THREAT RESEARCH PAPER

Credential Reuse Attacks in Cloud-Connected Environments

splunk>
turn data into doing™

The Splunk Research Team has developed a new analytic story addressing the recent SolarWinds campaign, which featured TTPs (Golden SAML) that target the extraction of credentials in cloud federated environments. Federation-enabling technologies such as Active Directory Federation Services (ADFS) compose these environments. These federations can be from inside the perimeter or between cloud vendors.

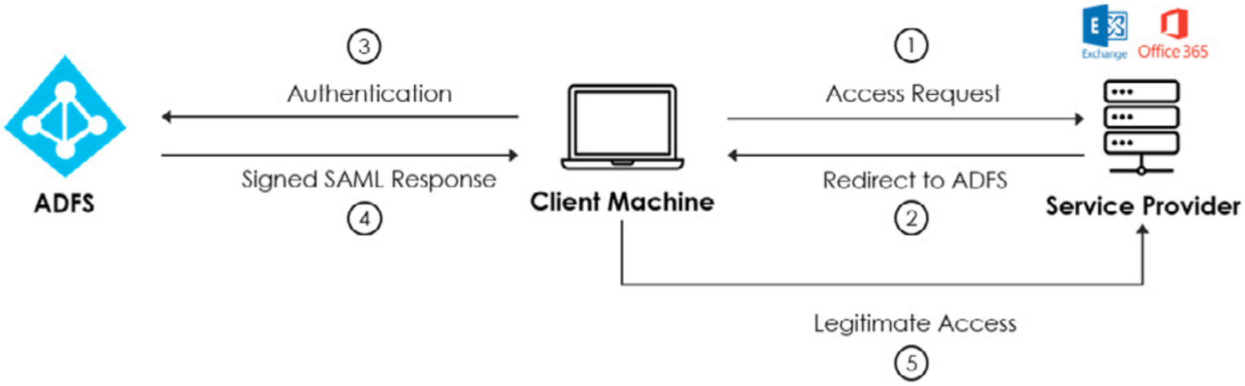
A recent alert from the Cybersecurity and Infrastructure Security Agency provides new attack vectors that target the reuse of credentials against cloud-connected infrastructures. With the widespread adoption of cloud technologies, many companies are now managing environments where the line between the perimeter and the internet is blurred. Such environments allow local users and applications to interact with cloud services. These interactions are usually done via REST API endpoints, and need to be easy, quick and efficient in order to provide a good user experience and use processing power effectively. The constant interaction of these services requires standards for authentication, authorization and validation of trust among the users, applications inside the perimeter and the connected cloud services. Two popular protocols help achieve this purpose: OAuth2 and Security Assertion Markup Language (SAML). These protocols have a similar goal — to allow users or applications to access multiple environments seamlessly. This is especially necessary when organizations use multicloud vendors and applications as part of their infrastructure.

In this research paper we delve into how these credentials operate and how these attacks work within the perimeter and between cloud environments.

OAuth2 tokens are used for making authorized calls to APIs on behalf of a user or application. They are usually stored within the endpoint application session variables and can be extracted and reused in many cases without re-validation against the issuer platform, providing attackers with a way to reuse them and access victim sessions and resources.

SAML is an open standard for exchanging authentication and authorization data between parties. One of the uses of SAML is the ability to perform single sign on (SSO) via the browser into multiple platforms. The SAML protocol includes the use of security assertions in order to grant access and determine the level of access. A security assertion is obtained via the interaction of a principal (the user), an identity provider (the system that issues the assertion) and a service provider (the system that accepts the assertion). These security assertions contain certificates and keys. These certificates and keys allow for identity verification and subsequent authorizations.

An efficient way to achieve seamless connectivity with cloud services is to implement federation technologies. Federation technologies use the aforementioned protocols in conjunction with inside- or outside-the-perimeter identity access management directory services in order to allow cross-environment access. The following is an example of ADFS (Active Directory Federation Service) authentication/authorization flow.



Source: Sygnia Advisory – Detection of Golden SAML attacks

The previous graphic could also apply to other cloud service providers besides Azure, as ADFS allows federation with AWS.

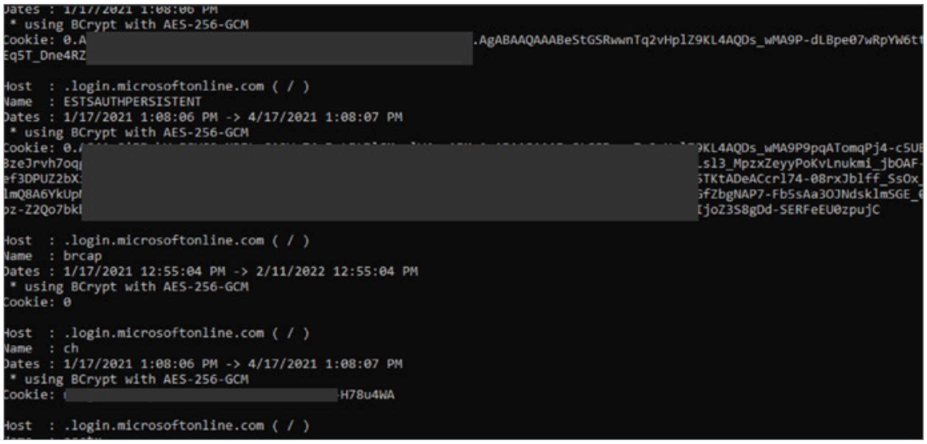
Recent reported attacks such as the SolarWinds campaign indicate that attackers are targeting SAML security assertions and OAuth2 tokens, especially where victims have cloud-connected environments.

Attack flow

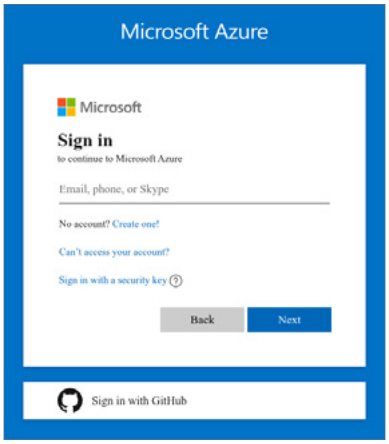
Based on a recent CISA alert, an attack that targets credentials of a cloud-linked or connected perimeter basically seeks to find one or more of these three items:

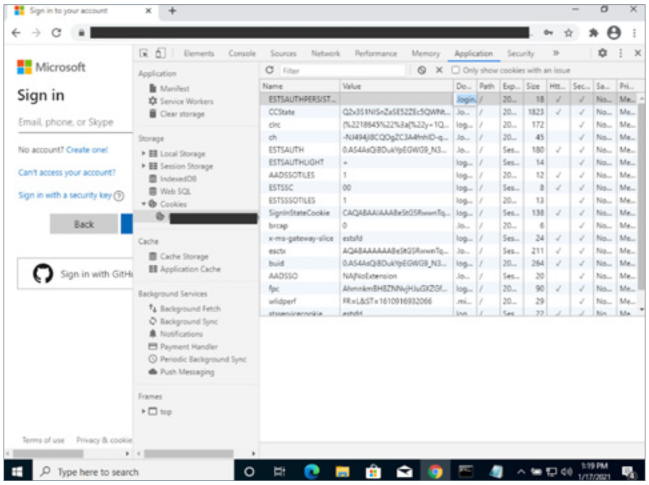
1. **OAuth2 token**

   The Splunk Threat Research team has previously researched GCP OAuth token hijack and reuse. Another example of OAuth2 token reuse can be executed against an Azure environment via pass-the-cookie. This attack bypasses multi-factor authentication as well. The following graphics show an example of a pass-the-cookie attack, after stealing the cookie via the Mimikatz tool.

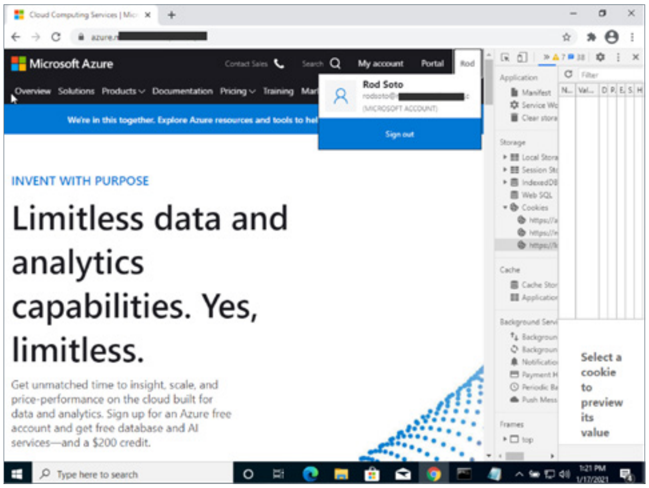


```
Dates : 1/17/2021 1:08:06 PM
 * using BCrypt with AES-256-GCM
Cookie: 0.A[redacted].AgABAAQAAABeStGSRwwnTq2vHplZ9KL4AQDs_wMA9P-dLBpe07wRpYW6t
Eq5T_Dne4RZ

Host  : .login.microsoftonline.com ( / )
Name  : ESTSAUTHPERSISTENT
Dates : 1/17/2021 1:08:06 PM -> 4/17/2021 1:08:07 PM
 * using BCrypt with AES-256-GCM
Cookie: 0.[redacted]9KL4AQDs_wMA9P9pqATomqPj4-c5UE
BzeJrvh7oq                                                  .sl3_MpzxZeyyPoKvLnukmi_jbOAF-
ef3DPUZ2bX                                                  5TKtADeACcrl74-08rxJblff_SsOx_
lmQ8A6YkUp                                                  GfZbgNAP7-Fb5sAa3OJNdsklmSGE_
pz-Z2Qo7bk                                                  IjoZ3S8gDd-SERFeEU0zpujC

Host  : .login.microsoftonline.com ( / )
Name  : brcap
Dates : 1/17/2021 12:55:04 PM -> 2/11/2022 12:55:04 PM
 * using BCrypt with AES-256-GCM
Cookie: 0

Host  : .login.microsoftonline.com ( / )
Name  : ch
Dates : 1/17/2021 1:08:06 PM -> 4/17/2021 1:08:07 PM
 * using BCrypt with AES-256-GCM
Cookie: [redacted]H78u4WA

Host  : .login.microsoftonline.com ( / )
```





Name	Value	Do...	Path	Exp...	Size	Htt...	Sec...	Sa...	Pri...
ESTSAUTHPERSIST...		login.	/	20...	18	✓	✓	No...	Me...
CCState	Q2x3S1NISnZaSE52ZEc5QWNt...	.lo...	/	20...	1823	✓	✓	No...	Me...
clrc	{%2218645%22%3a[%22y+1Q...	log...	/	20...	172		✓	No...	Me...
ch	-NJ494jl8CQOgZC3A4fnhID-q...	.lo...	/	20...	45		✓	No...	Me...
ESTSAUTH	0.AS4AsQiBDukYpEGWG9_N3...	.lo...	/	Ses...	180	✓	✓	No...	Me...
ESTSAUTHLIGHT	+	log...	/	Ses...	14		✓	No...	Me...
AADSSOTILES	1	log...	/	20...	12	✓	✓	No...	Me...
ESTSSC	00	log...	/	Ses...	8	✓	✓	No...	Me...
ESTSSSOTILES	1	log...	/	20...	13		✓	No...	Me...
SignInStateCookie	CAQABAAIAAABeStGSRwwnTq...	log...	/	Ses...	138	✓	✓	No...	Me...
brcap	0	.lo...	/	20...	6		✓	No...	Me...
x-ms-gateway-slice	estsfd	log...	/	Ses...	24	✓	✓	No...	Me...
esctx	AQABAAAAAAABeStGSRwwnTq...	.lo...	/	Ses...	211	✓	✓	No...	Me...
buid	0.AS4AsQiBDukYpEGWG9_N3...	log...	/	20...	264	✓	✓	No...	Me...
AADSSO	NA|NoExtension	.lo...	/	Ses...	20		✓	No...	Me...
fpc	AhmnkmBH8ZNNvjHJuGXZGf...	log...	/	20...	90	✓	✓	No...	Me...
wlidperf	FR=L&ST=1610916932066	.mi...	/	20...	29		✓	No...	Me...
stsservicecookie	estsfd	log...	/	Ses...	22	✓	✓	No...	Me...



We're in this together. Explore Azure resources and tools to hel

INVENT WITH PURPOSE

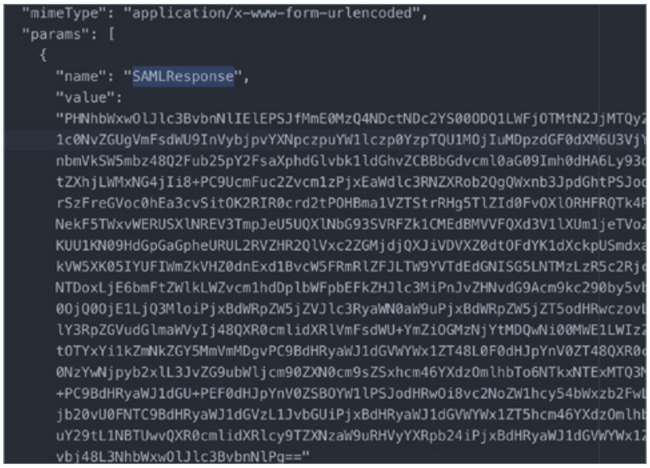
Limitless data and analytics capabilities. Yes, limitless.

Get unmatched time to insight, scale, and price-performance on the cloud built for data and analytics. Sign up for an Azure free account and get free database and AI services—and a $200 credit.

2. **SAML assertion**

   As stated above, if an attacker is able to obtain a valid SAML assertion, they can impersonate the victim and access the victim's cloud environment. This attack has a very low likelihood of success, as some providers like AWS expire assertions after five minutes, along with other SAML signature checks. This type of attack would require extra steps to keep the assertion valid; for example, the attacker would have to change the expiration date or modify some of the attributes (i.e., email address, name, etc.). Strict assertion verification measures prevent these types of attacks.

   A SAML assertion can be extracted by opening the developer's feature in most browsers.



```
"mimeType": "application/x-www-form-urlencoded",
"params": [
  {
    "name": "SAMLResponse",
    "value":
    "PHNhbWxwOlJlc3BvbnNlIElEPSJfMmE0MzQ4NDctNDc2YS00ODQ1LWFjOTMtN2JjMTQy
    1c0NvZGUgVmFsdWU9InVybjpvYXNpczpuYW1lczp0YzpTQU1MOjIuMDpzdGF0dXM6U3Vj
    nbmVkSW5mbz48Q2Fub25pY2FsaXphdGlvbk1ldGhvZCBBbGdvcml0aG09Imh0dHA6Ly93
    tZXhjLWMxNG4jIi8+PC9UcmFuc2Zvcm1zPjxEaWdlc3RNZXRob2QgQWxnb3JpdGhtPSJo
    rSzFreGVoc0hEa3cvSitOK2RIR0crd2tPOHBma1VZTStrRHg5TlZId0FvOXlORHFRQTk4
    NekF5TWxvWERUSXlNREV3TmpJeU5UQXlNbG93SVRFZk1CMEdBMVVFQXd3V1lXUm1jeTVo
    KUU1KN09HdGpGaGpheURUL2RVZHR2QlVxc2ZGMjdjQXJiVDVXZ0dtOFdYK1dXckpUSmdx
    kVW5XK05IYUFIWmZkVHZ0dnExd1BvcW5FRmRlZFJLTW9YVTdEdGNISG5LNTMzLzR5c2Rj
    NTDoxLjE6bmFtZWlkLWZvcm1hdDplbWFpbEFkZHJlc3MiPnJvZHNvdG9Acm9kc290by5v
    0OjQ0OjE1LjQ3MloiPjxBdWRpZW5jZVJlc3RyaWN0aW9uPjxBdWRpZW5jZT5odHRwczov
    lY3RpZGVudGlmaWVyIj48QXR0cmlidXRlVmFsdWU+YmZiOGMzNjYtMDQwNi00MWE1LWIz
    tOTYxYi1kZmNkZGY5MmVmMDgvPC9BdHRyaWJ1dGVWYWx1ZT48L0F0dHJpYnV0ZT48QXR0
    0NzYwNjpyb2xlL3JvZG9ubWljcm90ZXN0cm9sZSxhcm46YXdzOmlhbTo6NTkxNTExMTQ3
    +PC9BdHRyaWJ1dGU+PEF0dHJpYnV0ZSBOYW1lPSJodHRwOi8vc2NoZW1hcy54bWxzb2Fw
    jb20vU0FNTC9BdHRyaWJ1dGVzL1JvbGUiPjxBdHRyaWJ1dGVWYWx1ZT5hcm46YXdzOmlh
    uY29tL1NBTUwvQXR0cmlidXRlcy9TZXNzaW9uRHVyYXRpb24iPjxBdHRyaWJ1dGVWYWx1
    vbj48L3NhbWxwOlJlc3BvbnNlPg=="
```

3. **Certificate, key and directory service**

   Obtaining certificates from a federation service is the most difficult attack to execute but gives the attacker the most power in reusing or forging SAML-signed assertions. This technique can allow attackers to access cloud federated environments or even set up back doors by creating new federated entities via other cloud providers. Attackers can use post-exploitation tools such as ADFSDump, Mimikatz or even operating system tools like Certutil.exe in order to access certificates or keys, then proceed to use them to forge tokens or assertions using tools such as ADFSpoof or Shimit. An attacker needs credentials from a victim that has cloud federation configured and allows authentication flow from inside the perimeter to the cloud.

It is also important to consider that many inside-the-perimeter environments that do not have a formal federation with the cloud still have daily and persistent access to cloud environments. This is especially true for those who have DevOps environments, in which case some of these attacks are still valid.

Detection challenges and opportunities

These attacks are applicable to environments where there are significant interactions between inside-the-perimeter and cloud services and there is an established, formal federation via technologies such as Windows Active Directory Federation Services.

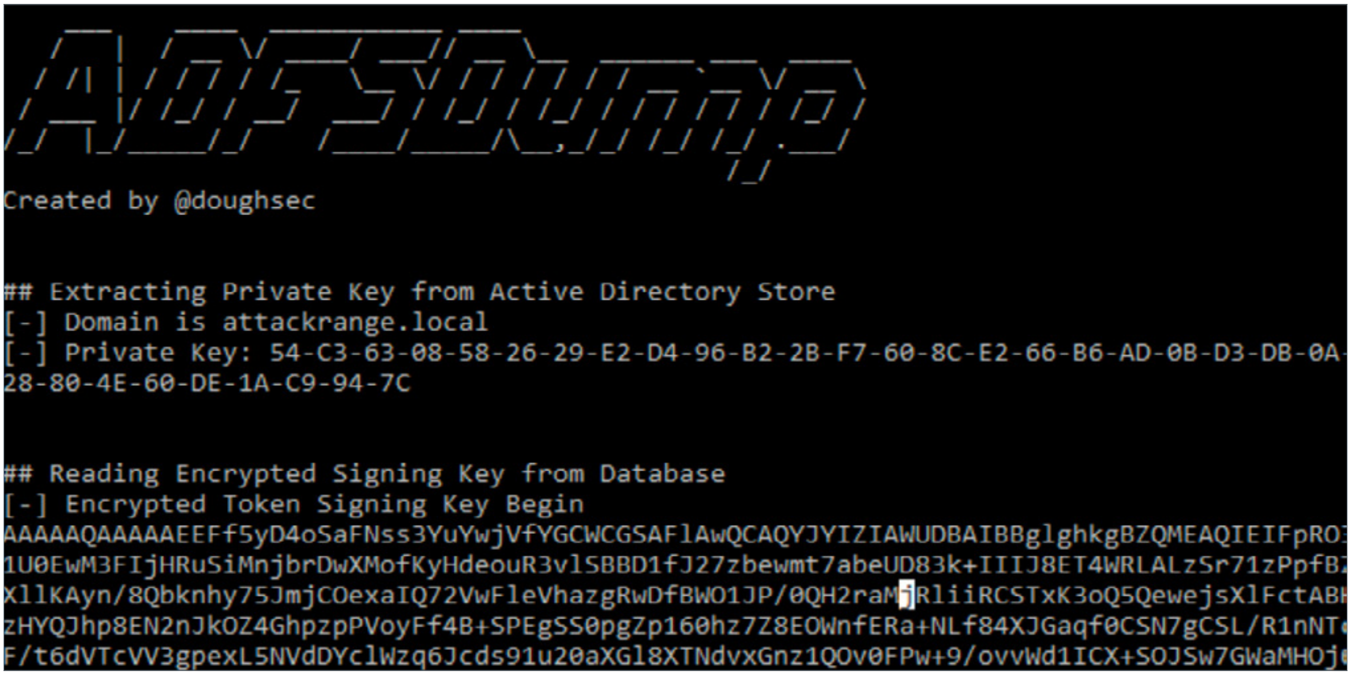
We should also consider informal federations. Those are environments where, even though there are no formal federation technologies in place such as ADFS, there are still many environments where a developer may be inside the perimeter and connected to cloud storage or compute instances. This flow of authentications is linked by their credentials residing at the same endpoint; data flows back and forth between cloud and local developer environment through this endpoint. This is technically an informal federation, and if an attacker can compromise the endpoint, then the attacker can reuse credentials to move north to south or east to west.

The aforementioned federation scenarios are ripe for credential reuse exploitation vectors targeting cloud services.

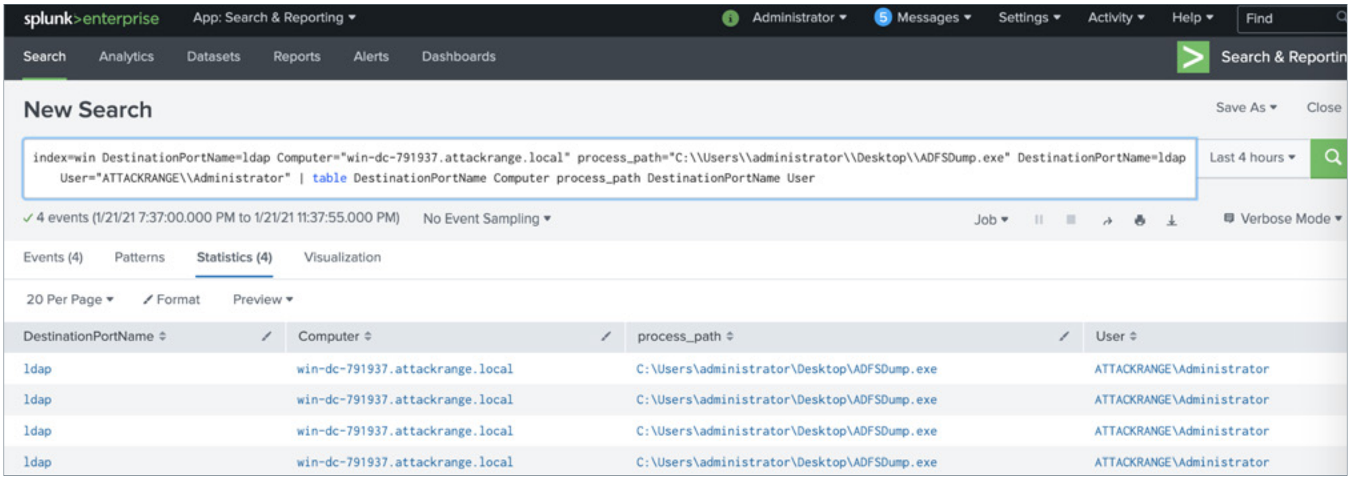
These attacks can be addressed from two different environments:

- Inside the perimeter: This is where the key objects for these types of attacks are located. Here we look at the attack surface of the services that provide identity directory services and federation services.
- The cloud: The infrastructure at the cloud provider is the target of this movement going south to north, since an attacker accesses the credentials first in a perimeter device and then moves onto cloud premises.

Credential reuse attacks specifically are very difficult to detect, because some of the tools that extract credentials from the desktops or servers are not detectable. When looking at the cloud traffic their use generates, it looks exactly like any other access from normal sessions. Take, for example, the use of ADFSDump post-exploitation tool at the desktop server level. This is a tool that dumps information from ADFS services, a prior step an attacker must execute to identify items needed in order to craft forged requests.



```
Created by @doughsec

## Extracting Private Key from Active Directory Store
[-] Domain is attackrange.local
[-] Private Key: 54-C3-63-08-58-26-29-E2-D4-96-B2-2B-F7-60-8C-E2-66-B6-AD-0B-D3-DB-0A-
28-80-4E-60-DE-1A-C9-94-7C


## Reading Encrypted Signing Key from Database
[-] Encrypted Token Signing Key Begin
AAAAAQAAAAAEEFf5yD4oSaFNss3YuYwjVfYGCWCGSAFlAwQCAQYJYIZIAWUDBAIBBglghkgBZQMEAQIEIFpRO
1U0EwM3FIjHRuSiMnjbrDwXMofKyHdeouR3vlSBBD1fJ27zbewmt7abeUD83k+IIIJ8ET4WRLALzSr71zPpfB
XllKAyn/8Qbknhy75JmjCOexaIQ72VwFleVhazgRwDfBWO1JP/0QH2raMjRliiRCSTxK3oQ5QewejsXlFctAB
zHYQJhp8EN2nJkOZ4GhpzpPVoyFf4B+SPEgSS0pgZp160hz7Z8EOWnfERa+NLf84XJGaqf0CSN7gCSL/R1nNT
F/t6dVTcVV3gpexL5NVdDYclWzq6Jcds91u20aXGl8XTNdvxGnz1QOv0FPw+9/ovvWd1ICX+SOJSw7GWaMHOj
```



```
index=win DestinationPortName=ldap Computer="win-dc-791937.attackrange.local" process_path="C:\\Users\\administrator\\Desktop\\ADFSDump.exe" DestinationPortName=ldap
    User="ATTACKRANGE\\Administrator" | table DestinationPortName Computer process_path DestinationPortName User
```

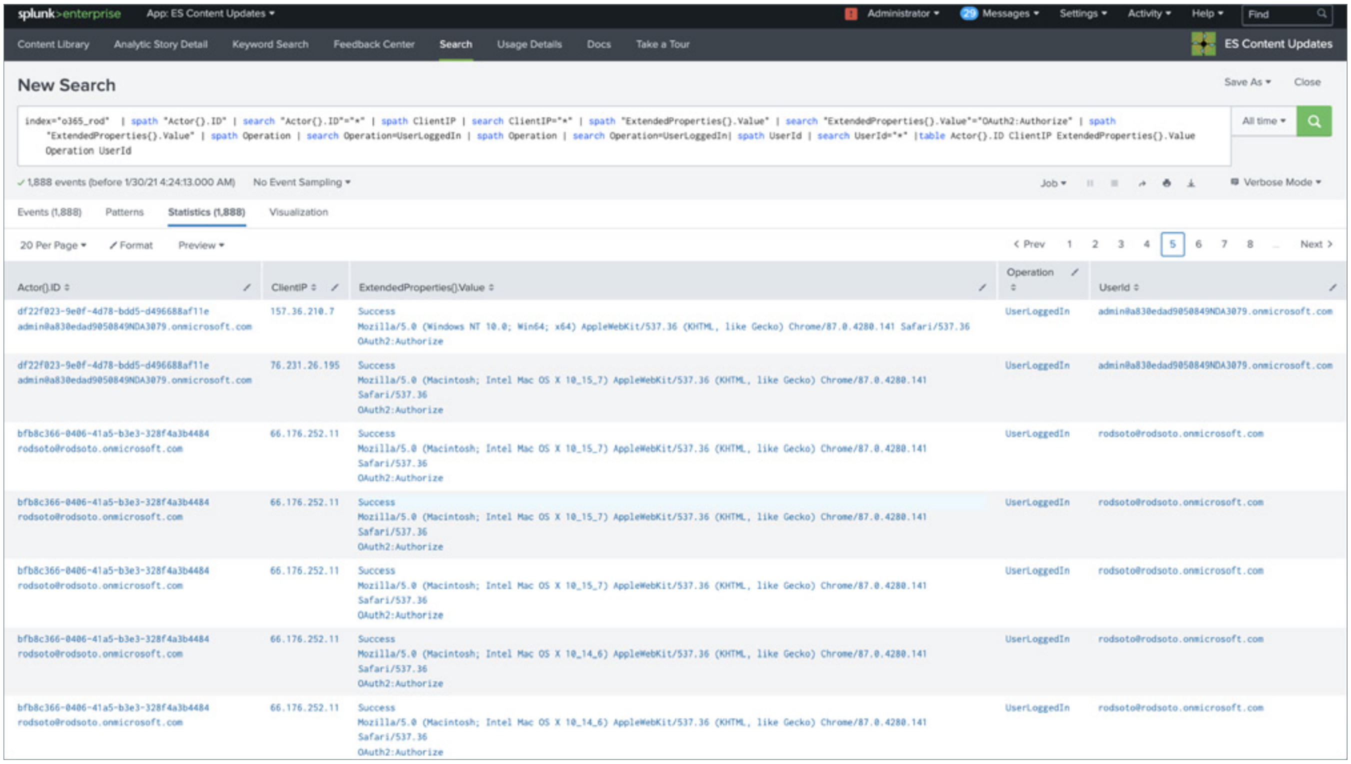
✓ 4 events (1/21/21 7:37:00.000 PM to 1/21/21 11:37:55.000 PM)

DestinationPortName	Computer	process_path	User
ldap	win-dc-791937.attackrange.local	C:\Users\administrator\Desktop\ADFSDump.exe	ATTACKRANGE\Administrator
ldap	win-dc-791937.attackrange.local	C:\Users\administrator\Desktop\ADFSDump.exe	ATTACKRANGE\Administrator
ldap	win-dc-791937.attackrange.local	C:\Users\administrator\Desktop\ADFSDump.exe	ATTACKRANGE\Administrator
ldap	win-dc-791937.attackrange.local	C:\Users\administrator\Desktop\ADFSDump.exe	ATTACKRANGE\Administrator

As seen in the above graphic, we executed the tool against a single instance of an ADFS server created using Splunk Attack Range. Antiviruses likely will not detect this tool. This is in part because it does not have a consistent signature, since it can be compiled using different variables.

In this specific case we compiled ADFSDump, then we ran it against a single ADFS server Windows Server 2016 using WID. We observed that instead of making a connection to SQL port, it used LDAP. This specific case differs from current expected indicators which suggest looking at a SQL pipe for discovery. In this case, using such an indicator for detection will not work.

From the cloud perspective, we executed the pass-the-cookie attack as outlined above, recorded it, then crafted a search to attempt detection. What we found is that the access footprint is exactly the same as normal logons.



```
index="o365_rod"  | spath "Actor{}.ID" | search "Actor{}.ID"="*" | spath ClientIP | search ClientIP="*" | spath "ExtendedProperties{}.Value" | search "ExtendedProperties{}.Value"="OAuth2:Authorize" | spath
    "ExtendedProperties{}.Value" | spath Operation | search Operation=UserLoggedIn | spath Operation | search Operation=UserLoggedIn| spath UserId | search UserId="*" |table Actor{}.ID ClientIP ExtendedProperties{}.Value
    Operation UserId
```

✓ 1,888 events (before 1/30/21 4:24:13.000 AM) No Event Sampling ▾

Actor{}.ID	ClientIP	ExtendedProperties{}.Value	Operation	UserId
df22f023-9e0f-4d78-bdd5-d496688af11e admin@a830edad9050849NDA3079.onmicrosoft.com	157.36.210.7	Success Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.141 Safari/537.36 OAuth2:Authorize	UserLoggedIn	admin@a830edad9050849NDA3079.onmicrosoft.com
df22f023-9e0f-4d78-bdd5-d496688af11e admin@a830edad9050849NDA3079.onmicrosoft.com	76.231.26.195	Success Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.141 Safari/537.36 OAuth2:Authorize	UserLoggedIn	admin@a830edad9050849NDA3079.onmicrosoft.com
bfb8c366-0406-41a5-b3e3-328f4a3b4484 rodsoto@rodsoto.onmicrosoft.com	66.176.252.11	Success Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.141 Safari/537.36 OAuth2:Authorize	UserLoggedIn	rodsoto@rodsoto.onmicrosoft.com
bfb8c366-0406-41a5-b3e3-328f4a3b4484 rodsoto@rodsoto.onmicrosoft.com	66.176.252.11	Success Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.141 Safari/537.36 OAuth2:Authorize	UserLoggedIn	rodsoto@rodsoto.onmicrosoft.com
bfb8c366-0406-41a5-b3e3-328f4a3b4484 rodsoto@rodsoto.onmicrosoft.com	66.176.252.11	Success Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.141 Safari/537.36 OAuth2:Authorize	UserLoggedIn	rodsoto@rodsoto.onmicrosoft.com
bfb8c366-0406-41a5-b3e3-328f4a3b4484 rodsoto@rodsoto.onmicrosoft.com	66.176.252.11	Success Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.141 Safari/537.36 OAuth2:Authorize	UserLoggedIn	rodsoto@rodsoto.onmicrosoft.com
bfb8c366-0406-41a5-b3e3-328f4a3b4484 rodsoto@rodsoto.onmicrosoft.com	66.176.252.11	Success Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.141 Safari/537.36 OAuth2:Authorize	UserLoggedIn	rodsoto@rodsoto.onmicrosoft.com

Because federation tokens are meant to be a feature that provides seamless access to cloud environments, they are not considered vulnerabilities. However, based on the TTPs of these attacks, we have developed an analytical story that covers the above two items (cloud, perimeter) where attackers access the credentials and then pass them to cloud environments. We looked specifically at scenarios such as the Golden SAML and other scenarios of credential abuse using OAuth tokens at the cloud level. On the perimeter level we focused on Windows Privilege Escalation (necessary in most cases to access credentials) and the use of tools such as Mimikatz and ADFSDump.

Perimeter-focused detection searches

Name	Technique ID	Tactic	Note
Certutil.exe certificate extraction	T1552.004	Credential access	New detection
Uncommon Processes on endpoint	T1204.002	Execution	Helps detect ADFSDump
Registry keys used for privilege escalation	T1546.012	Privilege escalation, persistence	
Detect Mimikatz using loaded images	T1003.001	Credential access	
Detect Mimikatz via PowerShell and event code 4703	T1003.001	Credential access	

New cloud-focused hunting and detection searches

Name	Technique ID	Tactic	Provider
AWS SAML access by provider user and principal	T1078	Defense evasion, persistence, privilege escalation, initial access	AWS
AWS SAML update identity provider	T1078	Defense evasion, persistence, privilege escalation, initial access	AWS
O365 Excessive SSO logon errors	T1556	Credential access, defense evasion	Azure
O365 added service principal	T1136.003	Persistence	Azure
O365 added service principal	T1136.003	Persistence	Azure
O365 new federated domain added	T1136.003	Persistence	Azure

Attacks like the Golden SAML are difficult to detect. However, by correlating both cloud provider and perimeter events, analysts are able to get meaningful information for detection. Without correlation, the cloud logins look like any other cloud login and the endpoint attack events do not reveal cloud federated abuse by themselves.

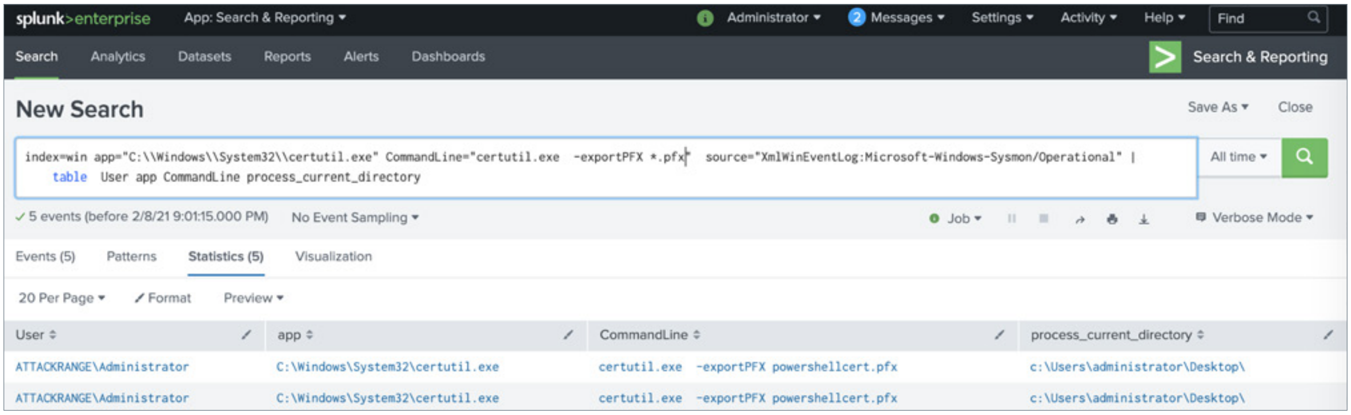
Here are some previews from our upcoming analytical story **Cloud Federated Credential Abuse** in the v3.15.0 release. These searches target OAuth2 token and SAML assertion abuse. These searches must be run under the context of investigation. If run individually, they cannot provide a full picture of a possible federation abuse considering how difficult it is to detect these attacks due to the abnormal-looking authentication flow.

Endpoint-focused

Certutil.exe certificate extraction

This search looks for arguments to certutil.exe indicating certificate manipulation or extraction. Attackers can then use this certificate to sign new authentication tokens, especially inside federated environments such as Windows ADFS.

```
index=win app="C:\\Windows\\System32\\certutil.exe" CommandLine="certutil.exe  -exportPFX *.pfx"
source="XmlWinEventLog:Microsoft-Windows-Sysmon/Operational" | table  User app CommandLine
process_current_directory
```



```
index=win app="C:\\Windows\\System32\\certutil.exe" CommandLine="certutil.exe  -exportPFX *.pfx"  source="XmlWinEventLog:Microsoft-Windows-Sysmon/Operational" |
    table  User app CommandLine process_current_directory
```

✓ 5 events (before 2/8/21 9:01:15.000 PM)

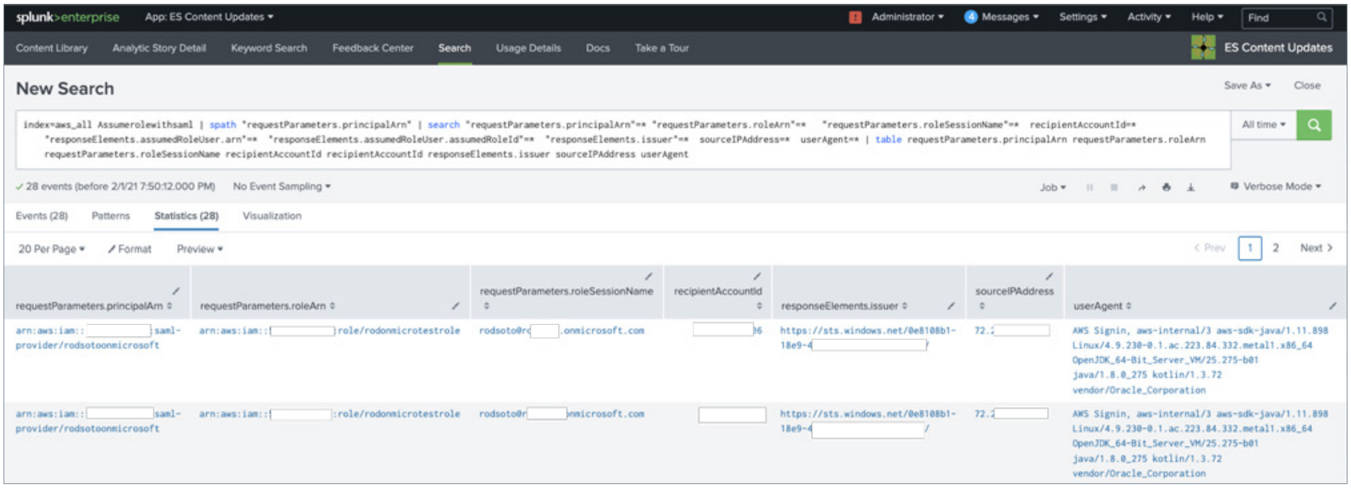
User	app	CommandLine	process_current_directory
ATTACKRANGE\Administrator	C:\Windows\System32\certutil.exe	certutil.exe -exportPFX powershellcert.pfx	c:\Users\administrator\Desktop\
ATTACKRANGE\Administrator	C:\Windows\System32\certutil.exe	certutil.exe -exportPFX powershellcert.pfx	c:\Users\administrator\Desktop\

Cloud-focused

AWS

AWS SAML access by provider user and principal

This search provides SAML access from specific service providers, users and targeted principals at AWS. It also gives certain information to detect abnormal access or potential credential hijack or forgery, especially in federated environments using SAML protocol inside the perimeter or cloud provider.



```
index=aws_all Assumerolewithsaml | spath "requestParameters.principalArn" | search "requestParameters.principalArn"=* "requestParameters.roleArn"=*   "requestParameters.roleSessionName"=*  recipientAccountId=*
    "responseElements.assumedRoleUser.arn"=*  "responseElements.assumedRoleUser.assumedRoleId"=*  "responseElements.issuer"=*  sourceIPAddress=*  userAgent=* | table requestParameters.principalArn requestParameters.roleArn
    requestParameters.roleSessionName recipientAccountId recipientAccountId responseElements.issuer sourceIPAddress userAgent
```

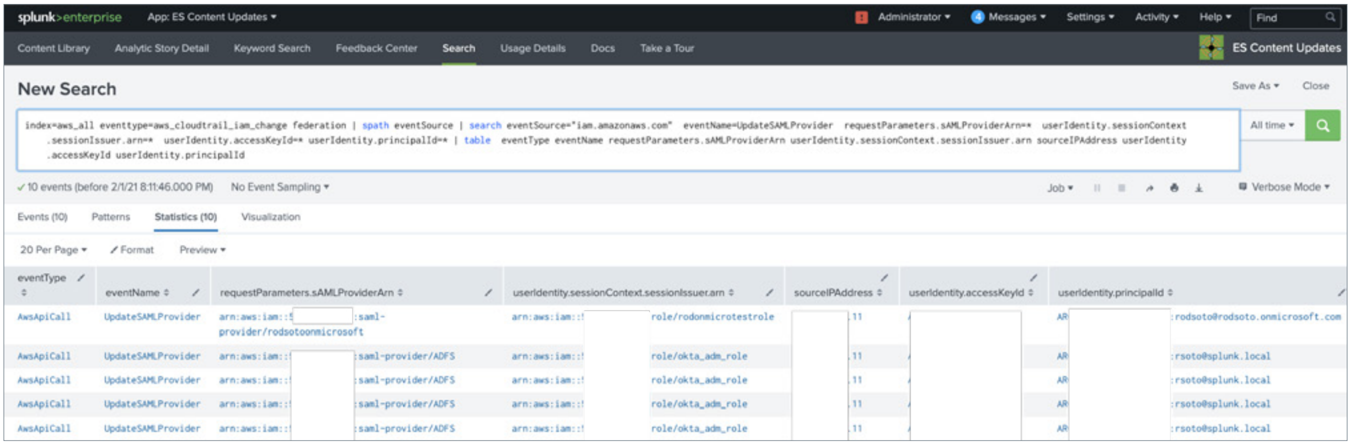
✓ 28 events (before 2/1/21 7:50:12.000 PM)

requestParameters.principalArn	requestParameters.roleArn	requestParameters.roleSessionName	recipientAccountId	responseElements.issuer	sourceIPAddress	userAgent
arn:aws:iam::[redacted]saml-provider/rodsotoonmicrosoft	arn:aws:iam::[redacted]role/rodonmicrotestrole	rodsoto@r[redacted].onmicrosoft.com	[redacted]6	https://sts.windows.net/0e8108b1-18e9-4[redacted]/	72.[redacted]	AWS Signin, aws-internal/3 aws-sdk-java/1.11.898 Linux/4.9.230-0.1.ac.223.84.332.metal1.x86_64 OpenJDK_64-Bit_Server_VM/25.275-b01 java/1.8.0_275 kotlin/1.3.72 vendor/Oracle_Corporation
arn:aws:iam::[redacted]saml-provider/rodsotoonmicrosoft	arn:aws:iam::[redacted]:role/rodonmicrotestrole	rodsoto@r[redacted]nmicrosoft.com	[redacted]	https://sts.windows.net/0e8108b1-18e9-4[redacted]/	72.[redacted]	AWS Signin, aws-internal/3 aws-sdk-java/1.11.898 Linux/4.9.230-0.1.ac.223.84.332.metal1.x86_64 OpenJDK_64-Bit_Server_VM/25.275-b01 java/1.8.0_275 kotlin/1.3.72 vendor/Oracle_Corporation

Note: This search by itself will not detect any SAML-related attack. However, if there are other contextual indicators, this search will provide the elements needed to investigate and pinpoint attack items, such as attributes in SAML assertion, principals, identity, service providers and, of course, the user per request and authentication granted by SAML assertion. This is a hunting query.

AWS update: SAML provider activity

This search provides detection of updates to SAML providers in AWS. Teams should monitor updates to SAML providers closely, as they may indicate possible perimeter compromise of federated credentials or backdoor access from another cloud provider set by the attacker.



New Search

```
index=aws_all eventtype=aws_cloudtrail_iam_change federation | spath eventSource | search eventSource="iam.amazonaws.com"  eventName=UpdateSAMLProvider  requestParameters.sAMLProviderArn=*  userIdentity.sessionContext
    .sessionIssuer.arn=*  userIdentity.accessKeyId=* userIdentity.principalId=* | table  eventType eventName requestParameters.sAMLProviderArn userIdentity.sessionContext.sessionIssuer.arn sourceIPAddress userIdentity
    .accessKeyId userIdentity.principalId
```

✓ 10 events (before 2/1/21 8:11:46.000 PM) No Event Sampling ▾

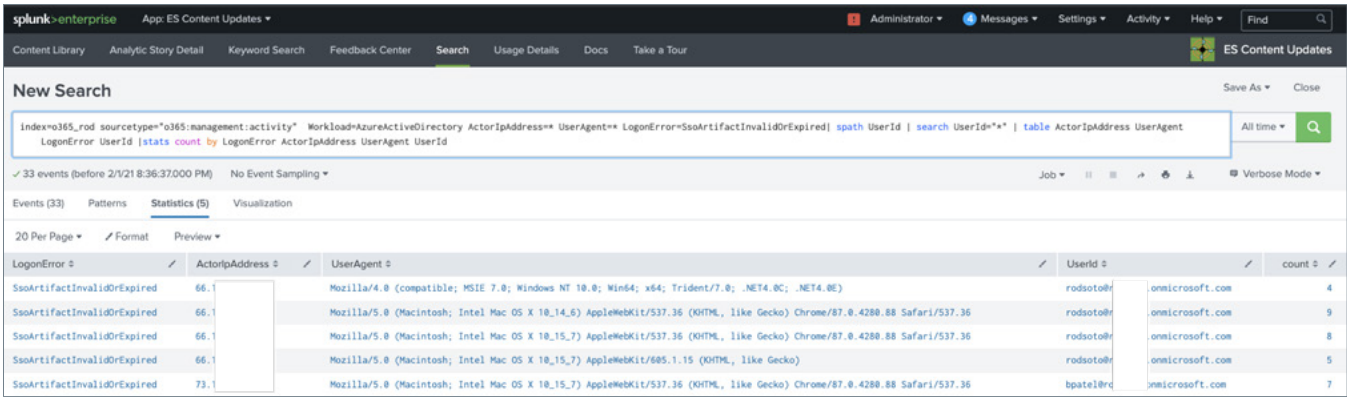
eventType	eventName	requestParameters.sAMLProviderArn	userIdentity.sessionContext.sessionIssuer.arn	sourceIPAddress	userIdentity.accessKeyId	userIdentity.principalId
AwsApiCall	UpdateSAMLProvider	arn:aws:iam::[redacted]:saml-provider/rodsotoonmicrosoft	arn:aws:iam::[redacted]role/rodonmicrotestrole	[redacted].11	[redacted]	AR[redacted]:rodsoto@rodsoto.onmicrosoft.com
AwsApiCall	UpdateSAMLProvider	arn:aws:iam::[redacted]:saml-provider/ADFS	arn:aws:iam::[redacted]role/okta_adm_role	[redacted].11	[redacted]	AR[redacted]:rsoto@splunk.local
AwsApiCall	UpdateSAMLProvider	arn:aws:iam::[redacted]:saml-provider/ADFS	arn:aws:iam::[redacted]role/okta_adm_role	[redacted].11	[redacted]	AR[redacted]:rsoto@splunk.local
AwsApiCall	UpdateSAMLProvider	arn:aws:iam::[redacted]:saml-provider/ADFS	arn:aws:iam::[redacted]role/okta_adm_role	[redacted].11	[redacted]	AR[redacted]:rsoto@splunk.local
AwsApiCall	UpdateSAMLProvider	arn:aws:iam::[redacted]:saml-provider/ADFS	arn:aws:iam::[redacted]role/okta_adm_role	[redacted].11	[redacted]	AR[redacted]:rsoto@splunk.local

Note: This search includes the creation of SAML providers, addition of roles or changes in the IDP document. It also shows federated domain users.

Azure

Excessive SSO logon errors

This search detects accounts with a high number of single sign on logon errors. Excessive logon errors may indicate attempts of a brute-force password attack, reuse or SSO token hijack.



New Search

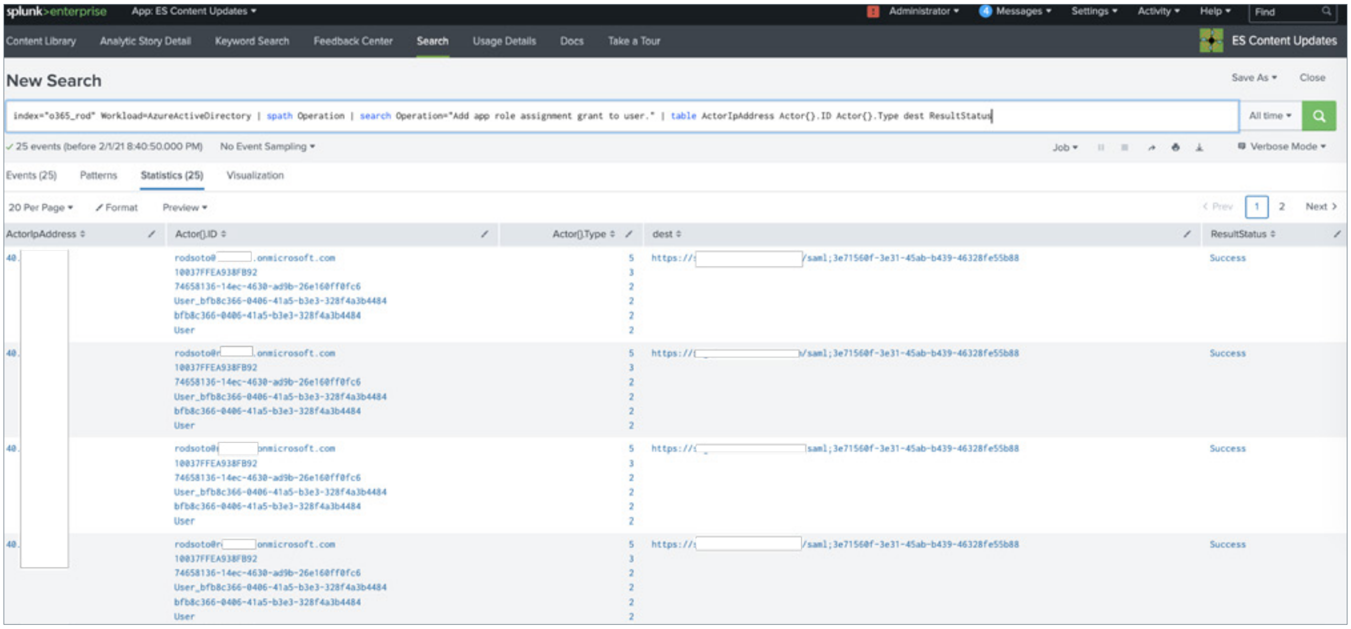
```
index=o365_rod sourcetype="o365:management:activity"  Workload=AzureActiveDirectory ActorIpAddress=* UserAgent=* LogonError=SsoArtifactInvalidOrExpired| spath UserId | search UserId="*" | table ActorIpAddress UserAgent
    LogonError UserId |stats count by LogonError ActorIpAddress UserAgent UserId
```

✓ 33 events (before 2/1/21 8:36:37.000 PM) No Event Sampling ▾

LogonError	ActorIpAddress	UserAgent	UserId	count
SsoArtifactInvalidOrExpired	66.[redacted]	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 10.0; Win64; x64; Trident/7.0; .NET4.0C; .NET4.0E)	rodsoto@[redacted].onmicrosoft.com	4
SsoArtifactInvalidOrExpired	66.[redacted]	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36	rodsoto@[redacted].onmicrosoft.com	9
SsoArtifactInvalidOrExpired	66.[redacted]	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36	rodsoto@[redacted].onmicrosoft.com	8
SsoArtifactInvalidOrExpired	66.[redacted]	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko)	rodsoto@[redacted].onmicrosoft.com	5
SsoArtifactInvalidOrExpired	73.[redacted]	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36	bpatel@[redacted]nmicrosoft.com	7

Add App Role Assignment grant to user

This search detects the creation of a new federation setting by alerting about a specific event related to its creation. In this case, the App Role Assignment is granted to a user, which is a necessary step in Azure to create a new federation.



New Search

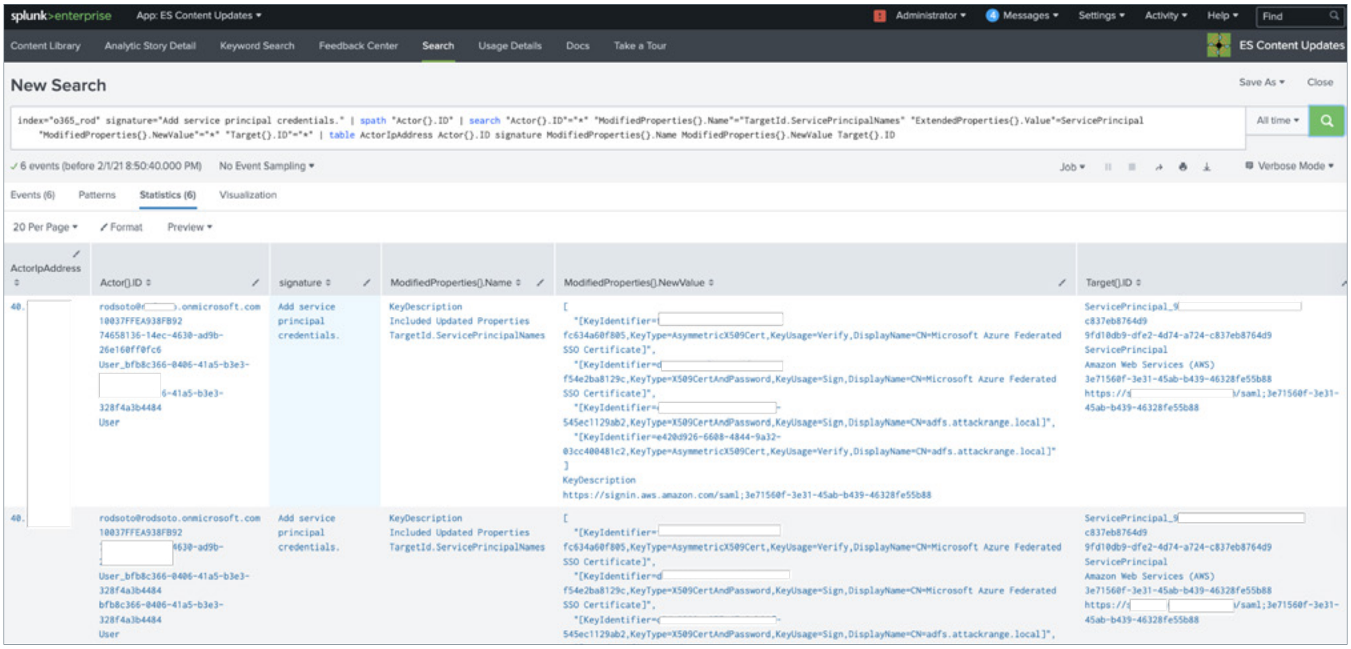
```
index="o365_rod" Workload=AzureActiveDirectory | spath Operation | search Operation="Add app role assignment grant to user." | table ActorIpAddress Actor{}.ID Actor{}.Type dest ResultStatus
```

✓ 25 events (before 2/1/21 8:40:50.000 PM) No Event Sampling ▾

ActorIpAddress	Actor{}.ID	Actor{}.Type	dest	ResultStatus
40.[redacted]	rodsoto@[redacted].onmicrosoft.com 10037FFEA938FB92 74658136-14ec-4630-ad9b-26e160ff0fc6 User_bfb8c366-0406-41a5-b3e3-328f4a3b4484 bfb8c366-0406-41a5-b3e3-328f4a3b4484 User	5 3 2 2 2 2	https://[redacted]/saml;3e71560f-3e31-45ab-b439-46328fe55b88	Success
40.[redacted]	rodsoto@[redacted].onmicrosoft.com 10037FFEA938FB92 74658136-14ec-4630-ad9b-26e160ff0fc6 User_bfb8c366-0406-41a5-b3e3-328f4a3b4484 bfb8c366-0406-41a5-b3e3-328f4a3b4484 User	5 3 2 2 2 2	https://[redacted]/saml;3e71560f-3e31-45ab-b439-46328fe55b88	Success
40.[redacted]	rodsoto@[redacted]onmicrosoft.com 10037FFEA938FB92 74658136-14ec-4630-ad9b-26e160ff0fc6 User_bfb8c366-0406-41a5-b3e3-328f4a3b4484 bfb8c366-0406-41a5-b3e3-328f4a3b4484 User	5 3 2 2 2 2	https://[redacted]saml;3e71560f-3e31-45ab-b439-46328fe55b88	Success
40.[redacted]	rodsoto@[redacted]onmicrosoft.com 10037FFEA938FB92 74658136-14ec-4630-ad9b-26e160ff0fc6 User_bfb8c366-0406-41a5-b3e3-328f4a3b4484 bfb8c366-0406-41a5-b3e3-328f4a3b4484 User	5 3 2 2 2 2	https://[redacted]/saml;3e71560f-3e31-45ab-b439-46328fe55b88	Success

Added service principal

This search detects the creation of a new federation setting by alerting about a specific event related to its creation, in this case the addition of a service principal.



New Search

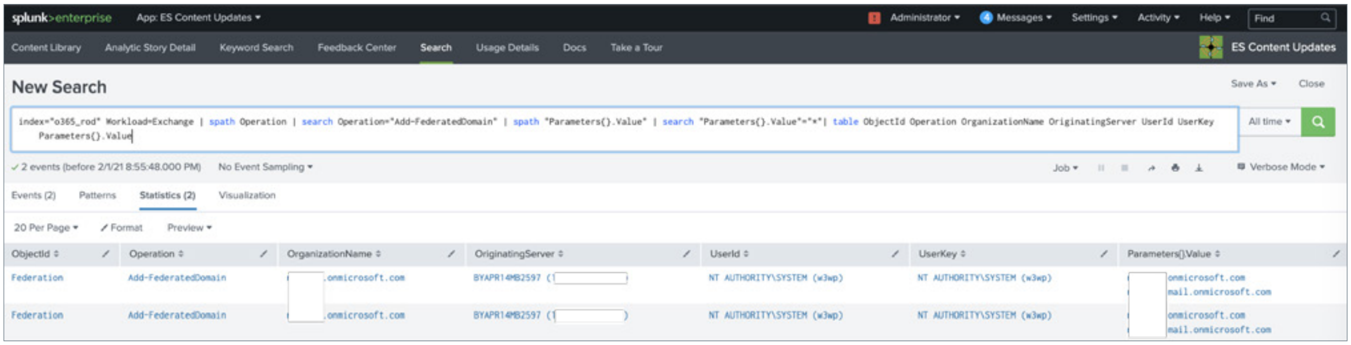
```
index="o365_rod" signature="Add service principal credentials." | spath "Actor{}.ID" | search "Actor{}.ID"="*" "ModifiedProperties{}.Name"="TargetId.ServicePrincipalNames" "ExtendedProperties{}.Value"=ServicePrincipal
    "ModifiedProperties{}.NewValue"="*" "Target{}.ID"="*" | table ActorIpAddress Actor{}.ID signature ModifiedProperties{}.Name ModifiedProperties{}.NewValue Target{}.ID
```

✓ 6 events (before 2/1/21 8:50:40.000 PM) No Event Sampling ▾

ActorIpAddress	Actor{}.ID	signature	ModifiedProperties{}.Name	ModifiedProperties{}.NewValue	Target{}.ID
40.[redacted]	rodsoto@[redacted].onmicrosoft.com 10037FFEA938FB92 74658136-14ec-4630-ad9b-26e160ff0fc6 User_bfb8c366-0406-41a5-b3e3-[redacted]6-41a5-b3e3-328f4a3b4484 User	Add service principal credentials.	KeyDescription Included Updated Properties TargetId.ServicePrincipalNames	["[KeyIdentifier=[redacted]fc634a60f805,KeyType=AsymmetricX509Cert,KeyUsage=Verify,DisplayName=CN=Microsoft Azure Federated SSO Certificate]", "[KeyIdentifier=[redacted]f54e2ba8129c,KeyType=X509CertAndPassword,KeyUsage=Sign,DisplayName=CN=Microsoft Azure Federated SSO Certificate]", "[KeyIdentifier=[redacted]-545ec1129ab2,KeyType=X509CertAndPassword,KeyUsage=Sign,DisplayName=CN=adfs.attackrange.local]", "[KeyIdentifier=e420d926-6608-4844-9a32-03cc400481c2,KeyType=AsymmetricX509Cert,KeyUsage=Verify,DisplayName=CN=adfs.attackrange.local]"] KeyDescription https://signin.aws.amazon.com/saml;3e71560f-3e31-45ab-b439-46328fe55b88	ServicePrincipal_9[redacted] c837eb8764d9 9fd10db9-dfe2-4d74-a724-c837eb8764d9 ServicePrincipal Amazon Web Services (AWS) 3e71560f-3e31-45ab-b439-46328fe55b88 https://[redacted]/saml;3e71560f-3e31-45ab-b439-46328fe55b88
40.	rodsoto@rodsoto.onmicrosoft.com 10037FFEA938FB92 [redacted]4630-ad9b- User_bfb8c366-0406-41a5-b3e3-328f4a3b4484 bfb8c366-0406-41a5-b3e3-328f4a3b4484 User	Add service principal credentials.	KeyDescription Included Updated Properties TargetId.ServicePrincipalNames	["[KeyIdentifier=[redacted]fc634a60f805,KeyType=AsymmetricX509Cert,KeyUsage=Verify,DisplayName=CN=Microsoft Azure Federated SSO Certificate]", "[KeyIdentifier=[redacted]f54e2ba8129c,KeyType=X509CertAndPassword,KeyUsage=Sign,DisplayName=CN=Microsoft Azure Federated SSO Certificate]", "[KeyIdentifier=[redacted]-545ec1129ab2,KeyType=X509CertAndPassword,KeyUsage=Sign,DisplayName=CN=adfs.attackrange.local]",	ServicePrincipal_9[redacted] c837eb8764d9 9fd10db9-dfe2-4d74-a724-c837eb8764d9 ServicePrincipal Amazon Web Services (AWS) 3e71560f-3e31-45ab-b439-46328fe55b88 https://[redacted] [redacted]/saml;3e71560f-3e31-45ab-b439-46328fe55b88

New federated domain added

This search detects the addition of a new federated domain.



```
index="o365_rod" Workload=Exchange | spath Operation | search Operation="Add-FederatedDomain" | spath "Parameters{}.Value" | search "Parameters{}.Value"="*"| table ObjectId Operation OrganizationName OriginatingServer UserId UserKey
    Parameters{}.Value
```

✓ 2 events (before 2/1/21 8:55:48.000 PM) No Event Sampling ▾

ObjectId	Operation	OrganizationName	OriginatingServer	UserId	UserKey	Parameters{}.Value
Federation	Add-FederatedDomain	[redacted].onmicrosoft.com	BYAPR14MB2597 ([redacted]	NT AUTHORITY\SYSTEM (w3wp)	NT AUTHORITY\SYSTEM (w3wp)	[redacted]onmicrosoft.com mail.onmicrosoft.com
Federation	Add-FederatedDomain	[redacted].onmicrosoft.com	BYAPR14MB2597 ([redacted])	NT AUTHORITY\SYSTEM (w3wp)	NT AUTHORITY\SYSTEM (w3wp)	[redacted]onmicrosoft.com mail.onmicrosoft.com

Some of these attack vectors are new and evolving and they seem to emulate past lateral movement techniques such as pass the hash or pass the ticket. Many vendors do not consider these attack vectors as vulnerabilities but rather an abuse of features. These types of attacks are bound to become more popular as enterprises continue to implement cloud services.

All the above searches are available for free today under the Cloud Federated Credential Abuse analytic story via Splunk Security Content and Splunk Security Essentials.

About the Splunk Threat Research Team

The Splunk Threat Research team is devoted to understanding actor behavior and researching known threats to build detections that the entire Splunk community can benefit from. The Splunk Threat Research team does this by building and open sourcing tools that analyze threats and actors like the Splunk Attack Range and using these tools to create attack data sets. From these data sets, new detections are built and shared with the Splunk community under Splunk Security Content. Various Splunk products like Enterprise Security, Splunk Security Essentials and Mission Control then consume these detections to help customers quickly and effectively find known threats.

Want to start using these pre-packaged detections to help your security operation center get started? Download the Enterprise Security Content Updates App on Splunkbase today!

splunk>

Learn more: www.splunk.com/asksales

www.splunk.com

Splunk, Splunk>, Data-to-Everything, D2E and Turn Data Into Doing are trademarks and registered trademarks of Splunk Inc. in the United States and other countries. All other brand names, product names or trademarks belong to their respective owners. © 2021 Splunk Inc. All rights reserved.