How to build a C2 framework from scratch

Jakob Friedl

BSidesVienna0x7E9

> whoami

- **Jakob Friedl**
- Penetration Testing @ MM Group
- ~3 years in offensive security
  - Hacking Lab Enjoyer
  - Certification Enjoyer 

  - Malware Development Enjoyer
- 6+ months, 200+ commits of “Conquest” C2 development

Chapter 1

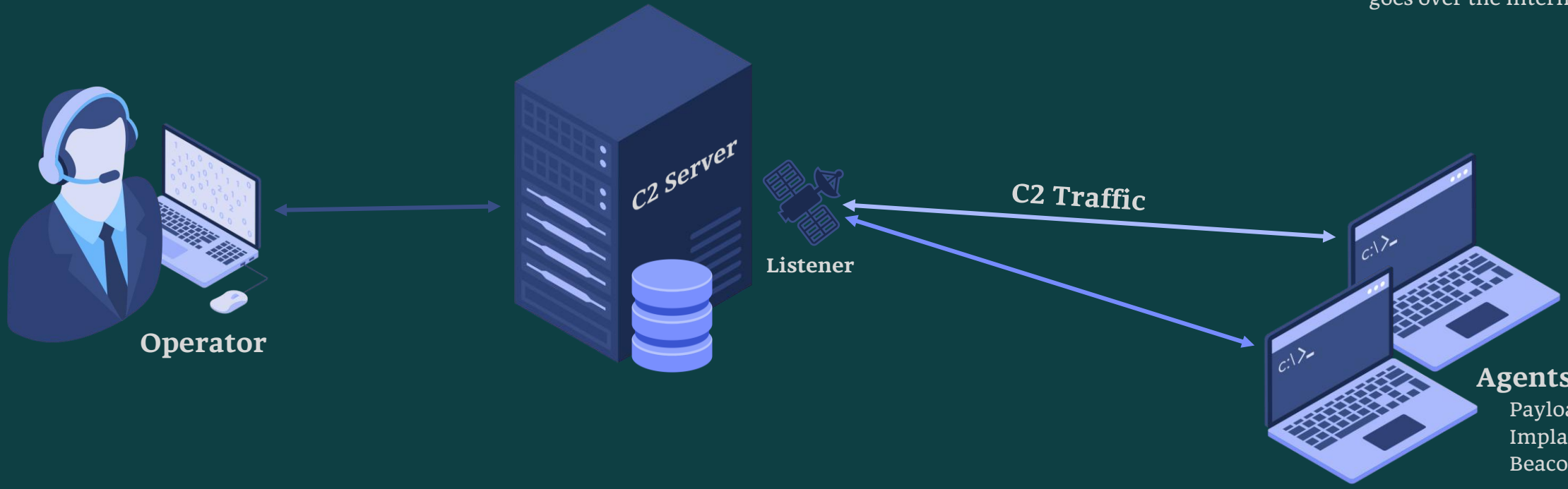
Command & Control 101

MITRE ATT&CK - TA0011

Command and Control consists of techniques that adversaries may use to communicate with systems under their control within a victim network.
Adversaries commonly attempt to mimic normal, expected traffic to avoid detection.

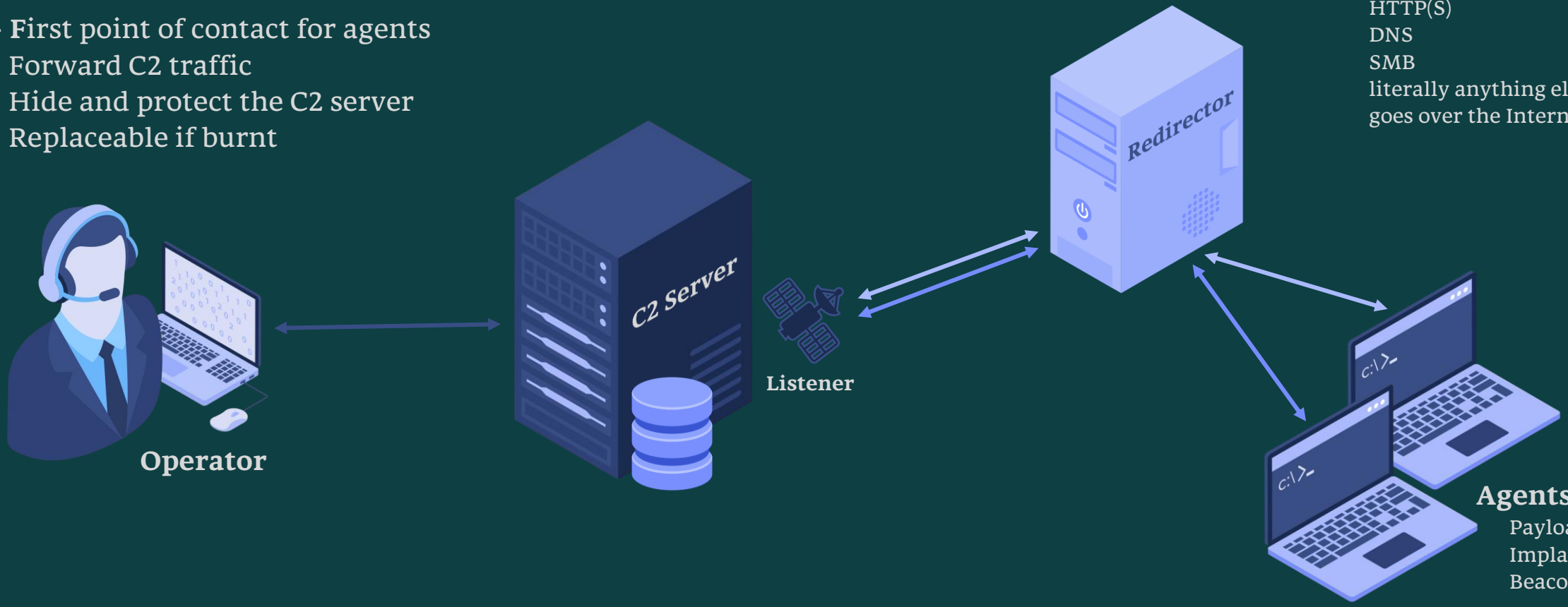
C2 Protocols:

HTTP(S)
DNS
SMB
literally anything else that goes over the Internet



C2 Redirectors

- First point of contact for agents
- Forward C2 traffic
- Hide and protect the C2 server
- Replaceable if burnt

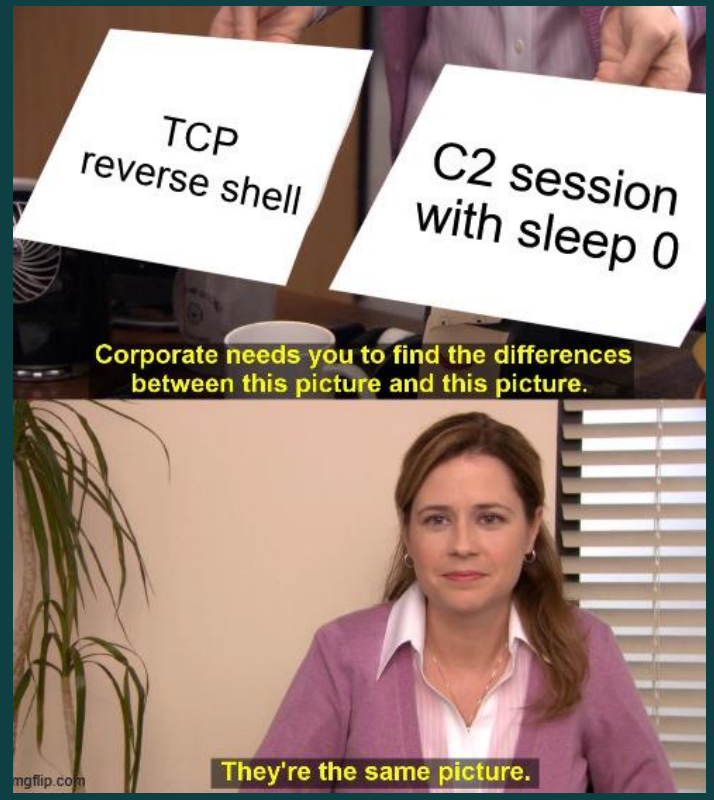


C2 Protocols:

HTTP(S)
DNS
SMB
literally anything else that goes over the Internet

Why C2?

- Task queueing
- Persistence
- Post-exploitation
- Multiple sessions & users



C2 Frameworks

More at:
https://howto.thec2matrix.com/

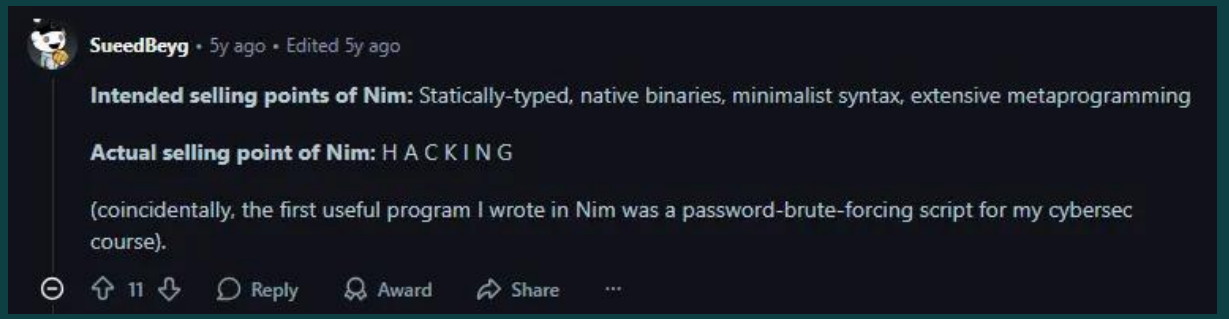
COBALT STRIKE
ADVANCED THREAT TACTICS FOR PENETRATION TESTERS

MYTHIC

Chapter 1.5
Nim 101 

Why Nim?

- Enjoyable development experience
- Static type system
- Low-level performance
- Cross-compilation
- Easy access to Windows API using *winim* library



Why Nim?

- Enjoyable deve
- Static type syst
- Low-level perfo
- Cross-compilat
- Easy access to V

11.8k results (2 s) Sort by: Most stars Save

nim-lang/Nim — Unstar Sponsor
Nim is a statically typed compiled systems programming language. It combines successful concepts from mature languages like Python, Ada a...
language nim compiler metaprogramming efficient
Nim · 17.5k · Updated 2 hours ago

zedeus/nitter — Star Sponsor
Alternative Twitter front-end
privacy twitter nim self-hosted x
Nim · 11.8k · Updated on 12 Oct

WyattBlue/auto-editor — Star
Auto-Editor: Efficient media analysis and rendering
audio video nim audio-editing video-processing
Nim · 3.7k · Updated 8 days ago

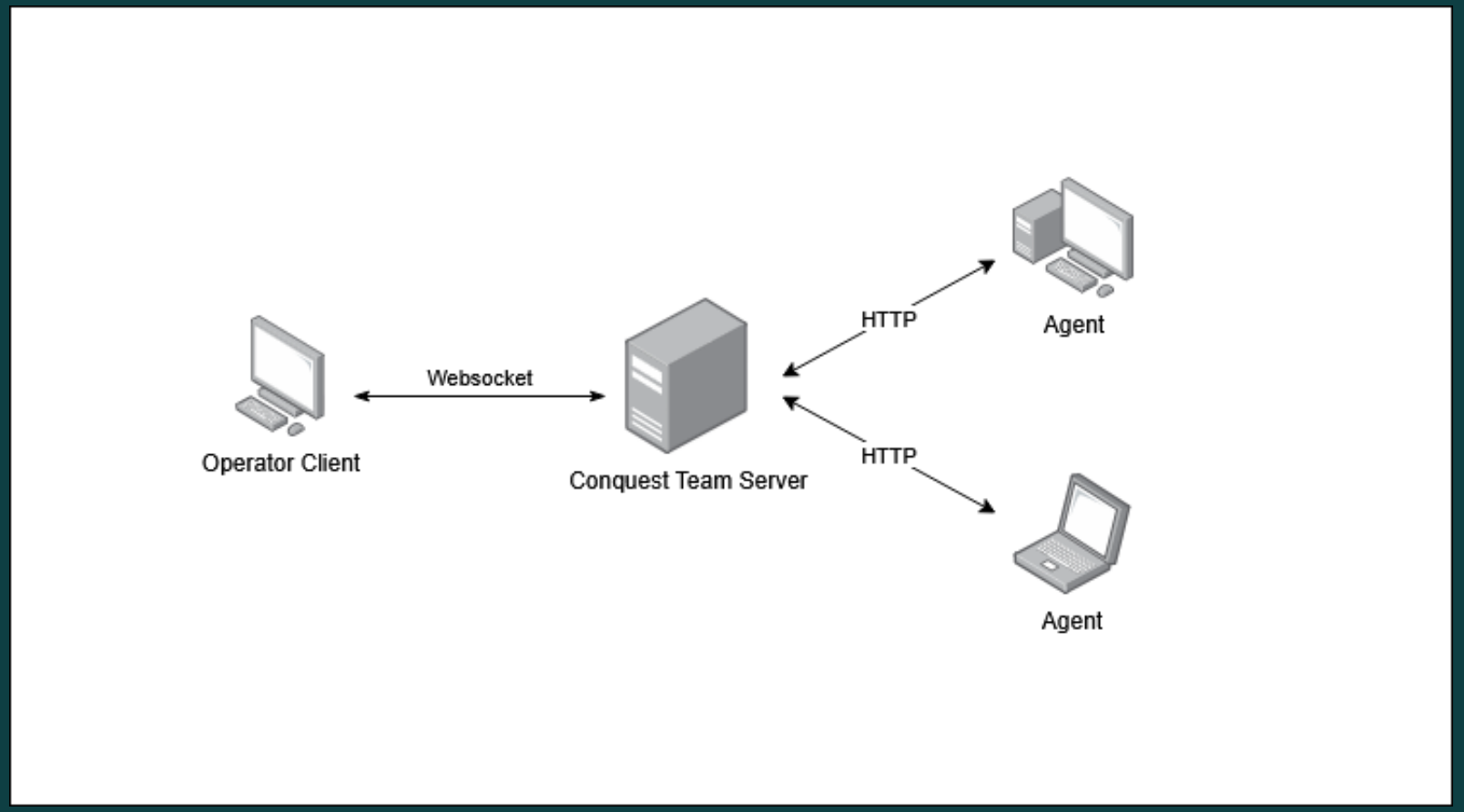
byt3bl33d3r/OffensiveNim — Unstar Sponsor
My experiments in weaponizing Nim (https://nim-lang.org/)
Nim · 3k · Updated on 13 May 2024

oakes/vim_cubed — Star
Vim rendered on a cube for no reason
Nim · 2.7k · Updated on 15 Apr 2022

chvancooten/maldev-for-dummies — Star Sponsor
A workshop about Malware Development
hacktoberfest
Nim · 1.7k · Updated on 2 Jun 2023

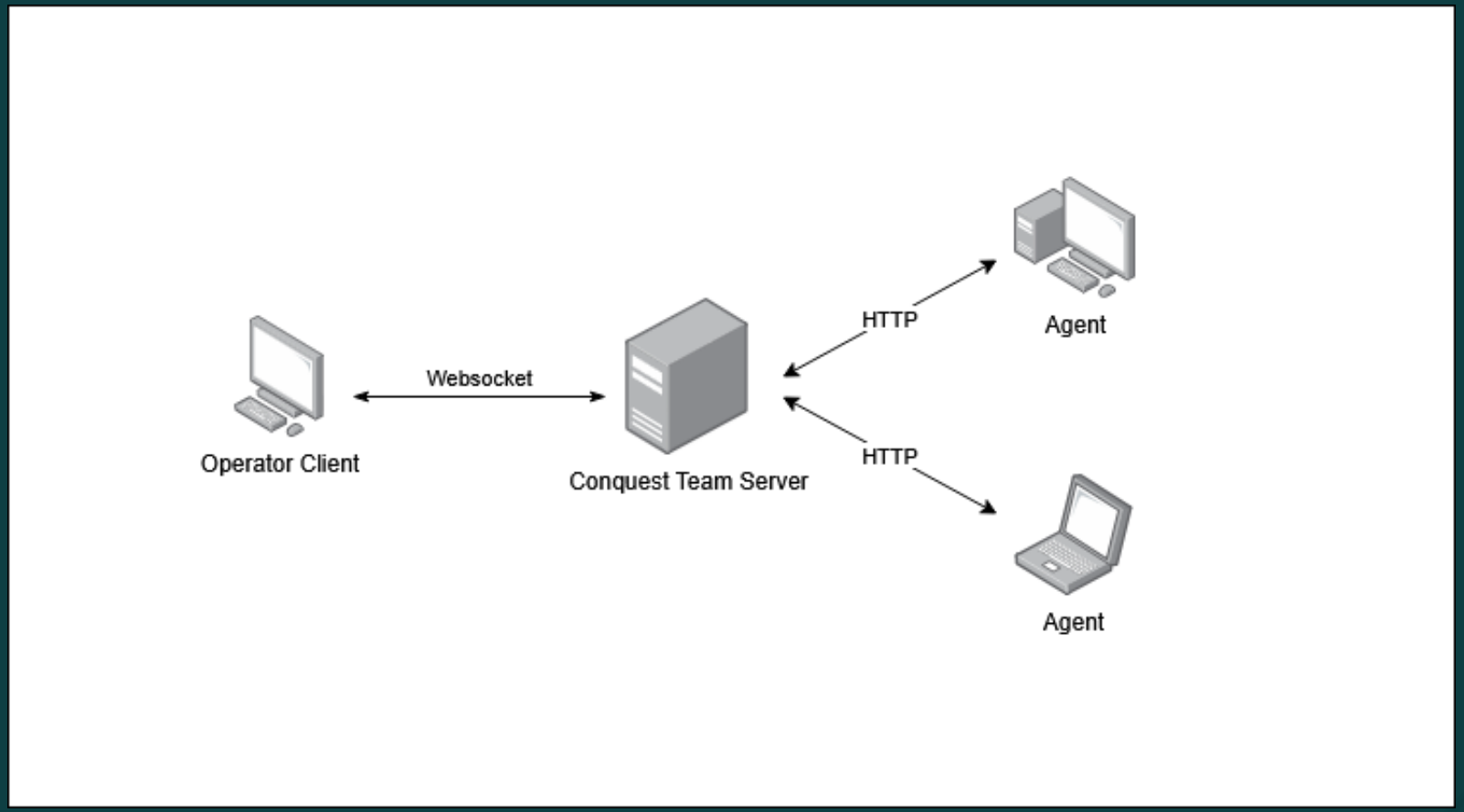
Chapter 2
Designing a C2 Framework

Architecture

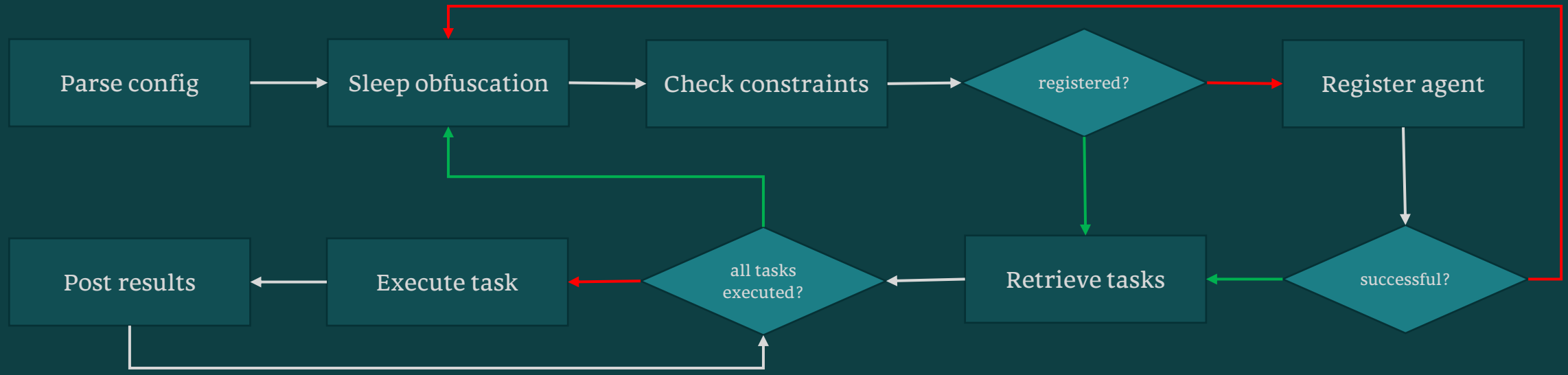


Architecture

- Client tells Server what to do
  - Start Listener
  - Stop Listener
  - Add new task to queue
  - …
- Server tells Client to update UI
  - New Agent connected
  - Task output received
  - …
- Agents use HTTP endpoints to GET tasks and POST results

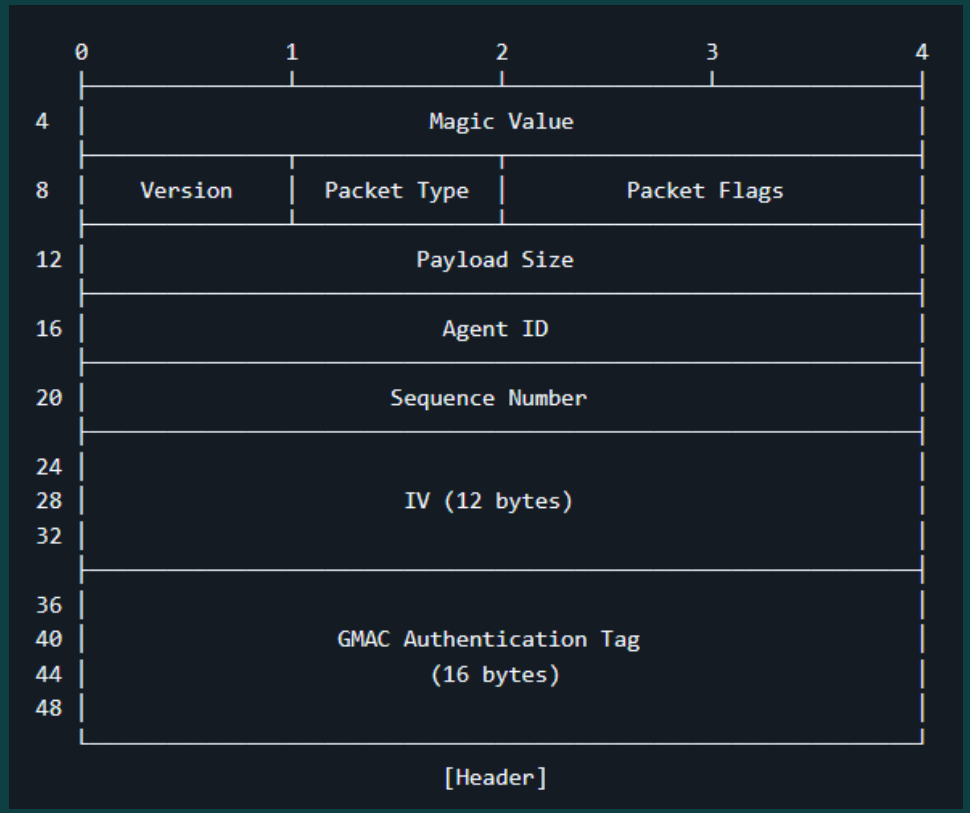


Agent Routine



C2 Communication

- 4 packet types
  - **Registration**
  - **Heartbeat**
  - **Task**
  - **Result**
- 48-byte header for packet & agent identification

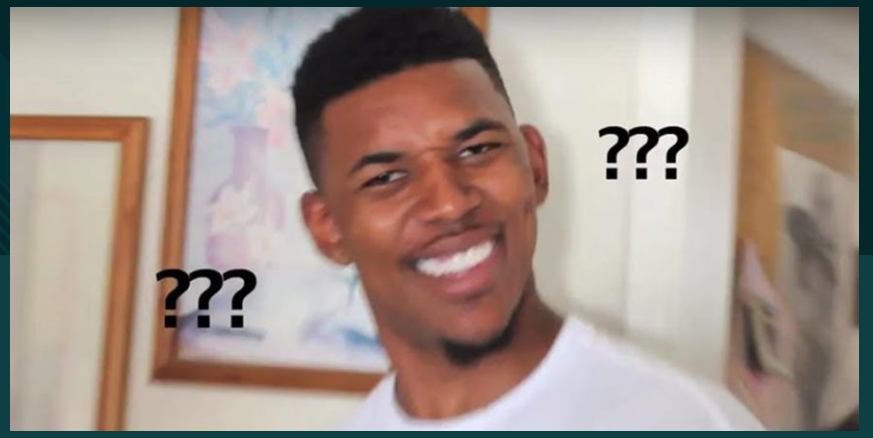


Protecting C2 Traffic

- C2 traffic contains highly-sensitive data
  - System information
  - Command output
  - Files
  - Credential material

- **Packet Encryption 🔒**
  - AES256-GCM

Key Exchange



- ECDH based on X25519
- Encryption key is derived from the shared secret

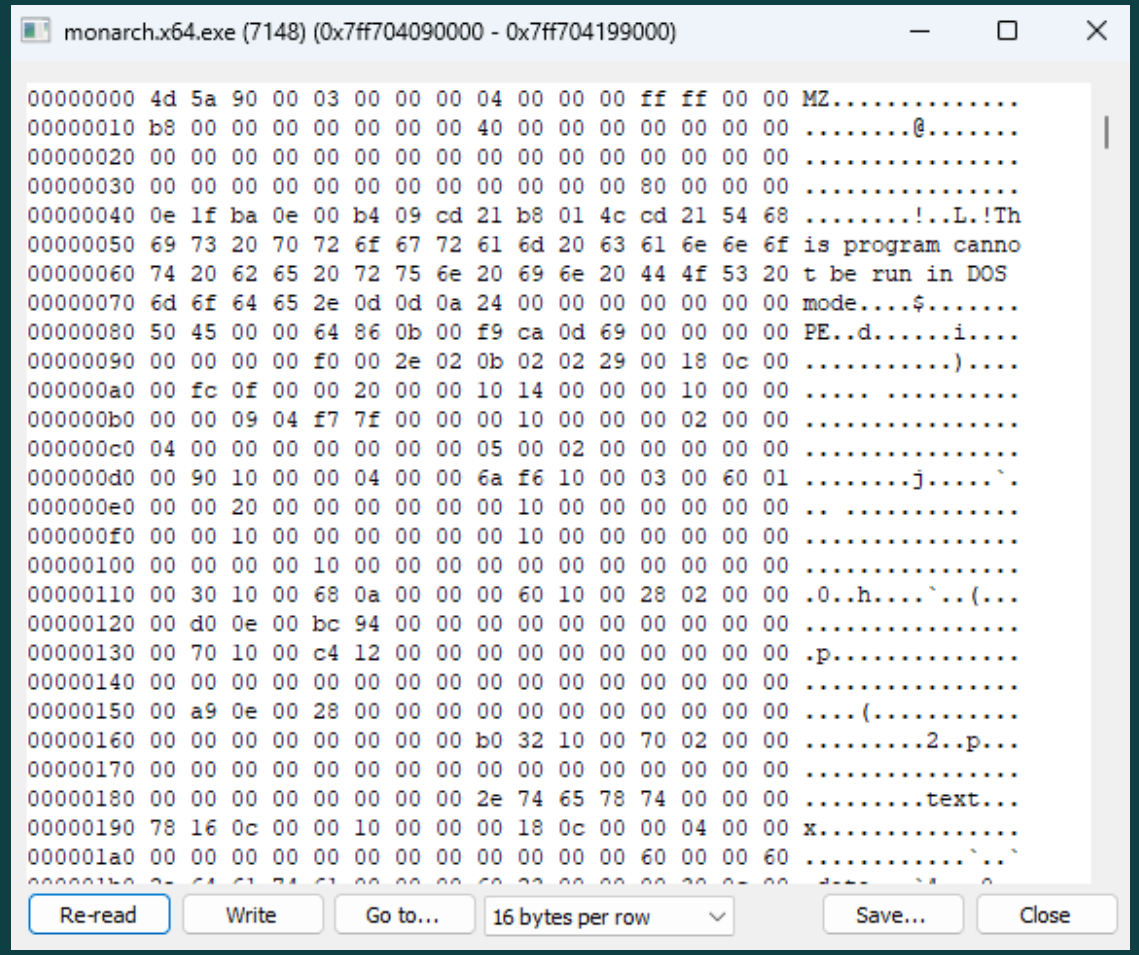
$$\text{derive}(\text{Server}_{\text{private}} + \text{Agent}_{\text{public}}) = \text{derive}(\text{Server}_{\text{public}} + \text{Agent}_{\text{private}})$$

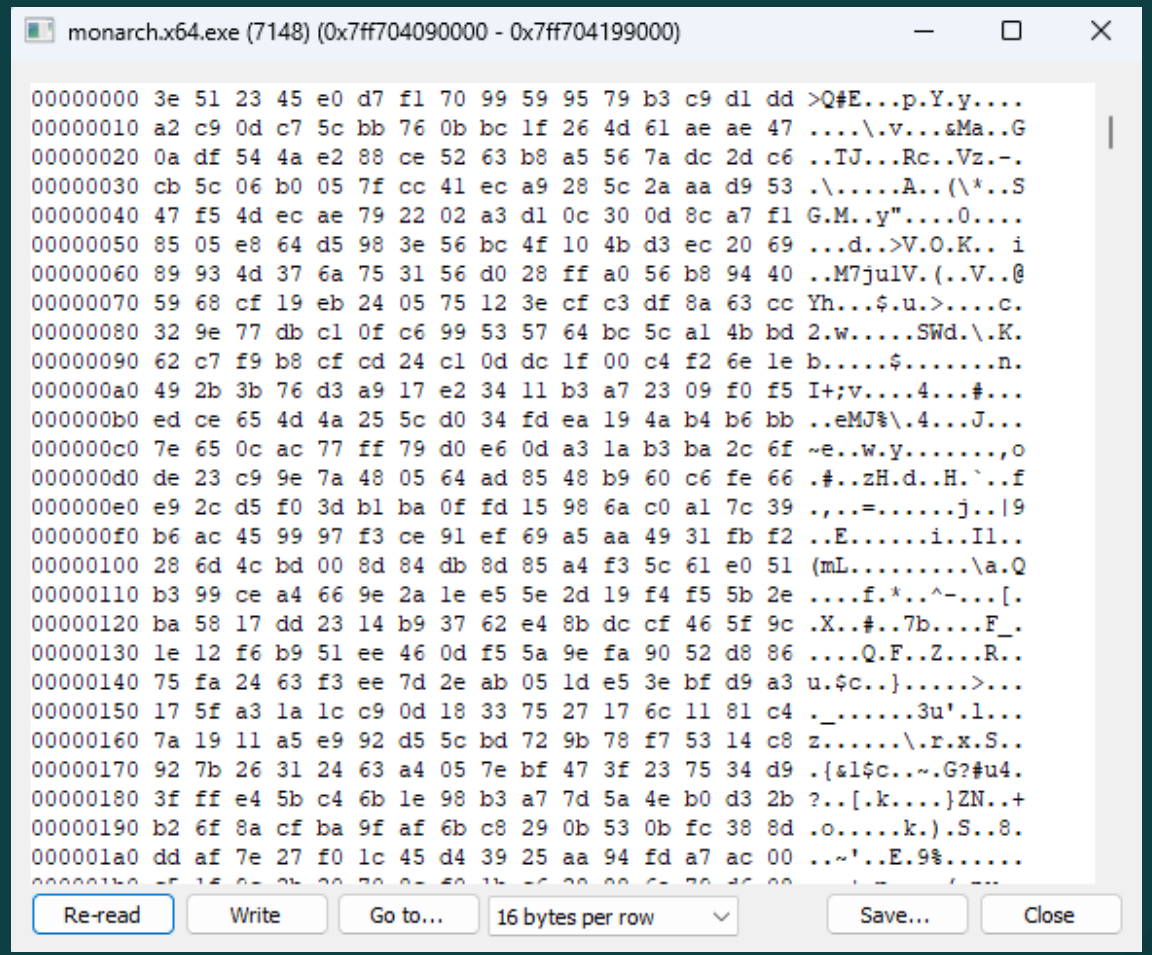
- Agent is generated with server's public key embedded
- Registration packet contains agent's public key
- Client uses Websocket events to initiate key exchange

Chapter 3
Agent Activities

Activity 1: Sleeping

Sleep Obfuscation





Activity 2: Requesting

- C2 Network traffic can be customized using C2 profiles
  - User-agents
  - HTTP endpoints
  - Headers
  - GET Parameters
  - …

```
# Defines URI endpoints for HTTP GET requests
# This has to be an array, even if it only has one member
endpoints = [
    "/get",
    "/api/v1.2/status.js"
]

# Defines arbitrary URI parameters that are added to the request
[http-get.agent.parameters]
id = "#####-#####"
lang = [
    "en-US",
    "de-AT"
]

# Defines arbitrary headers that are added by the agent when performing a HTTP GET request
[http-get.agent.headers]
Host = [
    "wikipedia.org",
    "google.com",
    "127.0.0.1"
]
Connection = "Keep-Alive"
Cache-Control = "no-cache"
```

Activity 2: Requesting

- C2 Network traffic can be customized using C2 profiles
  - User-agents
  - HTTP endpoints
  - Headers
  - GET Parameters
  - …

```
# Defines URI endpoints for HTTP GET requests
# This has to be an array, even if it only has one member
endpoints = [
    "/get",
    "/api/v1.2/status.js"
]

# Defines arbitrary URI parameters that are added to the request
[http-get.agent.parameters]
id = "#####-#####"
```

No.	Time	Source	Destination	Protocol	Length	Info
8	0.017898	172.29.176.1	172.29.177.43	HTTP	446	POST /api/v2/get.js?lang=en-US&page=16 HTTP/1.1 (text/plain)
10	0.189483	172.29.177.43	172.29.176.1	HTTP	91	HTTP/1.1 200
17	0.222307	172.29.176.1	172.29.177.43	HTTP	541	GET /api/v1.2/status.js?id=b3AKO-7fE90&lang=en-US HTTP/1.1
19	0.352693	172.29.177.43	172.29.176.1	HTTP	89	HTTP/1.1 200
26	5.557146	172.29.176.1	172.29.177.43	HTTP	524	GET /get?id=lAUQk-LSTob&lang=de-AT HTTP/1.1
28	5.688683	172.29.177.43	172.29.176.1	HTTP	89	HTTP/1.1 200
35	11.384841	172.29.176.1	172.29.177.43	HTTP	508	GET /get?id=zWgyL-eAVDN&lang=en-US HTTP/1.1
37	11.479281	172.29.177.43	172.29.176.1	HTTP	89	HTTP/1.1 200
44	17.036604	172.29.176.1	172.29.177.43	HTTP	551	GET /api/v1.2/status.js?id=VnQ4B-o5b5k&lang=de-AT HTTP/1.1
46	17.161741	172.29.177.43	172.29.176.1	HTTP	275	HTTP/1.1 200
54	17.189807	172.29.176.1	172.29.177.43	HTTP	1526	PUT /api/v2/get.js?lang=de-AT&page=17 HTTP/1.1 (text/plain)
57	17.243284	172.29.177.43	172.29.176.1	HTTP	91	HTTP/1.1 200
64	21.884782	172.29.176.1	172.29.177.43	HTTP	539	GET /api/v1.2/status.js?id=vkXB2-btoZb&lang=de-AT HTTP/1.1
66	21.998057	172.29.177.43	172.29.176.1	HTTP	89	HTTP/1.1 200

Data transformation

- Customize how packet data is represented

1. Encoding
2. Append/prepend strings
3. Placement

```
[http-get.agent.heartbeat]
placement = { type = "header", name = "Authorization" }
encoding = { type = "base64", url-safe = true }
prefix = "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9."
suffix = ".########################################-####"
```

Data transformation

```
▸ Frame 73: 509 bytes on wire (4072 bits), 509 bytes captured (4072 bits) on interface \Device\NPF_{42402AF3-E76F-4A86-9807-322510CEA966}, id 0
▸ Ethernet II, Src: Microsoft_16:13:82 (00:15:5d:16:13:82), Dst: Microsoft_16:1b:44 (00:15:5d:16:1b:44)
▸ Internet Protocol Version 4, Src: 172.29.176.1, Dst: 172.29.177.43
▸ Transmission Control Protocol, Src Port: 53612, Dst Port: 8080, Seq: 1, Ack: 1, Len: 455
▾ Hypertext Transfer Protocol
  ▾ GET /get?id=7cyub-sHSoi&lang=de-AT HTTP/1.1\r\n
      Request Method: GET
    ▾ Request URI: /get?id=7cyub-sHSoi&lang=de-AT
        Request URI Path: /get
      ▸ Request URI Query: id=7cyub-sHSoi&lang=de-AT
      Request Version: HTTP/1.1
    connection: Keep-Alive\r\n
    authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.QzBOUQFkAQAkAAAAID5E6AAAAAAQbtdtlhTVRrVaa6bIypGtjDLo3iKTqljRydyajCMPceUS1XA_bkIQSVDmNv5
    host: 127.0.0.1\r\n
    cache-control: no-cache\r\n
    user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36\r\n
    \r\n
    [Response in frame: 75]
    [Full request URI: http://127.0.0.1/get?id=7cyub-sHSoi&lang=de-AT]
```

Data transformation

```
Frame 73: 509 bytes on wire (4072 bits), 509 bytes captured (4072 bits) on interface \Device\NPF_{42402AF3-E76F-4A86-9807-322510CEA966}, id 0
Ethernet II, Src: Microsoft_16:13:82 (00:15:5d:16:13:82), Dst: Microsoft_16:1b:44 (00:15:5d:16:1b:44)
```

Recipe

From Base64

Alphabet
A-Za-z0-9+/=

☑ Remove non-alphabet chars

Input

start: 112
end: 112
length: 0

```
QzBOUQFkAQAkAAAAID5E6AAAAAAQbtdtlhTVRrVaa6bIypGtjDLo3iKTqljRydyajCMPceUS1XA_bkIQSVDmNv
```

Output

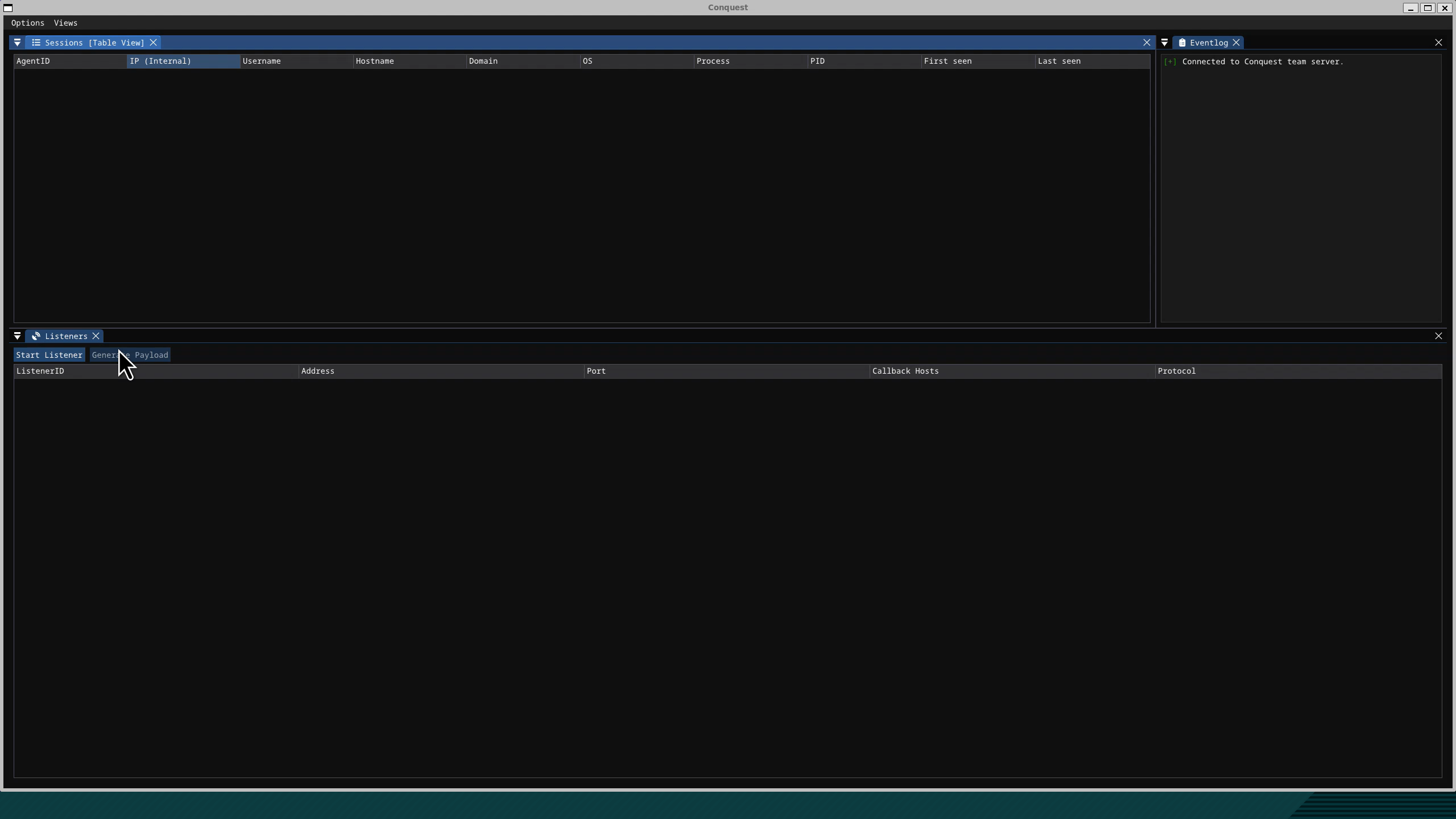
start: 84
end: 84
length: 0

```
C0NQ.d..$... >Dè.....n×m..ÕFµZk¦ÈÊ...2èÞ".ªXÑÉÜ..#.qå.Õp....T9.¿.ÚåÞTB¡¸.%.~.£.!
X.
```

```
[Response in frame: 75]
[Full request URI: http://127.0.0.1/get?id=7cyub-sH5oi&lang=de-AT]
```

Activity 3: Executing

Command execution	Screenshots	File upload & download
BOF execution	.NET execution	Token manipulation


Conquest
Options
Views
Sessions [Table View]
AgentID
IP (Internal)
Username
Hostname
Domain
OS
Process
PID
First seen
Last seen
Eventlog
[+] Connected to Conquest team server.
Listeners
Start Listener
Generate Payload
ListenerID
Address
Port
Callback Hosts
Protocol

Chapter 4
Should I build a C2 framework?

It depends…

Should I build a C2 framework?

YES, because …

- … you want to **learn how C2 frameworks work** in detail
- … you want **full control** over capabilities and features
- … you can spend time and energy on **designated malware/tool development**
- … you want to fill a **market or research gap**
- … you want to **challenge** yourself

NO, because…

- … it takes a lot of **effort**
- … you don’t want to **reinvent the wheel**
- … building a **custom agent/module** also gets the job done
- … **time is money**. You get paid for doing engagements, not writing a C2

What’s next for Conquest?

- *The more you know, the more you know that you don’t know*

- More listener & payload types
- Modular plugin system
- Stability - Usability - Functionality

https://github.com/jakobfriedl/conquest

Thank you!

jakobfriedl.github.io