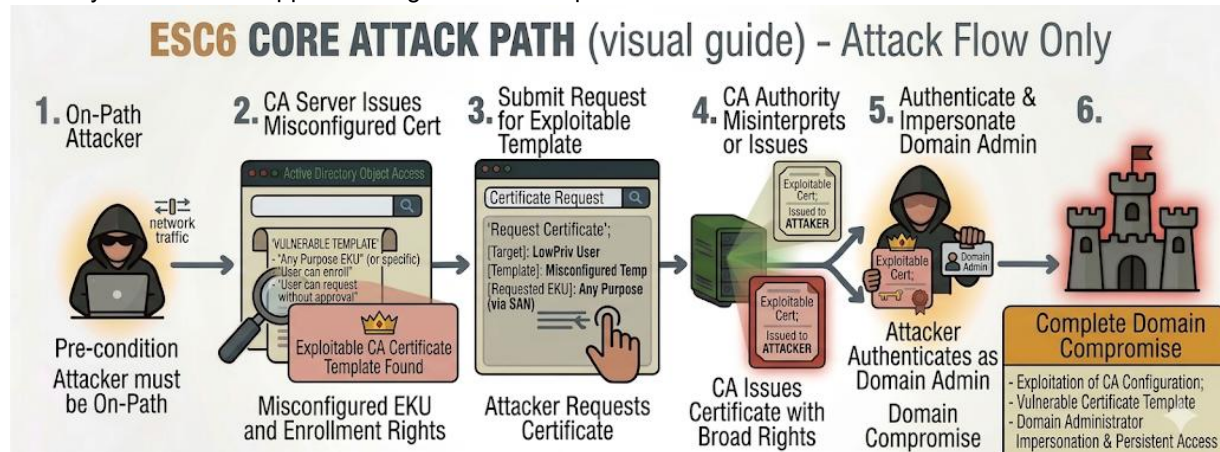


Figure 8 ESC6 Attack Flow

Detection Considerations

Detection requires correlating issued SAN values with requesting identities. CA configuration auditing is essential because the vulnerability exists at the infrastructure level rather than within templates.

Operational Impact

ESC6 converts standard authentication templates into identity spoofing mechanisms capable of impersonating privileged users.

Mitigation

- Disable EDITF_ATTRIBUTESUBJECTALTNAME2
- Restrict enrollment permissions
- Audit issued certificates
- Harden CA configuration

6.2.7. ESC7: Misconfigured CA Access Control

ESC7 occurs when low-privileged users possess dangerous permissions directly on the Certification Authority.

Attack Logic

Attackers abuse permissions such as ManageCA or ManageCertificates to approve pending requests, publish vulnerable templates, or modify CA behavior.

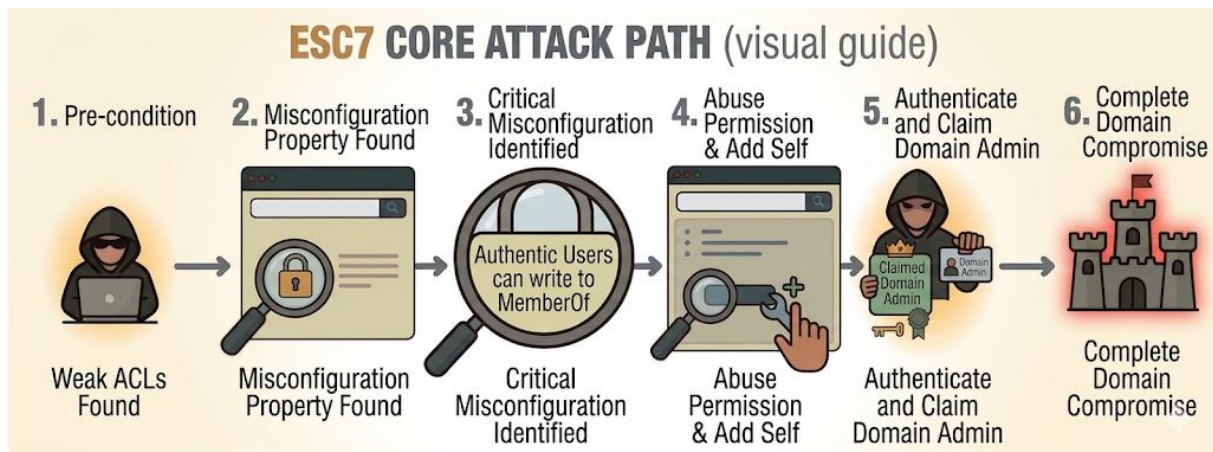


Figure 9 ESC7 Attack Flow

Detection Considerations

CA management actions generate strong telemetry and should rarely occur outside administrative maintenance. Unexpected certificate approvals are particularly suspicious.

Operational Impact

ESC7 enables complete CA control-plane compromise, bypassing template-level protections entirely.

Mitigation

- Restrict CA management permissions
- Audit CA ACLs
- Monitor approval workflows
- Harden CA administration

6.2.8. ESC8: NTLM Relay to AD CS Web Enrollment

ESC8 abuses NTLM relay attacks against AD CS web enrollment endpoints.

Attack Logic

The attacker relays NTLM authentication from a victim machine or user to the CA web enrollment service and requests a certificate in the victim's context.

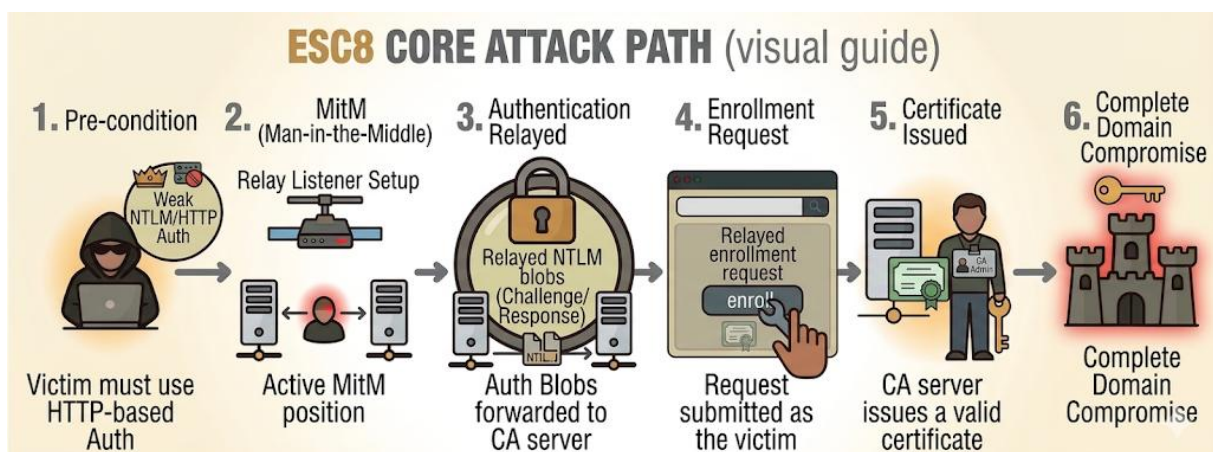


Figure 10 ESC8 Attack Flow

Detection Considerations

ESC8 generates network-level indicators including NTLM relay behavior and unusual web enrollment activity. Monitoring HTTP enrollment endpoints is critical.

Operational Impact

Successful exploitation allows attackers to obtain certificates for privileged systems or users without requiring credentials directly.

Mitigation

- Disable NTLM where possible
- Require HTTPS with Extended Protection
- Disable web enrollment if unnecessary
- Monitor enrollment endpoints

6.2.9. ESC9: No Security Extension (SID Mapping Bypass)

ESC9 occurs when certificates are accepted without strong SID-based mapping validation.

Attack Logic

The attacker abuses weak certificate-to-account binding to authenticate using manipulated certificate identities that bypass strong mapping protections.

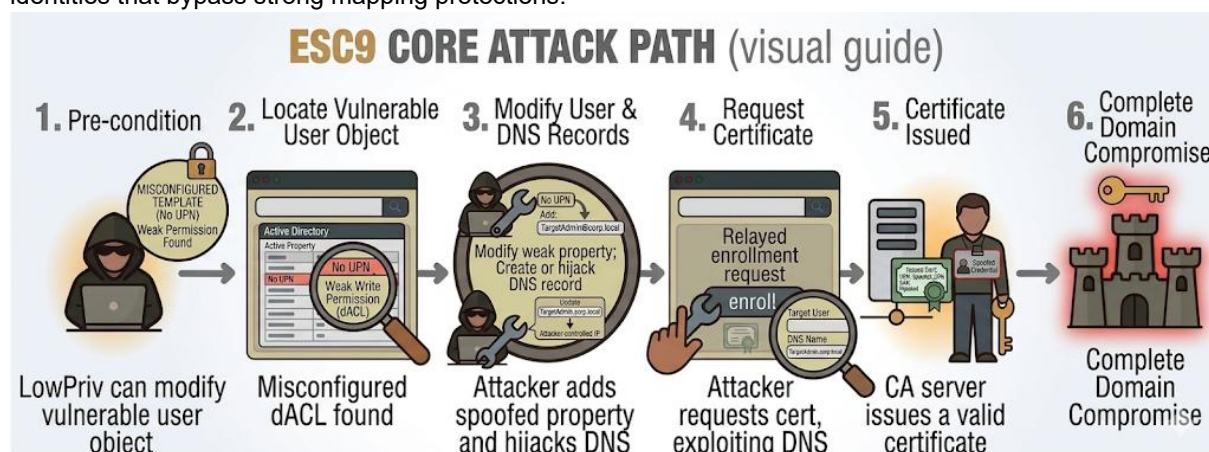


Figure 11 ESC9 Attack Flow

Detection Considerations

Detection depends heavily on identifying certificates lacking SID extensions or mismatched mapping attributes.

Operational Impact

ESC9 weakens Microsoft's post-2022 certificate mapping protections and enables identity spoofing attacks.

Mitigation

- Enforce strong certificate mapping
- Require SID extensions
- Audit certificate mappings
- Patch Domain Controllers

6.2.10. ESC10: Weak Certificate-to-Account Mapping

ESC10 occurs when alternative certificate mapping methods allow attackers to authenticate using manipulated certificate identities.

Attack Logic

Attackers abuse weak UPN or SAN-based mapping to authenticate as another user without proper SID validation.

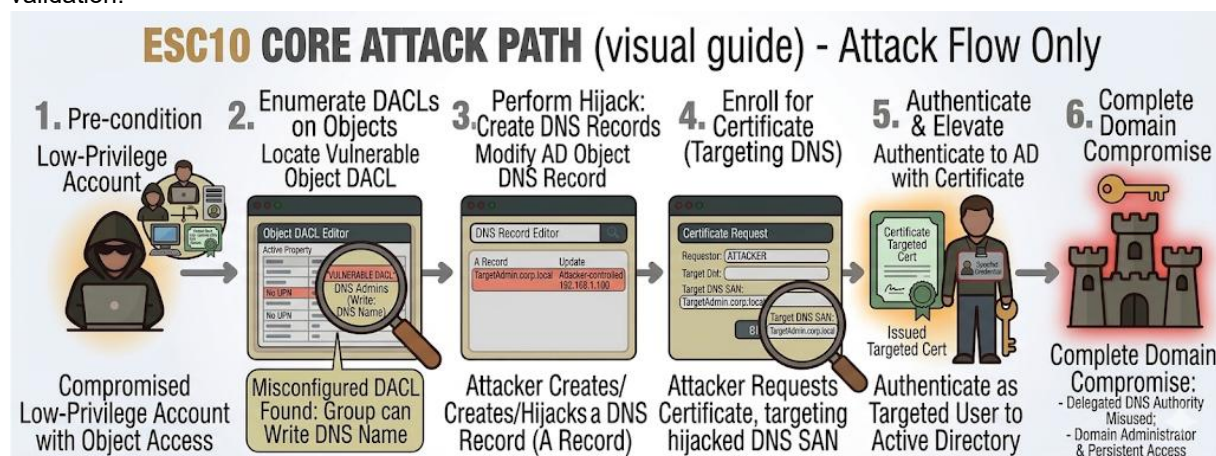


Figure 12 ESC10 Attack Flow

Detection Considerations

Authentication attempts relying on legacy certificate mapping methods should be considered suspicious in modern environments.

Operational Impact

ESC10 bypasses intended identity validation controls and enables certificate-based impersonation.

Mitigation

- Enforce strong mapping
- Disable legacy mappings
- Audit certificate authentication events
- Harden PKINIT configuration

6.2.11. ESC11: NTLM Relay Vulnerabilities in AD CS RPC Enrollment

ESC11 abuses NTLM relay attacks against the ICPR RPC interface exposed by AD CS.

Attack Logic

The attacker relays NTLM authentication to the ICPR interface and submits certificate requests using the victim's context.

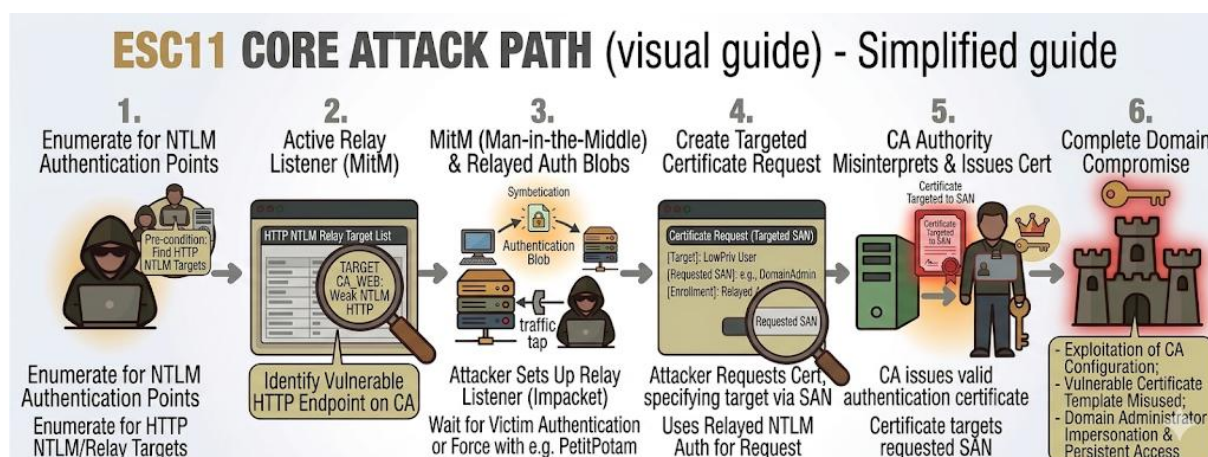


Figure 13 ESC11 Attack Flow

Detection Considerations

RPC-based certificate enrollment traffic should be monitored for unusual authentication patterns and relay indicators.

Operational Impact

ESC11 allows attackers to obtain valid certificates for privileged systems or users without direct credential compromise.

Mitigation

- Disable NTLM
- Harden RPC interfaces
- Restrict certificate enrollment
- Monitor RPC activity

6.2.12. ESC12: CA Private Key Provider Abuse

ESC12 targets environments where hardware security modules are improperly configured or integrated.

Attack Logic

Improper HSM protections expose sensitive CA operations or key material that should remain isolated from administrators and attackers.

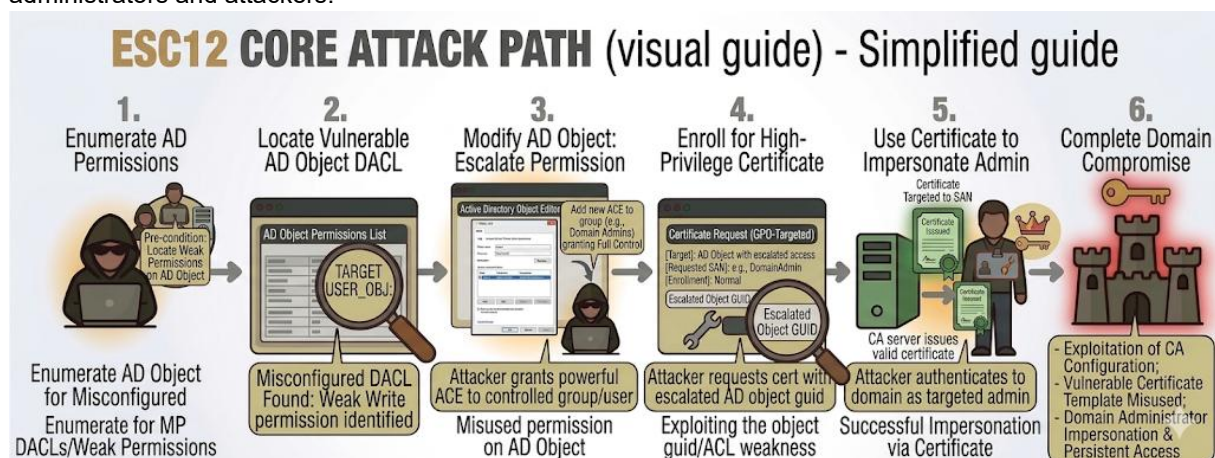


Figure 14 ESC12 Attack Flow

Detection Considerations

HSM administrative activity should be extremely limited and heavily audited.

Operational Impact

Compromise of HSM-backed CA infrastructure undermines the security guarantees of hardware-protected certificate issuance.

Mitigation

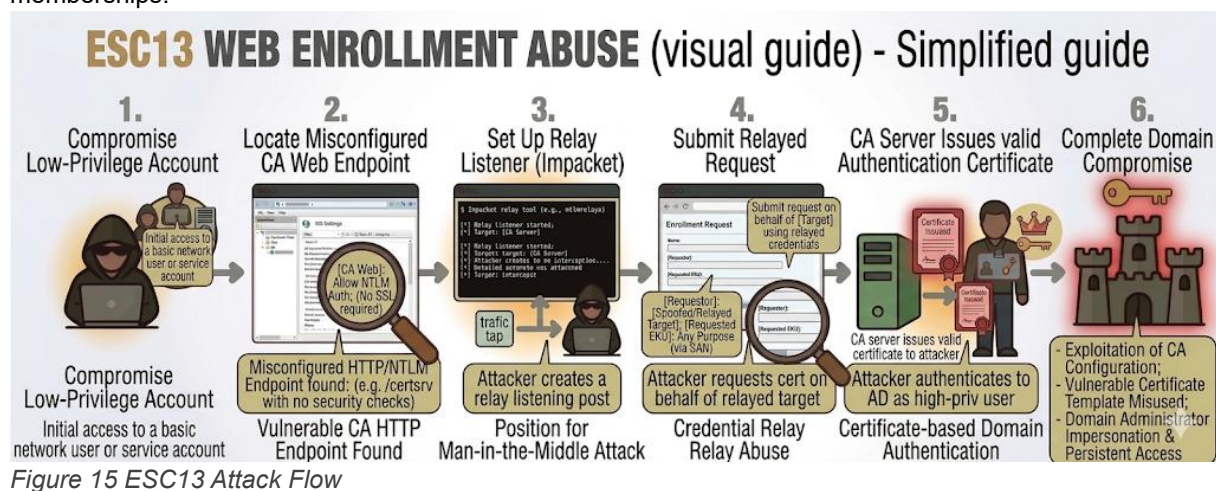
- Restrict HSM administration
- Audit HSM access
- Harden CA key management
- Enforce separation of duties

6.2.13. ESC13: Policy OID Group Linkage

ESC13 abuses issuance policy OIDs linked to privileged Active Directory groups.

Attack Logic

The attacker obtains certificates containing issuance policies that automatically map to privileged group memberships.



Detection Considerations

OID-linked group assignments should be monitored because they create implicit privilege relationships.

Operational Impact

ESC13 enables privilege escalation through certificate policy abuse rather than direct impersonation.

Mitigation

- Audit issuance policy mappings
- Restrict OID-linked groups
- Monitor certificate policy usage
- Remove unnecessary mappings

6.2.14. ESC14: Explicit Certificate Mapping Abuse

ESC14 occurs when explicit certificate mappings are configured insecurely within Active Directory.

Attack Logic

Attackers abuse manually configured certificate mappings to authenticate as another identity using attacker-controlled certificates.

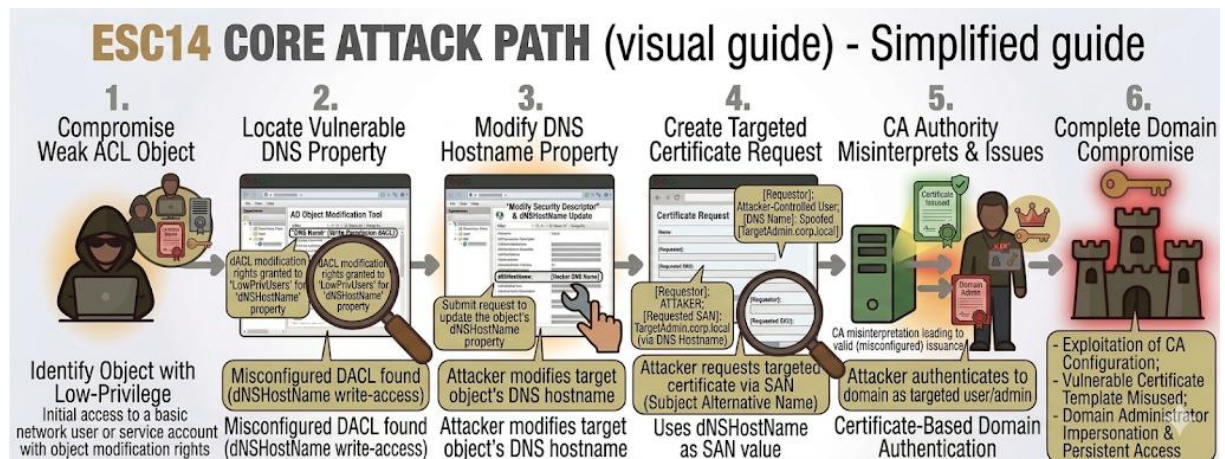


Figure 16 ESC14 Attack Flow

Detection Considerations

Explicit mappings are uncommon and should be monitored carefully for unauthorized changes.

Operational Impact

ESC14 bypasses intended certificate identity protections through insecure administrative mappings.

Mitigation

- Audit explicit mappings
- Remove unnecessary mappings
- Monitor account mapping changes
- Enforce strong mapping standards

6.2.15. ESC15: Application Policy Injection (EKUwu)

ESC15 abuses arbitrary application policies embedded within certificates.

Attack Logic

The attacker requests certificates containing dangerous or unintended application policies that permit privileged authentication behavior.

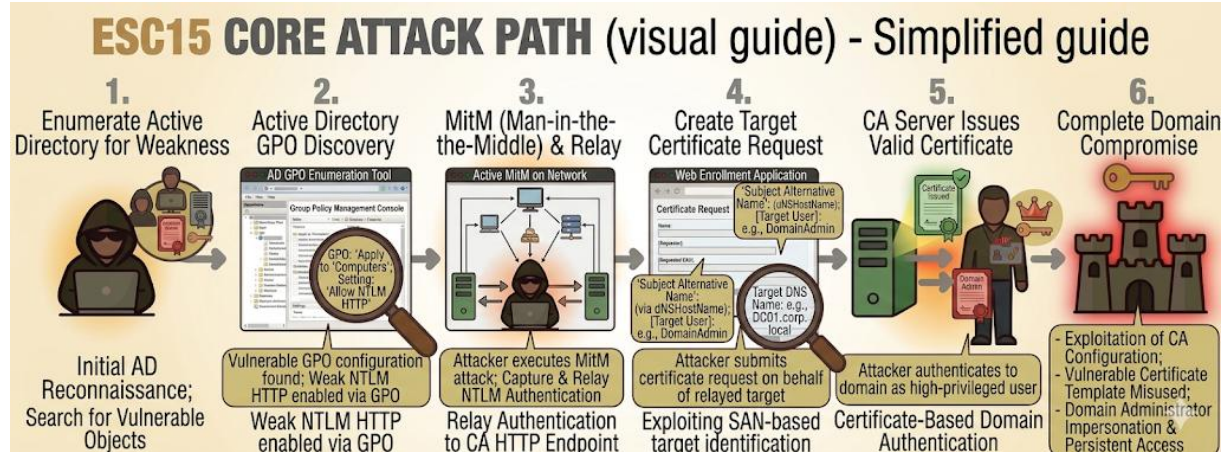


Figure 17 ESC15 Attack Flow

Detection Considerations

Unexpected application policy combinations within issued certificates should be investigated.

Operational Impact

ESC15 allows misuse of certificates outside their intended operational purpose.

Mitigation

- Restrict application policies
- Audit ECU configurations
- Monitor certificate issuance
- Harden template configurations

6.2.16. ESC16: CA-Wide SID Extension Suppression

ESC16 occurs when the CA does not embed SID security extensions into issued certificates.

Attack Logic

Without SID extensions, strong certificate mapping protections fail, allowing attackers to abuse SAN-based identity spoofing techniques.

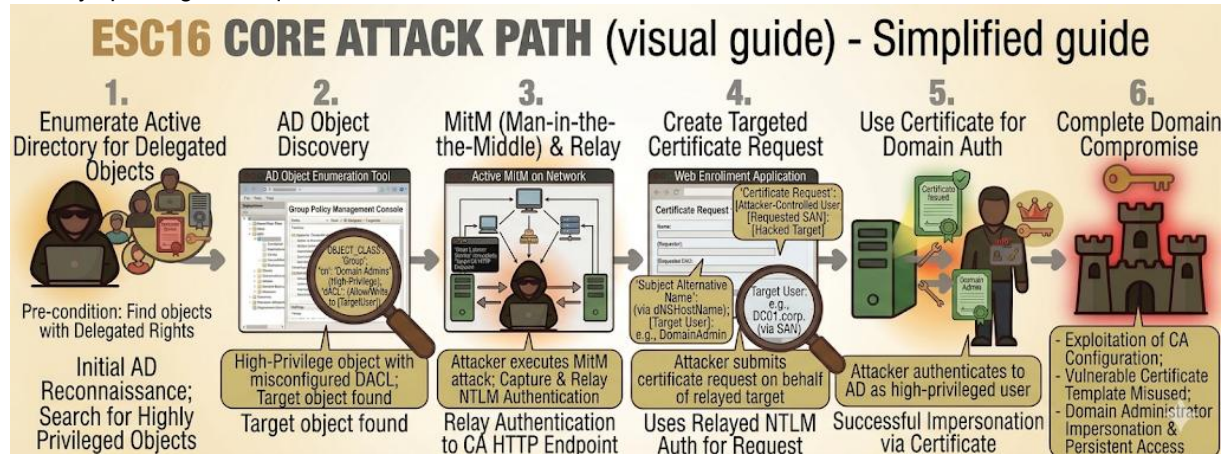


Figure 18 ESC 16 Attack Flow

Detection Considerations

Certificates lacking SID extensions should be considered suspicious in modern patched environments.

Operational Impact

ESC16 re-enables several older AD CS attack paths despite modern Microsoft hardening efforts.

Mitigation

- Enable SID extensions
- Enforce strong mapping
- Audit issued certificates
- Patch AD CS infrastructure

7. Advanced Identity Research: Trusts & Persistence

While AD CS provides a powerful entry point, modern enterprise networks often rely on complex trust relationships and automated service accounts. This chapter details my research into traversing these boundaries and establishing long-term persistence.

7.1. Cross-Domain Trust Attacks

Active Directory Trusts are designed to provide a security boundary while allowing resource sharing. My research focused on how an initial compromise in a child domain or a secondary forest can be leveraged to escalate privileges across these boundaries.

- **Research:** Using my lab environment, I simulated a multi-domain forest to study "**Trust Transitivity**." I analyzed how certificate-based authentication interacts with the **Global Catalog** and **Foreign Security Principals**.
- **The Attack Path:** I investigated how forged certificates can be used to request **Cross-Realm Ticket Granting Tickets (TGTs)**. If an attacker compromises a CA in a trusted domain, they can potentially forge identities that the trusting domain's KDC will accept, effectively by passing the security isolation intended by the Forest boundary.
- **Significance:** This demonstrates that a single misconfiguration in a "lesser" domain can compromise the "Root" domain, making AD CS security a forest-wide concern.

7.2. Golden gMSA (Group Managed Service Accounts)

As organizations move away from standard service accounts to **gMSAs** for automated password management, they introduce a new high-value target for persistence.

- **Technical Breakdown:** gMSAs are unique because their 240-character passwords are managed and rotated automatically by the Domain Controller. Access to these passwords is governed by the `msDS-AllowedToRetrieveManagedPassword` attribute.
- **The "Golden gMSA" Attack:** This technique involves extracting the **Key Identification (KID)** and the password blob directly from the Active Directory database. By obtaining the root key used to derive gMSA passwords, an attacker can "calculate" the current password of any gMSA at any time.
- **Impact on Persistence:** Unlike standard user accounts, gMSAs are often excluded from aggressive monitoring or MFA requirements. Compromising a "Golden gMSA" allows an attacker to maintain long-term, highly silent access to critical services (like SQL or IIS) even if other administrative passwords are changed.

7.3. Cross-Domain & Cross-Forest Trust Exploitation

In modern enterprise environments, the security boundary is rarely limited to a single domain. Active Directory Trusts are designed to facilitate resource sharing, but they also introduce complex attack surfaces, especially when integrated with AD CS. My research focused on how these trust primitives can be abused to move laterally across the forest.

1. Core Trust Attack Primitives

Before integrating AD CS, it is essential to understand the foundational Kerberos-based trust attacks:

- **SID History Injection:** Within a forest, Domain Controllers honour the SID History attribute in the Kerberos PAC without filtering. An attacker who compromises a child domain can inject the Enterprise Admins SID into their ticket, granting them forest-root administrative access.
- **Trust Key Golden Tickets:** By extracting the inter-realm trust key (the password hash of the trust account), an attacker can forge cross-realm TGTs for any user in the trusting forest.
- **Unconstrained Delegation:** Attackers can harvest TGTs from users of a trusted forest who authenticate to a compromised machine that holds unconstrained delegation.

2. AD CS as a Trust Amplifier

The introduction of X.509 certificates creates a new authentication primitive that often bypasses traditional Kerberos-based security boundaries like **SID Filtering**.

- **ESC8 Across Trusts (NTLM Relay):** This is a critical cross-boundary attack. Because NTLM relay is a raw protocol-level attack (HTTP/SMB), it does not respect Kerberos trust boundaries. An attacker in Forest A can coerce a Domain Controller in Forest B (via PetitPotam) and relay that authentication to Forest B's CA to obtain a DC-level certificate.
- **The NTAAuth Store Vulnerability:** For PKINIT (certificate-based logon) to work, the issuing CA must be present in the NTAAuthCertificates object of the domain. If a CA from Forest A is accidentally or intentionally added to the NTAAuth store of Forest B, any certificate issued by Forest A is implicitly trusted for authentication in Forest B, leading to a total collapse of the forest boundary.
- **SID Extension & Filter Bypass:** Certificates can embed the `szOID_NTDS_CA_SECURITY_EXT` extension containing a user's SID. Because the Domain Controller reads the SID directly from the certificate during PKINIT, the traditional **SID Filter** applied to cross-forest Kerberos traffic may be bypassed, allowing an attacker to "inject" a privileged SID directly via a certificate.

3. Mitigation & Defensive Controls

To secure trust boundaries against certificate-based attacks, the following controls are required:

- **Enforce SID Filtering:** Use `netdom trust /enablesidfiltering:yes` to prevent SID history injection across forest boundaries.
- **Audit NTAAuth Store:** Regularly review the NTAAuthCertificates container in Active Directory to ensure no unauthorized or foreign CAs are present.
- **Harden Web Interfaces:** Disable Web Enrollment or strictly enforce HTTPS and **Extended Protection for Authentication (EPA)** to neutralize ESC8 relay attacks.
- **Restrict Template ACLs:** Enrollment rights on sensitive templates should never be granted to broad groups like "Authenticated Users," as this often includes users from trusted foreign forests.

7.4. Case Study: Cross-Domain Exploitation (XYZ.COM → ABC.COM)

To evaluate the real-world impact of trust vulnerabilities, I conducted a simulated attack between two domains: **XYZ.COM** and **ABC.COM**. This case study demonstrates how compromising a single domain controller can lead to the total takeover of a trusting domain by abusing the inter-realm trust account.

1. Trust Topology & Setup

In this scenario, a bidirectional trust exists between the two domains. **ABC.COM** stores a trust account (xyz\$) containing a shared secret (NTLM/AES keys) used for Kerberos inter-realm validation. The objective was to demonstrate that a compromise of the trust-initiating domain allows for full administrative control over the trusting domain.

2. Technical Execution

The attack was executed in three distinct phases:

1. **Credential Extraction:** Using administrative access on a compromised DC, I extracted the trust account credentials from the **Local Security Authority (LSA)** database. By patching the LSA process memory, I retrieved the RC4 and AES keys for the xyz\$ trust account.
2. **Inter-Realm TGT Acquisition:** Leveraging the extracted NTLM hash, I utilized **Rubeus** to request a **Ticket Granting Ticket (TGT)** directly from the MULTINATIONAL.COM Key Distribution Center (KDC). Because the trust account is a valid domain principal, I was able to bypass the need for traditional referral tickets and inject the TGT directly into the current session.
3. **Privilege Escalation & Domain Takeover:** With a valid TGT for the trust account, I possessed sufficient LDAP write permissions in the target domain. I successfully:
 1. Created a new, attacker-controlled user account in MULTINATIONAL.COM.
 2. Modified the **Domain Admins** group membership to include this new user.

3. Attack Mechanics & Findings

This research highlights several critical security flaws in standard trust configurations:

- **Trust Account Over-Privilege:** Trust accounts often possess elevated LDAP permissions required for cross-domain operations. These rights can be abused to modify domain objects if the account is compromised.
- **Secret Symmetry:** Because both domains in a trust relationship store identical secrets, a breach in either environment exposes the entire trust boundary.
- **Lack of Selective Authentication:** Without **Selective Authentication** enabled, the trust account can perform privileged operations across the entire target domain without granular access controls.

4. Conclusion & Impact

The successful execution of this attack resulted in **complete domain administrative control** over ABC.COM. This case study proves that trust relationships are not just authentication bridges but significant security risks. It underscores the necessity of treating any trusted domain as a part of the local **Tier-0** security boundary.

8. Monitoring & Detection Analysis (Wazuh)

This chapter presents the defensive perspective of the project and evaluates the detectability of AD CS reconnaissance within a monitored Active Directory environment. The primary objective was to determine whether targeted enumeration techniques could be distinguished from legitimate LDAP activity by analyzing the telemetry generated within a Wazuh SIEM deployment.

8.1. Monitoring Architecture

Before conducting the research, a centralized logging pipeline was implemented to ensure visibility into both LDAP activity and PowerShell execution.

The monitoring environment consisted of the following components:

- **Wazuh Agent Integration:** Wazuh agents were deployed on both the Domain Controller and Certificate Authority to forward Windows event logs to the Wazuh manager in real time.
- **Sysmon (System Monitor):** Sysmon was configured using a community-based configuration focused on process creation, command-line logging, and network connections to improve visibility into administrative tooling and PowerShell execution.
- **Advanced Audit Policies:** Additional Windows audit policies such as “Audit Directory Service Access” were manually enabled to generate LDAP-related telemetry required for the research.

The monitoring environment was intentionally configured to remain realistic. Rather than enabling excessive auditing that would be impractical in production environments, only telemetry considered operationally useful within a SOC context was collected.

8.2. LDAP Enumeration Telemetry Analysis

One of the primary goals of this research was to compare the telemetry generated by publicly available AD CS enumeration tools against the custom tooling developed during the internship.

Initial assumptions suggested that tools such as Certipy would generate a substantial amount of detectable LDAP activity. However, practical testing demonstrated that modern AD CS enumeration tooling is significantly quieter than expected. During testing, Certipy consistently generated only a single LDAP-related Event ID 1644 (“LDAP search operation”) associated with queries against certificate-related Active Directory objects.

The custom PowerShell and Python tooling developed during the internship generated even less observable telemetry. Unlike Certipy, which retrieves larger portions of the certificate template container before performing client-side analysis, the custom scripts relied on highly targeted LDAP queries that delegated filtering directly to the Domain Controller. Only certificate templates matching specific vulnerable attributes were returned, significantly reducing unnecessary LDAP enumeration activity.

Because the filtering logic was handled server-side and the queries completed within only a few milliseconds, no alertable Event ID 1644 entries were generated under the configured monitoring conditions. As a result, the custom tooling blended into normal directory activity considerably more effectively than traditional enumeration approaches.

An additional finding was that aggressive LDAP-based alerting is operationally difficult to implement in realistic environments. Even within a small laboratory containing a single client system, legitimate Windows services continuously generated LDAP queries in the background. Lowering detection thresholds further would therefore introduce a substantial number of false positives and likely result in alert fatigue within a SOC environment.

8.3. Critical Telemetry & Key Event Ids

During testing, Event ID 1644 ("LDAP search operation") became the primary telemetry source used to identify LDAP-based reconnaissance activity against Active Directory Certificate Services.

Unlike the initially expected Event ID 4662 ("Directory Service Access"), Event ID 1644 provided significantly more reliable visibility into LDAP query behavior generated by AD CS enumeration tooling. The event exposed details regarding the LDAP query itself, including the targeted Distinguished Names (DNs) and the originating client IP address.

The event proved particularly valuable for identifying queries against certificate-related Active Directory containers such as:

- CN=Certificate Templates,CN=Public Key Services
- CN=Enrollment Services,CN=Public Key Services

PowerShell Script Block Logging (Event ID 4104) was also evaluated during testing. While PowerShell activity remained visible when logging was explicitly enabled, the generated telemetry was considerably less useful for reliably distinguishing targeted AD CS reconnaissance from legitimate administrative activity.

8.4. Wazuh Rule Customization

Custom Wazuh rules were developed to detect LDAP search operations associated with reconnaissance activity against Active Directory Certificate Services.

Rather than relying on complex detection logic, the implemented approach focused on identifying Event ID 1644 entries generated within the "Directory Service" event channel. The detection logic was intentionally kept simple to reduce parsing complexity and improve reliability.

The following custom Wazuh rule was implemented:

```
<group name="ad_recon">

  <rule id="110000" level="0">
    <if_sid>60000, 60009</if_sid>
    <field name="win.system.channel">^Directory Service$</field>
    <description>Group of Windows rules for Directory Service
channel</description>
  </rule>

  <rule id="110001" level="5">
    <if_sid>110000</if_sid>
    <field name="win.system.eventID">^1644$</field>
    <description>AD DS: LDAP search operation (Event 1644)</description>
    <group>windows,ad_recon,</group>
  </rule>

</group>
```

8.5. Preliminary Findings (The Detection Gap)

The testing phase revealed a substantial gap between theoretical detectability and practical detection capability within default Windows logging and Wazuh monitoring configurations.

- **Finding 1:** Modern AD CS enumeration tooling such as Certipy generated significantly less telemetry than initially expected, often producing only a single detectable LDAP-related audit event during enumeration activities.
- **Finding 2:** The custom PowerShell and Python scripts developed during this internship generated virtually no alertable telemetry under realistic monitoring conditions due to their highly targeted and short-lived LDAP queries.
- **Finding 3:** Detecting LDAP-based reconnaissance purely through frequency-based monitoring is impractical because legitimate Windows systems continuously generate LDAP traffic, even in small environments, resulting in a high risk of alert fatigue.
- **Finding 4:** Without Sysmon and advanced audit policy configuration, targeted reconnaissance performed through native Windows functionality becomes extremely difficult to distinguish from legitimate administrative activity.
- **Finding 5:** Advanced trust-related attacks such as Golden gMSA abuse and trust account manipulation remained largely invisible under default logging configurations and required additional object access auditing to produce meaningful telemetry.

9. Conclusion

The modern cybersecurity landscape has increasingly shifted from defending the network perimeter toward securing the identity perimeter. This internship at Refracted Security focused on researching the security implications of Active Directory Certificate Services (AD CS), with particular emphasis on offensive enumeration techniques, trust relationships, and defensive visibility.

The research demonstrated that AD CS introduces a large and often underestimated attack surface within Active Directory environments. Through the development of targeted PowerShell and Python-based tooling, multiple attack paths ranging from ESC1 to ESC16 were analyzed and validated within a controlled lab environment. The project further demonstrated that many AD CS attack paths can be enumerated with significantly lower telemetry generation than commonly used industry tooling, increasing the difficulty of detection in environments with limited monitoring capabilities.

Additional research into Cross-Domain Trust abuse and Golden gMSA techniques highlighted that identity-based vulnerabilities are not isolated to individual systems or domains. Instead, they frequently impact trust relationships across entire Active Directory forests, allowing attackers to bypass traditional segmentation boundaries and extend compromise across interconnected environments.

The monitoring and detection analysis performed using Wazuh confirmed the existence of a substantial detection gap surrounding certificate-based attacks and low-noise enumeration techniques. The findings showed that default logging configurations are often insufficient for detecting advanced AD CS abuse, emphasizing the importance of enhanced telemetry collection, certificate auditing, and custom detection engineering within enterprise environments.

This internship provided extensive practical experience in both offensive security research and defensive monitoring methodologies. Beyond the technical implementation itself, the project reinforced the importance of understanding identity infrastructure as a critical security boundary within modern enterprise networks. The experience gained during this internship forms a strong foundation for further specialization in Red Team operations, detection engineering, and identity-focused security research.

9.1. Acknowledgements

I would like to express my sincere gratitude to the team at Refracted Security for providing a technically challenging and motivating internship environment.

Special thanks go to my internship mentor, Yannick Hendrickx, for the continuous technical guidance, the freedom to shape the direction of the research, and the extensive knowledge shared throughout the internship period. The high-level technical discussions and practical insights into Active Directory and Red Teaming methodologies were instrumental in the success of this project and significantly contributed to my professional development.

Lastly, I would like to thank Thomas More and the instructors of the Ethical Hacking trajectory for providing the academic foundation that made this specialized research possible.

10. References

Christensen, L., & Schroeder, W. (2021). Certified Pre-Owned: Abusing Active Directory Certificate Services. SpecterOps. <https://specterops.io/blog/2021/06/17/certified-pre-owned/>

Microsoft. (2024). Active Directory Certificate Services documentation. Microsoft Learn. <https://learn.microsoft.com/en-us/windows-server/identity/ad-cs/>

Microsoft Sysinternals. (2024). Sysmon documentation. Microsoft Learn. <https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon>

Oliver Lyak. (2025). Certipy documentation. GitHub. <https://github.com/ly4k/Certipy/wiki/06-%E2%80%90-Privilege-Escalation>

The Hacker Recipes. (2025). Active Directory Certificate Services abuse techniques. <https://www.thehacker.recipes/ad/movement/adcs/>

Wazuh. (2025). Wazuh documentation. <https://documentation.wazuh.com>