GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Hacker Tools, Techniques, and Incident Handling (Security 504)"
at http://www.giac.org/registration/gcih

Extracting Timely Sign-in Data from Office 365 Logs

GIAC (GCIH) Gold Certification

Author: Mark J. Lucas, mjlucas62@mac.com
Advisor: Johannes Ullrich
Accepted: 05/22/2018

Abstract

Office 365 is quickly becoming a repository of valuable organizational information, including data that falls under multiple privacy laws. Timely detection of a compromised account and stopping the bad guy before data is exfiltrated, destroyed, or the account used for nefarious purposes is the difference between an incident and a compromise. Microsoft provides audit logging and alerting tools that can assist system administrators find these incidents. An examination of the efficacy and efficiency of these tools and the shortcomings and advantages provides insight into how to best use the tools to protect individual accounts and the organization as a whole.

© 2018 The SANS Institute, Author Retains Full Rights

© 2018 The SANS Institute Author retains full rights.

1. Introduction

Identity management is a challenge - it is vital to ensure that the credentials presented match the person entering them. Spencer Lee (2003) provides an extensive review of the difficulties of this process of credential validation and some solutions to address it. One method utilized by my institute is geographic location data as reflected by a sign-in source IP address. A relatively small number (under 500 members) of the institute's 6,000-member user base frequently travel worldwide. Members travel to locations known to be the sources of malicious activity. For these members, it is vital to quickly ascertain that a particular authentication event is genuine and not the result of compromised credentials. Microsoft provides tools on-premises and in Office 365 which facilitate the validation of these events. An examination of the usefulness of these tools provides system administrators a basis for choosing which tools are best suited for their environments.

1.1. On-Premises History

When servers were physically installed on the institution's property and all authentication events were recorded and logged locally, it was possible to use the fundamental but powerful tools of syslog and Perl-based grep scripts to, in near real time, determine the location of a sign-in and alert for unusual source locations. Windows Domain Controller event logs which contain source IPs (Smith, 2018) were included in the Linux-based syslogs with the use of Snare (Intersect Alliance, 2018). Source IP geographic locations were readily available from multiple providers (IP Location, 2018). Due to the relatively small number of world travelers, the security team was able to quickly confirm with department administration if the individual in question was physically present in the source location. While this confirmation did require a fair amount of human intervention, manually checking travel lists against source location warnings proved to be highly effective at this point in the on-premises evolution.

1.2. Cloud-based authentication

With the introduction of Microsoft Office 365 and Active Directory Federation Services (ADFS), filtering authentication events through cloud services presented new challenges. Locating the ADFS infrastructure in Amazon Web Services (AWS) (Amazon

© 2018 The SANS Institute Author retains full rights.

© 2018 The SANS Institute, Author Retains Full Rights

Web Services, 2018) improved redundancy and availability. A component of the redundant configuration is the use of an Elastic Load Balancer (ELB) (Amazon Web Services, 2018). In 2015, during the implementation of the AWS solution, the execution team was challenged to configure the ELB to reliably forward the source IP address to the ADFS Proxy Servers. The reevaluation of the ELB has been a low priority, even though new features have been added (Amazon Web Services, 2018). In the same timeframe, Microsoft moved the ADFS Proxy Server from Internet Information Services (IIS) to Web Application Proxy (Mathers, 2017). The Web Application Proxy increased the security of the proxy server (Bahall, Gremban, Tilman, Casey, & Notin, 2017) but also removed the extended logging capabilities of IIS which allowed the recording of X-Forwarded-For records from the Elastic Load Balancers (Cooper, 2011).

The use of a thick-client, such as Microsoft Outlook or Lync (now Skype) passes the credentials through Office 365 before sending them to the ADFS Proxy (Gregory, 2014). Thus, recording of a source IP address likely occurs at Office 365 and is lost by the time it reaches the Proxy servers. Logs that are available from Office 365 are often delayed by hours (Redmond, 2016). This delay could result in significant damage by a malicious actor before the intrusion is detected.

1.3. Current situation

Most recently, Microsoft has introduced Modern Authentication (Gunnemo, 2016) and the Microsoft Trust Center (Microsoft Corporation, 2018). Exploration of Modern Authentication, which utilizes components of Active Directory, Office 365 and ADFS Services and the tools available in the Trust Center has not been completed. The institution's security team lacks an understanding of potential advantages or hurtles present in the analysis of the source IP addresses and identification of source geographic location.

Also, in 2016, Microsoft introduced the Unified Audit Log (Microsoft Corporation, 2016). This log provided a single source for many types of events including sign-in events; however, logging of authentication events must be enabled on individual mailboxes (Redmond T, 2016). There is also currently no way to enable an alert in the Audit Log Search for authentication from suspect IP addresses.

© 2018 The SANS Institute Author retains full rights.

2. Case Study

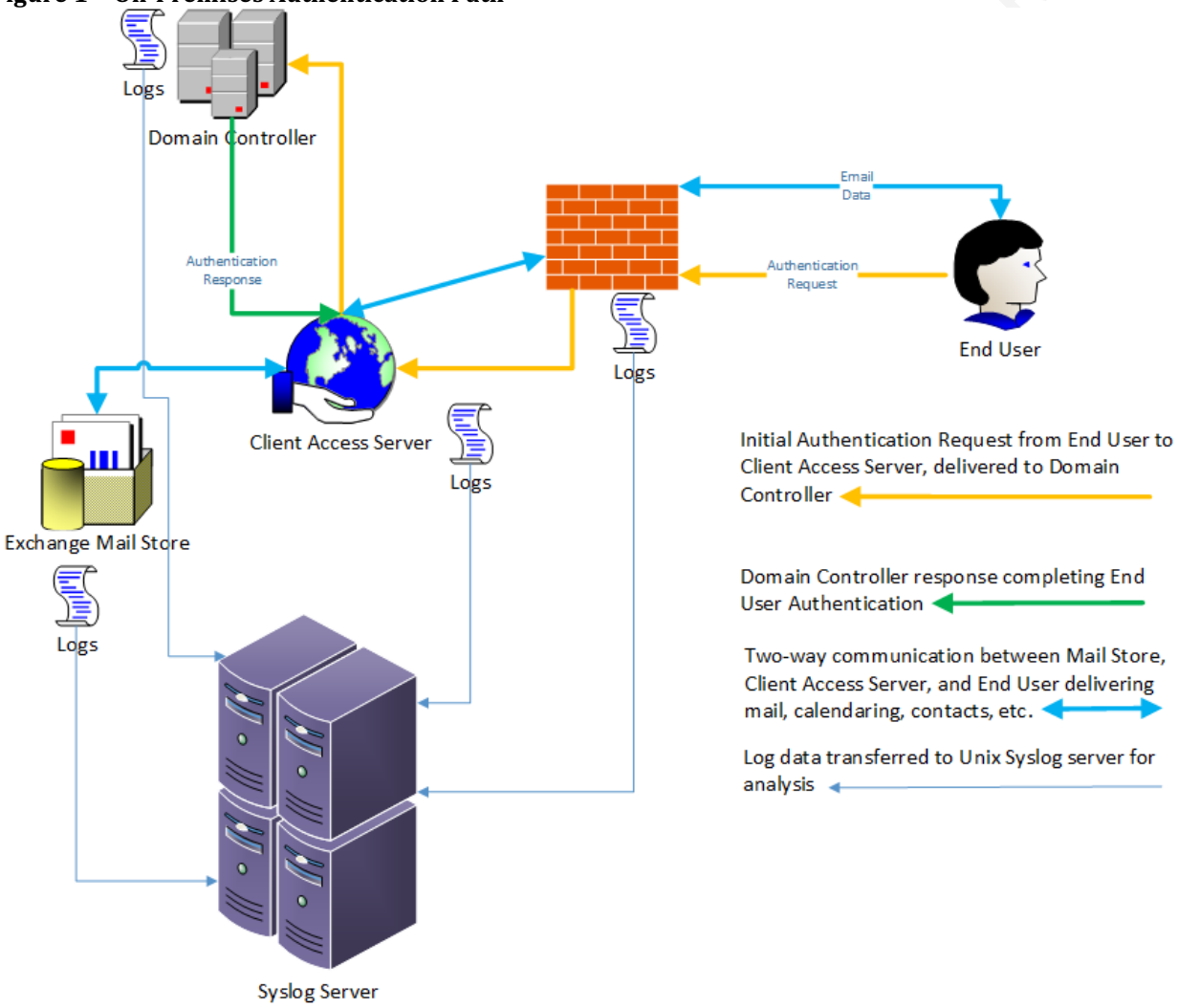
2.1. On-Premises History

Exchange was widely introduced at the institution with version Exchange 2003. Due to both budgeting and technical restrictions, Windows Network Load Balancing (NLB) was selected to load balance the two Exchange Client Access Servers (CAS). These servers handled MAPI, IMAP, and Web Browser connections (POP was turned off based on a policy decision by management). At the time, remote MAPI was not considered a vital concern because it was rarely, if ever, utilized. All MAPI authentication was handled directly by the domain controllers which had complete authentication logs which further mitigated the logging concerns. Internet Information Server (IIS) 6.0 logging handled and recorded all the logs for both IMAP and Web Browser connections. All logs from Exchange and the Domain Controllers were shunted to a standard Unix Syslog server using Snare (Intersect Alliance, 2018). The syslog server entries were monitored by the low tech, but effective use of GREP scripts that scanned for unusual source IP addresses. This configuration was largely unchanged with the upgrade to Exchange 2010. Figure 1 below shows the authentication path and logging locations for the on-premises configuration. All logs are copied to the syslog server for analysis and long-term storage.

© 2018 The SANS Institute, Author Retains Full Rights

© 2018 The SANS Institute Author retains full rights.

Figure 1 - On-Premises Authentication Path



2.2. Active Directory Federation Services

Active Directory Federation Services (ADFS), the Microsoft implementation of Federated Services for authentication between Office 365 and on-premises Active Directory (Mathers, Kumar, & Plett, Active Directory Federation Services, 2017) and the implementation in Amazon Web Services (Amazon Web Services, 2018) brought new challenges. ADFS 3.0 utilizes the role of Web Application Proxy which reduces the previous IIS logging capabilities to almost zero and eliminates the ability to capture source IP addresses of web-based sign-ins. Additionally, IMAP, ActiveSync, and HTTPS-based MAPI connections from Outlook and phone connections are routed through Microsoft-based authentication servers first and then are routed to the ADFS servers for final authentication. This authentication path completely removes the ability

© 2018 The SANS Institute Author retains full rights.

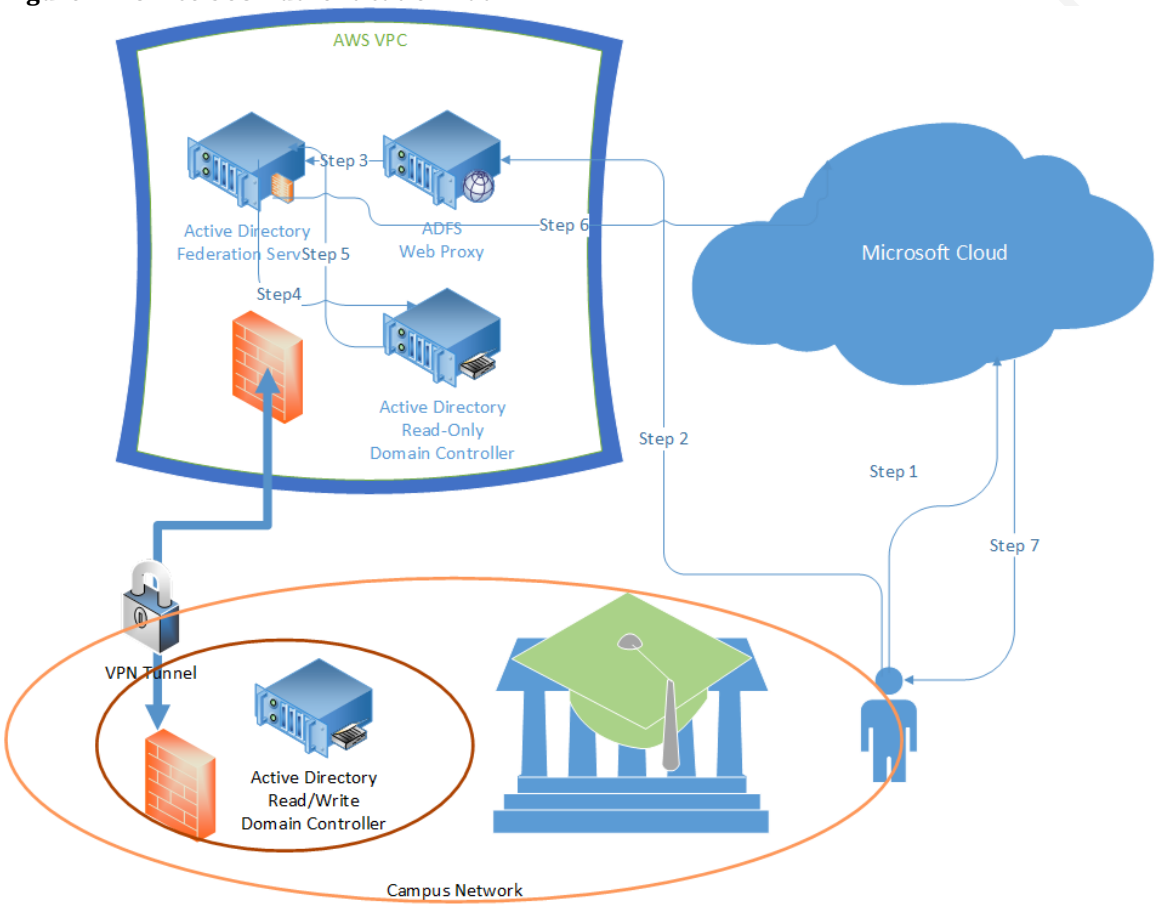
to capture the source IP address in logs controlled by the organization. Microsoft-based logging is required to obtain the source IP information.

As an example, Figure 2 below shows the path steps for login to Office 365 from a web browser. For simplicity, the load balancer and redundant servers have been excluded from the diagram.

1. The user enters their username on the Microsoft site, http://portal.office.com. In this case, the user can be on or off campus because the campus network is connected directly to the internet and all workstations have internet routable IP addresses.
2. Microsoft cloud determines the correct tenant for sign-in and redirects the browser to the ADFS servers.
3. Username and password is entered on the web proxy server.
4. Credentials are passed to the Federation Server.
5. Federation Server authenticates against the local read-only Active Directory Domain Controller (DC).
    a. The read-only DCs are synchronized to read-write DCs on-premises through a site-to-site VPN tunnel.
6. Federation Server passes an authentication token to the Microsoft Cloud.
7. Microsoft Cloud presents the user's data to the web browser.

© 2018 The SANS Institute Author retains full rights.

Figure 2 - Office 365 Authentication Path



2.3. Microsoft Trust Center – Graphical Interface

2.3.1. Data Availability

Before any owner mailbox sign-in activity can be monitored, the system administrator uses PowerShell to enable logging on the mailbox owner. By default, enabling auditing initiates logging for the Update, SoftDelete, HardDelete, SendAs, Create, and UpdateFolderPermissions actions for delegates, but not for owners. To permit default logging and audit owner sign-in activities, run the command:

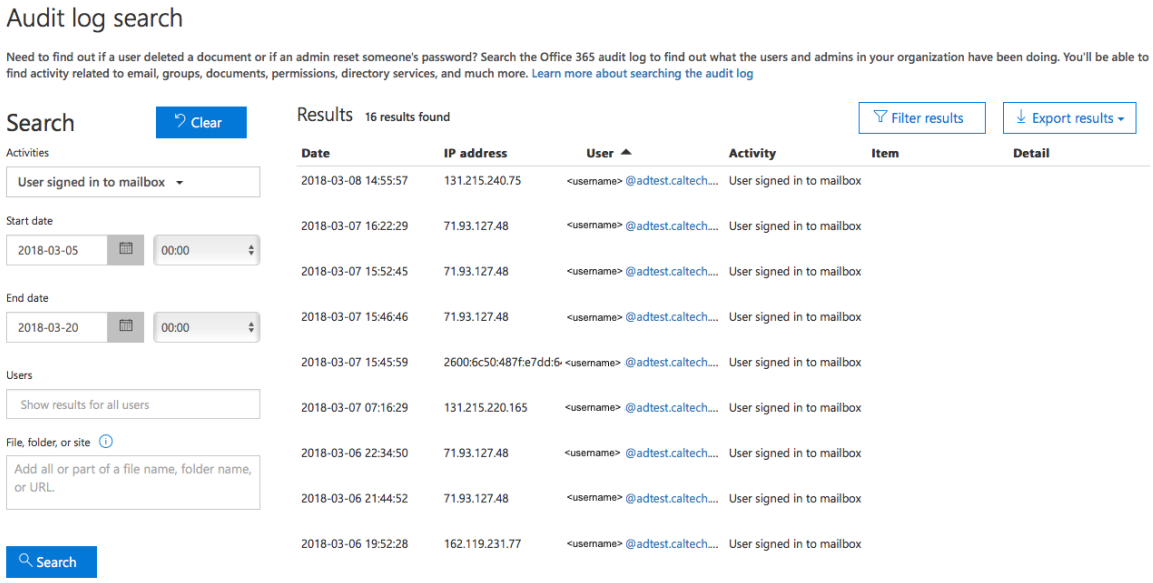
```
Set-Mailbox -AuditEnabled $true -AuditOwner MailboxLogin
```

The full list of auditable mailbox owner activities is: None, Create, HardDelete. MailboxLogin, Move, MoveToDeletedItems, SoftDelete, Update, UpdateFolderPermissions (cloud-based service only) (Microsoft Corporation, 2017).

© 2018 The SANS Institute Author retains full rights.

Discovery and analysis of sign-in information is available in the Office 365 browser-based administrative console. In the Security and Compliance Admin Center under Search & Investigation, an Audit Log search can be accessed. Here, searches based on the activity "User signed in to mailbox" can be performed based on date, time, and username. Alerts can be created based on specific user accounts that may sign into a mailbox. However, there is no facility to alert on a specific IP address or geographic location that signs into a mailbox. There is also no facility for finding or alerting on other types of sign-in activity other than mailbox sign-ins. Sign-ins for SharePoint, PowerBI, OneDrive, etc. are missing from this interface. Figure 1 shows typical log search results.

Figure 3 - Audit Log Search



It is possible to export data in comma-separated values (CSV) which permits easy sorting and searching. There are four attributes exported: CreationData, UserIDs, Operations, and Audit Data. Audit data contains all the details of the activity. Table 1 below shows the typical data detail gathered from a specific entry. In the CSV file, the attribute and value are separated by a colon (:) and each attribute/value pair is separated by a comma (,). The attribute/value pair is further sub-divided into Actor, ExtendedProperties, and Target attributes.

© 2018 The SANS Institute Author retains full rights.

Table 1 – Data detail

Details	
Date	03/7/2018 16:22
IP address	71.93.127.48
User	<username>@adtest.domain.com
Activity	User signed in to mailbox
Item	
Detail	
Id	1f3b13e3-6613-4b07-573f-0123456xxxx7
Logon Type	0
Mailbox Guid	9b93e132-4bc9-4c08-a839-013456xxxx89
Mailbox Owner UPN	<username>@adtest.domain.com
Mailbox Owner Sid	S-1-5-21-3539983850-1976339075-1450546321-xxxxxx
Logon User Sid	S-1-5-21-3539983850-1976339075-1450546321-xxxxxx
Record Type	2
External Access	FALSE
Client Info String	Client=POP3/IMAP4;Protocol=IMAP4
More information	
ClientIPAddress	71.93.127.48
ClientInfoString	Client=POP3/IMAP4;Protocol=IMAP4
CreationTime	2018-03-08T00:22:29
ExternalAccess	FALSE
Id	1f3b13e3-6613-4b07-573f-0123456xxxx7

© 2018 The SANS Institute Author retains full rights.

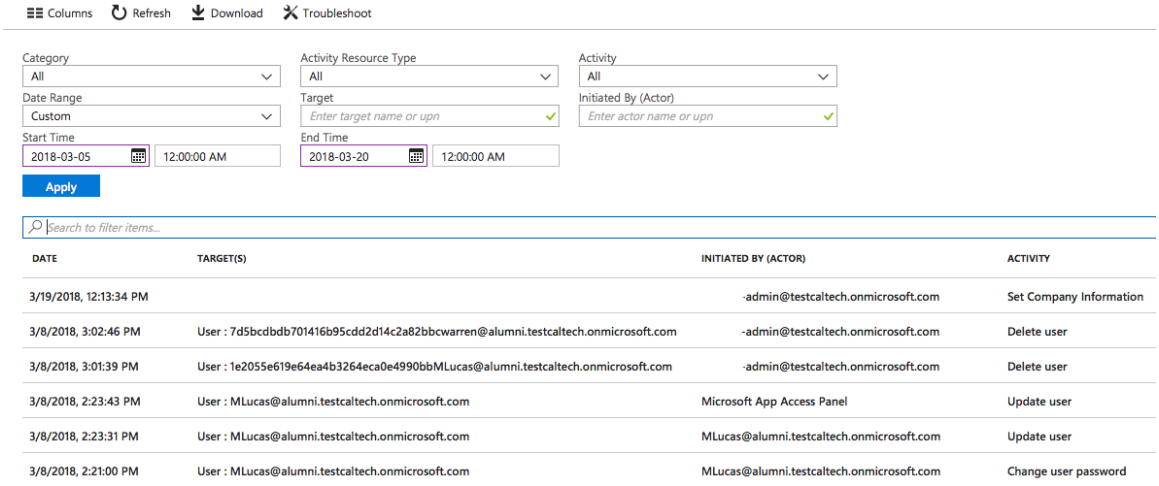
InternalLogonType	0
LogonType	0
LogonUserSid	S-1-5-21-3539983850-1976339075-1450546321-xxxxxx
MailboxGuid	9b93e132-4bc9-4c08-a839-013456xxxx89
MailboxOwnerSid	S-1-5-21-3539983850-1976339075-1450546321-xxxxxx
MailboxOwnerUPN	<username>@adtest.domain.com
Operation	MailboxLogin
OrganizationId	21248082-716c-4550-b30b-1234567zz890
OrganizationName	adtest.domain.com
OriginatingServer	
RecordType	2
ResultStatus	Succeeded
UserId	<username>@adtest.domain.com
UserKey	0123FFFF456FFF78
UserType	0
Version	1
Workload	Exchange

There is inconsistent terminology when comparing the "Activity" data (User signed in to mailbox) with the "Operation" data (MailboxLogin). There are two sign-in type attributes. One is "InternalLogonType" which is reserved for Microsoft use, while customer accessible documentation exists for the "LogonType" value (Microsoft Support, 2018). It is recommended, based on this information, that care be taken to search for the correct term under the correct attribute. Failure to do so will result in incorrect results which could lead to false conclusions.

© 2018 The SANS Institute Author retains full rights.

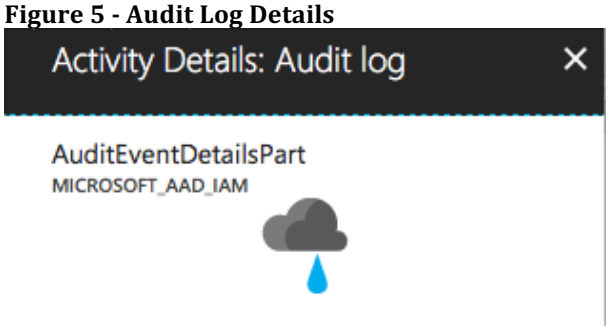
In the Azure Active Directory Admin Center, another version of Audit Logs is available. Despite Activity Resource types of "Authentication" and "User", there is no way to search for sign-ins. Because sign-in data is available in other locations, this is not a major issue; however, system administrators with this awareness will increase their efficiently by not unnecessarily clicking into this log when desiring to review sign-in data. At the time of this writing, clicking the "Download" button produces an error with no logs downloaded. Figure 4 shows typical log entries.

Figure 4 - Audit Logs



At the time of this writing, clicking on a specific event for details brought up a screen with no further information (Figure 5).

Figure 5 - Audit Log Details

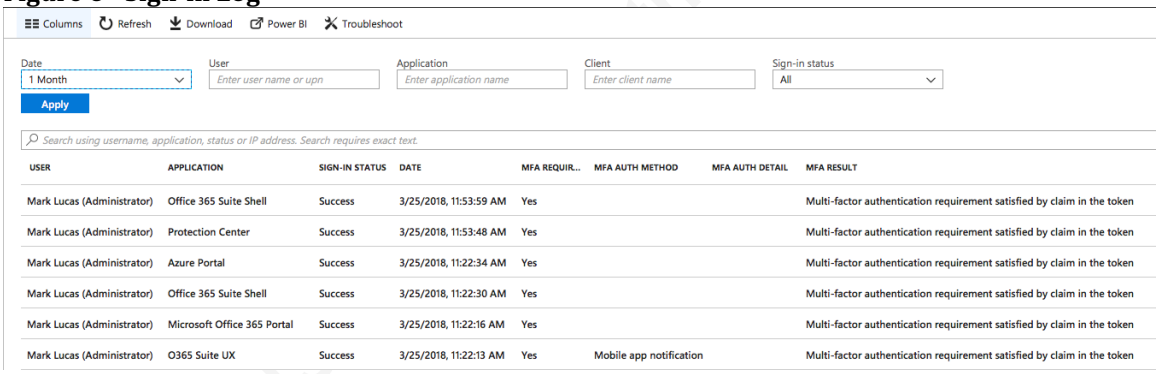


To view all sign-ins, system administrators can review the activity log of Sign-ins in the Azure Active Directory Admin Center. While valuable for specific searches, neither the sign-in log nor the activity log permits the creation of alerts as in the Audit log

© 2018 The SANS Institute Author retains full rights.

in the Security and Compliance Administrative Center. Sign-in detail of each authentication event is listed in Figure 6 below. In this example, the Administrator signed into Office 365 and had the Office Admin Center set as the startup location. The Azure Active Directory Administration Center was then opened, followed by the Security and Compliance Center. Each of these actions spawned a new browser tab. All authentication happened behind the scenes without user interaction despite the multiple log entries, while the audit log recorded each sign-in.

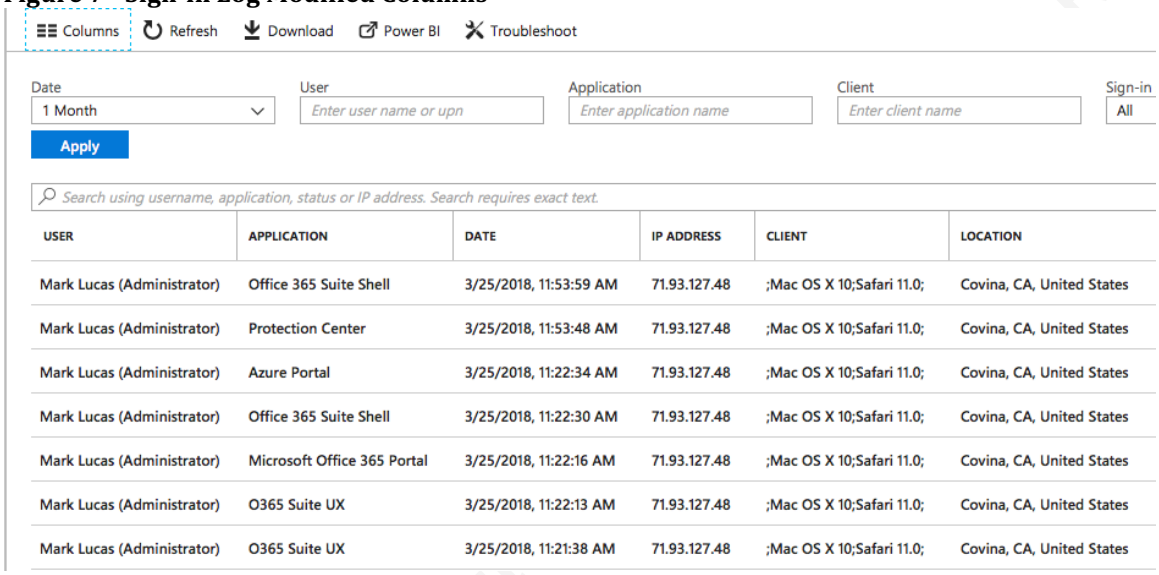
Figure 6 - Sign-in Log



The columns shown in the screenshot are the default columns. Other available columns are Date (UTC), IP Address, Client, Username, Location. Changing the view provides more information concerning sign-ins as Figure 7 shows. The location attribute is not entirely accurate. While Covina, CA, United States is within three miles of the actual location, this does not reflect the city of login. Depending on the incident investigation, this may be important. If the incident is extremely sensitive, the logs could be called into consideration in a court of law and the notation of an incorrect city may give credence to the opposition's case. System administrators might consider consulting with the legal teams and provide more exact location information to corroborate other evidence or testimony. This may require requests from service providers, which usually takes more time and may require a subpoena. These extra activities will require modifications in work schedules that managers will need to consider. If such inaccuracies are seen on a regular basis, system administrators and managers are advised to create contingency plans before an actual incident occurs.

© 2018 The SANS Institute Author retains full rights.

Figure 7 - Sign-in Log Modified Columns



While attempting to download the data, an error occurred stating, “Server Error in '/' Application. Access is denied.” This may have been a temporary service malfunction or there may be additional permissions needed to download data verses view data. Further work is needed in this area to determine the exact cause.

Azure Active Directory Admin Center provides automated alerting and mitigation of “Risky sign-ins”. All editions of Azure Active Directory afford some level of automated alerting and analysis. Paid versions Premium 1 and 2 provide extended levels of protection (Vilcinskas, Karlsson, Tilman, & Love, 2017). Six types of risk can be detected:

- Users with leaked credentials
- Sign-ins from anonymous IP addresses
- Impossible travel to atypical locations
- Sign-ins from infected devices
- Sign-ins from IP addresses with suspicious activity
- Sign-ins from unfamiliar locations

© 2018 The SANS Institute Author retains full rights.

Reporting in near-real-time is available for sign-ins from anonymous IP addresses and unfamiliar locations. These reports have a latency of five to ten minutes. All other activity is reported offline with a latency of two to four hours (Vilcinskas, Merger, Karlsson, Tillman, & Cristofor, 2017). It is worth noting several definitions of the types of risk reported from Vilcinskas et.al.:

"Impossible travel to atypical locations":

> *This risk event type identifies two sign-ins originating from geographically distant locations, where at least one of the locations may also be atypical for the user, given past behavior. Among several other factors, this machine learning algorithm takes into account the time between the two sign-ins and the time it would have taken for the user to travel from the first location to the second, indicating that a different user is using the same credentials.*

> *The algorithm ignores obvious "false positives" contributing to the impossible travel conditions, such as VPNs and locations regularly used by other users in the organization. The system has an initial learning period of 14 days during which it learns a new user's sign-in behavior.*

Impossible travel monitoring is extremely valuable because many malicious actors compromise accounts at some distance from the account owner. Chen, Ji, and Barford (2008) show that more than 80% of the malicious activity is from 20% of the total IPv4 address space which makes it likely that the malicious actor is not geographically close to the owner. The 2018 Verizon DBIR Report shows that only 15% of all incidents are due to insider threats (Verizon, 2017). Thus, watching for outside locations that access inside mailboxes will protect the organization from a majority of threats.

"Sign-in from unfamiliar locations":

> *This risk event type considers past sign-in locations (IP, Latitude / Longitude and ASN) to determine new / unfamiliar locations. The system stores information about previous locations used by a user, and considers these "familiar" locations. The risk event is triggered when the sign-in occurs from a location that's not already in the list of familiar locations. The system has an initial learning period of 30 days, during which it does not flag any new locations as unfamiliar*

© 2018 The SANS Institute Author retains full rights.

> *locations. The system also ignores sign-ins from familiar devices, and locations that are geographically close to a familiar location.*

"Leaked credentials":

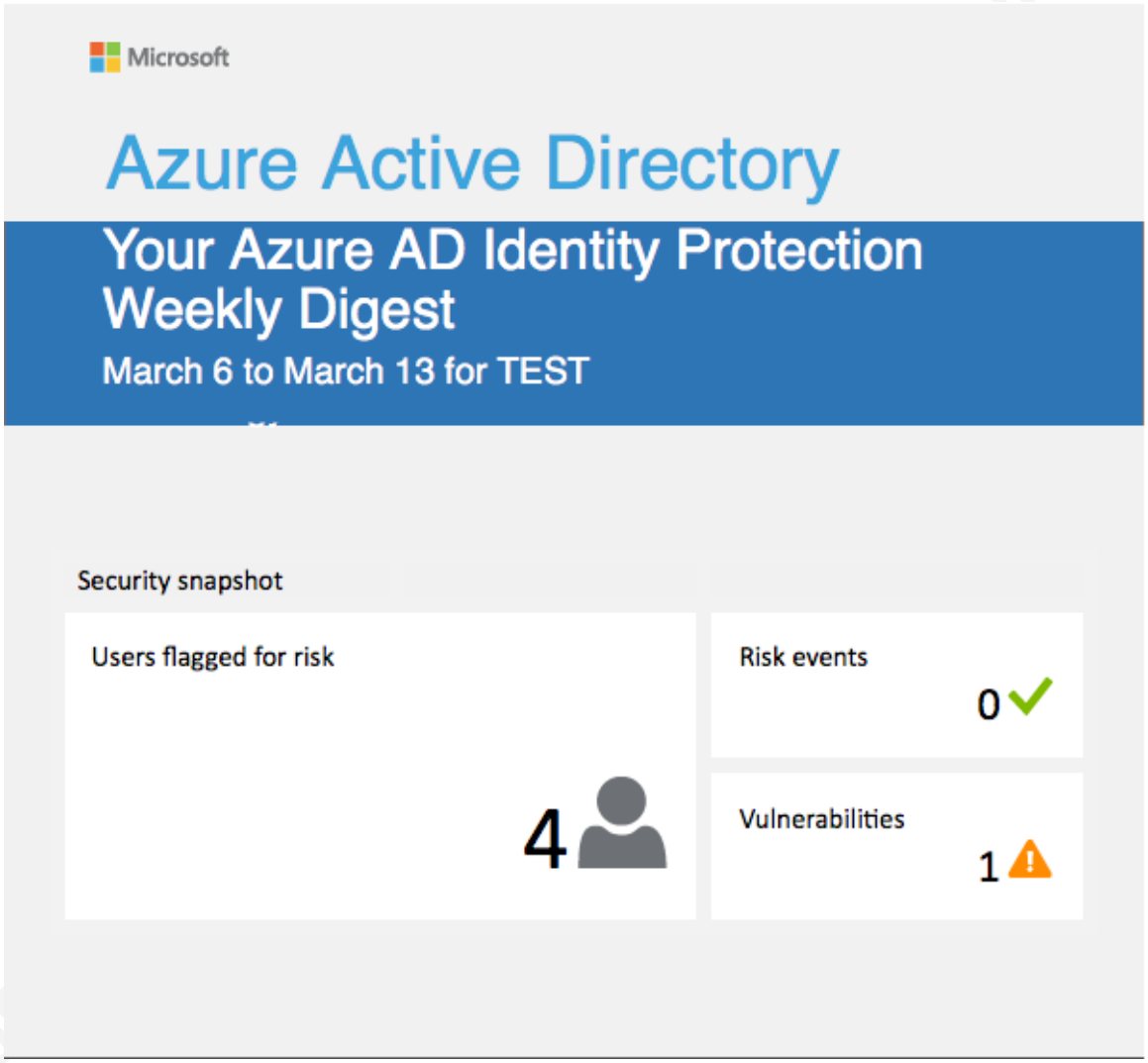
> *When cybercriminals compromise valid passwords of legitimate users, the criminals often share those credentials. This is usually done by posting them publicly on the dark web or paste sites or by trading or selling the credentials on the black market. The Microsoft leaked credentials service acquires username / password pairs by monitoring public and dark web sites and by working with:*
>
> - *Researchers*
> - *Law enforcement*
> - *Security teams at Microsoft*
> - *Other trusted sources*
>
> *When the service acquires username / password pairs, they are checked against AAD users' current valid credentials. When a match is found, it means that a user's password has been compromised, and a leaked credentials risk event is created.*

The check between exposed passwords and current valid credentials raises questions beyond the scope of this paper. Future research should address the issue as to whether Microsoft can read plain-text passwords or if password hashes can be compared. Additionally, consider what security measures surround the algorithms and systems making the comparison. If malicious actors were able to access these systems, it would provide access to not only the passwords, but access to which passwords are valid. Future research could consider how is this implemented for organizations using Azure AD Connect (Mathers, Karlsson, Tillman, & de Jong, 2018) when passwords are stored exclusively in locally controlled Active Directory.

Events reporting occurs on a weekly basis via email to the designated administrators of the online tenant. Figure 8 shows the Identity Protection Weekly Digest email. This email summarizes the events and accounts that may be causing risk to your organization. System administrators must log into the Office 365 Portal to gather in-depth knowledge of the summary report. From the Portal, they can mitigate the risks or acknowledge the alert as a non-risk.

© 2018 The SANS Institute Author retains full rights.

Figure 8 – Identity Protection Weekly Digest Email



Reports are also available in real-time on the Azure Active Directory Admin Center Risky sign-ins report page, as seen below in Figure 9:

Figure 9 - Sign-ins Report Page

Entering the "Sign-ins from anonymous IP addresses" item displays information concerning the account, locations, and time of the event as seen below in Figure 10:

© 2018 The SANS Institute Author retains full rights.

Figure 10 - Anonymous Sign-in Detail

USER	IP	LOCATION	SIGN-IN TIME (UTC)	STATUS
Mcqueen, Tyrus C.	5.9.158.75	Berlin, Berlin, Germany	3/2/2018 5:25 PM	Closed (password ...
Mcqueen, Tyrus C.	185.220.101.10	Craven, England, United Kin...	3/2/2018 3:51 PM	Closed (password ...
Mcqueen, Tyrus C.	193.15.16.4	Stockholm, Stockholm Cou...	3/2/2018 3:28 PM	Closed (password ...
Mcqueen, Tyrus C.	176.10.99.200	Zurich, ZH, Switzerland	3/2/2018 2:58 PM	Closed (password ...

As with the previous reports, attempts to download this report results in an access denied error, even though the account downloading it is a Global Admin for the tenant.

Administrators apply a Sign-in risk remediation policy to any of these risk event types. The policy basis is:

1) Who? Which groups are covered by the remediation policy?

2) Level? Should the policy apply to low-risk events and higher, medium-risk events and higher, or only high-risk events?

3) Control? Should the account at risk be denied access, or should it be forced to use Multi-Factor Authentication (MFA) or change the password?

If this were a risky event indicating an account compromise, forcing a password change or requiring MFA would lock out the genuine account owner but would possibly permit the malicious actor access to the account. A forced password change is only possible if the account resides in Office 365 or password write back (Flores, Phal, Tillman, Love, & Merger, 2018) is enabled. Requiring a password change or the implementation of MFA after a suspicious event would likely cause a cascading series of events to protect the account once the account owner calls the Help Desk for support. However, the potential damage caused between the time of compromise and the time of the report is immense.

2.3.2. Command-line options: PowerShell

Microsoft's flagship command line interface, PowerShell permits secure access to all aspects of Azure Active Directory, Exchange Online, and other Office 365 tools including the audit logs with the enforcement of MFA (Davies, 2017). Using PowerShell, automated downloads of the logs can be accomplished by creating CSV files which can

© 2018 The SANS Institute Author retains full rights.

be analyzed and stored locally for future reference. The format of the PowerShell downloads matches those of the Security & Compliance Audit Log download with only minor changes, as seen below:

```
RunspaceId   : 7c01bfb4-9b0b-4c4f-90c4-21ff0548f0b7
RecordType   : AzureActiveDirectoryStsLogon
CreationDate : 3/7/2018 7:37:35 PM
UserIds      : mjlucas-psadmin@test.onmicrosoft.com
Operations   : UserLoggedIn
AuditData
{
        "CreationTime":"2018-02-22T23:30:30",
        "Id":"8dcb27e2-267b-460d-a52f-1aaa1a1aa2b2",
        "Operation":"UserLoggedIn",
        "OrganizationId":"21248082-716c-4550-b30b-1aaa1a1aa2b222",
        "RecordType":15,
        "ResultStatus":"Succeeded",
        "UserKey":"10000AAA00BBBB333@adtest.domain.com",
        "UserType":0,
        "Version":1,
        "Workload":"Azure ActiveDirectory",
        "ClientIP":"111.222.10.192",
        "ObjectId":"Unknown",
        "UserId":"<username>@adtest.domain.com",
        "AzureActiveDirectoryEventType":1,
        "ExtendedProperties":[
                {"Name":"UserAgent","Value":"Microsoft Office\/16.0 (Windows
                   NT 10.0; Microsoft Outlook 16.0.4639; Pro)"},
                {"Name":"UserAuthenticationMethod","Value":"1"},
                {"Name":"RequestType","Value":"OrgIdWsTrust2:process"},
                {"Name":"ResultStatusDetail","Value":"Success"}
        ],
        "Actor":[
                {"ID":"a799c7de-8bbc-4bd7-b459-1a11111a111111","Type":0},
                {"ID":"<username>@adtest.domain.com","Type":5}
                ,{"ID":"10000AAA00BBBB333","Type":3}
        ],
        "ActorContextId":"21248082-716c-4550-b30b-1aaa1a1aa2b222",
        "ActorIpAddress":"111.222.10.192",
        "InterSystemsId":"db249c02-bea6-45f3-bb2c-04ffcad0954f",
        "IntraSystemId":"4e6abbbd-b053-4573-81f7-03bab3b90500",
        "Target":[
                {"ID":"Unknown","Type":0}
        ],
        "TargetContextId":"21248082-716c-4550-b30b-1aaa1a1aa2b222",
        "ApplicationId":"bfc44fc5-2fe3-4d02-98ec-1e5967475f68"
```


© 2018 The SANS Institute Author retains full rights.

```
}
ResultIndex  : 3
ResultCount  : 6
Identity     : a6dd0ec7-88a8-4428-bd1f-1a11111a111111
IsValid      : True
ObjectState  : Unchanged
```

The delivery of nested data in JavaScript Object Notation (JSON) is particularly challenging to parse. Some attributes are of interest and others are less relevant. All attributes are defined in the document "Detailed properties in the Office 365 audit log" (Microsoft Corporation, 2018). This study concentrates on the IP address and, secondarily, on the User Agent. These two attributes in addition to knowledge of email usage patterns makes it possible to ascertain if an event is suspicious. PowerShell has a built-in function to parse JSON and extract the values from the AuditData attribute. Further extraction is done by selecting the numbered attribute value within the resulting array of ExtendedProperties.

When accessing data using PowerShell, all times are displayed in Coordinated Universal Time (UTC). Therefore, data must be normalized to ascertain the appropriate event time.

Table 2 below shows early data downloaded using PowerShell without parsing and normalization:

Table 2 – CSV data downloaded using PowerShell

Date	IPAddress	UserIds	UserAgent	Operation	CheckTime
2018-02-23 16:04:37	10.1.1.75	mfa-admin@test.onmicrosoft.com	System.Object[]	UserLoggedIn	2/25/2018 7:37:49 AM
2018-02-23 15:44:09	10.1.1.75	admin@test.onmicrosoft.com	System.Object[]	UserLoggedIn	2/25/2018 7:37:49 AM
2018-02-23 13:51:17	10.1.1.185	user2@adtest.domain.com	System.Object[]	UserLoggedIn	2/25/2018 7:37:49 AM
2018-02-23 16:04:37	10.1.1.75	mfa-admin@test.onmicrosoft.com		UserLoggedIn	2/25/2018 7:03:29 AM
2018-02-23 15:44:09	10.1.1.75	admin@test.onmicrosoft.com		UserLoggedIn	2/25/2018 7:03:29 AM

© 2018 The SANS Institute Author retains full rights.

2018-02-23 13:51:17	10.1.1.185	user2@adtest.domain.com		UserLoggedIn	2/25/2018 7:03:29 AM

The following PowerShell script was utilized to check sign-in times against the time the event appeared in the audit log. The script checks the Audit Log every minute looking for the event, and once the event appears, it writes the data to the audit log, including the current time normalized to UTC.

```
$SearchMailbox =  "<username>@adtest.domain.com"
$StartTime = "03/07/2018 00:00"
$EndTime = "03/08/2018 11:00pm"
$OutputFile = "C:\Scripts\AuditSearch $(get-date -f
yyyyMMddTHHmmss).txt"

$a = 1
Do
{
     $AuditSearchUserLoggedIn = Search-UnifiedAuditLog -UserIDs
$SearchMailbox -StartDate $StartTime -EndDate $EndTime -
Operations "UserLoggedIn","User signed in to mailbox"
     #Write progress to screen to monitor script
     $PercentComplete = $a/500 * 100
     Write-progress -Activity "Searching Unified Audit Log for
     logon events for $SearchMailbox..." -PercentComplete
     $PercentComplete

     $Results = @()

     foreach ($Entry in $AuditSearchUserLoggedIn)
     {
          #Initialize a return object
          $return = "" | select
Date,CheckTime,IPAddress,UserId,Operation,UserAgent

          #Convert the JSON to a PSObject
          $data = $Entry | Select-Object -ExpandProperty
AuditData | ConvertFrom-Json

          #Populate the return object
          $return.Date = $data.CreationTime
          #Normalize the time of audit log search to UTC
          $return.CheckTime = ((get-
date).ToUniversalTime()).ToString("yyyy-MM-ddTHH:mm:ssZ")
          $return.IPAddress = $data.ClientIP
          $return.UserId = $data.UserId
          #Obtain the UserAgent string from inside the
ExtendedProperties Array
          $return.UserAgent = $data.ExtendedProperties[0].value
```


© 2018 The SANS Institute Author retains full rights.

```
            $return.Operation = $data.Operation

            #Returns the data to outside of the loop
            $Results += $return
      }

      #Export to csv
      $Results | Export-Csv -Delimiter "," -NoTypeInformation -
Path $OutputFile -append -force
      $a = $a + 1
      start-sleep -s 60
} While ($a -le 500)
Write-Progress -CurrentOperation "SearchingUnifiedAuditLog"
("Searching Unified Audit Log for $SearchMailbox Logon....Done!")
"Audit log search ended " + (get-date) | Out-File $OutputFile -
Append
```

Table 3 shows the data extracted from the JSON formatted data utilizing both PowerShell's built in convert from JSON built-in function and references to the data enclosed in arrays within the JSON element.

Table 3 - Sample logon data from the final script

Date	CheckTime	IPAddress	UserId	Operation	UserAgent
2018-03-07 19:36:34	2018-03-07 20:11:45Z	10.10.1.48	admin@test.onmicrosoft.com	UserLoggedIn	Microsoft WinRM Client
2018-03-07 19:37:55	2018-03-07 20:12:49Z	10.10.1.48	admin@test.onmicrosoft.com	UserLoggedIn	Mozilla/4.0*
2018-03-06 23:41:12	2018-03-07 00:06:22Z	10.1.1.164	user@adtest.domain.com	UserLoggedIn	CBAInPROD
2018-03-06 19:36:41	2018-03-06 23:37:31Z	10.1.1.165	admin@test.onmicrosoft.com	UserLoggedIn	Mozilla/5.0**
2018-03-06 19:33:34	2018-03-06 19:59:17Z	10.1.1.165	user@adtest.domain.com	UserLoggedIn	CBAInPROD
2018-03-06 17:31:08	2018-03-06 19:33:51Z	10.1.1.161	user@adtest.domain.com	UserLoggedIn	CBAInPROD

* Full Listing: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 10.0; Win64; x64; Trident/7.0; .NET4.0C; .NET4.0E; .NET CLR 2.0.50727; .NET CLR 3.0.30729; .NET CLR 3.5.30729)
** Full Listing: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.36 Edge/16.16299

Microsoft WinRM Client indicates a PowerShell Login regardless of whether or not the login uses Modern Authentication (The Office Team, 2017) or Basic Authentication. The Office 365 service connection does not affect the UserAgent value; all PowerShell logins have the same UserAgent string. The web browsers are very

© 2018 The SANS Institute Author retains full rights.

unclear. As shown above in the full listings starting "Mozilla/4.0" and "Mozilla/5.0", multiple browsers are listed. Using the Microsoft Edge produced the Mozilla/4.0 listing and using Safari on Mac High Sierra produced the Mozilla/5.0 listing. Neither listing is clear as to what browser was actually connected. Microsoft Edge isn't listed in the Mozilla 4.0 record but MSIE (Microsoft Internet Explorer) is and while Safari is listed in the second record, so is Chrome and Edge. Thunderbird on Windows and Mac returned the CBAInPROD listing. Clarifications for PowerShell logins are available within the web-based log; however, the web browser sign-ins are still vague.

There was no consistency regarding the time it took the data to be available for PowerShell download. The data appeared in as little as 25 minutes in one case and as long as 4:00 hours in a second with an average time of 1:20 hours and a median of 35:03 minutes. Further research and testing are required to determine if the longer times are anomalies or they are frequent enough to be of concern.

3. Conclusion

Microsoft offers multiple tools to monitor and mitigate account usage and possible compromise. When specific accounts are at risk and require monitoring, based on this case study, it is recommended that system administrators utilize Search and Investigation under the Security and Compliance Admin Center. In all cases, manually enable auditing on accounts via PowerShell to generate any alerts or logs of value. System Administrators can consider creating alerts for accounts then run reports based on sign-ins from specific at-risk accounts.

When searching for activities other than sign-ins, system administrators can consult the Audit Logs under Azure Active Directory Admin Center. These logs are extremely valuable when monitoring administrator privilege use and determining who took what action when. The logs are also useful to determine actions taken by mailbox delegates.

Microsoft alerting for sign-ins from anonymous IP addresses and unfamiliar locations is very valuable with a timely response of 5 to 10 minutes. If Microsoft could

© 2018 The SANS Institute Author retains full rights.

reduce alert latency for other activities from 2 to 4 hours to less than 30 minutes could make them worthwhile to enable. A blind 2-hour window is far too long, and significant damage can occur before taking mitigation steps.

If the account is sensitive such as an employee at the C-Suite level, System Administrator level, or personnel who regularly handle personally identifiable information (PII), then it is worth considering an automated account lockout. Account owner education is recommended when this is implemented to reduce or eliminate concern if the lockout is triggered. With proper training, the account owner will securely regain access to their account.

PowerShell dumps of the login data to local servers for offline analysis might be valuable in lieu of reliance on Microsoft algorithms if the reporting time can be proven to be consistently in the 30 minute and under availability time frame. Additionally, for system administrators to effectively utilize the data, the security team must know the usual workflow and geographic location of high-value employees.

Based on this case study, it is recommended that care be taken to search for the correct values in the correct attributes. Attributes have similar names and can easily be confused. PowerShell downloads are probably most valuable for smaller organizations with lower IT budgets and greater communication with employees. Larger organizations would do well to consider commercial products or Microsoft's premium tier products. In neither case (commercial products or premium tier) is the latency problem circumvented. Microsoft must still address this shortcoming by product development.

Most incidents will require researching at least two of the four reporting locations for complete data. This lack of a "single pane of glass" is somewhat mitigated by PowerShell searches, but there is still information that is only available via the web page such as the exact PowerShell service connection (Exchange, SharePoint, PowerBI).

Proper auditing and alerting will mitigate the damage of compromised accounts, and Microsoft has provided tools to assist in this challenge. The tools are not yet mature, but they show promise to combat malicious actors.

Mark J. Lucas, mjlucas62@mac.com

© 2018 The SANS Institute Author retains full rights.

References

Amazon Web Services. (2018). *Elastic Load Balancing documentation.* Retrieved February 03, 2018, from Amazon Web Services: https://aws.amazon.com/documentation/elastic-load-balancing/

Amazon Web Services. (2018). *Elastic Load Balancing features.* Retrieved February 3, 2018, from Amazon Web Services: https://aws.amazon.com/elasticloadbalancing/details/#compare

Amazon Web Services. (2018). *What is AWS? - Amazon Web Services.* Retrieved February 3, 2018, from Amazon Web Services: https://aws.amazon.com/what-is-aws/

Bahall, D., Gremban, K., Tilman, M., Casey, L., & Notin, C. (2017, September 8). *Security considerations for accessing apps remotely with Azure AD Application Proxy.* Retrieved from Microsoft Azure: https://docs.microsoft.com/en-us/azure/active-directory/application-proxy-security-considerations

Chen, Z., Ji, C., & Barford, P. (2008). Spatial-Temporal Characteristics of Internet Malicious Sources. *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE.* Phoenix: IEEE. Retrieved from http://jic.ece.gatech.edu/info-sec-08.pdf

Cooper, R. (2011, February 4). *IIS and X-Forwarded-For header.* Retrieved from Loadbalancer.org: http://www.loadbalancer.org/blog/iis-and-x-forwarded-for-header/

Davies, J. (2017, April 27). *Connect to Office 365 services with multifactor authentication (MFA) and PowerShell.* Retrieved from Microsoft Technet Blogs: https://blogs.technet.microsoft.com/solutions_advisory_board/2017/04/27/connect-to-office-365-services-with-multifactor-authentication-mfa-and-powershell/

Flores, J., Phal, B., Tillman, M., Love, C., & Merger, P. (2018, January 11). *Password writeback overview.* Retrieved from Microsoft Azure: https://docs.microsoft.com/en-us/azure/active-directory/active-directory-passwords-writeback

Gregory, D. (2014, November 23). *ADFS deep dive: Planning and design considerations [Web log post].* Retrieved from https://blogs.technet.microsoft.com/askpfeplat/2014/11/23/adfs-deep-dive-planning-and-design-considerations/

Gunnemo, J. (2016, September 30). *Dive into Modern Authentication - How it works and what to do when it doesn't [Video file].* Retrieved from https://www.youtube.com/watch?v=YtOufp8FN5Q

Intersect Alliance. (2018, February 03). *Snare by Intersect Alliance - Snare Agents.* Retrieved from Intersect Alliance: from https://www.intersectalliance.com/our-product/snare-agent/

IP Location. (2018). *IP location finder - Geolocation.* Retrieved February 3, 2018, from https://www.iplocation.net/

Mark J. Lucas, mjlucas62@mac.com

© 2018 The SANS Institute Author retains full rights.

Ipswitch, Inc. (2010). *Best practices: Event log management for security and compliance initiatives.* Retrieved from Ipswitch, Inc., Network Managment Division website: https://www.ipswitch.com/Ipswitch/media/Ipswitch/Documents/Resources/Whitepapers%20aan%20eBooks/ELM_Security_WP.pdf?ext=.pdf

Lee, S. C. (2003). *An introduction to identity management.* Retrieved from SANS.org: https://www.sans.org/reading-room/whitepapers/authentication/an-introduction-to-identity-management-852

Mathers, B. (2017, May 31). *Deploying Federation Server Proxies.* Retrieved from Windows IT Pro Center: https://docs.microsoft.com/en-us/windows-server/identity/ad-fs/deployment/deploying-federation-server-proxies-w2k12r2

Mathers, B., Karlsson, M., Tillman, M., & de Jong, R. (2018, March 19). *Integrate your on-premises directories with Azure Active Directory.* Retrieved from Microsoft Azure: https://docs.microsoft.com/en-us/azure/active-directory/connect/active-directory-aadconnect

Mathers, B., Kumar, S., & Plett, C. (2017, May 31). *Active Directory Federation Services.* Retrieved from Windows IT Pro Center: https://docs.microsoft.com/en-us/windows-server/identity/active-directory-federation-services

Microsoft Corporation. (2016, June 27). *Auditing and Reporting in Office 365.* Retrieved from Office 365: http://aka.ms/Office365AR

Microsoft Corporation. (2017, December 12). *Set-Mailbox.* Retrieved from Microsoft Technet: https://technet.microsoft.com/en-us/library/bb123981(v=exchg.160).aspx

Microsoft Corporation. (2018). *Detailed properties in the Office 365 audit log.* Retrieved March 26, 2018, from Office 365 Support: https://support.office.com/en-us/article/detailed-properties-in-the-office-365-audit-log-ce004100-9e7f-443e-942b-9b04098fcfc3

Microsoft Corporation. (2018). *Microsoft Trust Center | Microsoft Azure Security.* Retrieved January 28, 2018, from https://www.microsoft.com/en-us/trustcenter/security/azure-security

Microsoft Support. (2018). *Detailed properties in the Office 365 audit log.* Retrieved March 25, 2018, from Microsoft Support: https://support.office.com/en-us/article/detailed-properties-in-the-office-365-audit-log-ce004100-9e7f-443e-942b-9b04098fcfc3

Redmond, T. (2016, December 19). *How Office 365 collects and reports audit data.* Retrieved from Petri IT Knowledgebase: https://www.petri.com/office-365-audit-data

Redmond, T. (2016, January 12). *The woes of Exchange mailbox auditing [Web log post].* Retrieved from http://www.itprotoday.com/microsoft-exchange/woes-exchange-mailbox-auditing

Smith, R. F. (2018). *Windows security log event ID 4625 - An account failed to log on [Web log post].* Retrieved February 3, 2018, from https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/event.aspx?eventID=4625

Mark J. Lucas, mjlucas62@mac.com

© 2018 The SANS Institute Author retains full rights.

Sood, V. (2009, Mar 18). *Advanced logging for IIS - Custom logging.* Retrieved from Microsoft Docs: https://docs.microsoft.com/en-us/iis/extensions/advanced-logging-module/advanced-logging-for-iis-custom-logging

Splunk, Inc. (2018). *About the Splunk add-on for Microsoft Cloud Services - Splunk documentation.* Retrieved February 8, 2018, from https://docs.splunk.com/Documentation/AddOns/released/MSCloudServices/About

Splunk, Inc. (2018). *Authentication - Splunk Documentation.* Retrieved February 8, 2018, from http://docs.splunk.com/Documentation/CIM/4.9.1/User/Authentication

Swift, D. (2010, November 4). *Successful SIEM and log management strategies for audit and compliance.* Retrieved from SANS.org: https://www.sans.org/reading-room/whitepapers/auditing/successful-siem-log-management-strategies-audit-compliance-33528

The Office Team. (2017, August 1). *Updated Office 365 modern authentication.* Retrieved from Office Blogs: https://blogs.office.com/en-us/2015/11/19/updated-office-365-modern-authentication-public-preview/

Verizon. (2017). *2017 Data Breach Investigations Report.* Verizon Enterprises.

Vilcinskas, M., Karlsson, M., Tilman, M., & Love, C. (2017, November 14). *Risky sign-ins report in the Azure Active Directory portal.* Retrieved from Microsoft Azure: https://docs.microsoft.com/en-us/azure/active-directory/active-directory-reporting-security-risky-sign-ins

Vilcinskas, M., Merger, P., Karlsson, M., Tillman, M., & Cristofor, L. (2017, December 12). *Azure Active Directory risk events.* Retrieved from Microsoft Azure: https://docs.microsoft.com/en-us/azure/active-directory/active-directory-reporting-risk-events#impossible-travel-to-atypical-locations

© 2018 The SANS Institute Author retains full rights.