Laundering C2 Traffic

Through High-Reputation Services

IBM®

Who am I?

Ruben Boonen

Antiquarian, **10+** years in the industry

CNE Capability Lead @ **IBM Adversary Services**

Director @ Calypso Heavy Industries (**CHI**)

@FuzzySec

Windows

Post-Exploitation

Strange R&D

Vulnerability Research

Introduction

AGENDA

What are we talking about?

- Why talk about **C2**?
- **C2**, a history
  - Infrastructure **Old** vs **New**
  - Evolution of transport traffic
- **High friction** environments
  - We must go deeper
  - Taking advantage of **trusted cloud**
  - **Repurposing** corporate **services**
- Conclusions
  - Transport **middleware**
  - Service SOP's

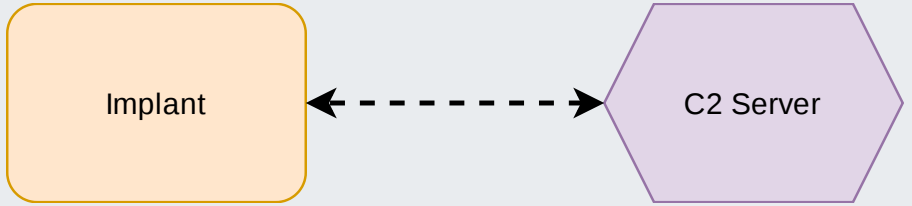
Why talk about C2?

- **Transport tradecraft** is pretty mature these days
  - Usually not a problem but **cost for failure is high**
  - Transports happen **after an attack was successful**
  - Transports should just work they are a **basic requirement**
- What **problems** can transports have?
  - Bad **endpoint tradecraft** (mostly a payload problem)
  - Good **security stack** and **security team**
    - EDR is getting better year-on-year
  - **Unknow** network **egress policy**

C2, a history

Old infrastructure

- Ancient history
  - C2 transports circa 2012
  - Bad endpoint and corporate security
- Not good today!
  - Users cannot reassure themselves by looking at the URL
  - Threat Intelligence (TI) teams signature C2 server responses
  - Abuse reports result in loss of infrastructure
  - C2 servers have exploitable bugs



Cobalt Strike 4.7.2 is now available. This is an out of band update to fix a remote code execution vulnerability that is rooted in Java Swing but which can be exploited in Cobalt Strike.

Remote Code Execution Vulnerability

I'd like to start by giving credit to Rio Sherri (0x09AL) and Ruben Boonen (FuzzySec) from the X-Force Red Adversary Simulation Team for their work in not only researching this vulnerability, but also sharing their findings with me and my team and helping us to mitigate it. They plan on publishing detailed information about this on their blog later today (if their blog post isn't live right now, check back later).

New infrastructure

- **Not new** but **evolving needs** over many years
- **Modern C2 infrastructure** has many components
  - Complex ⇨ needs **automation**
  - Can be updated dynamically
  - **Secure access** using VPN
  - Internal **service integration** (e.g., logging)
  - **Databases** (e.g. neo4j)
  - **Complex software** (e.g. ELK Stack)
  - Requirements for **operational success**
  - **OpSec**
  - Regulatory requirements**?**

Modern design

- More or less
  - 3+ years
  - This is simplified
  - Many moving parts
  - Fully automated, deployed every time
  - Hacking not just hands-on-keyboard

#DevOps 😂

Carrying Messages

- Transports are well understood
  - Send it over the **web**
  - **Front** the traffic
  - Use a **CDN**
- Mostly works fine
  - **Some considerations** for operational effectiveness
  - Big **vendors** may **hate you**
  - CDN's probably don't care

Transport Considerations

- Traditional **domains**
  - Ageing & reputation farming
  - Categorization services (**MacAfee**, **Cisco Talos**, **FortiGuard**, **Forcepoint**, **Palo Alto**, **Bluecoat**)
  - You likely need an **automated ageing solution** but be **careful** not to burn your infrastructure
- Domain **fronting**
  - There was a golden age for domain fronting
  - **Free reputation**
  - Careful, some vendors hate you (e.g. **Azure**)
  - TLS interception ⇨ Compare **SNI** and **HOST headers**
    - Prevalence of interception depends on **Geo** and **Industry**
- Just **CDN's**
  - Just use a CND ¯_(ツ)_/¯
  - Still **free reputation**, hard or impossible to block based on use
  - Careful, the domains **may not look as natural**
- Work on **your tradecraft**
  - You should have **a backup** and **a backup for your backup**
  - You should craft a **beautiful narrative** on **Endpoint**
    - **Does the traffic look normal** for the process
    - **Does the parent** of your process **look normal**
    - Are the **requests compliant** and does the traffic **volume seem ok**
    - **+** All the **implant in-memory hacking** (cryptography, deception, evasion)

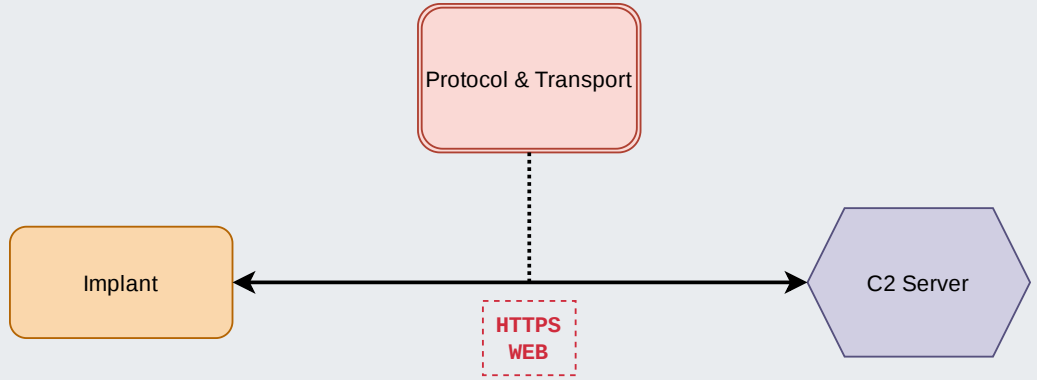
High-Friction Environments

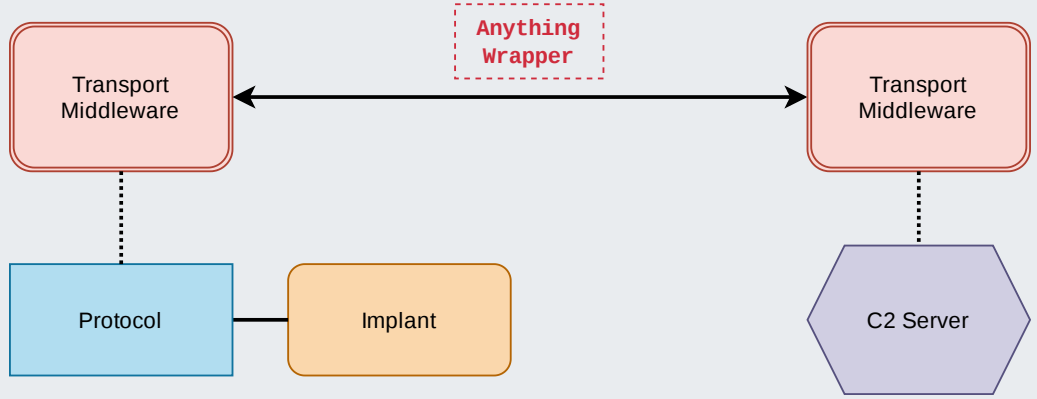
We must go deeper

- This seems pretty fine, **what is the problem**
  - Our team tests some of the **most mature environments**
  - Sometimes **egress** policies are **very strict**
  - We encounter combinations of **very expensive endpoint stacks** and **very good security teams**
- **How** can we make our transports look **even better**?
  - We can design our transports to **use real** business **cloud services**
  - We can **repurpose** authentication material from **endpoint services**

Level-setting transports

- Components of a **transport**
  - **Traditionally** we **couple** the implant **protocol** and the **transport** together
  - Not flexible, but functional for traditional C2 (**dumb HTTPS traffic**)
  - With **API based middleware**, operators can **create transports for anything** and **everything**



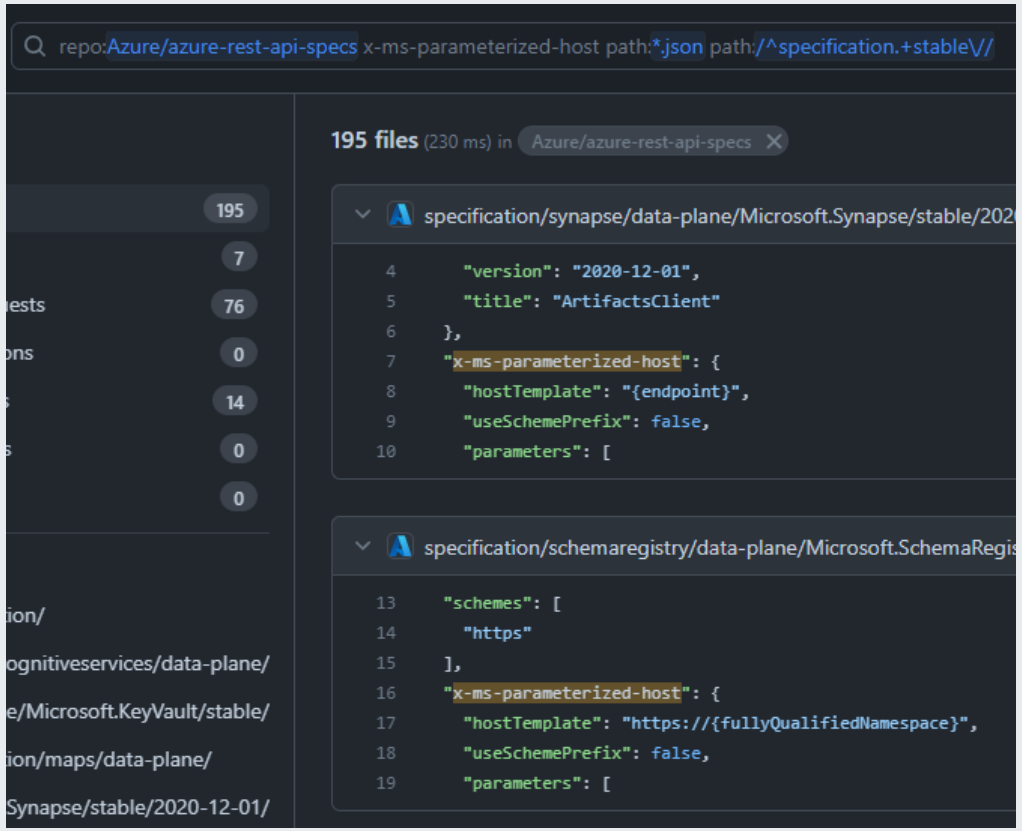


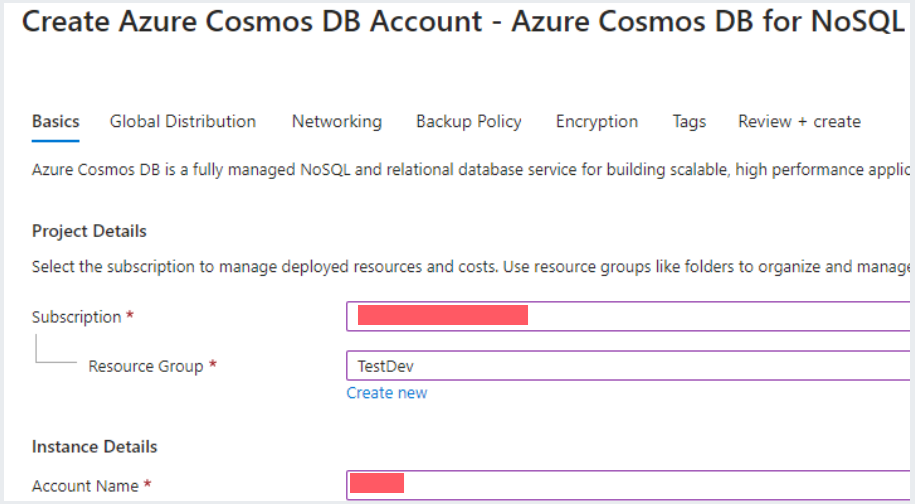
Trusted Cloud

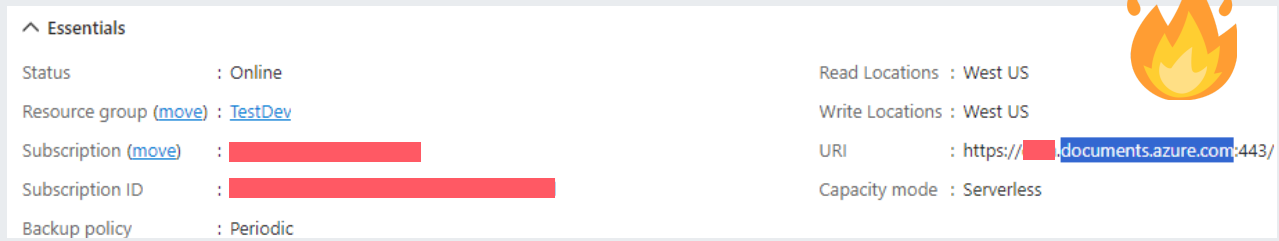
- Clouds, **not just rain** and **compute**
  - Cloud providers want to **offer you services**, they want to keep you on their platform
  - When you get a service, **you probably need a public endpoint for that** right?
- **Trust** in the clouds
  - Big cloud platforms are **used everywhere** in all businesses
  - Your target trusts their cloud 
  - Target usual suspects **Azure** & **AWS**

Azure methodology

- As a Microsoft fanboy **I love Azure**
- Finding services with public endpoints
  - Look at the **RESTful API specifications**
  - You are a developer now, **congratz**
- **Not all useful** (/ practical), but many are
  - **Verify** assumptions **in the portal**





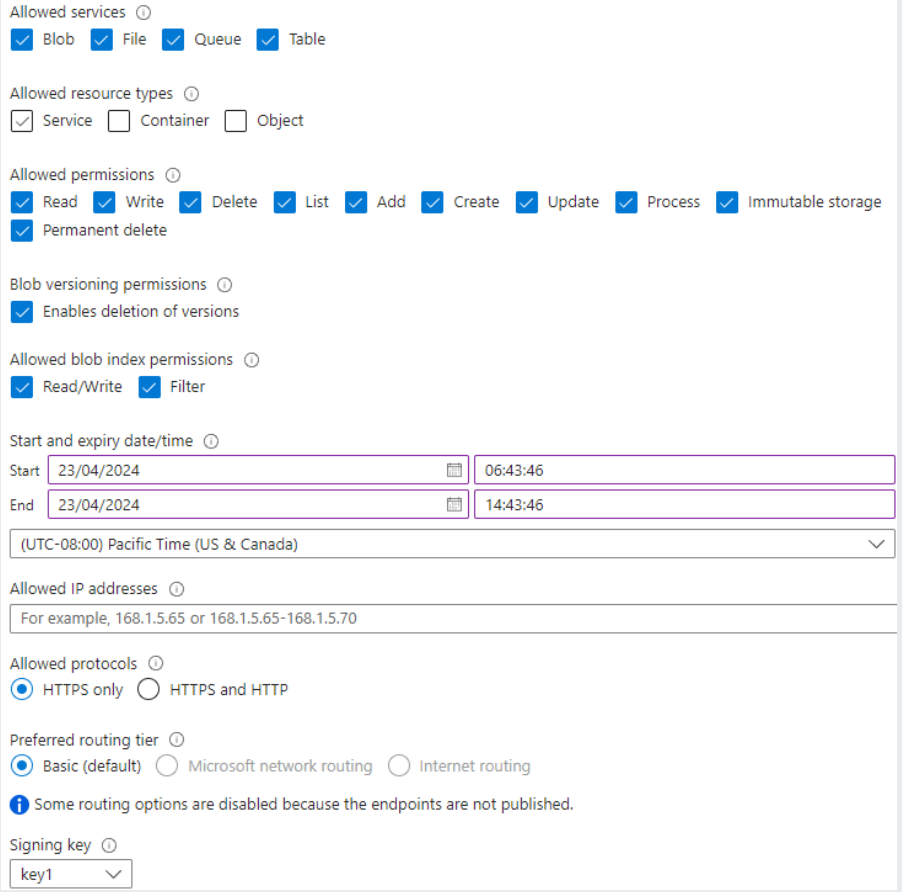


Now you just need to **write a CosmosDB transport**

Demo: Storage Accounts

- What does the **middleware** do?
  - **Implant middleware translates** the logic of the **C2 protocol** to Azure Storage Account tasking
    - For example, **implant registration** is translated to creating a **new GUID based container**
  - **Adds asymmetric key exchange** and session based cryptography **to the existing protocol**
  - **Server-side middleware unwraps everything** and turns it back into the original protocol
    - The **C2 server is not aware** that it is speaking a new protocol

- Services often have **special features** we can use!



What does it mean?

- **Very hard problem** for defenders
  - Company **relies** on a **specific cloud service**
  - Company **can't block cloud service**
  - Company security **trusts cloud service**
  - **Implant traffic** looks **exactly like real service traffic**
  - **Heuristic based monitoring**, why is this host doing something it has never done before?
  - **Blocking subdomains** for cloud services

- What should attackers do?
  - Make sure **service endpoint names** look good
  - Understand how the service is used in normal situations
  - Mask implant using **appropriate host/process context**
  - In-memory, payload hardening
  - **Backup transports** ok, we are not playing games

Repurpose & Reuse

- Endpoints are usually already operating many services
  - People want to **talk talk talk** (Slack / Teams)
  - Machines are domain-join, **Entra ID and hybrid environments**
- If the **attacker** owns the endpoint, they **own everything on the endpoint**
  - Use **DPAPI** to extract **Entra ID authentication JWT's** for dozens of services
  - **Electron** still **just SQLite on disk**, authenticate as the user to their applications

User Impersonation

Now you just need to write a bunch of **middleware transports**!

```
[+] Found DPAPI encrypted state key..
[>] Decrypted AES Key: 11E9CE75BF8FCD0EBAE3AFC269E128F3A49D2BD36319BAC5F8EC4BA9732292

[+] Parsing cookie store..
[+] Requesting API tokens..

[*] Workspace : [redacted]slack.com
    Token     : xoxc-4512[redacted]

[*] Workspace : [redacted]slack.com
    Token     : xoxc-6321[redacted]

[*] Workspace : [redacted]slack.com
    Token     : xoxc-3643[redacted]


[*] Workspace : redteamcabal.slack.com
    Token     : xoxc-1478[redacted]


[*] Workspace : fuzzyapt.slack.com
    Token     : xoxc-2466[redacted]


[*] Workspace : pssec.slack.com
    Token     : xoxc-8756[redacted]
```

¯_(ツ)_/¯

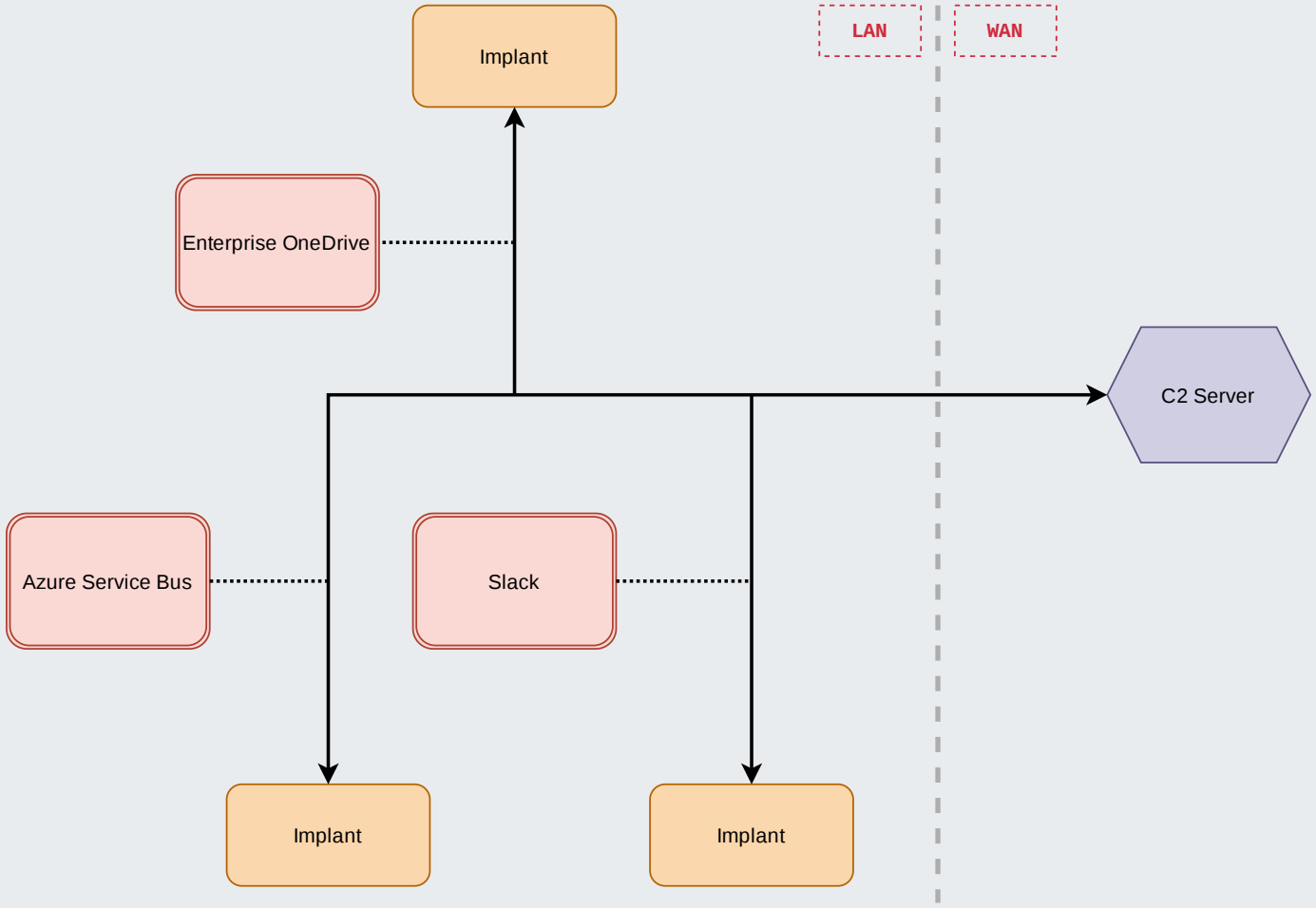
- Many **many** services
  - Microsoft Office
  - Microsoft Graph
  - Outlook O365
  - SharePoint
  - OneDrive
  - ...

```
[?] Processing file --> C:\Users\8J6670897\AppData\Local\Microsoft\TokenBroke
0.tbres
[+] Found JWT token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6IjlHbW55RlBraGMzaE91UjIybXZTdmduT
mZi1hYWQyMjkyYWIwMWMiLCJpc3MiOiJodHRwczovL2xvZ2luLm1pY3Jvc29mdG9ubGluZS5jb20v
5L3YyLjAiLCJpYXQiOjE2OTk4MDU3MTAsIm5iZiI6MTY5OTgwNTcxMCwiZXhwIjoxNjk5ODA5NjE
wbGlhbmNlX3VybCI6Imh0dHBzOi8vcG9ydGFsLm1hbmFnZS5taWNyb3NvZnQuY29tLz9wb3J0YWx
ybCI6Imh0dHBzOi8vZW5yb2xsbWVudC5tYW5hZ2UubWljcm9zb2Z0LmNvbS9lbnJvbGxtZW50c2V
lX3VybCI6Imh0dHBzOi8vcG9ydGFsLm1hbmFnZS5taWNyb3NvZnQuY29tL1Rlcm1zb2ZVc2UuYXN
jYmNmM2MtZmU2YS00NDY2LWFjYzctNTBhMmU5NGI3ZjQ3IiwicHJlZmVycmVkX3VzZXJuYW1lIjo
3MjVBNEUiLCJyaCI6IjAuQVJnQVYzRDJfTWxRMUVxWThfX0taSzNaNmRZT1dkT3pVZ0pCcnYtcTB
QNkVKRklSLUxqelR1S3NJeEFnUF9LZWFhYXMiLCJ0aWQiOiJmY2Y2NzA1Ny01MGM5LTRhZDQtOTh
jb20iLCJ2ZXIiOiIyLjAifQ.lu3R299BslW98xMBLisuNoarj19kkGmS6OHtotQPNGVqbR8b3t-PX
As8L9VJ5ld2kpq4AAbQFGMbN2V_HlhYmrkA0JU8q9cPnO19q04cnZYDeZ44873_jbj_dzwzUUi1Uz
u6bNFBRNCOl23DSvzxPwZSRqg119VS4oQNkFxiaLkHMPs26y-FzQW5zwGctIYlPxUkzvm3BZuyZZI
[+] JWT payload:
{"aud":"d3590ed6-52b3-4102-aeff-aad2292ab01c","iss":"https://login.microsoft
v2.0","iat":1699805710,"nbf":1699805710,"exp":1699809610,"email":"rboonen@ibm
ge.microsoft.com/?portalAction=Compliance","mdm_enrollment_url":"https://enro
scovery.svc","mdm_terms_of_use_url":"https://portal.manage.microsoft.com/Term
cf3c-fe6a-4466-acc7-50a2e94b7f47","preferred_username":"rboonen@ibm.com","pui
__kZK3Z6dYOWdOzUgJBrv-q0ikqsBwYAMk.","sub":"77CuplXKI8y__ee7P6EJFIR-LjzTuKsIx
64add9e9","upn":"rboonen@ibm.com","ver":"2.0"}
[+] Found JWT token:
eyJ0eXAiOiJKV1QiLCJub25jZSI6IjRacW10Ykp1cV9vMWoyMHNuaFgtZlpHSHUwM01lSDdRcHdmM
RlBraGMzaE91UjIybXZTdmduTG83WSIsImtpZCI6IjlHbW55RlBraGMzaE91UjIybXZTdmduTG83V
jb20vIiwiaXNzIjoiaHR0cHM6Ly9zdHMud2luZG93cy5uZXQvZmNmNjcwNTctNTBjOS00YWQ0LTk4
ibmJmIjoxNjk5ODA1NzEwLCJleHAiOjE2OTk4OTI0MTAsImFjY3QiOjAsImFjciI6IjEiLCJhaW8i
KY0VyIWtOT2xLk053Y1BzUlNTOEdaNWRxb1ZXU3R3MFZ2O3q2dWhGeUJMRkI4dzNLTS9Od0lzMWVM
```

Defenders want to know

AM I A JOKE TO YOU??

- **Very hard problem** for defenders
  - What is your threat model for an in-memory implant in the Slack process talking to Slack?
  - How do you know if the user is saving a document to SharePoint via OneDrive or not?
  - How can you tell if a service talking to your corporate message queue is on the level?
- Something, something .. **heuristics**



Conclusions

Research and Development

- C2 middleware
  - Obviously this requires that you can decouple the C2 protocol from the transport
  - Typically C2 products are not designed for this, they will support HTTP(s) and DNS
    - Some commercial products do, this is not an ad
  - Jury-rig your existing product (hard) write your own C2 (also hard)
- Custom transports
  - Building custom transports takes time but it's something you can work on gradually
    - Start with high-profile services that give the most returns
  - The research is interesting actually and you can find logic bugs in the protocols (like Slack👀)

Questions?

IBM®