New Phishing Attacks Exploiting OAuth Authorization Flows

August 7, 2021

Jenko Hwong
jhwong@netskope.com
@jenkohwong

$ az ad signed-in-user show


```
[
  {
    "jobTitle":   "Researcher",
    "department": "Threat Research Labs",
    "company":    "Netskope, Inc.",
    "email":      "jhwong@netskope.com"
    "twitter":    "@jenkohwong",

    "background": "vulnerability scanning, AV/AS, pen-testing/exploits,
                   L3/4 appliances, threat intel, windows security",
  }
]
```

Phishing Evolution: smtp, fake domain, ssl cert, user/pwd

in the beginning...

https://www.citibank-us.com

Apps | Trade | Netskope | Quest | Companies | Visualization | Baseball | Health | IT | Sweden | Tax | Housing | pb_save | my pinboard | read later | Quest for Space |...

citi

ATM / BRANCH

Credit Cards

How can we help? | Sign On

DST Root CA X3
Let's Encrypt Authority X3
www.citibank-us.com

www.citibank-us.com
Issued by: Let's Encrypt Authority X3
Expires: Tuesday, 7 November 2021 at 04:31:00 Central
This certificate is valid

Details

OK

otection Program.

purchases

Plus enjoy a low intro APR for 15 months on purchases and balance transfers.

Learn More

Sign On

User ID: joeblogs

Password: ••••••••••••

Remember my User ID

Sign On

Forgot User ID? | Activate a card

Forgot Password? | Register for online access

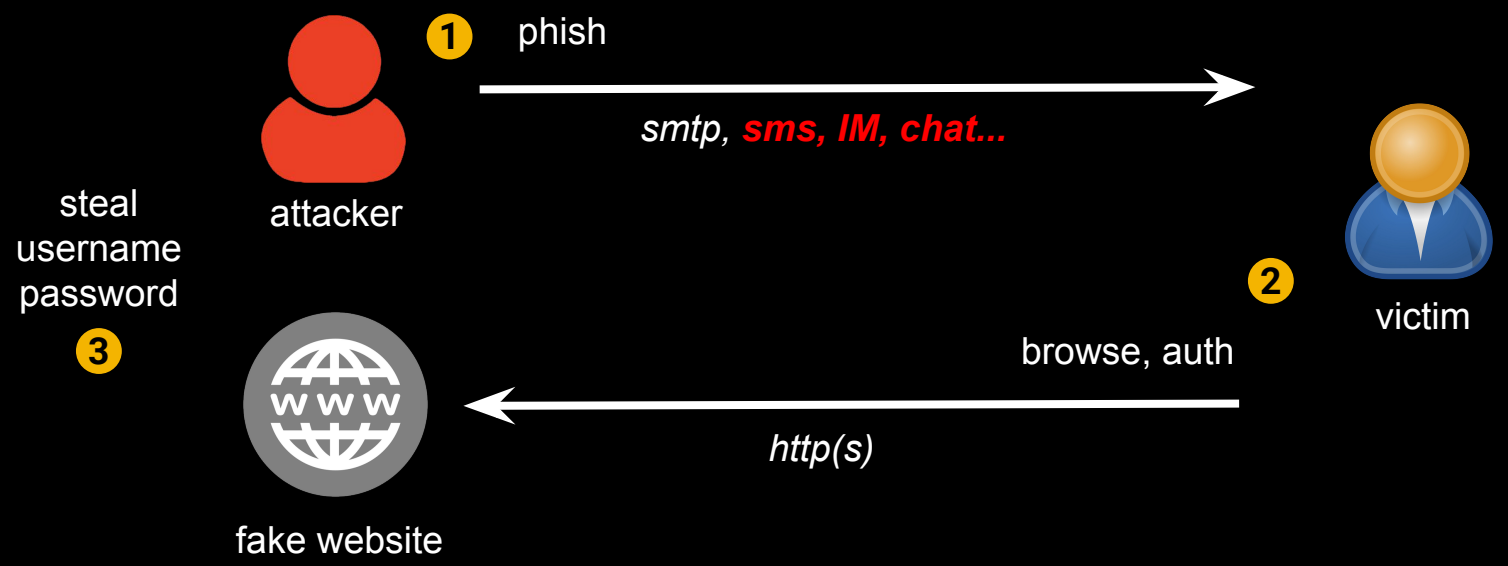
Low Intro APR Plus No Annual Fee

Choose the Right Citi® Credit Card for You

Earn Up To $1,500

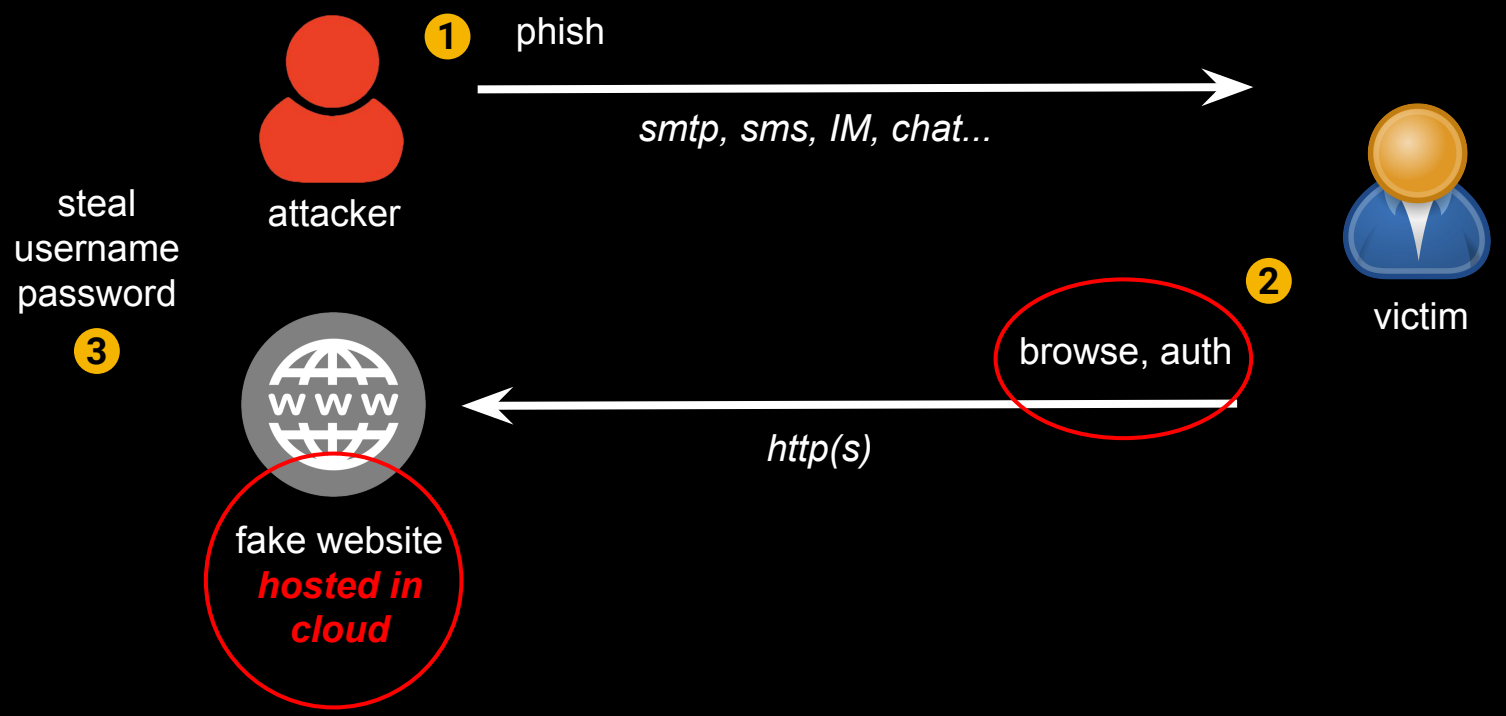
Phishing Evolution: apps, fake domain, ssl cert, user/pwd

+mobile



Phishing Evolution: apps, fake domain, ssl cert, user/pwd

+cloud



Phishing Evolution: apps, fake domain, ssl cert, user/pwd

+cloud

citi.cloudapp.azure.com

Click to go forward, hold to see history | Quest | Companies | Visualization | Baseball | Health | IT | Sweden | Tax | Housing | pb_save | my pinboard | read later | Quest for Space |...

citi

ATM / BRANCH | ESPAÑOL

Credit Cards | Banking | Lending | Investing | Wealth Management | Open an Account > | How can we help? | Sign On

View the latest details on our COVID-19 resources and the Paycheck Protection Program.

CITI CUSTOM CASH℠ CARD

New! Earn $200 cash back after qualifying purchases

Plus enjoy a low intro APR for 15 months on purchases and balance transfers.

Learn More

Sign On

User ID | Password

Remember my User ID

Sign On

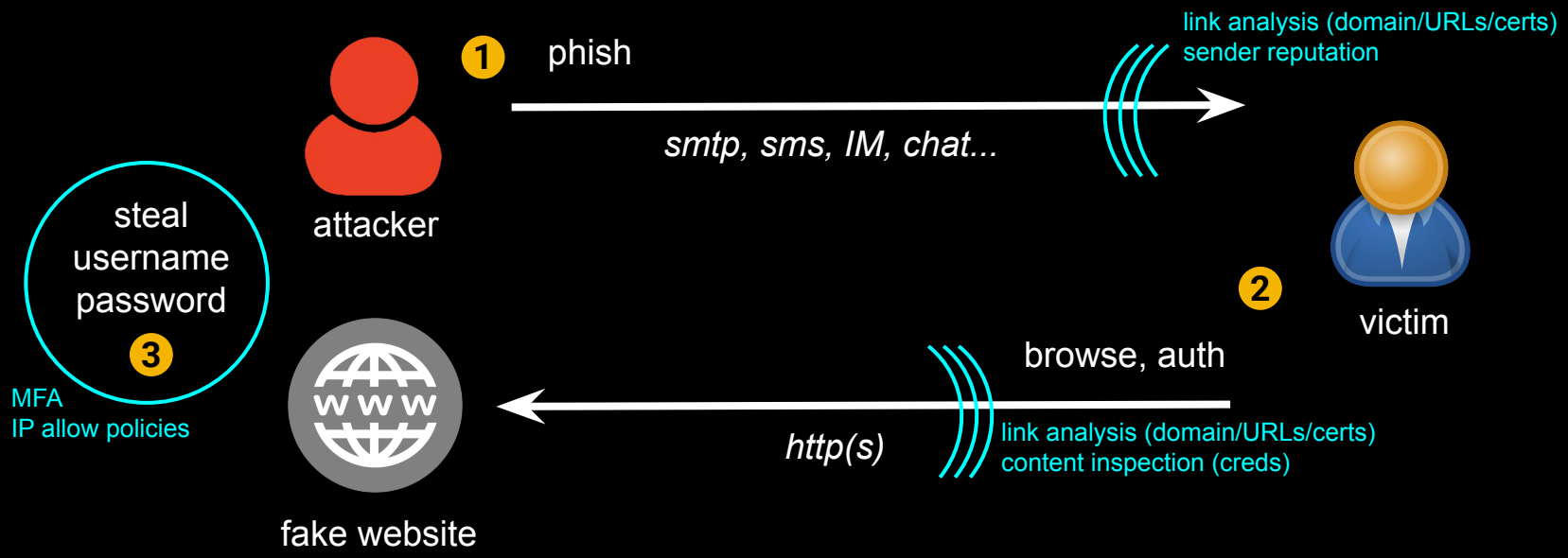
Forgot User ID? | Activate a card

Forgot Password? | Register for online access

Low Intro APR Plus No Annual Fee | Choose the Right Citi® Credit Card for You | Earn 0.50% APY High-Yield Interest

Phishing Evolution: fake domain, apps, ssl cert, user/pwd

controls



Phishing Evolution: OAuth 2.0 auth code grant[1]

+cloud app authorization

Azure AD

Google Identity

Identity Platform

3 **OAuth Tokens**
access token
refresh token

2 **Authenticate and Authorize**
Authenticate (MFA)
Authorize permissions (scopes)

Application
(client, device)

1 **Request Authorization**
Request permissions (scopes)
Redirect user to Identity Platform
(authorization service)

User

[1] https://datatracker.ietf.org/doc/html/rfc6749#page-24

Phishing Evolution: OAuth 2.0 auth code grant

+cloud app authorization: Payments

1 Item

Cart Subtotal

$229.99

CHECKOUT NOW

REP Sabre Olympic Bar - 20 kg

$229.99

Qty: 1

Remove

VIEW AND EDIT CART

PAYMENT METHOD

Credit Card

PayPal

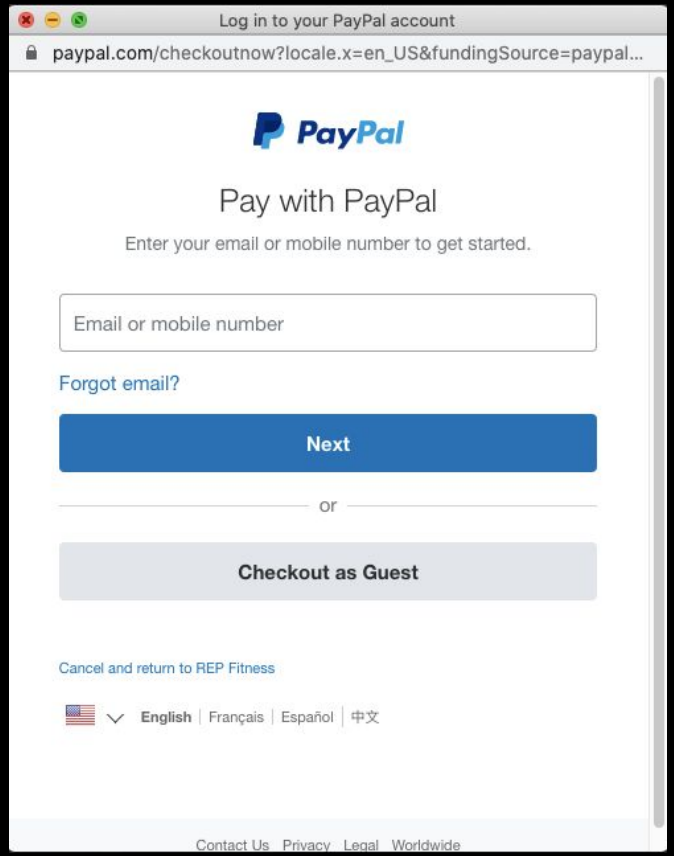
Pay with PayPal

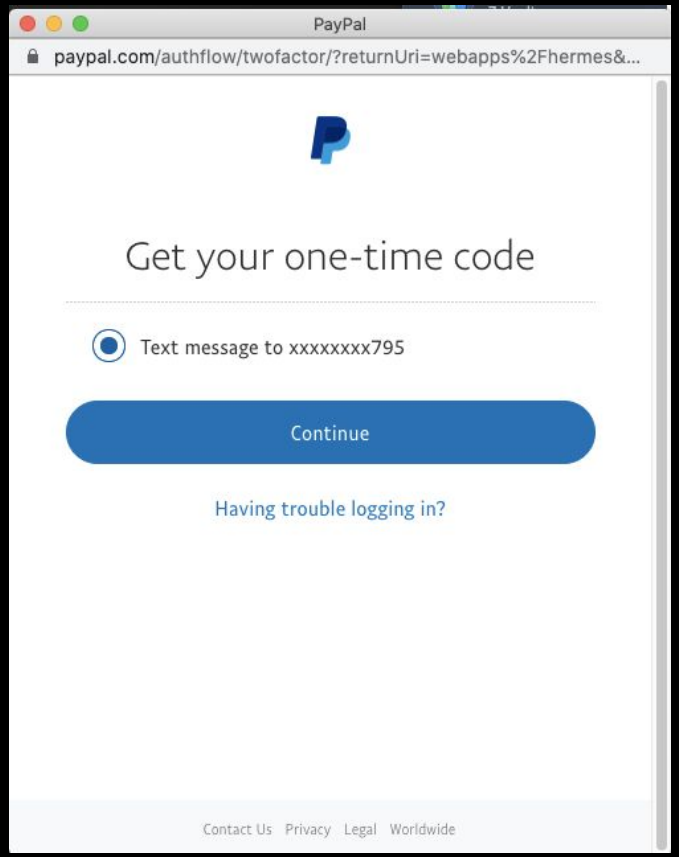
+ Apply Discount Code

+ Apply Gift Card

Phishing Evolution: OAuth 2.0 auth code grant

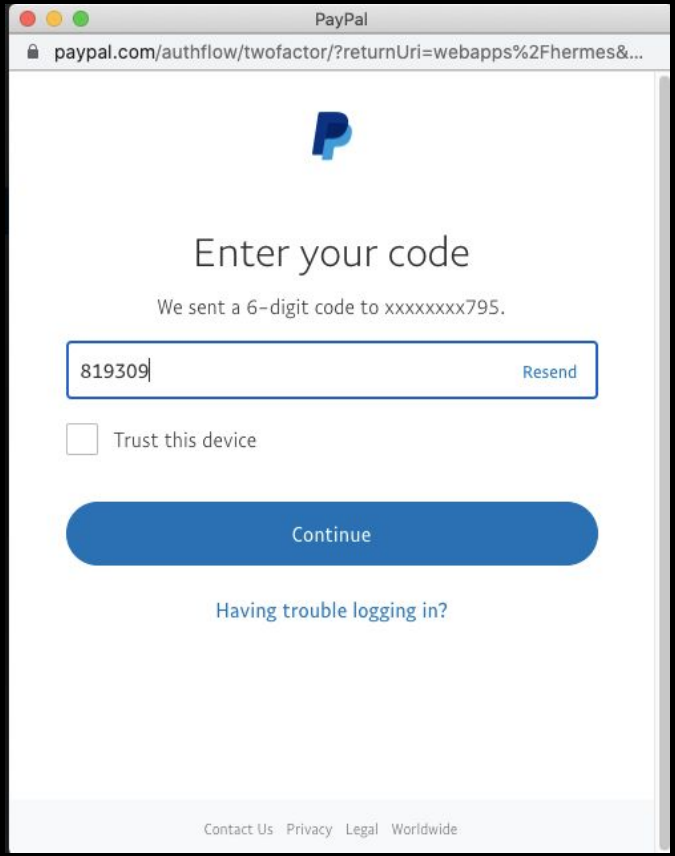
+cloud app authorization: Payments

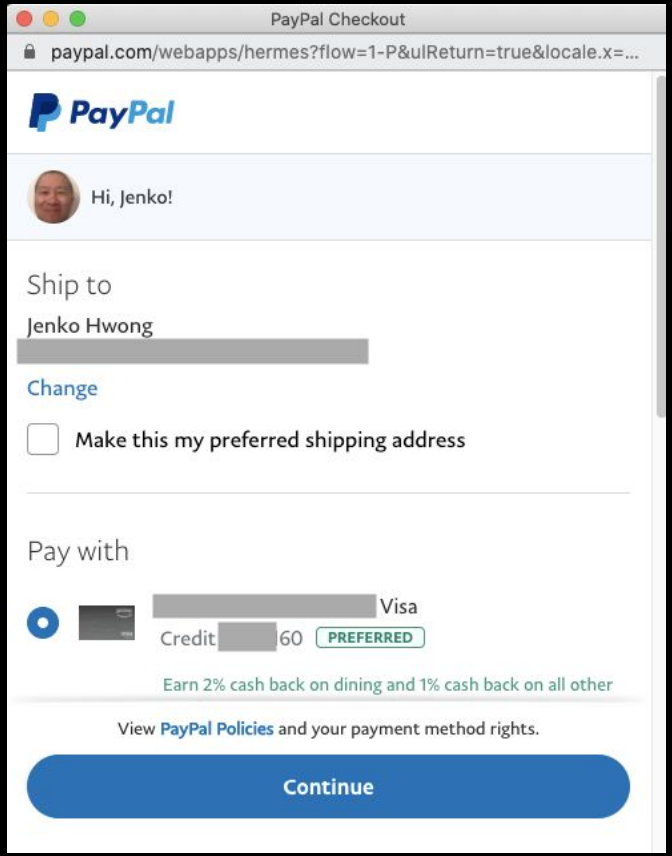




Phishing Evolution: OAuth 2.0 auth code grant

+cloud app authorization: Payments





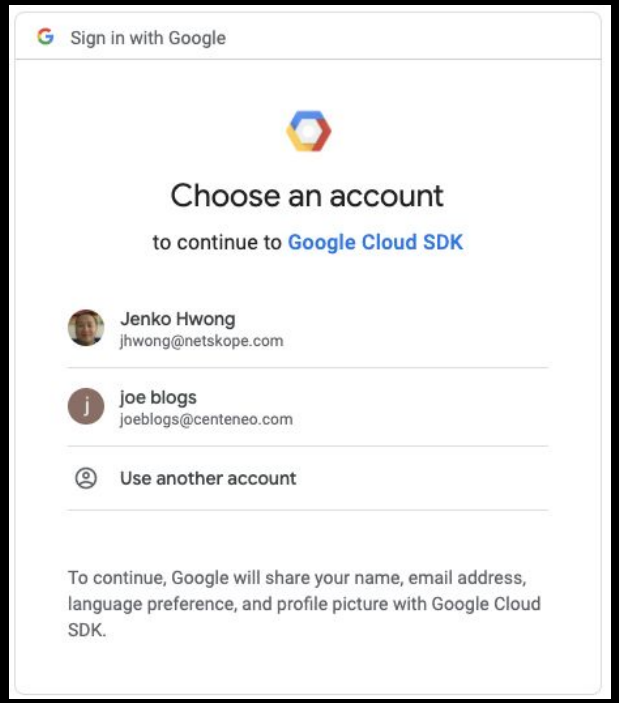
Phishing Evolution: OAuth 2.0 auth code grant

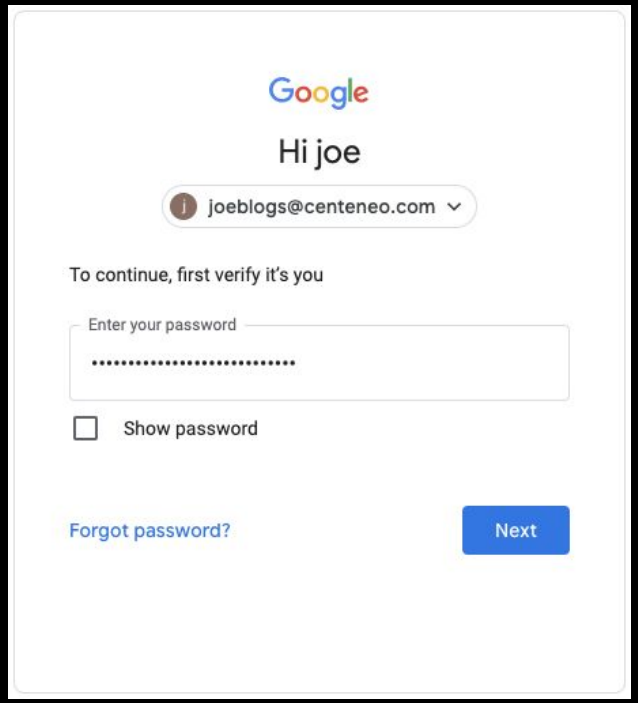
+cloud app authorization: GCP CLI

```
$ gcloud auth login joeblogs@centeneo.com --launch-browser --force

Your browser has been opened to visit:

https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=32555940559.apps.goo
gleusercontent.com&redirect_uri=http%3A%2F%2Flocalhost%3A8085%2F&scope=openid+https%3A%2F%2
Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud
-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fwww.googl
eapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts.reauth&state=
IMWlTK5Vlfab5gl4hKrleOxsylObop&access_type=offline&code_challenge=gU8ezZryqHCwAPyai2OLKaU-i
PvbR62biGjQgGV6IRE&code_challenge_method=S256
```

Phishing Evolution: OAuth 2.0 auth code grant

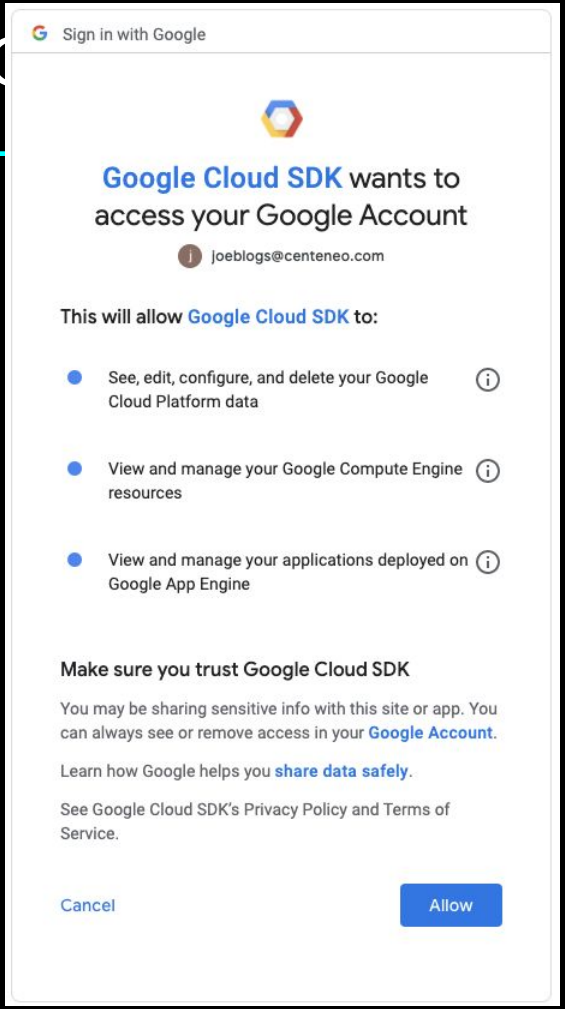
+cloud app authorization: GCP CLI





Phishing Evolution: G e grant

+cloud app authorization: GCP CL



Phishing Evolution

+cloud app authorization: GCP CL

Sign in with Google

Google Cloud SDK wants to

cloud.google.com/sdk/auth_success

Google Cloud

Why Google | Solutions | Products | Pricing | Getting Started

Cloud SDK: Command Line Interface

Overview | Guides | Reference | Support | Resources

Cloud SDK
- Product overview
- gcloud CLI overview
- gcloud CLI cheat sheet

Quickstarts
- All quickstarts
- Getting started with Cloud SDK

How-to guides
- All how-to guides
- Installing the SDK
- Setting up the SDK
- Managing SDK components
- Scripting guidelines

Cloud SDK: Command Line Interface > Documentation > Guides

You are now authenticated with the Google Cloud SDK!

The authentication flow has completed successfully. You may close this window, or check

Information about command-line tools and client libraries

To learn more about gcloud command-line tool commands, see the gcloud command-line

Phishing Evolution: OAuth 2.0 auth code grant

+cloud app authorization: GCP CLI

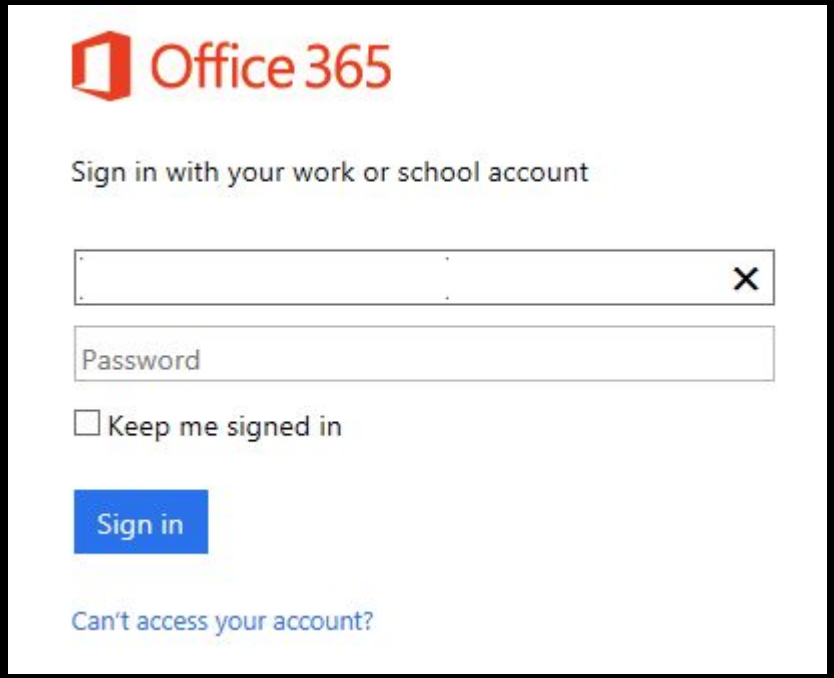
```
$ gcloud auth login joeblogs@centeneo.com --launch-browser --force

Your browser has been opened to visit:

https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=32555940559.apps.goo
gleusercontent.com&redirect_uri=http%3A%2F%2Flocalhost%3A8085%2F&scope=openid+https%3A%2F%2
Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud
-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fwww.googl
eapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts.reauth&state=
IMWlTK5Vlfab5gl4hKrleOxsylObop&access_type=offline&code_challenge=gU8ezZryqHCwAPyai2OLKaU-i
PvbR62biGjQgGV6IRE&code_challenge_method=S256

You are now logged in as [joeblogs@centeneo.com].

$
```

Phishing Evolution: fake OAuth login

+cloud app authorization

Phishing Evolution: fake OAuth login, check creds

+cloud app authorization



- Real-time creds validation (APIs)[1]


Azure AD

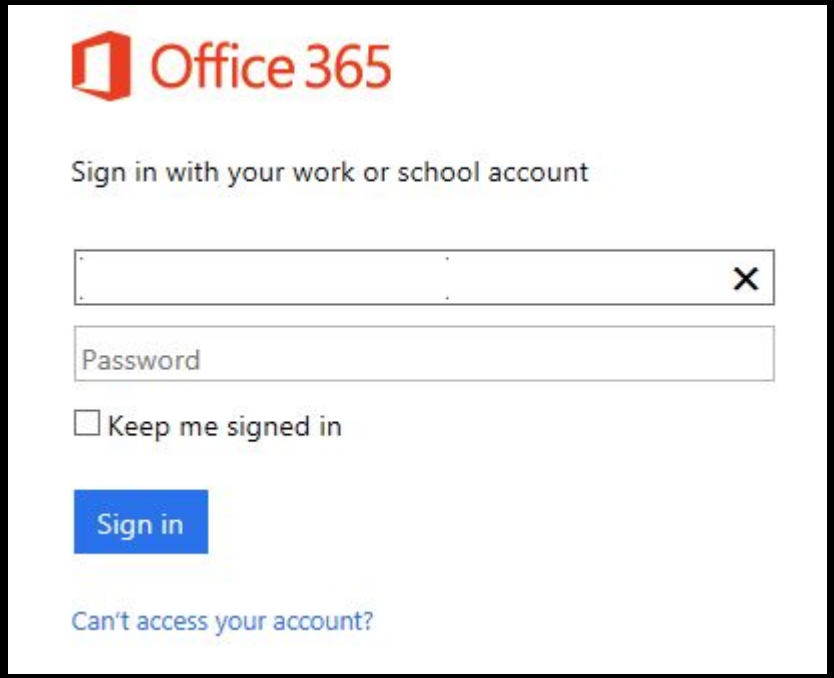

Google Identity

- Based on pass/fail, redirect user to valid domains (stealth, creds validation upfront)

[1] https://threatpost.com/office-365-phishing-attack-leverages-real-time-active-directory-validation/159188/

Phishing Evolution: fake OAuth login, check creds

+cloud app authorization



- Real-time creds validation (APIs)[1]


Azure AD


Google Identity

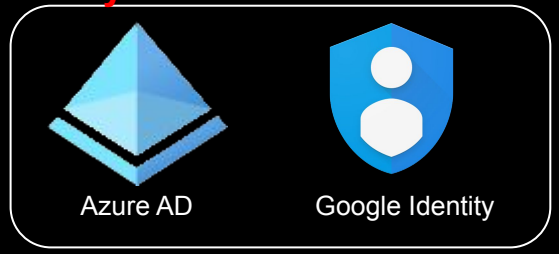
- Controls
  - MFA, IP allow policies
  - link analysis (domain/URLs/certs)
  - content inspection (creds)
  - sender reputation

[1] https://threatpost.com/office-365-phishing-attack-leverages-real-time-active-directory-validation/159188/

Phishing Evolution: OAuth 2.0 auth code grant

+cloud app authorization protocol -- why do we care ?

1. Hijack session tokens, not creds
2. REST APIs <=> remote exploit vs endpoint



Identity Platform

Oauth tokens

```
{ "access_token": "ya29.a0ARrdaM9...",
  "refresh_token": "1//06S3ISKyEHY…",
  "scope": "https://www.googleapis...",
  "expires_in": 3599,
  "token_type": "Bearer"
}
```

6

5

Request oauth tokens

POST https://www.googleapis.com/oauth2/v4/token
client_id=32555940559.apps.googleusercontent....&
scope=https://www.googleapis.com/auth/cloud...&
client_secret=JqQXA298PB…&
code=AwABAAAAvPM1KaP…&
redirect_uri=www.myapp.com:9000

3 **Authenticate and Authorize**

GET https://accounts.google.com/o/oauth2/v2/auth?
client_id=32555940559.apps.googleusercontent.com&
response_type=code&
scope=https://www.googleapis.com/auth/cloud-platform&
access_type=offline&redirect_uri=www.myapp.com:9000

(authenticate, MFA, consent to scopes)

4 **Redirect URL with Authorization Code**

GET http://www.myapp.com:9000?
code=AwABAAAAvPM1KaP...

2 **Redirect to Identity Platform**

1 **Login / Checkout / Install App**

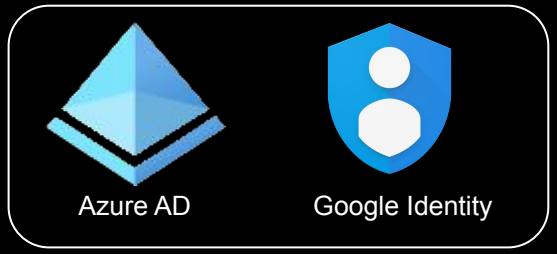
Application
(client, device)

User

Phishing Evolution: OAuth 2.0 illicit consent grants

+cloud app authorization protocol

1. Malicious registered application
2. Get user consent for wide scopes / permissions



Oauth tokens

6

```
{ "access_token": "ya29.a0ARrdaM9...",
  "refresh_token": "1//06S3lSKvEHY...",
  "scope": "https://www.googleapis...",
  "expires_in": 3599,
  "token_type": "Bearer"
}
```

Request oauth tokens

5

POST https://www.googleapis.com/oauth2/v4/token
client_id=32555940559.apps.googleusercontent....&
scope=https://www.googleapis.com/auth/cloud...&
client_secret=JqQXA298PB...&
code=AwABAAAAvPM1KaP...&
redirect_uri=www.myapp.com:9000

3 **Authenticate and Authorize**

GET https://accounts.google.com/o/oauth2/v2/auth?
client_id=32555940559.apps.googleusercontent.com&
response_type=code&
scope=https://www.googleapis.com/auth/cloud-platform&
access_type=offline&redirect_uri=www.myapp.com:9000

(authenticate, MFA, consent to scopes)

4 **Redirect URL with Authorization Code**

GET http://www.myapp.com:9000?
code=AwABAAAAvPM1KaP...

2 **Redirect to Identity Platform**

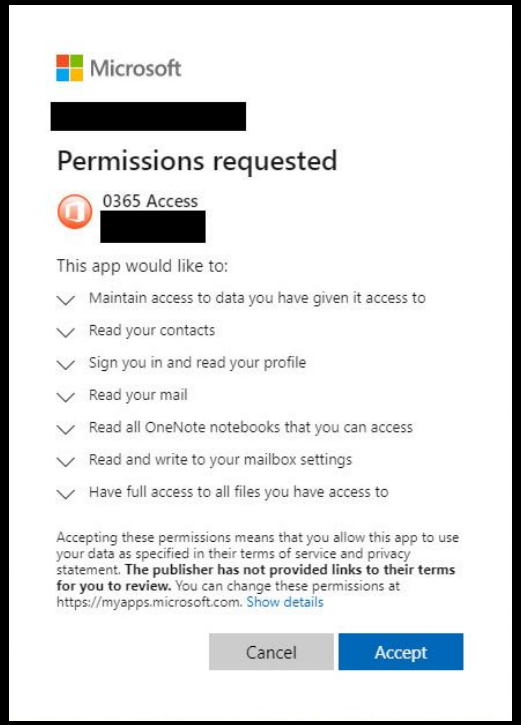
1 **Login / Checkout / Install App**

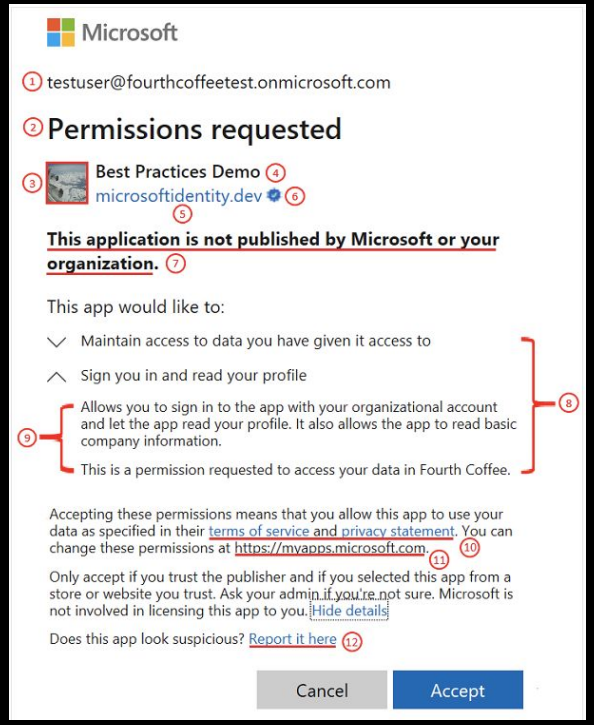
Application
(client, device)

User

Phishing Evolution: OAuth 2.0 illicit consent grants[1]

+cloud app authorization protocol





[2]

1. Malicious registered application
2. Get user consent for wide scopes / permissions

Controls

1. Prevent users from registering apps in AD
2. Prevent users from consenting

[1] https://www.bleepingcomputer.com/news/security/phishing-attack-hijacks-office-365-accounts-using-oauth-apps/
[2] https://docs.microsoft.com/en-us/azure/active-directory/develop/application-consent-experience

Phishing Evolution: OAuth 2.0 device code authorization[1]

what's the purpose? to provide easier authentication/authorization on limited input devices e.g. smart TVs

[1] https://datatracker.ietf.org/doc/html/rfc8628

“I think there's an RFC for that.”

datatracker.ietf.org/doc/html/rfc8628

[Search] [txt|html|pdf|bibtex] [Tracker] [WG] [Email] [Diff1] [Diff2] [Nits]

From: draft-ietf-oauth-device-flow-15 Proposed Standard
Errata exist

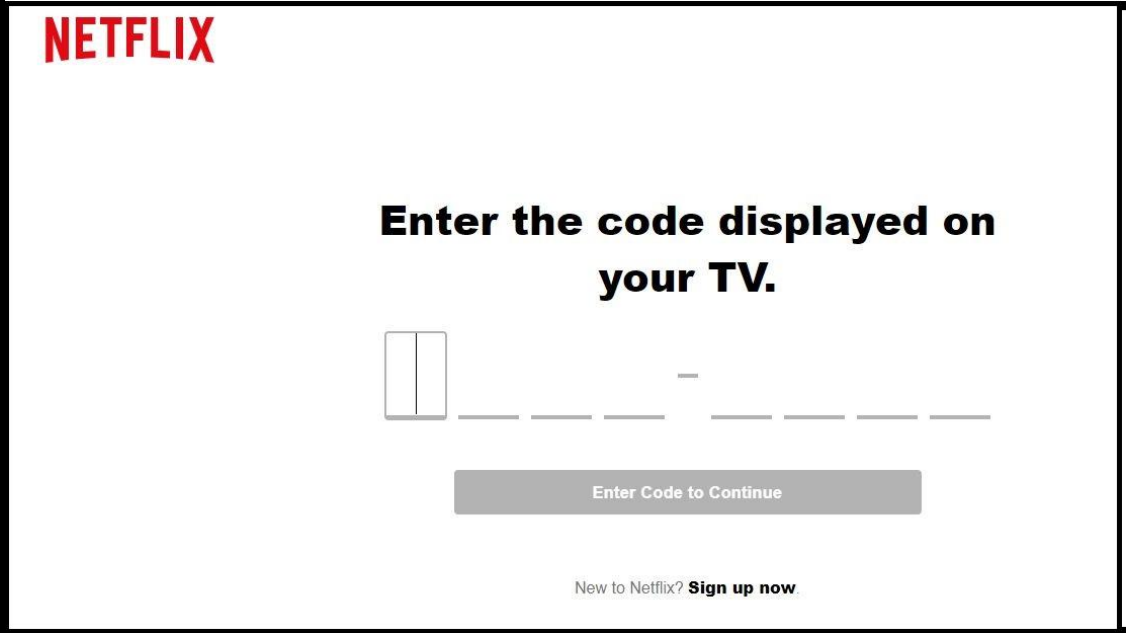
Internet Engineering Task Force (IETF)
Request for Comments: 8628
Category: Standards Track
ISSN: 2070-1721

W. Denniss
Google
J. Bradley
Ping Identity
M. Jones
Microsoft
H. Tschofenig
ARM Limited
August 2019

OAuth 2.0 Device Authorization Grant

which, when implemented, looks something like this on your TV

with the real sign-in on a computer or mobile phone



If your device generates an activation code, you will need to enter that code on our website by doing the following:

1.Navigate to Netflix.com/activate.

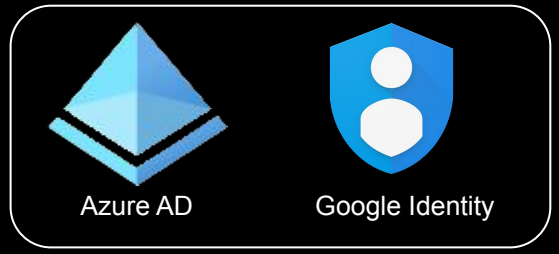
2.After signing in, select the profile you would like to watch Netflix from.

3.Enter the code in the Enter code field. Click Activate.

Unusability is the father of insecurity

Phishing Evolution: OAuth 2.0 device code authorization[1]

+cloud app authorization protocol



Identity Platform

2 **Get user/device codes**

Retrieve oauth tokens 5
client_id
device_code

3 **Instruct user to login**
on computer/smartphone
"1. Go to www.google.com/device
2. Enter user code: ZLGG-LOSP"

4 **Authenticate and Authorize**
1. Goes to www.google.com/device
2. Enters: ZLGG-LOSP
3. Authenticates, including MFA

1 **Login**

Device
(client, app)

User

[1] https://datatracker.ietf.org/doc/html/rfc8628

Demo: OAuth 2.0 device code authorization

- Dr. Nestori Syynimaa: https://o365blog.com/post/phishing/

- Usability => insecurity
- A different auth flow => opportunity
- Implementation quirks

Phishing Evolution: OAuth 2.0 device code authorization

+cloud app authorization protocol

Get user/device codes
POST
https://login.microsoftonline.com/common/oauth2/devicecode?api-version=1.0
client_id=d3590ed6-52b3-4102-aeff-aad2292ab01c&
resource=https://outlook.office365.com

2

3

User/device codes
{ "device_code": "AH-1NgM6boio...",
"verification_url":
"https://www.google.com/device",
"user_code": "ZLGG-LQSP",
"expires_in": 1800,
"interval": 5
}

Poll for oauth tokens
client_id
device_code

6

7

Oauth tokens
{ "access_token": "ya29.a0ARrdaM9...",
"refresh_token": "1//06S3ISKyEHY…",
}

Azure AD

Google Identity

Identity Platform

5 **Authenticate and Authorize**
1. Goes to www.google.com/device
2. Enters: ZLGG-LOSP
3. Authenticates, including MFA

4 **User code, verification URL**
manual instructions:
"1. Go to www.google.com/device
2. Enter: ZLGG-LOSP"

1 **Login**

Device
(client, app)

User

Phishing Evolution: OAuth 2.0 device code authorization microsoft phish

+cloud app authorization protocol

Get user/device codes

POST
https://login.microsoftonline.com/common/oauth2/devicecode?api-version=1.0
client_id=d3590ed6-52b3-4102-aeff-aad2292ab01c&
resource=https://outlook.office365.com

2

3

User/device codes

```
{ "device_code": "AH-1NgM6boio...",
  "verification_url":
  "https://www.google.com/device",
  "user_code": "ZLGG-LQSP",
  "expires_in": 1800,
  "interval": 5
}
```

Poll for oauth tokens
client_id
device_code

6

7

Oauth tokens

```
{ "access_token": "ya29.a0ARrdaM9...",
  "refresh_token": "1//06S3lSKyEHY...",
}
```

Azure AD

Google Identity

Identity Platform

5 **Authenticate and Authorize**
1. Goes to www.google.com/device
2. Enters: ZLGG-LOSP
3. Authenticates, including MFA

4 **Phish**
"Here's your promotional product code:
1. Go to www.google.com/device
2. Enter: ZLGG-LOSP"

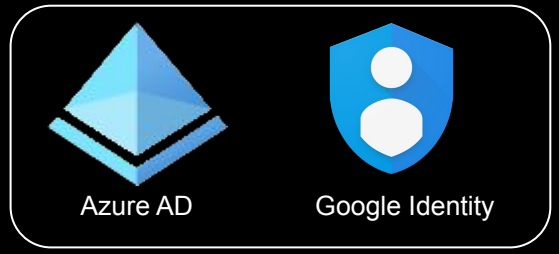
XX **Login**

Device
(client, app)

User

Phishing Evolution: OAuth 2.0 device code authorization microsoft phish

+cloud app authorization protocol



Identity Platform

Access Token


```
{ "scope": "user_impersonation",
  "resource": "https://management.azure.com",
  "access_token": "eyJ0eXAiOiJKV1QiLCJhbG...",
  "refresh_token": "0.AUYAAknJ93kbWUyXs2…",
}
```


9

8

Use refresh token to get new access token for Azure


```
{  "refresh_token": "1//06S3lSKyEHY…",
  "scope": "openid",
  "grant_type": "refresh_token"
  "resource": "https://management.azure.com",
  "client_id": "d3590ed6-52b3-4102-aeff-aad2292ab01c",
}
```


Device
(client, app)

Phishing Evolution: OAuth 2.0 device code authorization microsoft phish

+cloud app authorization protocol

1. No server infrastructure
2. No registered application, use existing vendor client app
3. No consent screen

Get user/device codes

POST
https://login.microsoftonline.com/common/oauth2/devicecode?api-version=1.0
client_id=d3590ed6-52b3-4102-aeff-aad2292ab01c&
resource=https://outlook.office365.com

2

3

User/device codes

```
{ "device_code": "AH-1NgM6boio...",
  "verification_url":
  "https://www.google.com/device",
  "user_code": "ZLGG-LQSP",
  "expires_in": 1800,
  "interval": 5
}
```

Azure AD

Google Identity

Identity Platform

Poll for oauth tokens
client_id
device_code

6

7

Oauth tokens

```
{ "access_token": "ya29.a0ARrdaM9...",
  "refresh_token": "1//06S3lSKyEHY…",
}
```

5 **Authenticate and Authorize**
1. Goes to www.google.com/device
2. Enters: ZLGG-LOSP
3. Authenticates, including MFA

4 **Phish**
"Here's your promotional product code:
1. Go to www.google.com/device
2. Enter: ZLGG-LOSP"

XX **Login**

Device
(client, app)

User

Phishing Evolution: OAuth 2.0 device code authorization microsoft phish

+cloud app authorization protocol

Get user/device codes

POST
https://login.microsoftonline.com/common/oauth2/devicecode?api-version=1.0
client_id=d3590ed6-52b3-4102-aeff-aad2292ab01c&
resource=https://outlook.office365.com

2

3

User/device codes

```
{ "device_code": "AH-1NgM6boio...",
  "verification_url":
  "https://www.google.com/device",
  "user_code": "ZLGG-LQSP",
  "expires_in": 1800,
  "interval": 5
}
```

Poll for oauth tokens
client_id
device_code

6

7

Oauth tokens

```
{ "access_token": "ya29.a0ARrdaM9...",
  "refresh_token": "1//06S3lSKyEHY...",
}
```

Azure AD

Google Identity

Identity Platform

5 **Authenticate and Authorize**
1. Goes to www.google.com/device
2. Enters: ZLGG-LOSP
3. Authenticates, including MFA

4 **Phish**
"Here's your promotional product code:
1. Go to www.google.com/device
2. Enter: ZLGG-LOSP"

1. No server infrastructure
2. No registered application, use existing vendor client app
3. No consent screen
4. Implicit, default scopes

Device
(client, app)

User

Phishing Evolution: OAuth 2.0 device code authorization microsoft phish

+cloud app authorization protocol

Microsoft Azure | Search resources, services, and docs (G+/) | ed@feasthealth.onmicr... FEAST HEALTHCARE (FEASTHEAL...

Home > Ed Van

Ed Van | Sign-ins

User

Download | Export Data Settings | Troubleshoot | Refresh | Columns | Got feedback?

Want to switch back to the default sign-ins experience? Click here to leave the preview. →

Date : **Last 7 days** | Show dates as : **Local** | User contains **e731a6d2-ba0c-46f3-84bb-167f488cecda** | Application contains **Microsoft Office** | Status : **Success** | Add filters

User sign-ins (interactive) | User sign-ins (non-interactive)

Date	Username	Application	Sta...	IP address	Location	Resource	Client app	Authentication requirement
7/14/2021, 11:30:45 AM	ed@feasthealth.onmicrosoft.com	Microsoft Office 365 Portal	Success	143.XXX.XXX.25	Sin City, Nevada, US	Windows Azure Active D...	Browser	Multi-factor authentication
7/13/2021, 12:16:30 PM	ed@feasthealth.onmicrosoft.com	Microsoft Office 365 Portal	Success	143.XXX.XXX.25	Sin City, Nevada, US	Windows Azure Active D...	Browser	Multi-factor authentication
7/12/2021, 8:25:17 AM	ed@feasthealth.onmicrosoft.com	Microsoft Office	Success	143.XXX.XXX.25	Sin City, Nevada, US	Microsoft Graph	Mobile Apps and Desktop clients	Multi-factor authentication
7/12/2021, 12:59:17 AM	ed@feasthealth.onmicrosoft.com	Microsoft Office	Success	143.XXX.XXX.25	Sin City, Nevada, US	Microsoft Graph	Mobile Apps and Desktop clients	Multi-factor authentication
7/12/2021, 12:56:43 AM	ed@feasthealth.onmicrosoft.com	Microsoft Office	Success	143.XXX.XXX.25	Sin City, Nevada, US	Microsoft Graph	Mobile Apps and Desktop clients	Multi-factor authentication
7/12/2021, 12:48:06 AM	ed@feasthealth.onmicrosoft.com	Microsoft Office	Success	143.XXX.XXX.25	Sin City, Nevada, US	Microsoft Graph	Mobile Apps and Desktop clients	Multi-factor authentication
7/12/2021, 12:32:28 AM	ed@feasthealth.onmicrosoft.com	Microsoft Office	Success	143.XXX.XXX.25	Sin City, Nevada, US	Microsoft Graph	Mobile Apps and Desktop clients	Multi-factor authentication
7/12/2021, 12:17:28 AM	ed@feasthealth.onmicrosoft.com	Microsoft Office	Success	143.XXX.XXX.25	Sin City, Nevada, US	Microsoft Graph	Mobile Apps and Desktop clients	Multi-factor authentication
7/12/2021, 12:09:05 AM	ed@feasthealth.onmicrosoft.com	Microsoft Office	Success	143.XXX.XXX.25	Sin City, Nevada, US	Microsoft Graph	Mobile Apps and Desktop clients	Multi-factor authentication
7/12/2021, 12:06:00 AM	ed@feasthealth.onmicrosoft.com	Microsoft Office	Success	143.XXX.XXX.25	Sin City, Nevada, US	Microsoft Graph	Mobile Apps and Desktop clients	Multi-factor authentication
7/11/2021, 11:50:52 PM	ed@feasthealth.onmicrosoft.com	Microsoft Office	Success	143.XXX.XXX.25	Sin City, Nevada, US	Microsoft Graph	Mobile Apps and Desktop clients	Multi-factor authentication
7/11/2021, 11:48:54 PM	ed@feasthealth.onmicrosoft.com	Microsoft Office	Success	143.XXX.XXX.25	Sin City, Nevada, US	Microsoft Graph	Mobile Apps and Desktop clients	Multi-factor authentication
7/11/2021, 11:43:05 PM	ed@feasthealth.onmicrosoft.com	Microsoft Office	Success	143.XXX.XXX.25	Sin City, Nevada, US	Microsoft Graph	Mobile Apps and Desktop clients	Multi-factor authentication
7/11/2021, 11:33:44 PM	ed@feasthealth.onmicrosoft.com	Microsoft Office	Success	143.XXX.XXX.25	Sin City, Nevada, US	Microsoft Graph	Mobile Apps and Desktop clients	Multi-factor authentication
7/11/2021, 11:31:16 PM	ed@feasthealth.onmicrosoft.com	Microsoft Office	Success	143.XXX.XXX.25	Sin City, Nevada, US	Microsoft Graph	Mobile Apps and Desktop clients	Multi-factor authentication
7/11/2021, 11:22:56 PM	ed@feasthealth.onmicrosoft.com	Microsoft Office	Success	143.XXX.XXX.25	Sin City, Nevada, US	Microsoft Graph	Mobile Apps and Desktop clients	Multi-factor authentication
7/11/2021, 11:14:42 PM	ed@feasthealth.onmicrosoft.com	Microsoft Office	Success	143.XXX.XXX.25	Sin City, Nevada, US	Microsoft Graph	Mobile Apps and Desktop clients	Multi-factor authentication
7/11/2021, 11:11:55 PM	ed@feasthealth.onmicrosoft.com	Microsoft Office	Success	143.XXX.XXX.25	Sin City, Nevada, US	Microsoft Graph	Mobile Apps and Desktop clients	Multi-factor authentication
7/11/2021, 8:35:26 PM	ed@feasthealth.onmicrosoft.com	Microsoft Office	Success	143.XXX.XXX.25	Sin City, Nevada, US	Microsoft Graph	Mobile Apps and Desktop clients	Multi-factor authentication

1. No server infrastructure
2. No registered application, use existing vendor client app
3. No consent screen
4. Implicit, default scopes
5. Move laterally to other services
6. Logging limited (initial token logged as sign-in, but lateral move is not)

Phish

+cloud ap

Activity Details: Sign-ins

Basic info | Location | Device info | Authentication Details | Conditional Access | Report-only | Additional Details

Date	7/12/2021, 8:25:17 AM
Request ID	ee30da7a-0f2e-4936-b64f-00da59f11200
Correlation ID	eba1a1ae-fec7-4670-b4be-d6cd063dc4b1
Authentication requirement	Multi-factor authentication
Status	Success
Continuous access evaluation	No
User	Ed Van
Username	ed@feasthealth.onmicrosoft.com
User ID	e731a6d2-ba0c-46f3-84bb-167f488cecda
Sign-in identifier	
User type	Member
Cross tenant access type	None
Application	Microsoft Office
Application ID	d3590ed6-52b3-4102-aeff-aad2292ab01c
Resource	Microsoft Graph
Resource ID	00000003-0000-0000-c000-000000000000
Resource tenant ID	f7c94902-1b79-4c59-97b3-62503ab64e53
Home tenant ID	f7c94902-1b79-4c59-97b3-62503ab64e53
Client app	Mobile Apps and Desktop clients
Token issuer type	Azure AD
Token issuer name	
Latency	612ms
Flagged for review	No
User agent	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.114 Safari/537.36

de authorization microsoft phish

1. No server infrastructure
2. No registered application, use existing vendor client app
3. No consent screen
4. Implicit, default scopes
5. Move laterally to other services
6. Logging limited (initial token logged as sign-in, but lateral move is not)

Phishing Evolution: OAuth 2.0 device code authorization microsoft phish

controls

1. Prevent: block verification URIs, use conditional access policies
   - https://oauth2.googleapis.com/device/code
   - https://microsoft.com/devicelogin
   - https://login.microsoftonline.com/common/oauth2/deviceauth
   - block access based on IP, location, endpoint characteristics
2. Detect
   - Difficult
3. Remediate
   - API to revoke all oauth tokens for a user

1. No server infrastructure
2. No registered application, use existing vendor client app
3. No consent screen
4. Implicit, default scopes
5. Move laterally to other services
6. Logging limited (initial token logged as sign-in, but lateral move is not)

Phishing Evolution: OAuth 2.0 device code authorization microsoft phish

controls

1. Prevent: block verification URIs, use conditional access policies
   - https://oauth2.googleapis.com/device/code
   - https://microsoft.com/devicelogin
   - https://login.microsoftonline.com/common/oauth2/deviceauth
   - block access based on IP, location, endpoint characteristics
2. Detect
   - https://login.microsoftonline.com/common/oauth2/devi
3. Remediate
   - API to revoke all oauth tokens for a user

1. No server infrastructure
2. No registered application, use existing vendor client app
3. No consent screen
4. Implicit, default scopes
5. Move laterally to other services
6. Logging limited (initial token logged as sign-in, but lateral move is not)

practical considerations

Short expiration of user/device codes (15-30mins)

- phishing numbers game
- incorporate hosted website, generate codes dynamically
- use images for user code (no javascript allowed in email clients)

OAuth 2.0 device code authorization

	Microsoft	Google
Server infrastructure	None required	None required
Application registration	None needed, can use large # of existing apps	Some limited vendor apps e.g. Chrome
Consent screens	No	**Partial** (limited vendor apps)
Scopes	Implicit, default scopes, wide-range	**Very limited** (user profile, drive access to app files, youtube info)
Lateral movement	Easy to switch among large number of services	**No**: strict limited scopes for device code flow
Logging	Partial (initial token access)	Partial
Prevention	block URIs, cond access	block URIs, VPC perimeters
Detection	Difficult	Difficult
Remediation	API to revoke user tokens	Delete/recreate user

Ongoing Research Areas

- Other flows[1]
- Any usability "requirements"
- Bypass consent e.g. implicit grants
- Default scopes[2]
- Consent[3]
- Browser auto-login and scope expansion e.g. Google uberauth (2013)[4][5]

With the plans for third party cookies to be removed from browsers, the **implicit grant flow is no longer a suitable authentication method**. The silent SSO features of the implicit flow do not work without third party cookies, causing applications to break when they attempt to get a new token. We strongly recommend that all new applications use the authorization code flow that now supports

Getting access tokens silently in the background

ⓘ **Important**

This part of the implicit flow is unlikely to work for your application as it's used across different browsers due to the **removal of third party cookies by default**. While this still currently works in Chromium-based browsers that are not in Incognito, developers should reconsider using this part of the flow. In browsers that do not support third party cookies, you will recieve an error indicating that no users are signed in, as the login page's session cookies were removed by the browser.

Incremental and dynamic user consent

With the Microsoft identity platform endpoint, you can ignore the static permissions defined in the app registration information in the Azure portal and request permissions incrementally instead. You can ask for a bare minimum set of permissions upfront and request more over time as the customer uses additional app features. To do so, you can specify the scopes your app needs at any time by including the new scopes in the `scope` parameter when requesting an access token - without the need to pre-define them in the application registration information. If the user hasn't yet consented to new scopes added to the request, they'll be prompted to consent only to the new permissions. Incremental, or dynamic consent, only applies to delegated permissions and not to application permissions.

[1] https://datatracker.ietf.org/doc/html/rfc6749#page-23
[2] https://docs.microsoft.com/en-us/azure/active-directory/develop/v2-permissions-and-consent
[3] https://docs.microsoft.com/en-us/azure/active-directory/develop/v2-permissions-and-consent
[4] https://gist.github.com/arirubinstein/fd5453537436a8757266f908c3e41538
[5] https://duo.com/blog/beyond-the-vulnerabilities-of-the-application-specific-password-exploiting-google-chrome-s-oauth2-tokens

Thank you

Questions

Open Source Tools

- Repo: https://github.com/netskopeoss/phish_oauth
- License: BSD-3-Clause

Contact

- jhwong@netskope.com
- @jenkohwong

References

1.0 Evolving Phishing Attacks

1.1 A Big Catch: Cloud Phishing from Google App Engine and Azure App Service: https://www.netskope.com/blog/a-big-catch-cloud-phishing-from-google-app-engine-and-azure-app-service

1.2 Microsoft Seizes Malicious Domains Used in Mass Office 365 Attacks: https://threatpost.com/microsoft-seizes-domains-office-365-phishing-scam/157261/

1.3 Phishing Attack Hijacks Office 365 Accounts Using OAuth Apps: https://www.bleepingcomputer.com/news/security/phishing-attack-hijacks-office-365-accounts-using-oauth-apps/

1.4 Office 365 Phishing Attack Leverages Real-Time Active Directory Validation: https://threatpost.com/office-365-phishing-attack-leverages-real-time-active-directory-validation/159188/

1.5 Demonstration - Illicit Consent Grant Attack in Azure AD: https://www.nixu.com/blog/demonstration-illicit-consent-grant-attack-azure-ad-office-365 https://securecloud.blog/2018/10/02/demonstration-illicit-consent-grant-attack-in-azure-ad-office-365/

1.6 Detection and Mitigation of Illicit Consent Grant Attacks in Azure AD: https://www.cloud-architekt.net/detection-and-mitigation-consent-grant-attacks-azuread/

1.7 HelSec Azure AD write-up: Phishing on Steroids with Azure AD Consent Extractor: https://securecloud.blog/2019/12/17/helsec-azure-ad-write-up-phishing-on-steroids-with-azure-ad-consent-extractor/

1.8 Pawn Storm Abuses OAuth In Social Engineering Attack: https://www.trendmicro.com/en_us/research/17/d/pawn-storm-abuses-open-authentication-advanced-social-engineering-attacks.html

2.0 OAuth Device Code Flow

2.1 OAuth 2.0 RFC: https://tools.ietf.org/html/rfc6749

2.2 OAuth 2.0 Device Authorization Grant RFC: https://datatracker.ietf.org/doc/html/rfc8628

2.3 OAuth 2.0 for TV and Limited-Input Device Applications: https://developers.google.com/identity/protocols/oauth2/limited-input-device

2.4 OAuth 2.0 Scopes for Google APIs: https://developers.google.com/identity/protocols/oauth2/scopes

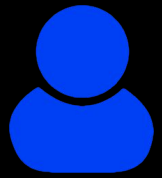
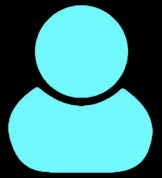
2.5 Introducing a new phishing technique for compromising Office 365 accounts: https://o365blog.com/post/phishing/#oauth-consent

2.6. Office Device Code Phishing: https://gist.github.com/Mr-Un1k0d3r/afef5a80cb72dfeaa78d14465fb0d333

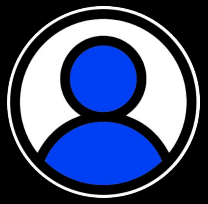
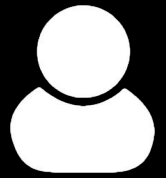
3.0 Additional OAuth Research Areas

3.1 Poor OAuth implementation leaves millions at risk of stolen data: https://searchsecurity.techtarget.com/news/450402565/Poor-OAuth-implementation-leaves-millions-at-risk-of-stolen-data

3.2 How did a full access OAuth token get issued to the Pokémon GO app?: https://searchsecurity.techtarget.com/answer/How-did-a-full-access-OAuth-token-get-issued-to-the-Pokemon-GO-app



1 2 3 4 5 6 7 8 9