RADIOACTIVE DIRECTORY

25 YEARS OF GIVING UP...

Human Resources R&D

> PERSONNEL RECORD

C:\Users\Dweller> whoami

DESIGNATION: Richard Belisle

CLASS: Full-Spectrum Security Nerd

AFFILIATION: MSU Denver | radioactivedirectory.com

MISSION: Bring knowledge to the irradiated ruins of AD

CONTACT: ribelisle@msudenver.edu

LINKEDIN

[SCAN TO CONNECT]

The Fallout: AD is TOXIC

T **Targeted**

Prime target for ransomware gangs and nation-state actors

O **Omnipresent**

Ubiquitous across enterprise environments. It's going nowhere.

X **Complex**

Hidden attack paths through permissions and delegations. Insecure by default.*

I **Isolated**

Knowledge scattered across talks, whitepapers, and blog posts

C **Core Infrastructure**

Controls authentication, authorization, and access to everything

What's a vulnerability?

Who's identifying and managing non-CVE vulnerabilities in your organization?

A Brief History

- **1993** Windows NT 3.1 introduced NTLMv1
- **1993** Kerberos v5 released
- **2000** Server 2000
- **2003** Server 2003
- **2008** Server 2008
- **2013** Server 2012 R2
- **2016** Server 2016
- **2025** 25 years of AD

Authentication Protocols

NTLM

Legacy challenge-response

- NTLMv1 — weak, easily cracked
- NTLMv2 — better, still relayable
- No mutual authentication
- Hash passed, not password
- Not-sensitive

Kerberos

Ticket-based (AD default)

- TGT from KDC (AS-REQ/REP)
- Service tickets (TGS-REQ/REP)
- Mutual authentication
- Sensitive (time, line of sight)

PKINIT

Certificate-based Kerberos

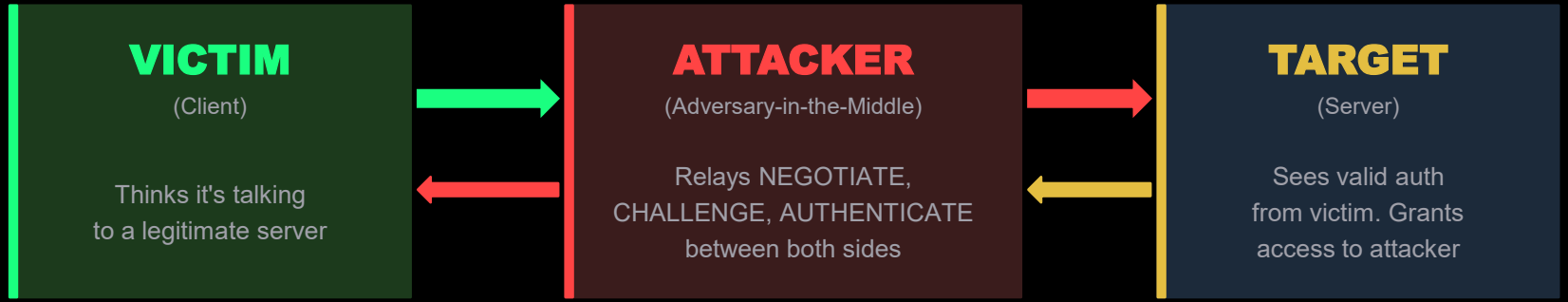
- Smart card authentication
- Uses X.509 certificates
- Ties into ADCS
- Shadow Credentials abuse

1. Client sends NEGOTIATE with supported flags and version. No credentials yet.
2. Server replies with 8-byte nonce (one-time random number to prevent replay) + agreed-upon flags for what it will/won't accept.
3. v1: DES-encrypts challenge with password hash (weak, crackable). v2: HMAC-MD5 with timestamp + client nonce (stronger, but still relayable).

The NTLM 3-way handshake: Negotiate > Challenge > Authenticate

NTLM Relay

NTLM connects via challenge-response, but all three messages can be proxied. Without channel binding, an adversary in the middle can own the entire exchange undetected.



WHY IT WORKS

T1557.001

Fully Proxiable

All the messages can be relayed. Attacker forwards the real server's challenge to the victim, victim signs it legitimately, attacker forwards it back.

Kerberos Is Different

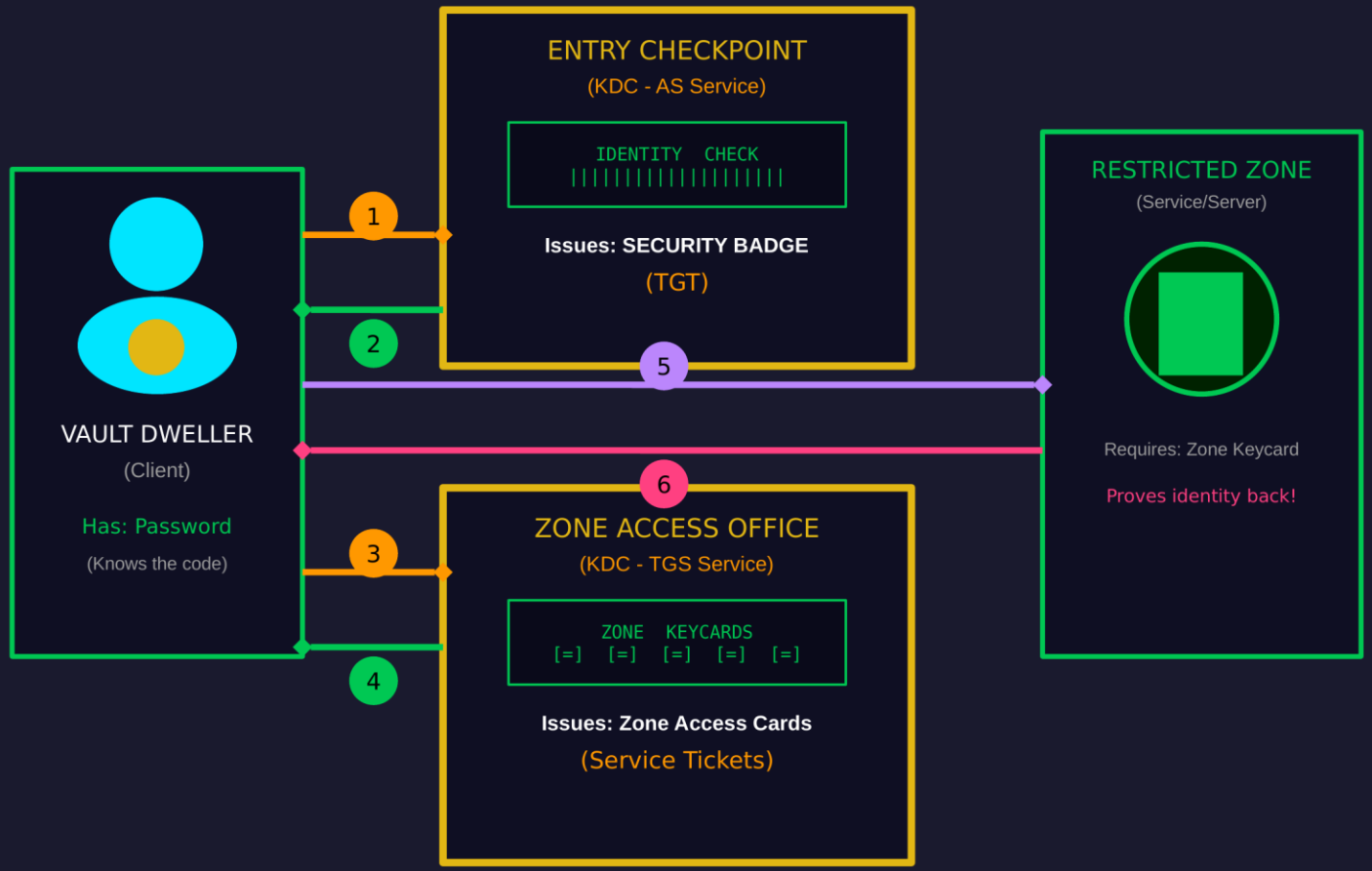
Kerberos tickets are SPN-specific. A ticket for FILE/srv1 cannot auth to LDAP/dc1. Cross-service relay is not possible.

LDAP Signing ≠ LDAPS

LDAP signing covers port 389 only. LDAPS (636) is a separate TLS channel. Channel binding (EPA) is required to close the LDAPS relay path.

Rarely Enforced

SMB signing, LDAP channel binding, and EPA all exist — but off by default on member servers. DCs require SMB signing since 2000. Enforce all three.



CLEARANCE PROTOCOL

1. **Show ID (password)**
   AS-REQ with username. Password derives key (never sent on wire).
2. **Get security badge**
   AS-REP: TGT + session key. Time-limited (10h, 7-day renewal).
3. **Present badge**
   TGS-REQ with TGT + authenticator + target SPN. No password.
4. **Get zone keycard**
   TGS-REP: Service Ticket + session key. Repeatable while TGT valid.
5. **Access the zone!**
   AP-REQ with Service Ticket. Server decrypts, validates, grants access.
6. **Zone confirms identity**
   AP-REP: Server returns timestamp encrypted with session key. This is mutual authentication.

TGT = Security Badge Service Ticket = Zone Keycard KDC = Key Distribution Center

TAKE THE TICKET!

SMB FILE ACCESS — NTLM PATH

PROTOCOL STACK

L6 File Access
Read/write files over authenticated SMB session

L5 Authorization
Server reads token, checks SIDs against share/NTFS ACLs

L4 SMB Session
Negotiate dialect, establish session, present NTLM auth

L3 NTLM Auth
Challenge-response directly with server — no KDC, no SPN

L2 Name Resolution
DNS, LLMNR, NBT-NS, mDNS — no auth on resolution

L1 TCP/IP (445)
Workstation → File Server (Port 445)

↑ Each layer depends on layers below

ATTACK SURFACE

L6 — **Authorization**
ACL misconfiguration grants excessive access — NTLM auth doesn't protect your ACLs

L5 — **Pass-the-Hash**
NT hash is a reusable credential. No plaintext needed — PtH bypasses this layer entirely

L4 — **SMB Session**
Relay succeeds if SMB signing not required — relayed session is fully authenticated as victim

L3 — **NTLM Auth**
Response not bound to destination — works against any server. Hash captured for offline cracking

L2 — **Name Resolution**
LLMNR/NBT-NS/mitm6 poisoning — attacker intercepts before connection is established

L1 — **TCP Connect**
Victim connects to attacker once name resolution is poisoned

Relay possible at every layer — SMB signing is your only defense

SMB FILE ACCESS — KERBEROS PATH

PROTOCOL STACK

L6 File Access
Read/write files over authenticated SMB session

L5 Authorization
Server reads PAC, checks SIDs against share/NTFS ACLs

L4 SMB Session
Present Kerberos ticket, negotiate SMB dialect

L3 Kerberos Auth
Get service ticket for cifs/fileserver SPN from KDC

L2 DNS Resolution
"fileserver" → IP via AD-integrated DNS + SPN lookup

L1 TCP/IP (445)
Workstation → File Server (Port 445)

↑ Each layer depends on layers below

ATTACK SURFACE

L6 — Authorization
ACL misconfiguration grants excessive access — Kerberos doesn't fix your ACLs

L5 — PAC Manipulation
PAC holds group memberships — historically forged (MS14-068), now validated by DC

L4 — Kerberoasting
Any authed user can request a service ticket for any SPN and crack the hash offline

L3 — Kerberos Auth
NOT relayable — ticket SPN-locked, encrypted with target's key. KDC-brokered mutual auth

L2 — DNS Resolution
Poisoning degrades to DoS only — SPN binding means attacker can't decrypt or reuse the ticket

L1 — TCP Connect
Client may connect to attacker — Kerberos auth fails. Attacker can't decrypt the service ticket

Kerberos mitigates auth relay — ACLs and DNS remain your responsibility

WHEN KERBEROS FAILS: NTLM FALLBACK

Kerberos requires a resolvable SPN and mutual authentication. When these conditions aren't met, Windows silently falls back to NTLM — changing the entire attack surface.

NTLM FALLBACK TRIGGERS

IP address used instead of hostname
\\192.168.1.50\share

Hostname not in DNS / no SPN registered
Mistyped or stale DNS records

Cross-forest without trust or SPN mapping
External partner resources

Legacy application hardcoded to NTLM
Older IIS, SQL, or LOB apps

Kerberos port blocked (88/TCP)
Network segmentation issues

Clock Skew or No Line of Sight to DC
More than 5 mins,

NTLM ATTACK SURFACE

Relay attacks
Forward auth to another host — no cracking needed. Mitigated by SMB signing + EPA.

Credential capture
Responder / mitm6 capture NTLMv2 hashes for offline cracking.

No mutual authentication
Client cannot verify server identity. Kerberos provides this by design.

Pass-the-Hash
Stolen NT hash authenticates directly — no password needed.

Downgrade to NTLMv1
If LmCompatibilityLevel not enforced, attacker negotiates weaker protocol.

KEY TAKEAWAY: The Kerberos path (slide 1) is not relayable and provides mutual auth. When NTLM fallback occurs (slide 2), the session becomes relayable, credential capture becomes possible, and the server cannot be verified.

NTLM & Kerberos -- Quick Wins

Zero risk, high impact. No change management ticket needed.

① QUICK WIN

Kerberos * -- Covered in Detail Later

Kerberoasting, delegation, and ticket attacks get dedicated slides. Key event IDs to enable now: 4769 RC4 (etype 0x17) = active roast. Honeypot SPN: any TGS request for it is an instant IOC. *Full coverage: Kerberoasting section.

```
Event 4769, etype 0x17 = roast attempt
Event 4768 bulk from one account = recon
Event 4771 = Kerberos pre-auth failed
```

② AUDIT FIRST

Enable NTLM Logging on DCs

NOT ON BY DEFAULT. Enable NTLM Operational log on DCs before blocking anything. Event 8001 = auth to domain server, 8002 = passthrough auth, 8003 = auth to DC. Map every NTLM source -- printers, legacy apps, vendor appliances -- before you touch a GPO.

```
Audit NTLM authentication
Events 8001/8002/8003 on DCs
```

③ AUDIT

Enable LDAP Logging on DCs

NOT ON BY DEFAULT. Set NTDS Diagnostics registry key to enable Events 2886-2889 (unsigned binds) and 3040-3041 (channel binding). Event 2889 gives you client IP for every unsigned LDAP bind (Type 0 = SASL, Type 1- Simple(cleartext). Zero-cost visibility before you enforce anything.

```
HKLM\SYSTEM\CurrentControlSet\
Services\NTDS\Diagnostics
"16 LDAP Interface Events" = 2
```

④ ENFORCE

NTLMv2 Minimum?*

Once audit data confirms NTLMv1 sources are gone, set LmCompatibilityLevel=5. Kills NTLMv1 and LM responses entirely. Verify via Event 4624 (LmPackageName = NTLM V1). Pilot one OU before domain-wide rollout.

```
GPO: Network Security:
LAN Manager auth level
Send NTLMv2 only
FOCUS ON SIGNING AND CHANNEL BINDING
```

⑤ PROJECT

Audit LDAP Signing + Channel Binding

Events 2886/2887 log unsigned LDAP bind summaries. Event 2889 gives client IP per unsigned bind -- use this to identify every legacy app before enforcing signing. Events 3040/3041 cover channel binding status. Enforce only after mapping all clients.

```
GPO: LDAP server signing
requirements -- Require
+ LdapEnforceChannelBinding = 2
```

⑥ QUICK WIN

ADCS -- Run Locksmith First

Before hardening ADCS, run Locksmith or PingCastle to surface all misconfigured certificate templates (ESC1-ESC13+). Know your full exposure before making changes.

```
Invoke-Locksmith
PingCastle --healthcheck
Review all ESC findings.
```

NTLM: relayable, hash-based, no mutual auth -- every unsigned service is a relay target. Kerberos: ticket-based, SPN-locked, mutual auth -- harder to relay, but tickets and delegation are the attack surface. Map before you block.

T1557 / T1558

NTLM & Kerberos -- Project Work

These need planning and testing. Map your environment before you touch anything.

① UNDERSTAND FIRST

NTLM Relay Attack Surface

Any service accepting NTLM without signing is a relay target: LDAP, ADCS, MSSQL, HTTP, SMB, etc.. NTLM relay does not require cracking -- attacker forwards the handshake in real time. Map exposure before touching any controls. No universal cure likely their own project per protocol.

```
NTLM Operational log -- enable via GPO first
(off by default).  Tells you who is using
NTLM.
PingCastle / Wireshark (LLMNR+NBNS filter).
```

② PROJECT

SMB Signing -- Member Servers

DCs already require SMB signing. Member servers and workstations are the gap. Caution: blocking port 445 inbound to DCs from workstations breaks SYSVOL and NETLOGON -- GPO processing depends on it. Enforce signing; do not blanket-block 445 to DCs but block between workstations.

```
GPO: Microsoft network
server: Digitally sign
communications (always)
```

③ PROJECT

LDAP Signing + Channel Bind

Two separate controls -- both required. Signing stops relay on port 389. EPA binds auth to the TLS session on LDAPS 636. One without the other leaves a path open.

```
GPO: LDAP server signing
requirements -- Require.
Enable both.
```

④ PROJECT

Resolve Locksmith Finding (ADCS) - Priority

Locksmith surfaces ESC vulnerabilities ranked by priority.

```
Invoke-Locksmith
PingCastle --healthcheck
Focus on ESC1, ESC4, ESC6, ESC8
```

⑤ PROJECT

gMSA for Service Accounts

AD manages the password -- 240 random chars, rotated every 30 days. Effectively mitigates Kerberoasting: ticket is requestable but offline cracking is infeasible. Verify app support first.

```
New-ADServiceAccount
-Name svc_gmsa
-PrincipalsAllowedToRetrieve
```

⑥ Phased

NTLM Block by OU / Group / Tier

Never block NTLM domain-wide, do enforce NTLMv2. Use 8001/8002 data to identify safe targets. Block by tier recommended. Research and clinical systems last -- and maybe not worth it.

```
GPO: Restrict NTLM:
Incoming NTLM traffic ← to that target
Deny all (per OU)

NOT: "NTLM authentication" - this is nuclear
```

The thread: audit first, block surgically. In higher ed, legacy systems are the constraint -- plan around them.

T1557 / T1558

Kerberos Delegation

THE PROBLEM

Kerberos authenticates users to services. But what if a service needs to call another service on the user's behalf?

User → Web App → SQL Server

Kerberos tickets are non-transferable. The web server can't hand the user's ticket to SQL Server.

⚠ Legitimate feature. Enormous attack surface.

THREE FLAVORS

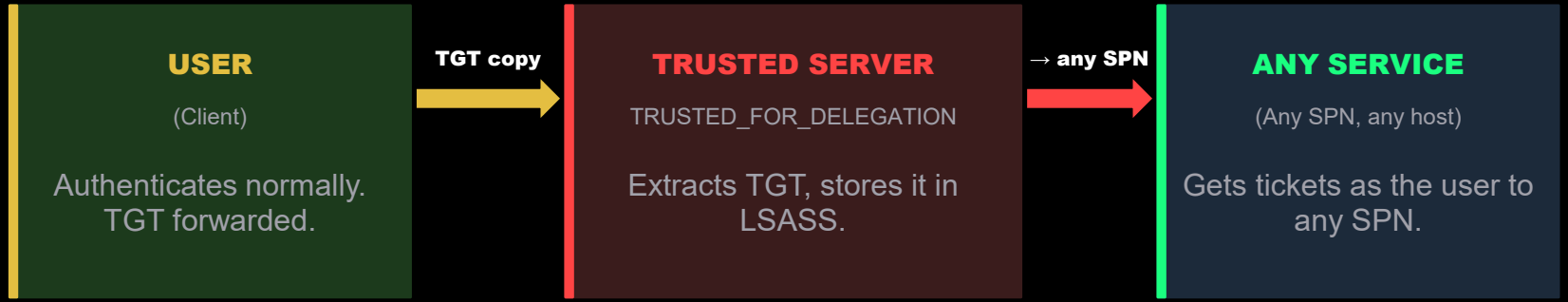
Unconstrained (2000)
Server caches user TGTs. Can delegate to any service anywhere. TGT theft → full domain pivot.

Constrained (2003)
Scoped to specific SPNs via S4U2Proxy. DA configures trust. Compromise the delegating server → impersonate any user to those SPNs.

RBCD (2012 R2)
Resource defines who can delegate to it (msDS-AllowedToActOnBehalfOfOtherIdentity). Write access to that attribute → impersonate anyone. No DA needed.

Unconstrained Delegation

A server with TRUSTED_FOR_DELEGATION caches every TGT that touches it — and can use them to impersonate any user to any service.



HOW IT WORKS

T1558

TGT embedded by KDC

When user requests a ticket to a TRUSTED_FOR_DELEGATION server, the KDC sets OK-AS-DELEGATE and embeds a forwardable TGT copy inside the service ticket.

TGT cached in LSASS

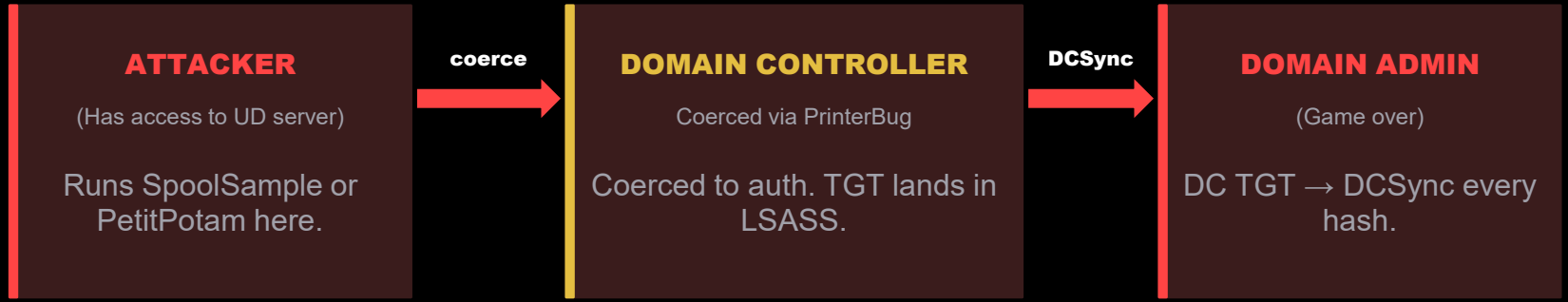
The server extracts the TGT from the service ticket and stores it in memory. Every user who authenticates leaves their TGT behind — accessible to any process on that server.

Impersonate anyone, anywhere

The server uses cached TGTs to request service tickets as any user to any SPN. No constraints. If the DC's TGT is in that cache, you own the domain.

Unconstrained Delegation — Attack

Coerce any privileged machine account to authenticate to your unconstrained server — catch the TGT, own the domain.



ATTACK CHAIN

T1187

Find UD servers

BloodHound: MATCH (c:Computer {unconstraineddelegation:true}) — every hit is a potential pivot point. DCs always have this flag but aren't useful targets.

Coerce DC authentication

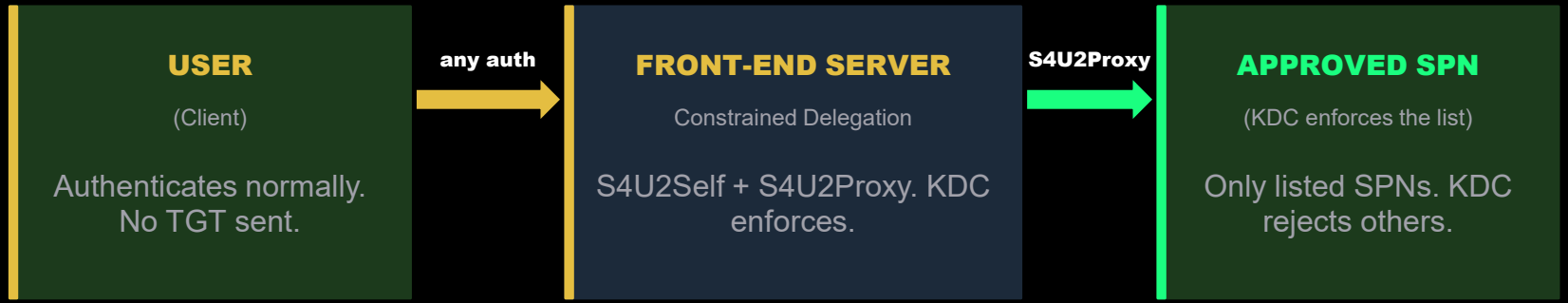
PrinterBug (MS-RPRN) or PetitPotam (MS-EFSRPC) force the DC's machine account to authenticate to your server. Unauthenticated on older systems.

Extract TGT → DCSync

Rubeus monitor /targetuser:DC$. Catch the TGT. Pass-the-Ticket to impersonate DC. Run DCSync. Every credential in the domain is now yours.

Constrained Delegation

S4U2Self + S4U2Proxy — the server impersonates users using its own credentials, but only to a pre-approved list of SPNs. No TGT ever leaves the user.



HOW IT WORKS

T1558

S4U2Self — no TGT needed

Server calls S4U2Self to obtain a service ticket to itself on behalf of the user — using the server's own machine credentials. User's TGT never touches this server.

KDC enforces the list

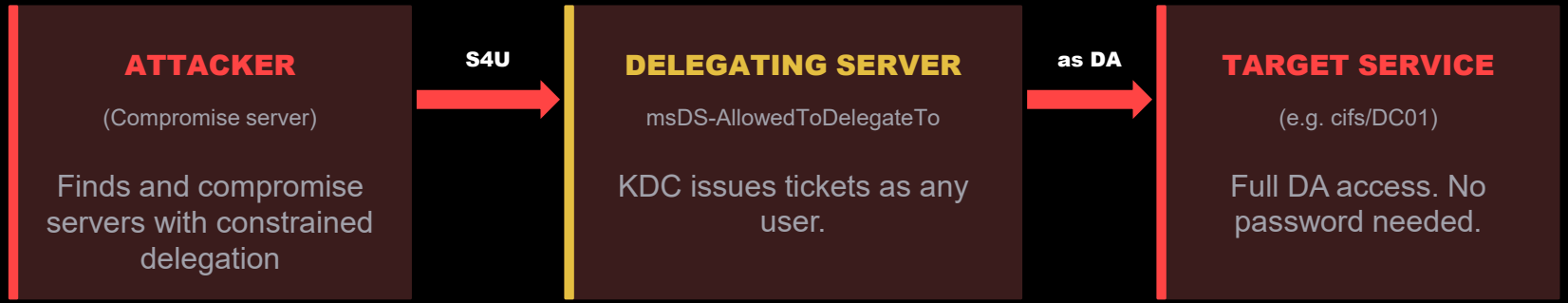
Server calls S4U2Proxy with that ticket to request a ticket to the target SPN. KDC checks msDS-AllowedToDelegateTo. If the SPN isn't listed, the request is denied.

Still dangerous if compromised

Compromise the front-end server's machine account → S4U2Self as any user (including DA) → S4U2Proxy to every approved SPN. One server = full access to everything it can delegate to.

Constrained Delegation — Attack

Compromise the delegating server's machine account — then impersonate any user to every SPN in its approved list.



ATTACK CHAIN

T1558.003

Find delegation targets

BloodHound: computers or users with msDS-AllowedToDelegateTo set. Accounts with cifs/DC, ldap/DC, or http/Exchange in the list are especially valuable.

Get machine account creds

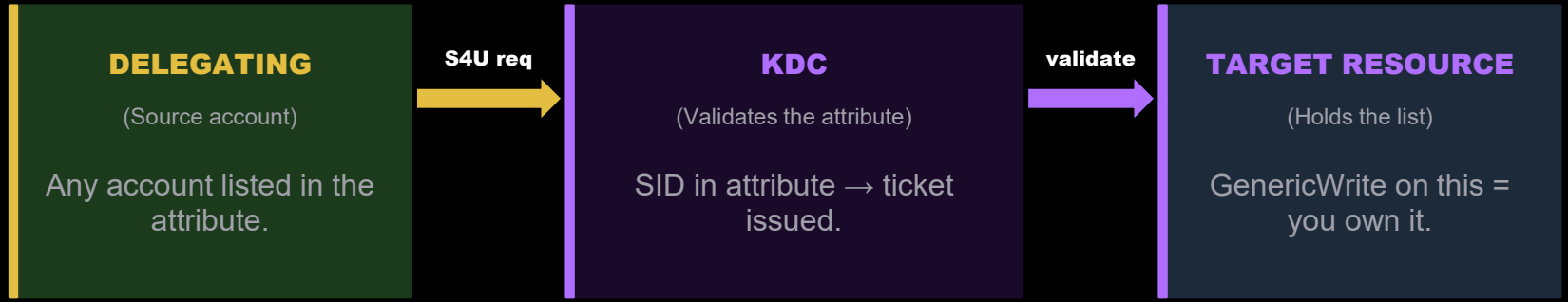
Machine account hash from secretsdump, LSASS dump, or Kerberoasting if it's a service account. The account itself is often low-visibility.

S4U as domain admin

Rubeus s4u /user:server$ /rc4:[hash] /impersonateuser:Administrator /msdsspn:cifs/DC01. Ticket issued. KDC won't stop you — this is by design.

Resource-Based Constrained Delegation

The trust arrow flips — in RBCD, the TARGET resource holds the list of accounts allowed to delegate to it. Not the source.



KEY DIFFERENCES FROM CONSTRAINED DELEGATION

Trust lives on the resource

In constrained delegation, a DA sets msDS-AllowedToDelegateTo on the source. In RBCD, anyone with GenericWrite on the target can set msDS-AllowedToActOnBehalfOfOtherIdentity. No DA needed.

MachineAccountQuota enables it

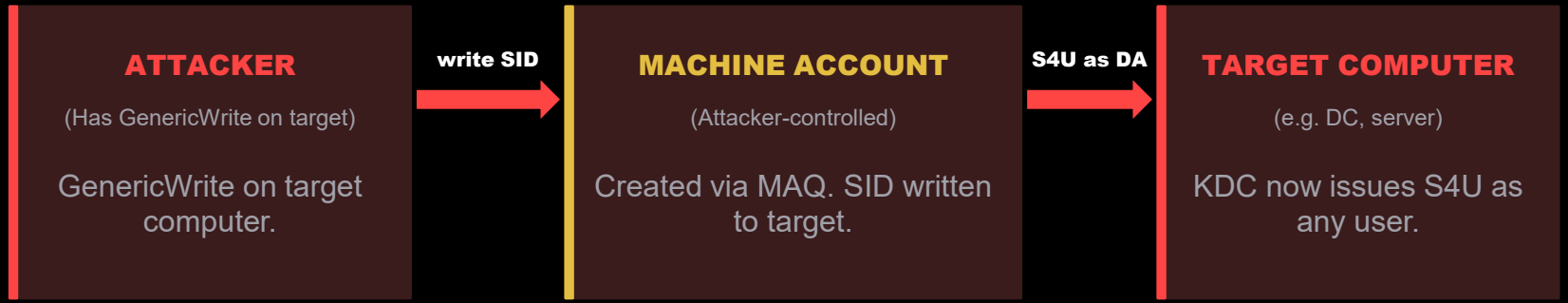
Default MAQ = 10. Any domain user can create up to 10 machine accounts. Attackers create one, write its SID into the target attribute, and use it to S4U as any user.

No TGT cached anywhere

Like constrained delegation, RBCD uses S4U2Self + S4U2Proxy. The user's TGT is never forwarded. This makes it stealthier — no credential material sits in LSASS on the delegating account.

RBCD — Attack

GenericWrite on any computer object = impersonate any user to that computer. No domain admin required.



ATTACK CHAIN

T1558

Find GenericWrite targets

BloodHound: MATCH p=(u)-[:GenericWrite]->(c:Computer). Misconfigurations in ACLs are common. LAPS write access also works. Any write-equivalent permission on the object.

Write the attribute

Set-ADComputer target -PrincipalsAllowedToDelegateToAccount attacker_machine$. Or write the raw SID directly into msDS-AllowedToActOnBehalfOfOtherIdentity via LDAP.

S4U as any user, clean exit

Rubeus s4u /user:attacker_machine$ /impersonateuser:Administrator /msdsspn:cifs/target. Present ticket. Full access as DA. Undo the attribute write afterwards. No persistent artefacts.

Delegation — Controls & Hardening

Most older delegation configs persist, not because of genuine technical necessity

① DO NOW

Protected Users Group

All tier-0 — DAs, EAs, privileged service accounts, etc/. Blocks all delegation, forces AES, disables NTLM. Free protection. Enable today, check for app breakage.

```
-Add-ADGroupMember
'Protected Users'
-Members DA_acct
```

② DO NOW

Sensitive: Not Delegatable

For all administrator accounts that cannot join PUG that may require NTLM. Weaker than PUG but better than nothing.

```
Set-ADUser DA_acct
-AccountNotDelegated $true
```

③ RESTRICT

MachineAccountQuota → 0

Setting MachineAccountQuota to 0 is the right call but do it alongside proper OU delegation, otherwise you'll get pushback from IT and it'll get quietly reverted.

```
Set-ADDomain -Replace
@{'ms-DS-MachineAccountQuota'=0}
```

④ PREFER

RBCD Over Constrained

If delegation is needed, RBCD is the better model. Config lives on the resource, easier to audit, no DA required to configure.

```
PingCastle flags it. ADUC Delegation tab.
Get-ADComputer -Filter
{msDS-AllowedToActOnBehalfOfOtherIdentity -
like '*'}
```

⑤ CAUTION

Constrained Delegation

Acceptable if RBCD isn't feasible. Scope is locked to approved SPNs. Protect the machine account hash — it's equivalent to full access.

```
PingCastle flags it. ADUC Delegation tab.
Get-ADComputer -Filter {msDS-
AllowedToDelegateTo -like '*'}
Event 4769 – alert on S4U tickets.
```

⑥ ELIMINATE

Unconstrained Delegation

No legitimate reason in modern environments excluding DCs. Find every non-DC with TRUSTED_FOR_DELEGATION and remove it. This one requires work.

```
PingCastle: flags directly in risk report.
Get-ADComputer -Filter {TrustedForDelegation -
eq $true}
```

Quick wins today: Protected Users, MAQ to 0. Bigger lift: audit and remove unconstrained delegation.

T1134.001

Common Attacks

Eight attack classes -- most require only valid domain credentials to start.

T1558.004

AS-REP Roasting

Accounts with Kerberos pre-auth disabled hand out encrypted AS-REP blobs to anyone -- no credentials needed. Request one, crack the hash offline. Simpler in execution than Kerberoasting, less common.

T1550.002

Pass-the-Hash

NTLM accepts the hash as proof of identity -- not just the password. Dump LSASS or SAM, authenticate directly with the hash. No cracking required. Lateral movement in seconds.

T1550.003

Pass-the-Ticket

Steal a live TGT or service ticket from memory with Mimikatz. Inject it into your session and become that account for the ticket's remaining lifetime. No hash, no crack -- just reuse.

T1003.006

DCSync

With Replicating Directory Changes rights, call the legitimate DC replication protocol and pull NTLM hashes and Kerberos keys for any account -- including KRBTGT. Never touch the DC physically.

T1558.001

Golden Ticket

With the KRBTGT hash, forge any TGT -- any user, any group, any lifetime. Survives password resets on user accounts. Only rotating KRBTGT twice kills it. Full domain persistence. Consider periodic KRBTGT password resets.

T1558.002

Silver Ticket

Forge a service ticket using only the target service account's hash. More targeted than Golden Ticket -- and much harder to detect since it never touches the DC after creation.

T1222.001

ACL Abuse

GenericWrite, WriteDACL, GenericAll on AD objects. Reset passwords, add to groups, configure delegation -- all without touching the target system. BloodHound makes these paths trivial to find.

T1558.003

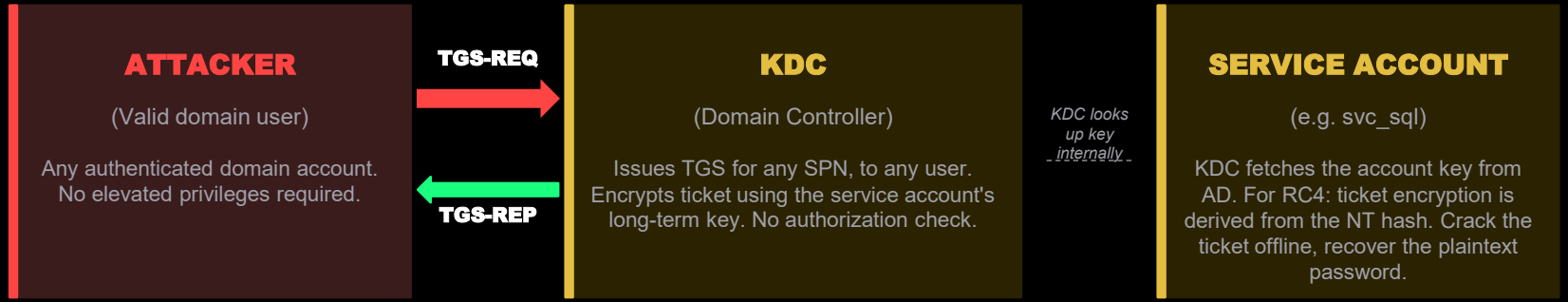
Kerberoasting

Request TGS tickets for any SPN as any domain user -- the KDC hands them out by design. Tickets are encrypted with the service account's password hash. Crack offline, no noise on the wire.

Golden and Silver Tickets require prior compromise to obtain hashes. Everything else starts from valid domain credentials.

Kerberoasting

Any domain user can request a service ticket for any SPN -- the KDC encrypts it with the service account's key and never checks authorization.



ATTACK CHAIN

T1558.003

① Enumerate + Request

Query AD for accounts with SPNs. Request TGS tickets via Rubeus or GetUserSPNs.py. The KDC hands them out -- this is legitimate Kerberos behavior.

② Extract + Crack Offline

Export the encrypted ticket from memory. Run Hashcat against it offline -- no DC contact, no failed logons, no lockouts. GPU cracks RC4 in minutes.

③ RC4 Downgrade

Even if the account supports AES, request RC4 explicitly (etype 0x17). KDC honors it unless msDS-SupportedEncryptionTypes=24 is set. Rubeus --rc4opsec automates this.

Kerberoasting -- Mitigations

Layered controls -- each one raises the bar. Stack them.

① BEST FIX

gMSA for Service Accounts

AD manages the password -- 240 random chars, rotated every 30 days. Ticket is still issuable but offline cracking is computationally infeasible. Verify app support first.

```
New-ADServiceAccount -Name svc
-PrincipalsAllowedToRetrieve
ManagedPassword computer$
```

② ENFORCE

AES-Only per Account

Set msDS-SupportedEncryptionTypes=24 on each service account. KDC refuses RC4 for that SPN -- closes the downgrade path without domain-wide crypto changes.

```
Set-ADUser svc_acct -Replace
@{msDS-SupportedEncryption
Types=24}
```

③ ENFORCE

Long Passwords on Svc Accts

If gMSA is not feasible, 25+ character random passwords make offline cracking infeasible regardless of encryption type. Rotate on a schedule.

```
Use LAPS or a PAM vault.
Target 25+ chars.
Document rotation schedule.
```

④ QUICK WIN

SPN Hygiene

Remove stale SPNs. Flag privileged accounts with SPNs -- a DA with an SPN is the worst case. Old PasswordLastSet is a priority target for attackers.

```
Get-ADUser -Filter {ServicePrincipal
Name -like '*'} -Properties
ServicePrincipalName,PasswordLastSet
```

⑤ DO NOW

Honeypot SPN Account

Fake service account with an SPN and a strong password. No legitimate system ever requests a ticket for it. Any 4769 is an instant IOC -- zero false positives.

```
Alert on any TGS request
for the honeypot SPN.
Little to no noise or false positives.
```

⑥ MONITOR

Event 4769 -- RC4 + Volume

Alert on 4769 with etype 0x17 for service accounts -- that is a downgrade request. Also flag bulk TGS requests from a single account -- enumeration before roasting is noisy.

```
Event 4769, etype 0x17.
Spike in TGS-REQ volume
from single account = recon.
```

Priority: gMSA where possible, AES-only everywhere else, honeypot always -- it costs nothing.

Advanced Attacks

Chained attacks — most require a foothold and one misconfiguration to reach Domain Admin.

T1187

Coercion Attacks

Force a machine to authenticate outbound via PrinterBug (MS-RPRN), PetitPotam (MS-EFSRPC), or DFSCoerce. No creds needed on unpatched or misconfigured DC. Chains directly into NTLM relay or ADCS ESC8.

T1649

ADCS ESC1

Template allows enrollees to specify a Subject Alternative Name. Any user can request a cert for any UPN -- including Domain Admin. Certificate = persistent credential, survives password reset.

T1649

ADCS ESC8

CA web enrollment (certsrv) accepts NTLM over HTTP. Relay DC$ auth from a coerce to certsrv, get a cert for DC$, then DCSync. Mitigate: HTTPS and EPA.

T1556.007

Shadow Creds

Write msDS-KeyCredentialLink on any object you have GenericWrite over. Adds a certificate-based credential -- authenticate as the target using PKINIT without knowing the password.

T1134

RBCD Abuse

GenericWrite on any computer = write msDS-AllowedToActOnBehalfOfOtherIdentity. S4U2Self + S4U2Proxy produces a service ticket impersonating any non-protected user. No admin needed.

T1558.001

Diamond Ticket

Modify the PAC of a legitimate TGT rather than forging from scratch. Stealthier than Golden Ticket -- ticket started valid. Requires KRBTGT hash but leaves less forensic residue.

T1557.001

ADIDNS Poisoning

Any authenticated user can create wildcard DNS records in AD. Poison responses domain-wide without broadcast protocols -- no LLMNR/NBNS noise. AitM from a standard account.

T1550.002

UnPAC the Hash

Authenticate via PKINIT using a certificate, then request a TGS with the U2U extension. The PAC contains the NT hash of the target account. No cracking -- just math.

These chain. Coercion + ESC8 = unauthenticated DA. RBCD + Shadow Credentials = foothold to DA with one GenericWrite.

☢ Coercion Attacks

Forcing a machine to authenticate to an attacker-controlled host — the launchpad for relay and ADCS attacks.

WHY COERCION MATTERS

Authentication can be weaponized before it reaches its destination. When you coerce a machine — especially a DC — to authenticate outbound, you control what happens to that credential.

Combined with NTLM relay or ADCS ESC8, coercing a DC gives you its certificate → PKINIT → NT hash → DCSync. Full domain from a single coerce.

THE COERCION TOOLKIT

`PrinterBug (no patch)`
MS-RPRN RPC call forces outbound SMB auth. Enabled by default if Print Spooler runs on DC. MS won't fix. ESC8.

`PetitPotam (partially patched)`
MS-EFSRPC coercion outbound SMB. Patched in 2021 but variants persist. No creds required (pre-patch). ESC8.

`DFSCoerce (no patch) / ShadowCoerce (patch)`
MS-DFSNM and MS-FSRVP variants outbound SMB auth. Patchwork approach means newer coercion primitives keep emerging.

ATTACKER RELAY FLOW

```
1. Start relay listener  (ntlmrelayx -t target)
2. Coerce DC auth  (PetitPotam / printerbug)
3. DC auth hits attacker → forwarded to target
4. Target grants access as DC machine account
```

Targets: LDAP (RBCD), certsrv HTTP (ESC8), SMB (if signing off), ADCS enrollment

MITIGATIONS

Disable Print Spooler on all DCs
Enable SMB signing + LDAP signing/channel binding
ADCS ESC8: require HTTPS and EPA on certsrv
Monitor: Event 4769 etype 0x17 from DC$ accounts (DC authenticating outbound = active coerce); Event 5145 for \\PIPE\\ access on DCs from non-DC hosts
RPC filter EFS/DFSNM/FSRVP UUIDs on DCs (netsh rpc filter -- test first)

☢ ADCS — AD Certificate Services

Your PKI is directly tied to AD auth. Certs are persistent, silent, and survive password resets.

WHAT IS ADCS

ADCS is Microsoft's PKI implementation built into Windows Server. It issues X.509 certificates used for authentication, encryption, and code signing — all tied to Active Directory identities.

ADCS integrates with Kerberos via PKINIT, allowing a certificate to prove identity to the KDC and obtain a TGT — without ever touching a password.

WHY CERTS ARE DANGEROUS

⚠ Survive password resets
Password change ≠ cert revocation. A cert obtained today is valid for 1–2 years.

⚠ No MFA bypass needed
PKINIT auth via a stolen cert bypasses password-based MFA entirely.

⚠ Hard to detect
Certificate-based auth looks like normal Kerberos to most SIEM rules.

WHO HAS ADCS

Roughly 90%+ of enterprise AD environments have at least one ADCS deployment. It ships with Windows Server and is routinely enabled for smart card auth, VPN, or WiFi 802.1X.

Most orgs set it up once and never audit it again. Templates accumulate. Permissions drift. ESC conditions go unnoticed for years.

THE ATTACK SURFACE

Certificate Templates — misconfigured enroll rights or EKUs
CA Configuration — weak CA ACLs, dangerous flags (ESC6/7)
Enrollment Interfaces — HTTP certsrv (ESC8 relay), RPC
Delegation of enrollment — ESC3 enrollment agent abuse

Will Schroeder & Lee Christensen (SpecterOps) — Certified Pre-Owned (2021) catalogued 8 ESC classes.

ADCS — Certificate Services Vulns

ESC1 — SAN Injection
T1649

Cert template lets requester supply Subject Alt Name. Request cert as any user including DA.

ESC2 — Any Purpose EKU
T1649

Template has Any Purpose or no EKU restriction. Valid for any use incl. client auth.

ESC3 — Enroll Agent
T1649

Template allows Enrollment Agent. Requester can enroll on behalf of any user.

ESC4 — Template ACL
T1649

Low-priv user has WriteDACL/WriteOwner on template. Modify template to enable ESC1.

ESC6 — CA Flag Abuse
T1649

EDITF_ATTRIBUTESUBJECTALTNAME2 set on CA. Any template becomes ESC1.

ESC7 — CA ACL
T1649

ManageCA or ManageCertificates right grants ability to issue/deny or edit CA config.

ESC8 — NTLM Relay
T1649

NTLM relay to HTTP web enrollment. Receive cert for any coerced machine/DC account.

ESC13 — OID Group Link
T1649

Template linked to AD group via OID. Obtained cert grants group membership.

☢ ADCS ESC1 — SAN Injection

Any authenticated user can request a cert that impersonates any account, including Domain Admin.

WHAT & WHY

A cert template has CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT set. This lets the requester specify a Subject Alternative Name (SAN) e.g. administrator@domain.com, in the CSR.

The KDC trusts the certificate's SAN for PKINIT authentication, so the resulting cert authenticates as whichever identity the attacker named.

REQUIRES:
Enroll/AutoEnroll right on template | Client Auth EKU | Manager approval OFF

ATTACK CHAIN

1. Enumerate vulnerable templates

```
Locksmith ☺ or certipy find -u user@dom -p pass -vulnerable
```

2. Request cert as DA

```
certipy req -u user@dom -p pass -ca CA-01 -template VulnTpl -upn administrator@dom
```

3. Authenticate + extract NT hash

```
certipy auth -pfx administrator.pfx -dc-ip 10.x.x.x
```

4. Pass-the-Hash / DCSync

DETECT

Event 4886 (cert requested) + 4887 (cert issued)
SAN != requester UPN in cert logs
Certipy / Certify enum from low-priv account
BloodHound: ESC1 edge on template nodes

TOOLS

Locksmtih
Certipy (Python)
Certify.exe (C#)
PKINITtools
BloodHound + ADCS edges

MITIGATE

Remove
CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT
Require manager approval on sensitive templates
Audit ADC periodically

☢ ADCS ESC8 — NTLM Relay to HTTP

WHAT & WHY

ADCS web enrollment (certsrv) accepts NTLM auth over HTTP. Without Extended Protection for Authentication (EPA) or HTTPS channel binding, NTLM can be relayed.

A coerced machine account (e.g. DC01$) authenticates to the attacker, who replays that auth to certsrv and obtains a valid certificate for DC01$.

REQUIRES:
ADCS web enrollment reachable | EPA disabled | HTTP (not HTTPS) OR HTTPS without channel binding

ATTACK CHAIN

1. Coerce DC authentication

```
PetitPotam / PrintSpooler / DFSCoerce → attacker
```

2. Relay to certsrv HTTP

```
ntlmrelayx.py -t http://ca/certsrv/certfnsh.asp --adcs --template DomainController
```

3. Authenticate as DC + UnPAC the hash

```
certipy auth -pfx dc01.pfx -dc-ip 10.x.x.x
```

4. DCSync → full domain compromise

DETECT

Event 4886/4887 spikes for machine accounts
IIS logs: POST /certsrv/certfnsh.asp from unexpected IPs
Unusual PKINIT auth from machine accounts
DC coercion events (5145, RPC abnormal outbound)

TOOLS

impacket ntlmrelayx.py
PetitPotam / PrinterBug / DFSCoerce
Certipy (enumerate + auth)
PKINITtools (UnPAC the Hash)
Locksmith

MITIGATE

Enable EPA on certsrv (requires HTTPS_
Disable Print Spooler + restrict RPC coercion on DCs
Restrict or disable the certsrv web enrollment- most clients use RPC auto-enrollment, not the web interface

☢ ADCS From Foothold to Domain – ESC8

1	2	3	4	5	6
Foothold	**Recon**	**Coerce**	**Relay**	**Auth**	**Domain**
Low-priv domain user or compromised workstation	BloodHound + Certipy find to map vuln templates & CAs	PetitPotam / PrinterBug forces DC01$ to authenticate out	ntlmrelayx → certsrv HTTP → cert for DC01$ (ESC8)	certipy auth -pfx dc01.pfx → TGT + NT hash	secretsdump.p y (DCSync)→ all hashes → Golden Ticket → persistence

PREREQUISITES

Any domain user account
Print Spooler running on DC (default)
ADCS HTTP/S enrollment reachable
EPA not enforced on certsrv
→ All four are common in enterprise

TIME TO DOMAIN ADMIN

The full chain from foothold to DCSync can be accomplished in under 5 minutes in an unpatched environment.

No exploit. No 0-day. Pure misuse of default configurations and legitimate Windows features.

BREAK THE CHAIN AT

Step 3: Disable Print Spooler on DCs
Step 4: Enable EPA on certsrv / HTTPS only
Step 2: Run Certipy/Locksmith periodically (find vulns before attacker)

☢ Hardening ADCS — Defending Your PKI

ADCS is set up once and forgotten. Misconfigurations accumulate silently for years.

AUDIT NOW

Run Locksmith or Certipy or Certify from a low-priv account periodically:

```
certipy find -u user@dom -p pass -vulnerable
```

Review all templates: who can enroll, what EKUs are present, whether SAN supply is permitted, whether manager approval is required.

BloodHound ADCS edges (ESC1–ESC8) show attack paths visually.

TEMPLATE HYGIENE

Remove CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT from all templates
Scope Enroll/AutoEnroll to only the accounts that need it.
Enable manager approval on any sensitive template
Delete or disable unused templates — they accumulate over time
Remove Any Purpose EKU; restrict to specific intended EKUs
Audit certificate issuance logs (Event 4886/4887) regularly

CA HARDENING

Remove EDITF_ATTRIBUTESUBJECTALTNAME2 flag from CA (ESC6)
Restrict ManageCA and ManageCertificates rights (ESC7)
Enable EPA (requires HTTPS) on certsrv IIS enrollment (ESC8)
Restrict or disable certsrv web enrollment — most clients use RPC auto-enrollment, not the web interface

CA server = Tier 0. Treat it like a DC.

ONGOING MONITORING

Alert on Events 4886 / 4887 spikes — especially from machine accounts
Alert on PKINIT Kerberos auth (Event 4768 with cert)
IIS logs on CA: POST /certsrv/certfnsh.asp from unexpected IPs
Enroll a 'honey cert' on a sensitive template — alert on any issuance

Treat any cert for DA-level accounts as an incident


THEY TOLD ME I COULD
BE ANYTHING
I WANTED
SO I BECAME A
DOMAIN CONTROLLER

Defense & Detection

Architecture

✓ Tiered admin (Tier 0/1/2)

✓ PAWs for privileged access (risk based)

✓ Harden NTLM relay

Credentials

✓ LAPS for local admin

✓ Protected Users group and/or Sensitive and Cannot Be Delegated

✓ Credential Guard

Kerberos

✓ Disable RC4, use AES

✓ Rotate KRBTGT periodically

✓ Enable FAST armoring – encrypts Kerberos pre-auth, Kerberoasting and Kerberos encryption downgrade attacks become harder

Detection

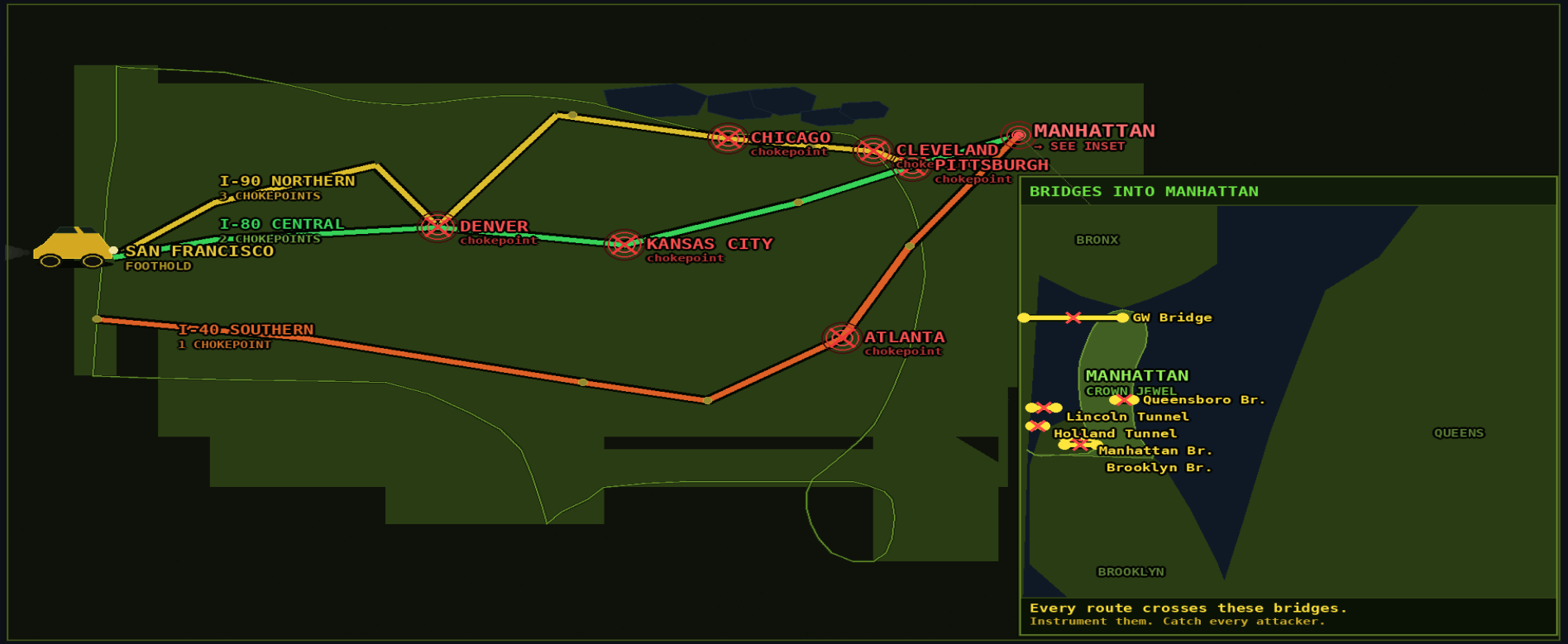
✓ Monitor replication (DCSync)

✓ Honey tokens/accounts

✓ Detect your vulnerabilities: Bloodhound, PingCastle, Locksmith are my favorites

Attack Paths -- Know Your Network

You cannot guard every road. Run BloodHound as a defender. Find your chokepoints -- then instrument and monitor those specifically.



BloodHound maps the graph. Find shortest paths to DA. Fix the chokepoints -- not everything. Monitor specifically. T1482 / T1087

The Tiered Administration Model

KEY RULE: Credentials from a tier 0 must NEVER be used on tier 1 or 2. Any account that can manage a different tier IS that tier, regardless of intent. Enforced via Group Policy (start here) + Kerberos Authentication Policies (see MS tiering guidance).

0

CROWN JEWELS

SYSTEMS

Domain Controllers
ADCS / PKI Infrastructure
AD Connect / Entra Sync
Backup Systems
Identity Governance Tools (e.g., SailPoint)
Hypervisors (vCenter, Hyper-V)
Security tooling (AV, EDR, SIEM)

ACCOUNTS

Domain Admins
Enterprise Admins
Schema Admins
ADCS Admins

Belisle_DA

1

SERVER INFRASTRUCTURE

SYSTEMS

Application Servers
File / Print Servers
Database Servers
Web Servers
Management Servers

ACCOUNTS

Server Admins
Application Admins
DBA Accounts
Service Accounts (scoped)

Belisle_Srvr

2

END USER ENVIRONMENT

SYSTEMS

Workstations / Laptops
Help Desk Systems
Standard User Devices

ACCOUNTS

Help Desk Admins
Workstation Admins
Standard User Accounts

Belisle_Wkstn

☢ Strategic Mitigations

Defense in depth — not a single control, but layered friction that raises attacker cost at every step. But don't lose site of signing and channel binding…

Privileged Access Workstations

Dedicated hardened devices used exclusively for Tier 0/1 admin tasks. Blocks credential theft via browser, email, or user-space malware on admin systems.

LAPS

Local Administrator Password Solution randomizes local admin passwords per-machine. Eliminates lateral movement via shared local admin creds (the 'pass around the hash' problem).

Protected Users Group

Members cannot use NTLM, DES, or RC4 Kerberos. Cannot cache credentials on hosts. Cannot be delegated. High-value accounts should always be in here.

Credential Guard

Virtualization-based security isolates LSASS in a protected process. Prevents NTLM hash extraction from memory and Kerberos ticket theft via Mimikatz-style attacks. Do a check.

Tiered Administration Model

Means nothing without enforcement. Start with GPO logon restrictions to prevent Tier 0 credentials from touching Tier 1/2 systems. Combine with Use Authentication Policy Silos (advanced)

BloodHound / Attack Path Mgmt

Run BloodHound regularly. Find your shortest path to DA. Attack path management programs turn adversarial graph analysis into a defensive capability.

Questions?

Non-Toxic Tools

Attack path view: https://github.com/SpecterOps/BloodHound
AD assessment via exe: https://www.pingcastle.com/
ADCS-defensive perspective https://github.com/jakehildreth/Locksmith
ADCS-attacker perspective: https://github.com/ly4k/Certipy

Non-Toxic References

- https://syfuhs.net/killing-ntlm-is-hard
- https://www.microsoft.com/en-us/security/blog/2024/10/11/microsofts-guidance-to-help-mitigate-kerberoasting/
- https://posts.specterops.io/certified-pre-owned-d95910965cd2
- https://www.microsoft.com/en-us/security/blog/2022/05/25/detecting-and-preventing-privilege-escalation-attacks-leveraging-kerberos-relaying-krbrelayup/
- https://en.hackndo.com/constrained-unconstrained-delegation/
- https://learn.microsoft.com/en-us/security/privileged-access-workstations/privileged-access-access-model
- https://techcommunity.microsoft.com/blog/coreinfrastructureandsecurityblog/protecting-tier-0-the-modern-way/4052851
- https://specterops.io/blog/2025/04/08/the-renaissance-of-ntlm-relay-attacks-everything-you-need-to-know/
- https://media.defense.gov/2024/Sep/25/2003553985/-1/-1/0/CTR-Detecting-and-Mitigating-AD-Compromises.PDF
- https://www.crowe.com/cybersecurity-watch/exploiting-ad-cs-a-quick-look-at-esc1-esc8
- https://blog.quest.com/implementing-a-tiered-administration-model-in-active-directory/
- https://learn.microsoft.com/en-us/windows-server/security/credentials-protection-and-management/authentication-policies-and-authentication-policy-silos