# [Attack]tive Directory

*Exploiting Active Directory for Offensive Purposes*

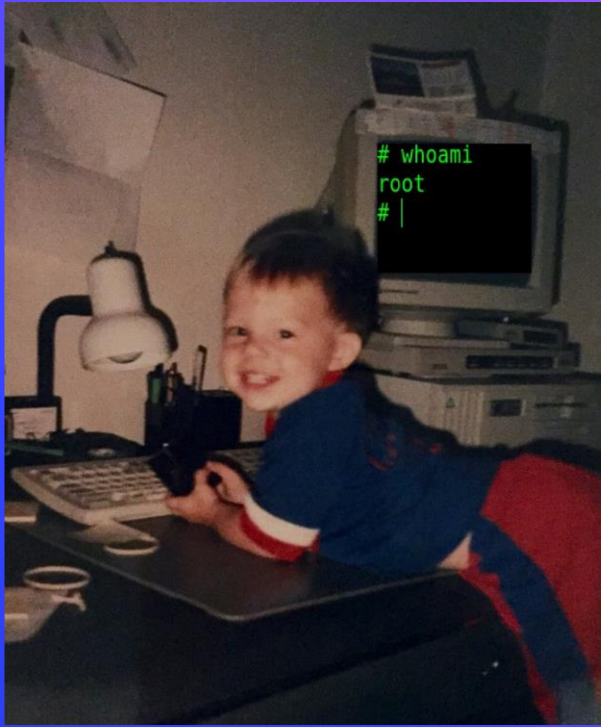
Presented by Ryan Hausknecht

“*We’re ready for a pentest, our vulnerability scan shows NO criticals!*”

# How their network got owned in fifteen minutes via Active Directory

# $whodat

- *Ryan Hausknecht* (House-neckt)
- Security Consultant @ SpecterOps
- Instructor at UNC Charlotte for Cyber Security
- Instructor & Organizer at FBI/Infragard Cyber Camp
- Blackhat 2019 Instructor – Red Team Operations
- GPEN, GWAPT, OSCP
- @haus3c

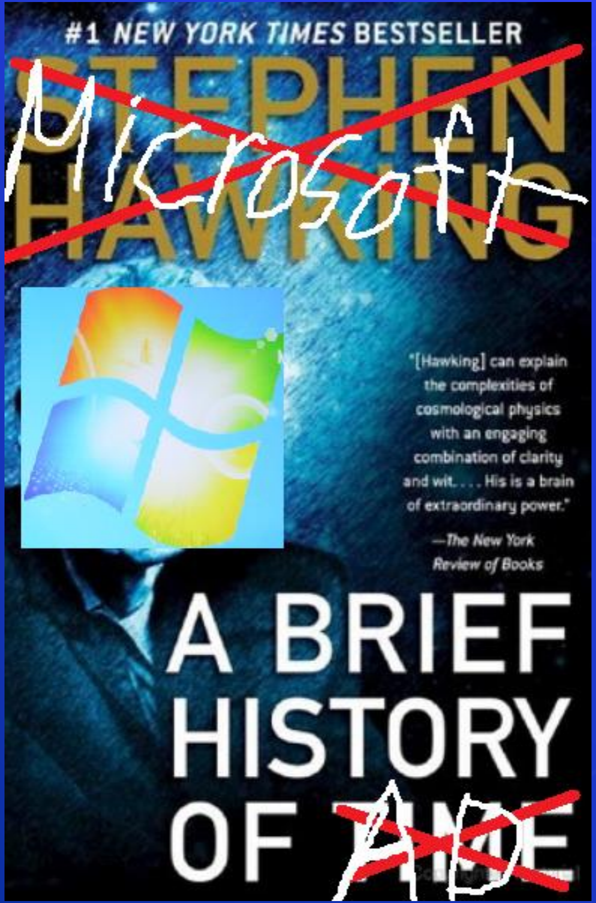


```
# whoami
root
# |
```

# What's this whole thing about?

- Massive discrepancy in maturity between enterprises and SMBs
- Red team exists to help blue team
- Many attacks over the years, these are the most common I've seen
- More-so in SMBs vs. enterprises, but still applicable
- A clean vulnerability scan does not mean a clean environment
- High quality ~~photoshop~~ Microsoft Paint edits.

# A Brief History of Active Directory (AD)


#1 NEW YORK TIMES BESTSELLER
~~STEPHEN HAWKING~~
Microsoft
"[Hawking] can explain the complexities of cosmological physics with an engaging combination of clarity and wit. . . . His is a brain of extraordinary power."
—The New York Review of Books
A BRIEF HISTORY OF ~~TIME~~
AD

- Directory services for Windows
- Introduced in Server 2000
- Used to control objects on the domain
  - Users, computers, policies, etc.
- Can group objects into Organizational Units (OU)
- Utilizes Group Policies to apply settings

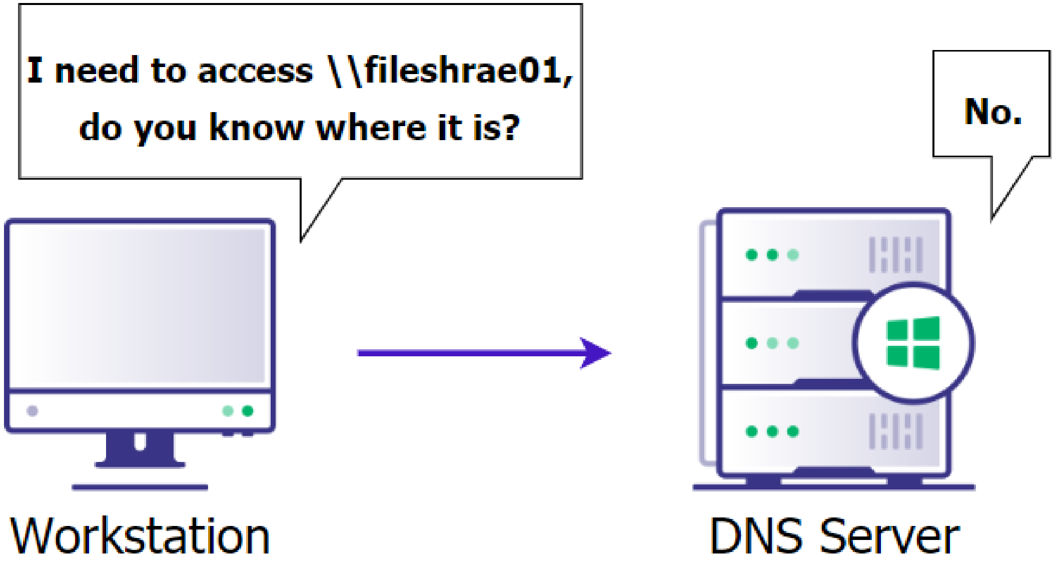
# Default Group Policy

- Security =/= Convenience
- Default GP is not meant to be secure!
- It's 💩

# 1. Gathering Credentials

# LLMNR & NBT-NS Spoofing

- Local Link Multicast Name Resolution
- NetBIOS Naming Service (Protocol in an API)
- “Backups” to DNS
  - E.g. \\fileshrae01\
- Natively insecure
  - Trusts any response!
- Commonly found on networks with decom’d file shares
- Enabled by DEFAULT

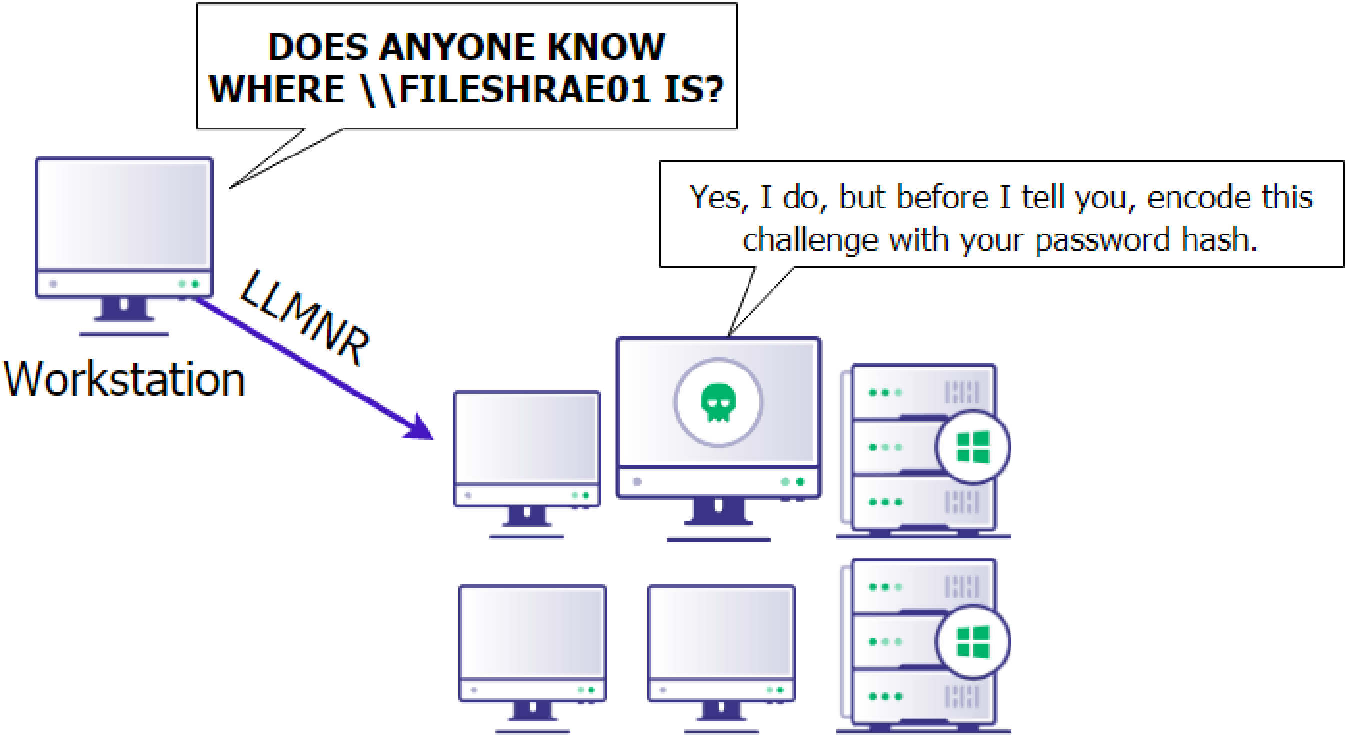


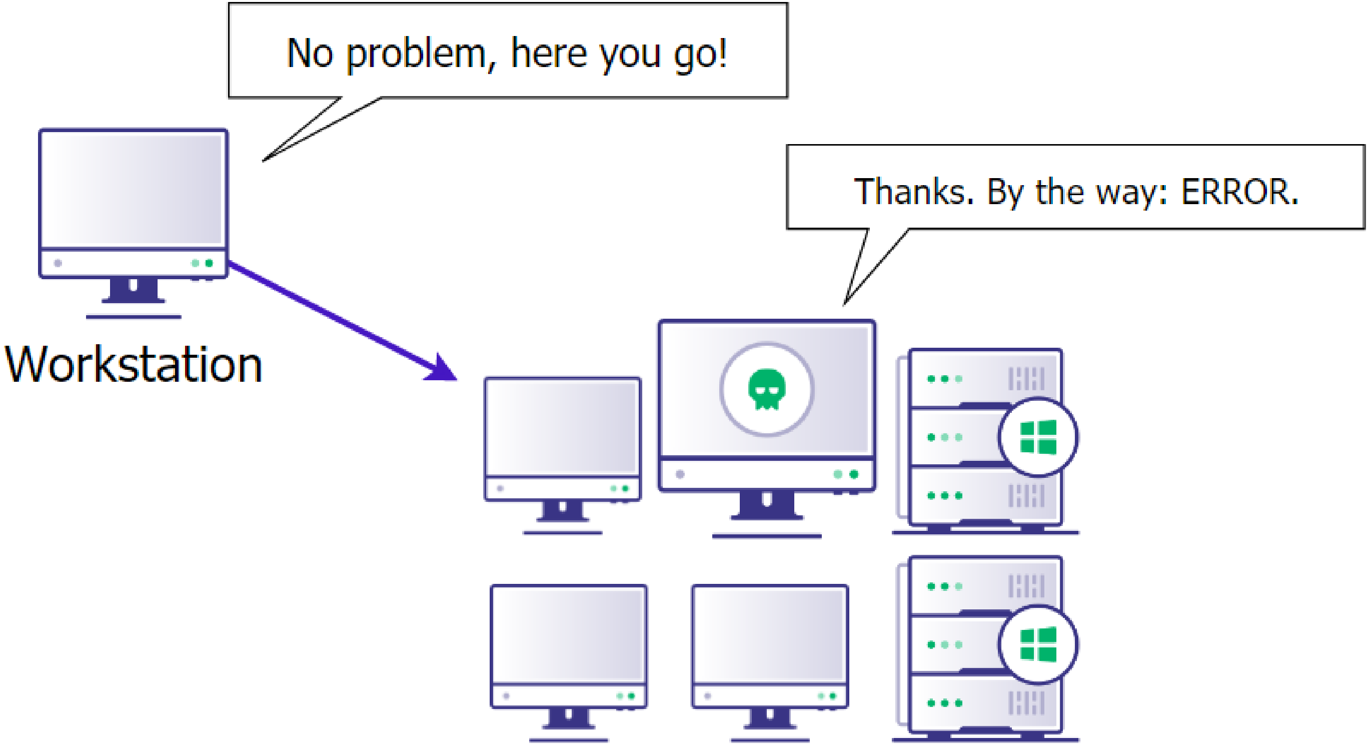
I need to access \\fileshrae01, do you know where it is?

No.

Workstation

DNS Server


DOES ANYONE KNOW WHERE \\FILESHRAE01 IS?
Yes, I do, but before I tell you, encode this challenge with your password hash.
LLMNR
Workstation



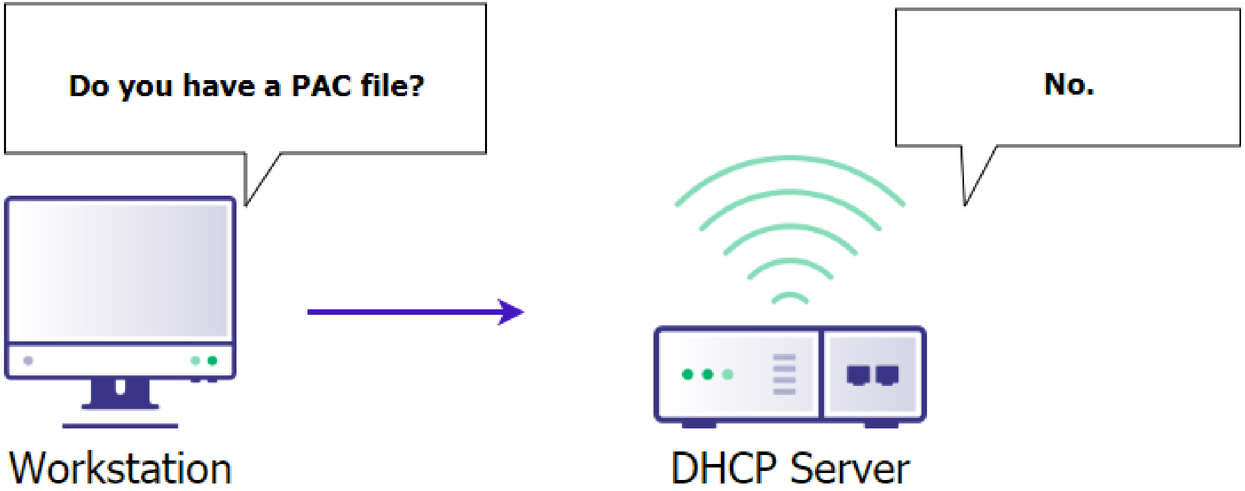
No problem, here you go!

Thanks. By the way: ERROR.

Workstation

# WPAD Spoofing

- Web Proxy Auto Discovery (WPAD) Protocol
- Outlines how to search for a Proxy Auto Connection (PAC) file any time internet is used.
- First searches via DHCP, then DNS, then LLMNR
- Enabled by DEFAULT

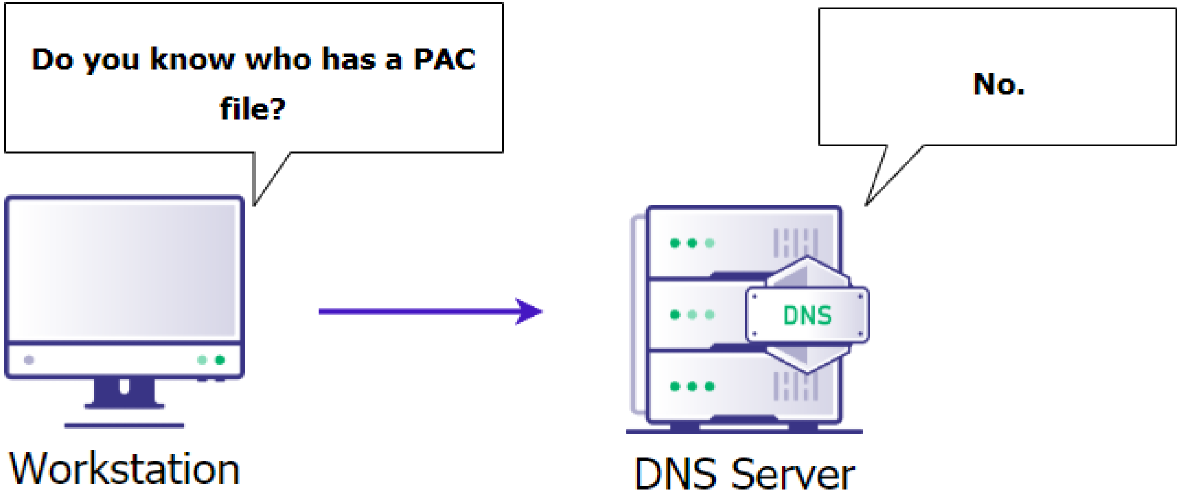


Do you have a PAC file?

No.

Workstation

DHCP Server

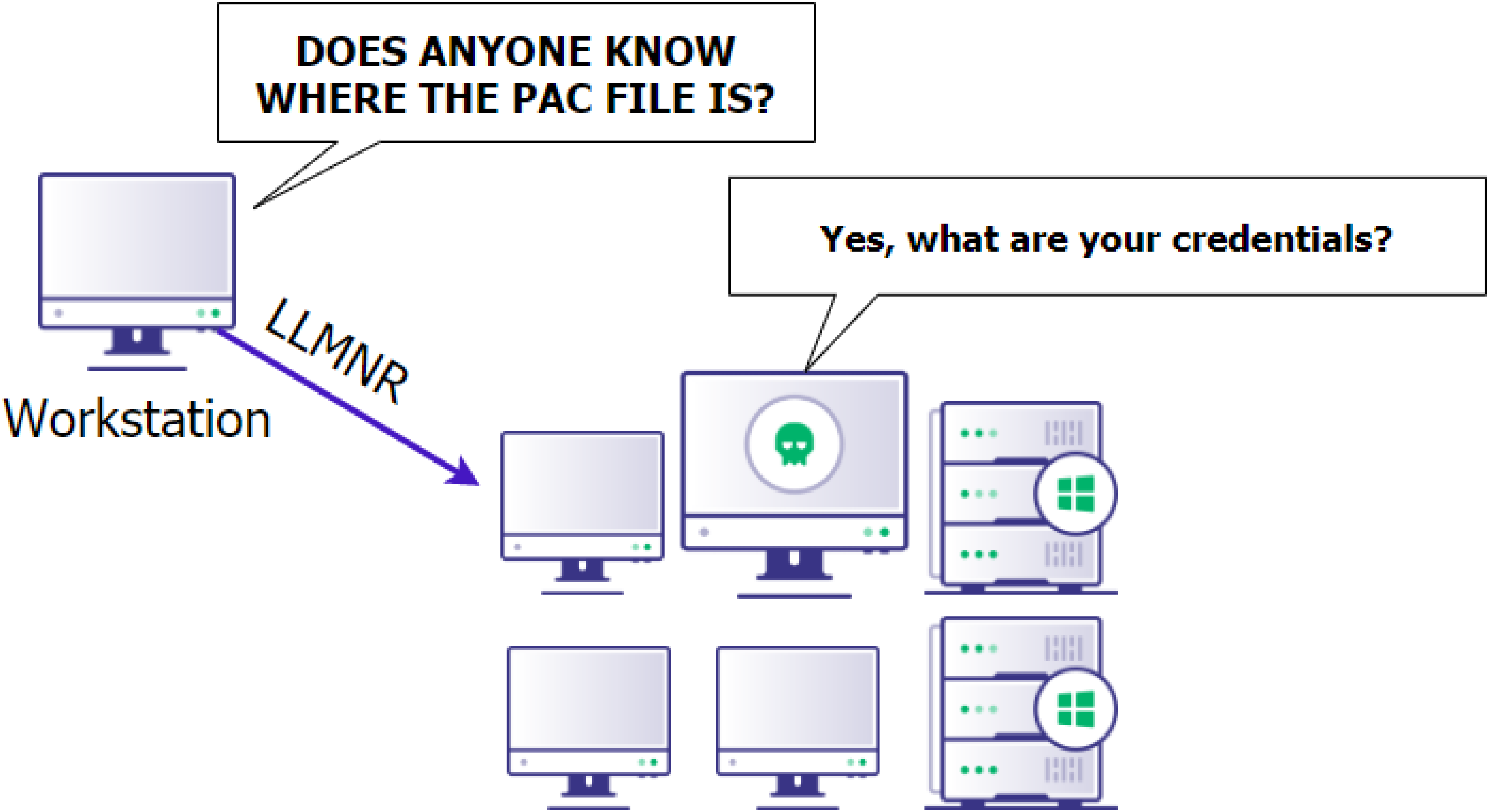


Do you know who has a PAC file?

No.

Workstation

DNS Server

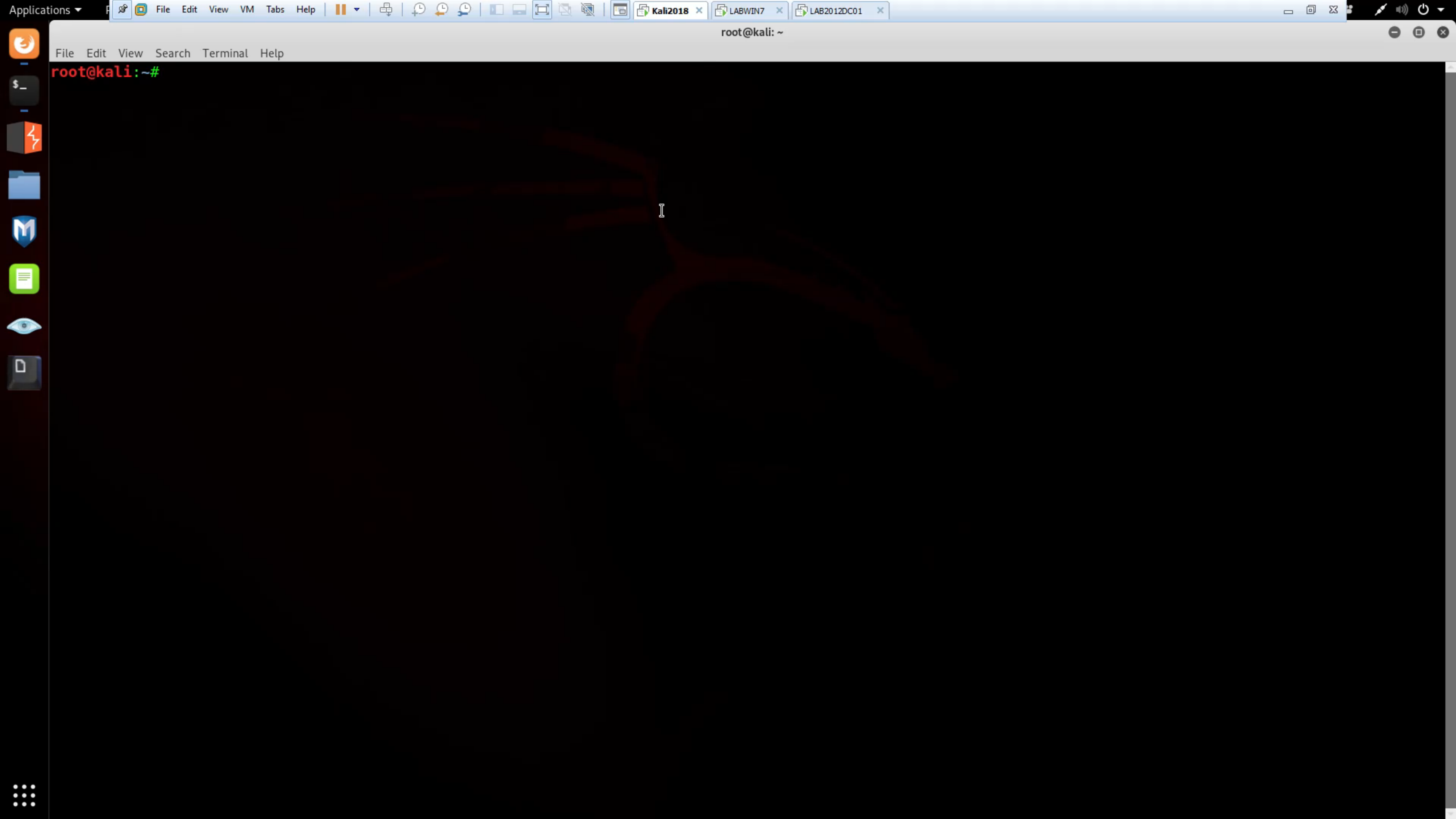


DOES ANYONE KNOW WHERE THE PAC FILE IS?

LLMNR

Workstation

Yes, what are your credentials?



Applications
File Edit View VM Tabs Help
Kali2018
LABWIN7
LAB2012DC01
root@kali: ~
File Edit View Search Terminal Help

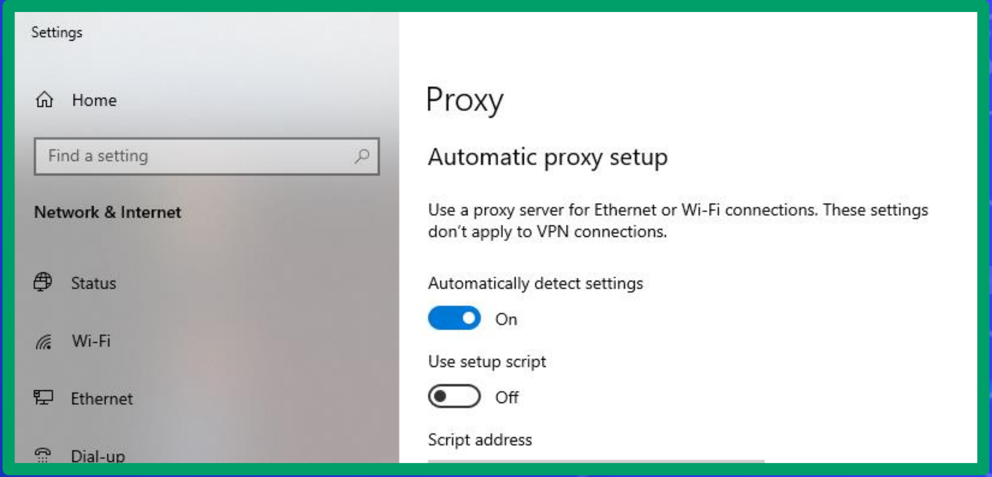
```
root@kali:~#
```

# Mitigations

- Turn off via Group Policy
- **LLMNR:** Computer Configuration -> Administrative Templates -> Network -> DNS ClientEnable Turn Off Multicast Name Resolution
- **NBT-NS**: Network Connection Properties -> TCP/IPv4 -> Advanced, WINS Tab -> Disable NetBIOS over TCP/IP
- Should not impact anything, if something is relying on LLMNR or NBTNS, it’s probably broken already
  - FQDNs

# Mitigations - WPAD

- Turn off via Group Policy
- Create a DNS entry for ‘wpad’
- Apply patch MS16-077
  - The location of the WPAD file is no longer requested via broadcast protocols, but only via DNS.



Proxy

Automatic proxy setup

Use a proxy server for Ethernet or Wi-Fi connections. These settings don’t apply to VPN connections.

Automatically detect settings

On

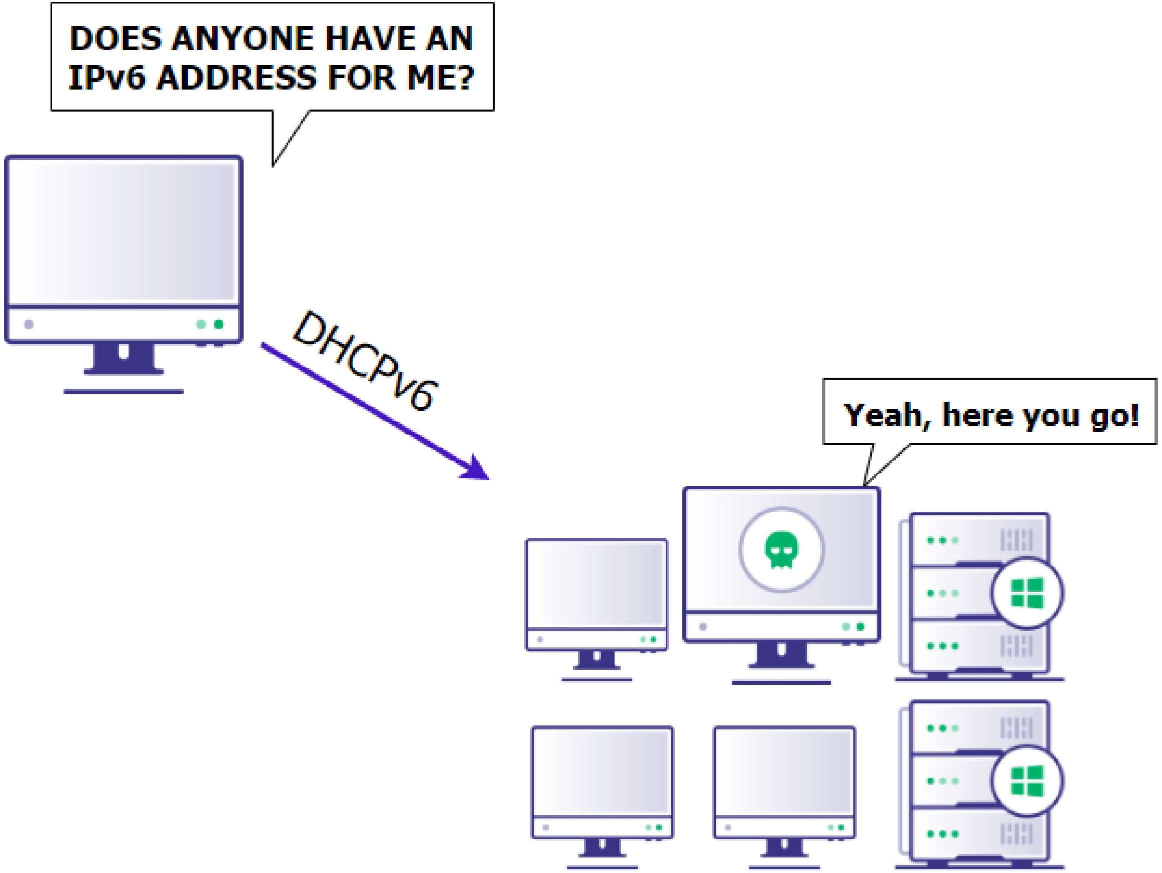
Use setup script

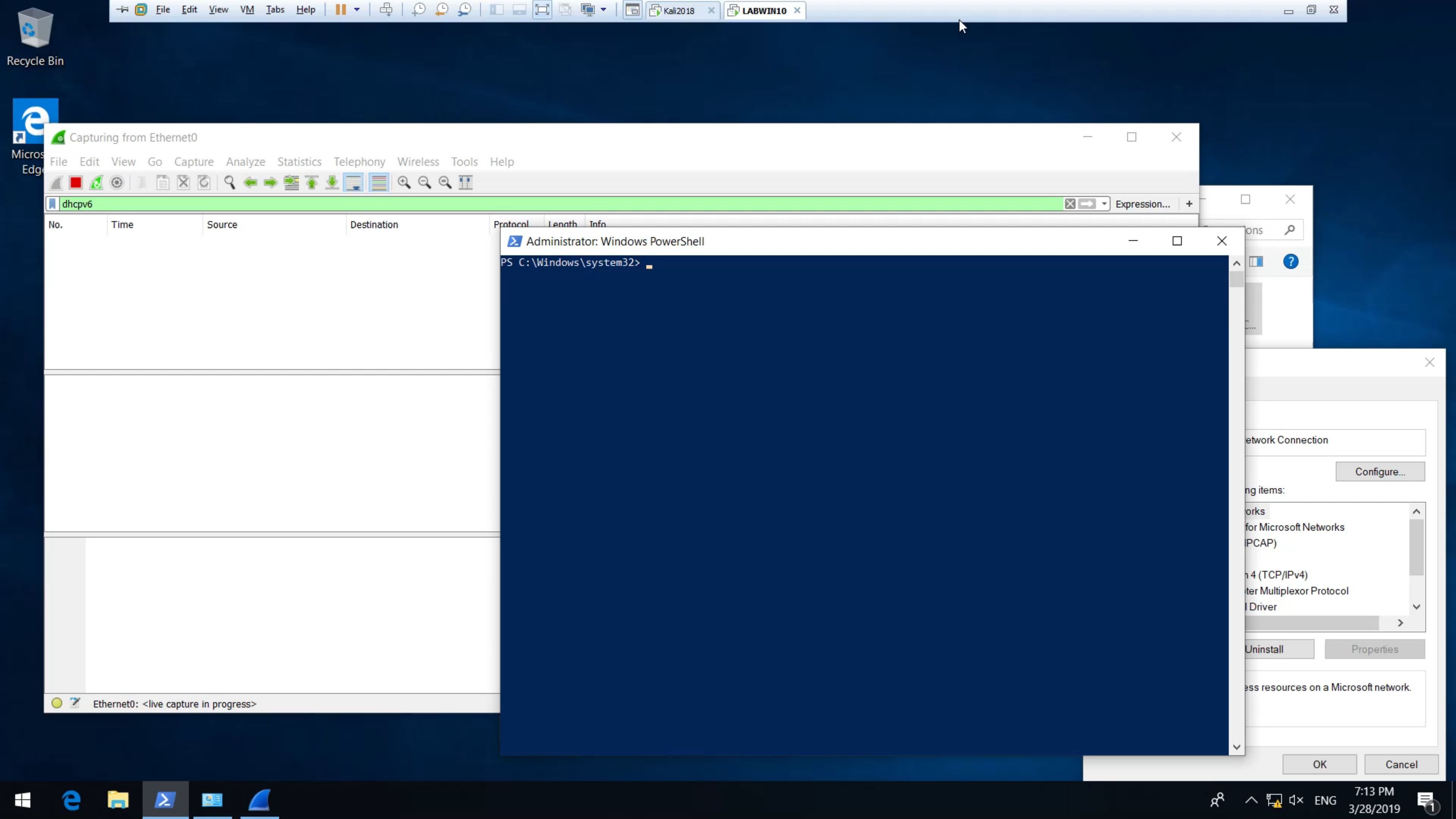
Off

Script address

# IPv6 Spoofing

- IPv6 – “Replacement” for IPv4
  - Not widely used for internal networks
- 192.168.1.1 – IPv4
- fe80::88ae:e421:f660:2616%9– IPv6
- Problem: DHCPv6
- Bigger Problem: Windows prefers IPv6 by default


DOES ANYONE HAVE AN IPv6 ADDRESS FOR ME?
DHCPv6
Yeah, here you go!



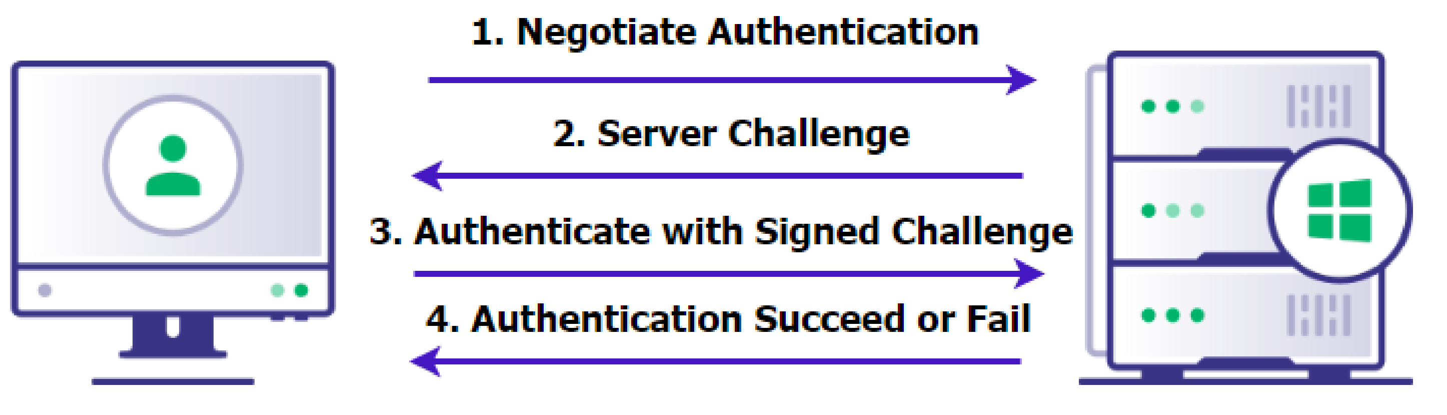
dhcpv6

```
PS C:\Windows\system32> _
```

Ethernet0: <live capture in progress>

# We control DNS. What now?

- NTLM Hashes can be passed (Pass-the-hash)
- Net-NTLMv2 Hashes cannot be passed
- Hash relaying



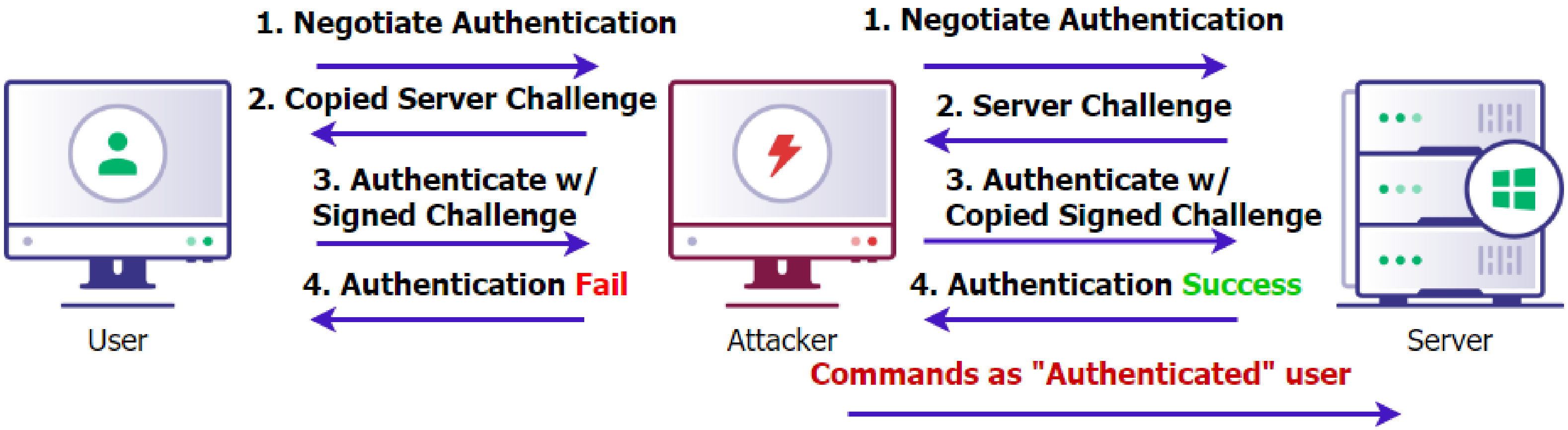
1. Negotiate Authentication

2. Server Challenge

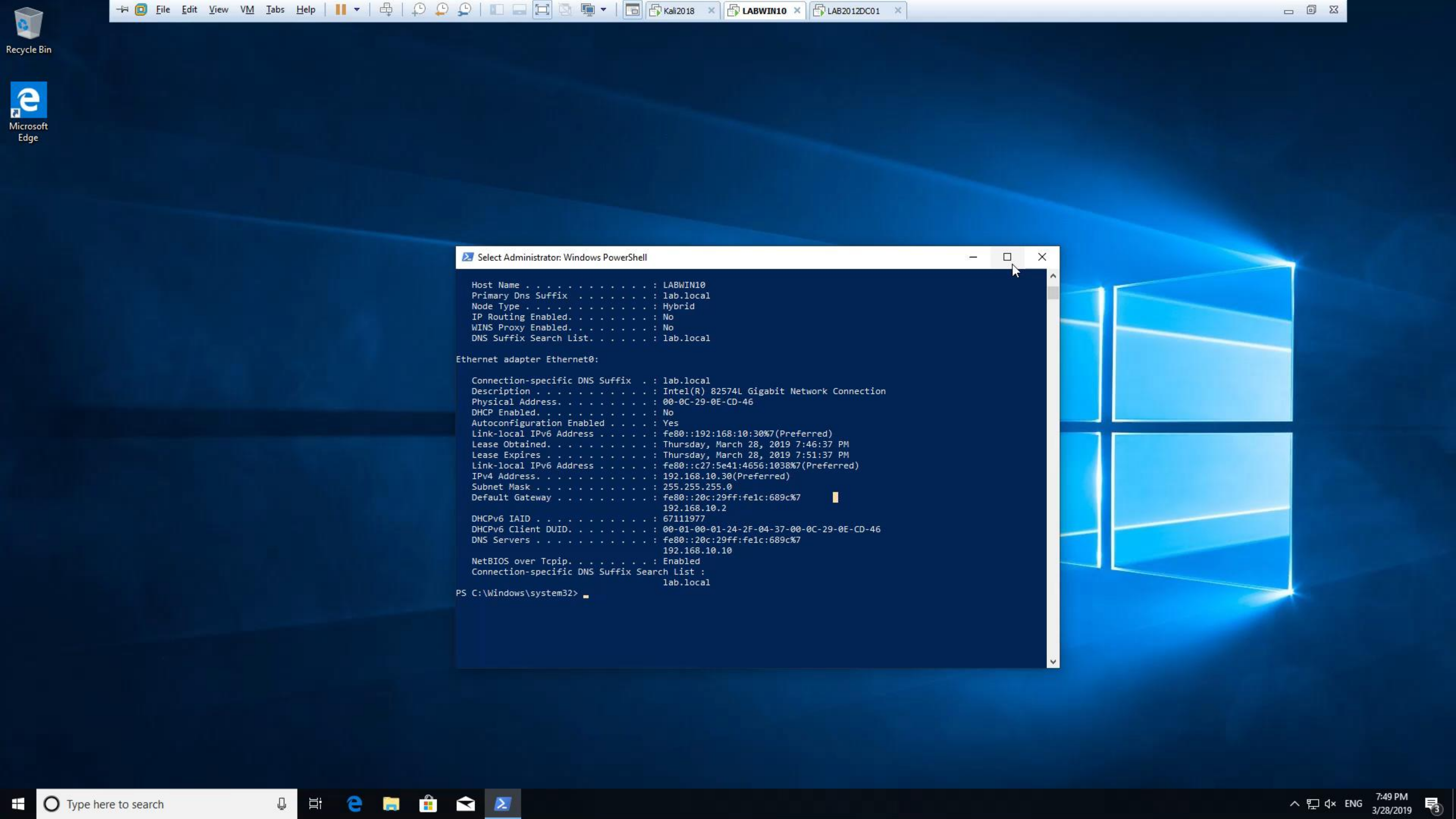
3. Authenticate with Signed Challenge

4. Authentication Succeed or Fail

# Hash Relaying Overview



1. Negotiate Authentication
2. Copied Server Challenge
3. Authenticate w/ Signed Challenge
4. Authentication Fail
User
Attacker
1. Negotiate Authentication
2. Server Challenge
3. Authenticate w/ Copied Signed Challenge
4. Authentication Success
Server
Commands as "Authenticated" user



```
   Host Name . . . . . . . . . . . . : LABWIN10
   Primary Dns Suffix  . . . . . . . : lab.local
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : lab.local

Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . : lab.local
   Description . . . . . . . . . . . : Intel(R) 82574L Gigabit Network Connection
   Physical Address. . . . . . . . . : 00-0C-29-0E-CD-46
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::192:168:10:30%7(Preferred)
   Lease Obtained. . . . . . . . . . : Thursday, March 28, 2019 7:46:37 PM
   Lease Expires . . . . . . . . . . : Thursday, March 28, 2019 7:51:37 PM
   Link-local IPv6 Address . . . . . : fe80::c27:5e41:4656:1038%7(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.10.30(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::20c:29ff:fe1c:689c%7
                                       192.168.10.2
   DHCPv6 IAID . . . . . . . . . . . : 67111977
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-24-2F-04-37-00-0C-29-0E-CD-46
   DNS Servers . . . . . . . . . . . : fe80::20c:29ff:fe1c:689c%7
                                       192.168.10.10
   NetBIOS over Tcpip. . . . . . . . : Enabled
   Connection-specific DNS Suffix Search List :
                                       lab.local
PS C:\Windows\system32>
```

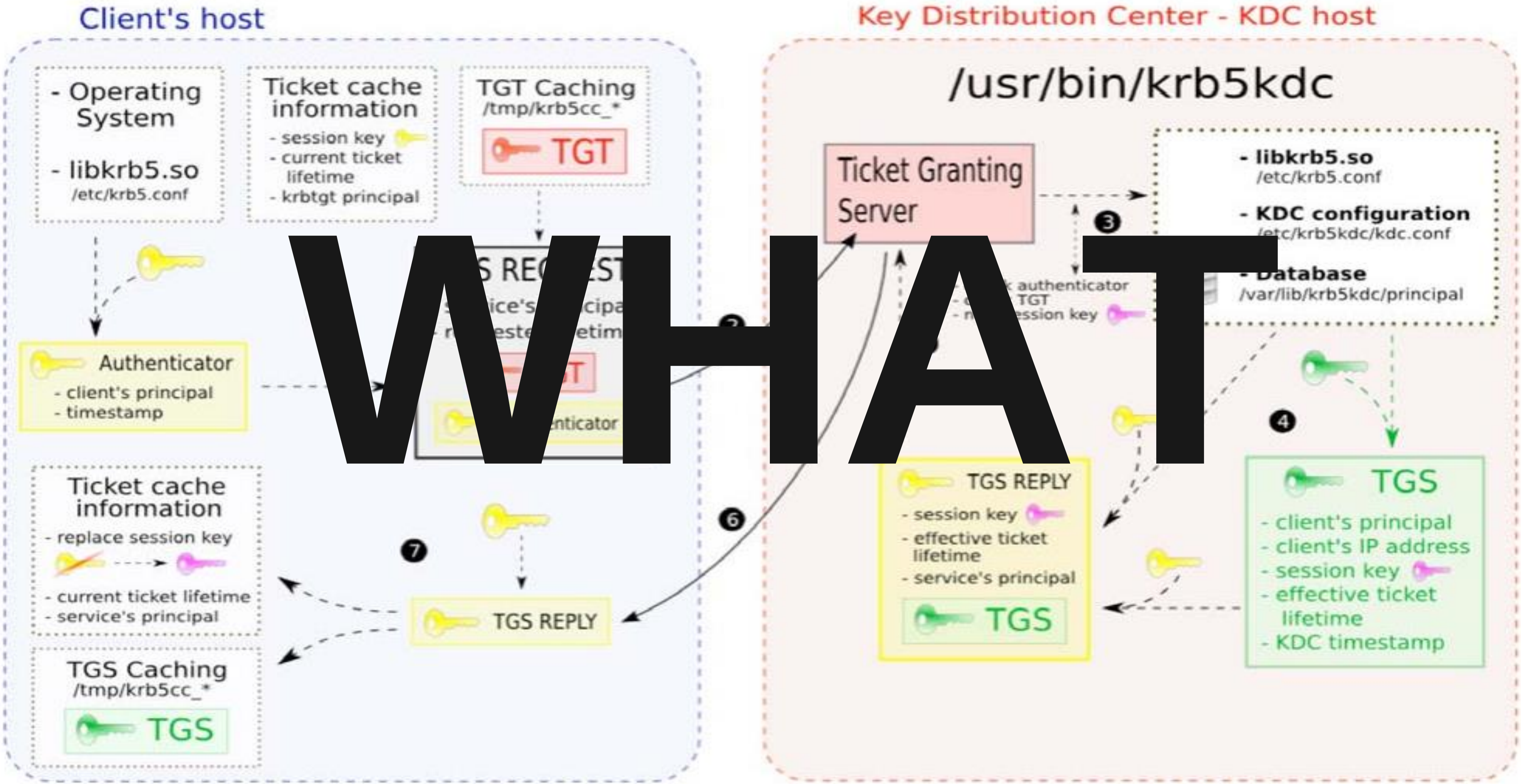
# Mitigations

-
  ```
  reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip6\Parameters" /v DisabledComponents /t REG_DWORD /d 0 /f
  ```
- DNSSEC
- SMB Signing

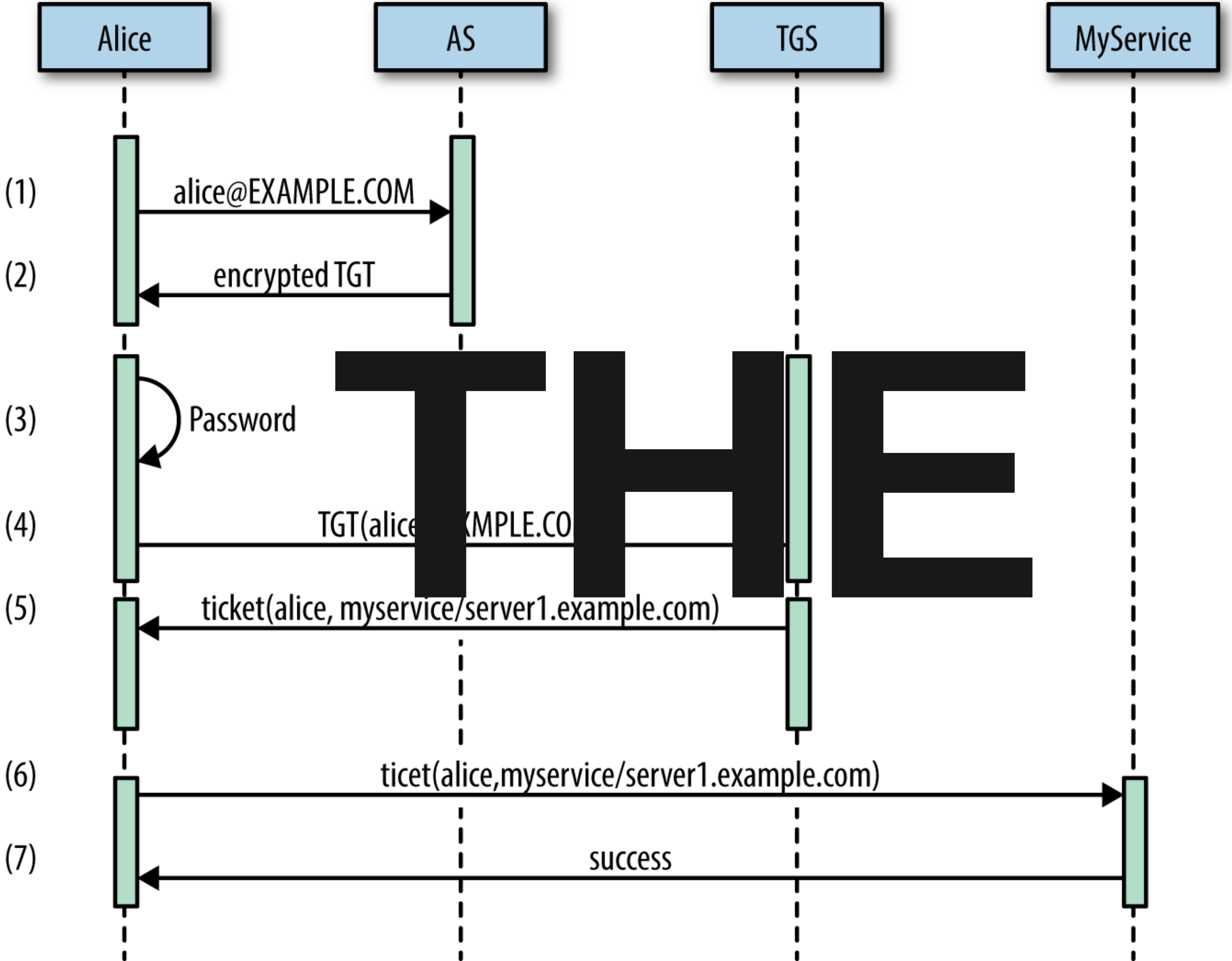
# 2. PrivEsc via Kerberos

Condensing chaos into 5 minutes

# KerberosV5 Ticket Granting Service - TGS delivery



WHAT


Alice
AS
TGS
MyService
(1)
alice@EXAMPLE.COM
(2)
encrypted TGT
(3)
Password
(4)
TGT(alice
XMPLE.CO
(5)
ticket(alice, myservice/server1.example.com)
(6)
ticet(alice,myservice/server1.example.com)
(7)
success

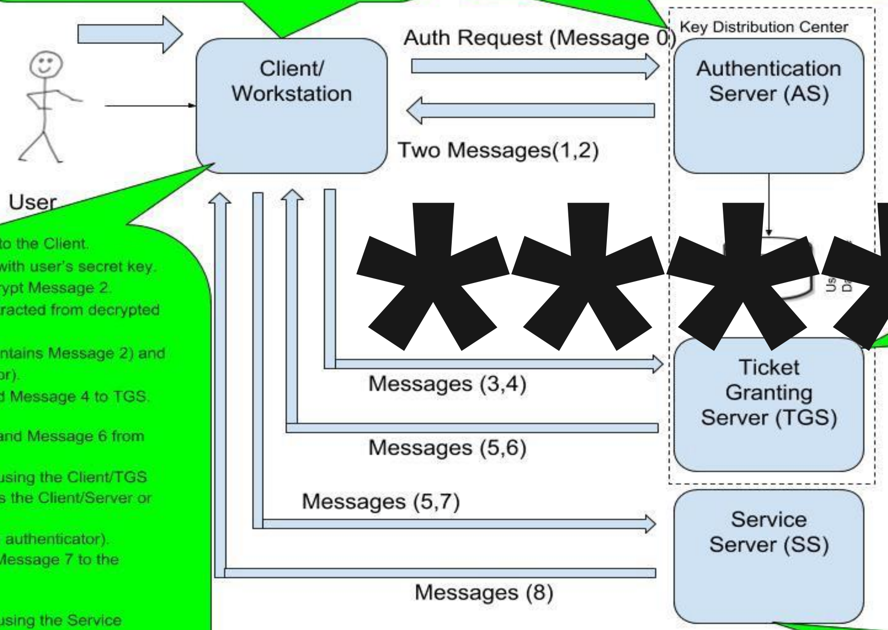
THE

# Kerberos Protocol v5

1. User logs into Client by providing username and password
2. Client converts password to a symmetric key (through scheduling algorithm or one-way hash)—this is the Client's secret key (or symmetric key).

3. Client sends authentication request (Message 0, see below) to the AS.
4. AS searches for user in its database.
5. If found, AS uses user's password to generate secret that then encrypts the response Message 1. If not found, an error message is returned to the Client.
6. AS uses the TGS secret key to encrypt Message 2.

**User:** The human being logging into the client
**Client:** The device (workstation, laptop, phone, tablet, etc) the user is logging into.
**Authentication Server (AS):** logical component of KDC that maintains a database of principals (users, clients, and servers) and their secret keys.
**Ticket Granting Server (TGS):** a logical KDC component that issues tickets.
**Service Server (SS):** Server hosting a service the client wishes to access securely.
**Key Distribution Center (KDC):** service that is responsible for authenticating users and issuing tickets (composed of AS + TGS).



7. AS returns Messages 1& 2 to the Client.
8. Client decrypts Message 1 with user's secret key. Note, the Client cannot decrypt Message 2.
9. Client/TGS Session Key extracted from decrypted Message 1.
10. Client builds Message 3 (contains Message 2) and Message 4 (the authenticator).
11. Client sends Message 3 and Message 4 to TGS.

---

19. Client receives Message 5 and Message 6 from TGS.
20. Client decrypts Message 6 using the Client/TGS Session Key. Client now has the Client/Server or Service Session Key.
21. Client builds Message 7 (an authenticator).
22. Client sends Message 5 & Message 7 to the Service Server (SS).

---

27. Client decrypts message 8 using the Service Session Key.
28. Check timestamp on Message 8.
29. The client now trusts the SS.
30. Client now interacts with SS as needed.

12. TGS intercepts Message 3 and Message 4.
13. Retrieve Message 2 from Message 3.
14. TGS decrypts Message 2 using the the TGS secret key. TGS now has the client/TGS session key.
15. TGS uses this key to decrypt Message 4 (the authenticator).
16. TGS compares client ID from Message 3 & Message 4.
17. If match, TGS generates Message 5 & Message 6. Otherwise, error.
18. TGS sends Message 5 and Message 6 to Client.

Steps 1-9: Authentication Service Exchange
Steps: 10-20: Ticket Granting Service Exchange
Steps: 21-30: Application Service Exchange

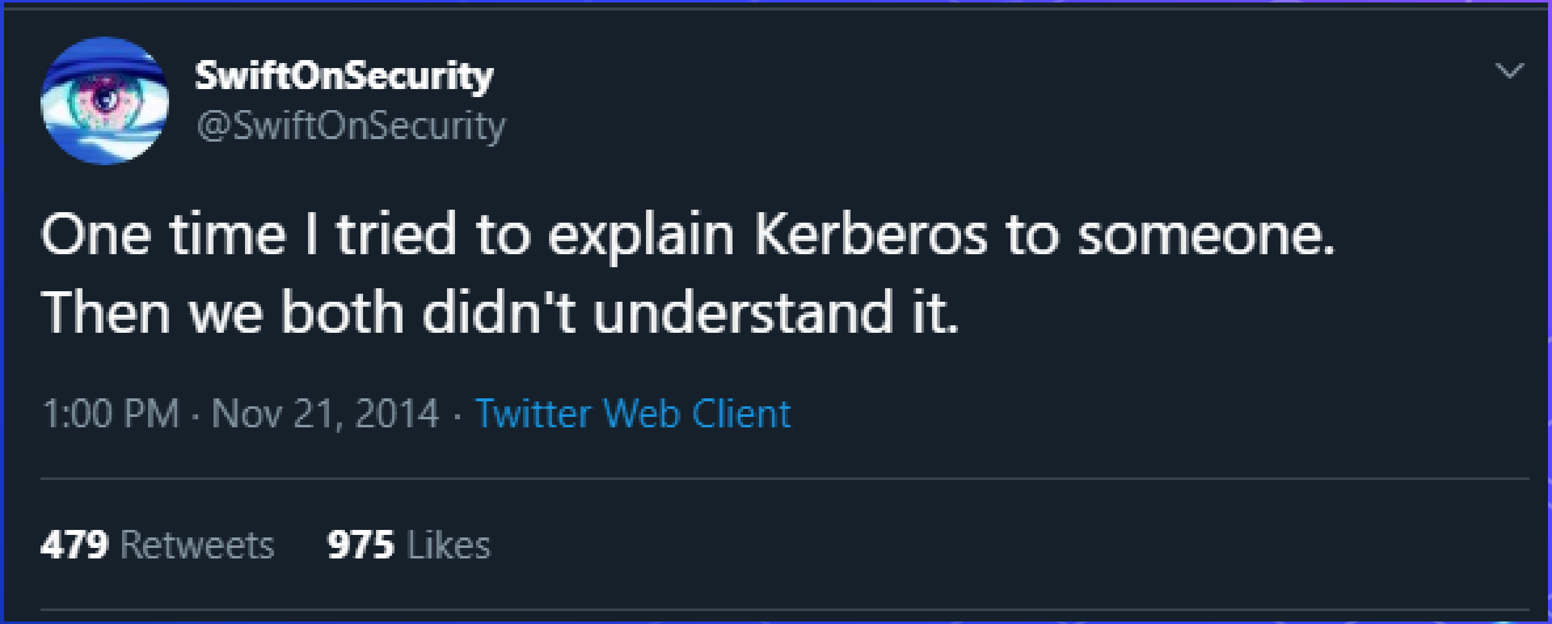
23. SS decrypts the ticket (Message 5) using its own secret key to retrieve the Client/Server Session Key.
24. Using the session key, SS decrypts the Authenticator.
25. Compare client ID from message 5 and 7.
26. If match, send Message 8 to client to confirm server identity and willingness to service client request (this part is optional).

User Login

Client-Service Authorization

Client-Service Request

Repeated once for each service client wants to access



SwiftOnSecurity
@SwiftOnSecurity

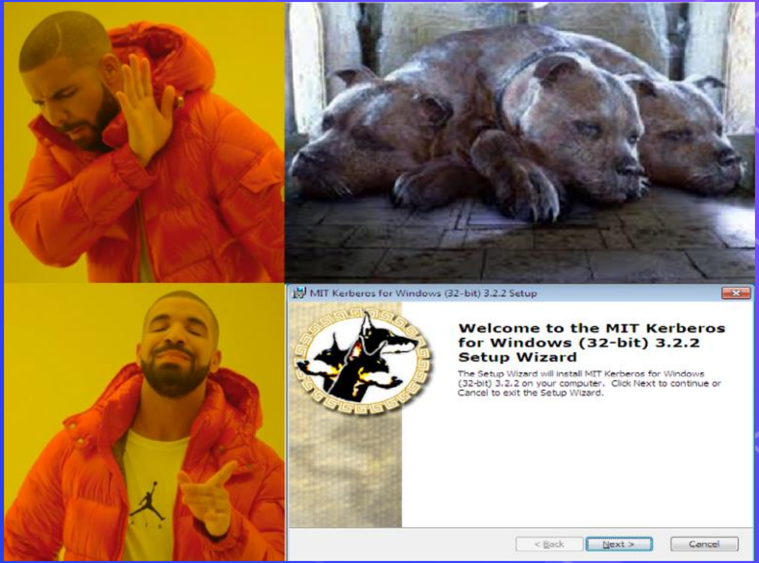
One time I tried to explain Kerberos to someone. Then we both didn't understand it.

1:00 PM · Nov 21, 2014 · Twitter Web Client

479 Retweets 975 Likes

# Kerberos Overview

- Protocol, Alternative to NTLM Authentication
- Preferred way of authentication via tickets
- Complex
- Really Complex
- Think SSO, but for Windows



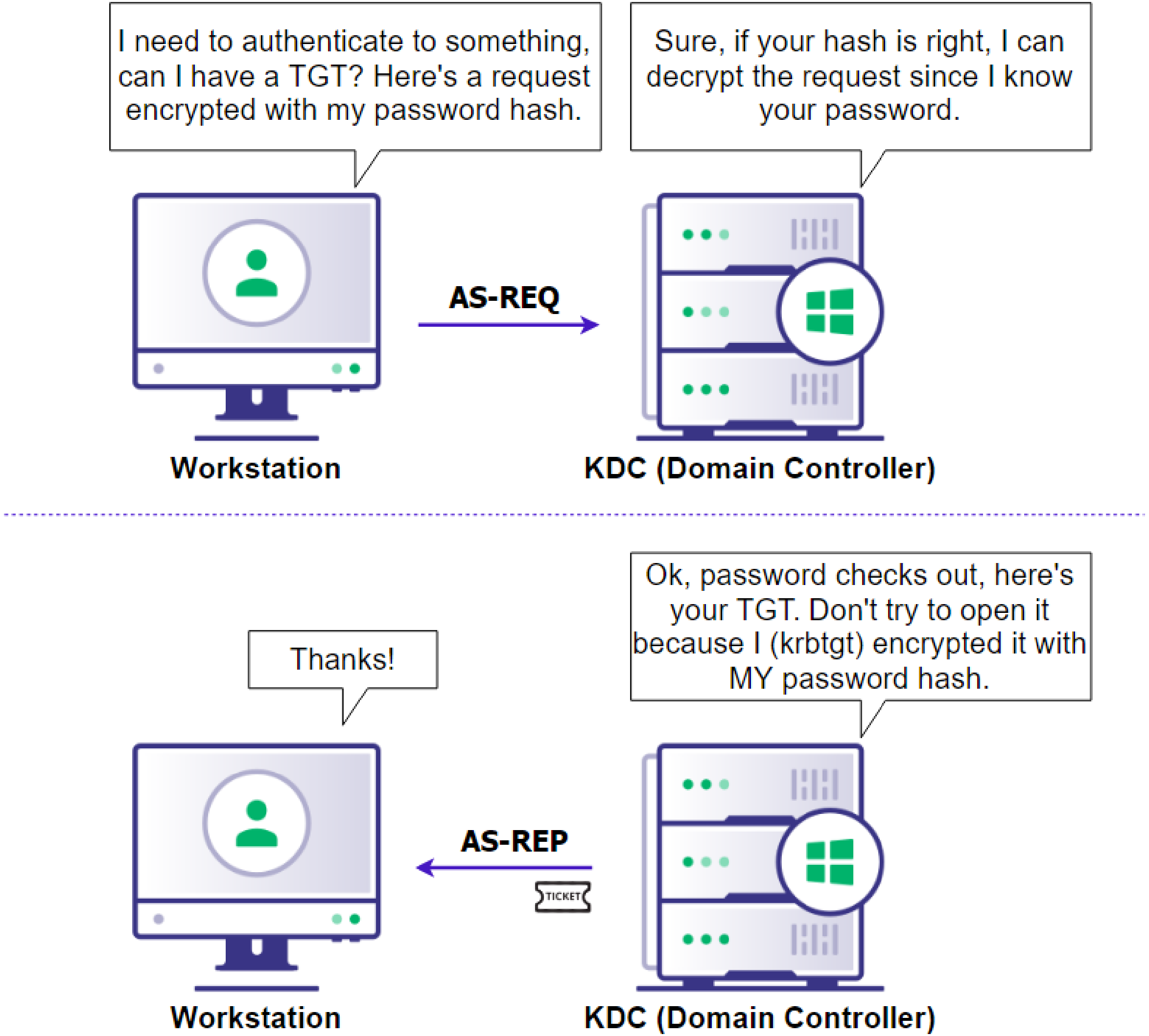
MIT Kerberos for Windows (32-bit) 3.2.2 Setup

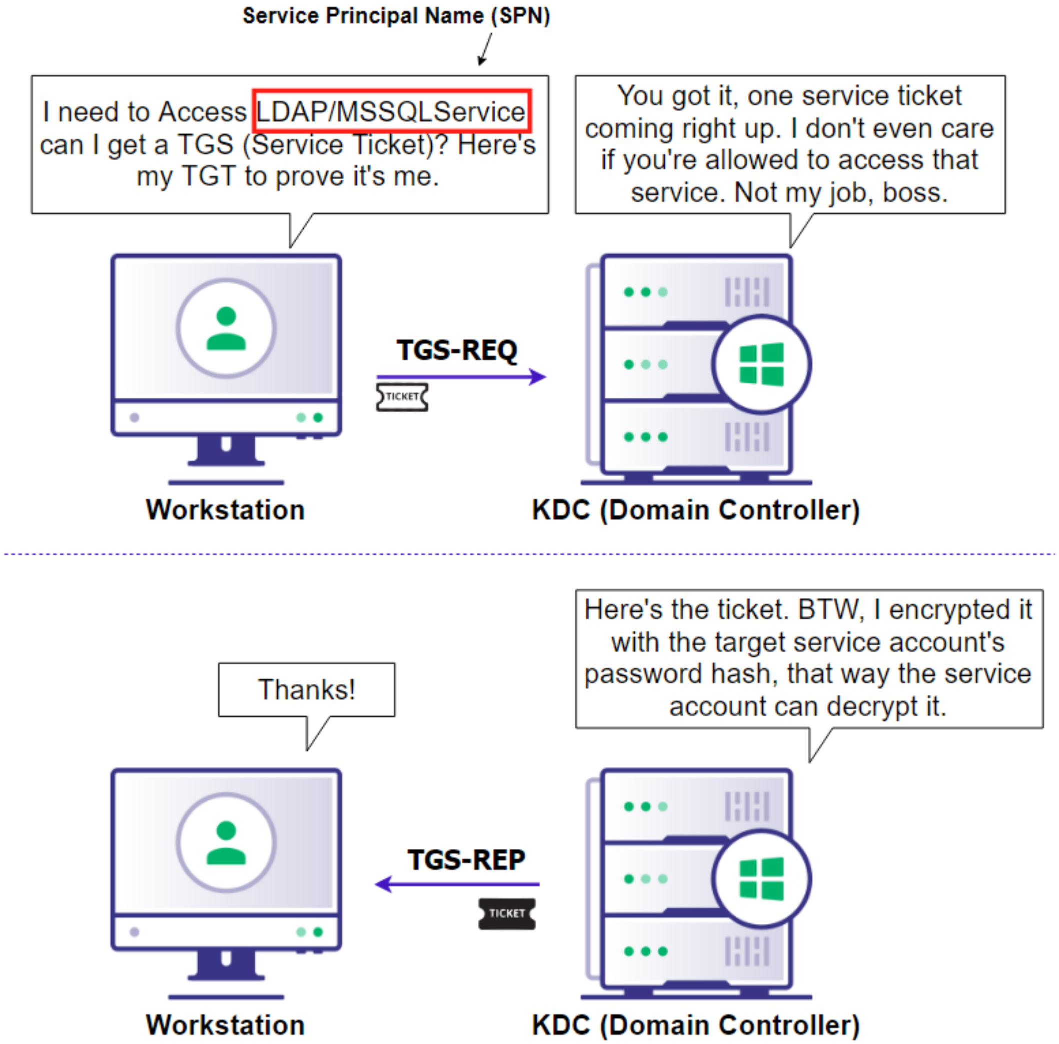
**Welcome to the MIT Kerberos for Windows (32-bit) 3.2.2 Setup Wizard**

The Setup Wizard will install MIT Kerberos for Windows (32-bit) 3.2.2 on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

# Kerberos-ulary

- Key Distribution Center (KDC) – Domain Controller that does the authentication
- Service Principal Name (SPN) – Unique name for a service account
  - E.g.: CIFS/LABDC01.LAB.LOCAL
  - Done through setspn.exe – Creates SPN for a user account
- Ticket Granting Ticket (TGT) – Used to authenticate to the KDC
- Ticket Granting Server (TGS) – A service ticket

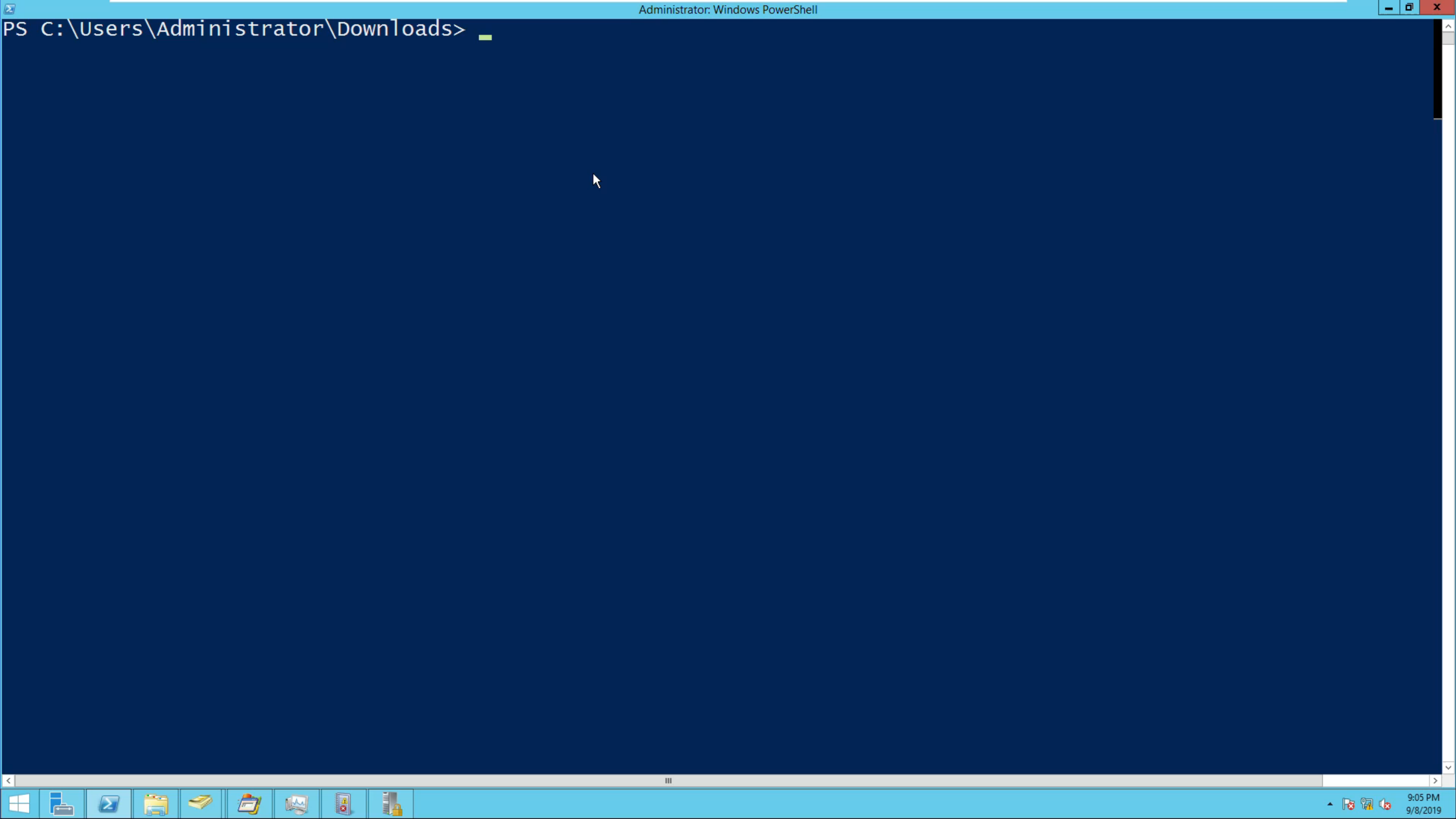

I need to authenticate to something, can I have a TGT? Here's a request encrypted with my password hash.
Sure, if your hash is right, I can decrypt the request since I know your password.
AS-REQ
Workstation
KDC (Domain Controller)
Ok, password checks out, here's your TGT. Don't try to open it because I (krbtgt) encrypted it with MY password hash.
Thanks!
AS-REP
TICKET
Workstation
KDC (Domain Controller)


Service Principal Name (SPN)
I need to Access LDAP/MSSQLService can I get a TGS (Service Ticket)? Here's my TGT to prove it's me.
You got it, one service ticket coming right up. I don't even care if you're allowed to access that service. Not my job, boss.
TGS-REQ
TICKET
Workstation
KDC (Domain Controller)
Here's the ticket. BTW, I encrypted it with the target service account's password hash, that way the service account can decrypt it.
Thanks!
TGS-REP
TICKET
Workstation
KDC (Domain Controller)

# Kerberoasting

- Requires credentials, but privileges are irrelevant
- Request the TGS ticket, which has the password hash of the SPN's account, then crack it offline.
  - Contains the hash because that's the only thing the DC and server have in common, so it's used for decryption
- Can be requested by any authenticated user

# Kerberoasting Demo



```
PS C:\Users\Administrator\Downloads> _
```

# Mitigations

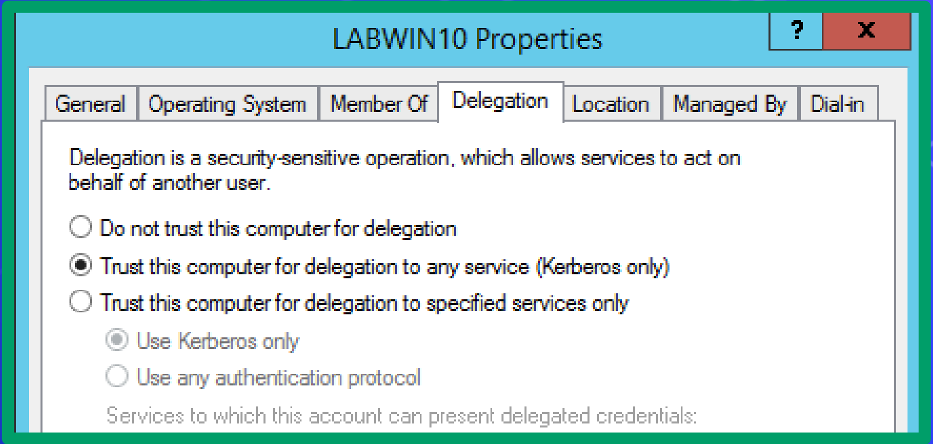
- Have a very long password for your accounts with SPNs
- Make sure no users have SPNs

# Delegation Attacks

- Delegation - A feature that allows a user or computer to impersonate another account
  - Unconstrained
  - Constrained*
  - Resource-Based Constrained*

# Unconstrained Delegation

- Unconstrained – User authenticates to a service on a server with a TGS for service. The service extracts the user’s TGT from the TGS to use for other TGS requests.
  - Ticket stored in memory
  - Printer bug



Delegation is a security-sensitive operation, which allows services to act on behalf of another user.

○ Do not trust this computer for delegation

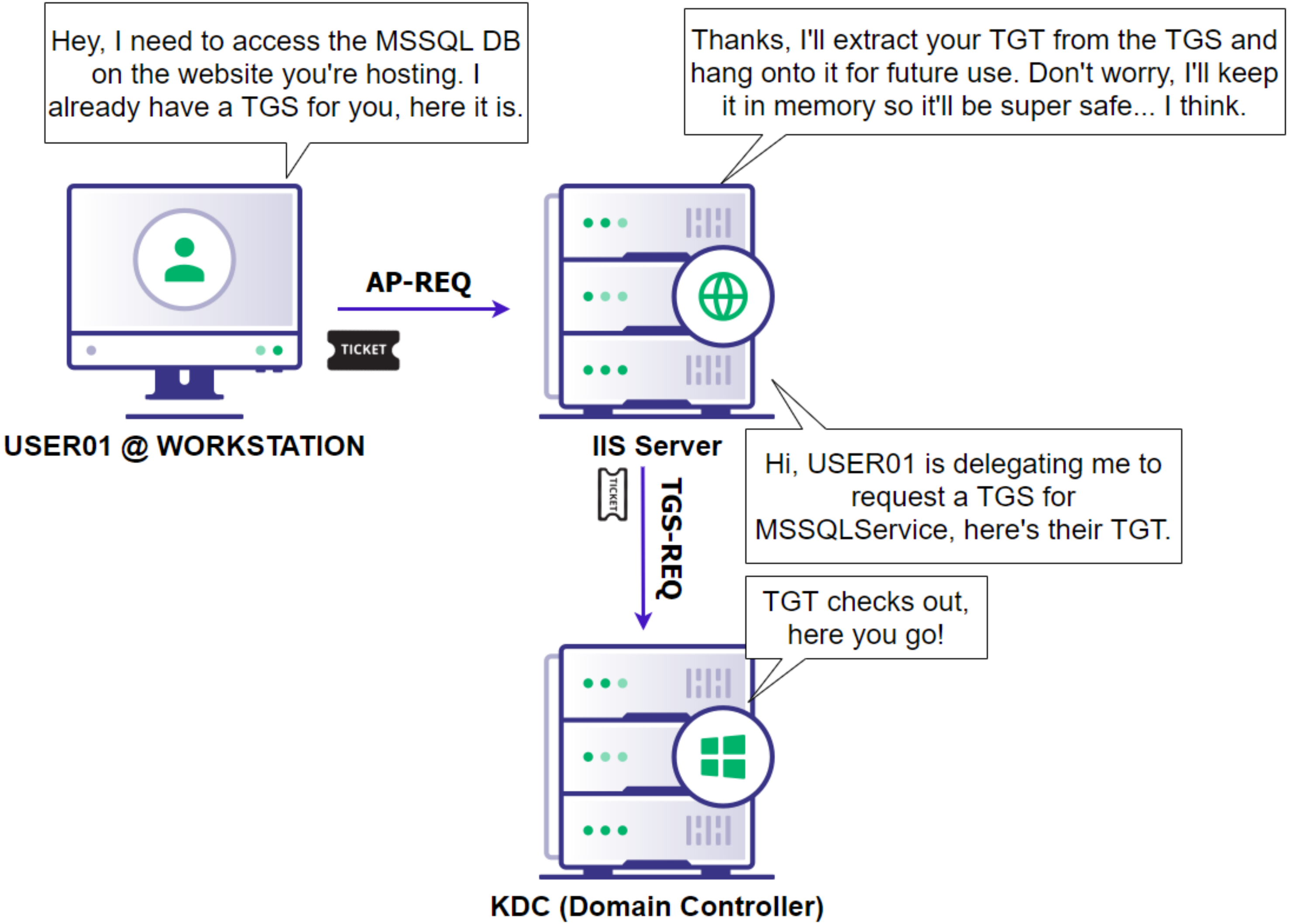
● Trust this computer for delegation to any service (Kerberos only)

○ Trust this computer for delegation to specified services only

● Use Kerberos only

○ Use any authentication protocol

Services to which this account can present delegated credentials:



Hey, I need to access the MSSQL DB on the website you're hosting. I already have a TGS for you, here it is.

Thanks, I'll extract your TGT from the TGS and hang onto it for future use. Don't worry, I'll keep it in memory so it'll be super safe... I think.

AP-REQ

TICKET

USER01 @ WORKSTATION

IIS Server

TICKET

TGS-REQ

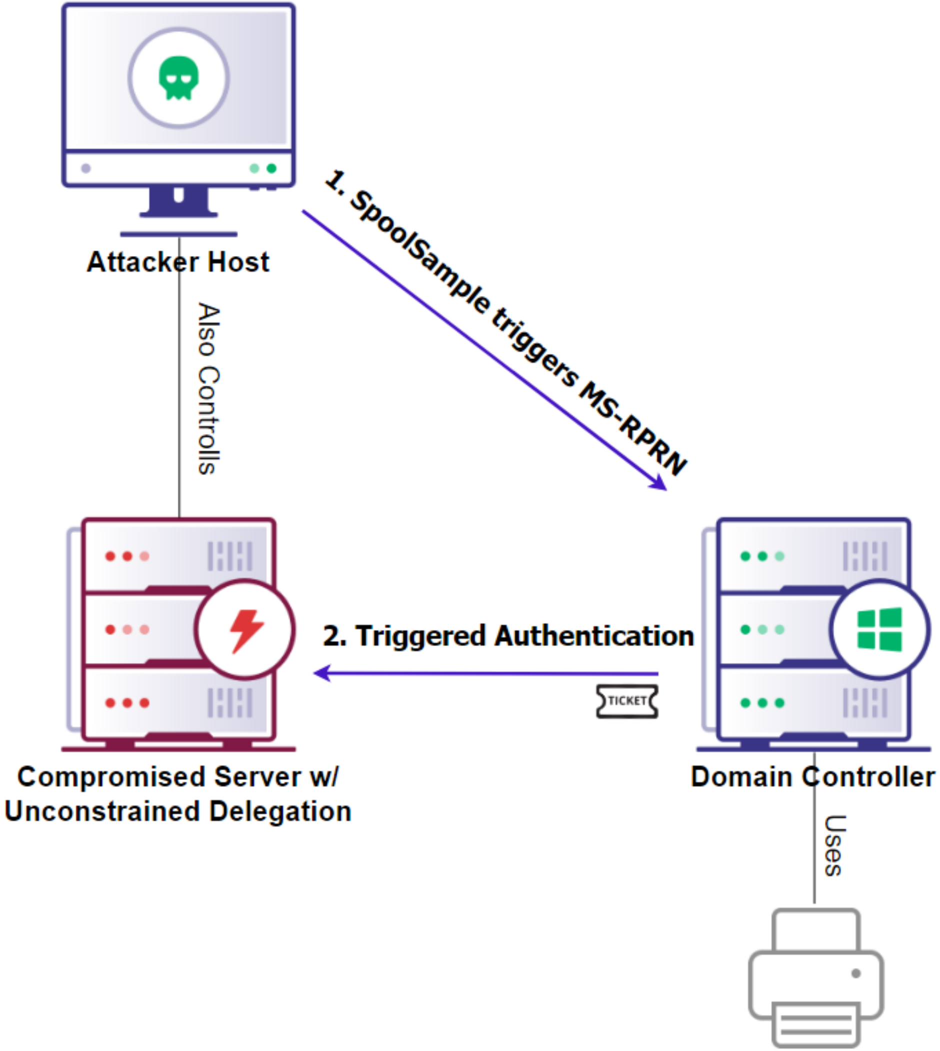
Hi, USER01 is delegating me to request a TGS for MSSQLService, here's their TGT.

TGT checks out, here you go!

KDC (Domain Controller)

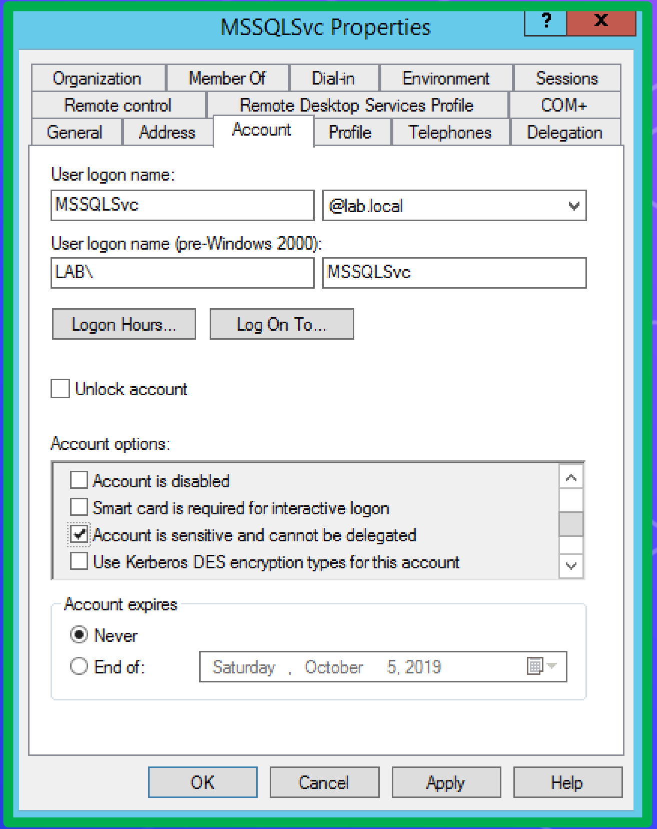
# Printer Bug

- Coerces a machine (e.g. domain controller) that has a printer setup on it, to authenticate to a host of our choosing via SpoolSample
  - Tool written by @tifkin_ to use the Print System Remote Protocol (MS-RPRN) to trigger authentication.
- Coerce a DC to authenticate to a host that we control that has unconstrained delegation on = win


Attacker Host
1. SpoolSample triggers MS-RPRN
Also Controlls
2. Triggered Authentication
TICKET
Compromised Server w/
Unconstrained Delegation
Domain Controller
Uses

# Mitigations

- Ensure sensitive accounts cannot be delegated



MSSQLSvc Properties

User logon name: MSSQLSvc @lab.local

User logon name (pre-Windows 2000): LAB\ MSSQLSvc

□ Unlock account

Account options:

□ Account is disabled
□ Smart card is required for interactive logon
☑ Account is sensitive and cannot be delegated
□ Use Kerberos DES encryption types for this account

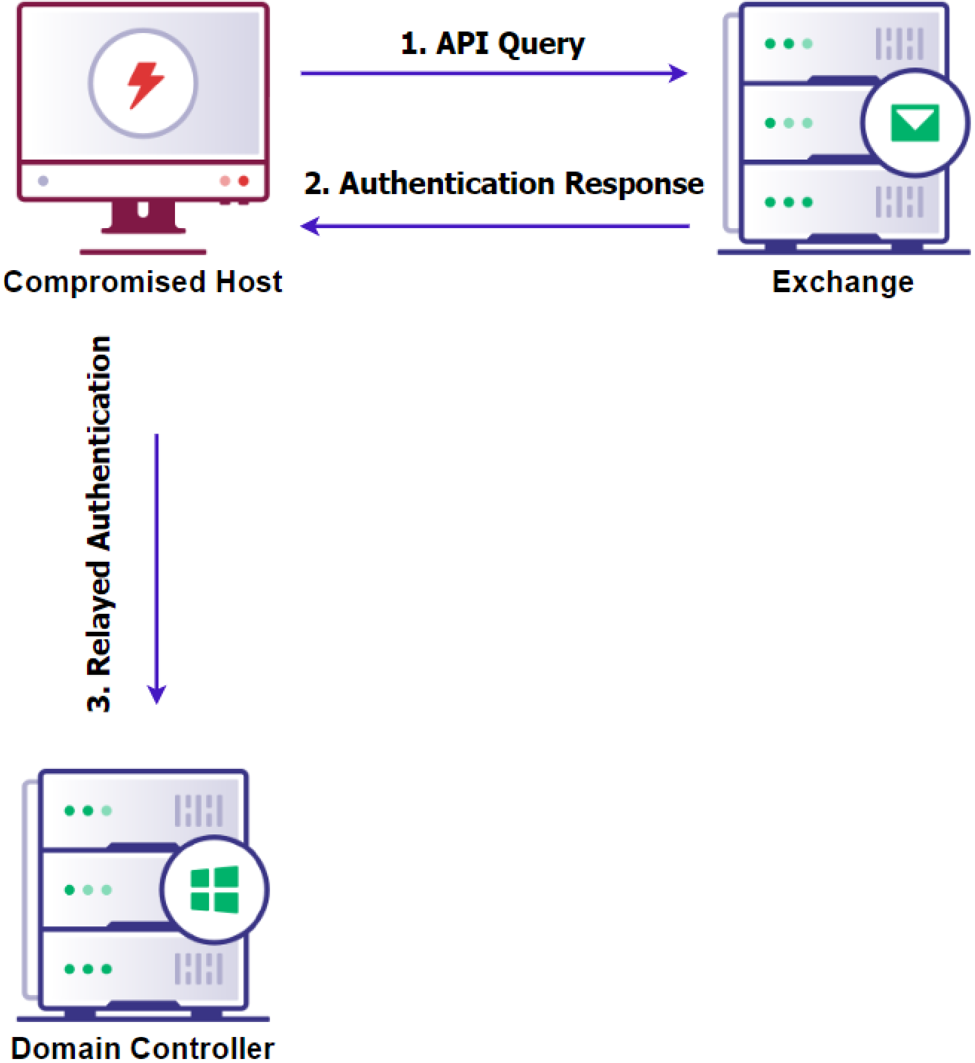
Account expires

● Never
○ End of: Saturday , October 5, 2019

BUT WAIT
E
THERE'S MORE

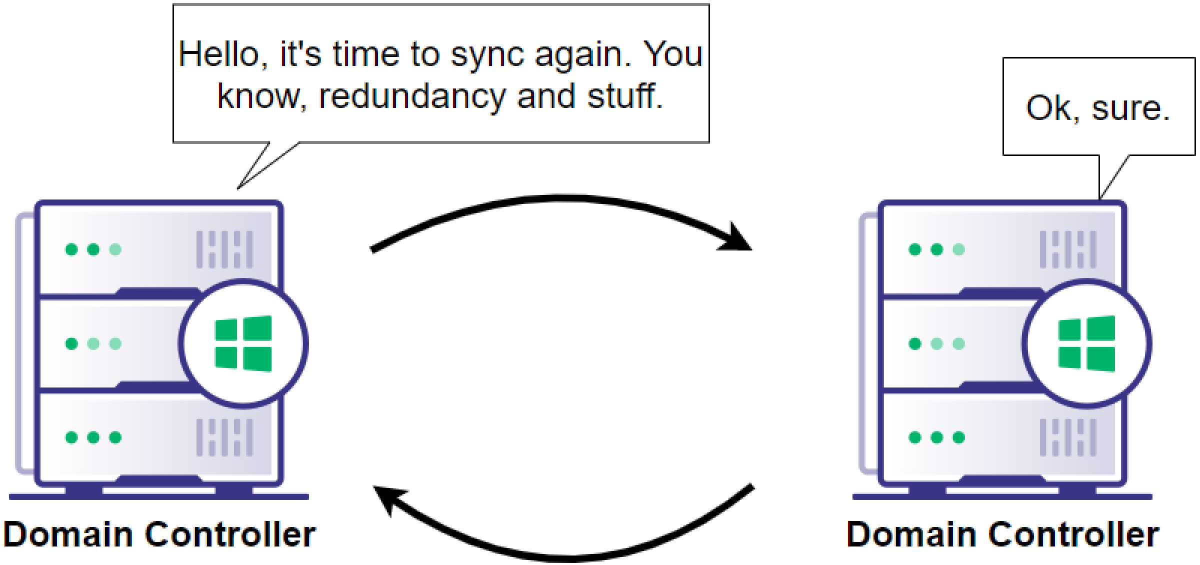
# PrivExchange

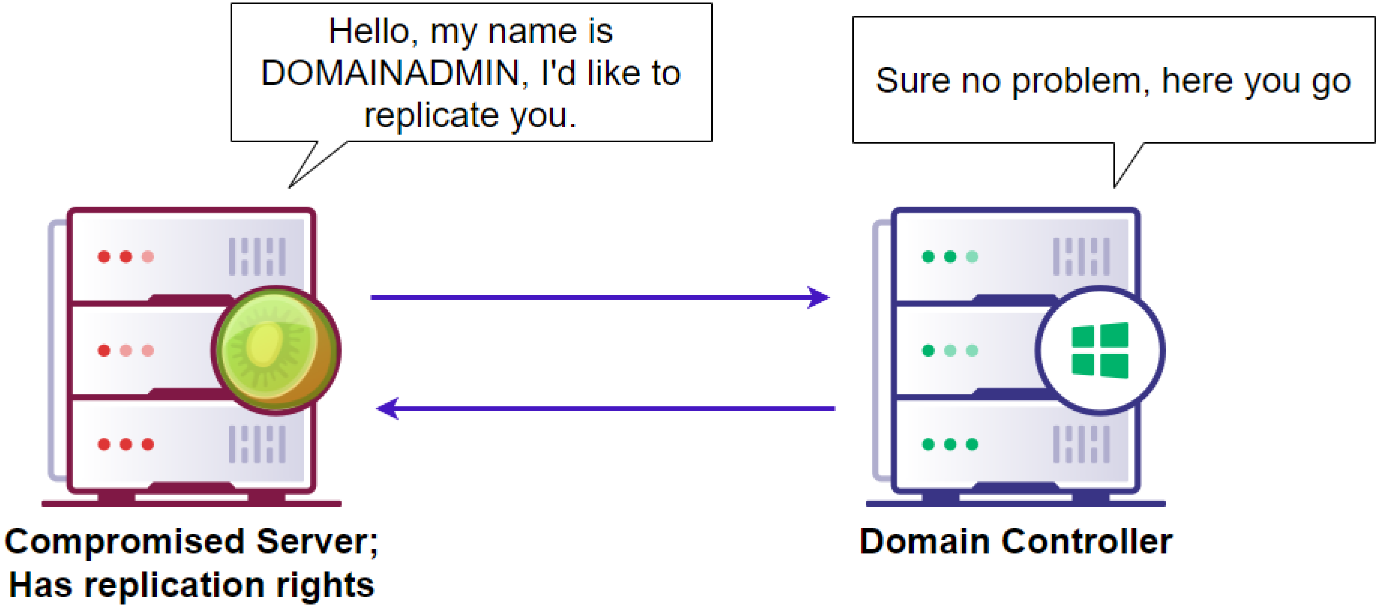
- Leverages the fact that Exchange servers are over-privileged
- Also done via relaying credentials
- Works by making an API call to Exchange, which sends a response with the Exchange server's credentials
    - Requires only a mailbox
- Once machine creds are relayed, the supplied user credentials are then added to the Enterprise Admins group


1. API Query
2. Authentication Response
Compromised Host
Exchange
3. Relayed Authentication
Domain Controller

# DCSync

- Feature
- Replication
- Mimikatz implemented the functionality


Hello, it's time to sync again. You know, redundancy and stuff.
Ok, sure.
Domain Controller
Domain Controller

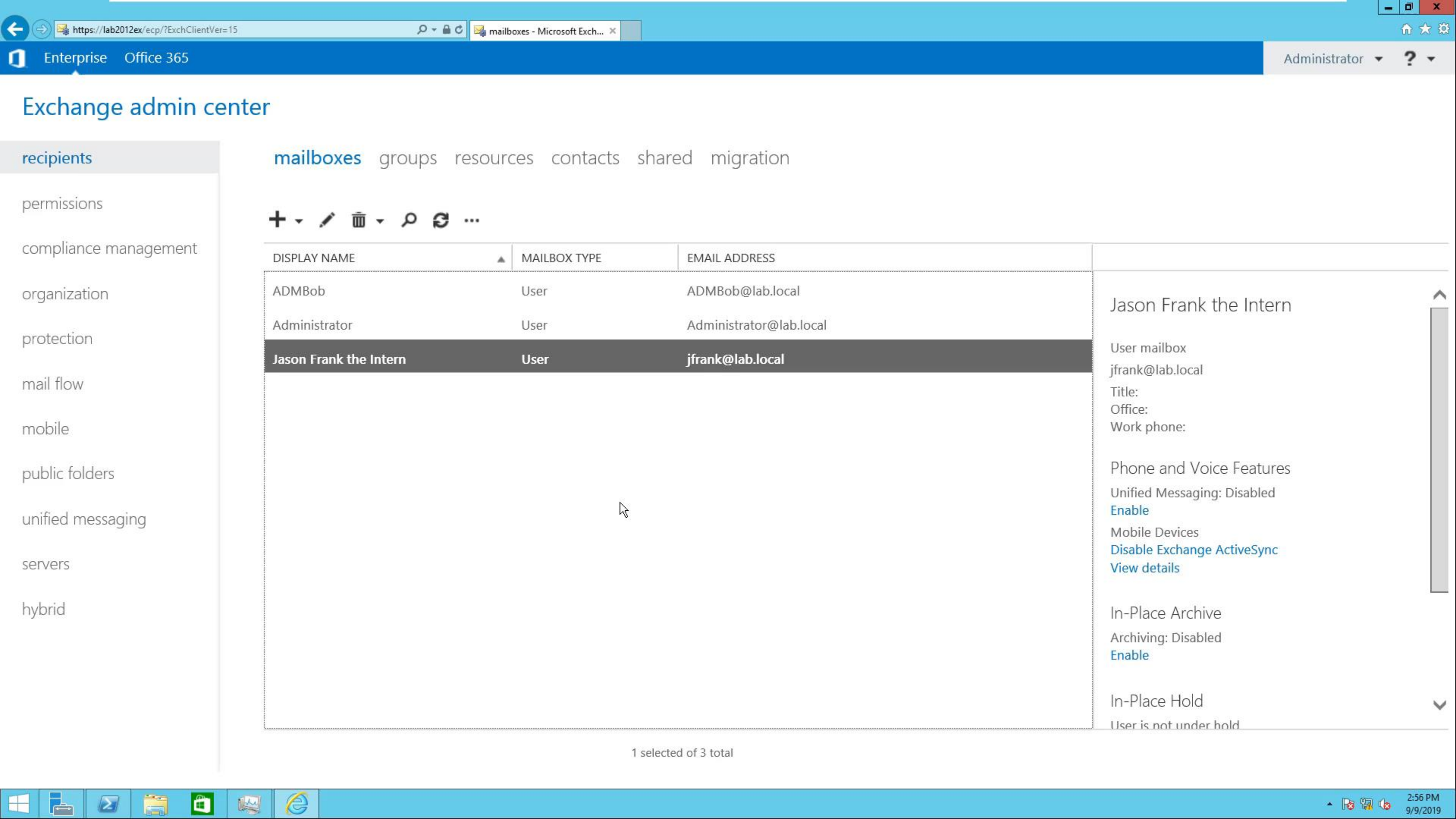


Hello, my name is DOMAINADMIN, I'd like to replicate you.

Sure no problem, here you go

**Compromised Server; Has replication rights**

**Domain Controller**



# Exchange admin center

| DISPLAY NAME | MAILBOX TYPE | EMAIL ADDRESS |
|---|---|---|
| ADMBob | User | ADMBob@lab.local |
| Administrator | User | Administrator@lab.local |
| **Jason Frank the Intern** | **User** | **jfrank@lab.local** |

1 selected of 3 total

## Jason Frank the Intern

User mailbox
jfrank@lab.local
Title:
Office:
Work phone:

### Phone and Voice Features

Unified Messaging: Disabled
Enable

Mobile Devices
Disable Exchange ActiveSync
View details

### In-Place Archive

Archiving: Disabled
Enable

### In-Place Hold

User is not under hold

# Mitigations

- KB4490060

# KRBTGT

- Account’s hash used to encrypt TGTs
- Created by default when installing AD DS
- Bad news if compromised

# Golden and Silver Tickets

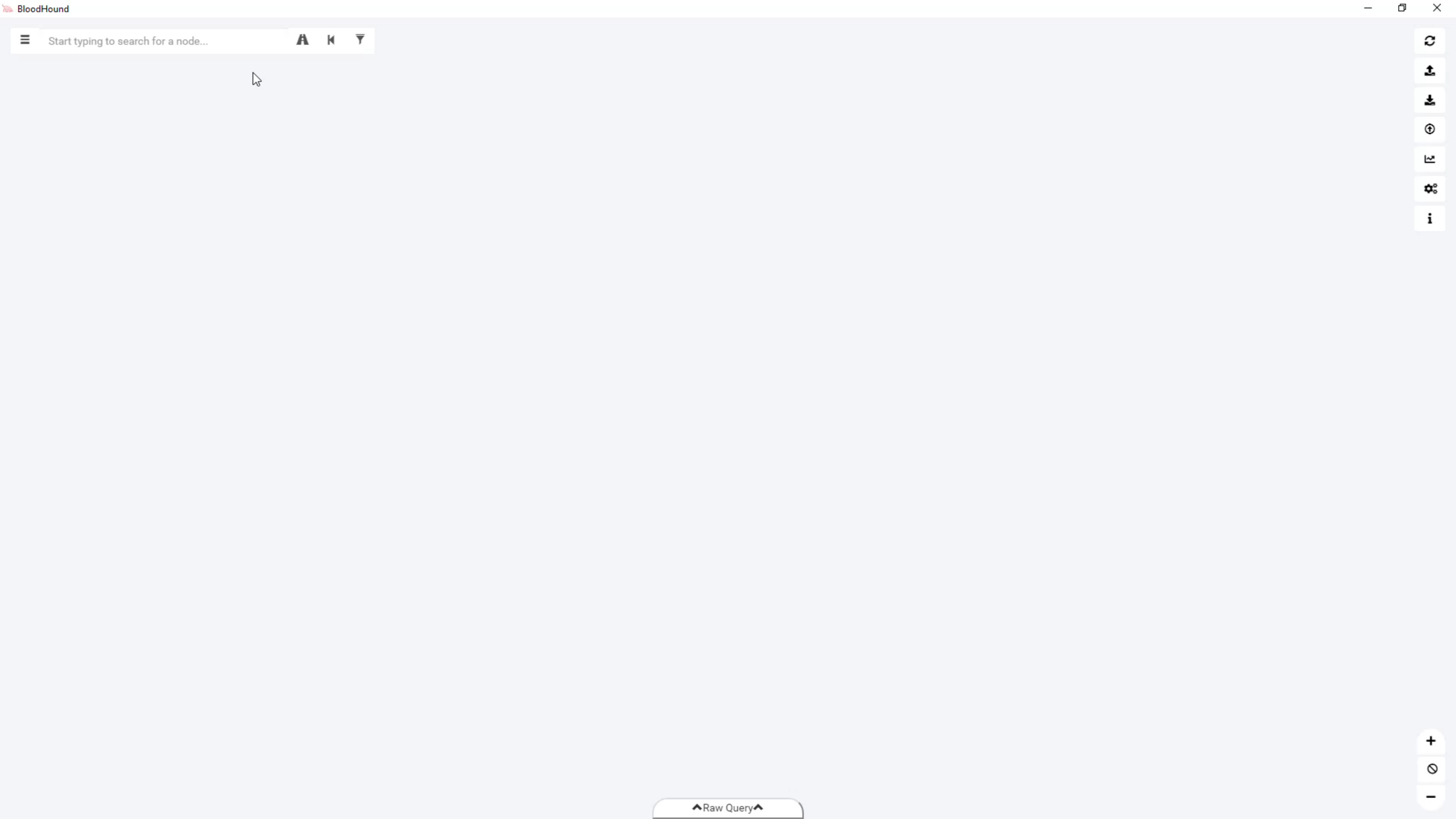
- Golden Ticket – When the KRBTGT account hash is compromised and the attacker can forge any ticket for any account.
- Silver Ticket – When the service or machine account hash is compromised and is used to forge a service ticket for that specific service

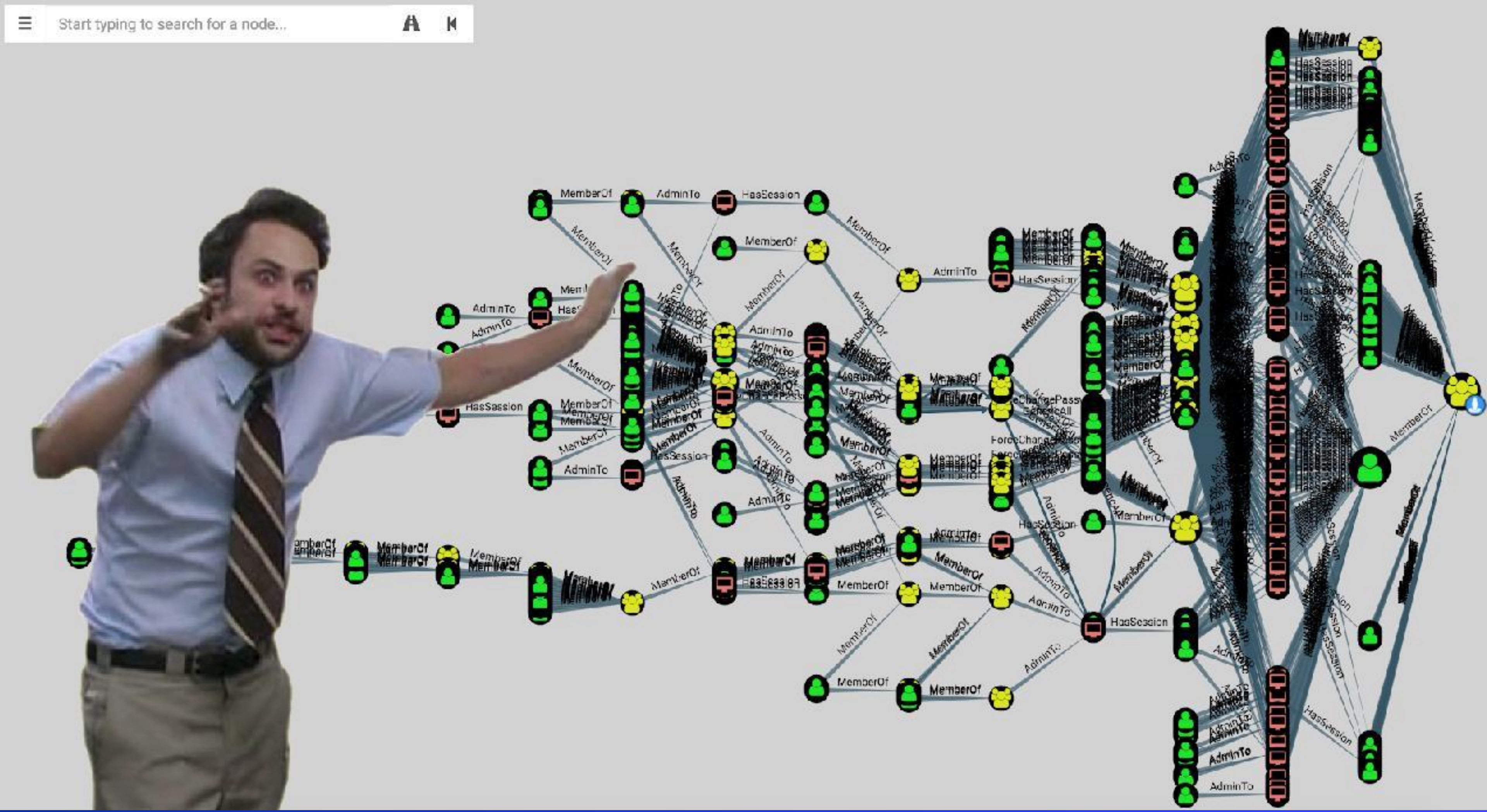
“*How can I defend myself against these attacks?*”

# Introducing the sniffy boi

# Bloodhound

- Graphs the domain to reveal relationships between objects within Active Directory
- EXTREMELY useful for an attacker
    - Shows attack paths
- EXTREMELY useful for the defense


BloodHound
Start typing to search for a node...
Raw Query



Resource: https://twitter.com/dudeslce/status/1111526726041890816?s=19

# Credits

- Will Schroeder for everything Kerberos and answering my questions
- Sean Metcalf for everything else Kerberos
- Lee Christensen for PrinterBug
- Dirk-Jan for PrivExchange
- Tim Medin for Kerberoasting
- SpiderLabs for Impacket
- Bloodhound Slack

# Can I just get the tl;dr please

- Default GP = bad
  - Disable LLMNR
  - Disable WPAD (Or create a DNS entry)
  - Disable IPv6 (If not in use)
- Don’t use Unconstrained Delegation
- Patch your Exchange server
- Use Bloodhound to identify attack paths
- Patch domain controllers
- Questions? @Haus3c
- Link to this deck: