Hiding in Plain Sight: Using the Office 365 Activities API to Investigate Business Email Compromises

By CrowdStrike Services Group

Archived: 2026-04-05 21:58:46 UTC

Update: While this blog post originally covered the Office 365 Activities API, that functionality has been disabled by Microsoft as of Friday, June 6, 2018. However, there are still data sources available within O365 to help investigate business email compromises (BECs). Please stay tuned for an updated blog post to address alternate methods for detecting and responding to BECs.

Introduction

Business email compromises (BECs) are a big problem across a multitude of industries. Just last week, the FBI participated in an international BEC takedown, arresting 74 individuals across the United States, Nigeria, Canada, Mauritius and Poland. In 2016, the Internet Crime Complaint Center (IC3) named BEC "the $3.1 Billion Scam", with some predicting losses to exceed $9 Billion in 2018. In May 2018, the Ghanaian Police Intelligence Unit arrested 42

Nigerians and two Ghanaians for committing "cyber fraud and robbery" related to BEC activity. Of course, the vast majority of BEC incidents do not end in arrests and convictions. In fact, the barriers to prosecution account in large part for the rampant growth of BEC crimes. In the course of the CrowdStrike® Services team's investigative work responding to BEC cases, we recently discovered a capability within Office 365 that allows for the retrieval of Outlook mailbox activity logs that far exceeds the granularity provided by existing, documented Office 365 log sources, such as the Unified Audit Log. This capability represents access to an always-on, mailbox activity recording system that is active by default for all users. This blog details CrowdStrike's knowledge of and experience with this remarkable Office 365 logging capability. This capability consists of a web API that uses Exchange Web Services (EWS) to retrieve Office 365 Outlook mailbox activities. The API can be accessed by anyone with knowledge of the API endpoint and a specific HTTP header. Activities are recorded for all users and are retained for up to six months. There are many activity types, including logins, messages deliveries, message reads, and mailbox searches. It is possible to acquire mailbox activities for specific time ranges and activity types. There are also some drawbacks to the API, such as the apparent inability to directly link activities to client sessions. Despite this, the API still provides enough detail to allow for the rapid identification of attacker activity, under most circumstances. (NOTE: In conjunction with this article, CrowdStrike is releasing a Python module that wraps the basic functionality of the Activities API. See the Python Module section below for full details.) Threat actors, such as Nigerian confraternities, often adhere to standardized playbooks when engaging in business email intrusions. The initial avenues of entry often consist of phishing emails that contain links to web-based credential stealers appearing to victims as legitimate Office 365 login forms, but occasionally leveraging keyloggers to capture credentials. These threat actors use the stolen credentials to log into the victims' mailboxes and begin collecting intelligence by issuing search queries and reading emails. The threat actors typically identify key

executives and employees involved in financial transactions such as wire transfers, and monitor their activity for periods ranging from weeks to months. Once the threat actors feel comfortable, they insert themselves into email conversations to conduct fraud. They often start by initiating wire transfer requests starting with smaller dollar amounts and working their way up. CrowdStrike has observed unauthorized wire transfer attempts ranging from thousands of dollars to as high as nearly fifteen million dollars. We've also seen these threat actors use the same access to conduct other forms of BEC fraud, such as redirecting payroll, emptying 401k accounts, and clearing out medical savings accounts.

As you will see, the techniques, tactics, and procedures (TTPs) employed by these threat actors can be discovered by analyzing the data returned from this powerful Office 365 API.

Accessing the API

Within the Outlook REST API, there is an undocumented API subset known as Activities. This API subset is included in all three versions (v1.0, v2.0, and beta) of the Outlook REST API. This article focuses primarily on the v2.0 implementation. As with other subsets, calls to the Activities API must be authenticated using a supported method — OAuth 2.0 or Basic Authentication. (Note that Microsoft will no longer support Basic Authentication in the Outlook REST API beginning on November 1, 2018.) If you intend to authenticate with OAuth 2.0, you can access the Activities API using any version of the Outlook REST API. All of the examples in this article use Outlook REST API v2.0 at the following endpoint: `https://outlook.office.com/api/v2.0/{user_context}/Activities` If you intend to authenticate with Basic Authentication, you can access the API using Outlook REST API v1.0 at the following endpoint: `https://outlook.office365.com/api/v1.0/{user_context}/Activities` All requests to the Activities API are issued via the HTTP GET method and must include the following HTTP header. Requests sent without this header will result in an `HTTP 400 Bad Request` response.

Prefer: exchange.behavior="ActivityAccess"

All requests to the Activities API must include an `Authorization` header. Requests that do not include a valid credential will result in an `HTTP 403 Unauthorized` response. In the case of OAuth 2.0, use the `Bearer` scheme.

Authorization: Bearer <access token>

For testing purposes, you can generate an OAuth 2.0 access token at the Outlook Dev Center – OAuth Sandbox. Authorize the application using a valid Office 365 credential and then proceed to acquire tokens. You may use the acquired Access Token with the examples below. (Keep in mind that this token will expire after 60 minutes.) Note that the minimum required scope for Activities API operations is `https://outlook.office.com/Mail.Read`. If you will be using Basic Authentication, supply a base64-encoded username and password in accordance with the `Basic` scheme.

Authorization: Basic <encoded username:password>

Note that with Basic Authentication, if the calling user has multifactor authentication enabled, it will be necessary to generate and supply an app password in place of the account password. It is advisable to include an `Accept`

header with a MIME type of `application/json` and the `odata.metadata` format parameter. All of the examples discussed in this article use the following header:

Accept: application/json; odata.metadata=none

Finally, it should be noted that the Activities API is subject to the same throttling limits imposed on all other subsets within the Outlook REST API. At the time of this writing, that limitation consists of a 10,000 request allowance per 10-minute window, per target user, per application. Requests issued in excess of this limitation will result in an `HTTP 429 Too Many Requests` response. (NOTE: The headers of these responses can be parsed to retrieve additional information about the throttling state.)

Retrieving Activities

The simplest call to the Activities API includes no parameters and uses the "me" user context shortcut. The following request will return a response containing the latest 10 mailbox activities of the calling user.

GET https://outlook.office.com/api/v2.0/me/Activities

This request can be issued using curl as seen here for v2.0 of the API with OAuth:

curl -H 'Prefer: exchange.behavior="ActivityAccess" ' -H 'Authorization: Bearer <access token>' -H 'Accept: application/json; odata.metadata=none' https://outlook.office.com/api/v2.0/me/Activities

And in v1.0 with Basic Authentication as seen here:

curl -H 'Prefer: exchange.behavior="ActivityAccess"' -H 'Accept: application/json; odata.metadata=none' -u <username> https://outlook.office365.com/api/v1.0/me/Activities

It's also possible to issue this request by using the "Edit Query" feature in the Outlook Dev Center - OAuth Sandbox. Make sure to supply the appropriate headers if you use this method. Below is a sample JSON response from the API (not all properties are displayed here):

{ "value": < { "Id":"WOGVSAiPKrfJ4apAPcBksT2en7whzDz4NIbUs3==", "ActivityCreationTime":"2010-04-01T12:34:56.789Z", "ActivityIdType":"ReadingPaneDisplayStart", "AppIdType":"Outlook", "ClientVersion":"15.00.0000.000", "ClientSessionId":"679126f3-02de-3513-e336-0eac1294b120", "ActivityItemId":"NjKG5m6OmaCjGKq6WlbjIzvp94czUDg30qGopD==", "TimeStamp":"2010-04-01T12:34:56.789Z", "TenantId":"679126f3-02de-3513-e336-0eac1294b120", } > }

Given the role that this API can play in incident response, organizations will typically want to retrieve activities for mailboxes not owned by the calling user. Fortunately, the Activities API supports that capability by allowing target users to be specified. To retrieve mailbox activities for another user, modify the user context as follows:

GET https://outlook.office.com/api/v2.0/Users('victim@contoso.com')/Activities

The necessary permission to retrieve mailbox activities for other users can be obtained in one of two ways: the Shared Mailbox or Application Permission method. **Shared Mailbox Method** With this method, the app performing the query should be configured to request the `Mail.Read.Shared` delegated user permission. (This is

not necessary if Basic Authentication is being used.) In addition, the calling user should be granted `FullAccess` permission to the target mailbox. This permission can be applied in the Office 365 Admin center by assigning the "Read and manage" permission for the desired user or by connecting to Exchange Online and running the following PowerShell cmdlet.

Add-MailboxPermission -Identity victim@contoso.com -User analyst@contoso.com -AccessRights FullAccess -InheritanceType All -AutoMapping:$false

This method will work with access tokens obtained by the OAuth Sandbox because the sandbox app includes `Mail.ReadWrite.Shared` in its list of requested scopes. This scope is a superset of the minimum scope required to read shared mailboxes. NOTE: Permission assignments are not reflected in real time. It can take minutes, or sometimes hours, for permission changes to take effect. When valid credentials are supplied, but without sufficient permission to the target mailbox, the API may return an `HTTP 404 Not Found` response with the error, *"The specified object was not found in the store. Default folder ActivityLogs not found."* If this error is returned after correctly granting `FullAccess` permission, you may need to allow additional time for the permission changes to propagate. **Application Permission Method** With this method, the app performing the query should be configured to request the `Mail.Read` *application permission*. If a Global Admin were to provide consent for this request, then activities for all users in the tenant could be retrieved without having to modify mailbox permissions. Note that the OAuth Sandbox cannot be used in conjunction with this method. Employing this method involves developing a web app, implementing a supported OAuth flow, and requesting admin consent from a tenant administrator.

Request Parameters

The following request parameters are supported by the Activities API:

Parameter	Description
`$orderby`	Orders the results by the specified expression
`$filter`	Filters the results by timestamp and/or activity type
`$select`	Selects a list of properties to return
`top`	Specifies the maximum number of activities to return
`$skip`	Specifies the number of activities to skip in the returned results

Table 1: Request Parameters

The order in which the API returns results is not always guaranteed. Results are normally sorted by `TimeStamp` in descending order, however certain `$filter` expressions can alter that behavior. To guarantee the order in which results are returned, use the `$orderby` parameter to specify a property name and sort order. For example, to return results in `TimeStamp` ascending order (oldest results first), issue the following query:

GET https://outlook.office.com/api/v2.0/Users('victim@contoso.com')/Activities?$orderby=TimeStamp+asc

(NOTE: There are restrictions on how the `$orderby` and `$filter` parameters can be used together.) The `$filter` parameter allows for an expression to be supplied that restricts results to a range of timestamps and/or a list of activity types. For example, to request activities that occurred in January of 2018, issue the following query:

GET https://outlook.office.com/api/v2.0/Users('victim@contoso.com')/Activities?$filter=(TimeStamp ge 2018-01-01T00:00:00Z and TimeStamp le 2018-01-31T23:59:59Z)

Timestamps are specified in ISO 8601 format `(2018-01-01T00:00:00Z)`. By default, the API will return all properties (in the default property set) for each activity type. The $select parameter can be used to specify a subset of properties to return instead. For example, to only return the `TimeStamp` and `ActivityIdType` properties for each activity, issue the following query:

GET https://outlook.office.com/api/v2.0/Users('victim@contoso.com')/Activities?$select=TimeStamp,ActivityIdType

If a `$top` value is not specified, the API will return a maximum of 10 activities. To change the default behavior, specify a `$top` value in the range of 1 to 1000. (The Activities API will not return more than 1000 results per request.) For example, to retrieve the last 500 activities that occurred in a mailbox, issue the following query:

GET https://outlook.office.com/api/v2.0/Users('victim@contoso.com')/Activities?$top=500

The API also supports pagination. In cases where a particular query generates a result count that exceeds the length specified by the `$top` value, multiple requests will need to be issued in order to retrieve the entire set. When constructing successive requests, it is necessary to skip the number of activities already retrieved in previous requests. This is accomplished by setting the value of the `$skip` parameter, as demonstrated in the second and third requests below. **First request**

GET https://outlook.office.com/api/v2.0/Users('victim@contoso.com')/Activities?$top=1000

Second request

GET https://outlook.office.com/api/v2.0/Users('victim@contoso.com')/Activities?$top=1000&$skip=1000

Third request

GET https://outlook.office.com/api/v2.0/Users('victim@contoso.com')/Activities?$top=1000&$skip=2000

Standard Properties

There are a number of standard properties that are present in all activities. The most commonly used ones are documented here.

Property Name	Description
`ActivityCreationTime`	The time that the activity was created
`ActivityIdType`	The type of activity (described in detail below)

`ActivityItemId`	The EWS ID of the activity object (can be used to query Mail API for additional message properties, including message body)
`AppIdType`	The client type (described in detail below)
`ClientSessionId`	A client-generated session identifier (has limited use in identifying actual user sessions)
`TimeStamp`	The time that the activity occurred

Table 2: Required Properties

Although other object identifiers (such as `Id` and `ActivityObjectId`) are included in most activities, the `ActivityItemId` property is the expected identifier when querying other subsets, such as the Outlook Mail API. For this reason, `ActivityItemId` is used as the preferred message identifier in this article. The `ClientSessionId` property has limited use in identifying actual user sessions. Although each login activity has an associated `ClientSessionId`, not all subsequent mailbox activity will be associated with that same identifier. There are normally many different client session identifiers per actual user session. This property is most likely a client-generated UUID that is designed to be used for request-response correlation. It may be of interest to note that as new requests are issued within each session, the `SequenceNumber` property is incremented.

Custom Properties

Most activity types include custom properties. Each custom property is represented as a single JSON object within a `CustomProperties` array.

These objects consist of properties that are unique to each activity type, such as the `ClientIP` property for `ServerLogin` activities and the `TargetFolder` property for Move activities. Most of the interesting metadata for each activity is contained within this array. Some of the more useful custom properties are discussed in detail in the Activity Types section below.

Activity Types

Activities are classified by the `ActivityIdType` property. CrowdStrike has identified at least 30 different activity types, ranging from message operations to login events to calendar updates. An exhaustive list and description of all of these activity types is outside the scope of this article, but some of the more useful ones are documented below.

Activity Type	**Description**
`Delete`	A message was deleted (by a user or by Exchange)
`Forward`	A message was forwarded
`LinkClicked`	A link in a message was clicked (does not apply to all application types)

`MarkAsRead`	A message was marked as read
`MarkAsUnread`	A message was marked as unread
`MessageDelivered`	A message was delivered to the mailbox
`MessageSent`	A message was sent from the mailbox
`Move`	A message was moved (by a user or by Exchange)
`OpenedAnAttachment`	An attachment was opened (does not apply to all application types)
`ReadingPaneDisplayEnd`	A message was deselected in the reading pane
`ReadingPaneDisplayStart`	A message was selected in the reading pane (a message was viewed)
`Reply`	A message was replied to (also ReplyAll)
`SearchResult`	Search results were generated
`ServerLogon`	A logon event occurred (may also be accompanied by a Logon activity)

Table 3: Activity Types

It is important to note that many activity types do not contain a substantial amount of message metadata. They do, however, include the `ActivityItemId` field, which can be used in multiple ways to retrieve all of the message metadata for an activity. One way to retrieve the relevant metadata is to query the Mail API for it. This will also allow you to retrieve the body of the message, if necessary. The following query uses Mail API to retrieve the `WebLink`, `BodyPreview`, and `Subject` properties for the message identified by `ActivityItemId`. (NOTE: If you use the beta endpoint, it's also possible to request the `InternetMessageHeaders` property, which can be useful in incident response.)

GET https://outlook.office.com/api/v2.0/Users('victim@contoso.com')/messages/< ActivityItemId>?$select=WebLink,BodyPreview,Subject

Another way to retrieve the metadata for the object is to search for it in corresponding `MessageDelivered` activities. `MessageDelivered` activities (and to a lesser extent, `MessageSent` activities) preserve important properties such as sender, recipient and subject. Activities can be linked together using the `ActivityItemId` property. In this way, it is possible to retrieve message metadata for many activities without issuing additional HTTP requests. For example, `MarkAsRead` activities do not include properties for the recipients or the subject of the message that was marked as read. However, you may already have the associated `MessageDelivered` activity for that message (stored in memory or on disk). Search for it by the `ActivityItemId` property of the `MarkAsRead` activity. If it exists, values for important fields (such as `Recipients` and `Subject`) can be reliably imported. Certain activity types are only generated when they are initiated by a supported application type. As an example, `LinkClicked` and `OpenedAnAttachment` activities are only recorded when those actions are performed using Outlook on the web (the Web application type).

Interesting Activity Types

These activities can provide significant value to analysts hunting for attacker activity in Outlook mailboxes. Accordingly, they are discussed in greater detail below. `MessageDelivered` — This activity indicates that a message was delivered to the mailbox. This activity type is important because it can be used to identify phishing emails and because it contains message metadata that can be used to enrich related activities. The `AppIdType` for this activity is always `Exchange`. Some of the most useful custom properties for this activity type are as follows:

- `SenderSmtpAddress` — The SMTP address of the sender of the message
- `From` — The originator of the message
- `Recipients` — The recipients of the message (limited to 128 entries)
- `Subject` — The subject of the message
- `SentTime` — The time the message was sent
- `InternetMessageId` — The Internet Message ID of the message
- `ItemClass` — The class of the message
- `AttachmentDetail` — A list of the types of attachments in the message

`ReadingPaneDisplayStart` — This indicates that a message was selected in the reading pane, making it viewable. This activity type may or may not contain a CustomProperties array, depending on which application type initiated the action. In all cases, however, there is a limited amount of metadata included in the activity, so the message being referenced must be looked up using one of the methods described in the Activity Types section above. A `ReadingPaneDisplayEnd` activity occurs when a message is deselected. `SearchResult` – These activities contain search queries that were performed within a mailbox. Each of these activities contains a `CustomProperties` array which contains a `Query` XML object. To retrieve the search query, parse this XML object for `/Query/RawQuery`. See the following example `Query` value:

<Query><RawQuery>payment</RawQuery><RankingQuery>any </RankingQuery><All>
<Token>payment</Token></All></Query>

`ServerLogon` – This activity type indicates that an Office 365 Outlook mailbox login event occurred. It contains many useful properties within the CustomProperties array.

Some of these are as follows:

- `ClientIP` — This is the source IP address for the login activity. If this field contains multiple comma-separated values, it indicates that an X-Forwarded-For header was included with the login submission.
- `UserName` — This is the username for the login activity.
- `Result` — This is the result of the login attempt (whether it was successful or not).
- `ExceptionInfo` — In the event of a login failure, exception info may be populated.
- `UserAgent` — This is the client's reported user agent.

Application Types

Each activity has an associated application type which is stored in the AppIdType property. This property indicates the type of application that initiated the activity. The most common application types are documented below.

Application Type	Description
`Exchange`	Exchange Online
`IMAP4`	IMAP4 client
`Lync`	Lync / Skype for Business
`MacMail`	MacOS Mail
`MacOutlook`	MacOS Outlook
`Mobile`	Mobile browser
`Outlook`	Windows Outlook
`POP3`	POP3 client
`Web`	Outlook on the web

Table 4: Application Types

It's important to note that some application types automatically begin synchronizing mailbox contents on first login. This should be taken into consideration when performing analysis because it means that the contents of a mailbox could have been exfiltrated, even without any "message read" activities being returned by the API. CrowdStrike has observed this most frequently occurring with IMAP4. At this time, there is no protocol logging available in Office 365 for IMAP4 or other comparable application types to determine exactly which messages were synchronized by a client.

Python Module

In conjunction with this article, CrowdStrike is releasing a Python module that wraps the basic functionality of the Activities API. It is accompanied by a command-line tool that can be used to retrieve activities for a user and write them to a CSV file. The tool requires a valid OAuth 2.0 access token, which for testing purposes can be obtained from the Outlook Dev Center – OAuth Sandbox. (To obtain access tokens for use with production software, you should implement a supported OAuth flow and register your application with Microsoft. For more details, see v2.0 Protocols - OAuth 2.0 & OpenID Connect.) **Using the command-line tool** This tool will continuously fetch and write activities to a CSV file until all activities matching the specified criteria have been retrieved. The access token can be supplied by setting an OAUTH_TOKEN environment variable (preferred method) or by including it as a command-line argument. Usage is as follows:

usage: retriever.py --user <username> --output <filename> <--token <token>> <--start <timestamp>> <--end <timestamp>> <--types <type> <<type> ...>> --user <username>

Target user (user principal name) --output <filename>

CSV output filename --token <token>

OAuth access token --start <timestamp>

Start timestamp (ISO 8601) --end <timestamp>

End timestamp (ISO 8601) --types <type> <<type> ...>

Space-delimited list of activity types

Example 1: Retrieve MessageDelivered activities that occurred after January 1:

python retriever.py --user victim@contoso.com --output activities.csv --types MessageDelivered --start 2018-01-01T00:00:00Z

Example 2: Retrieve ServerLogon and SearchResult activities that occurred in the month of May:

python retriever.py --user victim@contoso.com --output activities.csv --types ServerLogon SearchResult --start 2018-05-01T00:00:00Z --end 2018-05-31T23:59:59Z

Example 3: Retrieve the entire history of activities for a user. (NOTE: This may take a long time.)

python retriever.py --user victim@contoso.com --output activities.csv

NOTE: Ensure you've acquired the necessary permission to retrieve activities for other users by employing one of the methods described in the Retrieving Activities section above.

Techniques for Finding Attacker Activity

BEC threat actors rarely alter their TTPs and frequently reuse the same infrastructure across phishing campaigns. These characteristics, combined with the right analytical techniques, can allow incident responders to swiftly identify such intruders in their environments. Some of the techniques used by the CrowdStrike Services team are discussed below. **IP Geolocation** One of the simplest and most effective methods for distilling attacker activity from legitimate mailbox owner operations is performing geographic logon analysis. Logons from geographies that are not consistent with the mailbox owner's residence or travel patterns should be considered as immediate candidates for further inspection. Numerous free sources of IP geolocation data exist, all of which can be used to programmatically translate source IP addresses of logon activities to specific geographic locations.

Figure 1: IP Geolocation

Search Queries Search query analysis is particularly useful in BEC cases involving wire fraud (such as those commonly perpetrated by Nigerian confraternities) because mailbox searching is one of the most frequently employed techniques of these types of attackers. Upon gaining access to an Office 365 Outlook mailbox, these threat actors typically begin issuing queries for terms such as "wire transfer," "invoice" and "payment" in order to gather the necessary information to target specific people and processes. Analyzing SearchResult activities for suspicious search terms is one of the most efficient methods for isolating attacker activity. **Anomalous Client Types** Anomalous application types are frequent indicators of suspicious activity. Attacker activity can often be discovered by stacking application types to identify the instances that appear least over an extended period of time. For example, if a given user usually logs into his or her Office 365 Outlook mailbox via the Outlook client, activities performed via Web or IMAP4 constitute potential indicators of attacker activity. Identifying these types of outliers in mailbox activity history is essential to conducting successful BEC investigations.
DKIM/DMARC/SPF Failures The Office 365 Outlook Mail API can be called upon to retrieve message headers and bodies. By enriching activities with these additional properties, it is possible to search for other indications of malicious activity, most notably those involving DKIM, DMARC, and SPF failures. Messages that fail one or more of these checks may be associated with malicious activity, such as spoofed sender addresses or the use of unauthorized Mail Transfer Agents (MTAs). **Malicious Attachment Types** The properties of received messages are stored in MessageDelivered activities. Many of these properties can be used to identify suspicious emails. Among these is the AttachmentDetail property, which contains a list of attachment types and methods. By scanning this field for malware-related attachment types (such as HTA, EXE, and PDF), suspicious emails can be rapidly discovered and then analyzed for additional indicators.

Conclusion

The Office 365 Outlook Activities API provides a straightforward interface to a powerful mailbox logging subsystem that is vastly superior to existing Office 365 auditing capabilities. While the full potential of the API is unknown at this time, CrowdStrike has documented many of the features that can be used to derive maximum operational value from the API. Organizations and individuals alike can utilize this knowledge to respond to incidents, hunt for attackers, or simply to gain a better understanding of Office 365 Outlook mailbox operations. In CrowdStrike investigations of BEC cases, it is not unusual for the threat actor to have gained access or compromised the customer's environment outside of the Office 365 footprint. In such cases, CrowdStrike recommends a comprehensive Compromise Assessment / Threat Hunting engagement to determine the extent of the compromise and to ensure the threat actor can be adequately eradicated from the client's infrastructure.

For more information on CrowdStrike's Incident Response, Compromise Assessment or Threat Hunting offerings, visit the CrowdStrike Services page or please reach out to us at: Services@crowdstrike.com

---