TLP:CLEAR

MICROSOFT EXPANDED CLOUD LOGS IMPLEMENTATION PLAYBOOK

Enabling and operationalizing cloud logs to detect and defend against advanced intrusion techniques

Publication: January 2025

Cybersecurity and Infrastructure Security Agency

Cybersecurity Division

This document is marked TLP:CLEAR. Recipients may share this information without restriction. Information is subject to standard copyright rules. For more information on the Traffic Light Protocol, see https://www.cisa.gov/tlp.

TLP:CLEAR

ACKNOWLEDGEMENTS

CISA expresses appreciation to the following parties for their contributions to the development of this playbook:

Microsoft Federal Security
Microsoft Global Hunting, Oversight, and Strategic Triage (GHOST)
Microsoft Federal Customer Success Unit (CSU)
Microsoft Purview Engineering Team
U.S. Customs and Border Protection
U.S. Department of Commerce
U.S. Department of Energy
U.S. Department of State
Lawrence Livermore National Laboratory
Marine Corps Cyberspace Operations Group
National Labor Relations Board

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1. INTRODUCTION

The Cybersecurity and Infrastructure Security Agency (CISA) developed the original version of this playbook in order to help federal civilian executive branch (FCEB) agencies' network defenders use new Microsoft logging functionality for network defense purposes. This was done in light of the prevalence of Microsoft in FCEB networks and a threat actor's compromise of Microsoft (discovered in summer 2023). Now CISA is releasing this version to the public for the benefit of all network defenders whose organizations already subscribe to the relevant Microsoft services.

In summer 2023—about a month before U.S. Secretary of State Antony Blinken was slated to travel to Beijing—a hacker group backed by the People's Republic of China (PRC) infiltrated the email accounts of senior government officials from the departments of State and Commerce. The hackers compromised the Microsoft Exchange Online cloud-based service, which granted them access to legitimate user accounts that they then used to search emails, expose state secrets, and exfiltrate data. The U.S. Department of State discovered the anomalous activity because it had purchased Microsoft's G5 license, which included enhanced logging capabilities through Microsoft Purview Audit (Premium).

As adversaries with significant funding launch targeted intrusions against the United States federal government, this type of compromise event has become increasingly common and intricate. These adversaries are often nation-state actors, setting their sights on many targets around the globe with increasing frequency and sophistication and in support of government-led espionage campaigns. The 2023 incident involving the M365 ecosystem of products targeted senior officials from the U.S. government and other foreign enterprises. Organizational impacts from identity compromise can go beyond mere compromise to include loss of confidential information and reputational damage. The defense-in-depth approach to combatting identity-based intrusion techniques includes enhanced monitoring, alerting, and user behavior analytics techniques.

Operational lessons learned from these incidents and subsequent efforts to bolster cyber defenses produced this playbook. Here, organizations will find the information necessary to address the increased frequency and sophistication of identity-based compromises by advanced cyber threat actors.

1.1 OVERVIEW

This playbook provides an overview of the newly introduced logs in Microsoft Purview Audit (Standard), which enable organizations to conduct forensic and compliance investigations by accessing critical events, such as mail items accessed, mail items sent, and user searches in SharePoint Online and Exchange Online. In addition, the playbook also discusses significant events in other M365 services such as Teams. Lastly, administration/enabling actions and ingestion of these logs to Microsoft Sentinel and Splunk Security Information and Event Management (SIEM) systems are covered in detail.

The desired outcome of this playbook is to empower enterprises seeking to operationalize these expanded cloud logs in their M365 tenant. It provides guidance on how to navigate to the logs within M365 and how to perform administration actions to enable the logs. A key outcome from the playbook is making the newly available logs an actionable part of enterprise cybersecurity operations. The analytical methodologies tied to using these logs to detect advanced threat actor behavior are covered in detail.

1.2 BACKGROUND

Incident Highlights

The Microsoft Services Account (MSA) key compromise referenced in the introduction allowed a nation-state threat actor to forge customer tokens against this MSA key and access customer Exchange Online mailbox accounts. Late in 2021, Storm-0558, a hacker group with direct ties to the PRC, obtained access to this MSA key. On June 15, 2023, the U.S. Department of State Security Operations Center (SOC) used an in-house

detection tool to flag malicious behavior that was later attributed to Storm-0558. The tool the SOC used relied on enhanced logging capabilities available only to M365 Purview Audit Premium customers at that time. Further investigation determined that hackers used the same MSA key to compromise additional federal entities.

Microsoft Internal Investigation

On June 24, 2023, Microsoft invalidated the MSA key leveraged in the intrusions. In the following weeks, Microsoft notified affected customers across the U.S. government and several organizations in the United Kingdom. To identify victim organizations and individuals in this incident, Microsoft leveraged data fields in M365's Purview Audit Premium service offering, specifically **MailItemsAccessed** events with an unexpected **ClientAppID** and **AppID**.

Although the number of victims was relatively small, the hackers targeted a very high number of high-profile organizations. In total, they compromised email accounts from 22 enterprises, including several government agencies and three think tanks. Using these logs, Microsoft identified all individuals targeted in these intrusions, including personnel from Western European, Asia-Pacific (APAC), Latin American, and Middle Eastern countries.

Microsoft's Public Response

In response to the events and to provide its customer base with the tools required to identify similar future incidents, Microsoft announced it would provide this same set of Purview Audit (Premium) logs to all customers with E3/G3 licenses and above. These logs provide new telemetry to enhance threat-hunting capabilities for business email compromise (BEC), advanced nation-state threat activities, and possible insider-risk scenarios. Microsoft has said it automatically enabled all logs (except for two covered in section 2.1.2, which still require user activation) detailed in this playbook for E3/E5 and G3/G5 customers.

The Federal Government's Response

The federal government's response, which included the release of a playbook for federal agencies in February 2024, was led by the Cybersecurity and Infrastructure Security Agency (CISA), the White House's Office of Management and Budget (OMB), and the Office of the National Cyber Director (ONCD) with substantial collaboration with Microsoft.

This public version of the playbook includes feedback on the effectiveness of the initial federal playbook from multiple federal agency and U.S. Department of Defense (DoD) customers. In addition, CISA would like to recognize Microsoft for its significant contributions to this playbook and the inherent partnership—both with Microsoft and its customers—that made this effort possible.

1.3 AUDIT LOGS CHARACTERISTICS AND SCOPE

The newly enabled enhanced audit logs in Microsoft Purview for Audit (Standard) allow organizations to monitor and analyze thousands of user and admin operations performed in dozens of Microsoft services and solutions. These operations are captured, recorded, and retained in an organization's Unified Audit Log (UAL). Enhanced audit logs also facilitate reports and insights that can help improve data security, compliance, and quality. Enhanced audit logs differ from other logs (e.g., operational or diagnostic logs) in Purview because they provide more detailed and granular information about the data governance actions and events. Enhanced audit logs are available in the Microsoft Purview portal, the UAL, and can be exported to Azure Monitor, Azure Storage, Azure Event Hubs, or other technologies for further analysis and integration.

These audit logs provide the following:

1. Expanded Accessibility – Microsoft is expanding cloud logging accessibility and flexibility. As of May 2024, all Microsoft worldwide commercial customers have access to these expanded logs.
2. Microsoft Purview Audit – This tool enables customers to centrally visualize more types of cloud log data generated across their enterprise, helping them effectively respond to security events, forensic investigations, internal investigations, and compliance obligations.
3. Increased Visibility – Microsoft customers will receive deeper visibility into security data, including detailed logs of email access and several other types of log data previously only available at the Microsoft Purview Audit (Premium) subscription level.
4. Extended Retention Period – Microsoft increased the default retention period for Audit Standard customers from 90 days to 180 days.
5. Audit and Diagnostics – Microsoft Purview allows administrators to monitor audit and diagnostics logs captured from applications in the Microsoft Purview portal.

Microsoft began rolling out these changes, which are part of Microsoft's ongoing efforts to increase access to Microsoft Purview audit logging, to customers worldwide in September 2023. The rollout was completed in June 2024.

This playbook focuses on Microsoft Purview logs for the following workloads:

- Microsoft Exchange
- Microsoft SharePoint
- Microsoft Teams

The playbook also provides example scenarios that detail how to leverage the logs for both proactive and reactive investigations driven by specific threat-actor behavior. The logs detail access tracking (e.g., who, what, where, when) that can be correlated with other event data to support threat hunting and incident-response scenarios.

This playbook does not suggest alerting threshold values as these will be specific to each enterprise environment. Example scenarios are for informational purposes and are intended to show the value an enterprise can expect to receive by enabling and leveraging these logs in their Microsoft tenant.

1.4 AUDIENCE AND RELEASE TIMELINE

The playbook is specifically written for use by technical personnel responsible for log collection, aggregation, correlation, and incident-response orchestration at government agencies and enterprises with Microsoft E3/G3-and-above licensing. This release includes clients in all Microsoft identity boundaries. Previously, these logs were only available to Audit Premium subscription customers and were released first to DoD and federal agencies to protect U.S. national security interests.

1.5 PLAYBOOK FEEDBACK

To provide feedback on the playbook or to request additional information, please contact CISA's Federal Enterprise Improvement Team (FEIT) at CISA-FEIT@cisa.dhs.gov.

2. TECHNICAL GUIDANCE

2.1 GETTING STARTED

This section provides end users with prerequisite knowledge, administrative steps, and other nuances needed to fully leverage all expanded logging capabilities.

2.1.1 Permissions and Navigation

Permissions

To fully utilize the capabilities described in this playbook, users must possess the appropriate Entra ID roles. Reference Table 1 below to determine what role is necessary for accessing resources discussed throughout this playbook.

Viewing the logs described in this playbook via the Microsoft Purview portal requires that users be given specific roles that allow them access to Compliance Data from within Compliance Permissions (https://compliance.microsoft.com/permissions). Only a Global Administrator or a user with a *Role Management* role can access and edit these permissions. The roles contained here offer access to a wide array of functionality within the Compliance suite of tools. To search, view, and export the audit logs discussed in this playbook, the *Audit Reader* role will allow this functionality. To see more information regarding roles and permissions, click on the following resources:

1. https://learn.microsoft.com/en-us/defender-office-365/scc-permissions
2. https://learn.microsoft.com/en-us/purview/purview-compliance-portal-permissions

Note: When setting up roles and permissions, adhere to the principle of least privilege whenever possible.

Table 1: Entra ID Role(s) by Resource Type

Location	Type	Entra ID Role(s)
Microsoft Defender Portal (https://security.microsoft.com)	Investigation / viewing	Security Reader (Entra ID)
Microsoft Purview Portal (https://compliance.microsoft.com)	Investigation / viewing	Audit Reader (Purview Permissions) Or Compliance Administrator (Entra ID)
Exchange Online PowerShell	Investigation / viewing	Exchange Administrator (Entra ID) or Global Administrator (Entra ID)
Exchange Online PowerShell	Edit audit settings	Exchange Administrator (Entra ID) or Global Administrator (Entra ID)

Accessing the Microsoft Defender Portal and Advanced Hunting

The Microsoft Defender portal is used for investigating data from across the Microsoft security ecosystem. Specifically, the **CloudAppEvents** table is used in Section 2.3 for querying the log events described in this playbook.

1) To access the Microsoft Defender portal, navigate to https://security.microsoft.com.
2) In the Menu, select **Hunting** -> **Advanced Hunting.**

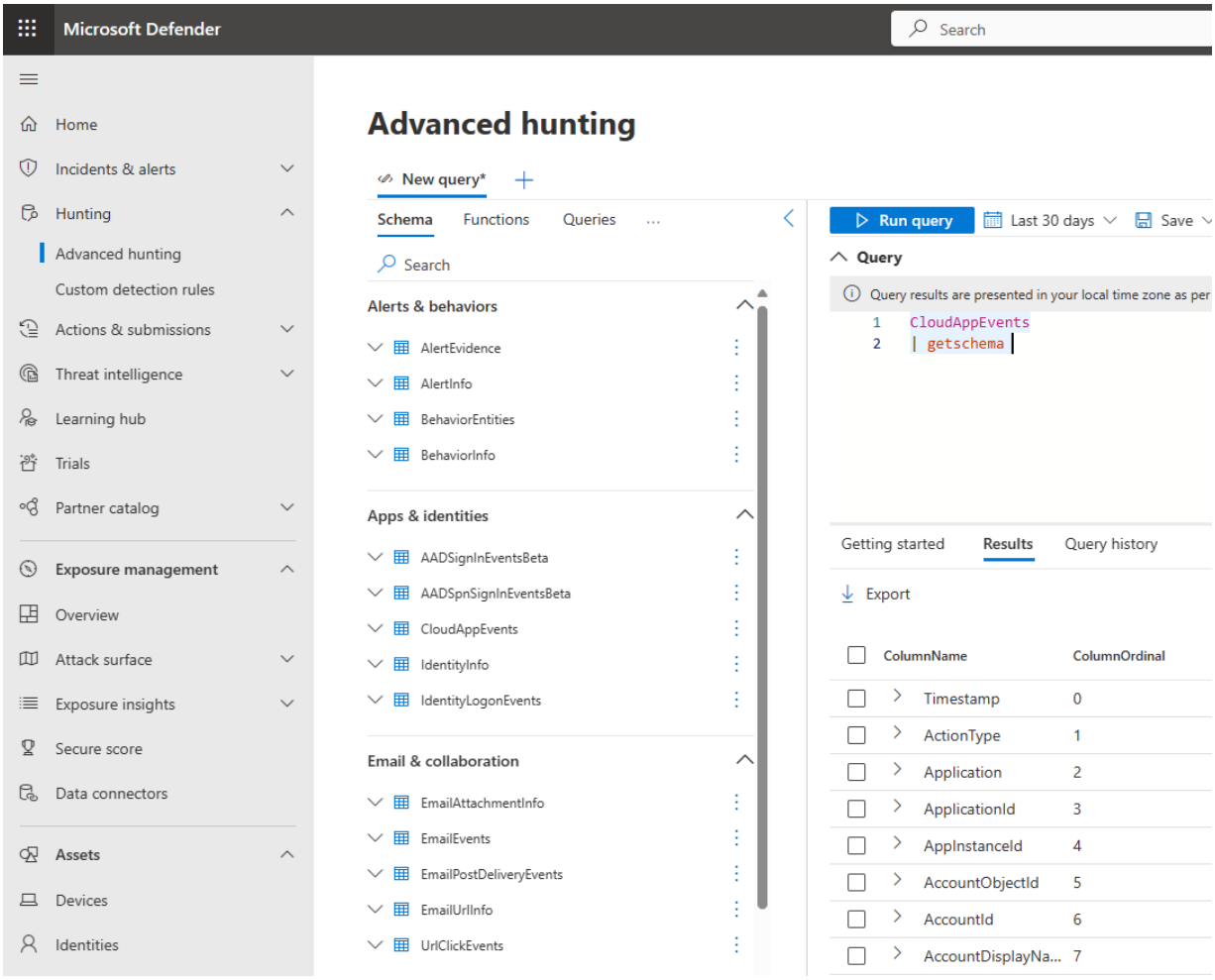


Figure 1: Microsoft Defender Portal

Accessing Microsoft Purview Portal and Purview Audit

The Microsoft Purview portal can be used for compliance, eDiscovery, and Data Loss Prevention (DLP) investigations. For this playbook, the Audit portion of the Microsoft Purview portal helps ensure expanded logging capabilities are enabled and functioning properly for a given user.

1) Navigate to https://compliance.microsoft.com.
2) In the menu select **Audit.**

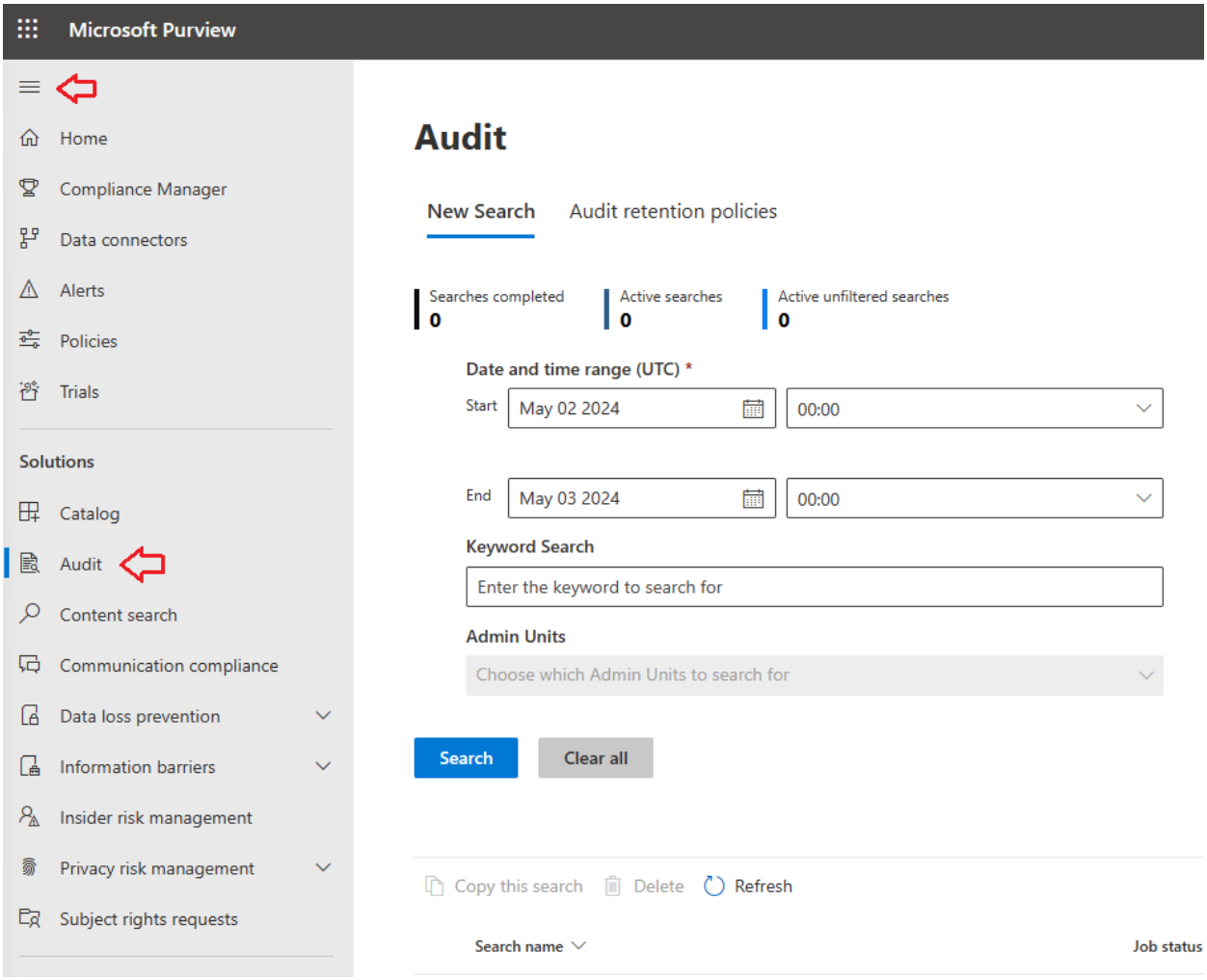


Figure 2: Microsoft Purview Portal

2.1.2 Enabling the New Logs

Default enablement is defined at a license level. For example, Auditing (Standard or Premium) is enabled by default for E3/E5/G3/G5 licenses. Some licenses, such as M365 Business Basic, M365 Business Standard, M365 Business Premium, and trial license accounts do provide access to Audit but do not currently have auditing enabled by default. These licenses will have Audit enabled by default in the future. If you are leveraging one of these license types, the steps below can be utilized to ensure that all audit features are enabled.

Note: Tampering with or disabling logging capabilities is a well-documented threat actor behavior. Given this fact, there may be scenarios where investigators wish to verify that logging is still enabled or may want to determine if any settings have been altered. Reference the commands that follow to determine audit log settings for a given user in your environment.

Useful terms:

- **DefaultAuditSet**: The value of this property indicates whether the default mailbox actions (managed by Microsoft) are being audited on the mailbox. When mailboxes are in the **DefaultAuditSet,** any new or added mailbox actions will be audited by default as they are released.
- Sign-in types
    - Owner – Mailbox owner (the account that is associated with the mailbox).
    - Delegate – A user who has been assigned **SendAs**, **SendOnBehalf**, or **FullAccess** permissions to another mailbox; or an Admin who has been assigned the **FullAccess** permission to a user's mailbox.
    - Admin – The mailbox is searched with one of the following Microsoft eDiscovery tools:
        - eDiscovery in the Microsoft Purview portal.
        - In-place eDiscovery in Exchange Online.
        - Mailbox accessed using Microsoft Exchange Server MAPI editor.
        - Mailbox accessed by an account impersonating another user. This occurs when the **ApplicationImpersonation** role is assigned to an account, such as an application, that is currently actively accessing the data.

Additional resources:

- https://learn.microsoft.com/en-us/purview/audit-solutions-overview
- https://learn.microsoft.com/en-us/purview/audit-mailboxes
- https://learn.microsoft.com/en-us/purview/audit-get-started#step-3-set-up-audit-premium-for-users

Enabling or Verifying Enablement

Users of specific license types (described above) may still be required to manually enable *Microsoft 365 Advanced Auditing* within the license properties. This can be done within Entra ID or the M365 Admin Center. Users will also need to enable **SearchQueryInitiated** logging for both Exchange and SharePoint since it is disabled by default.

1. Verify Microsoft 365 Advanced Auditing is enabled via a. M365 Admin Center or b. Entra ID.
    a. Verify Microsoft 365 Advanced Auditing is enabled via M365 Admin Center (https://admin.cloud.microsoft/).
        i. In the Microsoft 365 Admin Center, go to Users > Active users and select a user.
        ii. On the user properties flyout page, select Licenses and Apps.
        iii. In the Licenses section, verify that the user is assigned an E5 license or is assigned an appropriate add-on license. For a list of licenses that support Audit (Premium), see Audit licensing requirements.
        iv. Expand the Apps section and verify that the Microsoft 365 Advanced Auditing checkbox is selected.
        v. If the checkbox isn't selected, select it and then select Save.

OR

    b. Verify Microsoft 365 Advanced Auditing is enabled via Entra ID.
        i. In the Azure portal (https://portal.azure.com), go to Entra ID.
        ii. In the Manage section, go to Users, and select a user by clicking on the name.
        iii. In the Overview tab, select Assigned Licenses by clicking on the number to the right.
        iv. Select your license level (e.g., Microsoft 365 E5).
        v. If the slide bar is set to off, set it to on and then select Save.

2. Verify specified user is in **DefaultAuditSet** via Exchange Online PowerShell.

    a. Execute the command.

```
Get-Mailbox <identity> | FL *audit*
```

    b. The **DefaultAuditSet** property is returned by the **Get-Mailbox** cmdlet; a mailbox using the defaults will show the following result for all three sign-in types. This is important because the default audit set is managed by Microsoft and any new logs added in the future will be automatically enabled.

```
AuditEnabled     : True
AuditLogAgeLimit : 90.00:00:00
AuditAdmin       : {Update, MoveToDeletedItems, SoftDelete, HardDelete...}
AuditDelegate    : {Update, MoveToDeletedItems, SoftDelete, HardDelete...}
AuditOwner       : {Update, MoveToDeletedItems, SoftDelete, HardDelete...}
DefaultAuditSet  : {Admin, Delegate, Owner}
```

3. Verify audit actions for a specified user via Exchange Online PowerShell.

    a. Execute the command below for all three sign-in types to verify enabled mailbox actions:

```
Get-Mailbox <identity> | Select-Object -ExpandProperty <sign-in
type>
```

    b. The command output shown below for AuditOwner indicates **SearchQueryInitiated** mailbox action is not enabled**.** Note that **SearchQueryInitiated** logging is only available for **AuditOwner**; it cannot be enabled for **Admin** or **Delegate.**

```
Update
MoveToDeletedItems
SoftDelete
HardDelete
UpdateFolderPermissions
UpdateInboxRules
UpdateCalendarDelegation
ApplyRecord
MailItemsAccessed
Send
```

4. Enable **SearchQueryInitiated** mailbox action for specified user via Exchange Online PowerShell.

    a. Execute the command below to enable **SearchQueryInitiated** logging for **AuditOwner.**

```
Set-Mailbox <identity> -<sign-in type> @{Add="SearchQueryInitiated"}
```

    b. Note: Enabling **SearchQueryInitiated** mailbox action will remove the sign-in type from the **DefaultAuditSet** for the specified sign-in type.

5. Repeat step #3 above to ensure that **SearchQueryInitiated,** which was added in step #4, now appears in the mailbox actions for **AuditOwner**.

2.1.3 Verify Logs Are Flowing

In addition to checking the mailbox actions assigned to each sign-in type for each user (described in Section 2.1.2), two basic methods can be used to verify that logs are flowing for a given user. This will ensure they are operational.

1. Verify **MailItemsAccessed** logs are being generated for a specified user via Exchange Online PowerShell.

```
Search-UnifiedAuditLog -StartDate "M/DD/YYYY HH:MM AM/PM" -EndDate
"M/DD/YYYY HH:MM AM/PM" -Operation MailItemsAccessed
```

   a. The command output shown below indicates a single **MailItemsAccessed** record with the **AuditData** field truncated.

```
RecordType    : ExchangeItemAggregated
CreationDate  : 5/28/2024 3:18:02 PM
UserIds       : <username>.onmicrosoft.com
Operations    : MailItemsAccessed
AuditData     :
```

   b. Note**:** This command can be repeated with other operations discussed in this playbook (e.g., **SearchQueryInitiatedExchange**, **SearchQueryInitiatedSharePoint, Send**).

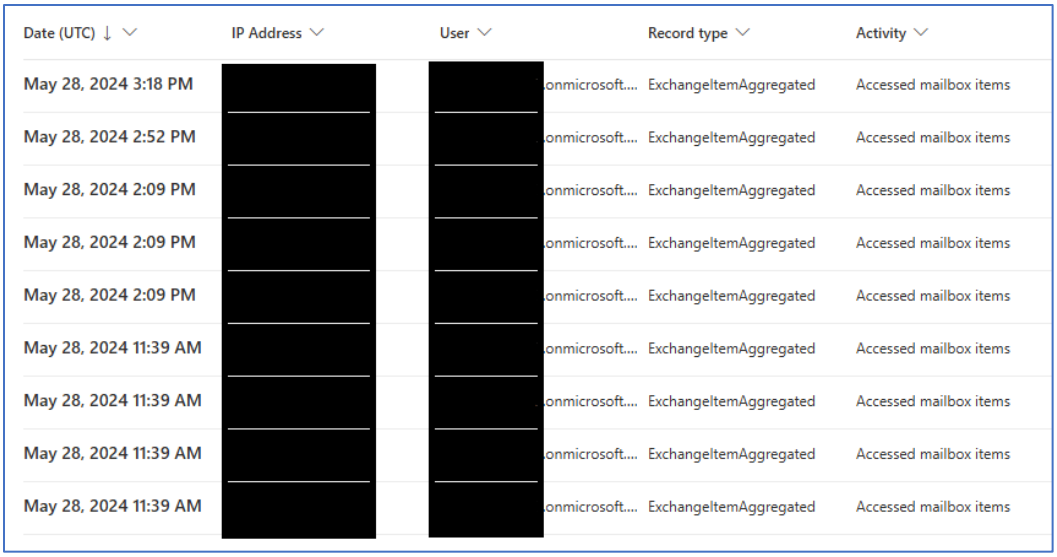
2. Verify **MailItemsAccessed** logs are being generated for a specified user via the Microsoft Purview portal.

   a. In the Microsoft Purview portal (https://compliance.microsoft.com), go to the Audit in the Solutions section.

   b. In the New Search tab, specify the following parameters:
      i. Start and end date (if testing for log generation, pick a small window of time to increase query efficiency).
      ii. Activities – Friendly names– select "Accessed mailbox items."
      iii. Users – add a desired user to check for **MailItemsAccessed** logs generation.

   c. Figure 3 shows the output of a successful query. Note: Each item can be expanded to view the details of each logged event (i.e., the AuditData field).

   d. Note**:** The Microsoft Purview portal can also be used to search the other operations discussed in this playbook (e.g., **SearchQueryInitiatedExchange**, **SearchQueryInitiatedSharePoint**, **Send**).



Date (UTC)	IP Address	User	Record type	Activity
May 28, 2024 3:18 PM	[redacted]	[redacted]	ExchangeItemAggregated	Accessed mailbox items
May 28, 2024 2:52 PM	[redacted]	[redacted]	ExchangeItemAggregated	Accessed mailbox items
May 28, 2024 2:09 PM	[redacted]	[redacted]	ExchangeItemAggregated	Accessed mailbox items
May 28, 2024 2:09 PM	[redacted]	[redacted]	ExchangeItemAggregated	Accessed mailbox items
May 28, 2024 2:09 PM	[redacted]	[redacted]	ExchangeItemAggregated	Accessed mailbox items
May 28, 2024 11:39 AM	[redacted]	[redacted]	ExchangeItemAggregated	Accessed mailbox items
May 28, 2024 11:39 AM	[redacted]	[redacted]	ExchangeItemAggregated	Accessed mailbox items
May 28, 2024 11:39 AM	[redacted]	[redacted]	ExchangeItemAggregated	Accessed mailbox items
May 28, 2024 11:39 AM	[redacted]	[redacted]	ExchangeItemAggregated	Accessed mailbox items

Figure 3: Results of MailItemsAccessed Search

2.2 SIEM INTEGRATION

2.2.1 Microsoft Sentinel

See Appendix C, "Enabling Logs in Microsoft Sentinel."

2.2.2 Splunk

See Appendix D, "Integrating Splunk and Microsoft Office 365."

See Appendix E, "Integrating Splunk With Azure & Sentinel."

2.3 OVERVIEW OF THE LOGS

This section provides a basic description of the newly added logs with significant forensic relevance, including what information they contain and an overview of their investigative value. The logs now included in Audit (Standard) will help organizations conduct forensic and compliance investigations by providing access to important events such as when mail items were accessed, when mail items were replied to and forwarded, and when and what a user searched for in Exchange Online and SharePoint Online. These events can help organizations investigate possible intrusions and determine the scope of a given compromise. In addition to these events in Exchange and SharePoint, Teams, and other M365 services possess important events and are detailed in the sections below. While they are now included in Audit (Standard) logs, Streams, and Viva Engage are outside the scope of this playbook.

2.3.1 Microsoft Exchange

MailItemsAccessed

This event can help investigators identify data loss and determine the scope of email messages that may have been read/exfiltrated. If a cyber threat actor gained access to email messages, the **MailItemsAccessed** action will be triggered even if there is no explicit signal that messages were read (i.e., the type of access, such as a bind or sync, defined below, is recorded in the audit record). **MailItemsAccessed** is typically used during forensic investigations after an incident has been remediated and the threat actor has been evicted. However,

in some scenarios, **MailItemsAccessed** can be used for proactive threat hunting or alerting for anomalous mailbox behavior. These will be discussed more in Section 2.4.

This event type's immediate value lies in determining:

- *Left-right timeline boundaries* – Unauthorized access to email can be used to scope the left-right time boundaries of an incident using **MailItemsAccessed** as a baseline.
- *Scope of compromise* – This leverages the output of behavioral analysis of threat actor methods/markers of access. For example, **ClientIPAddress** (infrastructure used), **AppId**, **ClientInfoString** (i.e., user agent), and **MailItemsAccessed** can be beneficial in identifying the scope (breadth) of a compromise by identifying which mailboxes may have been exposed or targeted.
- *Insights on threat actor collection tactics* – The **MailItemsAccessed** action can be a powerful artifact for ascertaining threat actor intent(s) such as intelligence collection or intellectual property theft or for categorizing targeted users to better understand threat actor objectives such as project alignments, team membership, or accesses to specific types of information.

MailItemsAccessed replaced **MessageBind** for mailbox auditing, providing the following improvements:

- Applies to all logon types (not just **AuditAdmin**).
- Auditing is triggered by both **Sync** and **Bind** access types.
- Enhanced aggregation techniques reduce audit noise through the creation of fewer audit records.

The **MailItemsAccessed** event is a mailbox auditing action triggered when mail data is accessed by mail protocols and mail clients. This includes all mail protocols (**POP**, **IMAP**, **MAPI**, **EWS**, **ExchangeActiveSync**, and **REST**) and covers both methods of accessing mail (**sync** and **bind**). The difference between these access methods is provided below:

- **Sync**: This is used by Outlook; entire folders are synced between Exchange in the cloud and Outlook desktop.
    - Due to the large volume of logs generated, it does not include information about individual messages.
    - Its presence indicates the entire folder was sync'd (i.e., accessed or exfiltrated).
- **Bind**: This is used by web clients such as Outlook on the Web (OWA), IMAP, and POP3.
    - Each individual email is recorded in the audit log.
    - This indicates individual access to an email message.
    - Bind operations within two-minute windows are aggregated in a single audit record.
    - Aggregated records can be identified with the **OperationCount** field in the **AuditData** property.

MailItemsAccessed events can be throttled if more than 1,000 records are generated on a mailbox in less than 24 hours. When this occurs, **MailItemsAccessed** activity will not be logged for 24 hours after the mailbox was throttled. It should be noted that throttling does not apply if a user is searching in other mailboxes (e.g., Admin searching user mailboxes). Microsoft indicates that fewer than one percent of all mailboxes in Exchange Online are throttled; therefore, *the presence of throttling could be an indication of mailbox misuse or compromise*.

MailItemsAccessed contains many subfields within the **AuditData** field, all of which can be of value to different investigative situations. The list below represents commonly used fields pertinent to most forensic investigations:

- **MailAccessType** – This indicates the **bind** or **sync** operation type.
- **ClientIPAddress** – This is the IP address of the client machine that accessed the mail.

- **ClientInfoString** – This is the user agent of the activity and is useful for detecting unusual activity and pivoting between other log events.
- **SessionID** – This differentiates threat actor actions versus day-to-day user activities with the same account. (Note: This is beneficial for pivoting between other log events.)
- **InternetMessageId** – This can be used to identify the specific email message accessed. (Note: This is beneficial for pivoting between log types, and for eDiscovery / compliance investigations.)
- **UserId** – This is the User Principal Name (UPN) of the user reading the message. (Note: This is beneficial for pivoting between other log events.)
- **AppId** – This is the unique identifier for the application that performed the access on behalf of the user. (Note: This is beneficial for statistical outlier analysis.)
- **ParentFolder** –This is the full folder path of the mail item that was accessed.
- **Logon_type** – This is the logon type of the user who performed the action. The logon types (and their corresponding Enum value) are Owner (0), Admin (1), or Delegate (2).
- **MailboxUPN** – This is the UPN of the mailbox where the message being read is located. (Note: This is beneficial for pivoting.)
- **OperationCount** – This is the count of emails accessed per operation.
- **IsThrottled** – This is useful for investigators to see if there was throttling.

MailItemsAccessed audit records can be natively searched within the Microsoft ecosystem in multiple ways.

- They can be searched in the *Accessed mailbox items* drop-down list in the audit log search tool (https://learn.microsoft.com/en-us/purview/audit-search?tabs=microsoft-purview-portal) in the Microsoft Purview portal.

- They can be found via the Exchange Online PowerShell:

```
Search-UnifiedAuditLog –Operations MailItemsAccessed
```

```
Search-MailboxAuditLog –Operations MailItemsAccessed
```

Send

By using the mailbox auditing action event Send, investigators can identify email sent from specified account(s). This auditing information can help investigators identify information about email messages sent from a compromised account, including those potentially sent by a threat actor. Additionally, investigators can use the Purview eDiscovery tool, message trace, or Graph API to search for the message (by using the subject line or **InternetMessageID**) to identify the recipients and actual content of the message. Users trigger this event by sending, replying to, or forwarding an email message. Other potential uses are detailed below:

- *Additional temporal context* – A second set of timestamps can be used to scope when a threat actor may have sent messages. When compared with timestamps in **MailItemsAccessed**, it can help investigators assemble a more complete picture of threat actor activity.
- *Differentiate scope or intent of threat actor's operation(s)* – The **Send** event helps to determine when a threat actor moves from internal reconnaissance or intelligence collection activities to externally focused activity (e.g., phishing). The presence or lack of presence of sent messages in the **Send** event can help investigators determine the scope of the threat actor's activity.
- *Subject line* – If a threat actor is carrying out phishing operations from your environment, determining the subject line(s) can be a useful indicator of compromise (IOC), which can inform defensive action to

counter the campaign.

The audit record for a **Send** event contains the following fields pertinent to most forensic investigations:

- **ClientIPAddress** – This is the IP address of the client that sent the email.
- **ClientInfoString** – This is the User Agent of the process that sent the email.
- **ClientProcessName** – This is the process name that initiated the sent email.
- **CreationTime** – These timestamps indicate when the message sends successfully.
- **InternetMessageID** – This is the unique identifier of the email message. (Note: This is beneficial for pivoting between other log events and for eDiscovery / compliance investigations.)
- **Subject** – This is the subject line used in the sent email.
- **Attachments** – This contains a list of the attachment names and file sizes (MAPI property size, not the physical attachment size).
- **MailboxOwnerSID** – This is the security identifier (SID) of the mailbox owner from which the message was sent.
- **MailboxOwnerUPN** – This is the UPN of the mailbox owner from which the message was sent.

Send event audit records can be natively searched within the Microsoft ecosystem in multiple ways.

- They can be searched through the *Sent messages* drop-down list in the audit log search tool in the Microsoft Purview portal.
- They can be searched via the Exchange Online PowerShell.

```
Search-UnifiedAuditLog -Operations Send
```

```
Search-MailboxAuditLog -Operations Send
```

SearchQueryInitiatedExchange

Investigators can use the **SearchQueryInitiatedExchange** event to determine if a threat actor compromised an account and searched for or tried to access sensitive information in the mailbox. This audit record contains the actual text of the search typed into the search bar. The audit record also indicates the Outlook environment in which the search was performed (e.g., Desktop, Android, iOS). By looking at the search queries that a threat actor may have performed, an investigator can better understand the threat actor's intent from a review of the email data that was searched. The other uses of the data are detailed below.

- Statistical and outlier analysis – Search terms can be baselined by a user to better understand what searches occur on a regular basis. Once a baseline is established, statistical outliers can be identified and flagged for further analysis (e.g., first seen, first seen in X time, one search term to many user accounts). Additionally, this event can be used to identify other outliners, such as off-hours searching or searches from an infrequently used device (e.g., normally uses Outlook desktop, new Search events appear from Outlook for iOS).
- Behavioral analysis of threat actor activity – Post incident, investigators can use this event to create a behavioral profile of the threat actor's activity in your environment. Identifying what emails or attachments threat actors searched for provides a powerful mechanism for understanding a threat actor's intentions. For example, intellectual property theft, intelligence collection, building target lists (e.g., building target lists or intelligence for potential supply chain compromise), or the targeting of specific user(s) can help determine why your organization was targeted. This behavioral profile can also be beneficial for predictive analysis purposes since understanding a threat actor's intentions may allow for the identification of follow-on objectives or additional targets.

- Sensitive terms alert list – Proactively maintain search term alert lists for various categories of activity within your organization. Sensitive projects, contractual negotiations, payroll/finance/human resources, and other search terms related to information pertinent to your organization's success can be maintained. Users who should not be searching for or accessing information in these categories can be flagged for additional analysis.

The **SearchQueryInitiatedExchange** event triggers when a person uses Outlook to search for items in a mailbox. Events are triggered when searches are performed in the following Outlook environments:

- Outlook (desktop client)
- Outlook on the web (OWA)
- Outlook for iOS
- Outlook for Android
- Mail app for Windows 10

SearchQueryInitiatedExchange contains many subfields within the **AuditData** field, all of which can be of value to different investigative situations. The list below represents only commonly used fields pertinent to most forensic investigations.

- **CreationTime** – This provides the timestamp of when the search was performed.
- **Client-IP** – This is the IP address of the client machine that performed the search.
- **Operation** – This indicates the operation (e.g., **SearchQueryInitiatedExchange**).
- **QueryText** – This provides the user-inputted string in the query bar.
- **ScenarioName** – This provides what was used for the search (e.g., OWA.react).
- **ClientUserAgent** – This is the User Agent of the client used to perform the search.
- **UserId** – This is the UPN of the user who performed the search.

SearchQueryInitiatedExchange event audit records can be natively searched within the Microsoft ecosystem in multiple ways.

- They can be searched through the *Performed email search* drop-down in the audit log search tool in the Microsoft Purview portal.
- They can be searched via Exchange Online PowerShell.

```
Search-UnifiedAuditLog -Operations SearchQueryInitiatedExchange
```

2.3.2 Microsoft SharePoint Online

SearchQueryInitiatedSharePoint

Like **SearchQueryInitiatedExchange**, investigators can use the **SearchQueryInitiatedSharePoint** event to determine if a threat actor tried to search for (and possibly access) information in SharePoint. The audit record for a **SearchQueryInitiatedSharePoint** event contains the actual text of the search query and indicates the type of SharePoint site that was searched. By looking at the search queries a threat actor may have performed, an investigator can better understand the intent and scope of the file data being sought. In addition to all the analytical methods described in Section 2.3.1 (**SearchQueryInitiatedExchange**), **SearchQueryInitiatedSharePoint** events can also be used in the following ways:

- They can enrich threat actor behavioral profile(s). With this event, search activity can be differentiated between Exchange (email) and SharePoint workloads. This can help narrow or categorize the threat actor's activity. Comparing terms used between the two workloads can also provide an additional layer of insights into threat actor activity (e.g., if the same terms were used on both Exchange and

SharePoint, or if different terms were used). This could help differentiate between your organization being a target of opportunity (e.g., general search terms used across the board) or one that was highly researched and targeted (e.g., specific search terms targeted at specific users/SharePoint sites).

The **SearchQueryInitiatedSharePoint** event is triggered when a user searches for items in SharePoint. Events are triggered when searches are performed on the root or default page of the following types of SharePoint sites:

- Home sites
- Communication sites
- Hub sites
- Sites associated with Microsoft Teams

The **SearchQueryInitiatedSharePoint** event contains many subfields within the **AuditData** field, all of which can benefit different investigative situations. The list below represents only commonly used fields pertinent to most forensic investigations.

- **CreationTime** – This provides the timestamp of when the search was performed.
- **Client-IP** – This is the IP address of the client machine that performed the search.
- **Operation** – This indicates the operation (e.g., **SearchQueryInitiatedSharePoint**).
- **QueryText** – This provides the user-inputted string in the query bar.
- **ScenarioName** – This shows what was used for the search (e.g., **TeamSiteSearch**).
- **ClientUserAgent** – This provides the User Agent of the client used to perform the search.
- **UserId** – This provides the UPN of the user who performed the search.

The **SearchQueryInitiatedSharePoint** event audit records can be natively searched within the Microsoft ecosystem in multiple ways.

- They can be searched through the *Performed SharePoint Search* drop-down in the audit log search tool in the Microsoft Purview portal.
- They can be searched via Exchange Online PowerShell.

```
Search-UnifiedAuditLog -Operations SearchQueryInitiatedSharePoint
```

2.3.3 Microsoft Teams

Microsoft Teams event logs provide vital information for forensics and incident response investigations. They provide a detailed record of user actions on the Teams workload, help reconstruct timelines, identify participants, and provide added context to threat actor behavior. These logs also facilitate anomaly detection and may support legal discovery and litigation.

MeetingParticipantDetail

This event provides information in Teams about the participants of a meeting, including the user ID of each participant, the time a participant joined the meeting, and the time a participant left the meeting. This can be used for establishing threat actor patterns of interaction with Teams meetings.

MeetingParticipantDetail can be valuable in the following investigative scenarios:

- *Temporal analysis:* Three different timestamp fields in **MeetingParticipantDetail** can be used to scope the different interactions a threat actor may have with Teams meetings. **InviteTime** can be used to ascertain when an invitation was sent, which can be useful for determining if a threat actor was

adding participants to a meeting. **JoinTime** and **LeaveTime** can be used to determine how long an attendee was in a meeting. This can help determine what information may have been exposed to a threat actor during the course of a meeting.
- *Timestamp comparison:* **InviteTime** and **JoinTime** in **MeetingParticipantDetail** can be compared to known accessed email in **MailItemsAccessed** to ascertain prior knowledge of meeting(s) or near-real-time intelligence collection (i.e., leveraging intelligence collected in email to join meetings).
- *Insights on threat actor interactions with Teams meetings:* Meetings joined by threat actor controlled/created identities can be used to identify targeting and can be an added data point for behavioral analysis of the threat actor's intentions (e.g., which meetings and topics were targeted).

MeetingParticipantDetail contains many subfields within the **AuditData** field, all of which can be of value to different investigative situations. The list below represents commonly used fields pertinent to most forensic investigations:

- **UserID** – This log records the identities of all participants in a meeting. This includes the organization's employees and external participants, if any. Such information is crucial in establishing who was present during a particular discussion.
- **Attendees/Role** – This log can detail the roles of participants within the meeting (e.g., organizer, presenter, or attendee), along with the level of access and control different participants had in the meeting, which could be important in cases of information leaks or unauthorized disclosures.
- **ClientIP** – Depending on the organization's privacy and logging settings, this log may also include information about where participants joined the meeting from, such as IP addresses. The geographical locations of the participants can be crucial in certain investigations, especially user identity compromise cases.
- **DeviceID** – Logs about the devices used by participants to join the meeting—laptop, mobile phone, or tablet, and their operating systems—can be recorded. This can be useful in pinpointing security incidents or in understanding access patterns.

MessageSent

This event provides a record of outgoing communications, capturing detailed information about when messages are sent, the sender, the recipient, and associated metadata of the message (e.g., channel name, meeting name, or chat name). This event helps to identify communication patterns and establish timelines in threat actor interaction with Teams messages. This event is only triggered if the Graph API is used or if there are guests, federated, and/or anonymous users.

MessageSent can be valuable in the following investigative scenarios:

- *Temporal analysis:* **MessageSent** is an additional data point for timeline analysis and provides insights on when Teams messages were sent through the Graph API or with guest, federated, or anonymous users. This can be valuable for ascertaining threat actor behavior and identification of shifting intent from data/intelligence collection to possible interaction with users via Teams chat messages.
- *Identify identities of Teams chats sent:* **ClientIP** and **UserId** fields within **MessageSent** can be used to uniquely identify who has sent messages via Graph API or to chats with guest, federated, or anonymous users. This can be beneficial for identifying and baselining which identities a threat actor may have used to generate or send messages.
- *Determine meeting interaction:* If the chats sent were part of a meeting with guest, federated, or anonymous users, the **MessageSent** log identifies the chat name and/or the meeting name through the **ChatName** and **ItemName** fields respectively. This can help investigators differentiate a threat actor who may have been joining meetings to passively listen (collect intelligence) versus those in

which a threat actor may have joined and interacted with attendees in the meeting (e.g., sending messages).

MessageSent contains many subfields within the **AuditData** field, all of which are of value in different investigative situations. The list below represents commonly used fields pertinent to most forensic investigations:

- **ClientAppId** – A unique identifier indicating the application associated with the Graph API call.
- **ClientAppName** – A human-readable name for the application associated with the Graph API call.
- **MessageId** – The message's unique identifier allows incident investigators to quickly pinpoint the specific message and examine it in the context of the broader conversation or investigation.
- **UserId** – The identity of the user who sent the message.
- **CreationTime** – The exact time when the message was sent.
- **ParticipantInfo** – Identifies the presence of foreign tenant users, guests, or unauthenticated users in a conversation and can be critical in understanding the scope of information sharing and potential exposure. The following fields only display if the chat has guests, federated, and/or anonymous users.
    - **HasForeignTenantUsers** – This indicates whether users from other M365 tenants (i.e., outside the organization's primary domain) were part of the communication. This can be important in cases of cross-organizational collaboration.
    - **HasGuestUsers** – This shows if guest users (users invited to join a specific team or group but not part of the organization's main tenant) were involved.
    - **HasOtherGuestUsers** – This is like **HasGuestUsers** but specifically refers to guests not belonging to the primary or secondary domains.

MessagesListed

This event is triggered when messages are listed or viewed leveraging the Graph API. For example, using the Graph API to call **/v1.0/chats/<chatID>/messages** would generate a list of messages in a chat thread (**ChatID**) and trigger the **MessagesListed** event. This can be a useful data point for establishing Teams messages read or accessed by a threat actor and can aid in identifying malicious scripting and automation leveraging the Graph API.

MessagesListed can be valuable in the following investigative scenarios:

- *Profiling access type*: **AppAccessContext**, APIID, ClientAppID, and **ClientAppName** can be used to identify what tool(s) a threat actor may have been using to interact with the Teams messages (e.g., Graph Explorer).
- *Determine scope of Teams chat messages accessed*: **ChatThreadId** provides a unique identifier for chat threads that may have been listed. Messages contained within these threads, when corroborated with other data sources, such as identities known to have been used, left/right timelines of their activity, and known infrastructure (IPs, **UserAgents**) leveraged can be used to ascertain Teams messages that may have been read or accessed by a threat actor.
- *Identifying threat actor use of scripting and automation*: Using **ClientAppId** and **ClientAppName** in conjunction with **CreationTime,** an investigator can identify patterns in API calls and use. Threat actors leveraging scripting and automation will typically generate a high volume of requests in a short period of time. Although a threat actor may vary the tools leveraged to obfuscate their activity, hunting for anomalous **AppIds** in **MessagesListed** can be an effective way to identify a threat actor's use of scripting and automation.

- *Threat actor scripting and automation profiling:* **ClientAppId** and **ClientAppName** can be beneficial for identifying how a threat actor is interacting with the Graph API and what tools that threat actor may be leveraging (e.g., Graph Explorer).

MessagesListed contains many subfields within the **AuditData** field, all of which can be of value to different investigative situations. The list below represents commonly used fields pertinent to most forensic investigations:

- **ClientAppId** – A unique identifier indicating the application associated with the Graph API call.
- **ClientAppName** – A human-readable name for the application associated with the Graph API call.
- **CreationTime** – The time when messages were listed via the Graph API. This information is crucial for constructing an accurate timeline of events. This allows incident responders to correlate the message with other events and understand the sequence of activities during an incident.
- **UserId** – The UPN of the identity that initiated the Graph API call.
- **Messages**
  - **ChatThreadId** – The chat ID that was listed.
  - **Id** – The message ID that was listed.
  - **Version** – The version of the message that was listed.
- **ItemName** – This indicates the name of the item (such as a channel or chat) where the message was listed.

MeetingDetail

This event provides specific details of a Teams meeting, such as organizer, start time, duration, type (scheduled, reoccurring), and more. This event type can be valuable for the investigation of Teams meetings.

MeetingDetail can be valuable in the following investigative scenarios:

- *Temporal analysis:* **MeetingDetail** can be used to determine when a meeting was created via the **CreationTime** field and when it was started via the **StartTime** field. This is valuable in scenarios where a threat actor acquired/created identity is scheduling and starting meetings.
- *Threat actor infrastructure:* **MeetingDetail** contains a **ClientIP** field that indicates the IP address of the organizer. In addition, it also provides **UserAgent** and **DeviceType** fields. These fields can be valuable for identifying infrastructure used by the threat actor. Threat actor objectives and intent (determined via email, Teams meetings, SharePoint, etc.) can be compared with the infrastructure used to ascertain if multiple threat actor teams or entities are responsible for different parts of the campaign.

MeetingDetail contains many subfields within the **AuditData** field, all of which can be of value to different investigative situations. The list below represents commonly used fields pertinent to most forensic investigations:

- **CreationTime** – This timestamp indicates when the meeting record was created and can help establish a timeline, which is crucial to understanding the sequence of events in an investigation.
- **Id** (Meeting ID) – This unique identifier distinguishes this specific meeting from others. It is used to correlate activities or issues to this event.
- **UserId** – The identity of the user who scheduled the meeting.
- **ChatThreadId** – The unique identifier for the meeting's chat thread; this is useful for correlating chat communications with specific meetings.
- **StartTime** – The actual start time of the meeting.
- **EndTime** – The actual end time of the meeting.
- **MeetingURL** – Tracking the URL used to join the meeting enables investigators to track how participants accessed the meeting or verify if the meeting link was shared outside the intended group.

- **Modalities** – This indicates the types of communication that were used in the meeting (e.g., audio, video, chat). Understanding what modalities were used can provide insights into how information was communicated and shared.
- **ICalUid** – This is a unique identifier for the meeting in calendar formats and is useful for correlating the meeting with calendar entries and invites.
- **ProviderTypes** – These indicate the platform used for the meeting, which is relevant to understanding the technical context of the meeting.

MessageUpdated

This event is triggered when a user edits a chat message within Teams. The event log provides essential insights into the modifications made to Teams communications, which helps to establish timelines and determine threat actor interaction with Teams chats within an organization.

MessageUpdated can be valuable in the following investigative scenarios:

- *Temporal analysis:* The **CreationTime** field can be used to ascertain when a given message was edited. When investigating the totality of threat actor activity, this is another timestamp that can be leveraged.
- *Mis/Disinformation and data manipulation:* When corroborated with other known factors of threat actor activity such as threat actor controlled/created identities, timelines of activity, and known infrastructure utilized, **MessageUpdated** can be used to determine Teams messages that may have been intentionally edited with malicious purpose. Note: Purview eDiscovery or other means would need to be used to identify chat content.
- *Threat actor infrastructure:* **MessageUpdated** contains a **ClientIP** field that indicates the IP address of the user who updated the message. This field can be valuable for identifying infrastructure used by the threat actor. Threat actor objectives and intent (determined via email, Teams meetings, SharePoint, etc.) can be compared with infrastructure used to ascertain if multiple threat actor teams or entities are responsible for different parts of the campaign.

MessageUpdated contains many subfields within the **AuditData** field, all of which can be of value to different investigative situations. The list below represents commonly used fields pertinent to most forensic investigations:

- **ClientAppId** – A unique identifier indicating the application associated with the Graph API call.
- **ClientAppName** – A human-readable name for the application associated with the Graph API call.
- **CreationTime** – The timestamp of when the message was updated.
- **UserId** – This identifies who edited the message. This is key to attributing the action to a specific individual.
- **MessageId** – This is the unique identifier of the message that was updated; it allows the specific message in question to be pinpointed and examined in the context of the broader conversation or investigation.
- **ItemName** – This indicates the name of the item (such as a channel or chat) where the message was updated. It provides context to the location within Teams where the edit occurred.
- **ExtraProperties** – This provides additional information about the environment from which the message was updated, including time zone, operating system, country, client name, and version. This data can help investigators understand the technical context of the edit and may provide clues about the device or location used.
- **AADGroupId**, **TeamGuid**, **ChannelGuid** – These identifiers correlate the updated message to a specific group, team, or channel within Teams.
- **MessageVersion** and **ParentMessageId** – These fields can be important in tracking the history of a message, including its original version and how it has evolved over time through edits.

- **ParticipantInfo** – While not directly related to the message content, understanding the composition of participants (e.g., the presence of guests or external users) in the conversation can provide additional context.

ChatRetrieved

This event is only triggered when a chat is retrieved or accessed via the **/v1.0/chats/<chatID>** Graph API call. The API call would provide a threat actor context of Teams chat threads, including chat type (one-on-one), created dates, last updated times, and last message read times. The corresponding event log provides details of chat activities, including timestamps, context on the application used to initiate the API call (e.g., Graph Explorer), and other metadata potentially valuable to forensic investigations. This event can be valuable for establishing what Teams chats were profiled by a threat actor and can aid in identifying scripting and automation used in conjunction with the Graph API.

ChatRetrieved can be valuable in the following investigative scenarios:

- *Identifying threat actor use of scripting and automation:* Using **ClientAppId** and **ClientAppName** in conjunction with **CreationTime,** an investigator can identify patterns in API calls and use. Threat actors leveraging scripting and automation will typically generate a high volume of requests in a short period of time. Although threat actors may vary the tools leveraged to obfuscate their activity, hunting for anomalous **AppIds** in **ChatRetrieved** can be an effective technique to identify threat actor use of scripting and automation.
- *Threat actor scripting and automation profiling:* **ClientAppId** and **ClientAppName** can be beneficial for identifying how threat actors are interacting with the Graph API and what tools they are leveraging (e.g., Graph Explorer).
- *Temporal analysis:* **CreationTime** in **ChatRetrieved** can assist in building a complete timeline of threat actor activity by highlighting Graph API usage by the threat actor.

The following **AuditData** subfields in **ChatRetrieved** are valuable for most forensic investigations:

- **ClientAppId** – A unique identifier indicating the application associated with the Graph API call.
- **ClientAppName** – A human-readable name for the application associated with the Graph API call.
- **CreationTime** – Timestamp indicating the time when the Graph API was called.
- **UserID** – The UPN of the identity that initiated the Graph API call.

MessageRead

This event is only triggered when a chat is read via the Graph API. For example, using the Graph API to call **/beta/chats/<ChatId>/messages/<MessageID>** would retrieve a message (**MessageId**) in a chat thread (ChatID) and trigger the **MessageRead** event. This can be a useful data point for establishing Teams messages read or accessed by a threat actor and can aid in identifying malicious scripting and automation leveraging the Graph API.

MessageRead can be valuable in the following investigative scenarios:

- *Targeted chat message collection:* **MessageRead** could facilitate the identification of targeted chat collection by a threat actor. Given the nature by which this event is triggered (the retrieval of a single message identified by a **MessageId**), it could indicate the threat actor is performing near-real-time intelligence collection in the environment. For example, harvesting **ChatId** and **MessageIds** surrounding users or subjects pursuant to their objectives and intent and then performing retrievals of specific messages based on this targeted collection.

- *Identifying threat actor use of scripting and automation*: Using **ClientAppId** and **ClientAppName** in conjunction with **CreationTime,** an investigator can identify patterns in API calls and use. Threat actors leveraging scripting and automation will typically generate a high volume of requests in a short period of time. Although a threat actor may vary the tools leveraged to obfuscate their activity, hunting for anomalous **AppIds** in **MessageRead** can be an effective technique to identify threat actor use of scripting and automation. Specifically, with **MessageRead**, if high volumes of **ChatId** and **MessageId** retrievals are being initiated in a short window of time, it may indicate harvesting of this information from elsewhere and adapting it with scripted or automated collection tools.
- *Threat actor tool profiling:* **ClientAppId** and **ClientAppName** can help identify how threat actors are interacting with the Graph API and what tools they may be leveraging (e.g., Graph Explorer).

MessageRead contains many subfields within the **AuditData** field, all of which can be of value to different investigative situations. The list below represents commonly used fields pertinent to most forensic investigations:

- **ClientAppId** – A unique identifier indicating the application associated with the Graph API call.
- **ClientAppName** – A human-readable name for the application associated with the Graph API call.
- **CreationTime** – The timestamp of when the message was read.
- **UserId** – This identifies the user or the application that read the message.
- **MessageId** – This is the unique identifier of the message that was read; it allows the specific message in question to be pinpointed and examined in the context of the broader conversation or investigation.
- **ItemName** – This indicates the name of the item (such as a channel or chat) where the message was read. It provides context to the location within Teams where the edit occurred.
- **MessageVersion** – This field can be important in tracking the history of a message, including its original version and how it has evolved over time through edits.

MessageHostedContentRead

This event is only triggered when Teams-hosted content (e.g., code snippets, hosted images) is retrieved via the Graph API. For example, using the Graph API to call **/chats/<ChatId>/messages/<MessageId>/hostedContents/<HostedContentId>** would retrieve the hosted content (e.g., code snippet) from a message (**MessageId**) in a specific chat (**ChatId**). It should be noted that file attachments are not classified as hosted content; they are stored in SharePoint or OneDrive. This event can be useful for establishing threat actor usage of the Graph API and for establishing non-file-hosted content that may have been accessed or acquired.

MessageHostedContentRead can be valuable in the following investigative scenarios:

- *Identify code and other non-standard chats targeted*: Corroborated with other data sources such as threat actor-controlled/-created identities, timelines of their actor activity, and infrastructure leveraged, **MessageHostedContentRead** can be used to determine non-standard chat content that may have been targeted. For example, a threat actor may target code snips and other hosted content to proliferate access via embedded secret material for intellectual property theft or for exploit development.
- *Threat actor scripting and automation profiling:* **ClientAppId** and **ClientAppName** can be beneficial for identifying how threat actors are interacting with the Graph API and what tools they may be leveraging (e.g., Graph Explorer).

MessageHostedContentRead contains many subfields within the AuditData field, all of which can be of value to different investigative situations. The list below represents commonly used fields pertinent to most forensic investigations:

- **ClientAppId** – A unique identifier indicating the application associated with the Graph API call.
- **ClientAppName** – A human-readable name for the application associated with the Graph API call.
- **CreationTime** – The timestamp of when the message's hosted content was read; it is critical to establishing a timeline of events, particularly in relation to the incident being investigated.
- **UserId** – The user or the application that read the message's hosted content, enabling investigators to attribute the action to a specific individual or application.
- **MessageId** – The unique identifier of the message that has hosted content. It allows the specific message in question to be pinpointed and examined in the context of the broader conversation or investigation.
- **ItemName** – Indicates the name of the item (such as a channel or chat) where the message was read, providing context to the location within Teams where the edit occurred.
- **HostedContentsId** – Identifies what hosted content was accessed.

2.4 SCENARIO-BASED ANALYSIS

Compromise comes in many forms, and threat actors typically follow the path of least resistance. Determined and persistent threat actors are driven by their objectives and will find whatever means necessary to achieve success. While each compromise scenario is different and encompasses unique behavioral characteristics, some tactics are more favorable than others. For example, many threat actor groups target **user identities** (T1586 -- https://attack.mitre.org/techniques/T1586/) because of the positive returns on investment.

In fact, threat actors target user identities for a variety of additional reasons. The end users have been, and will be for the foreseeable future, the weakest link in the chain of cyber defense. Fallible human nature, a widening schism between generational knowledge in cyber, and continued reliance on simple passwords are all contributing factors. Additionally, the increased prevalence of artificial intelligence (AI) in threat actor operations has created a multitude of new challenges. For example, what once was a staple detection method—language barrier-induced grammatical inconsistencies in phishing messages—has been antiquated by the onset of AI-assisted phishing and deception campaigns. Although challenges in defending against identity-based intrusions will continue to exist, there are steps that can be taken by defenders to counter the threat actor's reliance on this tactic.

This scenario and the following sections focus on how cyber defenders can utilize Microsoft's expanded logging capabilities to facilitate an intelligence-driven approach. This approach equips defenders with the tools and tactics necessary to detect and defend against identity compromises. It focuses on three primary tactics related to identity-based compromises:

- **Credential Access** (TA0006 -- https://attack.mitre.org/tactics/TA0006/)
- **Exfiltration** (TA0010 – https://attack.mitre.org/tactics/TA0010/)
- **Impact** (TA0040 -- https://attack.mitre.org/tactics/TA0040/)

2.4.1 Detect Credential Access through Accessed Mail

Threat actors acquire credentials (identities) in a variety of ways. Phishing-based techniques, Adversary-in-the-middle (AiTM), acquisition through compromise (e.g., dumping New Technology Directory Services Directory Information Tree, or NTDS.dit, from a domain controller, acquiring intentionally stored credentials in a file), brute force, or even the purchase of ill-gotten credentials from the dark web are all methods that can be used. The section focuses on how to detect the unauthorized use of legitimate credentials in your environment by leveraging Microsoft's expanded logging capabilities.

Threat Actor Behavior: Accesses M365 mailbox using legitimate identity (T1586 -- https://attack.mitre.org/techniques/T1586/) from virtual private server (VPS) or proxy infrastructure. Searches mailbox(es) for the purpose of intelligence collection (T1114 -- https://attack.mitre.org/techniques/T1114) pursuant to broader campaign goals.

Analytical methods: Microsoft's expanded logging capabilities can be used to aid in the detection of illicit access to mailbox(es) in the M365 tenant through the **MailItemsAccessed** event type. This can be accomplished in a proactive manner, with no prior knowledge of the compromise described above, or as a reactive mechanism when supplemental threat intelligence has been identified (e.g., **AppId**, **ClientIP**).

Proactively, **MailItemsAccessed** can be leveraged to trend access to mailboxes from unique **AppId**s over time. Employing statistical outlier methods of analysis, investigators can identify and flag for further triage the following categories of anomalous activity:

- **AppId**'s observed accessing mailbox(es) for the first time.
- **AppId**'s observed accessing multiple mailboxes.
- Large spikes in **AppId** mailbox activity (stacked by **AppID**).
- **AppId** historical trending – first seen in X time (tunable threshold, varies by use case).
- Analyze statistical outliers in mail clients accessing mailboxes in **ClientInfoString** (e.g., a spike in Android use in an environment where only iOS devices are issued).
- Baseline/trending analysis of mail clients accessing mailboxes in **ClientInfoString** (e.g., Bob in accounting typically uses Outlook desktop and switches to iOS app).

Reactive detection of the behavior can also be accomplished with the **MailItemsAccessed** event type. This is predicated on prior knowledge of the activity. In this analytical method, **SessionId** is critically important for building a more complete picture of all threat actor activity in your environment. As a globally unique identifier (GUID) included in the Entra ID (Active Directory) token, **SessionId** allows investigators to group activities performed in a single logon session together, regardless of ephemeral indicators that the threat actor may arbitrarily manipulate (e.g., IP address, User Agent). The **SessionId** changes only when a user reauthenticates to Entra ID. **SessionIds** are logged in Mailbox and Admin audit logs. Understanding the totality of the threat actor's activity can be accomplished with the following analytical methods.

- Identify accessed mail by using **ClientIP.** Note that nation-state threat actors pivot infrastructure quickly; IP addresses should not be the only point of focus.
- Leverage **SessionID**(s) to discern the totality of threat actor activity across entire logon sessions. Note that this can also help distinguish threat actor activity from benign user activity for the same account/identity.

Sample queries: Proactively, the Advanced Hunting query below can be used to visualize an application's daily mail access count. Large spikes in usage or uncommon applications should be investigated. For a list of commonly used Microsoft applications, visit the Microsoft Learn article Verify first-party Microsoft applications in sign-in reports (https://learn.microsoft.com/en-us/troubleshoot/azure/entra/entra-id/governance/verify-first-party-apps-sign-in),

```
CloudAppEvents
| where ActionType == “MailItemsAccessed”
| extend Accessing_AppId = tostring(RawEventData.AppId)
| summarize count() by bin(Timestamp,1d),Accessing_AppId
| render timechart

OfficeActivity
| where Operation == “MailItemsAccessed”
| summarize count() by bin(TimeGenerated,1d),AppId
| render timechart
```

Reactively, the Advanced Hunting query below can be used to identify mail items accessed by a specific IP address.

```
//Get sessions associated with suspicious IP address
let bad_sessions = materialize (
AADSignInEventsBeta
| where IPAddress == 'x.x.x.x' //Replace with IP of interest
| where isempty(SessionId) == false
| distinct SessionId
);
//Get any mail accessed during suspicious sessions
CloudAppEvents
| where ActionType == 'MailItemsAccessed'
| where RawEventData.SessionId has_any (bad_sessions)
```

2.4.2 Detect Exfiltration Through Anomalous Search Activity

Threat Actor Behavior: The threat actor leverages Outlook and SharePoint query bars to perform searches for desired information across multiple compromised identities (T1594 – https://attack.mitre.org/techniques/T1594/) and/or leverages pre-compiled list of keywords for scaling of operations and reduced overhead (e.g., single threat actor engaging with multiple compromised environments simultaneously).

Analytical methods: Microsoft's expanded logging capabilities contain unparalleled insights into understanding end-user behavior. These insights allow investigators to assemble a behavioral profile of threat-actor motivation and intent through visibility on user-entered searches in both Exchange and SharePoint workloads. This is accomplished through the **UserSearchQueryInitiatedExchange** and **UserSearchQueryInitiatedSharePoint** event types, respectively.

Figure 4: Pyramid of Pain (Bianco, 2013)

The analytical methodologies at the top of the Pyramid of Pain (see Figure 4) detect advanced state-sponsored threat actors most effectively, reducing the reliance on ephemeral indicators that are easily changed (e.g., IP address, UserAgent, hashes). These event types arm investigators with the data necessary to characterize even the most sophisticated threat actors. Proactive analytical methodologies for these critical event types are detailed below.

- Hunt for anomaly-based user search queries, search for statistical outliers such as off-hours searching, and look for anomalous search terms (e.g., Bob in accounting searching SharePoint sites for information on sensitive government projects).
- Search query baselining can attempt to ascertain normal patterns of activity throughout the course of daily business. Outliers, such as "first seen" or "first seen in X time," can be flagged for additional analysis. (Note: This method would net high false positive rates and would need to be tuned to fit each distinct use case.)
- Proactively assemble search query alert lists based on organizational needs. Terms associated with highly sensitive work can be flagged for triage if they are searched outside the approved team (e.g., Bob in accounting searching for "incident response plan" or "network diagrams").
- Look for one-to-many search patterns as a method to proactively detect anomalous search behavior. Given the threat actor scenario described in the section above, this method could be used if search terms are copied out of a playbook and the same (or similar) terms are used across multiple compromised identities or mailboxes. While it may be normal for Alice in cybersecurity to search for "incident response plan," it may be anomalous for Alice in cybersecurity, Bob in accounting, and John in payroll to all search for "incident response plan" in a short window of time.

UserSearchQueryInitiatedExchange and **UserSearchQueryInitiatedSharePoint** are also invaluable for reactive-analysis scenarios. After an incident has been identified and supplemental threat intelligence established (e.g., **ClientIPs**, **SessionIds**, **UPNs**), these event types can be used to illustrate the threat actor's motive, intent, and objectives.

In building a profile of what types of information were sought, investigators may be able to understand why the threat actor targeted them. This level of insight can be a form of predictive analysis, allowing investigators to potentially identify the threat actor's goals or next steps. This information can be used to bolster defenses. Reactive analytical methodologies are detailed below:

- Corroborate other activity, such as SharePoint file operations and **MailItemsAccessed**. This can be beneficial for indicating data access or exfiltration (e.g., searched a file on SharePoint, with accompanying **FileAccessed**, **FileCopied**, or **FileDeleted** operations).
- Build a profile of intent by categorizing threat actor-used search terms across both Exchange and SharePoint workloads. For example, creating buckets and grouping similar search terms can help determine the types of information sought. For example, "cybersecurity insurance," "incident response plan," and "incident response team" might be categorized as *cybersecurity capabilities* whereas "contracts," "RFI," and "RFP" might be categorized as *contracts*.
- Similarly, the individuals targeted can be added to the profile described above. While targeting Bob in accounting might seem benign, if Bob oversaw financials for a contract related to a sensitive project and the threat actor was searching for *contracts*-related information, this could be a powerful mechanism to derive adversary intent.
- Through a combination of user search query and targeted user analysis, investigators may infer threat actor's interests and assemble a possible target list. This list may encompass entities internal or external to your organization, and it may be beneficial for determining next steps in the campaign (i.e., follow-on actions, in your organization, targets for supply chain compromise).

Sample Queries:

Here is a Proactive Advanced Hunting query for summarizing user searches outside of normal working hours that contains sensitive keywords.

```
    let keywords = dynamic(['secret','password','vpn']); //replace with org
specific keywords or remove
    let utc_working_hours = range(2,13); //replace with org specific working hours
    CloudAppEvents
    | extend client_ip = tostring(RawEventData.ClientIP)
    | extend query_text = tostring(RawEventData.QueryText)
    | where ActionType == "SearchQueryInitiatedExchange" or ActionType ==
"SearchQueryInitiatedSharePoint"
    | where not (datetime_part("Hour",Timestamp) in (utc_working_hours))
    | where query_text has_any (keywords)
    | take 100
    | summarize search_number=count(), make_set(query_text), make_set(client_ip)
by AccountDisplayName
```

Here is a Proactive Advanced Hunting query for identifying a user that searches for five or more keywords of interest in a 15-minute time block. CountofSharePointSearches, CountofExchangeSearches, timerange, and KeywordsofInterest can be adjusted to fit the needs of your organization's specific use case(s).

```
let
keywordsOfInterest=dynamic(["vpn","password","anyconnect","pfx","credential","credent
ials","work from home","cisco","palo alto","virtual
desktop","key","secret","confidential","certificate"]);
let timerange=15m;
CloudAppEvents
| where ActionType in
("SearchQueryInitiatedSharePoint","SearchQueryInitiatedExchange")
| extend QueryText=tostring(RawEventData.QueryText)
| extend Workload=tostring(RawEventData.Workload)
| extend UserId=tostring(RawEventData.UserId)
| where QueryText has_any (keywordsOfInterest)
| project Timestamp, ActionType, UserId, Workload, QueryText
| summarize ExchangeSearches=make_list_if(QueryText,Workload ==
"Exchange"),DistinctExchangeSearches=make_set_if(QueryText,Workload == "Exchange"),
SharePointSearches=make_list_if(QueryText,Workload ==
"SharePoint"),DistinctSharePointSearches=make_set_if(QueryText,Workload ==
"SharePoint") by UserId, bin(Timestamp, timerange)
| extend
CountofExchangeSearches=array_length(ExchangeSearches),CountofDistinctExchangeSearche
s=array_length(DistinctExchangeSearches),
CountofSharePointSearches=array_length(SharePointSearches),CountofDistinctSharePointS
earches=array_length(DistinctSharePointSearches)
| project-reorder Timestamp, UserId, CountofExchangeSearches, ExchangeSearches,
CountofDistinctExchangeSearches, DistinctExchangeSearches, CountofSharePointSearches,
SharePointSearches, CountofDistinctSharePointSearches, DistinctSharePointSearches
| where CountofSharePointSearches >= 5 or CountofExchangeSearches >= 5
```

Here is a Proactive Advanced Hunting query for identifying a medium- to high-risk sign-in followed by searches of known terms within 60 minutes (30 minutes on either side of the risky sign-in). **CountofSharePointSearches**, **CountofExchangeSearches**, **timerange**, and **KeywordsofInterest** can be adjusted to fit the needs of your organization's specific use case(s).

```
let
keywordsOfInterest=dynamic(["vpn","password","anyconnect","pfx","credential","credent
ials","work from home","cisco","palo alto","virtual
desktop","key","secret","confidential"]);
AADSignInEventsBeta
| where RiskLevelDuringSignIn in ("50","100")
| project RiskySignInTime=Timestamp, AccountUpn, RiskLevelDuringSignIn,
SignInIPAddress=IPAddress
| join kind=inner(
CloudAppEvents
| where ActionType in
("SearchQueryInitiatedSharePoint","SearchQueryInitiatedExchange")
| extend QueryText=tostring(RawEventData.QueryText)
| extend Workload=tostring(RawEventData.Workload)
| extend UserId=tostring(RawEventData.UserId)
| where QueryText has_any (keywordsOfInterest)
| project SearchTime=Timestamp, UserId, Workload, QueryText, IPAddress
) on $left.AccountUpn==$right.UserId
| extend ['Time Between Risky Sign in and
search']=datetime_diff('minute',SearchTime,RiskySignInTime)
| where ['Time Between Risky Sign in and search'] between (-30 .. 30)
| project RiskySignInTime, SearchTime, AccountUpn, Workload, QueryText, ['Time
Between Risky Sign in and search'], SignInIPAddress, SearchIPAddress=IPAddress
```

2.4.3 Determine the Impact of a Compromise Through Teams Interactions

Threat Actor Behavior: The threat actor carries out follow-on objectives within the compromised environment by joining meetings with a compromised identity for the purpose of intelligence collection (T1113 – https://attack.mitre.org/techniques/T1113/, T1125 – https://attack.mitre.org/techniques/T1125/) and/or using compromised identities to propagate phishing messages to known contacts/business relationships (T1566 – https://attack.mitre.org/techniques/T1566/, T1534 – https://attack.mitre.org/techniques/T1534/).

Analytical methods: Threat actor activities in a compromised environment are not limited to searching for documents and reading email. In some instances, they execute follow-on objectives intended to interact with or impact the environment and its users. As described in the scenario above, joining meetings/calls and sending phishing emails are a few examples of impact-level behavior. Microsoft's expanded logging capabilities can provide investigators with the critical data necessary to determine if impact-level activities occurred.

Proactive analytical methodologies include:

- Utilize **DeviceInformation** in combination with **UserAgent** in **MeetingParticipantDetail** to hunt for anomalous devices joining meetings (e.g., a MacOS device when your organization uses only Windows).
- EntraID's Risky User's report can be compared to **UserIDs** in **MeetingParticipantDetail** to identify users flagged for risky behavior that may be joining meetings.
- **RecipientType** in **MeetingParticipantDetail** can be leveraged to hunt for meeting attendees from external or federated tenants. (Note: The presence of external users by itself does not indicate compromise.)

Reactive analytical methods are detailed below:

- Leverage Teams **MeetingParticipantDetail** event type in conjunction with additional derived threat intelligence (**ClientIP**) to determine if the threat actor attempted to join Teams meetings.
- **JoinTime** and **LeaveTime** can be used to establish how long the attendee was in the meeting (useful for timelines and to establish knowledge of meeting contents).
- **MessageSent**/**MessageRead** can be used to establish interaction with and confirmation of received messages.
- The Exchange workload's **Send** event type can be used to identify a threat actor using a compromised identity/mailbox to propagate phishing messages from your organization. Use the **HasAttachments** and **Subject** fields.
- Add infrastructure used across workloads to build the behavioral profile and ascertain different teams/individuals responsible for specific goals/objectives (e.g., unique sets of infrastructure used for accessing email versus accessing Teams and interacting with meetings or messages).

Sample Queries:

Here is a Reactive Advanced Hunting query to display participation duration of account associated with a suspicious IP address:

```
CloudAppEvents
| where ActionType == 'MeetingParticipantDetail'
| extend clientip = tostring(RawEventData.ClientIP)
| extend meeting_id = tostring(RawEventData.MeetingDetailId)
| extend join_time = todatetime(RawEventData.JoinTime)
| extend leave_time = todatetime(RawEventData.LeaveTime)
| extend min_duration = datetime_diff('minute',leave_time,join_time)
| where clientip == 'x.x.x.x' //Replace with suspicious IP
| project AccountDisplayName, meeting_id, join_time,leave_time,min_duration
```

APPENDIX A: DATA DICTIONARY

This table provides a breakdown of the newly available logs, their fields, and a summary of the purpose/description value of the log and its exposed events. Fields with "." indicate a nested field within a field. For example, **Item.Id** is the parent field name and **Id** is a subfield within the Item field.

Table 2: Data Dictionary

Log	Fields Included	Purpose/Description/Value
Exchange		
Send	• AppID • ClientAppId • ClientInfoString • ClientIP • ClientIPAddress • ClientProcessName • ClientVersion • CreationTime • ExternalAccess • Id • InternalLogonType • Item.Attachments • Item.Id • Item.InternetMessageId • Item.ParentFolder.Id • Item.ParentFolder.Path • Item.SizeInBytes • Item.Subject • LogonType • LogonUserSid • MailboxGuid • MailboxOwnerSid • MailboxOwnerUPN • Operation • OrganizationId • OrganizationName • OriginatingServer • RecordType • ResultStatus • SessionId • UserId • UserKey • UserType • Version • Workload	Investigators can use the **Send** event to identify email(s) sent from specified account(s). This auditing information can help investigators identify information about email messages sent from a compromised account, including those potentially sent by a threat actor.

MailItemsAccessed	• AppId • ClientAppId • ClientInfoString • ClientIP • ClientIPAddress • CreationTime • ExternalAccess • Folders.FolderItems.ClientRequestId • Folders.FolderItems.InternetMessageId • Folders.FolderItems.Sensitivity • Folders.FolderItems.SizeInBytes • Folders.Id • Folders.Path • Id • InternalLogonType • Item.Id • Item.ParentFolder.Id • Item.ParentFolder.Name • Item.ParentFolder.Path • LogonType • LogonUserSid • MailboxGuid • MailboxOwnerSid • MailboxOwnerUPN • Operation • OperationCount • OperationProperties.Name • OperationProperties.Value • OrganizationId • OrganizationName • OriginatingServer • RecordType • ResultStatus • SessionId • UserId • UserKey • UserType • Version • Workload	This event type can help investigators identify data exfiltration and determine the scope of messages that may have been compromised. **MailItemsAccessed** is typically used during forensic investigations after an incident has been remediated and threat actor eviction has occurred. However, there are scenarios where it could be used for proactive threat hunting or alerting for anomalous mailbox behavior.
SearchQueryInitiatedExchange	• AppAccessContext.APIId • AppAccessContext.ClientAppId • AppAccessContext.CorrelationId • ClientIP • ClientUserAgent • CreationTime • CustomProperties.Name • CustomProperties.Value • Id • Operation • OrganizationId • QuerySource • QueryText • RecordType	Investigators can use the **SearchQueryInitiatedExchange** event to determine if a threat actor compromised an account searched for or tried to access sensitive information in the mailbox. This audit record contains the actual text of the search typed into the search bar. The audit record also indicates the Outlook environment the

	• ScenarioName • UserId • UserKey • UserType • Version • Workload	search was performed in (e.g., Desktop, Android, iOS).
SharePoint Online		
SearchQueryInitiatedSharePoint	• AppAccessContext.APIId • AppAccessContext.ClientAppId • AppAccessContext.CorrelationId • ClientIP • ClientUserAgent • CreationTime • CustomProperties.Name • CustomProperties.Value • Id • Operation • OrganizationId • QuerySource • QueryText • RecordType • ScenarioName • UserId • UserKey • UserType • Version • Workload	Investigators can use the **SearchQueryInitiatedSharePoint** event to determine if a threat actor searched for or tried to access information in SharePoint. The audit record for a **SearchQueryInitiatedSharePoint** event also contains the actual text of the search query and indicates the type of SharePoint site that was searched.
Microsoft Teams		
MeetingParticipantDetail	• CreationTime • Id • Operation • OrganizationId • RecordType • UserKey • UserType • Version • Workload • ClientIP • UserId • ItemName • Attendees • Attendees.OrganizationId • Attendees.RecipientType • Attendees.UserObjectId • Attendees.Role • Attendees.DisplayName • Attendees.UPN • DeviceId • ExtraProperties.Key • ExtraProperties.Value	This event type includes information about the participants of a Teams meeting including the user ID of each participant, the time a participant joined the meeting, and the time a participant left the meeting. This can be used for establishing identity-based forensic evidence.

	• IsJoinedFromLobby • JoinTime • LeaveTime • MeetingDetailId • DeviceInformation	
MessageSent	• AADGroupId • AppAccessContext.APIId • AppAccessContext.ClientAppId • AppAccessContext.ClientAppName • AppAccessContext.CorrelationId • AppAccessContext.IssuedAtTime • AppAccessContext.UniqueTokenId • ChannelGuid • ChannelName • ChannelType • ChatName • ChatTheadId • ClientIP • CommunicationType • CreationTime • ExtraProperties • ExtraProperties.Key • ExtraProperties.Value • HasUnauthenticatedUsers • Id • ItemName • MessageId • MessageSizeInBytes • MessageVersion • Operation • OperationScope • OrganizationId • ParentMessageId • ParticipantInfo.HasForeignTenantUsers • ParticipantInfo.HasGuestUsers • ParticipantInfo.HasOtherGuestUsers • ParticipantInfo.HasUnauthenticatedUsers • ParticipantInfo.ParticipatingTenantIds • RecordType • ResourceTenantId. • TargetUserId • TeamGuid • TeamName • UserId • UserKey • UserType • Version • Workload	This event type is a comprehensive record of outgoing communications, capturing detailed information about when messages are sent, by whom, to whom, along with their content and context. This makes it an invaluable tool in digital forensic investigations and helps to unravel communication patterns, identify key actors, and establish timelines in various scenarios.
MessagesListed	• AppAccessContext.APIId • AppAccessContext.ClientAppId • AppAccessContext.ClientAppName • AppAccessContext.CorrelationId • ChatThreadId	**Graph API Triggered** Teams will generate two different **MessagesListed**

	• CreationTime • Id • Operation • OrganizationId • RecordType • UserKey • UserType • Version • Workload • UserId • AADGroupId • ChannelGuid • CommunicationType • ExtraProperties • ExtraProperties.Key • ExtraProperties.Value • Messages • Messages.AADGroupId • Messages.ChannelGuid • Messages.ChatThreadId • Messages.Id • Messages.SizeInBytes • Messages.TeamGuid • Messages.Version • OperationScope • ChannelName • ItemName • TeamName	events based on the type of chat listed with the Graph API. The first type generated is messages listed from a **Chat** (one-on-one or group). The identifier of this chat is the **ChatThreadId** and the messages read are listed as the Id. These messages can be summarized with up to 20 IDs in a single **MessageListed** event. The second type generated is messages listed from a **Channel**. In this event, the **TeamGuid** and **AADGrouID** identify the team (values are the same) and the **ChannelGuid** specifies the channel. Messages are listed in the ID field; these can also be summarized with up to 20 IDs listed in a single event.
MeetingDetail	• CreationTime • Id • ClientIP • Operation • OrganizationId • RecordType • UserKey • UserType • Version • Workload • UserId • ChatThreadId • CommunicationSubType • CommunicationType • ConferenceUri • EndTime • ICalUid • MessageId • Modalities • Organizer • Organizer.OrganizationId • Organizer.RecipientType • Organizer.UserObjectId • Organizer.Role • ProviderTypes • StartTime	This event type provides specifics of a Teams meeting like duration, participants, and more. Investigators can verify attendance, meeting content, and meeting duration. It provides valuable information in scenarios where the details of a meeting such as attendee behavior, content shared, and meeting access are relevant to the investigation, whether for security incidents, compliance audits, or internal inquiries.

	• ItemName • MeetingURL • ResultIndex • ResultCount • Identity • QueryGuid • RunspaceId • ExtraProperties.Key • ExtraProperties.Value	
MessageUpdated	• AppAccessContext.IssuedAtTime • AppAccessContext.UniqueTokenId • CreationTime • Id • Operation • OrganizationId • RecordType • UserType • Version • Workload • ClientIP • UserId • AADGroupId • ChennelGuid • ExtraProperties • ExtraProperties.Key • ExtraProperties.Value • MessageId • MessageVersion • ParentMessageId • ParticipantInfo.HasForeignTenantUsers • ParticipantInfo.HasGuestUsers • ParticipantInfo.HasOtherGuestUsers • ParticipantInfo.HasUnauthenticatedUsers • ParticipantInfo.ParticipatingTenantIds • ResourceTenantId • TeamGuid • ChannelName • ItemName • TeamName • ChatThreadId	This event type provides essential insights into the modifications made to communications and helps to establish timelines, determine accountability, and understand the dynamics and integrity of conversations within an organization.
ChatRetrieved	• AppAccessContext • AppAccessContext.APIId • AppAccessContext.ClientAppId • AppAccessContext.ClientAppName • AppAccessContext.CorrelationId • CreationTime • Id • Operation • OrganizationId • RecordType • UserKey • UserType • Version	Graph API Triggered A Microsoft Teams chat was retrieved.

	• Workload • UserId • ChatThreadId • CommunicaitonType • ExtraProperties • ExtraProperties.Key • ExtraProperties.Value • OperationScope • ChatName • ItemName	
MessageRead	• AppAccessContext.APIId • AppAccessContext.ClientAppId • AppAccessContext.ClientAppName • AppAccessContext.CorrelationId • ChatThreadId • CommunicationType • CreationTime • ExtraProperties • Id • ItemName • MessageId • MessageSizeInBytes • MessageVersion • MessageVisibilityTimer • Operation • OperationScope • OrganizationId • ParticipantInfo • ParticipantInfo.HasForeignTenantUsers • ParticipantInfo.HasGuestUsers • ParticipantInfo.HasOtherGuestUsers • ParticipantInfo.HasUnauthenticatedUsers • ParticipantInfo.ParticipatingDomains • ParticipantInfo.ParticipatingSIPDomains • ParticipantInfo.ParticipatingTenantIds • RecordType • ResourcetenantId • UserId • UserKey • UserType • Version • Workload	A message of a chat or channel was retrieved.
MessageHostedContentRead	• AppAccessContext.APIId • AppAccessContext.ClientAppId • AppAccessContext.ClientAppName • AppAccessContext.CorrelationId • CreationTime • Id • Operation • OrganizationId • RecordType • UserKey • UserType	Graph API Triggered All hosted content in a message (e.g., images or code snippets) was retrieved.

	• Version • Workload • UserId • ChatThreadId • CommunicationType • ExtraProperties • HostedContents • HostedContents.Id • HostedContents.SizeInBytes • MessageId • OperationScope • ChatName • ItemName	
SubscribedToMessages	• AppAccessContext • AppAccessContext.APIId • AppAccessContext.ClientAppId • AppAccessContext.ClientAppName • AppAccessContext.CorrelationId • CreationTime • Id • Operation • OrganizationId • RecordType • UserKey • UserType • Version • Workload • UserId • ExtraProperties • ExtraProperties.Key • ExtraProperties.Value • OperationScope • SubscriptionId • ItemName	Graph API Triggered A subscription was created by a listener application to receive change notifications for messages.
MessageHostedContents Listed	• AADGroupId • AppAccessContext.APIId • AppAccessContext.ClientAppId • AppAccessContext.ClientAppName • AppAccessContext.CorrelationId • ChannelGuid • ChatThreadId • CommunicationType • CreationTime • ExtraProperties • HostedContents.Id • HostedContents.SizeInBytes • Id • ItemName • MessageId • Operation • OperationScope • OrganizationId • RecordType	Graph API Triggered All hosted content in a message (e.g., images or code snippets) was retrieved.

	• TeamName • UserId • UserKey • UserType • Version • Workload	
ChatCreated	• AppAcessContext • AppAccessContext.APIId • AppAccessContext.ClientAppId • AppAccessContext.ClientAppNam • AppAccessContext.CorrelationId • CreationTime • Id • Operation • OrganizationId • RecordType • UserKey • UserType • Version • Workload • UserId • ChatThreadId • CommunicationType • OperationScope • ItemName	**Graph API Triggered** When the GraphAPI is used to create a chat (one-on-one or group), **ChatCreated** triggers. (https://learn.microsoft.com/en-us/graph/api/chat-post?view=graph-rest-beta&tabs=http) The **AppAccessContext.AppId** field will be "00000003-0000-0000-c000-000000000000". The **AppAccessContext.ClientAppName** will indicate the method of access used (e.g., Graph Explorer). This event helps determine if a threat actor is creating chats or interacting with users in your environment via Graph API.
ChatUpdated	• AppAcessContext • AppAccessContext.APIId • AppAccessContext.ClientAppId • AppAccessContext.ClientAppNam • AppAccessContext.CorrelationId • CreationTime • Id • Operation • OrganizationId • RecordType • UserKey • UserType • Version • Workload • UserId • ChatThreadId • CommunicationType • OperationScope • ChatName • ItemName • NewValue	**Graph API Triggered** This event is triggered when the Graph API is used to update a chat (https://learn.microsoft.com/en-us/graph/api/chat-patch?view=graph-rest-beta&tabs=http). This method can only be used on Group chats. The **AppAccessContext.AppId** field will be "00000003-0000-0000-c000-000000000000". The **AppAccessContext.ClientAppName** will indicate the method of access used (e.g., Graph Explorer).

		All of the properties of chatMessage can be updated in delegated permissions scenarios except for the policyViolation property and read-only properties. The policyViolation property is the only property that can be updated in application permissions scenarios. This event can be useful for determining threat-actor interactions in your environment if properties like the body, attachments, reactions, timestamps, or subject are manipulated or changed.
MessageCreatedNotification	• AppAcessContext • AppAccessContext.APIId • AppAccessContext.ClientAppId • AppAccessContext.ClientAppNam • AppAccessContext.CorrelationId • CreationTime • Id • Operation • OrganizationId • RecordType • UserKey • UserType • Version • Workload • UserId • ChatThreadId • ExtraProperties • ExtraProperties.Key • ExtraProperties.Value • MessageId • MessageSizeInBytes • MessageVersion • OperationScope • SubscriptionId • ChatName • ItemName	Graph API Triggered A change notification was sent to notify a subscribed listener application of a new message.
MessageDeletedNotification	• AppAcessContext • AppAccessContext.APIId • AppAccessContext.ClientAppId • AppAccessContext.ClientAppNam • AppAccessContext.CorrelationId • CreationTime • Id	Graph API Triggered A change notification was sent to notify a subscribed listener

	• Operation • OrganizationId • RecordType • UserKey • UserType • Version • Workload • UserId • ChatThreadId • ExtraProperties • ExtraProperties.OriginEnvironment • MessageId • MessageSizeInBytes • MessageVersion • OperationScope • SubscriptionId • ChatName • ItemName	application of a deleted message.
MessageUpdatedNotification	• AppAcessContext • AppAccessContext.APIId • AppAccessContext.ClientAppId • AppAccessContext.ClientAppNam • AppAccessContext.CorrelationId • CreationTime • Id • Operation • OrganizationId • RecordType • UserKey • UserType • Version • Workload • UserId • ChatThreadId • ExtraProperties • ExtraProperties.Key • ExtraProperties.Value • MessageId • MessageSizeInBytes • MessageVersion • OperationScope • SubscriptionId • ChatName • ItemName	**Graph API Triggered** A change notification was sent to notify a subscribed listener application of an updated message.

APPENDIX B: ADDITIONAL RESOURCES

Table 3: Additional Resources

#	Title / URL / Summary
1.	**CISA Cybersecurity Advisory, "Enhanced Monitoring to Detect APT Activity Targeting Outlook Online," July 12, 2023** URL: https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-193a Summary: In June 2023, a Federal Civilian Executive Branch (FCEB) agency identified suspicious activity in their Microsoft 365 (M365) cloud environment. The agency reported the activity to Microsoft and the Cybersecurity and Infrastructure Security Agency (CISA); subsequently, Microsoft determined that advanced persistent threat (APT) actors accessed and exfiltrated unclassified Exchange Online Outlook data. CISA and the Federal Bureau of Investigation (FBI) released this joint Cybersecurity Advisory to provide guidance to critical infrastructure organizations on enhancing monitoring of Microsoft Exchange Online environments.
2.	**Microsoft blog post, "Analysis of Storm-0558 techniques for unauthorized email access," July 14, 2023** URL: https://www.microsoft.com/en-us/security/blog/2023/07/14/analysis-of-storm-0558-techniques-for-unauthorized-email-access/ Summary: On July 11, 2023, Microsoft published two blogs detailing a malicious campaign by a threat actor, known as Storm-0558 that targeted customer email. This malicious activity was detected and mitigated. Microsoft detailed its deployment of defense-in-depth measures to harden all systems involved and the deeper analysis of the observed actor techniques for obtaining unauthorized access to email data, tools, and unique infrastructure characteristics.
3.	**Microsoft blog post, "Expanding cloud logging to give customers deeper security visibility," July 19, 2023** URL: https://www.microsoft.com/en-us/security/blog/2023/07/19/expanding-cloud-logging-to-give-customers-deeper-security-visibility/ Summary: In response to the increasing frequency and evolution of nation-state cyber threats, Microsoft is taking additional steps to protect our customers and increase the secure-by-default baseline of our cloud platforms. These steps are the result of close coordination with commercial and government customers, and with the Cybersecurity and Infrastructure Security Agency (CISA) about the types of security log data Microsoft provides to cloud customers for insight and analysis.
4.	**Microsoft blog post "Expanding audit logging and retention within Microsoft Purview for increased security visibility," October 18, 2023** URL: https://www.microsoft.com/en-us/security/blog/2023/10/18/expanding-audit-logging-and-retention-within-microsoft-purview-for-increased-security-visibility/ Summary: Since our announcement in July 2023, we have made significant efforts to enhance the access to Microsoft Purview's audit logging. This ongoing work expands accessibility and flexibility to cloud security logs, which began rolling out to customers around the world in September 2023. Our decision to update the scope of log data accessible from Microsoft's cloud infrastructure resulted from a close collaboration with both commercial and government customers, as well as ongoing engagement with the Cybersecurity and Infrastructure Security Agency (CISA).
5.	**CISA press release, "CISA, OMB, ONCD and Microsoft Efforts Bring New Logging Capabilities to Federal Agencies," February 21, 2024** URL: https://www.cisa.gov/news-events/news/cisa-omb-oncd-and-microsoft-efforts-bring-new-logging-capabilities-federal-agencies

	Summary: The Cybersecurity and Infrastructure Security Agency (CISA), Office of Management and Budget (OMB), Office of the National Cyber Director (ONCD), and Microsoft announce today further progress in ensuring that Federal Civilian Executive Branch (FCEB) agencies have access to necessary logging capabilities. Over the past six months, Microsoft has worked closely with CISA, OMB, and ONCD to roll out expanded logs to a pilot group of agencies. Beginning this month, expanded logging will be available to all agencies using Microsoft Purview Audit regardless of license tier.
6.	**Microsoft blog post, "CISA, OMB, ONCD and Microsoft collaborate on new logging playbook for Federal agencies," Feb 21, 2024** URL: https://techcommunity.microsoft.com/t5/public-sector-blog/cisa-omb-oncd-and-microsoft-collaborate-on-new-logging-playbook/ba-p/4063661 Summary: As part of our efforts to increase security defaults and follow the principle of secure by design, we are happy to share that a feature change initiated by Microsoft engineering will enable more logging capabilities for Purview Audit (Standard). We have worked closely with the Executive Office of the President (EOP), the Office of the National Cyber Director (ONCD), and the Cybersecurity and Infrastructure Security Agency (CISA) to prioritize this effort for U.S. government customers. This data will provide new telemetry to assist in meeting OMB 21-31 logging requirements for customers without E5 capability.
7.	**CISA publication, "Cyber Safety Review Board Releases Report on Microsoft Online Exchange Incident from Summer 2023," March 20, 2024** URL: https://www.cisa.gov/sites/default/files/2024-04/CSRB_Review_of_the_Summer_2023_MEO_Intrusion_Final_508c.pdf Summary: In May and June 2023, a threat actor—known as Storm-0558 and affiliated with the People's Republic of China—compromised the Microsoft Exchange Online mailboxes of 22 organizations and over 500 individuals around the world in pursuit of espionage objectives. This intrusion compromised senior United States government representatives working on national security matters.
8.	**CISA CyberStat Capacity Building Workshop, "Operationalizing Microsoft's Expanded Cloud Logs to Detect Advanced Attacks," May 29, 2024** URL: https://community.connect.gov/pages/viewpage.action?pageId=2499646656 Summary: Representatives from CISA, ONCD, and Microsoft presented a one-hour workshop on Microsoft's expanded cloud logs. Workshop topics included a detailed overview of the expanded cloud logs, steps for getting started and enabling the logs, Splunk and Microsoft Sentinel SIEM integration, and operational scenarios for using the logs to detect advanced adversarial behavior.

APPENDIX C: ENABLING LOGS IN MICROSOFT SENTINEL

To enable the logs in Microsoft Sentinel, there are two possible avenues. Each is described in detail below.

Enable Logs in Sentinel via Microsoft 365 Connector

The Microsoft 365 Connector in Sentinel is a quick method for capturing a large percentage of the logs discussed in this playbook in a few simple steps. It should be noted, at the time of this writing, this connector does not capture the **QueryText** data from **SeachQueryInitiated** logs (Exchange or SharePoint). This field represents the actual search term entered in the search bar in SharePoint or Exchange. The connector does log that a search was performed, providing the workload (SharePoint or Exchange), timestamp, **UserID**, **ClientIP**, and other identifying information. To configure the connector, follow the steps provided below.

1. In Sentinel in the Configuration section, click Data Connectors.
2. In the search bar type “Microsoft 365.”
3. Enable the connector.
4. With the Microsoft 365 connector highlighted, click Open Connector Page.
5. In the Instructions tab, under the Configuration section, check the boxes for Exchange, SharePoint, and Teams and click Apply Changes as shown in Figure 5 below.

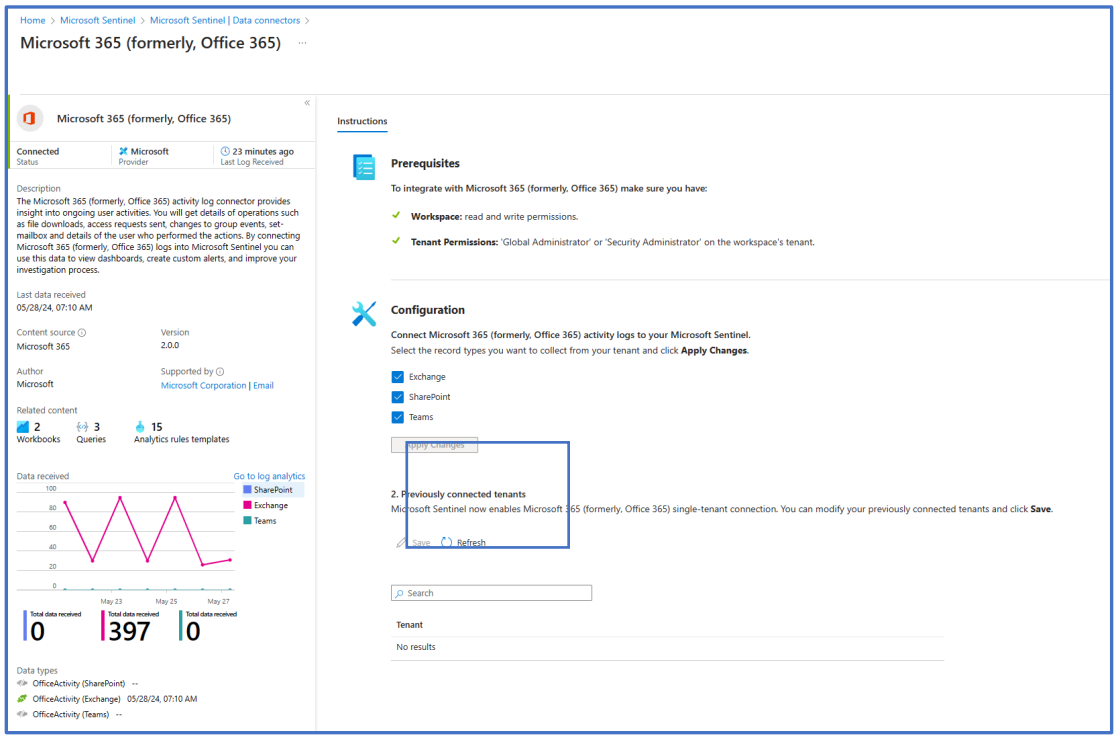


Figure 5: Configuring Microsoft 365 Connector

6. The logs ingested by this connector will be populated in the **OfficeActivity** table within Sentinel
    a. The Operation field categorizes the logged events by type and can be used to identify the logs discussed in this playbook.
    b. The KQL query below can be used to quickly ascertain what events are being captured.

```
OfficeActivity
| summarize count() by Operation
```

Enable Sentinel integration in Microsoft Defender XDR

At the time of this writing, Sentinel and Microsoft Defender XDR integration is available as a Public Preview in commercial environments, with support for government environments scheduled to follow shortly after. This integration facilitates collocating Security Information and Event Management (SIEM) and Extended Detection and Response (XDR) data, providing analysts with the ability to query security data from multiple sensor locations (EDR, Office365, Cloud Apps, etc.) in a single location. To enable this functionality, it must be configured both in Sentinel and in Microsoft Defender XDR. The instructions below provide guidance on how to complete both.

1. Configure Defender XDR Connector in Sentinel.
    a. In Sentinel in the Configuration section, click Data Connectors.
    b. In the search bar type “Defender XDR.”
    c. Enable the connector.
    d. Open connector page.
    e. In the Configuration section click Connect Incidents and Alerts.

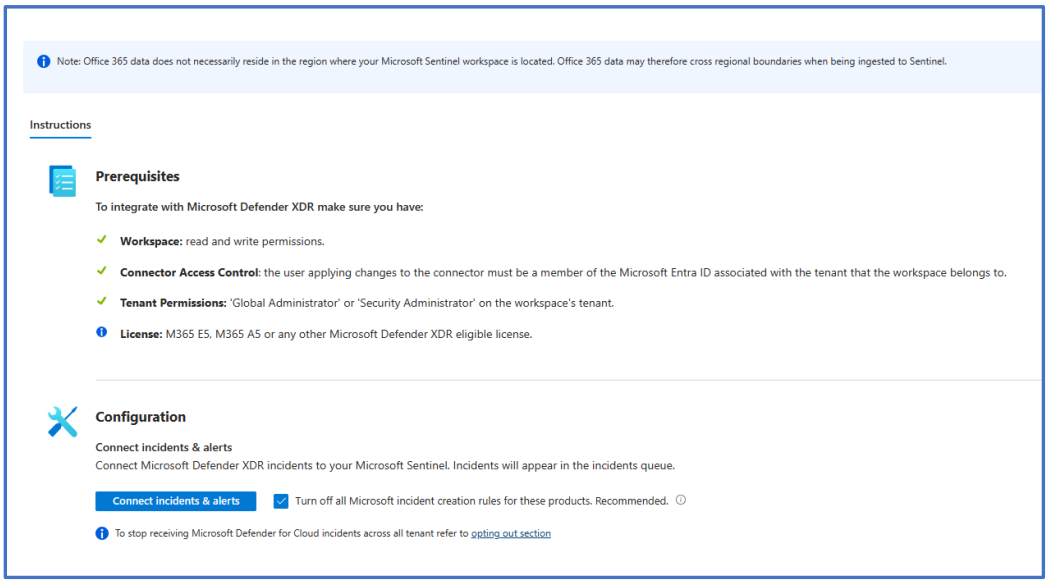


Figure 6: Configure XDR Connector in Sentinel

    f. In the Configuration section under Connect Events, click the drop down for Defender for Cloud Apps and check the box for **CloudAppEvents**. Note: connecting other data sources will provide increased visibility of XDR data in Sentinel, but for the purposes of this playbook, only **CloudAppEvents** is required.

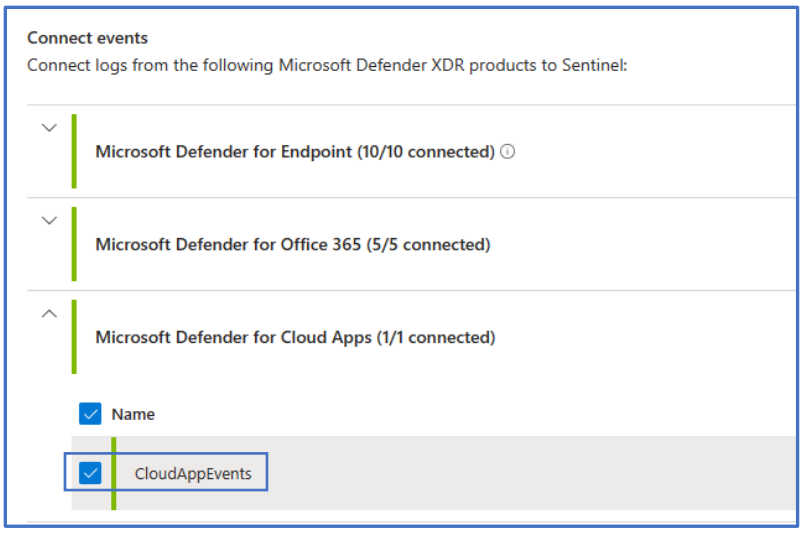


Figure 7: Enable CloudAppEvents

2. Connect Sentinel Workspace in Defender XDR
    a. In the Microsoft Defender portal (https://security.microsoft.com) under the System section. select Settings.
    b. Select Microsoft Sentinel.
    c. In the list of available Log Analytics Workspaces, select the check box next to the one which you wish to connect to XDR (currently only one connection is supported).
    d. Click Connect Workspace and, when the acknowledgement box appears, select Confirm and proceed.
3. The logs ingested by this connector will appear in the **CloudAppEvents** table in both Sentinel and Defender XDR (i.e., advanced hunting).
    a. The **ActionType** field categorizes the logged events by type and can be used to identify the logs discussed in this playbook.
    b. The KQL query below can be used to quickly ascertain what events are being captured.

```
CloudAppEvents
| summarize count() by ActionType
```

    c. Note: The **RawEventData** section contains the field **QueryText**, which captures the terms entered in the search bars of Exchange and SharePoint for the **SearchQueryInitiated** logs discussed throughout this playbook. With this connection configured, this data is available both in Advanced Hunting in the Microsoft Defender portal (https://security.microsoft.com) and in Microsoft Sentinel in the **CloudAppEvents** table.

APPENDIX D: INTEGRATING SPLUNK AND MICROSOFT OFFICE 365

Integrating Splunk with Azure and Office 365 connects Splunk's powerful data analysis capabilities with Azure and Office 365's productivity tools. This integration collects data from sources such as Exchange Online, Microsoft Entra ID, Azure Security Center, Microsoft Sentinel, Microsoft Defender for Endpoint, Microsoft Teams, and more. Integrating Splunk with Azure and Office 365 can be accomplished using Splunk's technical add-ons in concert with configuration settings implemented in Azure and Office 365. With this information, Splunk can monitor and analyze activities in real time or through historical analysis, helping detect anomalies, identify trends, and gain visibility into user activities, system performance, security incidents, and compliance issues.

Alongside this analysis, Splunk provides customizable dashboards and reports, visualizing data to display user login trends, email traffic patterns, file access activities, security incidents, and compliance statuses. It can also generate alerts based on predefined conditions or anomalies, triggering automated responses or notifications to IT teams for proactive resolution.

Through this integration, organizations can enhance their compliance and governance capabilities and monitor access controls, data retention policies, user behavior, and adherence to regulatory requirements like the Health Insurance Portability and Accountability Act of 1996 (HIPAA). This comprehensive approach optimizes the performance, security, and compliance of an organization's Azure cloud ecosystem, enhancing visibility, enabling proactive monitoring, and supporting informed decision-making for IT administrators and security teams. Several key aspects of integrating Splunk with Office 365 and Azure are:

1. Data Collection**:** Splunk collects data from various Office 365 and Azure sources such as Exchange Online, SharePoint Online, OneDrive for Business, Microsoft Entra ID, and Microsoft Teams. This data can include logs, events, usage metrics, and audit trails.
2. Monitoring and Analysis**:** Once the data is collected, Splunk can monitor and analyze it in real time or through historical analysis. This allows organizations to detect anomalies, identify trends, and gain visibility into user activities, system performance, security incidents, and compliance issues within their Office 365 and Azure environment.
3. Dashboards and Reporting**:** Splunk provides customizable dashboards and reports that visualize the collected data in meaningful ways. These dashboards can display information such as user login trends, email traffic patterns, file access activities, security incidents, and compliance status.
4. Alerting and Automation**:** Splunk can be configured to generate alerts based on predefined conditions or anomalies detected in Office 365 and Azure data. These alerts can trigger automated responses or notifications to IT and/or cybersecurity teams, helping them proactively address issues or security threats.
5. Compliance and Governance**:** By integrating Splunk with Office 365 and Azure, organizations can enhance their compliance and governance capabilities. Splunk can assist in monitoring access controls, data retention policies, user behavior, and adherence to regulatory requirements such as HIPAA.

Enabling logging between Splunk and Microsoft Office 365 can be accomplished through two primary Splunk add-ons: Splunk Add-on for Microsoft Office 365 and Microsoft Graph Security API Add-On for Splunk. These plugins allow for collection of a broad and encompassing data set that can exponentially boost security operation center visibility into the Microsoft Office and Azure environments.

SPLUNK ADD-ON FOR OFFICE 365

The Splunk Add-on for Microsoft Office 365 allows a Splunk software administrator to pull service status, service messages, and management activity logs from the Office 365 Management Activity API and the Office 365 Service Communications API. You can collect:

- Audit logs for Microsoft Entra ID, SharePoint Online, and Exchange Online, supported by the Office 365 Management API. For more details, see the Office 365 Management Activity API reference on the Microsoft website.
- Historical and current service status and service messages for the corresponding Office 365 Service Communications API.
- Data loss prevention events via the Office 365 Management Activity API.
- Message trace events via the Office 365 Message Trace Report API.

Steps to Enablement

Personnel: Splunk Admin, O365 Admin; **Technical Requirements:**
https://docs.splunk.com/Documentation/AddOns/released/MSO365/Hardwareandsoftwarerequirements

High-level steps to install the Splunk Add-on for Office 365:

1. Download the Splunk Add-on for Microsoft Office 365 from Splunkbase.
2. Determine where and how to install this in your deployment.
3. Perform any prerequisite steps before installing.
4. Complete your installation.

During development of this section and consultation with other Federal Civilian Executive Branch (FCEB) constituents, the following challenges were shared:

- Permissions for the API account were not adequately explained before add-on installation.
- Azure Government Cloud (GCC) requires a separate tenant for EventHub to collect logs from.
- Logs extracted from this add-on require normalization efforts and additional tuning before incorporating into their correlation logic in Splunk.
- This add-on feed does not include MailOpen events. These events must be searched manually in the Azure/Office 365 portal.

Splunkbase - The add-on can be found for download with associated documentation here:
https://splunkbase.splunk.com/app/4055

Documentation – Additional documentation is available here:
https://docs.splunk.com/Documentation/AddOns/released/MSO365/About

How Can I Use This?

The following are potential use cases for the Splunk Add-on for Microsoft Office 365:

Log Monitoring & Troubleshooting Query

The query below is a general-purpose query that summarizes the events received from the Splunk Add-on for Microsoft Office 365. This query extracts all events available by the add-on feed, categorizes them by workload type, highlights whether the logs have been received, and provides a count of how many logs have been received. This query can be extremely helpful in troubleshooting the log feed itself or identifying when an issue occurs with the feed, such as a drop in connectivity.

```
index="your_index_for_o365_data" earliest=-30d@d latest=now
    (Operation IN ("MailItemsAccessed", "Send","SearchQueryInitiatedExchange") OR
Operation IN ("SearchQueryInitiatedSharePoint") OR Operation IN
("MeetingParticipantDetail", "MessageSent", "MessagesListed", "MeetingDetail",
"MessageUpdated", "ChatRetrieved", "SubscribedToMessages", "MessageRead", "ChatCreated",
"MessageCreatedNotification", "MessageUpdatedNotification", "MessageHostedContentRead",
"MessageHostedContentsListed", "ChatUpdated", "MessageDeletedNotification"))
| fields Operation, Workload
| stats count by Operation, Workload
| eval operation_status = "identified"
| append
    [| makeresults
    | eval Operation =
split("MailItemsAccessed,Send,SearchQueryInitiatedExchange,SearchQueryInitiatedSharePoin
t,MeetingParticipantDetail,MessageSent,MessagesListed,MeetingDetail,MessageUpdated,ChatR
etrieved,SubscribedToMessages,MessageRead,ChatCreated,MessageCreatedNotification,Message
UpdatedNotification,MessageHostedContentRead,MessageHostedContentsListed,ChatUpdated,Mes
sageDeletedNotification", ",")
| mvexpand Operation
| fields - _time]
| stats values(count) as num_events, values(operation_status) as operation_status,
values(Workload) as Workload by Operation
| fields Operation, Workload, operation_status, num_events
| sort 0 Workload
| eval
    Workload = case(Operation="MailItemsAccessed", "Exchange",
    Operation="SearchQueryInitiatedExchange", "Exchange",
    Operation="Send", "Exchange",
    Operation="ChatCreated", "MicrosoftTeams",
    Operation="ChatRetrieved", "MicrosoftTeams",
    Operation="MeetingDetail", "MicrosoftTeams",
    Operation="MeetingParticipantDetail", "MicrosoftTeams",
    Operation="MessageCreatedNotification", "MicrosoftTeams",
    Operation="MessageRead", "MicrosoftTeams",
    Operation="MessageSent", "MicrosoftTeams",
    Operation="MessageUpdated", "MicrosoftTeams",
    Operation="MessageUpdatedNotification", "MicrosoftTeams",
    Operation="MessagesListed", "MicrosoftTeams",
    Operation="SubscribedToMessages", "MicrosoftTeams",
    Operation="SearchQueryInitiatedSharePoint", "SharePoint",
    Operation="ChatUpdated", "MicrosoftTeams",
    Operation="MessageDeletedNotification", "MicrosoftTeams",
    Operation="MessageHostedContentRead", "MicrosoftTeams",
    Operation="MessageHostedContentsListed", "MicrosoftTeams", 1==1, null()),
    operation_status = case(isnull(operation_status) OR len(operation_status)<=0, "not
identified in data", 1==1, operation_status)
| fillnull value="0" num_events
      | rename num_events as "# of Events", operation_status as "Operation Status"
```

The image below demonstrates the output of this query:

100 Per Page ▾ | Format | Preview ▾

Operation	Workload	Operation Status	# of Events
MailItemsAccessed	Exchange	identified	98799316
SearchQueryInitiatedExchange	Exchange	identified	1806924
Send	Exchange	identified	5717641
ChatCreated	MicrosoftTeams	identified	2113
ChatRetrieved	MicrosoftTeams	identified	20
MeetingDetail	MicrosoftTeams	identified	11
MeetingParticipantDetail	MicrosoftTeams	identified	107
MessageCreatedNotification	MicrosoftTeams	identified	36796
MessageRead	MicrosoftTeams	identified	159
MessageSent	MicrosoftTeams	identified	303050
MessageUpdated	MicrosoftTeams	identified	95156
MessageUpdatedNotification	MicrosoftTeams	identified	3536
MessagesListed	MicrosoftTeams	identified	4576
SubscribedToMessages	MicrosoftTeams	identified	488
SearchQueryInitiatedSharePoint	SharePoint	identified	195201
ChatUpdated	MicrosoftTeams	not identified in data	0
MessageDeletedNotification	MicrosoftTeams	not identified in data	0
MessageHostedContentRead	MicrosoftTeams	not identified in data	0
MessageHostedContentsListed	MicrosoftTeams	not identified in data	0

Figure 8: Splunk Troubleshooting Query Output

Authentication, SharePoint Usage, and One-Drive Usage Use-Cases

Authentication

Disclaimer: The Authentication data model doesn't work perfectly for monitoring Entra ID authentication because there are several messages that register authorization issues as authentication failures. There is no detail in the data-model record to allow us to filter out the failure/error messages "InvalidReplyTo," "SsoArtifactExpiredDueToConditionalAccess," and "BlockedByConditionalAccess," which leads to many false positives. The following Splunk search processing language can be used to filter out these failure/error messages.

```
sourcetype=o365:management:activity eventtype="o365_authentication"
action=failure NOT LogonError IN ("InvalidReplyTo",
"SsoArtifactExpiredDueToConditionalAccess", "BlockedByConditionalAccess")
NOT user IN (service_test_user@nosuchagency.gov)
| eval l_user=lower(user)
| stats min(_time) AS FT, max(_time) AS LT, values(LogonError) AS errors,
values(src) as src, count by l_user
| where count > 7
| eval first = strftime(FT, "%Y-%m-%dT%H:%M:%S")
| eval last = strftime(LT, "%Y-%m-%dT%H:%M:%S")
| fields l_user errors src first last count
```

OneDrive and SharePoint activity

Malware Detections

This shows alerts by O365 from malware detection of a static file.

```
sourcetype="o365:management:activity" Workload IN ("OneDrive","SharePoint")
Operation=FileMalwareDetected
| table _time, VirusInfo, Workload, ObjectId
```

Distinct Files Modified

This query supports identification of large numbers of file modifications. Alerts triggered by this logic are suspected to be indicative of a cloud-savvy threat actor attempting to destroy or encrypt file contents. This query regularly triggers false positives when mass renaming of files is performed by authorized users.

```
sourcetype="o365:management:activity" Workload IN
("SharePoint","OneDrive") Operation=FileModified | stats sparkline count
AS Total_Changes, dc(file_name) AS Distinct_Files_Changed by UserId
|  where Distinct_Files_Changed>=50
|  sort -Distinct_Files_Changed
```

Large Number of Files Accessed

This query supports identification of unusually large numbers of distinct file reads or downloads. This experimental query is an attempt to identify insider threats or compromised credentials attempting file exfiltration. This query regularly triggers false positives when local drives sync to the cloud. Caution is advised when implementing this query as it has the potential to become overwhelming.

```
sourcetype="o365:management:activity" Workload IN
("SharePoint","OneDrive") Operation=FileAccessed UserId!=app@sharepoint
| stats sparkline count AS Total_Accessed, dc(file_name) AS
Distinct_Files_Accessed by UserId
|  where Distinct_Files_Accessed >=200
|  sort -Distinct_Files_Accessed
```

SharePoint Top Activity

This query identifies the top 10 highest-activity users of SharePoint. This query provides mostly supplemental information and is predominately used during investigation of anomalies as it can identify regular, potentially known-good user accounts. Sudden changes to this top 10 list could be used to support identification of compromise. The “app@sharepoint” user ID is excluded because it is in constant use within the infrastructure. There is potential to model normal activity using trending/baselining techniques, but this is not available currently.

```
sourcetype="o365:management:activity" Workload=SharePoint
UserId!="app@sharepoint"
| timechart span=1d count by UserId useother=f where max in top10
```

OneDrive Top Activity

This query, much like the previous one, identifies the top 10 highest-activity users of OneDrive. This query provides mostly supplemental information and is predominately used during investigation of anomalies as it can identify regular, potentially known-good user accounts. Sudden changes to this top 10 list could be used to support identification of compromise. There is potential to model normal activity using trending/baselining techniques, but this is not available currently.

```
sourcetype="o365:management:activity" Workload=OneDrive | timechart
span=1d count by UserId useother=f where max in top10
```

Table 4: Additional Splunk Add-on Resources

Technical Add-On	Description / Highlights	URL
Splunk Add-on for Microsoft Cloud Services	Provides preconfigured inputs for data sources to ingest security alerts and events. Collects security event data from various MS cloud services.	https://splunkbase.splunk.com/app/3110
Microsoft Teams Add-on for Splunk	Provides access to call record data and collects aggregate statistics, including network metrics, call quality telemetry, and participant and device information. Collects call telemetry and meeting information from MS Teams.	https://apps.splunk.com/app/5818/
Splunk Add-on for Microsoft Azure	Provides visibility into Azure resource utilization, performance metrics, security incidents, and compliance events. Collects logs and metrics data from Azure services.	https://splunkbase.com/app/3757/
Splunk Add-on for Microsoft Security	Provides visibility into incidents and related information from MS 365 Defender and alerts from MS Defender for Endpoint. Also collects simulation data from MS Defender for Endpoint and MS 365 Defender Advanced Hunting events data from Azure Event Hubs. Collects security relevant information and simulation data from MS 365 Defender, Defender for Endpoint.	https://splunkbase.splunk.com/app/6207

MICROSOFT GRAPH SECURITY API ADD-ON

The Microsoft Graph Security API Add-On allows Splunk users to ingest all security alerts for their organization using the Microsoft Graph Security API. Supported products include Microsoft Defender for Identity, Microsoft Entra ID Protection, Azure Security Center, Microsoft Sentinel, Azure Information Protection, Microsoft Defender for Cloud Apps, Microsoft Defender for Office 365, Microsoft Defender for Endpoint, and many more.

title	severity	category	description	vendorInformation.provider
Creation of forwarding/redirect rule	informational	ThreatManagement	This alert is triggered when someone in your organization sets up auto-forwarding, email forwarding, redirect rule or a mail flow rule - V1.0.0.5	Office 365 Security and Compliance
Email sending limit exceeded	medium	ThreatManagement	User has exceeded their email sending limit and the action defined within the Outbound Spam policy has been applied. -V1.0.0.0	Office 365 Security and Compliance
Unusual volume of external file sharing	medium	DataGovernance	This alert is triggered when the volume of external file sharing activities in your organization becomes unusual -V1.0.0.1	Office 365 Security and Compliance

Figure 9: Graph API Add-On Output

Security alerts ingested through this add-on are mapped to the Splunk Common Information Model, which allows you to easily integrate the alerts into your existing processes and dashboards.

Steps to Enablement

Personnel: Splunk Admin, Azure Admin; **Technical Requirements:** Splunk with access to Azure environment, Azure environment

High-level steps to install the Microsoft Graph Security API Add-On for Splunk:

1. Determine where and how to install this add-on in your deployment.
2. Register a new application for the Splunk add-on.
3. Install the add-on.
4. Configure Microsoft Graph Security data inputs.
5. *Optional:* Configure proxy settings.

During development of this section and consultation with other FCEB constituents, the following challenges were shared:

- Logs extracted from this add-on required normalization efforts and additional tuning before incorporation into their correlation logic in Splunk.

Additional Information

Splunkbase – The add-on can be found for download with associated documentation here: https://splunkbase.splunk.com/app/4564

Graph API Documentation – Documentation for Microsoft Graph Security can be found here: https://learn.microsoft.com/en-us/graph/security-concept-overview

APPENDIX E: INTEGRATING SPLUNK WITH AZURE & SENTINEL

Two key technical add-ons are required to integrate Splunk with Azure and Sentinel--Splunk Add-on for Microsoft Cloud Services and the Splunk Add-on for Microsoft Azure. The Splunk Add-on for Microsoft Cloud Services integrates with Event Hubs, storage accounts, and the activity log. The Microsoft Azure Add-on for Splunk integrates with various REST Application Programming Interfaces (APIs). Notice that the Splunk Add-on for Microsoft Cloud Services can get the activity log via the REST API or Event Hub.

SPLUNK ADD-ON FOR MICROSOFT CLOUD SERVICES

The Splunk Add-on for Microsoft Cloud Services allows a Splunk software administrator to pull activity logs, service status, operational messages, Azure audit, Azure resource data and Azure Storage Table and Blob data from a variety of Microsoft cloud services using Event Hubs, Azure Service Management APIs, and Azure Storage API.

This add-on provides the inputs and Common Information Model (CIM) knowledge to use with other Splunk apps such as Splunk Enterprise Security, Splunk App for Payment Card Industry (PCI) compliance, and Splunk IT Service intelligence.

Steps to Enablement – High-level steps to install the Splunk Add-on for Microsoft Cloud Services:

1. Get the Splunk Add-on for Microsoft Cloud Services by downloading it from Splunkbase or browsing to it using the app browser within Splunk Web.
2. Determine where and how to install this add-on in your deployment.
3. Perform any prerequisite steps before installing, if required.
4. Complete your installation.

Personnel: Splunk Admin, O365 Admin; **Technical Requirements:** https://splunkbase.splunk.com/app/3110, https://splunk.github.io/splunk-add-on-for-microsoft-cloud-services/

SPLUNK ADD-ON FOR MICROSOFT AZURE

The Splunk Add-on for Microsoft Azure integrates data from Microsoft Azure into Splunk, providing extensive monitoring, analysis, and operational insights. This powerful add-on provides a comprehensive solution for collecting, monitoring, and analyzing data from Azure services. It enhances visibility, security, compliance, and operational efficiency, supporting informed decision-making and proactive management of Azure resources and security incidents.

This add-on collects data from Microsoft Azure, including the following: Azure AD Data, Azure Log Analytics (KQL), metrics, estimated billing and consumption, inventory metadata, Azure Security Center, and Azure Resource Graph.

Steps to Enablement – High-level steps to install the Splunk Add-on for Microsoft Azure:

1. Get the Splunk Add-on for Microsoft Azure by downloading it from Splunkbase or browsing to it using the app browser within Splunk Web.
2. Determine where and how to install this add-on in your deployment.
3. Perform any prerequisite steps before installing, if required.

Personnel: Splunk Admin, O365 Admin; **Technical Requirements:** https://splunkbase.splunk.com/app/3757, https://github.com/splunk/splunk-add-on-microsoft-azure

APPENDIX F: FREQUENTLY ASKED QUESTIONS

1. **Is [SearchQueryInitiatedExchange] going to always require enabling Audit (Premium), as documented here (date March 2024):** https://learn.microsoft.com/en-us/purview/audit-get-started?

   Yes. You must enable two Audit (Premium) events (**SearchQueryInitiatedExchange** and **SearchQueryInitiatedSharePoint**) to be logged when users perform searches in Exchange Online and SharePoint Online.

2. **Isn't the new audit log age limit up to 180 for standard audit?**

   Yes. At a minimum, Microsoft will log and store on your behalf for 180 days in the Microsoft Purview portal. No additional action is required for that change. This is an increase from 90 days for previous Audit Standard customers.

3. **Will Get/Set-Mailbox commands also need to be executed for all new mailboxes going forward?**

   G3/G5 and E3/E5 all logging is enabled by default (except for **SearchQueryInitiated**). However, there are still select license types that require verification or enablement steps (e.g., M365 Business Basic). These types are on the Microsoft 365 roadmap (https://www.microsoft.com/en-us/microsoft-365/roadmap) to be enabled by default.

4. **Any there native SIEM integrations other than Splunk and MS Sentinel?**

   Yes, other integrations are possible, but these are not covered in this playbook.

5. **Are these all post-incident investigation logs?**

   No, these logs can be used for proactive threat hunt. However, doing so requires some effort to parse log data.

6. **Could CISA offer a training class on these topics for us?**

   Training is planned for release in early 2025.

7. **Audit events show up in Splunk with missing data but show the data when viewed directly in the M365 tenant. Why is this?**

   Contact your Splunk representative as we believe this a Splunk issue and not a Microsoft API issue. If you feel this is a Microsoft API issue, contact your Microsoft Customer Success Account Manager (CSAM).

8. **We're only seeing some of those operations in our logs; does that mean those logs were not enabled by whoever turns that on in a console?**

   Ensure enabling steps described in the playbook are being followed correctly. There are nine Teams logs that are only triggered through Graph API.

9. **We are not seeing XDR portal in Government Cloud Community (GCC). How do we fix this?**

   The Sentinel connection to XDR is not currently available in GCC. They are on the Microsoft 365 roadmap in private preview.

10. **Is this expanded logging size and Splunk commercial-off-the-shelf (COTS) solution being offered at no added cost to agencies? Or is the cost to agencies absorbed by CISA or OMB?**

    Neither CISA nor OMB is absorbing the increased costs on behalf of agencies. You need to address increased storage costs with your vendor.

11. **In order to estimate log sizes, where would you find statistics on logs?**

    See the Sentinel Ingestion portal for estimated log size increases, up to a tenfold increase.

12. **We are not seeing SQL logs for ingestion into Splunk. Why are the logs not flowing for some customers?**

    We recommend contacting your Splunk representative or Microsoft CSAM for support.