Decreasing Attacker Dwell Time in Azure Active Directory

GIAC GMON Gold Certification

Author: mark.morowczynski@students.sans.edu
Advisor: Bryan Simon

Accepted: *April 26th, 2021*

Abstract

As companies continue to embrace the cloud, attackers also have shifted their attack methods to target cloud infrastructure. A popular target in 2020 has been an identity-based compromise (Verizon, 2020). Azure Active Directory is the identity provider behind Office 365, Azure, and thousands of applications for 200,000 companies, processing 30 billion authentications a day (Microsoft Corporation, 2021). Reducing attacker dwell time for any infrastructure is one of the most fundamental ways to minimize a breach's scope and financial impact. This paper provides an investigation into the effects on attacker dwell time when leveraging Microsoft's Security Operation Guide for Azure Active Directory.

1. Introduction

Cloud services adoption is expanding at a rapid rate. The benefits of scalability, flexibility, and in recent times, a forced shift to a fully remote workforce has required even the most critical business functions to migrate to the cloud. With this shift to cloud services, identity plays an even more crucial role and becomes the new perimeter for many organizations. Azure Active Directory (Azure AD), Microsoft’s cloud-based identity as a service (IDaaS), was named a leader in this space by Gartner. By the numbers, Azure AD has 200,000 customers, 425 million monthly active users, and processes 30 billion authentications a day. It is also the identity provider behind Office 365 and Azure (Microsoft Corporation, 2021).

This increased importance of the cloud and identity is not lost on attackers. According to Verizon's 2020 DBIR, cloud assets were involved in 24% of breaches (Verizon, 2020). Of those attacks, identity-based compromises were the primary target. As the report states, “Additionally, 77% of those cloud breaches also involved breached credentials. This is not so much an indictment of cloud security as it is an illustration of the trend of cybercriminals finding the quickest and easiest route to their victims” (Verizon, 2020).

Looking one layer deeper into these credential-based attacks, Microsoft can determine that most identity-based attacks are made up of three different categories: phishing, credential reuse, and password spray (Microsoft Corporation, 2020). Identity-based breaches also are more expensive. The 2020 IBM Cost of Data Breach report by the Ponemon Institute found that compromised credentials increased the total cost of a breach by nearly $1 million (Ponemon Institute, 2020).

Attackers are still unfortunately in the lead against defenders. Attacks are measured in minutes to hours. Dwell time, though improving, is still measured in weeks or months, depending on the report. IBM Cost of Data Breach report by the Ponemon Institute has dwell time at 280 days (Ponemon Institute, 2020). FireEye’s M-Trends 2000 report has dwell time at 56 days (FireEye Mandiat, 2020).

This long dwell time is terrible for the overall cost of breaches. The average breach costs $3.86 million (Ponemon Institute, 2020). The longer they go, the more expensive they become. Detecting and containing a breach in under 200 days can save $1.12 million (Ponemon Institute, 2020). Attacks are going to get through and how fast defenders can detect, and respond is critical (Nemeretes, 2019). Organizations should continue to focus on driving this dwell time down. As part of this effort, they need to ensure cloud identity compromises are part of their detection and monitoring capabilities.

Recently Microsoft released a security operations guide for Azure AD, https://aka.ms/AzureADSecOps. This guidance gives SOCs and overall defenders the recommended activities and alerts they should monitor for and take action on in their Azure AD environments. This paper aims to analyze if the dwell time of common identity attacks would be reduced by following this security operations guidance.

2. Why Azure AD

Azure AD is Microsoft's cloud-based identity as a service (IDaaS) offering. It is a whole Identity and Access Management (IAM) product that is the Identity Provider (IdP) for Microsoft applications such as Office365 and Azure. However, it also leverages modern authentication protocols such as SAML, OAuth, and OpenID Connect to integrate with non-Microsoft applications. In 2020, Gartner named Azure AD a leader in the 2020 Magic Quadrant for Access Management. Azure AD has 200,000 customers, with 425 Million monthly active users, and processes 30 billion authentications a day (Microsoft Corporation, 2021).

There are two types of ways for companies to authenticate to Azure AD: Federated Authentication or Managed Authentication. The authentication method a company chooses will impact the detectability of some of these identity-oriented attacks.

2.1 The Need For Federation

Most customers authenticate to their corporate on-prem Active Directory through their domain-joined computer. On-prem Active Directory is used in 95% of the Fortune 500 companies/organizations (Anderson, 2018). On-prem Active Directory provides them with authentication, usually through a single sign-on experience to local resources such as file servers and applications local to the intranet. The boundary of corporate resources is expanding from the local intranet to the internet with cloud technologies and SaaS applications. As more companies embrace the cloud and SaaS applications, users need a way to authenticate to these resources using their corporate identities, otherwise several problems can occur.

The first problem is that the user needs to have a username/password pair in each application to authenticate. This requires each application to implement its own authentication stack and everything that comes with it, such as password resets and storage of these credentials. There are numerous cases where a vendor does not correctly store these usernames and passwords. Troy Hunt’s website, “Have I been pwned?" is an excellent source for the latest credential compromise information (Hunt, 2021).

To make matters worse, users are likely to reuse their credentials across multiple applications. Since they are using this application for work; therefore, they are likely to reuse their same corporate credentials. Credential reuse presents an even bigger issue as the number of SaaS applications increases. The credentials are spread across all these applications, and a compromise of one can lead to a compromise of all the remaining applications (Microsoft Corporation, 2020).

Second, applications typically need more than just a username to be useful to a user. Information like job title, department, manager, etc., are all leveraged in applications to allow functionality and provide authorization for user actions. What the head of HR can view, add, delete, and change is much different from what an individual employee can do in that HR application. This additional data would also need to be present in each of these SaaS applications.

The good news is the industry saw these problems and started moving to a federated model. At a high-level, federation works on trust between an Identity Provider (IdP) and an application (also called a Resource Provider (RP) in many modern authentication protocols). The IdP handles the user's authentication and then provides some information to the application about who that user is and any additional information about the user that the application needs, usually in the form of a claim that is signed by the IdP (Bertocci, 2016).

2.2 Federation Components

A detailed breakdown of the inner workings of federation protocols such as WS-Fed, SAML, OAuth, and OpenID Connect is well beyond the scope of this paper. However, some components and concepts apply to many modern authentication protocols that are worth understanding.

The Identity Provider- As stated, the identity provider handles the authentication of the user. The authentication can be via a web browser using forms-based authentication or integrated windows authentication (IWA) or a native application using a web API. Common examples of IdPs are Azure AD, Active Directory Federation Services (ADFS), and Ping Federate. The IdP issues trusted claims to the application/resource provider (Bertocci, 2016).

Claims- This is information that is sent to the application/resource provider that, in this case, identifies the user and any additional information about the user that the application needs to function. This varies from application to application. This claim is signed by the IdP using the private key of the IdP (Bertocci, 2016).

Certificates- Public key cryptography is used to digitally sign claims by the IdP using its private key. The application/resource provider uses the public key to validate the claim. The application validates that the claim did come from the IdP and the claims data has not been modified since it was signed (Bertocci, 2016).

Trust and Metadata- Before a user can authenticate and have information sent as a claim to the application, a trust needs to be set up first. The setup details vary between federation protocols, but the IdP and the application will essentially exchange some information, such as the IdP public key and the application's endpoints for authentication (Bertocci, 2016).

Application/Resource provider- This is the application or resource that the user is accessing. Because the trust and exchange of metadata have happened previously, the application will trust the signed claims from the IdP. There are thousands of applications that support this federated authentication model, such as Office365, Salesforce, and Workday.

An example diagram below depicts these components (Figure 1).

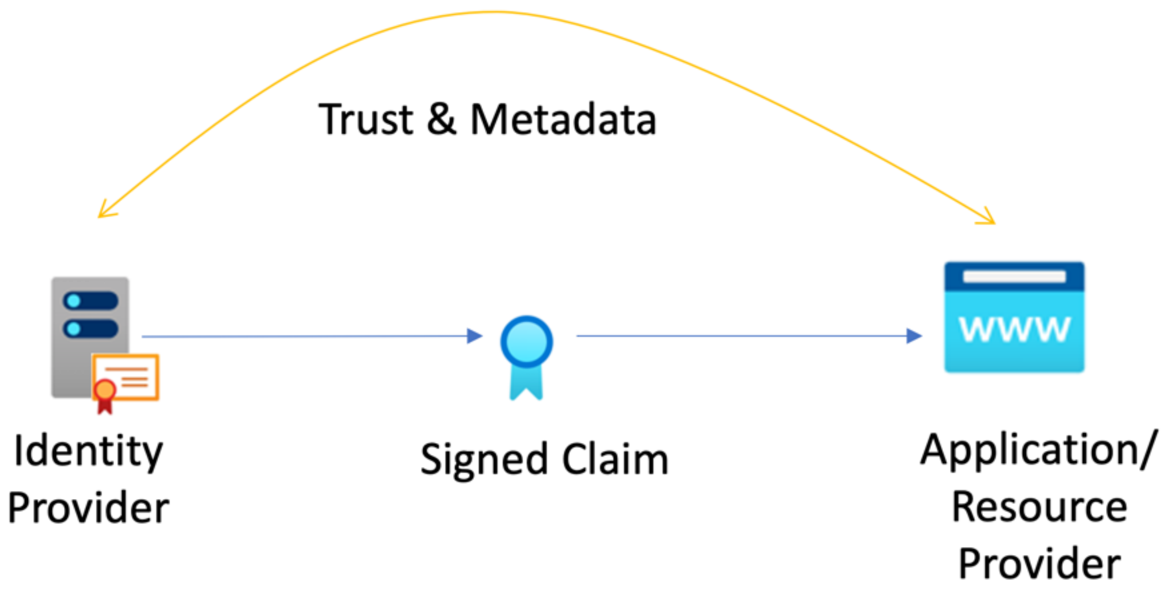


Figure 1. Common federation components

2.3 Federation with Azure AD

There are two aspects to federating with Azure AD. The first aspect is Azure AD is itself an IdP. That means applications can trust claims issued from Azure AD. Office 365 only trusts Azure AD as its IDP. Additional applications can be configured to trust Azure AD as the IDP (Chik, 2021).

The second aspect is when Azure AD is acting as the resource provider for an existing IdP. In this scenario, an existing IdP like ADFS or Ping Federate authenticates the user to their on-prem Active Directory; then the IdP sends the claim up to Azure AD. There are 25 IdPs that have been tested and work with Azure AD (Microsoft, 2021).

When authenticating to Azure AD through a federated model, there are a few things to note. First, the protection of the private keys of the IdP is critical. If the certificates are compromised, an attacker would have the ability to issue a claim that Azure AD would trust and accept the authentication of whoever the user was in the claim. This type of attack was recently seen as part of the Solarwinds incident in December 2020 (Weinert, 2020).

Secondly, depending on configuration, Azure AD will only have authentication logs of the successful logins. Failed logins will be in the IdP. When the user is authenticating to Azure

AD, they are redirected to the IdP. If they fail to login, no claim is sent back to Azure AD. This is important to understand when investigating attacks since logs from the federated IdP will also be needed to see those failed login attempts. If the IdP is ADFS and ADFS Connect Health is installed, then the sign-in logs of both successful and failed attempts from ADFS will be included in the Azure AD sign-in logs (Microsoft, 2021). If ADFS Connect Health is not installed on ADFS or the IdP is not ADFS, then the Azure AD logs will only include successful logins. The failed attempts will be located on the IdP.

2.4 Managed Authentication with Azure AD

The second way organizations authenticate against Azure AD is through managed authentication. There are two choices for managed authentication: Pass-through Authentication (PTA) and Password Hash Sync (PHS).

2.4.1 Pass-through Authentication (PTA)

PTA uses authentication agents on-prem that reach outbound to Azure AD, take the encrypted credentials, and then validate them against on-prem Active Directory. Next, the agent sends a message back up to Azure AD if the authentication is successful or failed (Microsoft, 2021).

When authenticating to Azure AD through PTA, it's highly critical that the servers running the PTA agents are protected and treated like a tier 0 resource. If an attacker could access this resource, they could potentially intercept or modify authentication requests destined for on-prem Active Directory.

2.4.2 Password Hash Sync (PHS)

The current credentials in on-prem Active Directory are synced up to Azure AD through Azure AD Connect. This is the MD4 hash in on-prem Active Directory that is processed with per-user salt and inputted into the PBKDF2 function for 1,000 iterations of HMAC-SHA256. The resulting hash, the per-user salt, along with the number of iterations, are transmitted to Azure AD. The user then authenticates to Azure AD directly with the same password they use for their corporate credentials, where it is put through the same MD4 hash, per-user salt, PBKDF2, HMAC-SHA256 process. If the hash matches the hash stored in Azure AD, the user is authenticated (Microsoft, 2021).

2.5 Lab Setup

Three different lab environments based on these three different authentication method types with Azure AD will be set up to simulate three possible different corporate environments.

2.5.1 Federation

This first part of the lab (Figure 2) will be set up to mirror a real customer environment using federation:

- An on-prem Active Directory forest (corp.markmorow.com) on Server 2019 (Version 1809 OS Build 17763.1757).
- Azure AD Connect (version 1.5.45.0, the most recent at this time of writing).
- ADFS and Web Application Proxy (WAP) servers are also running the same OS version as the domain controllers.
- One Windows 10 (20H2 Build 19042.804) joined to the domain.

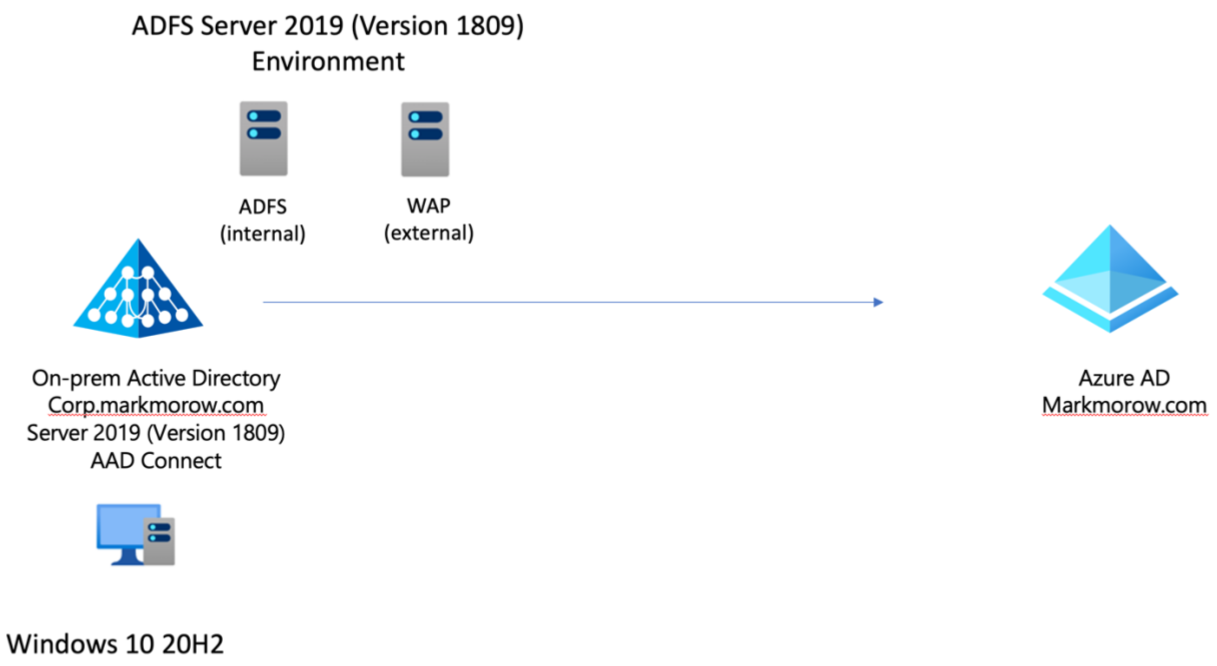


Figure 2. Federation lab setup

2.5.2 Pass Through Authentication (PTA)

The second part of the lab (Figure 3) will be set up to mirror a real customer environment using PTA:

- An on-prem Active Directory forest (corp.markmorowpta.com) on Server 2019 (Version 1809 OS Build 17763.1757).
- Azure AD Connect (version 1.5.45.0, the most recent at this time of writing) with the PTA agent.
- One Windows 10 (20H2 Build 19042.804) joined to the domain.

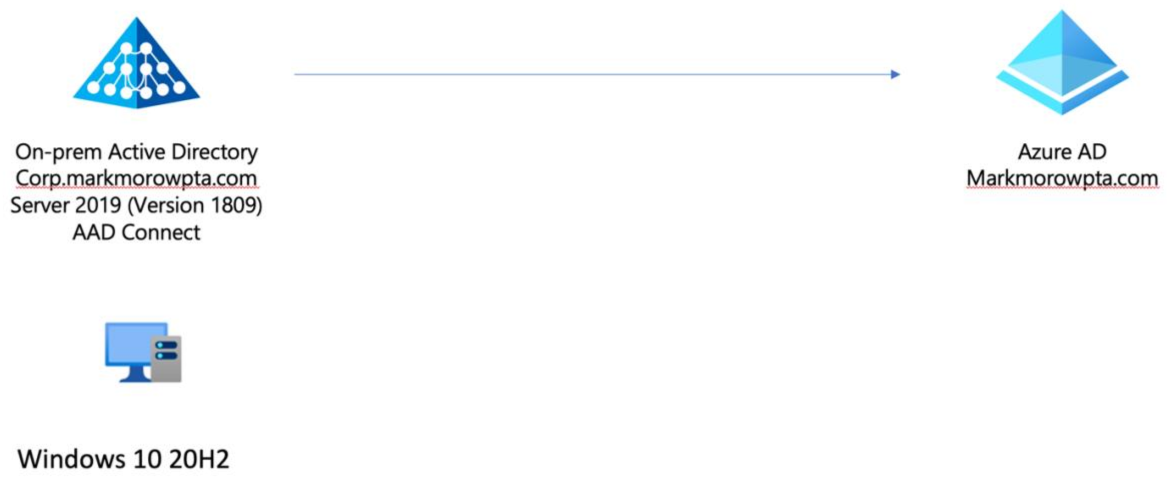


Figure 3. PTA lab setup

2.5.3 Password Hash Sync (PHS)

The final part of the lab (Figure 4) will be set up to mirror a real customer environment using PHS:

- An on-prem Active Directory forest (corp.markmorowphs.com) on Server 2019 (Version 1809 OS Build 17763.1757).
- Azure AD Connect (version 1.5.45.0, the most recent at this time of writing).
- One Windows 10 (20H2 Build 19042.804) joined to the domain.

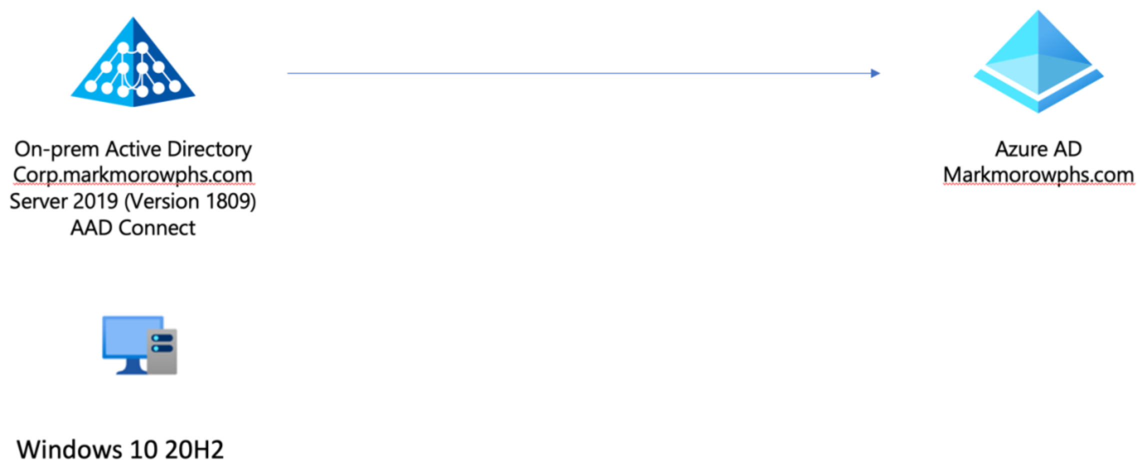


Figure 4. PHS lab setup

3. Cloud Identity Attacks to Test Dwell Time

Organizations have limited time and a limited budget. Focusing on the most common identity attacks is critical to ensure detection efforts reduce dwell time. According to Microsoft, the most common identity attacks are credential reuse, password spray, and phishing (Microsoft Corporation, 2020). Another emerging identity-based attack is application consent phishing (Gupta, 2018). Application consent phishing will also be performed in the lab to determine if the dwell time can be reduced following the security operations guide.

3.1 Credential Reuse

A user will use the same username and password at many different sites and applications. Frequently it's the same password they use for their corporate credentials. When one of these sites or applications is compromised, the attacker will try those same credentials against many other resources, including Azure AD (Hunt, 2021).

3.2 Password Spray

Users follow very predictable password patterns. Many times, this is caused by corporate password policies. Requiring users to change their password every 30 days will frequently have users selecting the month, the year, then the special character— "May2021!" for example, or a season such as "Summer2021!" if passwords are changed quarterly. Attackers then try this same password against all the users in the directory, frequently leveraging legacy protocols that cannot do MFA, such as IMAP4, POP3, or SMTP (Microsoft Corporation, 2020).

3.3 Phishing

The identity attack that people are most familiar with is phishing. This is when an attacker tries to impersonate a legitimate service and tries to get the user to enter some sort of personal information. In the context of identity attacks, this personal information is usually their username and passwords. They then use these credentials against the service to impersonate the user. In some of the more advanced cases, the attacker will also try to get the user to input their MFA prompt as well (Microsoft Corporation, 2020).

3.4 Application Consent Phishing

An emerging attack for identities is application consent phishing. Similar to other phishing attempts, the attacker tries to get the user to consent to a malicious application masquerading as an application the user would use. In the act of consenting to the application, the malicious application would, depending on the permissions it's asking for, be able to access resources on behalf of that user. For example, users can read the directory. If an application asked a user to consent to use an application and the permissions that the application was requesting were also to read the directory, it would then be able to read the directory on behalf of the signed-in user (Gupta, 2018).

Though this attack is not as prevalent as the other three, it will become more popular in the future as the previous three become less frequent. This attack is already being seen in the wild. SANS themselves were hit with this type of attack on August 6th, 2020 (SANS, 2021). Thus, this attack will be included in our evaluation which will include a sample application for others to test this scenario in their environments.

4. Simulating The Attack

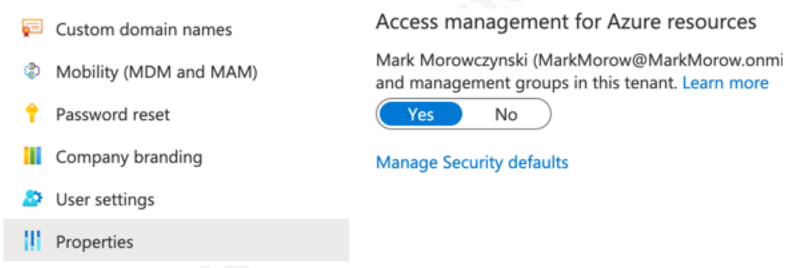
Determining a definitive answer for the average attacker dwell time is difficult as there is no industry consensus on this time amount. For this research, the FireEyes MTrends average dwell time finding for an attack will be utilized which is 56 days. The Azure AD security operations guide also does not set a specific response time for their high, medium, and low alerts. These actions should be taken in alignment with the SOCs SLA for those levels of severity. For this analysis, a high alert SLA of 24 hours, a medium alert SLA of 72 hours, and a low alert at 120 hours (one day, three days, and five days) will be used. An analyst's SOC response might be shorter or longer based on staffing, resourcing, and business needs.

Azure AD also has built-in risk events to alert to many of these common identity attacks. They use a high, medium, and low-risk scale to indicate priority and how confident the system is in the risk event itself (Microsoft, 2021).

4.1 Azure AD Security Defaults

When setting up a new Azure AD tenant, the feature Security Defaults is enabled by default. This feature is designed to help mitigate some of these most common identity attacks like password spray by having legacy authentication disabled. Security Defaults also requires users to register for MFA and to complete an MFA challenge when necessary which helps mitigate when credentials are compromised. There is no configuration to change or update. Security Defaults is ideal for small and medium customers who do not have a dedicated security or identity team. More fine-grain control is usually needed for larger enterprises and provided by other Azure AD areas, such as Conditional Access. For these tests, Security Defaults have been disabled to simulate the larger enterprise (Microsoft, 2021).

To disable Security Defaults, the administrator should login to portal.azure.com and into their Azure AD tenant. They would then click on “Properties” and then “Mange Security Defaults” in blue (Figure 5). Another section will then be displayed (Figure 6).



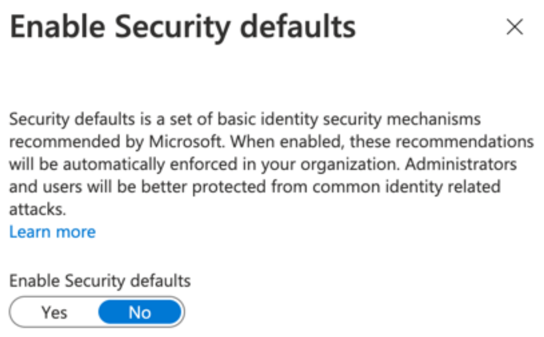
Access management for Azure resources

Mark Morowczynski (MarkMorow@MarkMorow.onmi
and management groups in this tenant. Learn more

[✓ Yes] No

Figure 5. Azure AD tenant properties

The administrator should select “No” on the switch “Enable Security defaults”, then click “Save.”



Enable Security defaults

Security defaults is a set of basic identity security mechanisms recommended by Microsoft. When enabled, these recommendations will be automatically enforced in your organization. Administrators and users will be better protected from common identity related attacks.
Learn more

Enable Security defaults

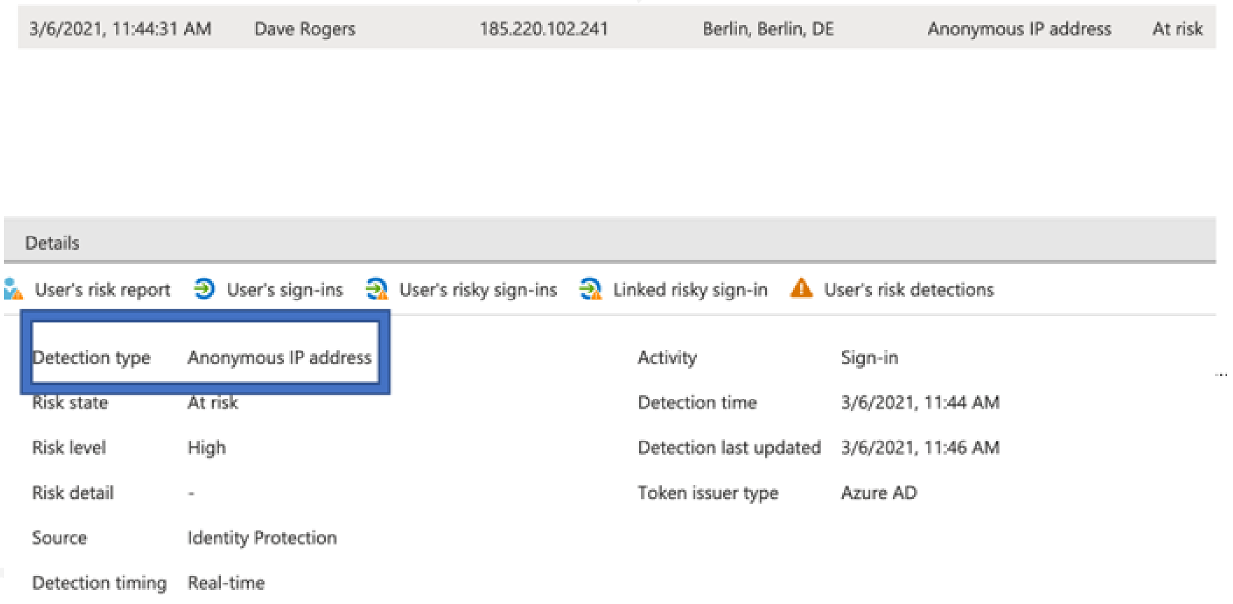
Yes [✓ No]

Figure 6. Disabling Security defaults

4.2 Credential Reuse Attack

Azure AD alerts to this attack under the signal of a "Leaked Credential." Azure AD processes these credentials whenever they are discovered. These credentials come from underground sites such as Pastebin as well as from security researchers and law enforcement. When a leaked credential is detected, the user automatically goes to a risk state of high (Microsoft, 2021).

To simulate that same outcome, an Azure AD user logs in leveraging a TOR browser. This will generate an alert that the user has signed-in as an “Anonymous IP Address." To view this alert in the Azure AD portal, click on “Security”, then “Risky sign-ins” (Figure 7). Then locate this sign-in and click on "Confirm sign-in compromised" (Figure 8). This will move the user to a state of high risk to simulate a leaked credential found. The behavior was seen consistent across federated, PTA, and PHS authentication.

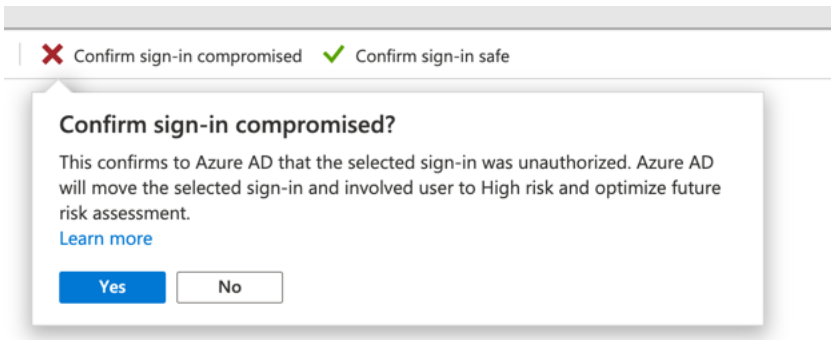


3/6/2021, 11:44:31 AM	Dave Rogers	185.220.102.241	Berlin, Berlin, DE	Anonymous IP address	At risk

Details

Detection type	Anonymous IP address	Activity	Sign-in
Risk state	At risk	Detection time	3/6/2021, 11:44 AM
Risk level	High	Detection last updated	3/6/2021, 11:46 AM
Risk detail	-	Token issuer type	Azure AD
Source	Identity Protection		
Detection timing	Real-time		

Figure 7. Risky sign-in from TOR browser

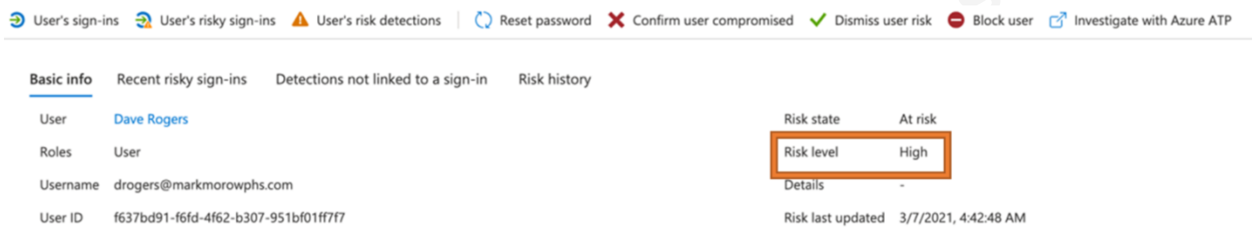


Confirm sign-in compromised?

This confirms to Azure AD that the selected sign-in was unauthorized. Azure AD will move the selected sign-in and involved user to High risk and optimize future risk assessment.
Learn more

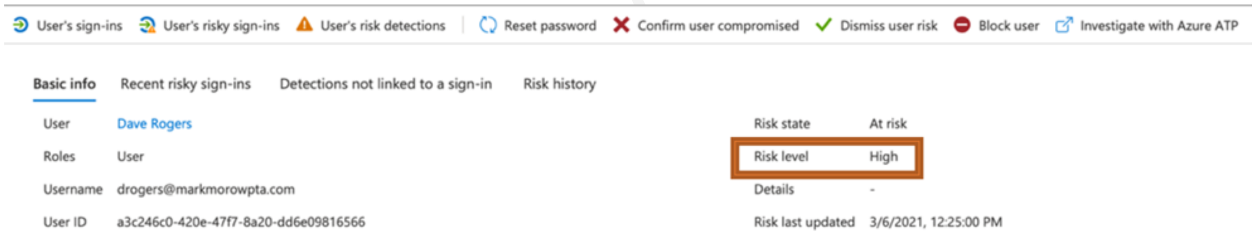
Figure 8. Confirm sign-in compromised

The result is that the user is now at high risk which can be seen in the PHS (Figure 9), PTA (Figure 10), and federated (Figure 11) environments by clicking on “Risky users”.



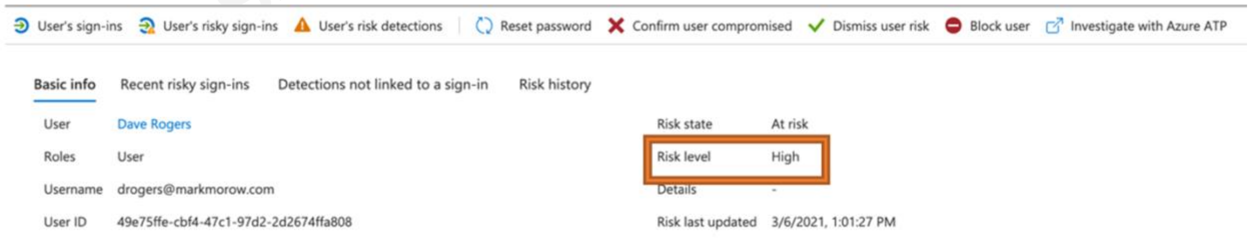
User	Dave Rogers	Risk state	At risk
Roles	User	Risk level	High
Username	drogers@markmorowphs.com	Details	-
User ID	f637bd91-f6fd-4f62-b307-951bf01ff7f7	Risk last updated	3/7/2021, 4:42:48 AM

Figure 9. PHS environment risk level high



User	Dave Rogers	Risk state	At risk
Roles	User	Risk level	High
Username	drogers@markmorowpta.com	Details	-
User ID	a3c246c0-420e-47f7-8a20-dd6e09816566	Risk last updated	3/6/2021, 12:25:00 PM

Figure 10. PTA environment risk level high



User	Dave Rogers	Risk state	At risk
Roles	User	Risk level	High
Username	drogers@markmorow.com	Details	-
User ID	49e75ffe-cbf4-47c1-97d2-2d2674ffa808	Risk last updated	3/6/2021, 1:01:27 PM

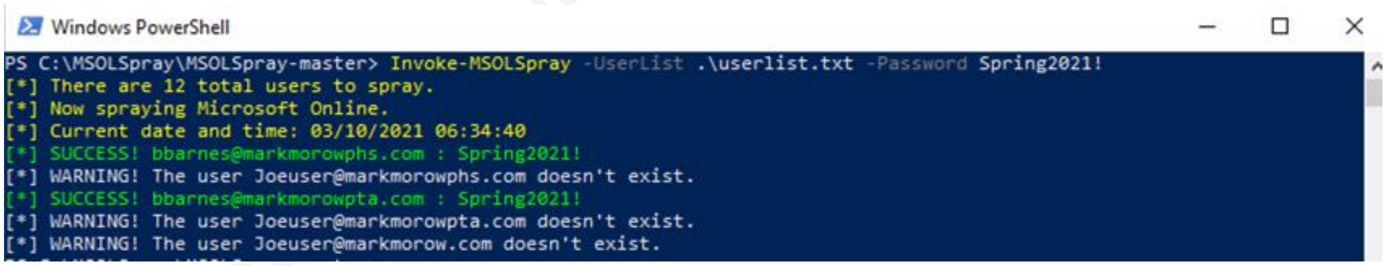
Figure 11. Federated environment risk level high

4.2.1 Following Guidance on Credential Reuse

A leaked credential is a high alert from the Azure AD security operations guide. These should be acknowledged and remediated within 24 hours per this experiment’s SLA. The remediation guidance is for the user to reset their password following their normal corporate password reset procedures. A 24-hour detection would be much improved over our default of 56 days.

4.3 Password Spray Attack

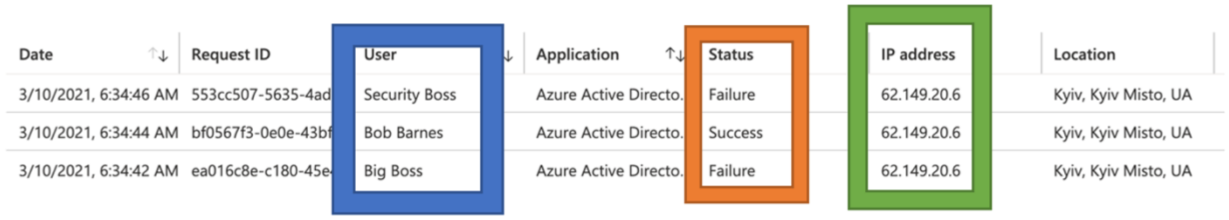
There are various password spray tools available that typically leverage legacy authentication protocols such as SMTP, IMAP4, and POP3. For this study, a user will be enabled for an Exchange Online mailbox on Office365. To simulate this attack, the tool MSOLSpray in conjunction with a VPN will be used to generate logins that originate from a non-normal location for these users. This script is available at https://github.com/dafthack/MSOLSpray. As per the tool's instructions, a list of users was passed to the tool along with the easily guessable password to attempt (Figure 12). The list of users was a mix of valid users in all three environments and some users that don't exist. As noted in the output of the tool there was a successful password spray against the Bob Barnes account.



```
PS C:\MSOLSpray\MSOLSpray-master> Invoke-MSOLSpray -UserList .\userlist.txt -Password Spring2021!
[*] There are 12 total users to spray.
[*] Now spraying Microsoft Online.
[*] Current date and time: 03/10/2021 06:34:40
[*] SUCCESS! bbarnes@markmorowphs.com : Spring2021!
[*] WARNING! The user Joeuser@markmorowphs.com doesn't exist.
[*] SUCCESS! bbarnes@markmorowpta.com : Spring2021!
[*] WARNING! The user Joeuser@markmorowpta.com doesn't exist.
[*] WARNING! The user Joeuser@markmorow.com doesn't exist.
```

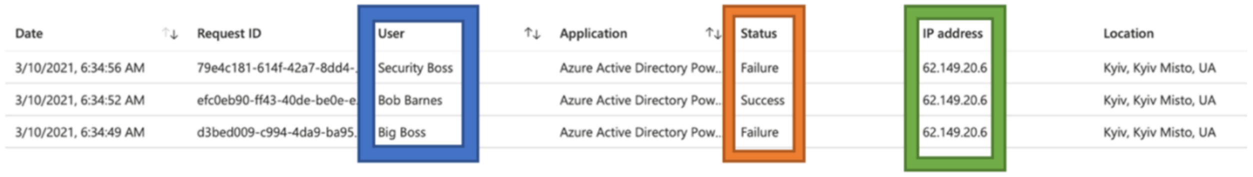
Figure 12. MSOLSpray password spraying accounts with password Spring2021!

By inspecting the PHS (Figure 13) and PTA (Figure 14) tenants' logs by going to the Azure AD portal under the Monitoring section selecting "Sign-ins", the failed sign-in attempts (orange box) for the different users (blue box) and then the successful sign-in for another user can be seen, all from the same IP address (green box).



Date	Request ID	User	Application	Status	IP address	Location
3/10/2021, 6:34:46 AM	553cc507-5635-4ad	Security Boss	Azure Active Directo.	Failure	62.149.20.6	Kyiv, Kyiv Misto, UA
3/10/2021, 6:34:44 AM	bf0567f3-0e0e-43bf	Bob Barnes	Azure Active Directo.	Success	62.149.20.6	Kyiv, Kyiv Misto, UA
3/10/2021, 6:34:42 AM	ea016c8e-c180-45e	Big Boss	Azure Active Directo.	Failure	62.149.20.6	Kyiv, Kyiv Misto, UA

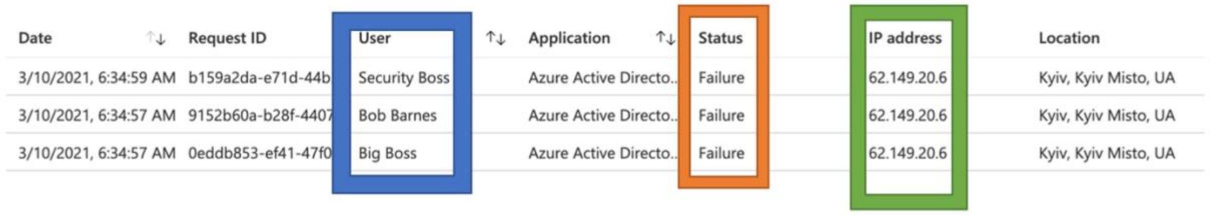
Figure 13. PHS environment password spray



Date	Request ID	User	Application	Status	IP address	Location
3/10/2021, 6:34:56 AM	79e4c181-614f-42a7-8dd4-	Security Boss	Azure Active Directory Pow..	Failure	62.149.20.6	Kyiv, Kyiv Misto, UA
3/10/2021, 6:34:52 AM	efc0eb90-ff43-40de-be0e-e	Bob Barnes	Azure Active Directory Pow..	Success	62.149.20.6	Kyiv, Kyiv Misto, UA
3/10/2021, 6:34:49 AM	d3bed009-c994-4da9-ba95	Big Boss	Azure Active Directory Pow..	Failure	62.149.20.6	Kyiv, Kyiv Misto, UA

Figure 14. PTA environment password spray

For the federated tenant, a different behavior was displayed. All attempts were blocked. This was not what was expected based on the understanding of the authentication flows. What was expected was these authentication requests would be passed back to the IDP, in this case, ADFS 2019, and only the successful logins would be visible in the Azure AD logs. Instead, failure for all attempts were present in the Azure AD logs (Figure 15).



Date	Request ID	User	Application	Status	IP address	Location
3/10/2021, 6:34:59 AM	b159a2da-e71d-44b	Security Boss	Azure Active Directo..	Failure	62.149.20.6	Kyiv, Kyiv Misto, UA
3/10/2021, 6:34:57 AM	9152b60a-b28f-4407	Bob Barnes	Azure Active Directo..	Failure	62.149.20.6	Kyiv, Kyiv Misto, UA
3/10/2021, 6:34:57 AM	0eddb853-ef41-47f0	Big Boss	Azure Active Directo..	Failure	62.149.20.6	Kyiv, Kyiv Misto, UA

Figure 15. Federated environment password spray

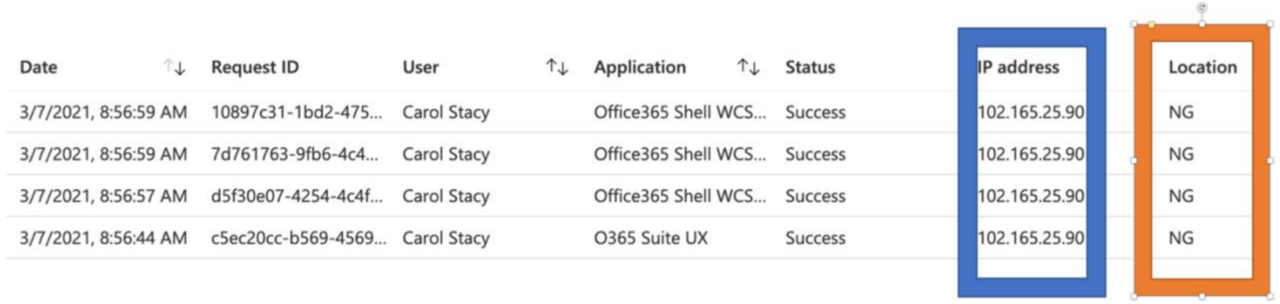
Investigation of the Azure AD documentation revealed that in this case, this is actually expected behavior. This tool was using an ROPC (Resource Owner Password Credentials) flow. These flows are blocked by default in Azure AD for federated tenants. An admin would need to take action to enable this (Microsoft, 2021). Discussing the type of application authentication flows in Azure AD is outside this paper's scope, but more details can be found in the Azure AD developer documentation (Microsoft, 2021). Other password spray tools may use different protocols that may give the original expected behavior.

4.3.1 Following Guidance on Password Spray

Following the Azure AD security operations guide, a successful password spray would be a high alert that should be investigated within 24 hours per our SLA. The remediation guidance is for the user to reset their password following their normal corporate password reset procedures. A 24-hour detection would be much improved over the default of 56 days.

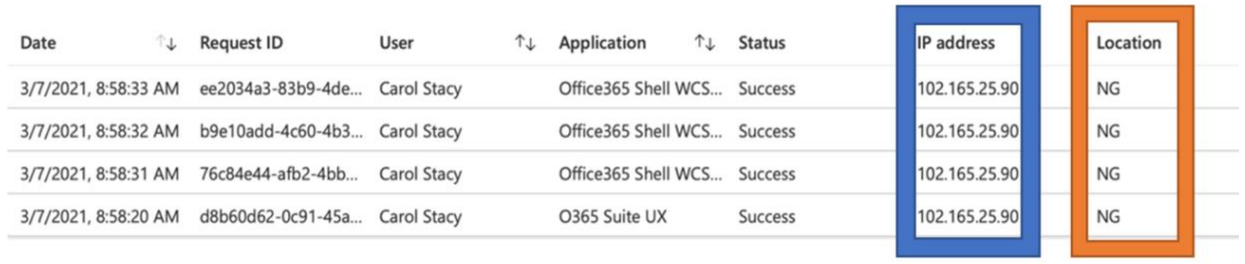
4.4 Phishing

To simulate a phishing attack, the user has already given up their credentials to an attacker and the attacker is now leveraging them. The attacker will typically try to login from an anonymizing service or possibly a location near to them. Since TOR was previously utilized and the results were observed, a VPN will be used to specify the location exit point. A user will then login from Nigeria. The results below from the PHS (Figure 16), PTA (Figure 17), and federated (Figure 18) environments, are visible in the Azure AD Sign-in logs like before with the IP (blue) and the location (orange) Nigeria (NG).



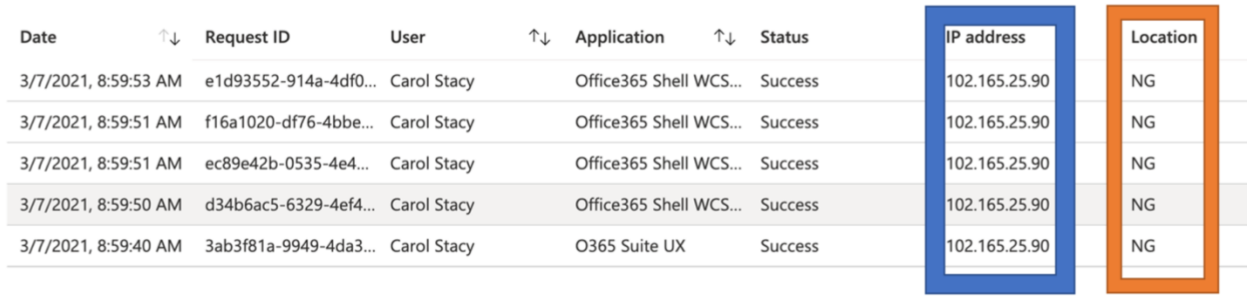
Date	Request ID	User	Application	Status	IP address	Location
3/7/2021, 8:56:59 AM	10897c31-1bd2-475...	Carol Stacy	Office365 Shell WCS...	Success	102.165.25.90	NG
3/7/2021, 8:56:59 AM	7d761763-9fb6-4c4...	Carol Stacy	Office365 Shell WCS...	Success	102.165.25.90	NG
3/7/2021, 8:56:57 AM	d5f30e07-4254-4c4f...	Carol Stacy	Office365 Shell WCS...	Success	102.165.25.90	NG
3/7/2021, 8:56:44 AM	c5ec20cc-b569-4569...	Carol Stacy	O365 Suite UX	Success	102.165.25.90	NG

Figure 16. PHS environment login from Nigeria



Date	Request ID	User	Application	Status	IP address	Location
3/7/2021, 8:58:33 AM	ee2034a3-83b9-4de...	Carol Stacy	Office365 Shell WCS...	Success	102.165.25.90	NG
3/7/2021, 8:58:32 AM	b9e10add-4c60-4b3...	Carol Stacy	Office365 Shell WCS...	Success	102.165.25.90	NG
3/7/2021, 8:58:31 AM	76c84e44-afb2-4bb...	Carol Stacy	Office365 Shell WCS...	Success	102.165.25.90	NG
3/7/2021, 8:58:20 AM	d8b60d62-0c91-45a...	Carol Stacy	O365 Suite UX	Success	102.165.25.90	NG

Figure 17. PTA environment login from Nigeria



Date	Request ID	User	Application	Status	IP address	Location
3/7/2021, 8:59:53 AM	e1d93552-914a-4df0...	Carol Stacy	Office365 Shell WCS...	Success	102.165.25.90	NG
3/7/2021, 8:59:51 AM	f16a1020-df76-4bbe...	Carol Stacy	Office365 Shell WCS...	Success	102.165.25.90	NG
3/7/2021, 8:59:51 AM	ec89e42b-0535-4e4...	Carol Stacy	Office365 Shell WCS...	Success	102.165.25.90	NG
3/7/2021, 8:59:50 AM	d34b6ac5-6329-4ef4...	Carol Stacy	Office365 Shell WCS...	Success	102.165.25.90	NG
3/7/2021, 8:59:40 AM	3ab3f81a-9949-4da3...	Carol Stacy	O365 Suite UX	Success	102.165.25.90	NG

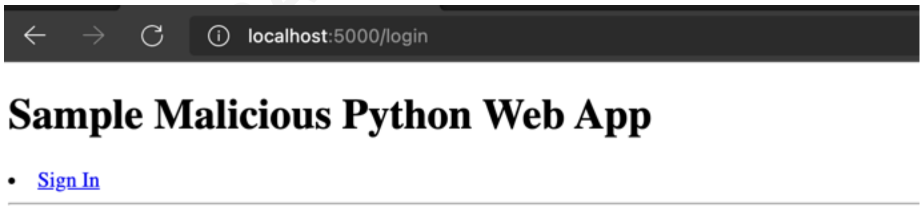
Figure 18. Federated environment login from Nigeria

4.4.1 Following Guidance on Phishing

A successful login from a country where the business is not usually done is a medium alert from the Azure AD security operations guide. These should be acknowledged and remediated within 72 hours per the experiment SLA. The remediation guidance is for the user to reset their password following their normal corporate password reset procedures. A 72-hour detection would be much improved over our default of 56 days.

4.5 Application Consent Phishing

A sample Python web app was made to simulate a malicious multi-tenant application. A multi-tenant application would best simulate a real-world application consent phishing attack. This application is hosted in another Azure AD test tenant, MarkMorowsMaliciousAppLab.com. The setup for making your own multi-tenant application and application source is also provided at https://github.com/MarkMorow/AppConsentPhishingSample. To step through this attack, the web app is running on the localhost (Figure 19).


localhost:5000/login

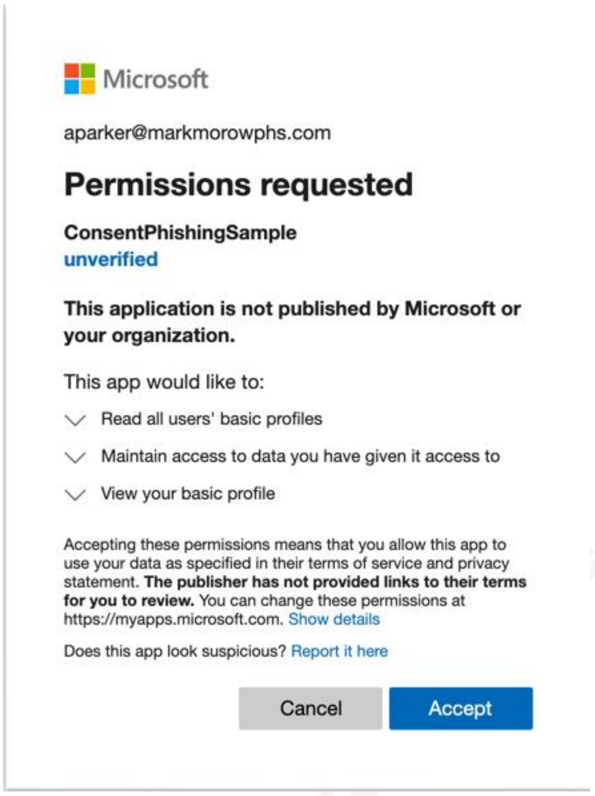
Sample Malicious Python Web App

- Sign In

Figure 19. Sample Python web app running on the localhost

Clicking on "Sign In" prompts the user to authenticate to Azure AD. In this example, the user Alice Parker in the PHS environment has successfully authenticated. They are then presented with this application consent screen below (Figure 20). In an actual attack, this is where the user would land after clicking on the malicious link. If they were already signed-in to Azure AD, that existing session token would be used, and they would not have to sign-in. The same experience was observed for the PTA (Figure 21) and federated (Figure 22) environments.

Mark Morowczynski, mark.morowczynski@students.sans.edu



aparker@markmorowphs.com

Permissions requested

ConsentPhishingSample
unverified

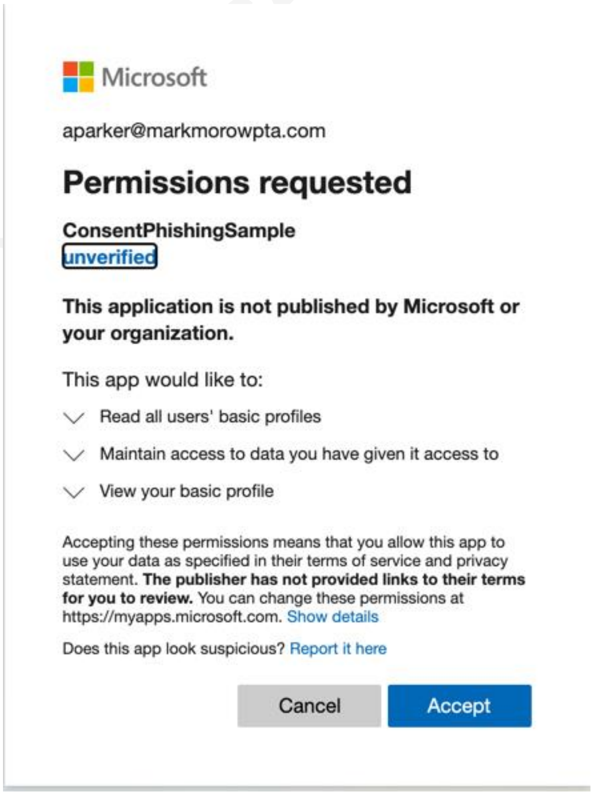
This application is not published by Microsoft or your organization.

This app would like to:
- Read all users' basic profiles
- Maintain access to data you have given it access to
- View your basic profile

Accepting these permissions means that you allow this app to use your data as specified in their terms of service and privacy statement. **The publisher has not provided links to their terms for you to review.** You can change these permissions at https://myapps.microsoft.com. Show details

Does this app look suspicious? Report it here

Figure 20. PHS environment application consent



aparker@markmorowpta.com

Permissions requested

ConsentPhishingSample
unverified

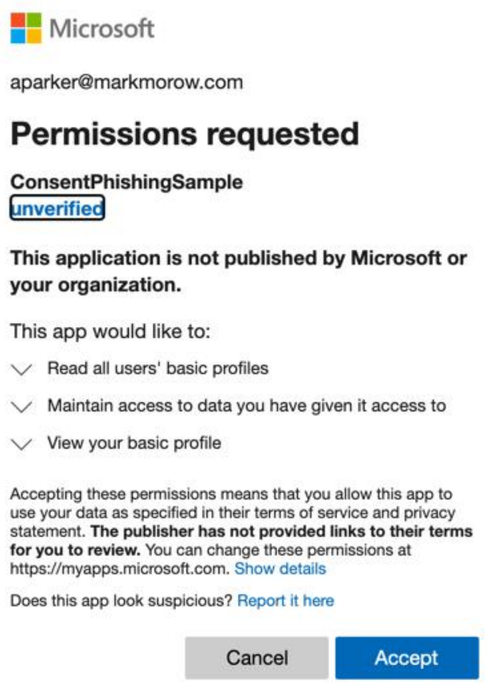
This application is not published by Microsoft or your organization.

This app would like to:
- Read all users' basic profiles
- Maintain access to data you have given it access to
- View your basic profile

Accepting these permissions means that you allow this app to use your data as specified in their terms of service and privacy statement. **The publisher has not provided links to their terms for you to review.** You can change these permissions at https://myapps.microsoft.com. Show details

Does this app look suspicious? Report it here

Figure 21. PTA environment application consent


Microsoft

aparker@markmorow.com

Permissions requested

ConsentPhishingSample
unverified

This application is not published by Microsoft or your organization.

This app would like to:

- Read all users' basic profiles
- Maintain access to data you have given it access to
- View your basic profile

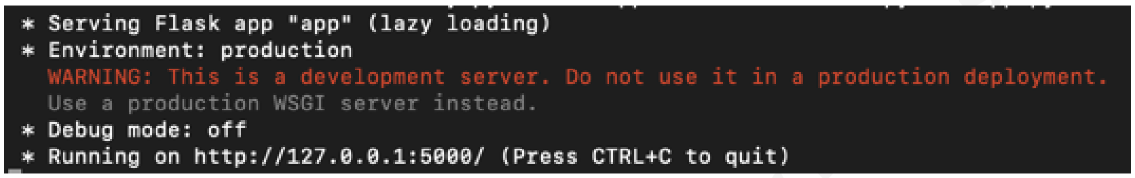
Accepting these permissions means that you allow this app to use your data as specified in their terms of service and privacy statement. **The publisher has not provided links to their terms for you to review.** You can change these permissions at https://myapps.microsoft.com. Show details

Does this app look suspicious? Report it here

Figure 22. Federated environment application consent

The application is requesting basic permissions (User.ReadBasic.All) that users can consent to by default. If the application permissions were more privileged, the user would not be able to consent to them and would require administrator permissions to complete the consent request. When the user clicks accept, they are granting this application delegated permissions with their account. An administrator account is not immune to this type of attack and can consent to an application for themselves or all users in the directory. Permissions can vary to the ability to read mail (Mail.Read) or even read and write objects to the directory (Directory.ReadWrite.All).

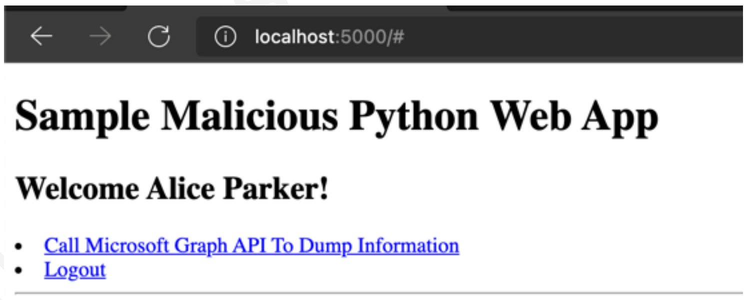
Readers should note that when using this sample application, the application is configured to redirect to the URL http://localhost. To access this correctly you will need to type in your browser URL http://localhost:5000. If you use http://127.0.0.1:5000 to access the web app, after you click accept on the application consent you will get a redirect URL error since the address you are redirected to, http://127.0.0.1, doesn't match http://localhost. This is an easy mistake to make if you are reading the Python webserver output and navigate to that exact address (Figure 23).



```
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Figure 23. Python webserver output

At this point, the malicious app can now call the GraphAPI on behalf of this signed-in user and see what that user would be able to see (Figure 24). Members of the PHS environment directory including the CEO of the company, aka Big Boss, and their user principal name are displayed in the web app (Figure 25). In many environments, this is the same as their email address. The output has been truncated for readability. The experience was the same for both PTA (Figure 26) and federated (Figure 27) environments.



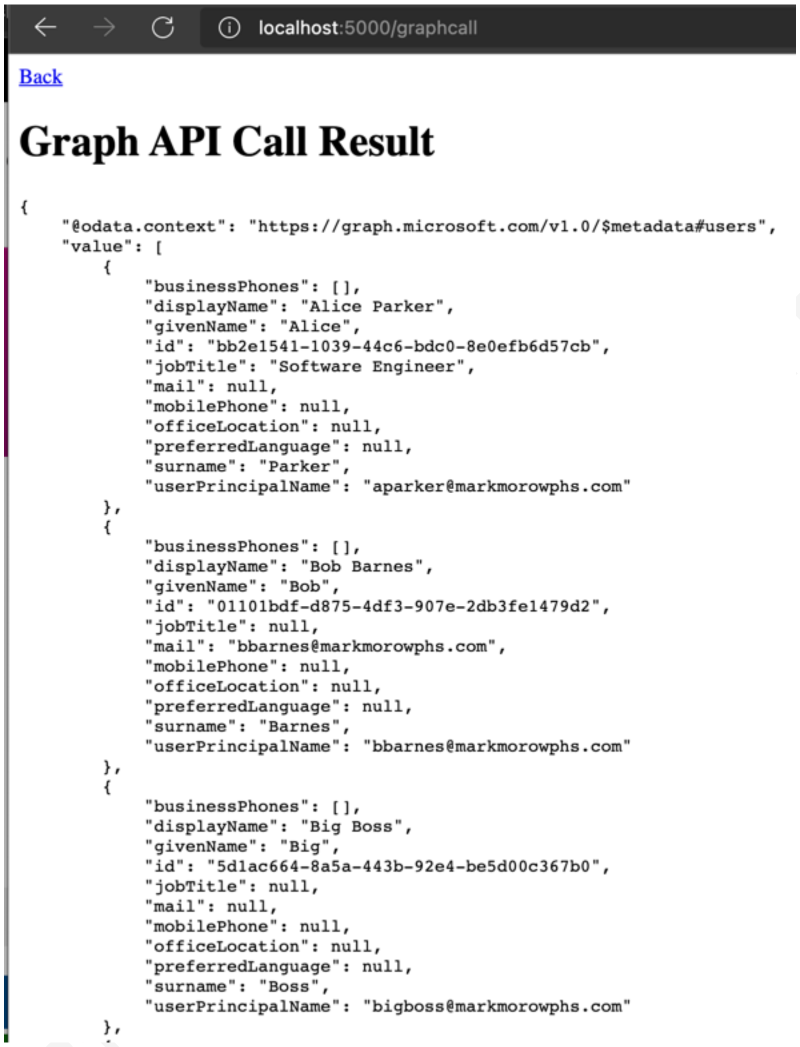
Sample Malicious Python Web App

Welcome Alice Parker!

- Call Microsoft Graph API To Dump Information
- Logout

Figure 24. Malicious app signed-in as the user

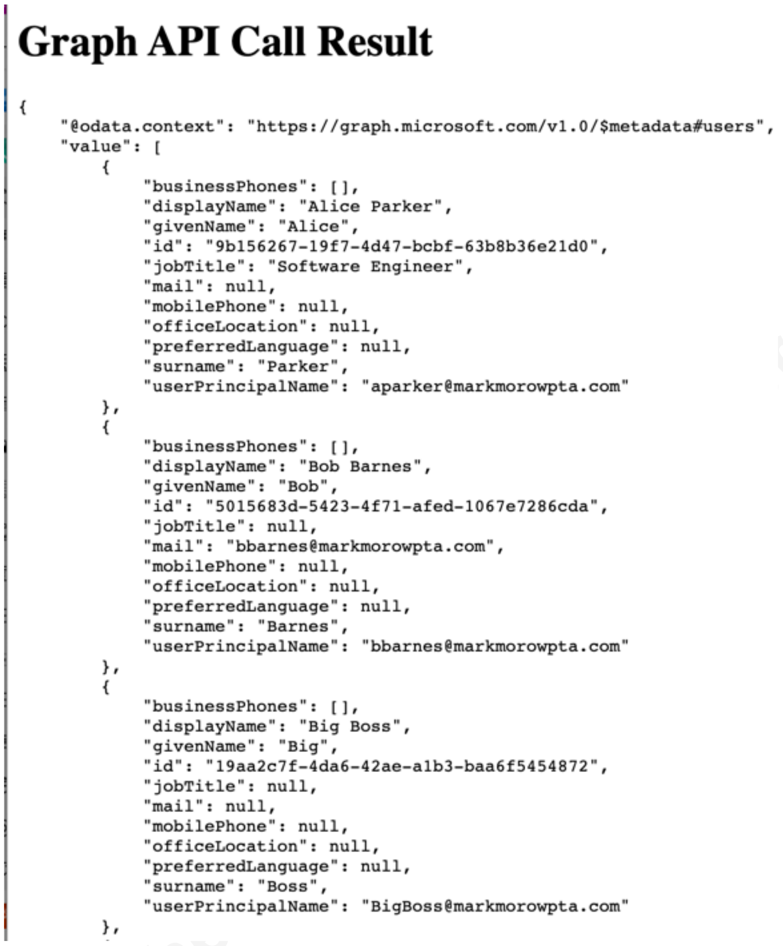
Mark Morowczynski, mark.morowczynski@students.sans.edu



Graph API Call Result

```
{
    "@odata.context": "https://graph.microsoft.com/v1.0/$metadata#users",
    "value": [
        {
            "businessPhones": [],
            "displayName": "Alice Parker",
            "givenName": "Alice",
            "id": "bb2e1541-1039-44c6-bdc0-8e0efb6d57cb",
            "jobTitle": "Software Engineer",
            "mail": null,
            "mobilePhone": null,
            "officeLocation": null,
            "preferredLanguage": null,
            "surname": "Parker",
            "userPrincipalName": "aparker@markmorowphs.com"
        },
        {
            "businessPhones": [],
            "displayName": "Bob Barnes",
            "givenName": "Bob",
            "id": "01101bdf-d875-4df3-907e-2db3fe1479d2",
            "jobTitle": null,
            "mail": "bbarnes@markmorowphs.com",
            "mobilePhone": null,
            "officeLocation": null,
            "preferredLanguage": null,
            "surname": "Barnes",
            "userPrincipalName": "bbarnes@markmorowphs.com"
        },
        {
            "businessPhones": [],
            "displayName": "Big Boss",
            "givenName": "Big",
            "id": "5d1ac664-8a5a-443b-92e4-be5d00c367b0",
            "jobTitle": null,
            "mail": null,
            "mobilePhone": null,
            "officeLocation": null,
            "preferredLanguage": null,
            "surname": "Boss",
            "userPrincipalName": "bigboss@markmorowphs.com"
        },
```

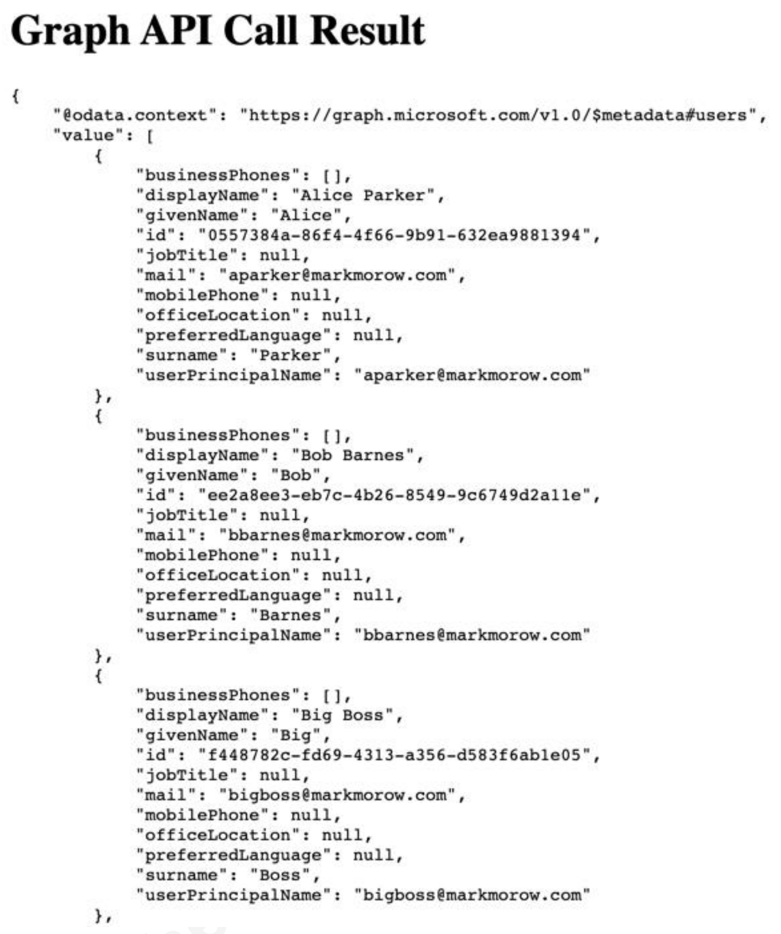
Figure 25. PHS environment GraphAPI Directory Read output



Graph API Call Result

```
{
    "@odata.context": "https://graph.microsoft.com/v1.0/$metadata#users",
    "value": [
        {
            "businessPhones": [],
            "displayName": "Alice Parker",
            "givenName": "Alice",
            "id": "9b156267-19f7-4d47-bcbf-63b8b36e21d0",
            "jobTitle": "Software Engineer",
            "mail": null,
            "mobilePhone": null,
            "officeLocation": null,
            "preferredLanguage": null,
            "surname": "Parker",
            "userPrincipalName": "aparker@markmorowpta.com"
        },
        {
            "businessPhones": [],
            "displayName": "Bob Barnes",
            "givenName": "Bob",
            "id": "5015683d-5423-4f71-afed-1067e7286cda",
            "jobTitle": null,
            "mail": "bbarnes@markmorowpta.com",
            "mobilePhone": null,
            "officeLocation": null,
            "preferredLanguage": null,
            "surname": "Barnes",
            "userPrincipalName": "bbarnes@markmorowpta.com"
        },
        {
            "businessPhones": [],
            "displayName": "Big Boss",
            "givenName": "Big",
            "id": "19aa2c7f-4da6-42ae-a1b3-baa6f5454872",
            "jobTitle": null,
            "mail": null,
            "mobilePhone": null,
            "officeLocation": null,
            "preferredLanguage": null,
            "surname": "Boss",
            "userPrincipalName": "BigBoss@markmorowpta.com"
        },
```

Figure 26. PTA environment GraphAPI Directory Read output



Graph API Call Result

```
{
    "@odata.context": "https://graph.microsoft.com/v1.0/$metadata#users",
    "value": [
        {
            "businessPhones": [],
            "displayName": "Alice Parker",
            "givenName": "Alice",
            "id": "0557384a-86f4-4f66-9b91-632ea9881394",
            "jobTitle": null,
            "mail": "aparker@markmorow.com",
            "mobilePhone": null,
            "officeLocation": null,
            "preferredLanguage": null,
            "surname": "Parker",
            "userPrincipalName": "aparker@markmorow.com"
        },
        {
            "businessPhones": [],
            "displayName": "Bob Barnes",
            "givenName": "Bob",
            "id": "ee2a8ee3-eb7c-4b26-8549-9c6749d2a11e",
            "jobTitle": null,
            "mail": "bbarnes@markmorow.com",
            "mobilePhone": null,
            "officeLocation": null,
            "preferredLanguage": null,
            "surname": "Barnes",
            "userPrincipalName": "bbarnes@markmorow.com"
        },
        {
            "businessPhones": [],
            "displayName": "Big Boss",
            "givenName": "Big",
            "id": "f448782c-fd69-4313-a356-d583f6ab1e05",
            "jobTitle": null,
            "mail": "bigboss@markmorow.com",
            "mobilePhone": null,
            "officeLocation": null,
            "preferredLanguage": null,
            "surname": "Boss",
            "userPrincipalName": "bigboss@markmorow.com"
        },
```

Figure 27. Federated environment GraphAPI Directory Read output

4.5.1 Following Guidance on Application Consent Phishing

A user accepting an application consent does not necessarily indicate malicious behavior. Most application consent users accept are not malicious. However, the Azure AD Audit logs should be reviewed for application consent as a low alert for the SOC. Looking for other indicators of a suspicious app such as the app trying to masquerade as another well-known app, blend in as a bland app (‘email’), and misspellings all require further investigation.

A 120-hour detection would be much improved over our default of 56 days. In reality, this type of attack would likely go even longer from detection due to its newness and rarity.

5. Recommendations and Future Research

Four cloud identity-based attacks consisting of credential reuse, password spray, phishing, and application consent phishing were all performed to simulate an adversary against the three typical Azure AD tenant authentication setups of password hash sync, pass-through authentication, and federation.

The original question proposed was: does following the Azure AD security operations guidance reduce the dwell time for common cloud identity attacks? The data shows the answer is yes, drastically. Using the FireEye average of 56 days to detect, credential reuse, password spray, and phishing was detected within our high alert SLA of 24 hours. Following the security operations guidance detected our emerging application consent attack as a low alert SLA of 120 hours. Though these SLA numbers were simulated for the purpose of the experiment, any SOC should be able to apply these detection priorities to their existing schemes and significantly improve their detective controls for their cloud-based identities.

The cloud will continue to grow in usage, and further research will need to be done. This research was focused on using recommend detection controls in Azure AD. These could also be applied to other cloud-based identity providers. Furthermore, there are recommended preventive controls that could be applied to Azure AD (Gupta, 2018). The next step would be to apply those controls and determine if this enhances these detective controls to decrease dwell time even further or prevent many of these attacks outright.

Finally, there are many research opportunities in these modern cloud applications and the underlying protocols. Further research should be done on implementation mistakes of SAML, OAuth, and OpenID Connect and the different authentication flows such as ROPC and implicit grant, to name a few. Lastly, the emerging application consent phishing attack has additional research opportunities as it becomes more well known by both attackers and defenders.

6. Conclusion

Cloud adoption has grown rapidly and will continue to grow as its benefits are too great to ignore. Attackers targeting the cloud will also continue to rise, with identity being a primary target. Reducing the dwell time of a breach is critical to reducing the cost and the compromise scope. On the low end, the average dwell time is 56 days, and on the higher end, it is 280 days. All hope for defenders is not lost, though. Focusing on detecting the most common identity attacks and a new emerging attack will allow companies to focus their resources to be the most beneficial for the defense. By following the Azure AD security operations guidance, companies can drastically reduce the dwell time of attackers.

References

Anderson, B. (2018, September 8). Success with Hybrid Cloud: Getting deep – Azure Active Directory . Retrieved from Microsoft Security and Compliance: https://techcommunity.microsoft.com/t5/microsoft-security-and/success-with-hybrid-cloud-getting-deep-8211-azure-active/ba-p/248213

Bertocci, V. (2016). Modern Authentication with Azure Active Directory for Web Applications. Redmon: Microsoft Press.

Chik, J. (2021, January 27). The state of apps by Microsoft Identity: Azure AD app gallery apps that made the most impact in 2020. Retrieved from Microsoft Security Blog: https://www.microsoft.com/security/blog/2021/01/27/the-state-of-apps-by-microsoft-identity-azure-ad-app-gallery-apps-that-made-the-most-impact-in-2020/

FireEye Mandiat. (2020). M-Trends 2020. Milpitas: FireEye Mandiant.

Gupta, N. &. (2018). Shut the door to cybercrime with identity-driven security. Ignite 2018. Orlando: Microsoft.

Hunt, T. (2021, March 3). Have I Been Pwned? Retrieved from Have I Been Pwned? : https://haveibeenpwned.com/

Microsoft. (2021, March 13). Authentication Flows. Retrieved from Documentation: https://docs.microsoft.com/en-us/azure/active-directory/develop/msal-authentication-flows#usernamepassword

Microsoft. (2021, March 3). Azure Active Direcotry Pass-though Authentication: Technical deep dive. Retrieved from Documentation: https://docs.microsoft.com/en-us/azure/active-directory/hybrid/how-to-connect-pta-how-it-works

Microsoft. (2021, March 7). Azure Active Directory Identity Protection - Security overview. Retrieved from Documentation Identity Protection: https://docs.microsoft.com/en-us/azure/active-directory/identity-protection/concept-identity-protection-security-overview#high-risk-users

Microsoft. (2021, March 2). Azure AD federation compatibility list . Retrieved from Documentation: https://www.microsoft.com/en-us/download/confirmation.aspx?id=56843

Microsoft. (2021, March 13). Configure Azure Active Directory sign in behavior for an application by using a Home Realm Discovery policy. Retrieved from Documentation: https://docs.microsoft.com/en-us/azure/active-directory/manage-apps/configure-authentication-for-federated-users-portal#enable-direct-ropc-authentication-of-federated-users-for-legacy-applications

Microsoft. (2021, March 27). Docs. Retrieved from AD FS sign-ins in Azure AD with Connect Health - preview: https://docs.microsoft.com/en-us/azure/active-directory/hybrid/how-to-connect-health-ad-fs-sign-in

Microsoft. (2021, March 3). Implement password hash synchronization with Azure AD Connect sync. Retrieved from Documentation: https://docs.microsoft.com/en-us/azure/active-directory/hybrid/how-to-connect-password-hash-synchronization

Microsoft. (2021, March 7). What are security defaults? Retrieved from Azure Active Directory Fundamentals: https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/concept-fundamentals-security-defaults

Microsoft. (2021, March 7). What is Risk? Retrieved from Identity Protection Documentation: https://docs.microsoft.com/en-us/azure/active-directory/identity-protection/concept-identity-protection-risks#leaked-credentials

Microsoft Corporation. (2020). Microsoft Digital Defense Report. Redmond: Microsoft.

Microsoft Corporation. (2021, March 3). Azure Active Directory. Retrieved from https://www.microsoft.com/en-us/security/business/identity-access-management/azure-active-directory

Nemeretes. (2019). Cutting SecOps breach response time is key to success: A new survey measures the success of security operations breach response by how long it takes to complete a three-step process to detect, understand and contain incidents. Information Security, 1.

Ponemon Institute. (2020). Cost of Data Breach Report. Armonk: IBM.

SANS. (2021, March 7). Data Incident 2020. Retrieved from SANS: https://www.sans.org/dataincident2020?msc=data-incident-2020

Verizon. (2020). 2020 Data Breach Investigations Report. New York City: Verizon.

Weinert, A. (2020, December 21). Understanding "Solorigate's" Identity IOCs- For Identity Vendors and their customers. Retrieved from Azure Active Directory Identity Blog: https://techcommunity.microsoft.com/t5/azure-active-directory-identity/understanding-quot-solorigate-quot-s-identity-iocs-for-identity/ba-p/2007610