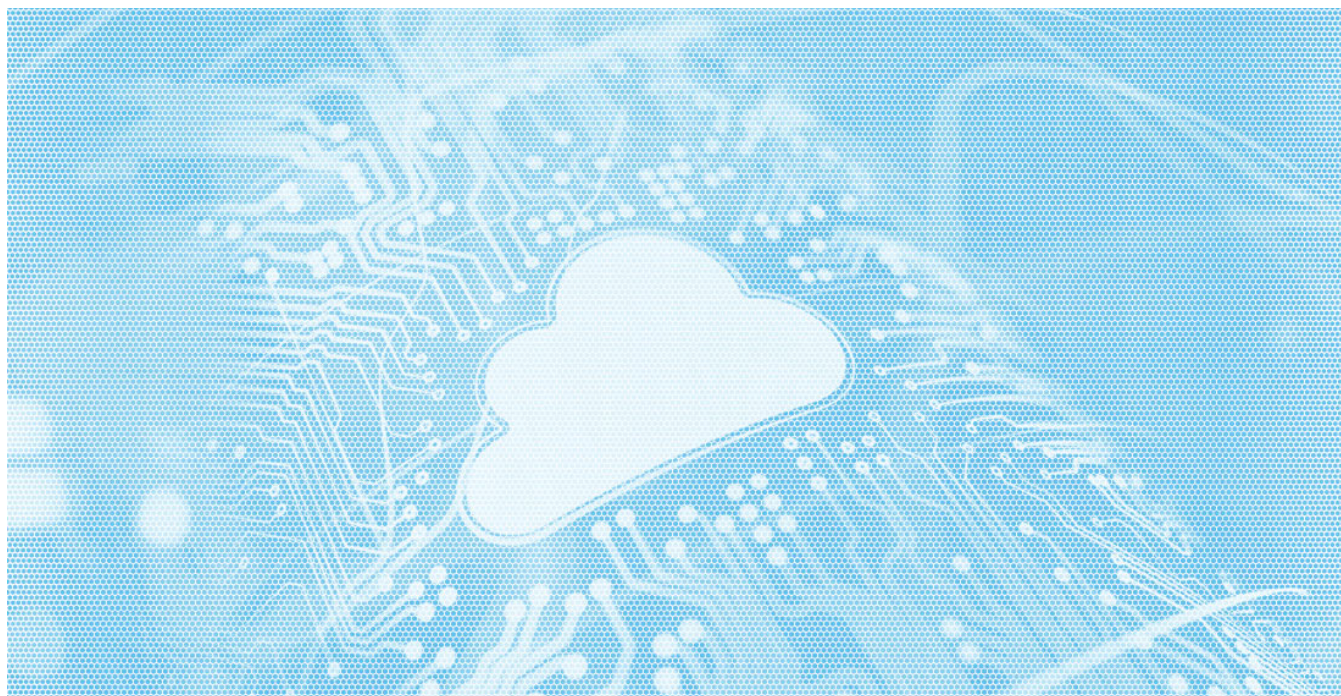


Cloudy with a Chance of Persistent Email Access

aon.com/cyber-solutions/aon_cyber_labs/cloudy-with-a-chance-of-persistent-email-access/



How an Advanced Threat Group Leveraged Microsoft Azure to Gain Persistent Access to Emails

Introduction

Over the last few weeks, [Stroz Friedberg Incident Response](#) has led multiple investigations related to the nation-state threat group being tracked under several names – UNC2452, Dark Halo, StellarParticle and SolarStorm. Stroz Friedberg has tracked some of the activities tied to this threat group going back to January 2020. It has been widely reported that this threat group used the [SUNBURST backdoor](#) to gain initial access to victim networks. However, Stroz Friedberg is investigating incidents where the same threat group gained initial access through alternative means, without any evidence of SUNBURST use.

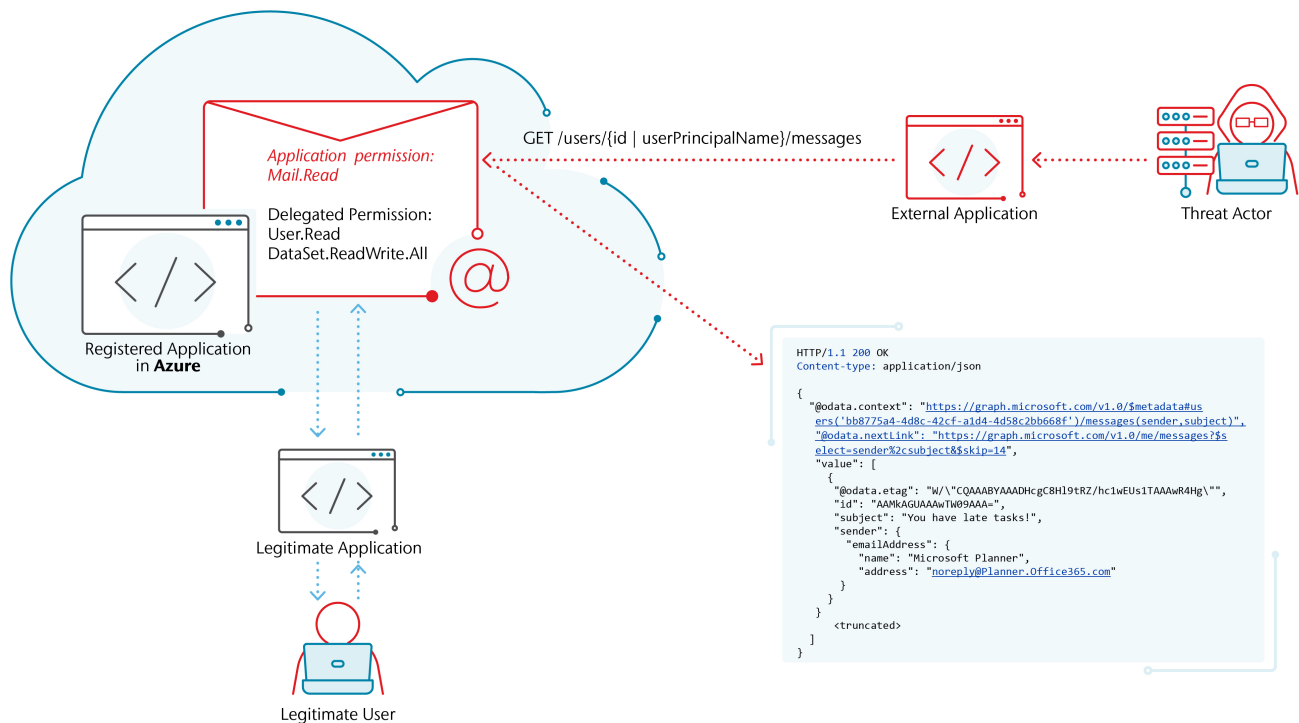
Based on open-source reporting, a common theme associated with this threat group is its focus on victims' email data – be it in [O365 or on-premise Exchange](#). On December 13, 2020, Microsoft issued an [advisory](#) on this threat group detailing the tactics that enabled persistent long-term access to users' email in Office 365 (“O365”) – specifically by adding the application-level permissions “Mail.Read” or “Mail.ReadWrite” to cloud applications in Azure.

In this post, we're going to enumerate (1) the attack chain wherein the threat group gains long-term access to all users' mailboxes within an O365 tenant and (2) the evidence sources that track such activity.

Attack Chain

A crucial component needed to gain long-term persistent access to read all users' mailboxes in O365 is access to an administrator account in Azure Active Directory. This is especially easy for a threat group to accomplish when an administrator account is not protected by multi-factor authentication (“MFA”). Even in cases where MFA is implemented, this threat group has taken [steps to bypass MFA](#). In our investigations, the threat group took the following steps:

1. Compromise administrator account credentials and/or forge authentication tokens for an administrator account and log into the account.
2. Modify existing Azure applications to add “Mail.Read” or “Mail.ReadWrite” application permissions.
3. Create a client secret to enable [“client credentials grant”](#) authentication to modified Azure applications with the newly added permissions.
4. With access to (1) the client ID of the targeted Azure application, (2) the Azure tenant ID and (3) the client secret, connect to victim's Azure tenant via the registered application to read email from any mailbox in the tenant.



Sample threat workflow involving modification of Azure application permissions

It's helpful to visualize this and walk through an example scenario. When an application is registered in Azure, a delegated permission "User.Read" is automatically assigned to the application. In Azure, delegated permissions differ from application permissions as [described by Microsoft](#) below:

Delegated Permissions: *Delegated permissions are used by apps that have a signed-in user present. For these apps, either the user or an administrator consents to the permissions that the app requests. The app is delegated permission to act as the signed-in user when it makes calls to the target resource.*

Application Permissions: *Application permissions are used by apps that run without a signed-in user present, for example, apps that run as background services or daemons. Only an administrator can consent to application permissions.*

In this example, suppose the organization also adds the permission "Dataset.ReadWrite.All" for their purposes. Once a threat group gains access to Azure using an administrator account, they can add an application permission like "Mail.Read" to this existing application in Azure. Adding this permission will not impact the legitimate usage of the application by the organization, and as a result may go undetected. Upon adding the "Mail.Read" permission, the threat group can then create a client secret to authenticate to this application. Armed with the client ID, tenant ID and the client secret, the threat group can now connect to Azure and read all email via an *external* application. That external application could be a commercial product designed to export mail from O365 or could be a custom application to pull mail via the Microsoft Graph API. Alternatively, the threat group could register a new application of their own to Azure as a method of persistent access. This method is much more likely to be detected, depending on the tenant configurations and diligence of the organization.

As an example, issuing an HTTP GET request like `GET /users/{id | userPrincipalName}/messages` would result in retrieving all email messages for a specific user. A list of Graph API methods that can be used is documented by Microsoft [here](#).

Using the Microsoft Graph API, a threat group can also issue requests based on certain filters. For example, they can retrieve all emails within a tenant that contain specific keywords such as "PRIVILEGED AND CONFIDENTIAL" in the body of an email.

As a proof of concept, Stroz Friedberg created an external application called "MailFetch" that takes as inputs client ID, tenant ID and the client secret and issues Microsoft Graph API requests through the registered Azure application associated with the client ID. To manage the output and to prevent reading all email within the tenant, the application takes as an input a "username" which is used as a filter. Using MailFetch, you can verify that the "Mail.Read" permission grants read access to all users' mailboxes. MailFetch is publicly available on our [GitHub page](#).

Organizations impacted by this threat workflow may be interested to know the following data that was accessible through the Graph API permission "Mail.Read" during our testing:

1. Message__body__content
2. Message__body__contentType
3. Message__bodyPreview
4. Message__conversationId
5. Message__conversationIndex
6. Message__flag__flagStatus
7. Message__from__emailAddress__address
8. Message__from__emailAddress__name
9. Message__hasAttachments
10. Message__importance
11. Message__inferenceClassification
12. Message__internetMessageId
13. Message__isDraft
14. Message__isRead
15. Message__isReadReceiptRequested
16. Message__parentFolderId
17. Message__receivedDateTime
18. Message__replyTo__emailAddress__address
19. Message__replyTo__emailAddress__name
20. Message__sender__emailAddress__address
21. Message__sender__emailAddress__name
22. Message__sentDateTime
23. Message__subject
24. Message__toRecipients__emailAddress__address
25. Message__toRecipients__emailAddress__name
26. Message__webLink
27. Message__changeKey
28. Message__createdDateTime
29. Message__lastModifiedDateTime
30. Message__id
31. Message__isDeliveryReceiptRequested
32. Attachments__contentBytes
33. Attachments__contentType
34. Attachments__isInline
35. Attachments__lastModifiedDateTime
36. Attachments__name
37. Attachments__size
38. Attachments__id
39. Attachments__contentId
40. Attachments__contentLocation

As we see from the email properties exposed above, a threat group using an Azure application with the “Mail.Read” application permission has access to the body of the email, email senders/recipients and attachments in addition to other email properties.

As mentioned in the example scenario, the threat group is also able to filter on email properties to retrieve only emails of interest. Per Microsoft’s [documentation](#), filters can be applied to the following properties:

1. attachment
2. bcc
3. body
4. cc
5. from
6. hasAttachment
7. importance
8. kind
9. participants
10. received
11. recipients
12. sent
13. size

14. subject

15. to

Examples on how these HTTP requests can be crafted can be found [here](#).

Detection Mechanisms

On December 16, 2020, Microsoft released [Azure Sentinel queries](#) that detect the Techniques, Tactics and Procedures (“TTPs”) mentioned in this post, including but not limited to: (1) the creation of service principals; and (2) addition of permissions to applications. Leveraging these, Stroz Friedberg recommends organizations review the following areas to detect the attack chain detailed above.

Sign-in Activity

When investigating sign-in activity for this attack chain, focus on sign-ins to Office 365 and Azure Service Principals.

As with any investigation into malicious activity in Office 365, look out for user sign-ins that originate from anomalous IP addresses, especially those from anonymization services. Administrator accounts are of particular interest since these are the accounts that can modify cloud applications in Azure. Gaining access to an administrator account is the first step in this attack chain.

In addition to common methods such as phishing or credential stuffing, one potential avenue for O365 account compromise that this threat group has been known to use is SAML token forgery. If the threat group has stolen the SAML-signing certificate from the organization’s network, or added their own certificate to the tenant, they can sign their own tokens to impersonate any user in the tenant, including administrator accounts. A [January 2021 CISA alert](#) addressing ways to detect malicious activity in Azure notes a specific “UserAuthenticationValue” of 16457, which is described as an indicator of potential SAML forgery. Looking for non-guest users whose sign-in events in the Unified Audit Log contain this value could identify sign-ins that have used forged SAML tokens.

Additionally, Service Principal sign-ins should be reviewed for unauthorized activity. These events can be found in the Azure Active Directory portal under Sign-Ins. Within Azure Sentinel, this data is tracked in a table named “AADServicePrincipalSigninLogs”, which displays the source IP address that the sign-in attempt originated from. As with Office 365 user logins, anomalous IP addresses could be indicative of compromise.

Please note that AADServicePrincipalSigninLogs is only enabled in Azure Sentinel when diagnostic logging is enabled on the workspace. Microsoft has listed the steps to enable diagnostic logging [here](#).

Modifications to Azure Applications

A core component of these types of attacks involves modifying existing applications with additional permissions. There are two ways to detect this, (1) via the Unified Audit Log which shows historical activity (retention will vary depending on the environment), and (2) the active application-level permissions within Azure.

Unified Audit Log

Organizations should review operations in the Unified Audit Log that relate to application creations, changes, and deletions. Any of the following Unified Audit Log events could indicate changes to an Azure application:

1. Update application.
2. Consent to application.
3. Add OAuth2PermissionGrant.
4. Add app role assignment to service principal.
5. Add app role assignment grant to user.
6. Add service principal.
7. Update service principal.
8. Update application.
9. Update application – Certificates and secrets management

Although each operation is formatted slightly differently, you will most often find the Application/Service Principal name in the **AdditionalDetails** portion of the log; specifically, under the **targetname** attribute.

A User Agent attribute can also be found in the **AdditionalDetails** section of the log entry which can also be used as a potential indicator of compromise, the exact attribute name is “User-Agent”.

Audit of active application-level permissions in registered Azure Apps

In addition to a review of the Unified Audit Log, Stroz Friedberg also recommends an audit of *active* application-level permissions assigned to registered Azure applications to ensure there is no active compromise.

Apart from email access via an Azure application with “Mail.Read” or “Mail.ReadWrite” permissions, threat actors could plausibly add other application-level permissions such as “Sites.Read.All” or “Files.Read.All” which would grant read access to documents stored on all SharePoint/OneDrive sites in the organization.

To review these application permissions within Azure, one can perform the following steps:

1. Go to the [Azure Portal](https://portal.azure.com) (https://portal.azure.com)
2. Search for **Enterprise Applications**, and navigate to the page
3. Select the application to review
4. On the left-hand side, select **Permissions**
5. Look for permissions that are potentially overinclusive such as Mail.Read permissions. If the Type is labeled as “Application” this indicates that there is potential for global read abilities.

See below for an example of an application with malicious “Mail.Read” permissions added:

Permissions

« Refresh Review permissions Got feedback?

Permissions

Applications can be granted permissions to your directory by an admin consenting to the application for all users (Admin consent), a user consenting to the application for him or herself (User consent), or an admin integrating an application and enabling self-service access or assigning users directly to the application. As an administrator you can grant consent on behalf of all users in this directory, ensuring that end users will not be required to consent when using the application. Click the button below to grant admin consent.

As an administrator you can grant consent on behalf of all users in this directory, ensuring that end users will not be required to consent when using the application. Click the button below to grant admin consent.

Grant admin consent for

Admin consent User consent

Search permissions

API Name	Permission	Type	Granted through	Granted by
Microsoft Graph				
Microsoft Graph	Existing Permissions	Maintain access to data you have given it access to	Delegated	Admin consent An administrator
Microsoft Graph		Read all users' full profiles	Application	Admin consent An administrator
Microsoft Graph	Malicious Permissions Added	Read mail in all mailboxes	Application	Admin consent An administrator

Azure application with “Mail.Read” permissions

If you have more than a couple of applications in Azure, Stroz Friedberg recommends using [Sparrow](#), a tool created by Cybersecurity & Infrastructure Security Agency (“CISA”), to perform this audit across all applications in your Azure tenant. Upon successful execution, this tool creates a report called “ApplicationGraphPermissions.csv” that contains all the application-level permissions assigned to applications in Azure.

A list of all application-level graph permissions can be found in Microsoft’s documentation [here](#). If a compromised application contains unexpected permissions that could access Mail, Sites or Files, Stroz Friedberg recommends reviewing the Unified Audit Log for associated activities. Below is advice on identifying access to mail by a compromised application.

Emails Accessed through Compromised Application

In some cases, organizations can determine the exact emails that were accessed via the compromised Azure application. To perform this analysis, organizations must already have premium Office 365 E5 subscriptions. Only users that were already assigned the Office 365 E5 license will be eligible for this investigation. Enabling E5 licenses at the start of the O365/Azure investigation *will not* provide historical data.

To start this workflow, organizations will need to get the Application ID of the application(s) deemed compromised.

To find the Application ID:



1. Go to the [Azure Portal](https://portal.azure.com) (https://portal.azure.com)
2. Search for **Enterprise Applications**, and navigate to the page
3. Select the application you have deemed as compromised
4. Select **Properties** on the left-hand side of the screen
5. Note and save the Application ID; this will be needed later.


See below for an example of the Application Properties page and location of Application ID:


Enabled for users to sign-in? Yes No

Name * ✓

Homepage URL

Logo  

Application ID 

Object ID 

User assignment required? Yes No

Visible to users? Yes No

Notes

Sample of

an Application ID from <https://portal.azure.com>

With the Application ID documented, use the ID to determine all emails accessed by the application.

Unified Audit Log / Mailbox Audit Log Analysis

Within the Unified Audit Log and Mailbox Audit Logs, analyze the "MailItemsAccessed" operation. Both the Unified Audit Log and Mailbox Audit Logs contain this operation, and the client ID can be found in the "ClientAppID" field.

See below for a sample of attributes found in one MailItemsAccessed Unified Audit Log entry (values shown below are randomized or generic). Stroz Friedberg identified the user agent "**Client=REST;Client=RESTSystem;;**" utilized across multiple cases, however it is possible the threat actor could utilize a different user agent in other environments.

Highlighted in **red** is the compromised client application ID. Highlighted in **blue** are the Internet Message IDs of emails accessed – multiple messages may be aggregated into one MailItemsAccessed event. These Internet Message IDs can then be correlated to metadata within acquired mailboxes to extract the exact emails accessed.

Attribute Name	Value
CreationTime	2021-01-01T12:00:00
Id	a123bcde-1a2b-3c4c-1234-abcde12f34g5
Operation	MailItemsAccessed
OrganizationId	abcdef-7247-92fj-b184-24789fde193e
RecordType	50,"ResultStatus" Succeeded
UserKey	123456789A12345
UserType	0,"Version" 1,"Workload" Exchange

UserType.Version	1
UserType.Workload	Exchange
UserId	jane.doe@YourDomain.com
ClientAppId	8db2c6d7-1abc-123a-a123-ab12cd34567e
ClientIPAddress	168.63.129.16
ClientInfoString	Client=REST;Client=RESTSystem;;
ExternalAccess	FALSE
ExternalAccess.InternalLogonType	0
ExternalAccess.LogonUserSid	S-1-2-34-1234567891-234567890-123456789-12345678
MailboxGuid	a12345bc-67d8-1234-56ef-1234g567h890
MailboxOwnerSid	S-1-2-34-1234567891-234567890-123456789-12345678
MailboxOwnerUPN	jane.doe@YourDomain.com
OperationProperties.MailAccessType	Bind
OperationProperties.IsThrottled	FALSE
OrganizationName	YourDomain.com
OriginatingServer	AB1CD2345EF3456 (12.34.5678.901)u111bu222c
SessionId	12a34567-1234-123b-a12b-a123bcd45e6f
Folders.FolderItems.InternetMessageId	<12345678a1b2345c1a12345ab1a12ab@EXCH01.YourDomain.com>
Folders.FolderItems.InternetMessageId	<43215678a1b2345c1a12345ab1a12ab@EXCH01.YourDomain.com>
Folders.FolderItems.InternetMessageId	<34565678a1b2345c1a12345ab1a12ab@EXCH01.YourDomain.com>
Folders.FolderItems.InternetMessageId	<22345678a1b2345c1a12345ab1a12ab@EXCH01.YourDomain.com>
Folders.FolderItems.Id	LgAAAAB1aABCDefABCDEFgHij1aABCD+AbBcdEFGHiJ1aB1CDEFGHIJKLM1ABCD
Folders.Path	Inbox
Folders.OperationCount	4
ResultIndex	1
ResultCount	4500
Identity	a123bcde-1a2c-3d4e-1234-a1ab123a45b6
IsValid	TRUE
ObjectState	Unchanged

Sample MailItemsAccessed

MailItemsAccessed logging does have some limitations; for instance, if an account has over 1,000 items accessed in a single day, MailItemsAccessed logging will pause for the next 24 hours. Additionally, users with lower levels of licensing will not have this logging available, limiting visibility into this activity for non-E5 users. Additional information related to MailItemsAccessed analysis can be found in Microsoft's documentation [here](#).

Authors: Partha Alwar, Carly Battaile, Alex Parsons

Special Thanks: Zack Weger, Daniel Spicer, Sankara Shanmugam, Mahmoud El Halabi, Noah Rubin

January 29, 2021

Cyber security services offered by Stroz Friedberg Inc. and its affiliates. Insurance products and services offered by Aon Risk Insurance Services West, Inc., Aon Risk Services Central, Inc., Aon Risk Services Northeast, Inc., Aon Risk Services Southwest, Inc., and Aon Risk Services, Inc. of Florida and their licensed affiliates. Aon UK Limited is authorised and regulated by the Financial Conduct Authority in respect of insurance distribution services. FP.AGRC.238.JJ The following products or services are not regulated by the Financial Conduct Authority:

- Cyber risk services provided by Aon UK Limited and its affiliates
- Cyber security services provided by Stroz Friedberg Limited and its affiliates.

Copyright 2021 Aon plc. All Rights Reserved.

Cyber security services offered by Stroz Friedberg Inc. and its affiliates. Insurance products and services offered by Aon Risk Insurance Services West, Inc., Aon Risk Services Central, Inc., Aon Risk Services Northeast, Inc., Aon Risk Services Southwest, Inc., and Aon Risk Services, Inc. of Florida and their licensed affiliates. Aon UK Limited is authorised and regulated by the Financial Conduct Authority in respect of insurance distribution services. FP.AGRC.238.JJ The following products or services are not regulated by the Financial Conduct Authority:

- Cyber risk services provided by Aon UK Limited and its affiliates
- Cyber security services provided by Stroz Friedberg Limited and its affiliates.