SPECTEROPS

Putting NTLM in the Doghouse

The journey of modeling NTLM relay and authentication coercion attacks in BloodHound

SPECTEROPS SO-CON 2025

Lee Chagolla-Christensen & Rohan Vazarkar

SpecterOps

Introductions

RESEARCH

ENGINEERING

NEW STUFF

BACKGROUND

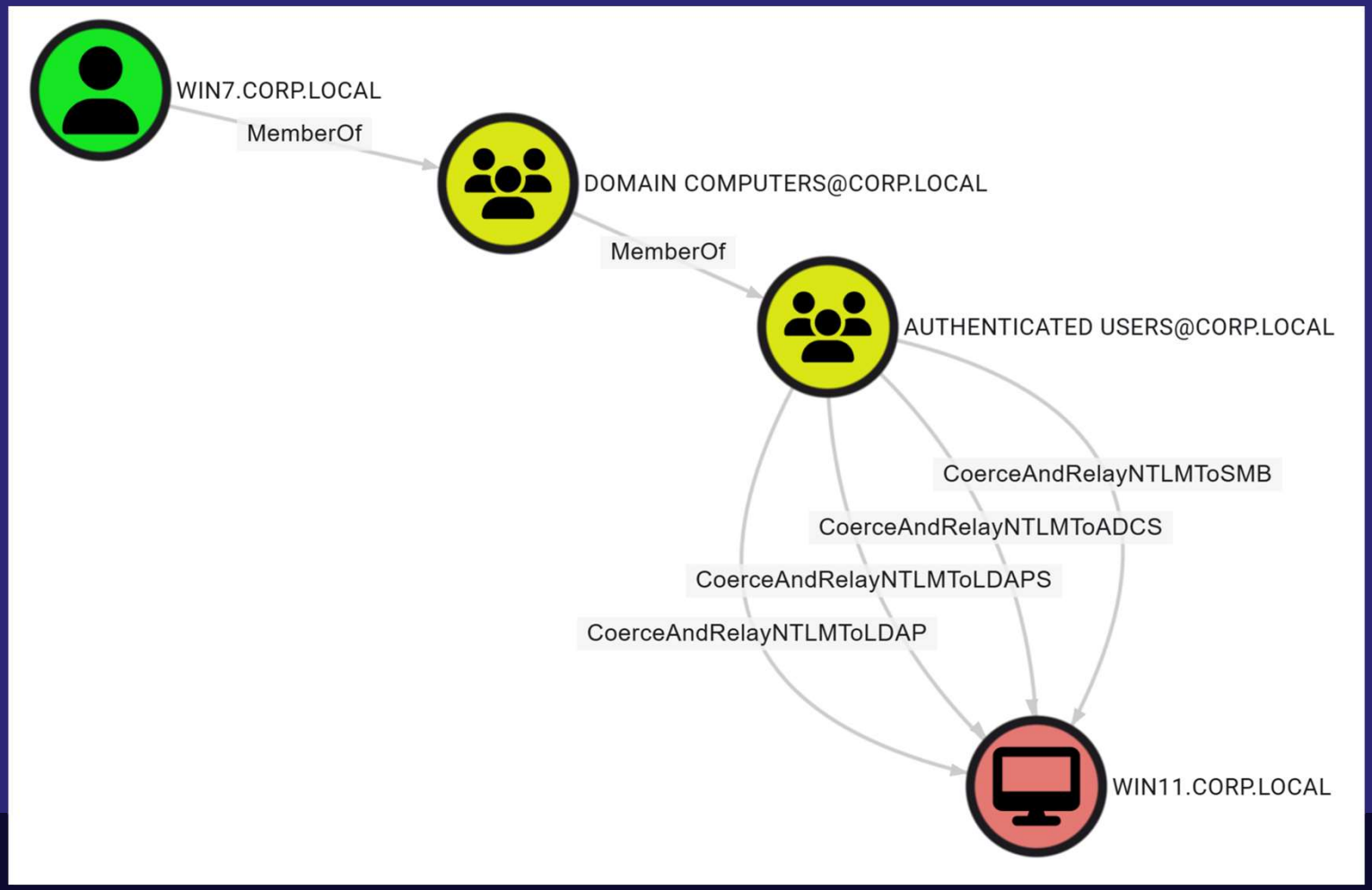
LESSONS LEARNED

TLDR: 4 New BloodHound Edges

A powerful combo: Authentication Coercion + NTLM Relay

- CoerceAndRelayNTLMToSMB
- CoerceAndRelayNTLMToADCS
- CoerceAndRelayNTLMToLDAP
- CoerceAndRelayNTLMToLDAPS

A Powerful Combo: Authentication Coercion + NTLM Relay



Collection Options

WebClientService, LdapServices, SmbInfo, NTLMRegistry

```
PS C:\> .\SharpHound.exe --help
2025-03-30T13:00:18.9526342-07:00|INFORMATION|This version of SharpHound is compatible with the 5.0.
0 Release of BloodHound
SharpHound 2.6.1+340aaa6c3f765960645caf012eee7a35550129ce
Copyright (C) 2025 SpecterOps

  -c, --collectionmethods      (Default: Default) Collection Methods: Group, LocalGroup, LocalAdmin,
                               RDP, DCOM, PSRemote, Session, Trusts, ACL, Container, ComputerOnly,
                               GPOLocalGroup, LoggedOn, ObjectProps, SPNTargets, UserRights,
                               Default, DCOnly, CARegistry, DCRegistry, CertServices,
                               WebClientService, LdapServices, SmbInfo, NTLMRegistry, All
```

All collectable as a low-priv user!*

Background

The basics of Authentication Coercion and NTLM Relay

Why did we model NTLM relay?

Prioritization:

- Intuition
- RICE: Priority = (Reach * Impact * Confidence) / Effort
- Assessment work shows prevalence of relay attack paths
- Very ripe set of impactful targets (AD CS, SCCM, LDAP, MSSQL)
- Current assessment/operator workflows weren’t ideal

Isn’t Microsoft getting rid of NTLM?

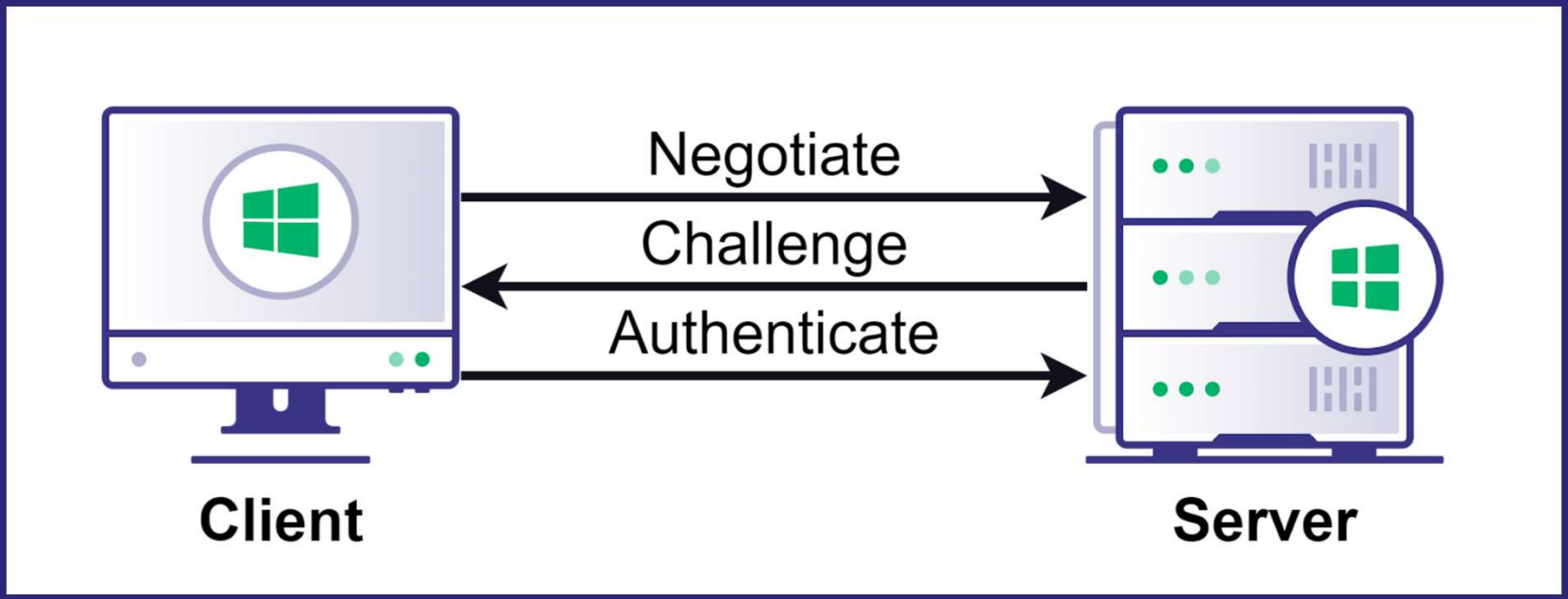
NTLM: “I’m not dead yet!”

- Legacy systems
- IAKERB is not enabled…yet
- Still enabled by default
  - Deprecated so far, not disabled / removed

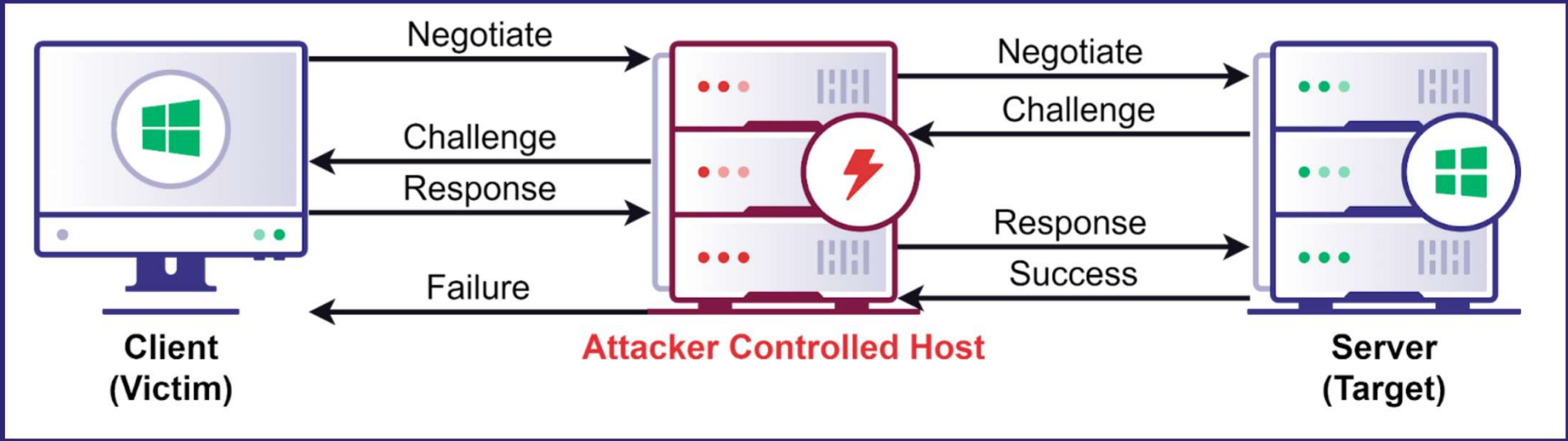
NTLM	NTLMv1 is removed. LANMAN and NTLMv2 are no longer under active feature development and are deprecated. NTLMv2 will continue to work but will be removed from Windows Server in a future release. Replace calls to NTLM to calls to Negotiate, which try to authenticate with Kerberos and only fall back to NTLM when necessary. For more information, see The evolution of Windows authentication.

NTLM Authentication Messages

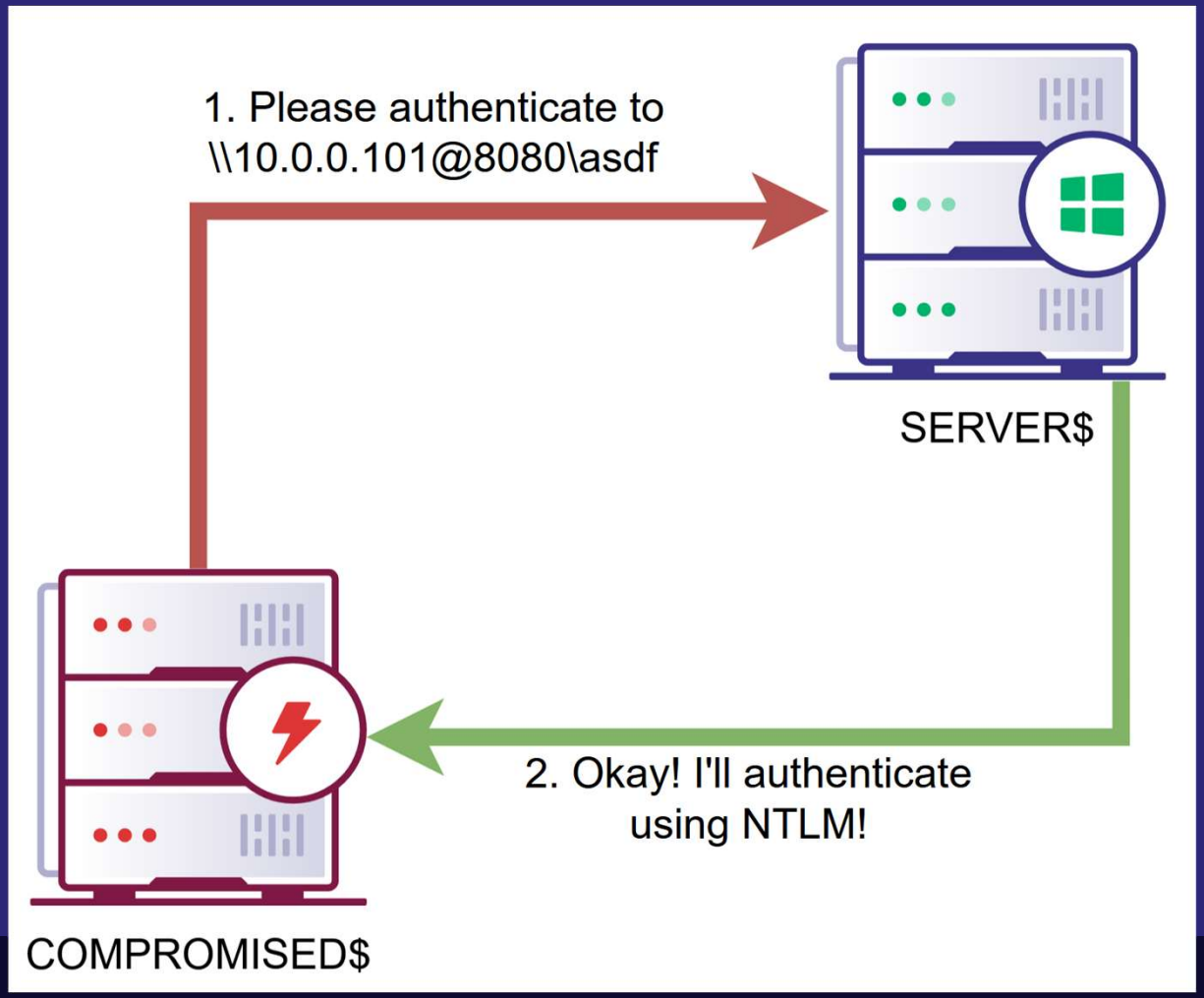
The basic, big idea



Edge Component 1: NTLM Relay



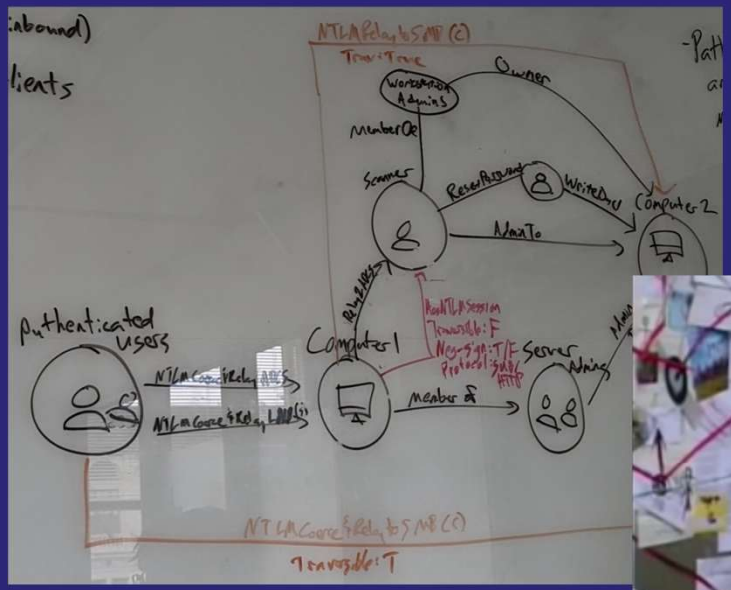
Edge Component 2: Coerced Computer Authentication

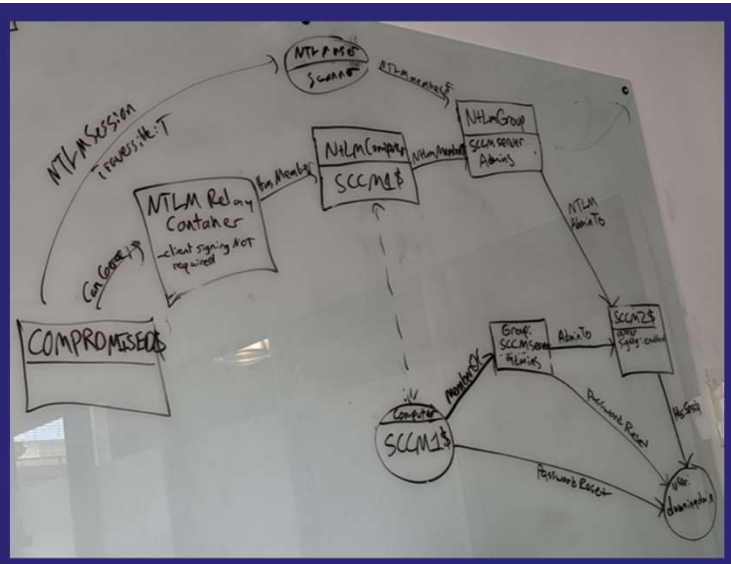


Examples:
- SpoolSample
- PetitPotam
- Coercer

Early Relay Modeling

Lee, Will, and Elad

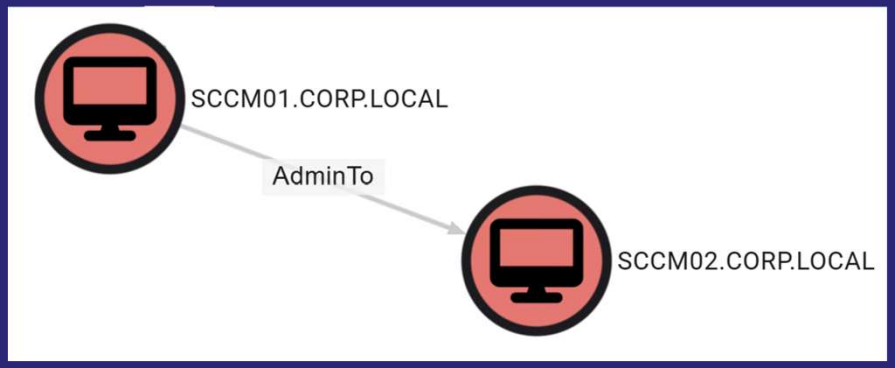


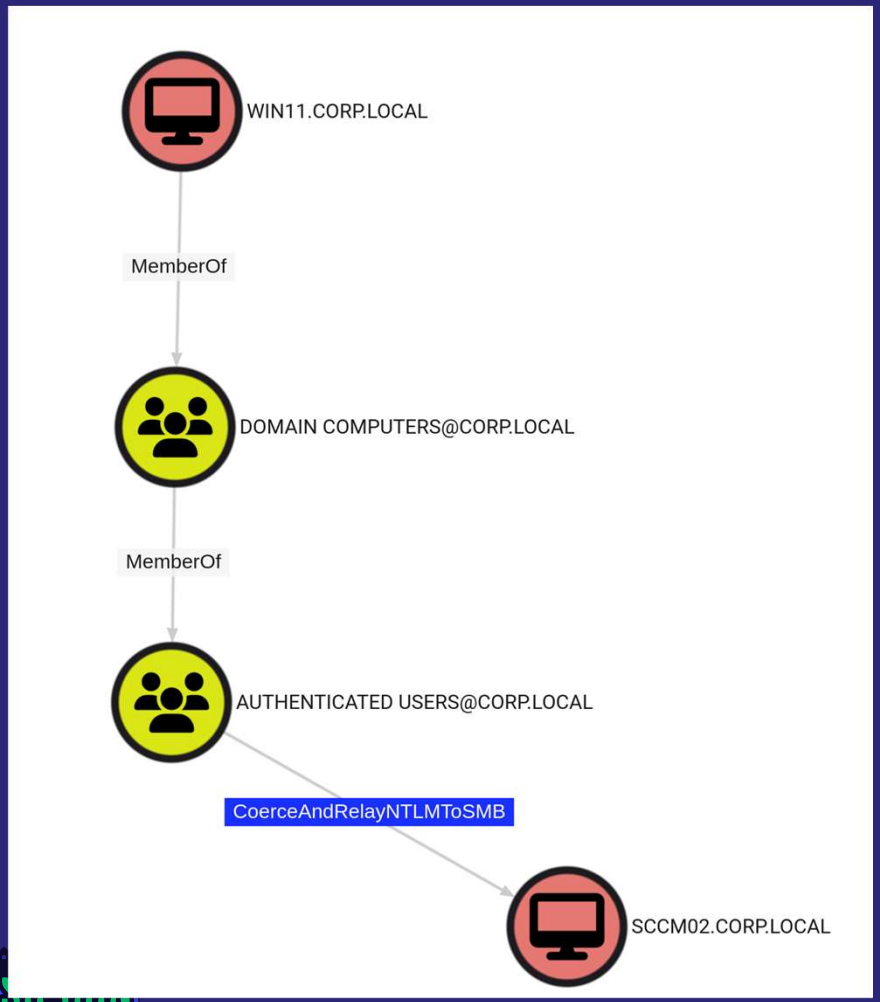


CoerceAndRelayNTLMToSMB

Setup:

- Attacker compromises WIN11
- SCCM01 has local admin to SCCM02





CoerceAndRelayNTLMToSMB

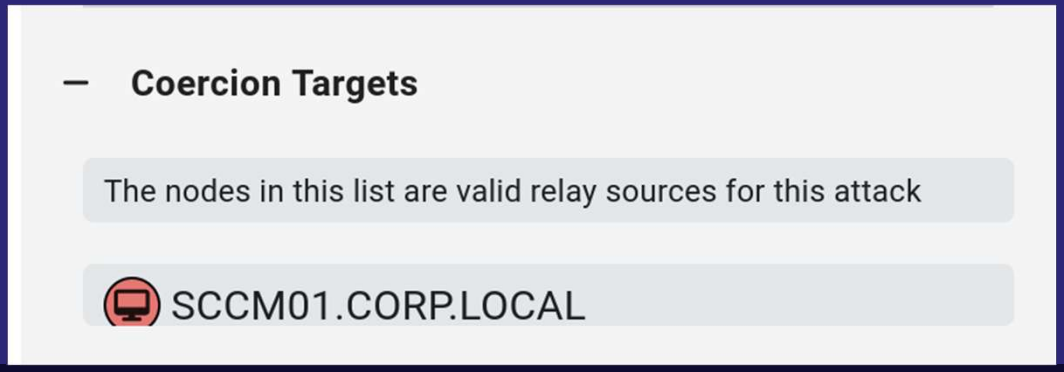
Coercion (Source) Target Requirement

- Outbound NTLM Allowed

Key: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0\
Value: RestrictSendingNTLMTraffic

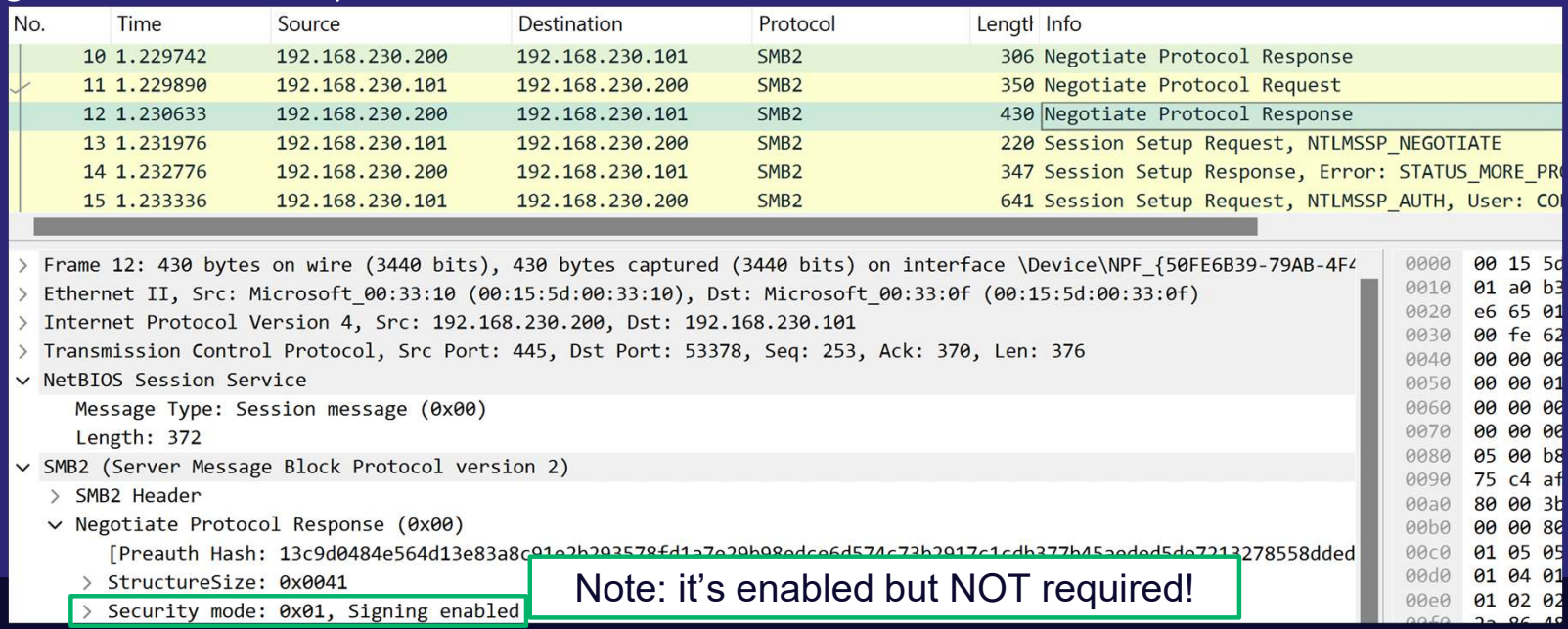
Relay Target Requirements:

- SMB signing not required



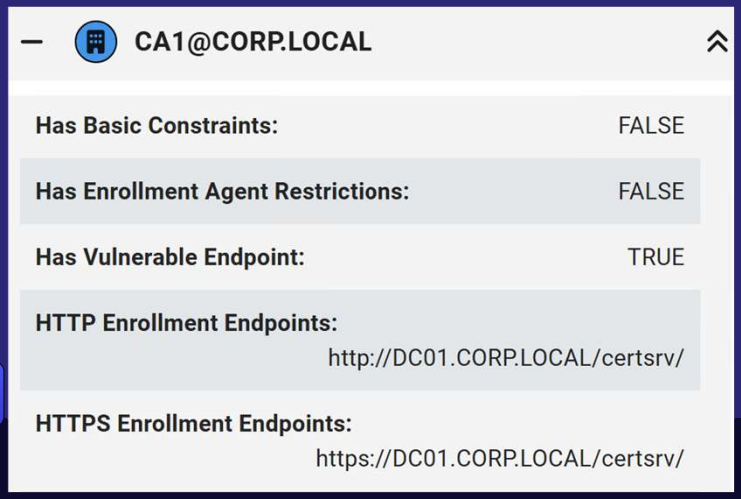
Mitigation: SMB Signing

- Protection negotiated between client/server in the SMB1/2 negotiate messages' "Security Mode" field
  - Note: this is different than signing/sealing (session security) negotiated in the NTLM protocol
  - Controlled by HKLM\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters
    - Value: EnableSecuritySignature / RequireSecuritySignature
- Can be enumerated unauthenticated (nmap, nxc,Invoke-SMBEnum, Responder's RunFinger.py, FeigongSec/NTLMINFO)



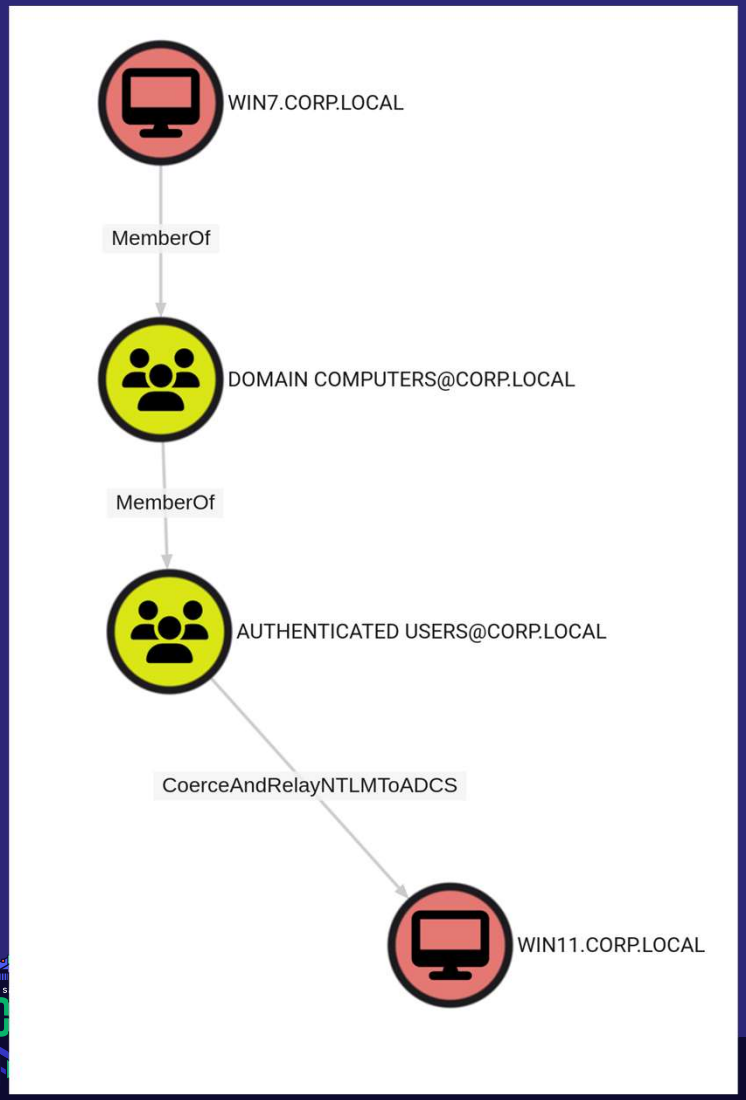
Note: it's enabled but NOT required!

CoerceAndRelayNTLMToADCS (ADCS ESC8)



Setup:

- Attacker compromises WIN7
- ADCS installed with web enrollment endpoints (new property on CA)
- Target machine can enroll in an applicable certificate template



CoerceAndRelayNTLMToADCS (ADCS ESC8)

Coercion (Source) Target Requirement

- Outbound NTLM Allowed

Relay Target (AD CS) Requirements:

- An HTTP enrollment endpoint
- HTTP enabled or HTTPS w/o Extended Protection for Authentication (EPA, a.k.a. channel binding)

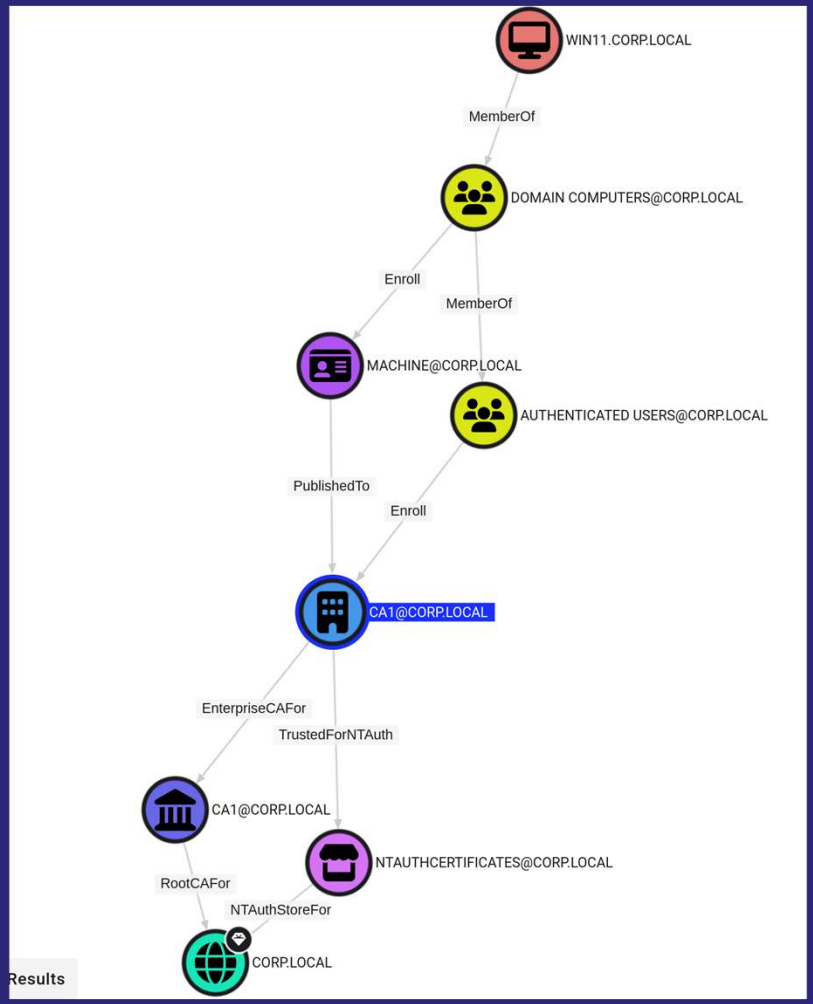
Edge Composition

There's a lot going on there!

— Composition

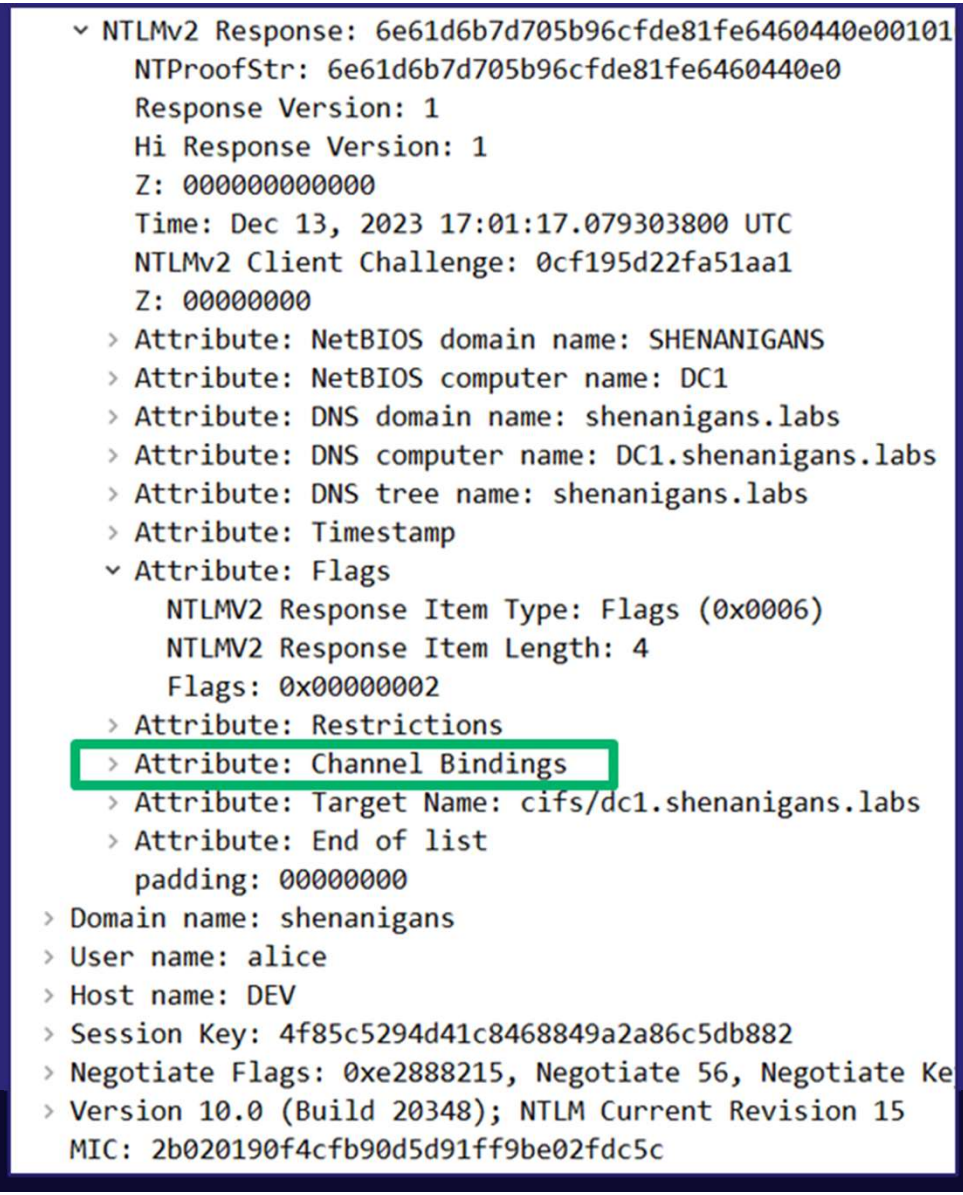
The relationship represents the effective outcome of the configuration and relationships between several different objects. All objects involved in the creation of this relationship are listed here:

- MACHINE@CORP.LOCAL
- DOMAIN COMPUTERS@CORP.LOCAL
- SCOMSERVER.CORP.LOCAL
- CORP.LOCAL
- AUTHENTICATED USERS@CORP.LOCAL
- CA1@CORP.LOCAL
- NTAUTHCERTIFICATES@CORP.LOCAL
- CA1@CORP.LOCAL

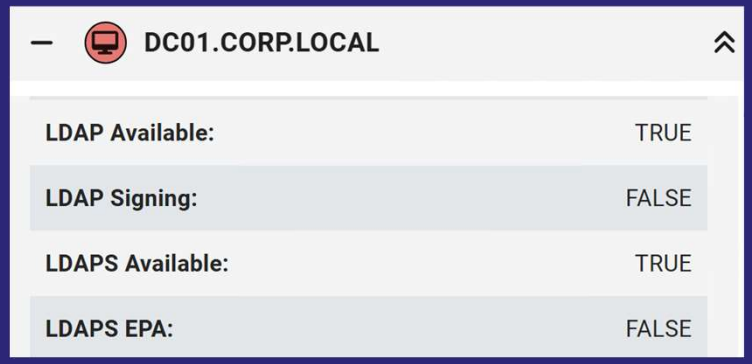


Mitigation: Removal or Extended Protection for Authentication(EPA) / Channel Binding?

- Binds a token from the outer secure protocol (TLS) into an NTLM Authenticate message
- SharpHound currently requires a low privileged user to enumerate it

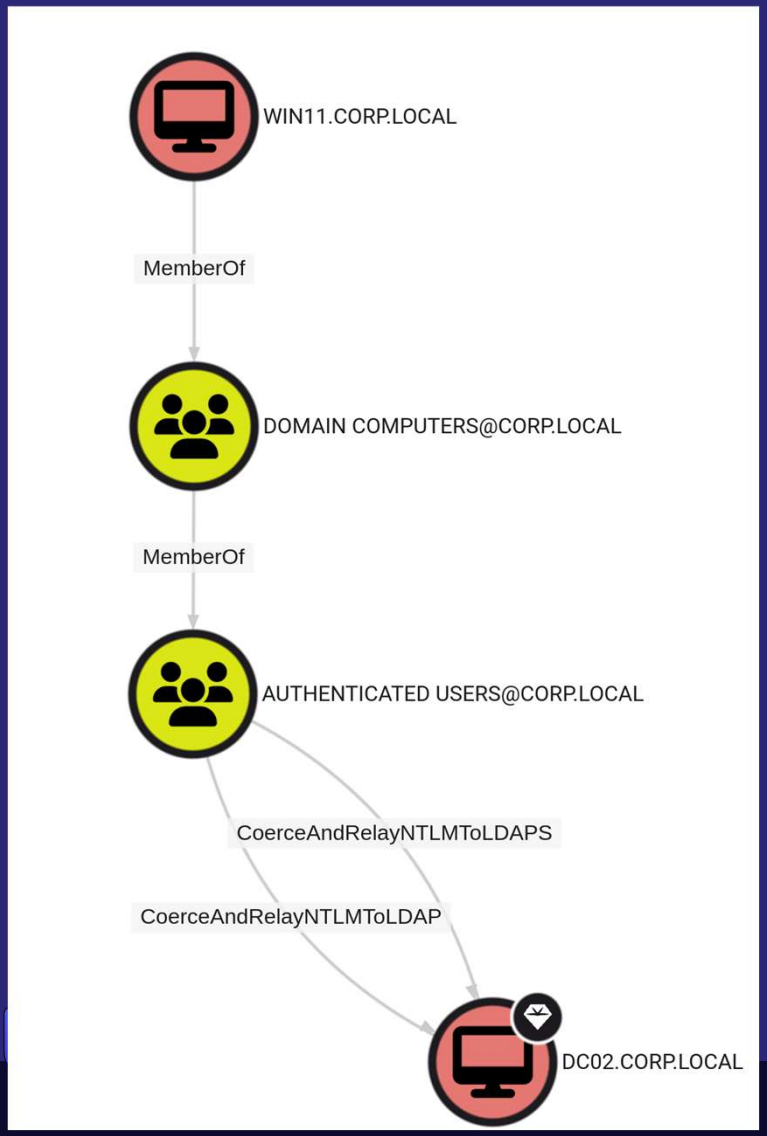


CoerceAndRelayNTLMToLDAP(S)



Setup:

- Attacker compromises WIN11
- Domain Controller
- Target computer (DC02) has the WebClient service running / installed



CoerceAndRelayNTLMToLDAP(S)

<u>Coercion (Source) Target Requirement</u>

- Outbound NTLM Allowed
- WebClient service running

<u>Relay Target (LDAP) Requirements:</u>

- LDAP: No signing

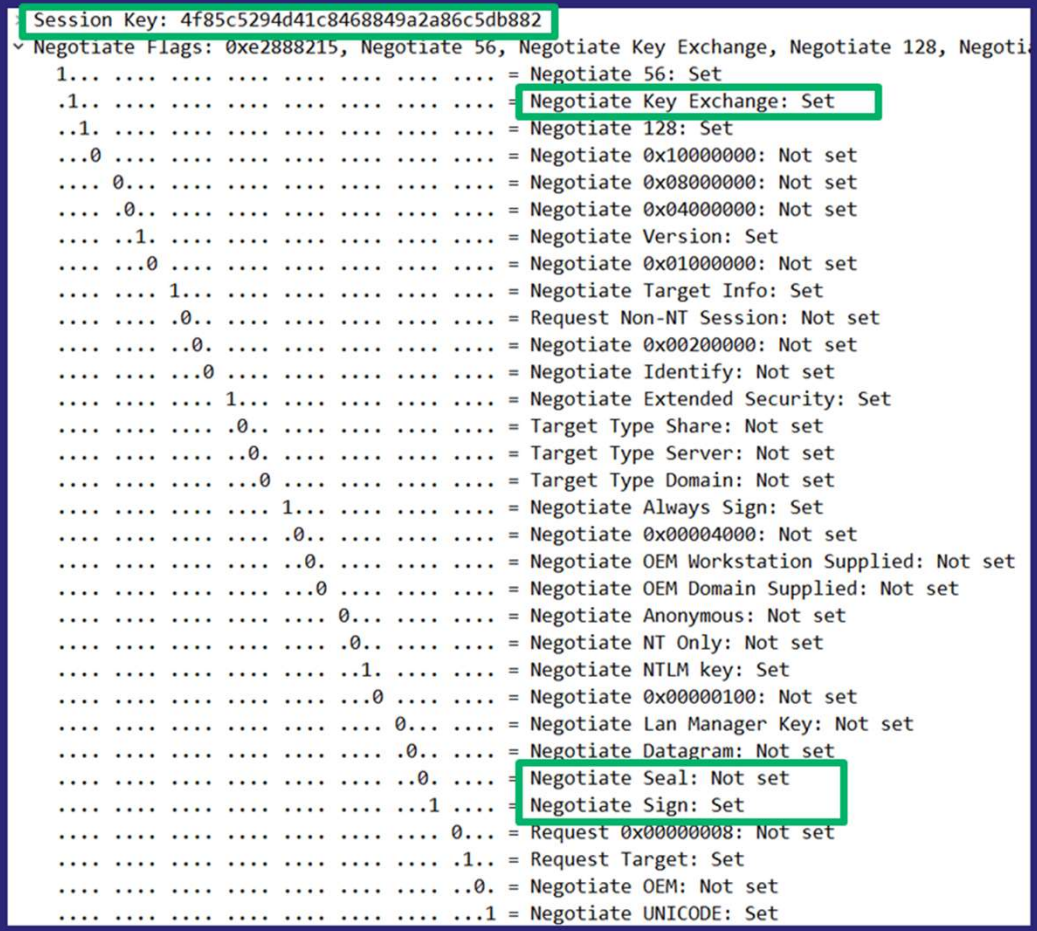
- LDAPS:
  - Extended Protection for Authentication
  - OR LDAP signing disabled

https://offsec.almond.consulting/bypassing-ldap-channel-binding-with-starttls.html

Mitigation: LDAP Signing + LDAPS Channel Binding

Enforce both!

- Signing in this case refers to the signing bit in the NTLM messages

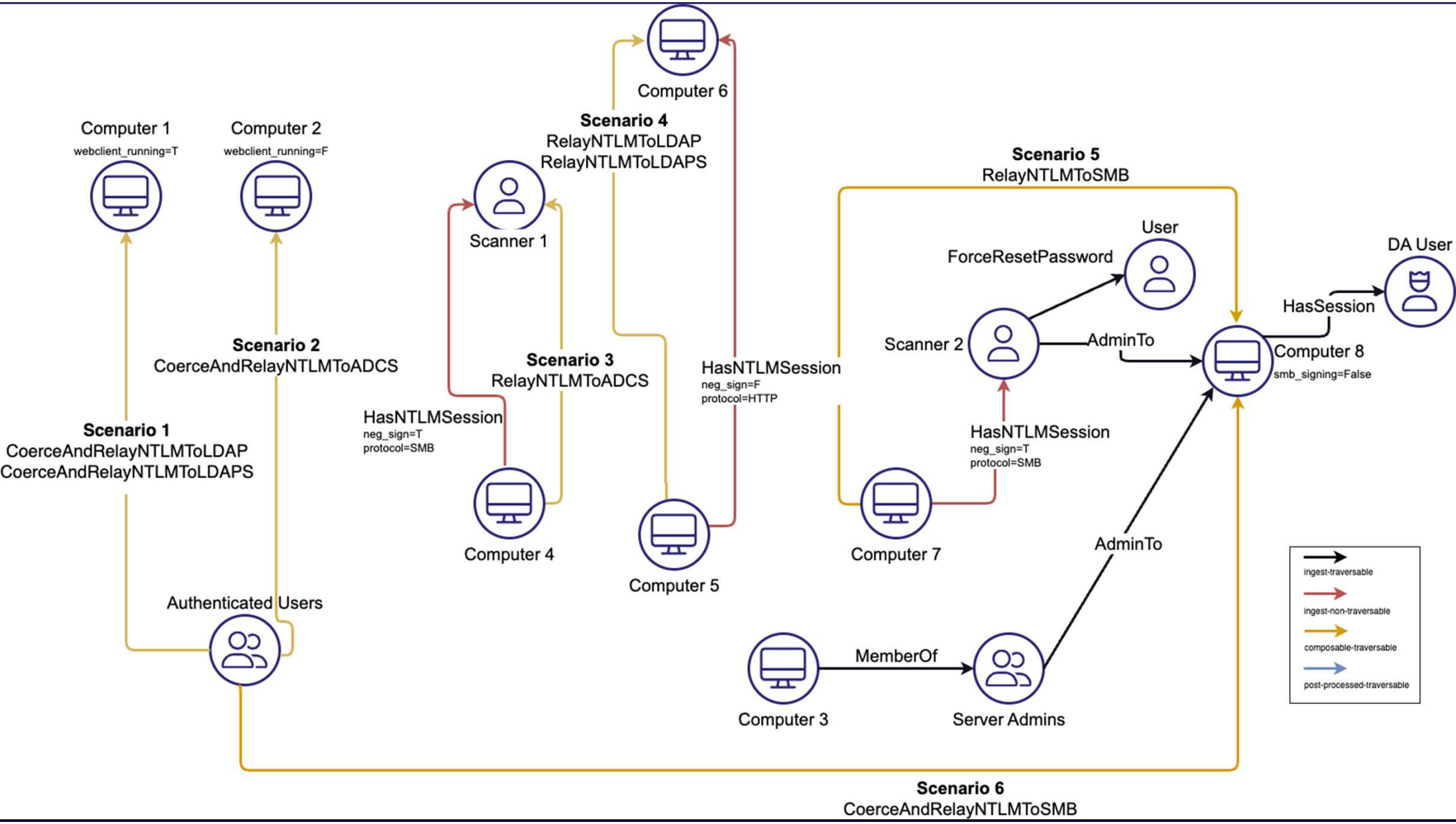


Engineering

The boring part of the talk

Engineering Handoff Process

- Research provided documentation on all the new edges in a private github repo.
- Each attack primitive was provided with all the information needed for each edge's info panels
- Rough post processing guidelines were created to show the ideal workflow and test harnesses were created using arrows.app to allow us to model expected behavior for positive cases


Computer 1
webclient_running=T
Computer 2
webclient_running=F
Computer 6
Scenario 4
RelayNTLMToLDAP
RelayNTLMToLDAPS
Scenario 5
RelayNTLMToSMB
Scanner 1
User
ForceResetPassword
DA User
HasSession
Scenario 2
CoerceAndRelayNTLMToADCS
Scenario 3
RelayNTLMToADCS
HasNTLMSession
neg_sign=F
protocol=HTTP
Scanner 2
AdminTo
Computer 8
smb_signing=False
HasNTLMSession
neg_sign=T
protocol=SMB
Scenario 1
CoerceAndRelayNTLMToLDAP
CoerceAndRelayNTLMToLDAPS
HasNTLMSession
neg_sign=T
protocol=SMB
Computer 4
Computer 5
Computer 7
AdminTo
Authenticated Users
MemberOf
Computer 3
Server Admins
ingest-traversable
ingest-non-traversable
composable-traversable
post-processed-traversable
Scenario 6
CoerceAndRelayNTLMToSMB

Engineering Process

- Always have a Jonas on your team
  - Did rough translation between expectations and actual code.
  - Ended up doing 50% of the translation work for us.
- Decomposed work into JIRA tickets and linked them all into a big initiative.
- Did estimations on time taken
  - (which were as per usual wildly off)
- Assigned an engineering team to go to work (including yours truly)

Engineering Process

- Deployed our first release with NTLM information in early access a few weeks ago.
- Saw immediate jumps in exposure across many environments (more work!)
- Identified bugs from our early adopters and worked on triaging them rapidly. Final NTLM release will likely be next week barring any other bugs being found!

Triumphs!

Making a note here, huge success

Greatly improved handoff process

- Our previous research to engineering handoff was ADCS, which ended up being a big of a disaster internally (for many, many reasons).
- ADCS Post Processing was a "from-scratch" which blew out our timeline significantly. NTLM got to leverage a lot of that hard work from before
- Learned tons of stuff from that process and implemented lessons learned.
- This process was significantly smoother than our previous effort, which shows great progress

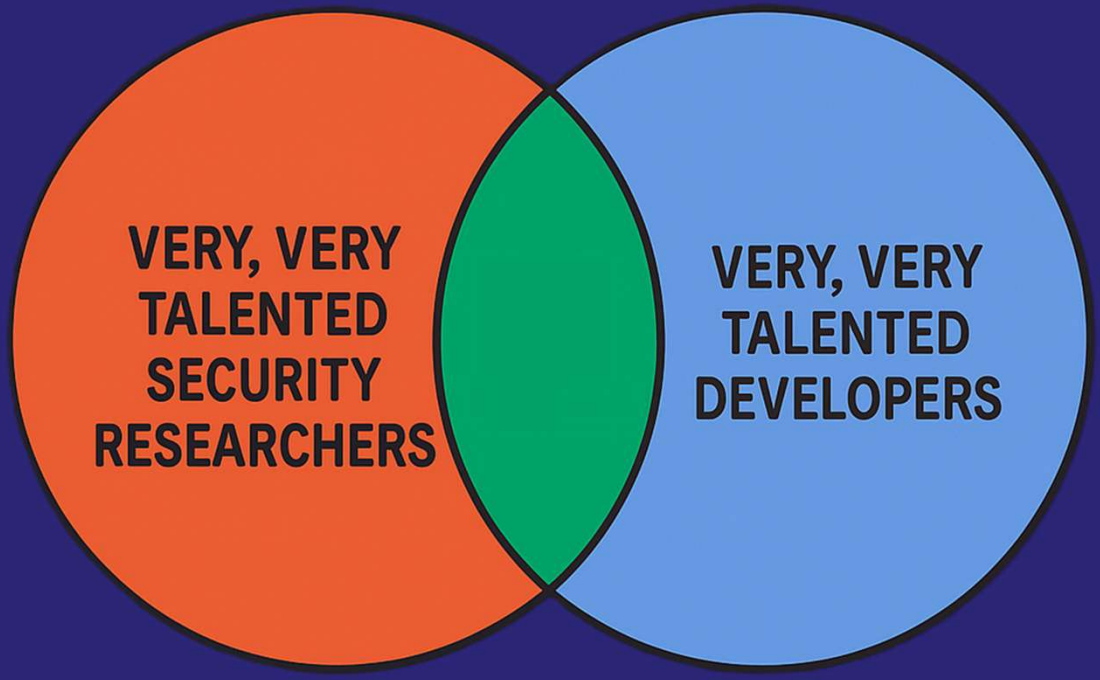
Proving out complex non-standard primitives

- NTLM coercion is different from any of our other attack primitives in that its directionality is different
- Any user can perform NTLM relay, so how do we model this appropriately?
- Research team came up with a clever idea to tie this to authenticated users, allowing us to model within the confines of our graph and reflect the severity of these issues appropriately

Lessons Learned

Oh man...so many

Engineers are not researchers (and vice versa)



- Understanding AD primitives is hard as a domain expert.
  - It's impossible as an engineer with no background

Engineers are not researchers and vice versa

- Many of the post processing primitives were sufficiently complicated that without extremely specific instructions (handholding) it was difficult for engineers to produce a working example
- LDAP vs LDAPS – Engineers think this is plural LDAP. We know differently
- Nuances around how AD behaves complicates the problem (Protected Users, Restrict Outbound NTLM)

The more time that passes, the harder it gets

Research and development are not carried out in tandem

- Result: when engineering was happening, the researchers involved were no longer fresh on the research and were working on other things
- Created a delay for interfacing with research due to assessment priorities

Lessons Learned

- Minimize time of research to time of engineering,
- Have a researcher available for testing near end of engineering

Communication about specifics

- During review, engineering determined based on naming that a file was part of a third-party dependency, and so vendored it (referenced a specific commit, changed the file to the original version with minimal modifications)
  - Accidentally removed a very subtle but important modification.
- Resulted in Lee spending 3+ days tracking down something that was extremely basic

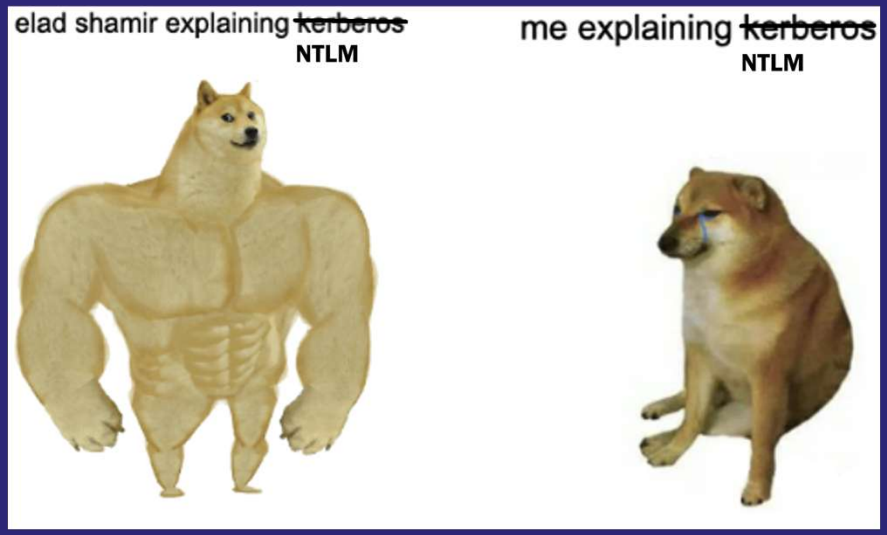
Lesson Learned: Communicate changes

Test Data

- Good and comprehensive test data
  - Covers both positive AND negative test cases for each primitive.
  - Test data needs to be representative of a real environment,
    - Not a simple harness that illustrates the issue
- Test data needs to cover edge cases properly, so we accurately simulate real environments
- End to end testing is important! Harnesses are prone to human error during generation or fixing things that are not reflective of what the collector is actually sending over
- Thank you early access users!

The Future and Final Thoughts

- Expand beyond coerced computer authentication
  - E.g. network scanner accounts?
- Data collection for other NTLM attacks/defenses.
- IAKERB kills NTLM?
- Keep an eye out for Elad Shamir's upcoming post!



SPECTEROPS

Thank you!

Any Questions?

SPECTEROPS
SO·CON
2025

Lee Chagolla-Christensen | lee@specterops.io

Rohan Vazarkar | rvazarkar@specterops.io