Search for and delete chat messages in Teams

By Kas Tsaedu

This guide primarily outlines the process for searching for and deleting chat messages in Teams using eDiscovery (Premium) and the Microsoft Graph Explorer. I developed this draft due to the numerous discrepancies and inconsistencies found in the official Microsoft documentation titled "Search for and delete chat messages in Teams" While attempting to follow this document in my job, I discovered significant errors and a lack of clarity in the official Microsoft KBA. This motivated me to write a clearer and more user-friendly set of instructions. Administrators can use this guide to find and remove sensitive or inappropriate content or to respond to data spillage incidents where confidential or malicious information has been released through Teams chat messages in Microsoft 365.

Minimum Admin role required to complete the task:

— eDiscovery Manager: To create an eDiscovery (Premium) case and use collections to search for chat messages
— Search and purge: To run the purge (To delete chat messages)
— Graph: API Permission to eDiscovery.Read.All and eDiscovery.ReadWrite.All permissions: To complete the purge process in Graph Explorer or PowerShell

Steps to Search and delete

Step 1: Create a case in eDiscovery (Premium)

— Go to https://compliance.microsoft.com and sign in.
— select eDiscovery > **Premium** > select the **Cases** tab, and then select **Create a case**.
— In the Name and description page provide Name and description for your case
— Select the New (recommended) option and then select **Next**

Step 2: Create a collection estimate

— Navigate to the **Cases** tab in eDiscovery (Premium) > select the case created for this purpose
— select the **Data Sources** tab > click Add Data Source > specify the data source: Add new custodians > search for the custodians' UPN /user or group/ or import if you have csv or another file list.

— select the **Collections** tab > select new collection > provide Name and description. Note after the collection is created, you can't change the name, but you can modify the description.

— On the Custodial data sources page > Select the **Select all** toggle to search all custodians that were added to the case. When you select this option, all data sources for all custodians are searched. If you need to search to only specific custodians that were added to the case, click on the Select **custodians** list of the case custodians is displayed and select one or more custodians.

— (This is not required if you select all custodians in option C above) On the **Non-custodial** data sources page, select one of the following options to identify the non-custodial data sources to collect content from:

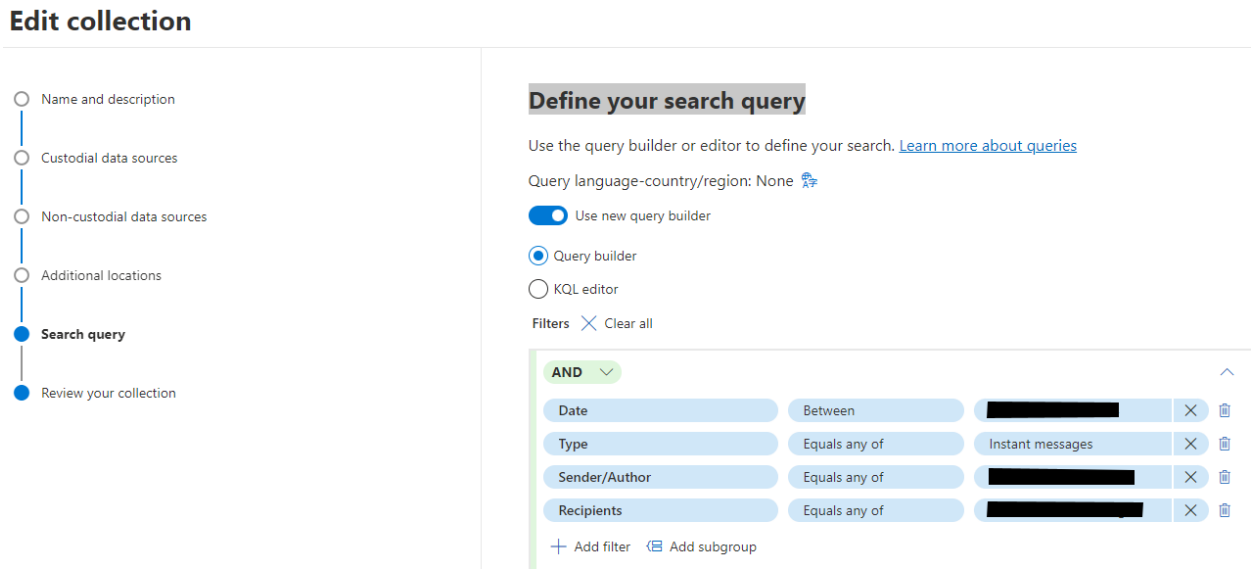
— Select **"Select non-custodial"** data sources to select specific non-custodial data sources that were added to the case.

— Select the **Select all** toggle to select **all** non-custodial data sources that were added to the case.

— Click NEXT

— On the **Additional locations** page, you can select other mailboxes and sites to search as part of the collection.

— On the Search Query page: Define your search query as required:



Note: Administrators can add multiple collection as required.

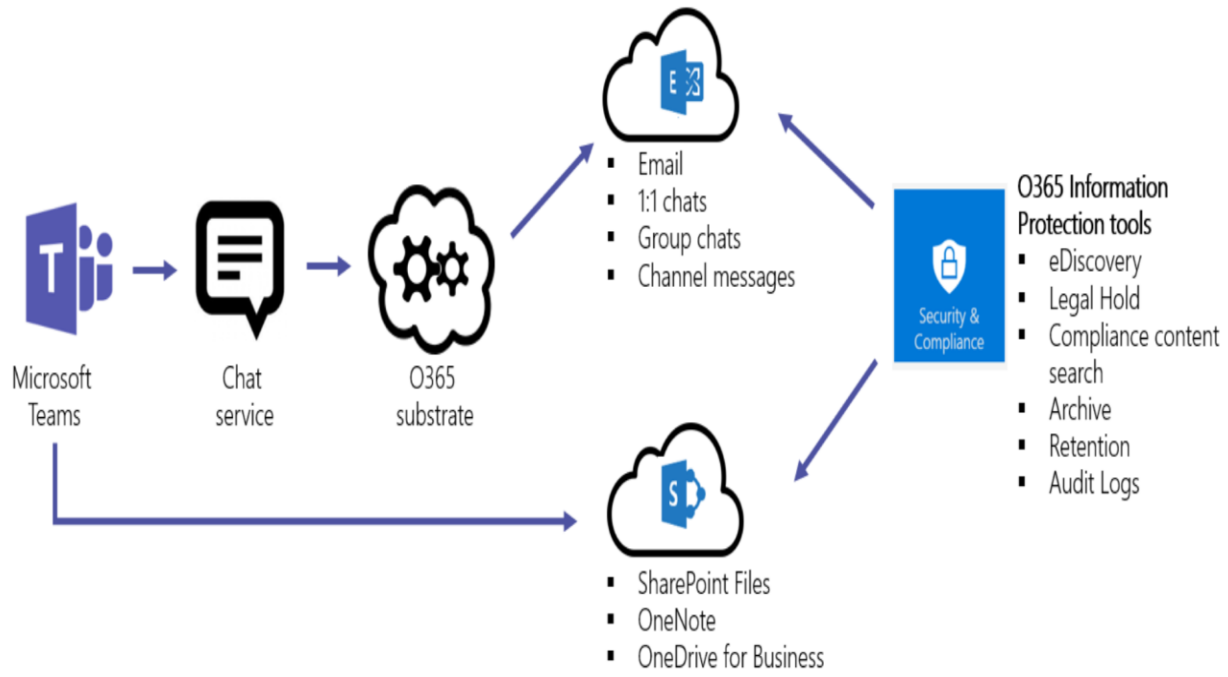
For searching for chat messages: use the **Type** condition and select the **Instant messages** option when you build the search query for the collection estimate.

Adding a date range or several keywords to narrow the scope of the collection to items relevant to your search a delete investigation is recommended.

Step 3: Review and verify chat messages to delete:

- Review a sample of items in a collection estimate and confirm if there is data available in the collection and if the data is the target information looking to purge.
- Additionally, you can use the collection statistics (specifically the Top Locations statistics) to generate a list of the data sources that contain items returned by the collection. Use this list in the next step to remove hold and retention policies from the data sources that contain search results. For more information, see Collection statistics and reports.

The following figure indicates the ingestion flow of Teams data to both Exchange and SharePoint for Teams Files and Messages. Target the correct data location path accordingly.



Step 4: Remove all holds and retention policies from data sources

Before we delete the chat messages from a mailbox, administrators might need to review and get their organization approval to remove all organization-wide holds, site holds, or retention policy for the targeted mailboxes. If not, the chat we're trying to delete is retained. When retention policies and holds applied are removed, there is a risk of further data loss during that time frame. It can be because of the user have the capability of deleting items at the time frame.

Once the main data source location are identified, identifying the organizational wide hold, retention assigned to the specific data source (for 1:1 chat for example the data source location path shows like this: Primary*, <GUID>\<username>(Primary)**__*TeamsMessagesData*__) this shows we need to remove the user from the teams retention policy assigned for teams chat. For Group chat, chats like 1 to many or vice versa, chats in channel, we might need to remove the users from the other retention policies like retention for channel etc.

=> Be sure to write down these settings or save them to a text file because you'll change some of these properties and then revert back to the original values in Step 6, after you delete items from the Recoverable Items folder.

Here is list of the mailbox properties you might need to collect as required for the identified data location/path.

```
Get-Mailbox <username> | FL SingleItemRecoveryEnabled,RetainDeletedItemsFor
Get-Mailbox <username> | FL *hold*
Get-mailbox <username> | fl userp*, guid, inp*, ret*, lit*, sing*, delay*, elc*, *quo*
```

=> IF LitigationHoldEnabled, InPlaceHolds, DelayHoldApplied, DelayReleaseHoldApplied, RetentionHoldEnabled are both setting to False. NO action required!

- Run the set- command as required if any of the hold are enable to True and if it blocks the purging process.

Example: If any of the hold are enabled, you'll have to disable it as required. If the deleted item retention period isn't set for 30 days (Max value in EXO), then you can increase it.

You can run the following to disable the hold assigned to the affected user:

```
Set-Mailbox <username> -SingleItemRecoveryEnabled $false #To enable single item recovery:
Set-Mailbox <username> -RetainDeletedItemsFor 30       #To increase the deleted item retention period
(This assumes that the current setting is less than 30 days) #
Set-Mailbox <username> -RemoveDelayHoldApplied         #To remove the delay hold #
Set-Mailbox <username> -RemoveDelayReleaseHoldApplied       #To remove the delay release hold #
```

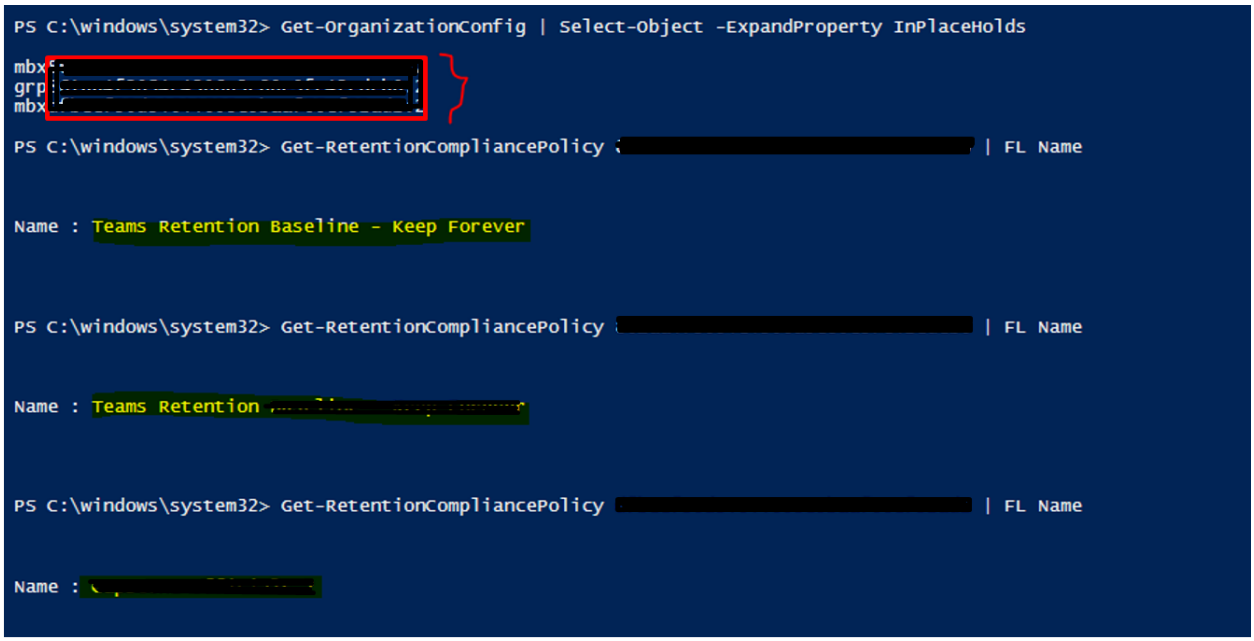
Run the following command to get information about any organization-wide retention policies

```
Get-OrganizationConfig | Select-Object -ExpandProperty InPlaceHolds

Get-organizationconfig | fl OrganizationId, name, guid, InPlaceHolds, ElcProcessingDisabled
```

Run the following command in Exchange Online PowerShell to identify the In-Place Hold that's placed on the mailbox. Use the GUID for the In-Place Hold that you identified

```
Get-RetentionCompliancePolicy <retention policy GUID without prefix> | FL Name
```



After you identify the retention policy, go to the **Data lifecycle management** > **Microsoft 365** > **Retention page** in the compliance portal, edit the retention policy that you identified in the previous step, and remove the mailbox from the list of recipients that are included in the retention policy.

If a mailbox is excluded from an organization-wide Microsoft Purview retention policy, the GUID for the retention policy that the mailbox is excluded from is displayed in the InPlaceHolds property and is identified by the -mbx prefix.

Get-Mailbox <username> | FL InPlaceHolds Check the InPlaceHolds property andm make sure the prefix shows empty or (-MBX)

Step 5: Delete chat messages from Teams

We can use one of the following two options to perform the purge request.

Option One: Microsoft Graph Explorer

To complete the purge process in Graph Explorer, you may have to consent to the correct graph explorer permissions.

We will use PowerShell script and Microsoft Graph Explorer to perform the following three tasks:

- Get the ID of the eDiscovery (Premium) case that we created in Step 1. This is the case that contains the collection created in Step 2.
- Get the ID of the collection that we created in Step 2 and verified the search results in Step 3. The search query in this collection returns the chat messages that will be deleted.
- Delete the chat messages returned by the collection.

Follow the following:

- Get the ID of the eDiscovery (Premium) case (eDiscoveryCaseId also called Id) that we created in Step 1. This is the case that contains the collection created in Step 2.
- Go to https://developer.microsoft.com/graph/graph-explorer and sign in to the Graph Explorer with an account that's assigned the ***Search And Purge*** role in the Microsoft Purview compliance portal.
- Run the following **GET** request to retrieve the ID for the eDiscovery (Premium) case. Use the value https://graph.microsoft.com/v1.0/security/cases/ediscoveryCases in the address bar of the request query. Be sure to select v1.0 in the API version dropdown list. This request returns information about all cases in your organization on the Response preview tab.
- Scroll through the response to locate the eDiscovery (Premium) case. Use the **displayName** property to identify the case.
  - Copy the corresponding ID (or copy and paste it to a text file). You'll use this ID in the next task to get the collection ID.

Tip: Alternatively, to obtain the case Id, you can open the case in the Microsoft Purview compliance portal and copy the case Id from the URL. It shows something like URL…Id=…….

- Get the eDiscovery Search Id (SearchId)

- In Graph Explorer, run the following **GET** request to retrieve the ID for the collection that you created in Step 2, and contains the items you want to delete. Use the value https://graph.microsoft.com/v1.0/security/cases/ediscoveryCases/{ediscoveryCaseID}/searches in the address bar of the request query, where *{ediscoveryCaseID}* is the CaseID that you obtained in the previous procedure.
- Scroll through the response to locate the collection that contains the items that you want to delete. Use the ***displayName*** property to identify the collection that you created in Step 3. In the response, the search query from the collection is displayed in the *contentQuery* property. Items returned by this query is deleted in the next task.
  - Copy the corresponding **ID** (or copy and paste it to a text file). You'll use this ID in the next task to delete the chat messages.

- Delete the chat messages with Graph API
- In Graph Explorer, run the following **POST** request to delete the items returned by the collection that you created in Step 2.
- Use the value https://graph.microsoft.com/v1.0/security/cases/ediscoveryCases/{ediscoveryCaseID}/searches/{ediscoverySearchID}/purgeData in the address bar of the request query,

where *{ediscoveryCaseID}* and *{ediscoverySearchID}* are the IDs that you obtained in the previous procedures.

- If the POST request is successful, an HTTP response code: 200 is displayed in a green banner stating that the request was accepted. For more information on purge Data, see sourceCollection: purgeData.

  Option Two: Delete chat messages with PowerShell

  You can also delete chat messages using PowerShell. For example, to delete messages in the US Government cloud you could use a command similar, but you need to specify the environment to USGov. Commercial tenant doesn't need the environment to be specified.

```
    - Connect-MgGraph -Scopes "ediscovery.ReadWrite.All" -Environment USGov
    ⇨ Use the:  Connect-MgGraph -Scopes "ediscovery.ReadWrite.All"   #without specifying the
      environment if you are not able to connect with the environment specified.
#Run the following to check the search collection"
    - Invoke-MgGraphRequest -Method Get -Uri
      '/v1.0/security/cases/ediscoveryCases/{ediscoveryCaseID}/searches/{ediscoverySearchID}'
# Run the following to delete chat messages
    - Invoke-MgGraphRequest -Method POST -Uri
      '/v1.0/security/cases/ediscoveryCases/<ediscoveryCaseID>/searches/<search ID>/purgeData'
```

A maximum of 10 items per mailbox are deleted when you run the previous command. For multiple items (more than 10, you need to rerun the script in different time interval).

Step 6: Verify chat messages are deleted

After you run the POST request to delete chat messages, these messages are removed from the Teams client and replaced with an automatically generated message stating "This message was deleted by an admin". Admins can confirm with end users if the original chat messages are replaced by the auto generated messages is replaced.

Alternatively, administrators can rerun content search and verify if any matching messages are found.

Deleted chat messages are moved to the Substrate Holds folder, which is a hidden mailbox folder. Deleted chat messages are stored there for at least 1 day, and then are permanently deleted the next time the timer job runs (typically between 1-7 days).

Note: SingleItemRecoveryEnabled can be switched to True by itself (**broken by design**) after we run the purge request, and you might need to recheck after you run the purge request in step 5 if items are not purged successfully.

Step 7: Reapply holds and retention policies to data sources

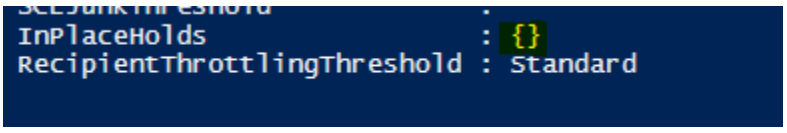
After verifying that chat messages are deleted and removed from the Teams client, we will reapply the holds and retention policies that we removed in the earlier **Step 4.**

Note: end users have the capability to purge items when we are preforming in the windows after we remove the hold and retentions (step 4) and before reapplying the hold (step7). Reapply the holds and retention policies we removed in step 4 as soon as we complete and verified the purge is must and minimizes the risk of losing data

Example: if we disabled the Single Item Recovery in step 4, we need to reapply the hold by running the following script:

Set-Mailbox <username> -SingleItemRecoveryEnabled $True #To enable single item recovery. Follow the same process to all removed holds/retention

Remove users from the team's retention exclusion (if any): After the users are removed from the team's retention exception (users are now subject to the organizational wide retention policy:



Follow the same process and reapply ALL the holds and retention policies that we removed in the earlier Step 4

Reference:

Search for and delete chat messages in Teams | Microsoft Learn

Delete items in the Recoverable Items folder | Microsoft Learn

Learn about retention for Teams | Microsoft Learn