black hat®
BRIEFINGS

AUGUST 6-7, 2025
MANDALAY BAY / LAS VEGAS

# Ghost Calls: Abusing Web Conferencing for Covert Command & Control

Adam Crosser

# Introduction

Adam Crosser

Praetorian

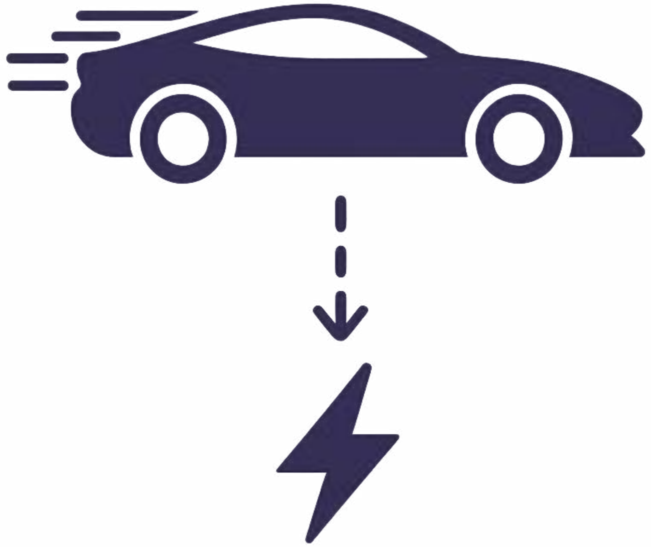
LinkedIn: https://www.linkedin.com/in/adam-crosser-366263265

X: https://x.com/UNC1739

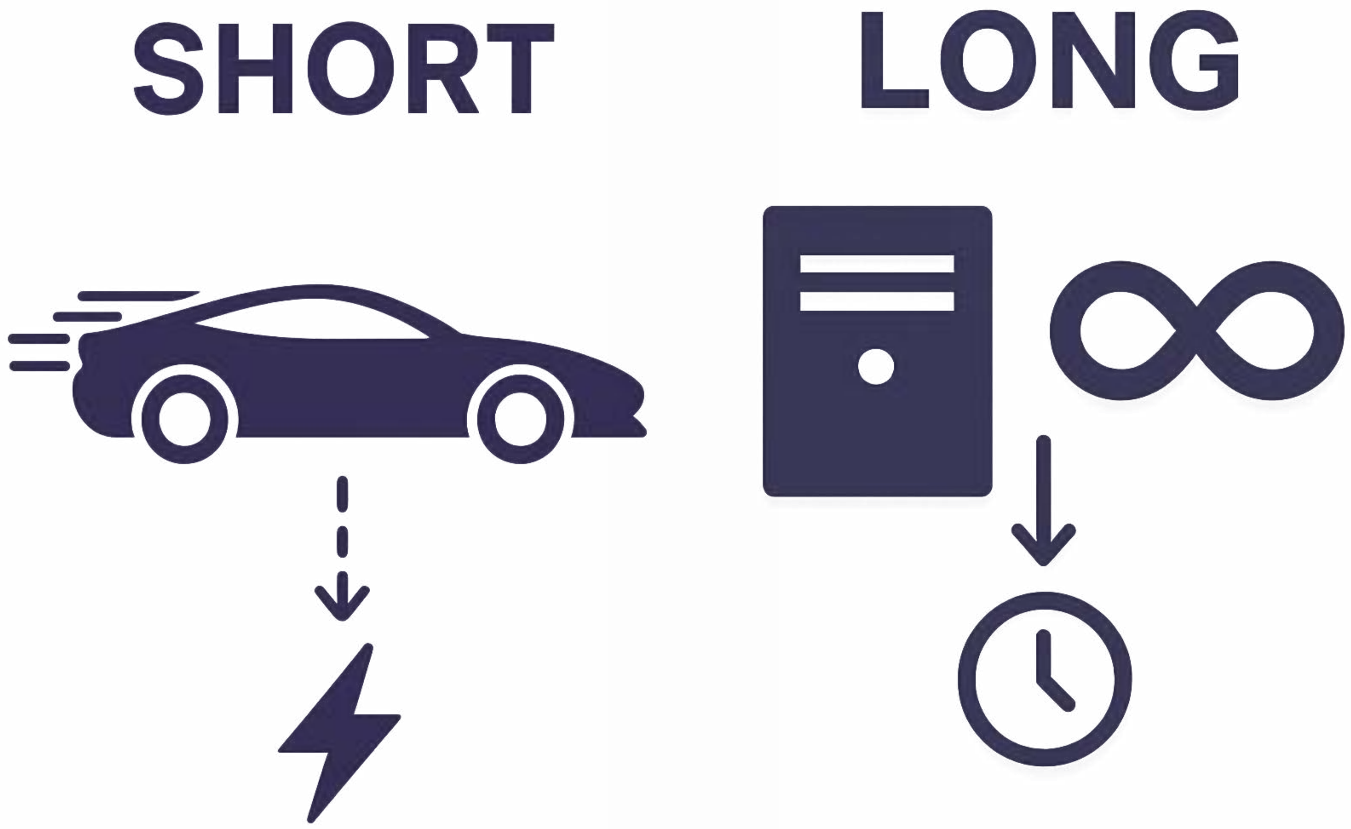
# Types of Command-and-Control Channels

# Types of Command-and-Control Channels

## SHORT



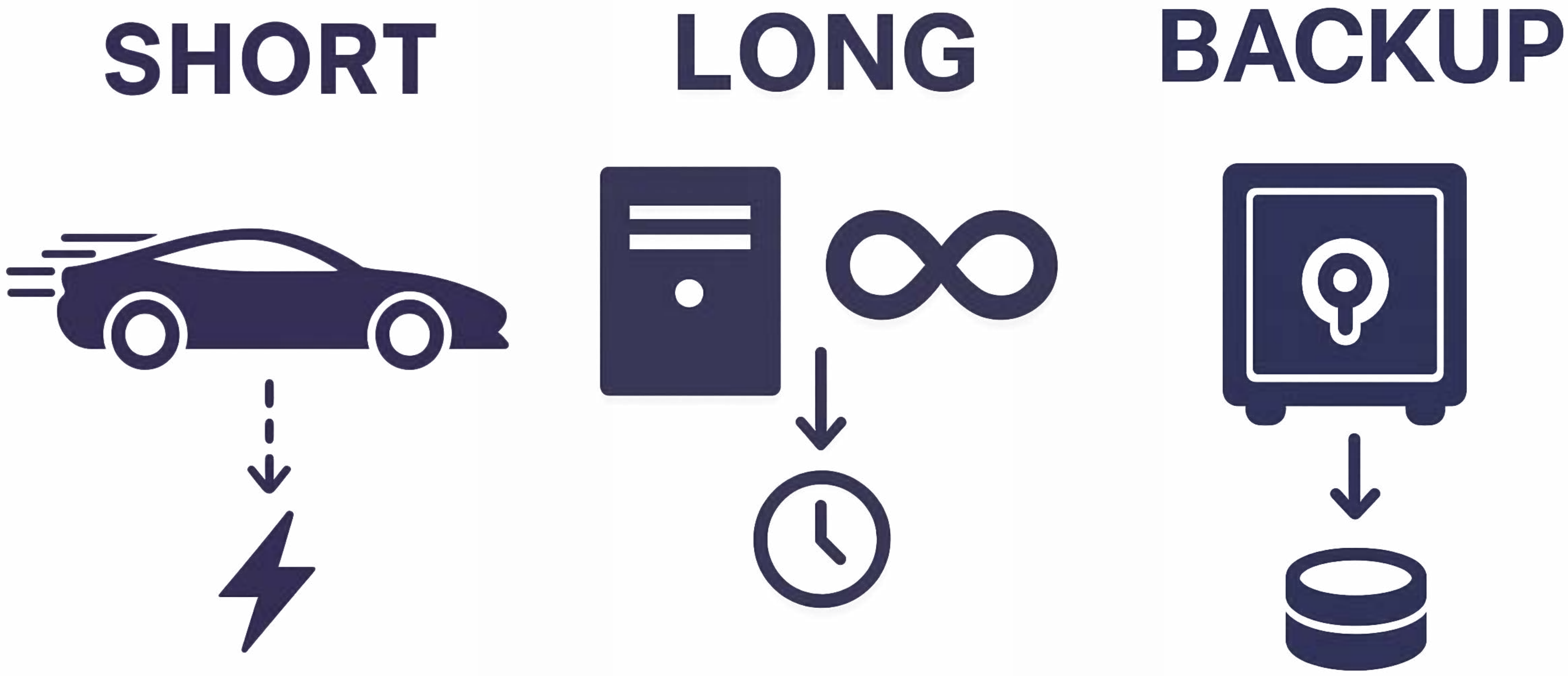
# Types of Command-and-Control Channels



SHORT

LONG

# Types of Command-and-Control Channels



SHORT

LONG

BACKUP

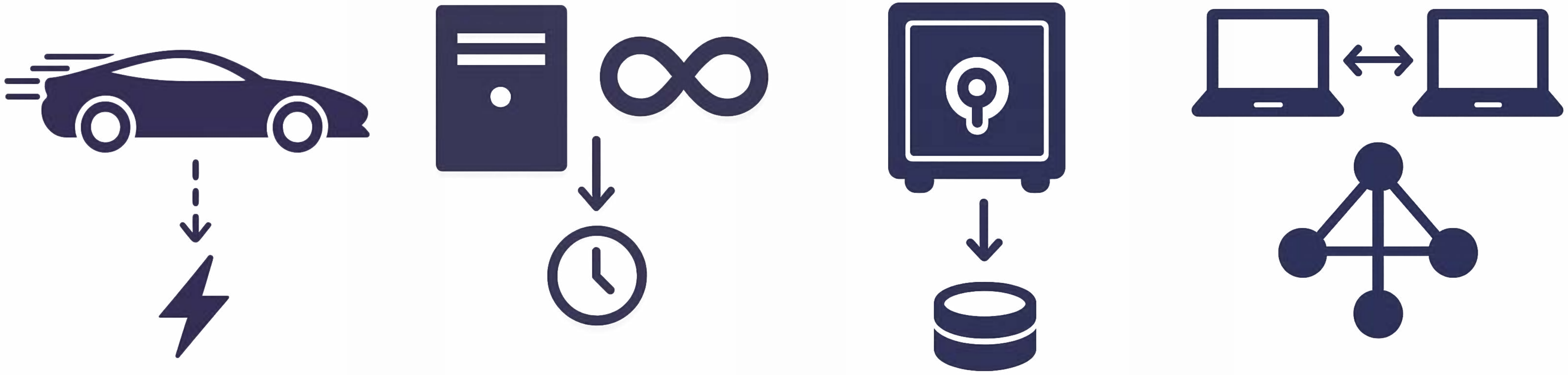
# Types of Command-and-Control Channels

SHORT

LONG

BACKUP

P2P



# Brainstorming Solutions

# Ideal Short-Term Command and Control

# Ideal Short-Term Command and Control

## LATENCY

# Ideal Short-Term Command and Control

LATENCY

THROUGHPUT

# Ideal Short-Term Command and Control

LATENCY

THROUGHPUT

REACH

# Ideal Short-Term Command and Control

**LATENCY**

**THROUGHPUT**

**REACH**

**TRUST**

# Selection Criteria

- Focused on services egressing from user devices
- Must be broadly used across enterprise roles
- Applicable to non-technical departments (e.g., HR, sales)
- Protocols favored by technical users were excluded
- Thought through common workflows and use-cases

HTTP

DNS

SMB

RDP

# DNS over HTTP (DoH)

- ✓ LATENCY
- ✗ THROUGHPUT
- ● REACH
- ✓ TRUST

# Cloud File Storage

- ✗ LATENCY
- ✓ THROUGHPUT
- ✓ REACH
- ● TRUST

# Attacker VM with Classified Domain

- ✓ LATENCY
- ✓ THROUGHPUT
- ✗ REACH
- ✗ TRUST

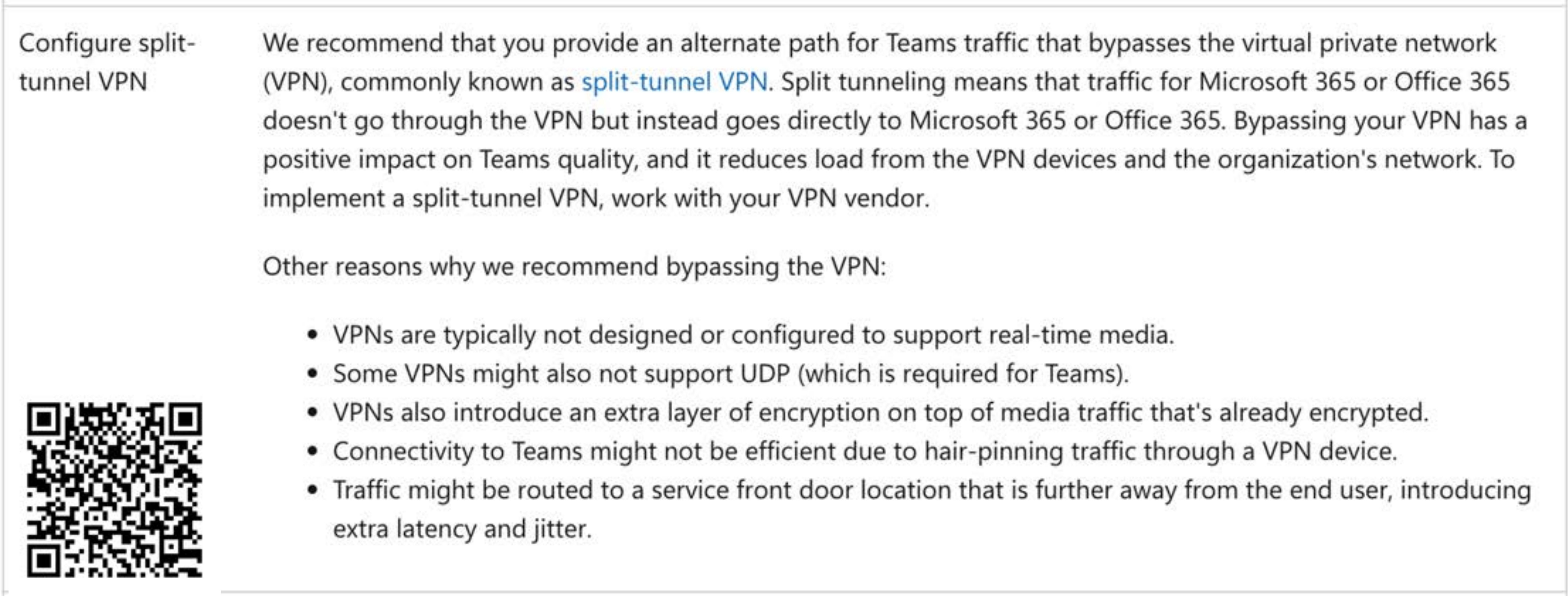
# Email and Messaging Applications

- ● LATENCY
- ✗ THROUGHPUT
- ✓ REACH
- ✓ TRUST

# Web Conferencing

- ✓ LATENCY
- ✓ THROUGHPUT
- ✓ REACH
- ✓ TRUST

# Microsoft Teams Split Tunneling Guidelines



Configure split-tunnel VPN

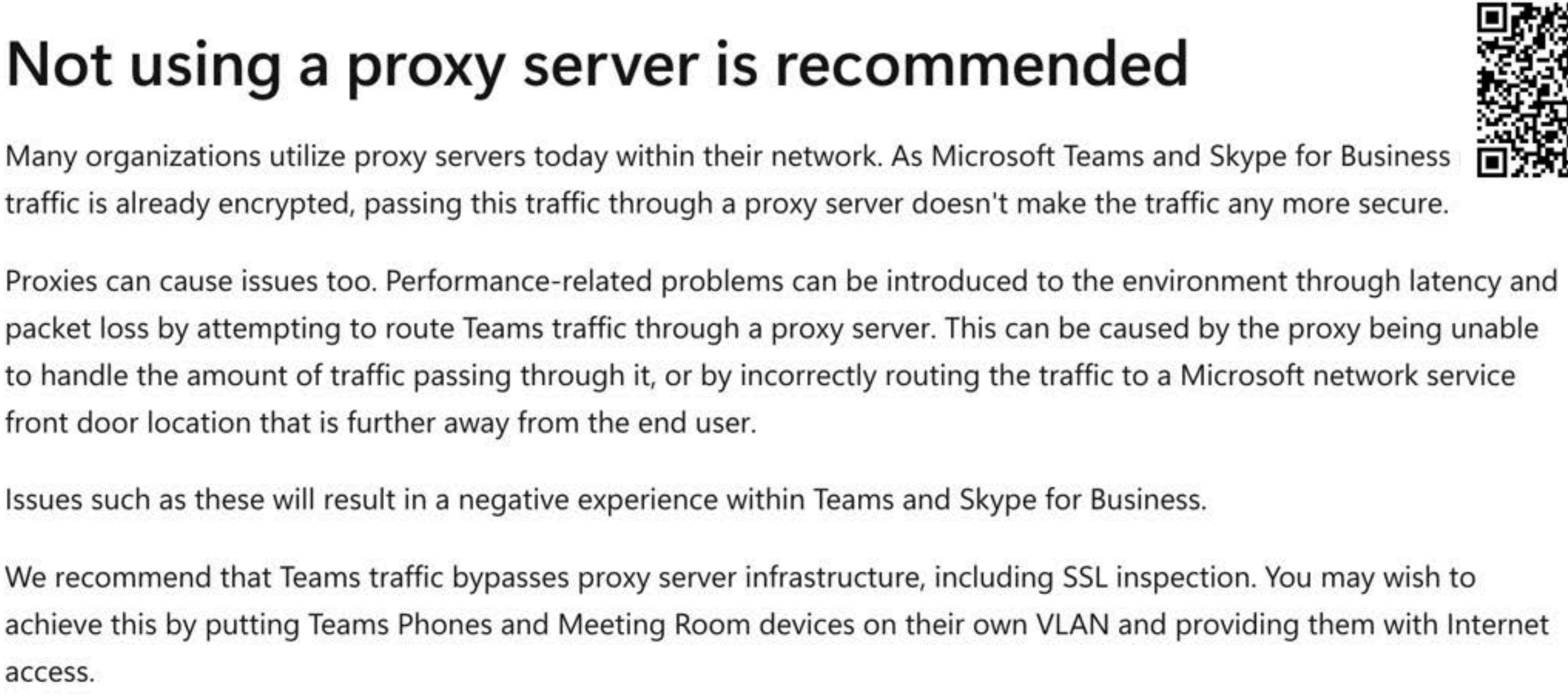
We recommend that you provide an alternate path for Teams traffic that bypasses the virtual private network (VPN), commonly known as split-tunnel VPN. Split tunneling means that traffic for Microsoft 365 or Office 365 doesn't go through the VPN but instead goes directly to Microsoft 365 or Office 365. Bypassing your VPN has a positive impact on Teams quality, and it reduces load from the VPN devices and the organization's network. To implement a split-tunnel VPN, work with your VPN vendor.

Other reasons why we recommend bypassing the VPN:

- VPNs are typically not designed or configured to support real-time media.
- Some VPNs might also not support UDP (which is required for Teams).
- VPNs also introduce an extra layer of encryption on top of media traffic that's already encrypted.
- Connectivity to Teams might not be efficient due to hair-pinning traffic through a VPN device.
- Traffic might be routed to a service front door location that is further away from the end user, introducing extra latency and jitter.

https://learn.microsoft.com/en-us/microsoftteams/prepare-network

# Microsoft Teams TLS Inspection



## Not using a proxy server is recommended

Many organizations utilize proxy servers today within their network. As Microsoft Teams and Skype for Business traffic is already encrypted, passing this traffic through a proxy server doesn't make the traffic any more secure.

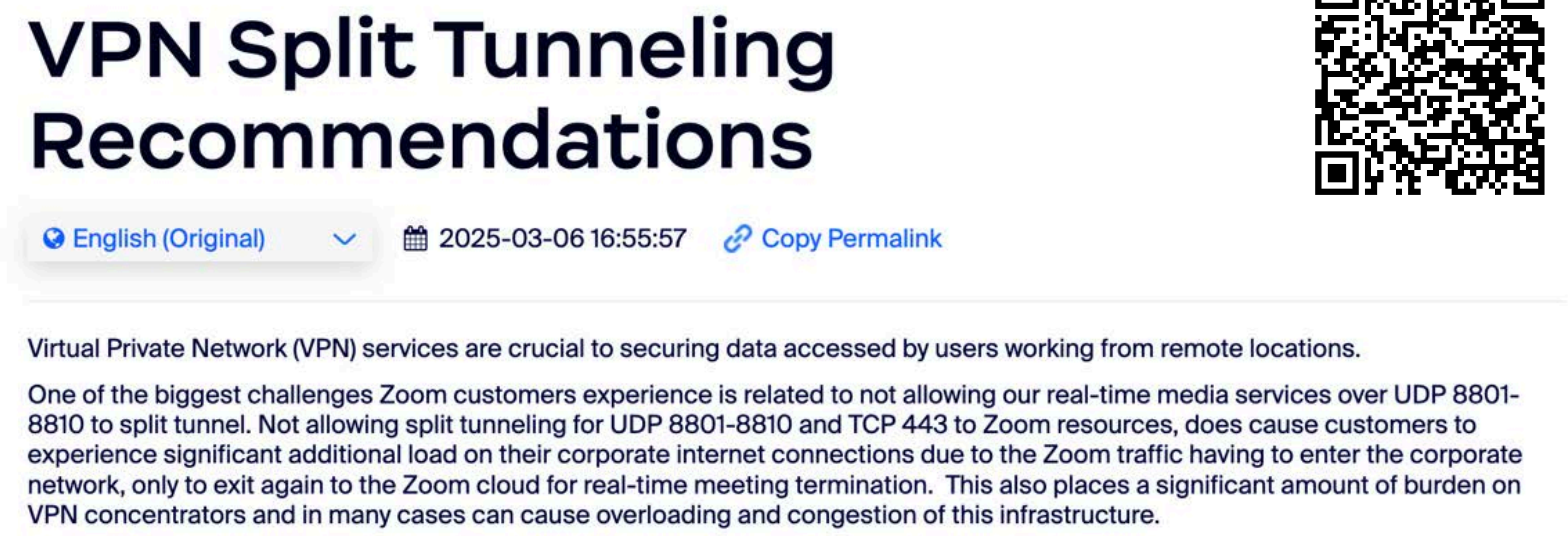
Proxies can cause issues too. Performance-related problems can be introduced to the environment through latency and packet loss by attempting to route Teams traffic through a proxy server. This can be caused by the proxy being unable to handle the amount of traffic passing through it, or by incorrectly routing the traffic to a Microsoft network service front door location that is further away from the end user.

Issues such as these will result in a negative experience within Teams and Skype for Business.

We recommend that Teams traffic bypasses proxy server infrastructure, including SSL inspection. You may wish to achieve this by putting Teams Phones and Meeting Room devices on their own VLAN and providing them with Internet access.

https://learn.microsoft.com/en-us/microsoftteams/proxy-servers-for-skype-for-business-online

# Zoom Split Tunneling Recommendations



## VPN Split Tunneling Recommendations

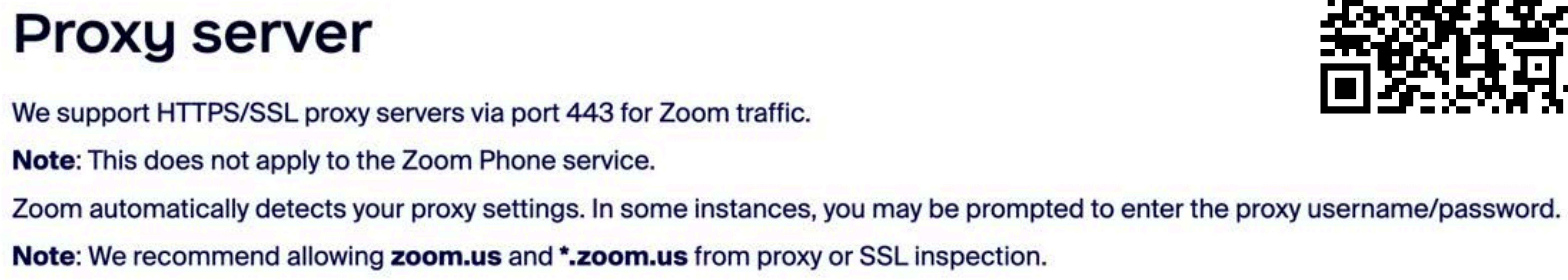
2025-03-06 16:55:57

Virtual Private Network (VPN) services are crucial to securing data accessed by users working from remote locations.

One of the biggest challenges Zoom customers experience is related to not allowing our real-time media services over UDP 8801-8810 to split tunnel. Not allowing split tunneling for UDP 8801-8810 and TCP 443 to Zoom resources, does cause customers to experience significant additional load on their corporate internet connections due to the Zoom traffic having to enter the corporate network, only to exit again to the Zoom cloud for real-time meeting termination.  This also places a significant amount of burden on VPN concentrators and in many cases can cause overloading and congestion of this infrastructure.

https://support.zoom.com/hc/en/article?id=zm_kb&sysparm_article=KB0065998

# Zoom TLS Inspection Recommendations



## Proxy server

We support HTTPS/SSL proxy servers via port 443 for Zoom traffic.

**Note**: This does not apply to the Zoom Phone service.

Zoom automatically detects your proxy settings. In some instances, you may be prompted to enter the proxy username/password.

**Note**: We recommend allowing **zoom.us** and ***.zoom.us** from proxy or SSL inspection.

https://support.zoom.com/hc/en/article?id=zm_kb&sysparm_article=KB0060548

# Quick Disclaimer

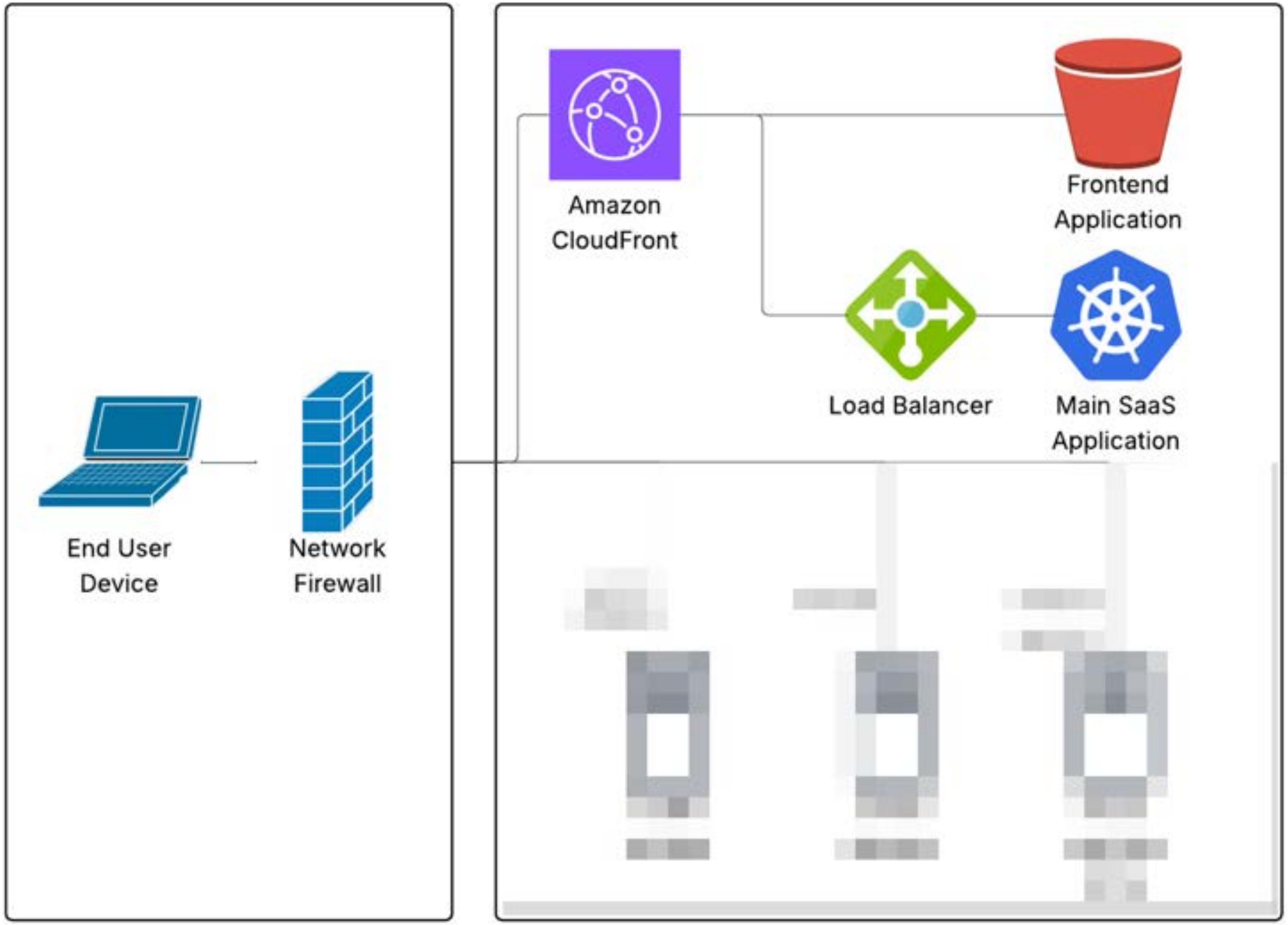
- Providers aren’t being malicious
- Performance is the main design driver
- Latency must be minimized for app reliability
- These configs are often intentional not careless
- Inspection or routing can overwhelm systems

# How does it Work?

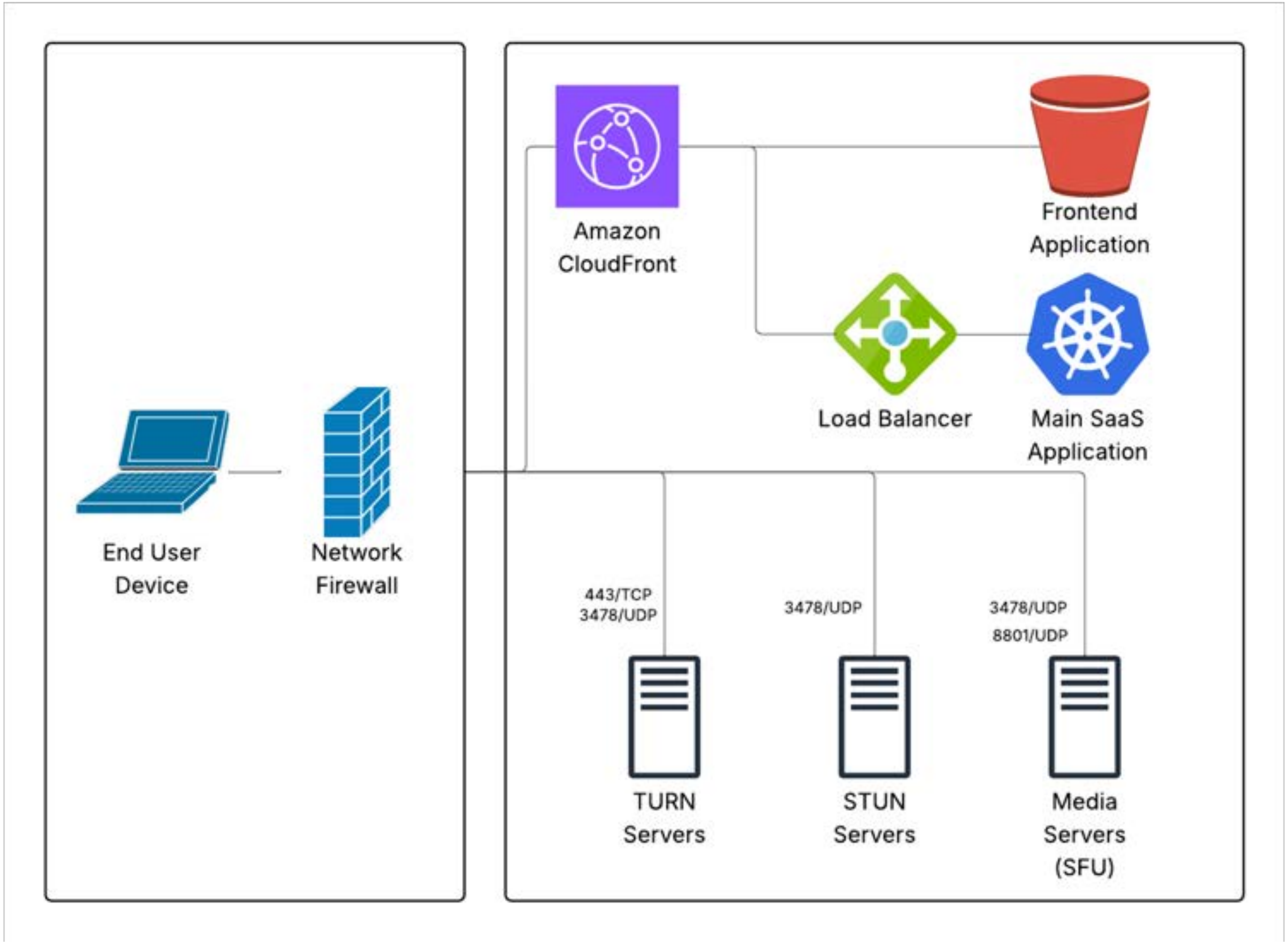
# General Web Conferencing Architecture

End User Device
Network Firewall

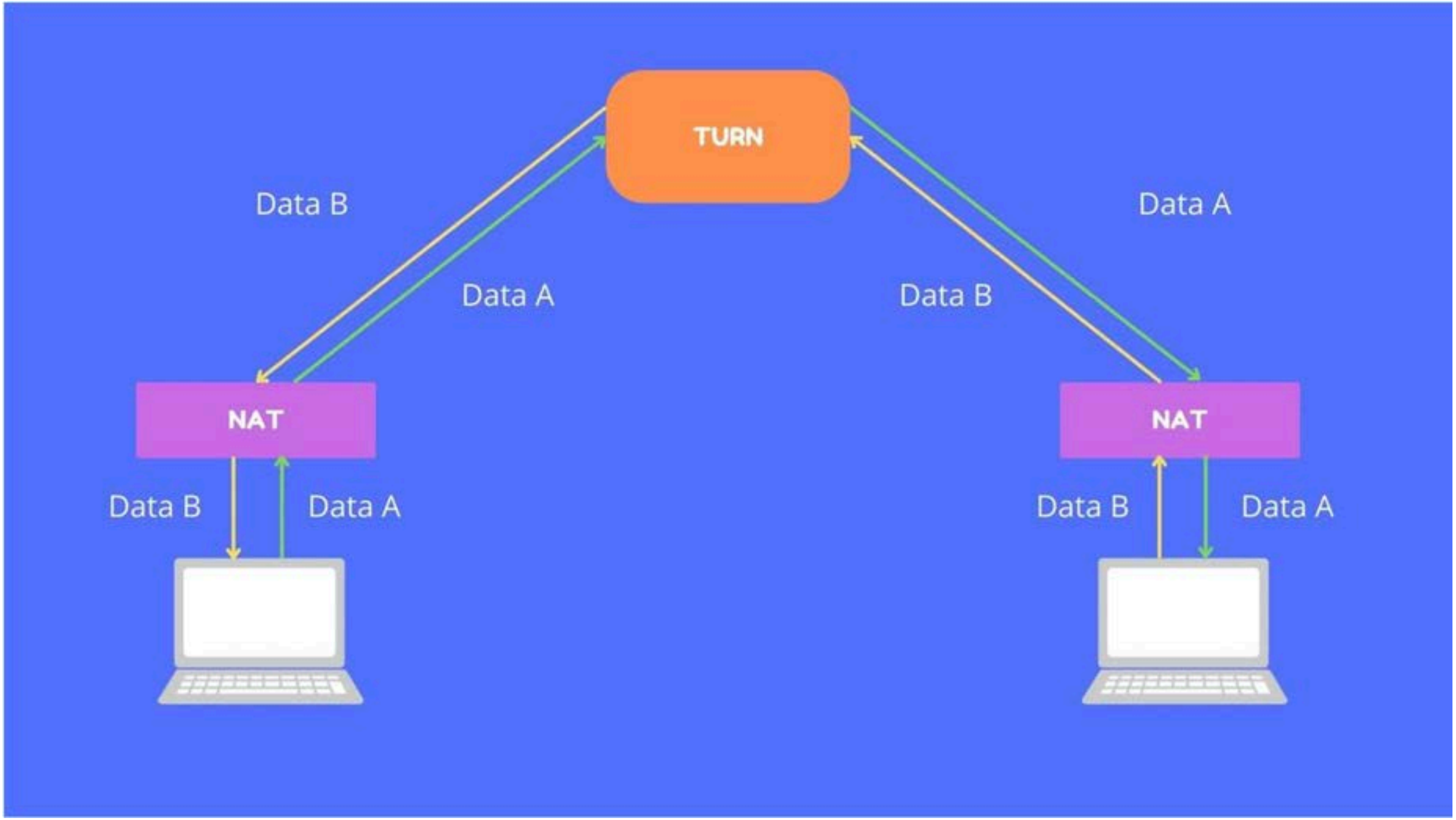
# General Web Conferencing Architecture



# General Web Conferencing Architecture



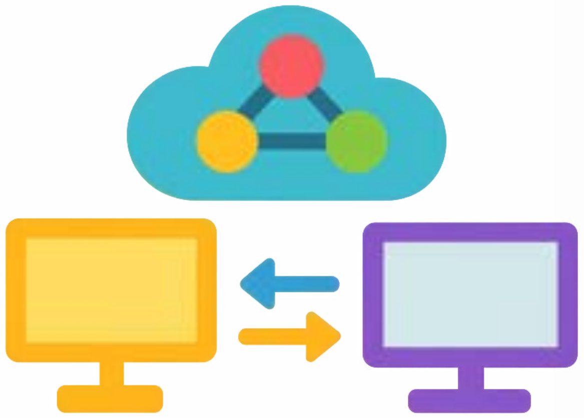
# What is TURN?



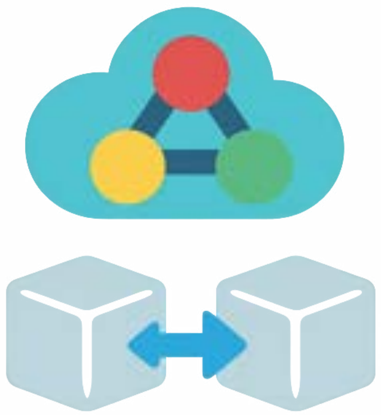
https://www.100ms.live/blog/webrtc-turn-server

# WebRTC Core Protocols

DTLS



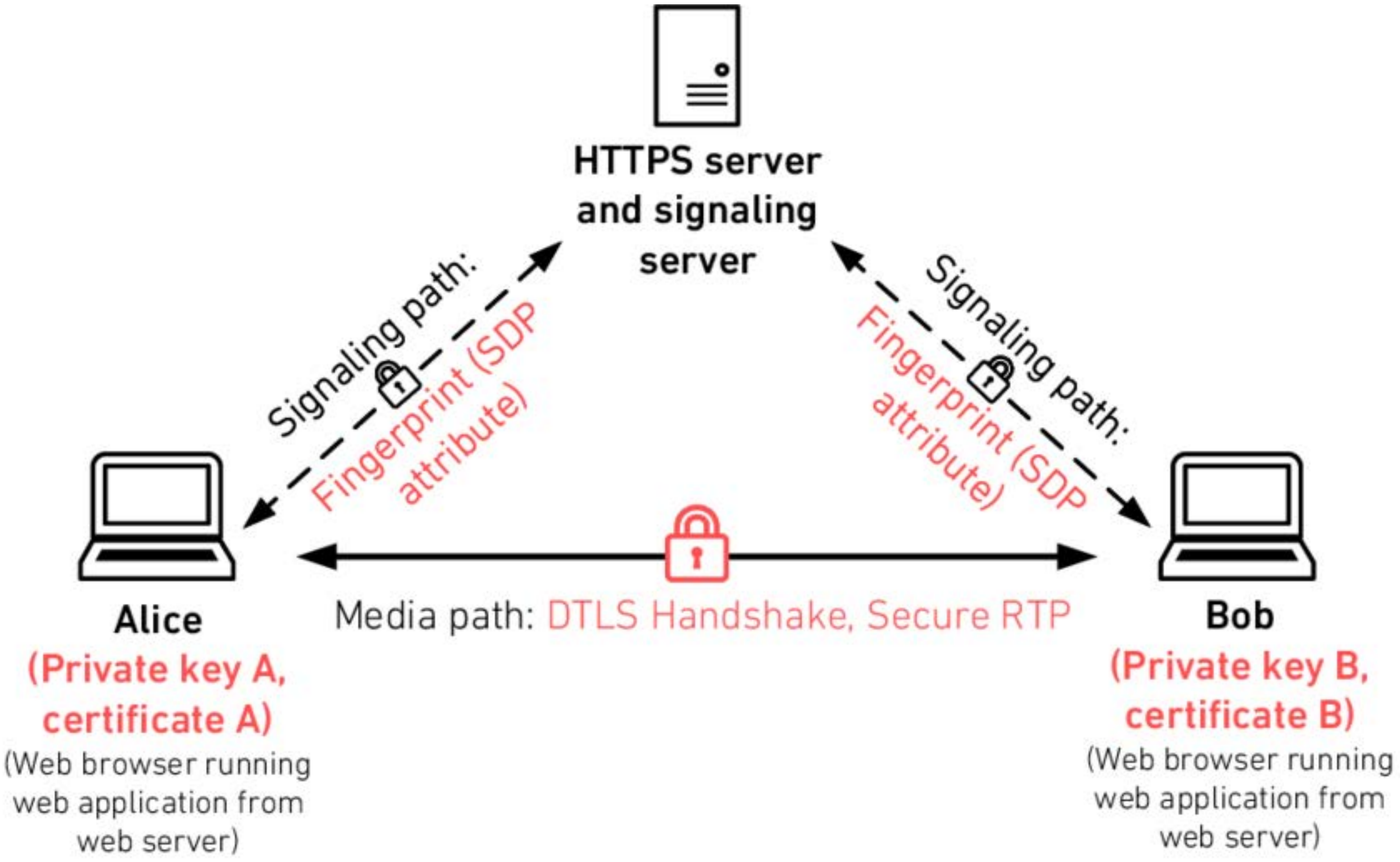
SCTP



ICE

SRTP

# WebRTC Handshake Process



HTTPS server and signaling server

Signaling path: Fingerprint (SDP attribute)

Signaling path: Fingerprint (SDP attribute)

Media path: DTLS Handshake, Secure RTP

**Alice**

(Private key A, certificate A)

(Web browser running web application from web server)

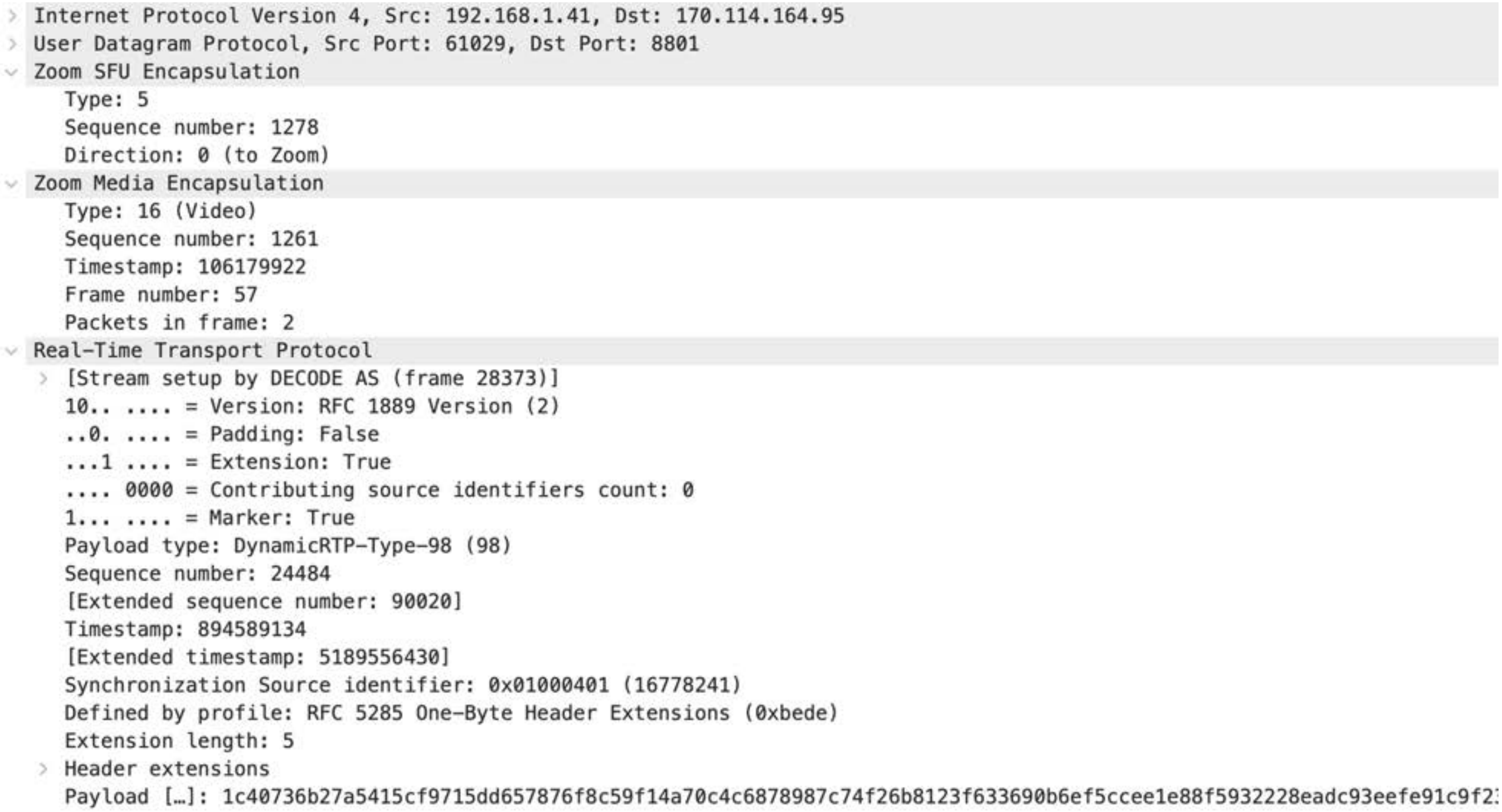
**Bob**

(Private key B, certificate B)

(Web browser running web application from web server)

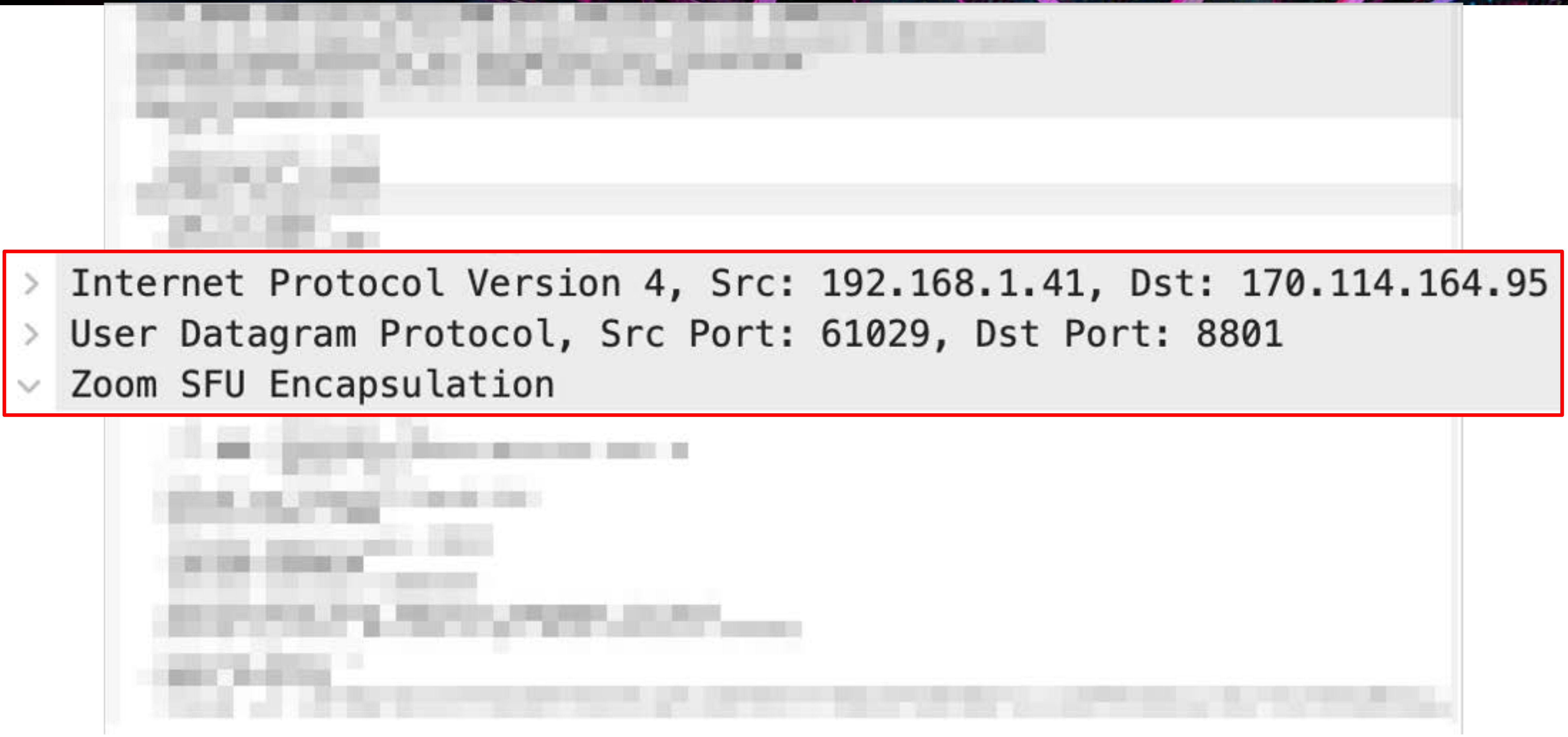
https://www.researchgate.net/figure/WebRTC-triangle-with-DTLS-key-exchange_fig8_328334940

# Reverse Engineering Zoom



```
Internet Protocol Version 4, Src: 192.168.1.41, Dst: 170.114.164.95
User Datagram Protocol, Src Port: 61029, Dst Port: 8801
Zoom SFU Encapsulation
    Type: 5
    Sequence number: 1278
    Direction: 0 (to Zoom)
Zoom Media Encapsulation
    Type: 16 (Video)
    Sequence number: 1261
    Timestamp: 106179922
    Frame number: 57
    Packets in frame: 2
Real-Time Transport Protocol
    [Stream setup by DECODE AS (frame 28373)]
    10.. .... = Version: RFC 1889 Version (2)
    ..0. .... = Padding: False
    ...1 .... = Extension: True
    .... 0000 = Contributing source identifiers count: 0
    1... .... = Marker: True
    Payload type: DynamicRTP-Type-98 (98)
    Sequence number: 24484
    [Extended sequence number: 90020]
    Timestamp: 894589134
    [Extended timestamp: 5189556430]
    Synchronization Source identifier: 0x01000401 (16778241)
    Defined by profile: RFC 5285 One-Byte Header Extensions (0xbede)
    Extension length: 5
    Header extensions
    Payload […]: 1c40736b27a5415cf9715dd657876f8c59f14a70c4c6878987c74f26b8123f633690b6ef5ccee1e88f5932228eadc93eefe91c9f2
```

# Reverse Engineering Zoom

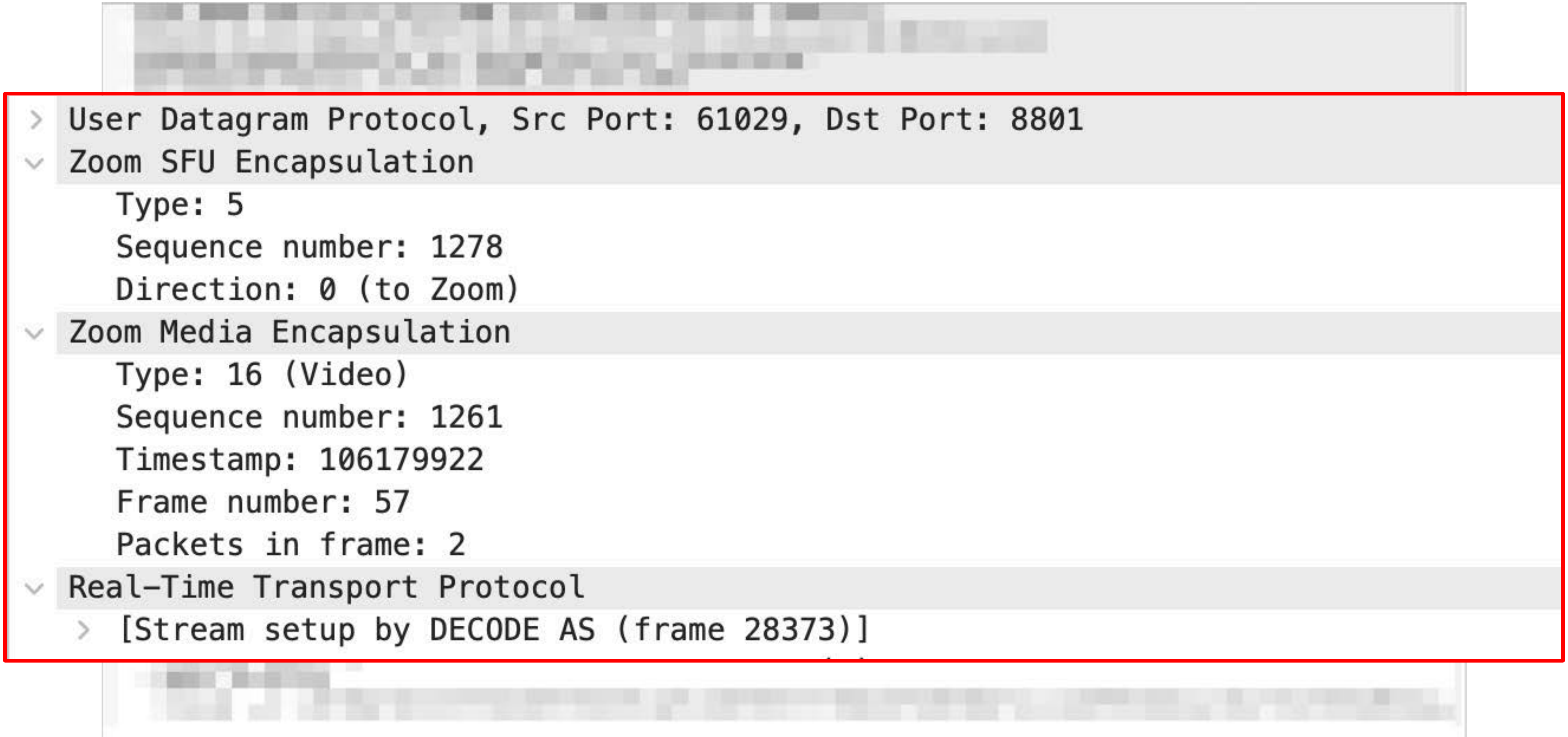


[redacted]

```
> Internet Protocol Version 4, Src: 192.168.1.41, Dst: 170.114.164.95
> User Datagram Protocol, Src Port: 61029, Dst Port: 8801
v Zoom SFU Encapsulation
```

[redacted]

# Reverse Engineering Zoom



[redacted]

```
> User Datagram Protocol, Src Port: 61029, Dst Port: 8801
v Zoom SFU Encapsulation
    Type: 5
    Sequence number: 1278
    Direction: 0 (to Zoom)
v Zoom Media Encapsulation
    Type: 16 (Video)
    Sequence number: 1261
    Timestamp: 106179922
    Frame number: 57
    Packets in frame: 2
v Real-Time Transport Protocol
  > [Stream setup by DECODE AS (frame 28373)]
```

[redacted]

# Building on Existing Work

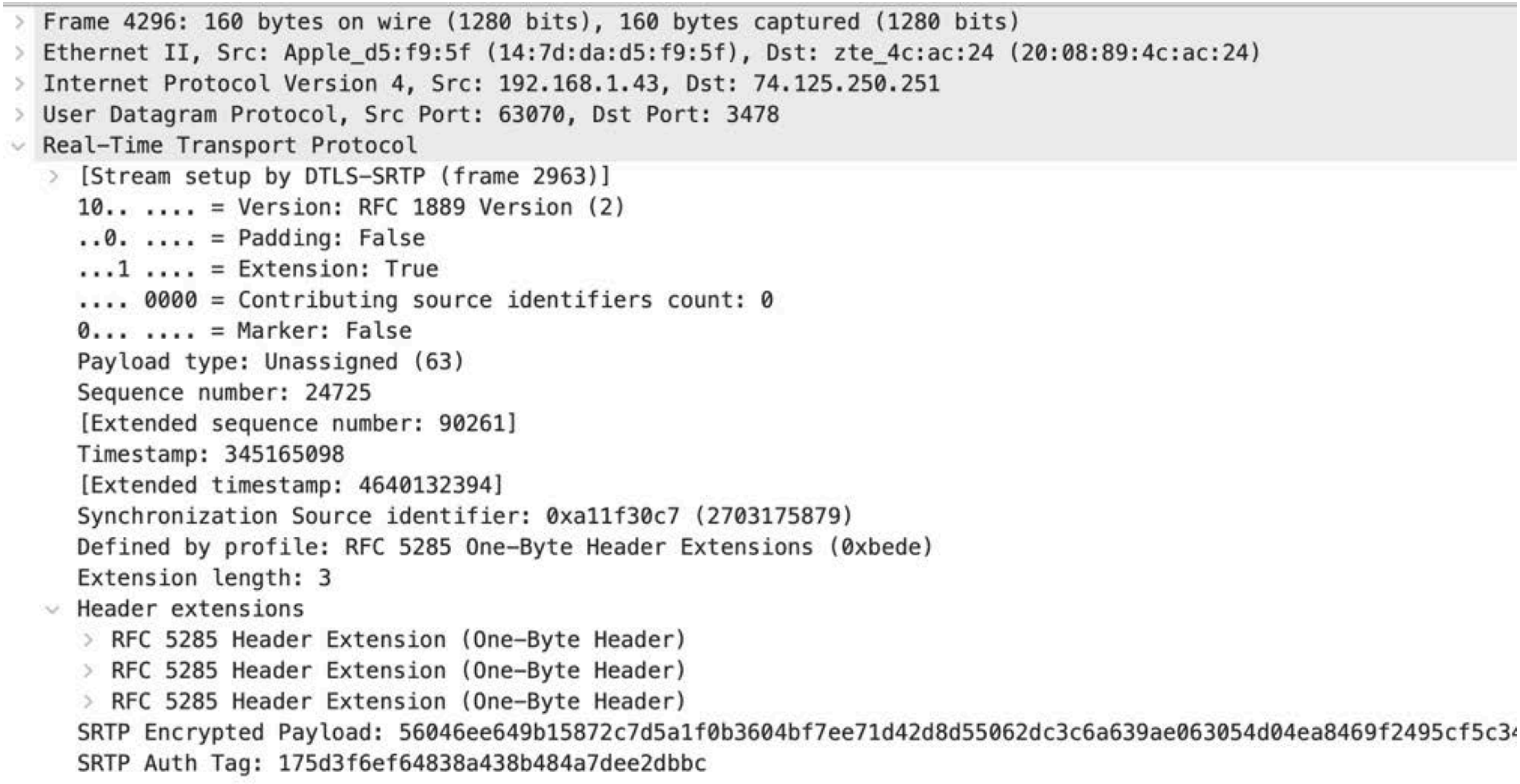
Enabling Passive Measurement of Zoom Performance in Production Networks

https://dl.acm.org/doi/pdf/10.1145/3517745.3561414

Custom Wireshark Analyzer for Zoom Desktop Media Traffic

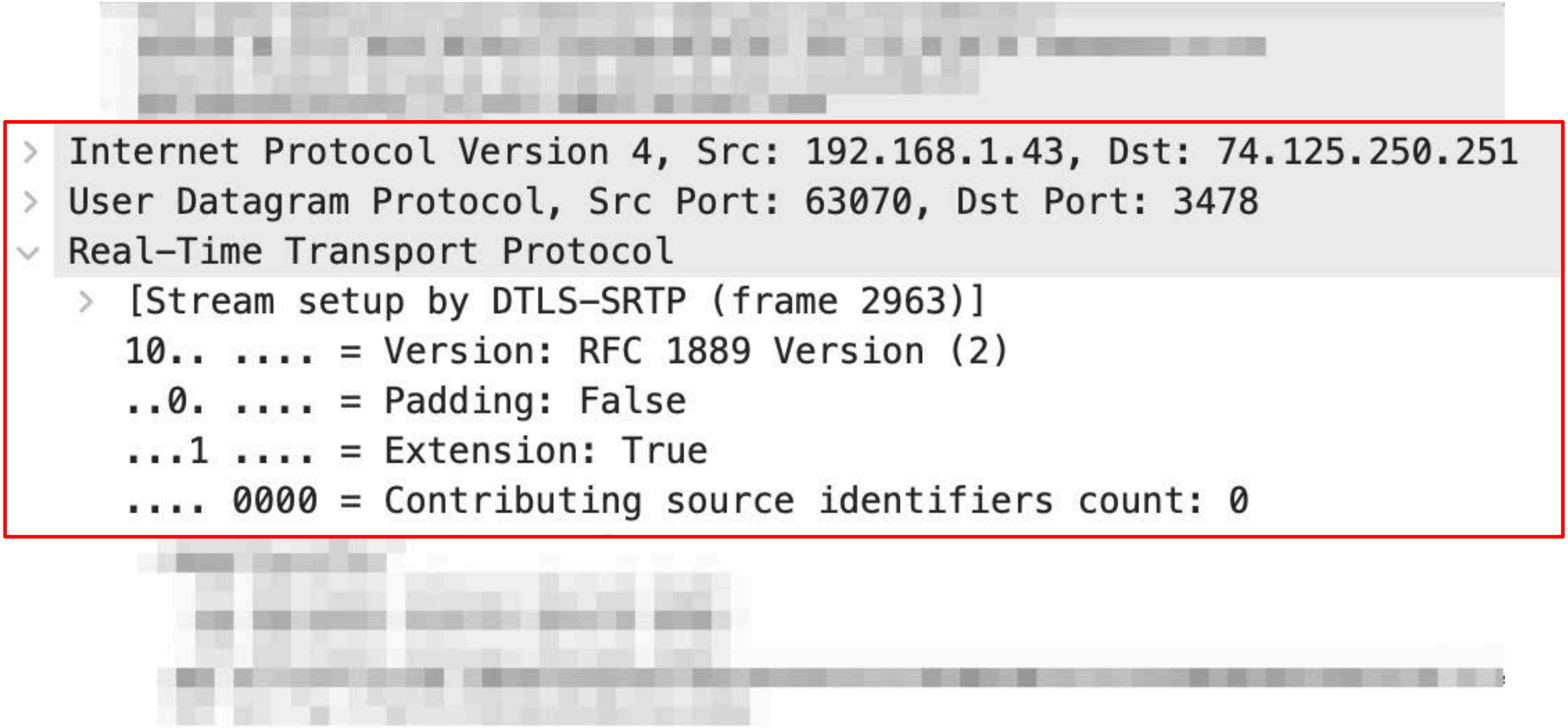
https://github.com/Princeton-Cabernet/zoom-analysis

# Reverse Engineering Google Meet



```
> Frame 4296: 160 bytes on wire (1280 bits), 160 bytes captured (1280 bits)
> Ethernet II, Src: Apple_d5:f9:5f (14:7d:da:d5:f9:5f), Dst: zte_4c:ac:24 (20:08:89:4c:ac:24)
> Internet Protocol Version 4, Src: 192.168.1.43, Dst: 74.125.250.251
> User Datagram Protocol, Src Port: 63070, Dst Port: 3478
v Real-Time Transport Protocol
  > [Stream setup by DTLS-SRTP (frame 2963)]
    10.. .... = Version: RFC 1889 Version (2)
    ..0. .... = Padding: False
    ...1 .... = Extension: True
    .... 0000 = Contributing source identifiers count: 0
    0... .... = Marker: False
    Payload type: Unassigned (63)
    Sequence number: 24725
    [Extended sequence number: 90261]
    Timestamp: 345165098
    [Extended timestamp: 4640132394]
    Synchronization Source identifier: 0xa11f30c7 (2703175879)
    Defined by profile: RFC 5285 One-Byte Header Extensions (0xbede)
    Extension length: 3
  v Header extensions
    > RFC 5285 Header Extension (One-Byte Header)
    > RFC 5285 Header Extension (One-Byte Header)
    > RFC 5285 Header Extension (One-Byte Header)
    SRTP Encrypted Payload: 56046ee649b15872c7d5a1f0b3604bf7ee71d42d8d55062dc3c6a639ae063054d04ea8469f2495cf5c3
    SRTP Auth Tag: 175d3f6ef64838a438b484a7dee2dbbc
```

# Reverse Engineering Google Meet

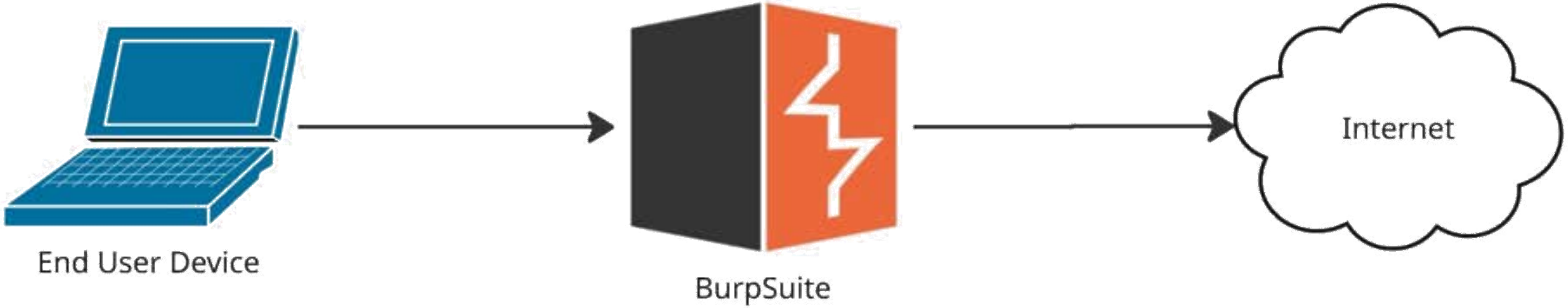


[redacted]

```
> Internet Protocol Version 4, Src: 192.168.1.43, Dst: 74.125.250.251
> User Datagram Protocol, Src Port: 63070, Dst Port: 3478
v Real-Time Transport Protocol
    > [Stream setup by DTLS-SRTP (frame 2963)]
      10.. .... = Version: RFC 1889 Version (2)
      ..0. .... = Padding: False
      ...1 .... = Extension: True
      .... 0000 = Contributing source identifiers count: 0
```

[redacted]

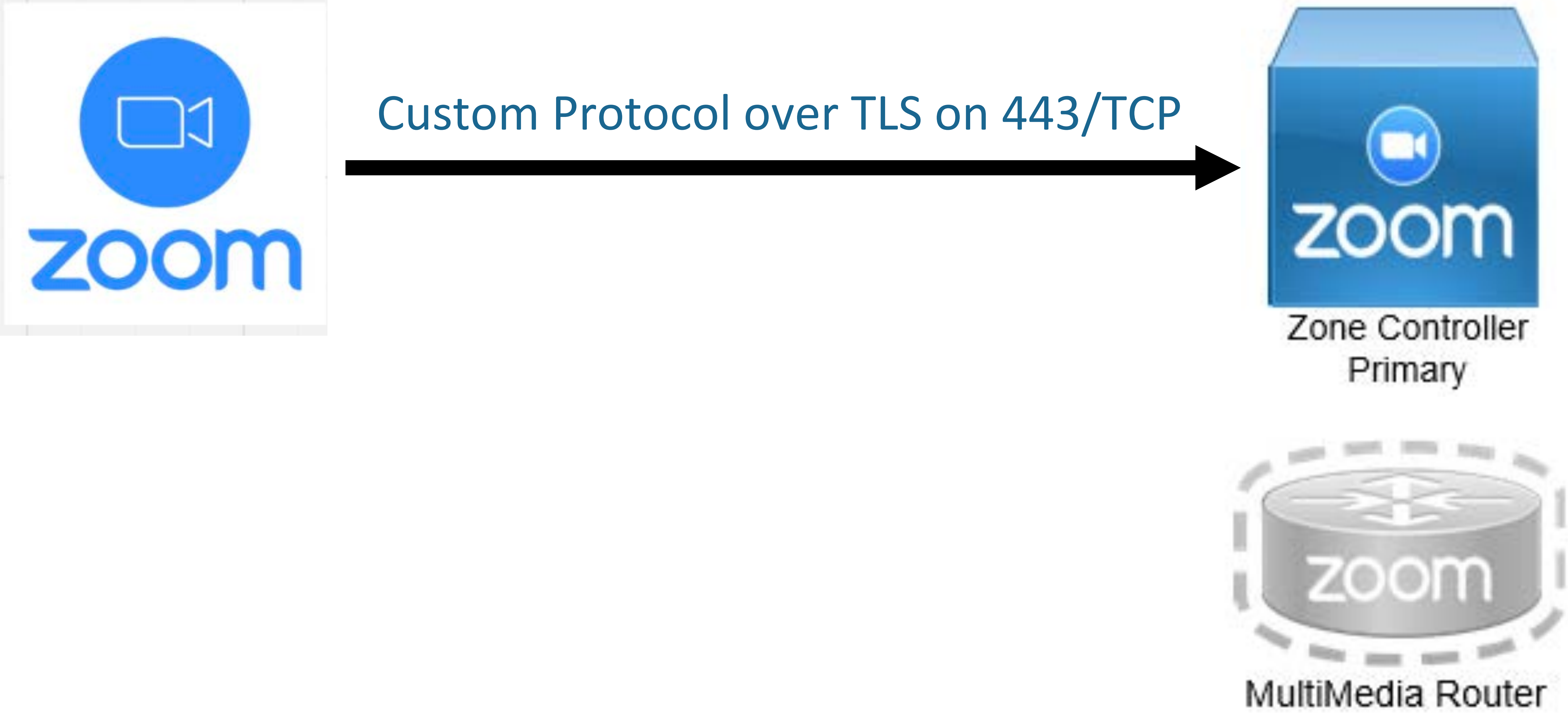
# Case Study in Egress Resilience



# Example Zoom Desktop Egress Attempts

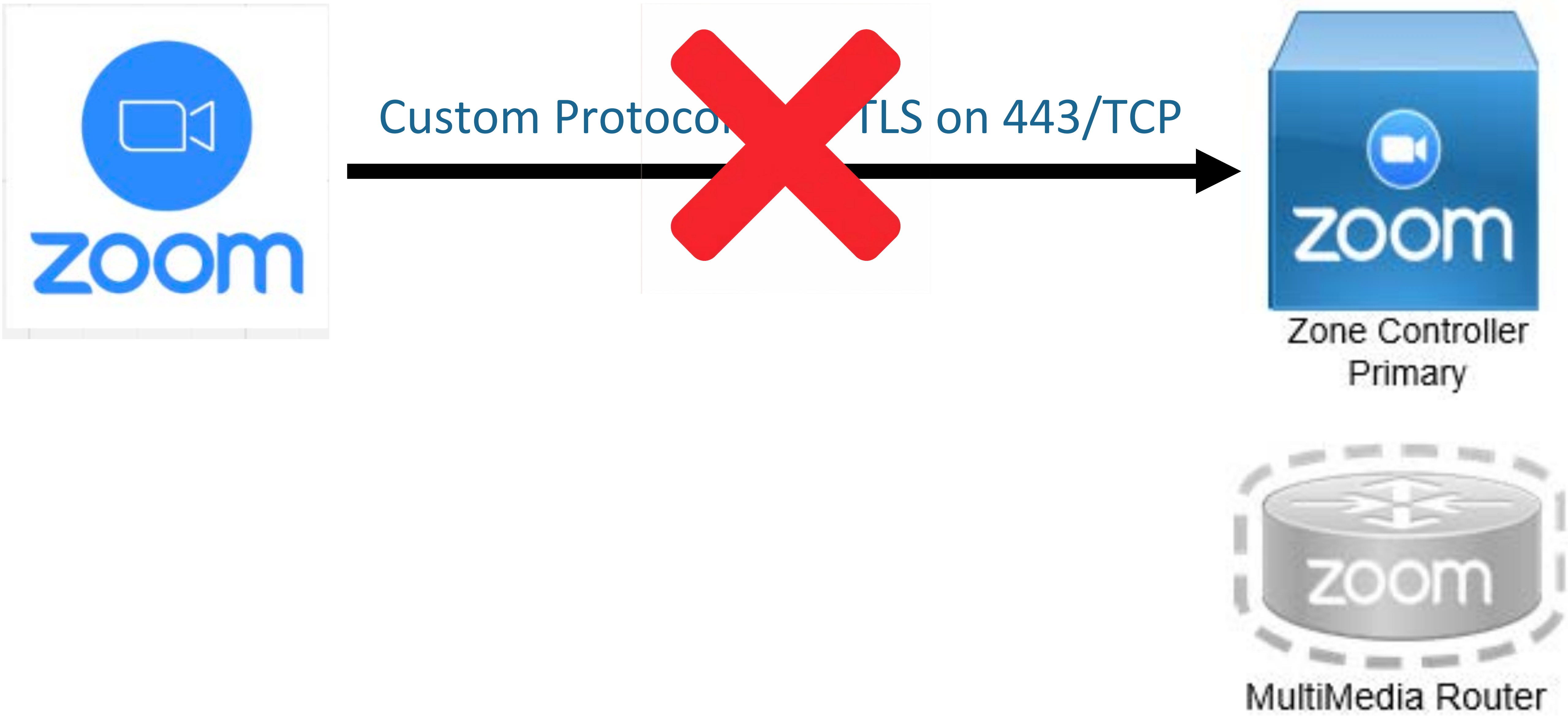
zoom
zoom
Zone Controller Primary
zoom
MultiMedia Router

# Example Zoom Desktop Egress Attempts



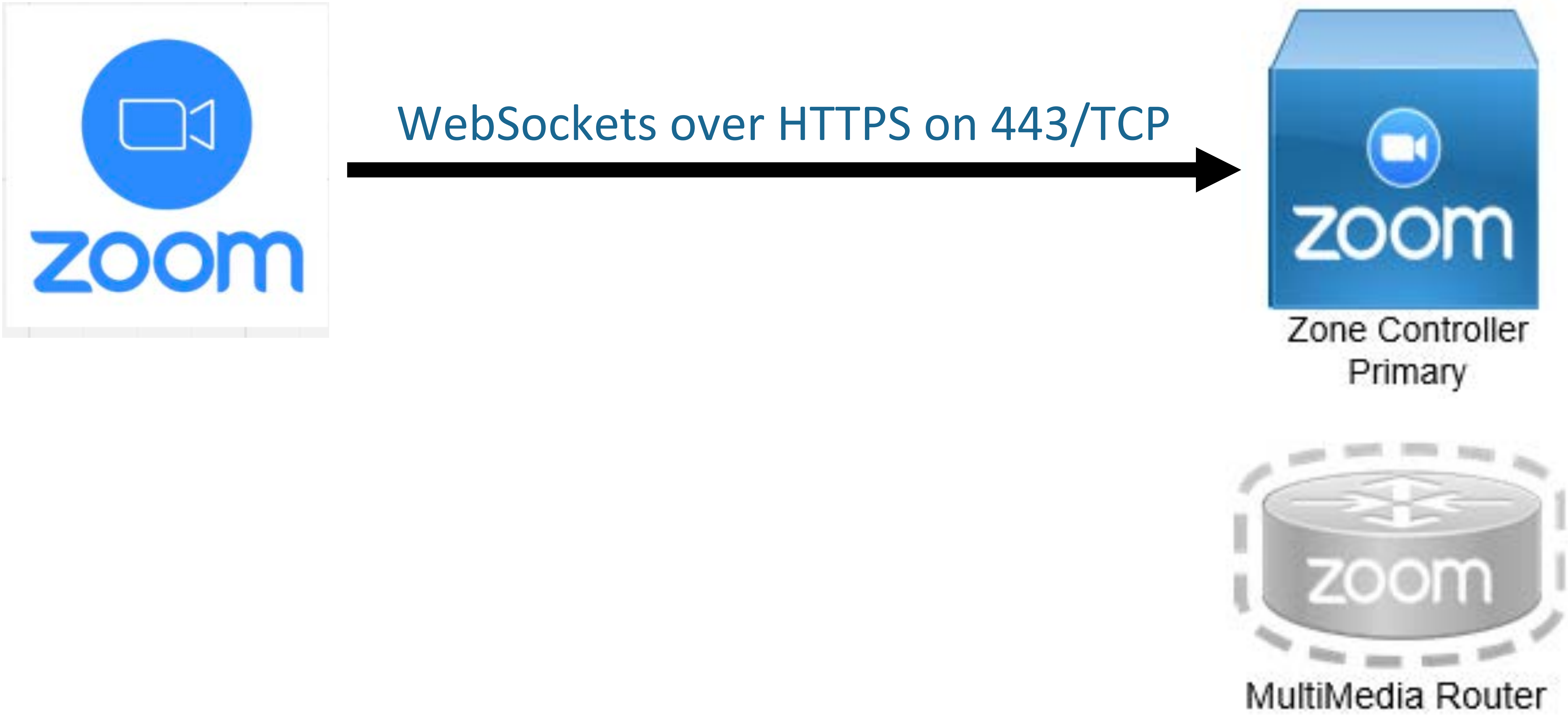
Custom Protocol over TLS on 443/TCP

# Example Zoom Desktop Egress Attempts



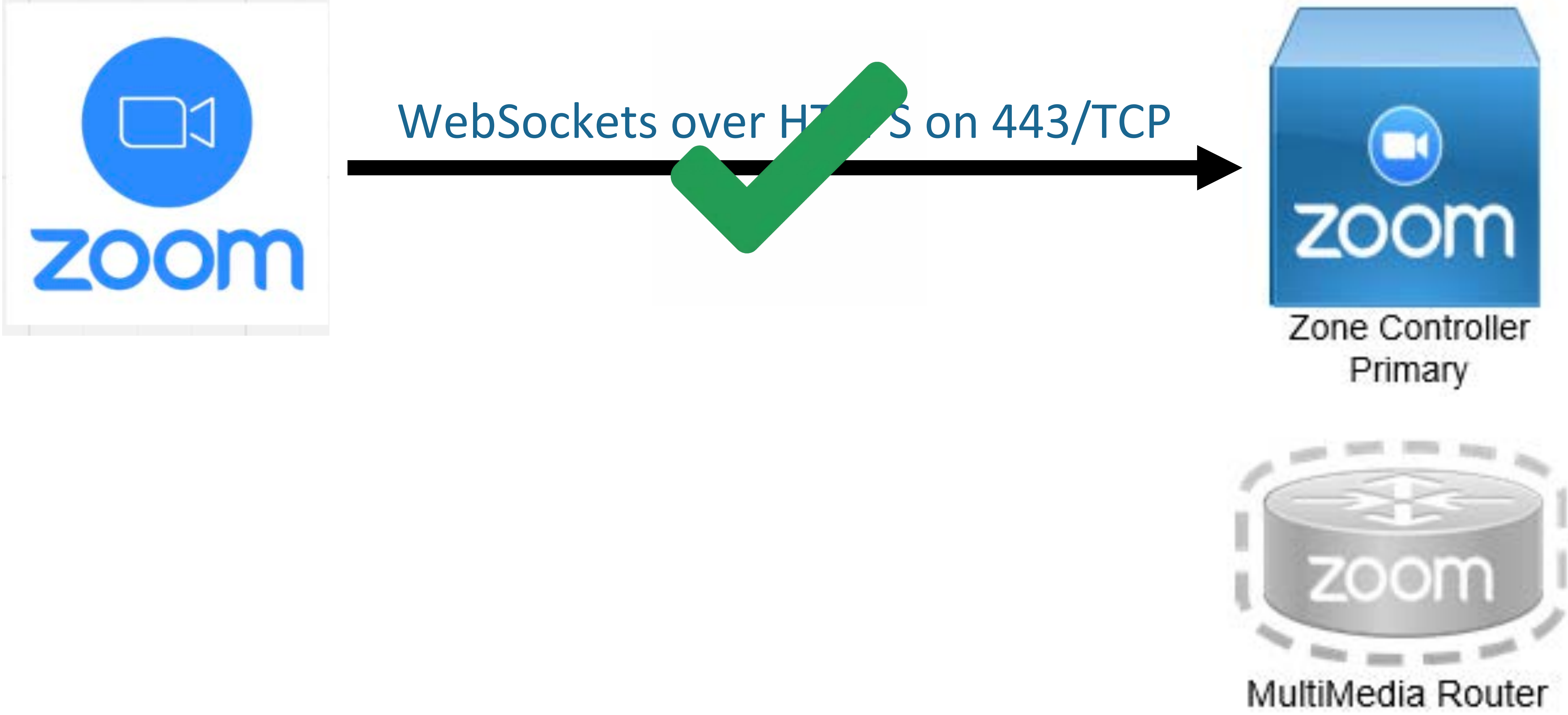
Custom Protoco TLS on 443/TCP

# Example Zoom Desktop Egress Attempts



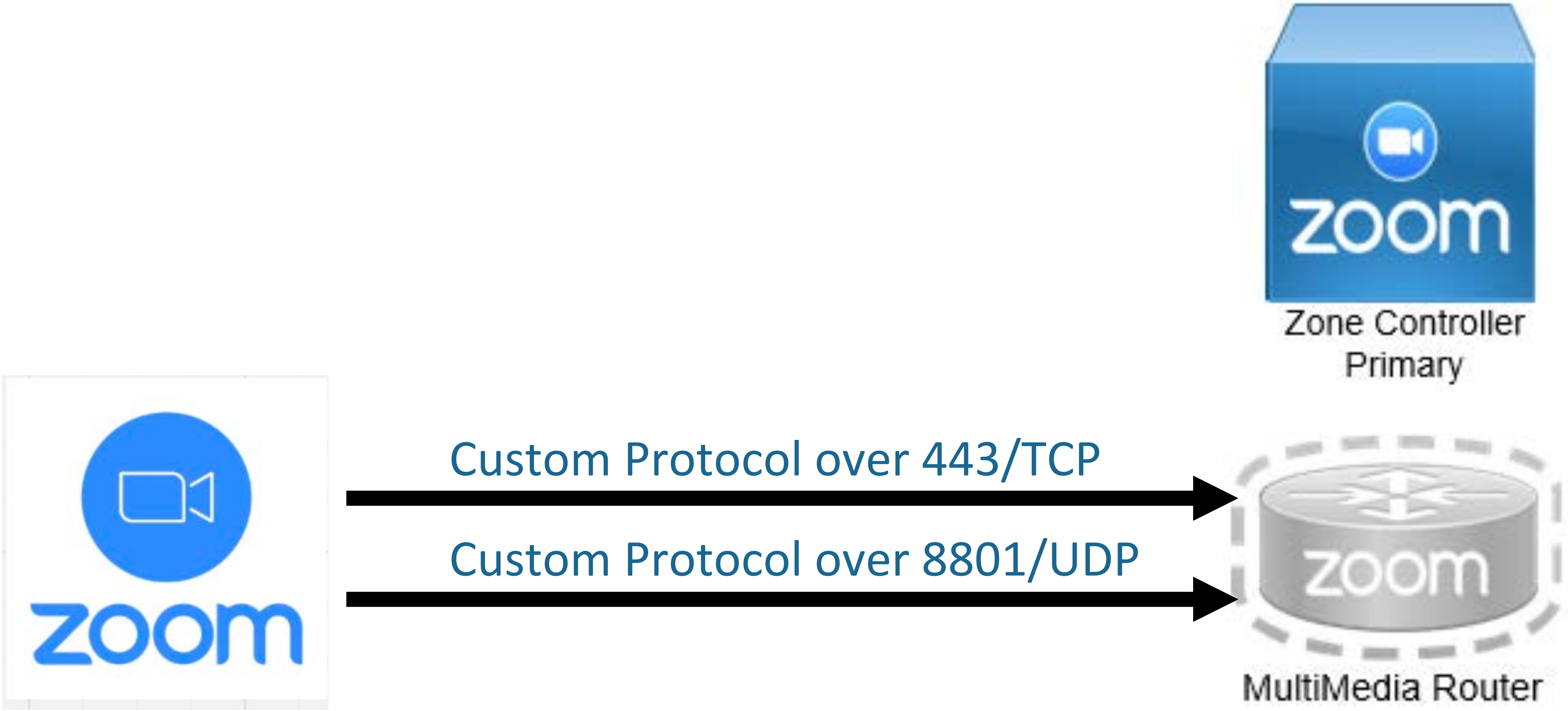
WebSockets over HTTPS on 443/TCP

# Example Zoom Desktop Egress Attempts

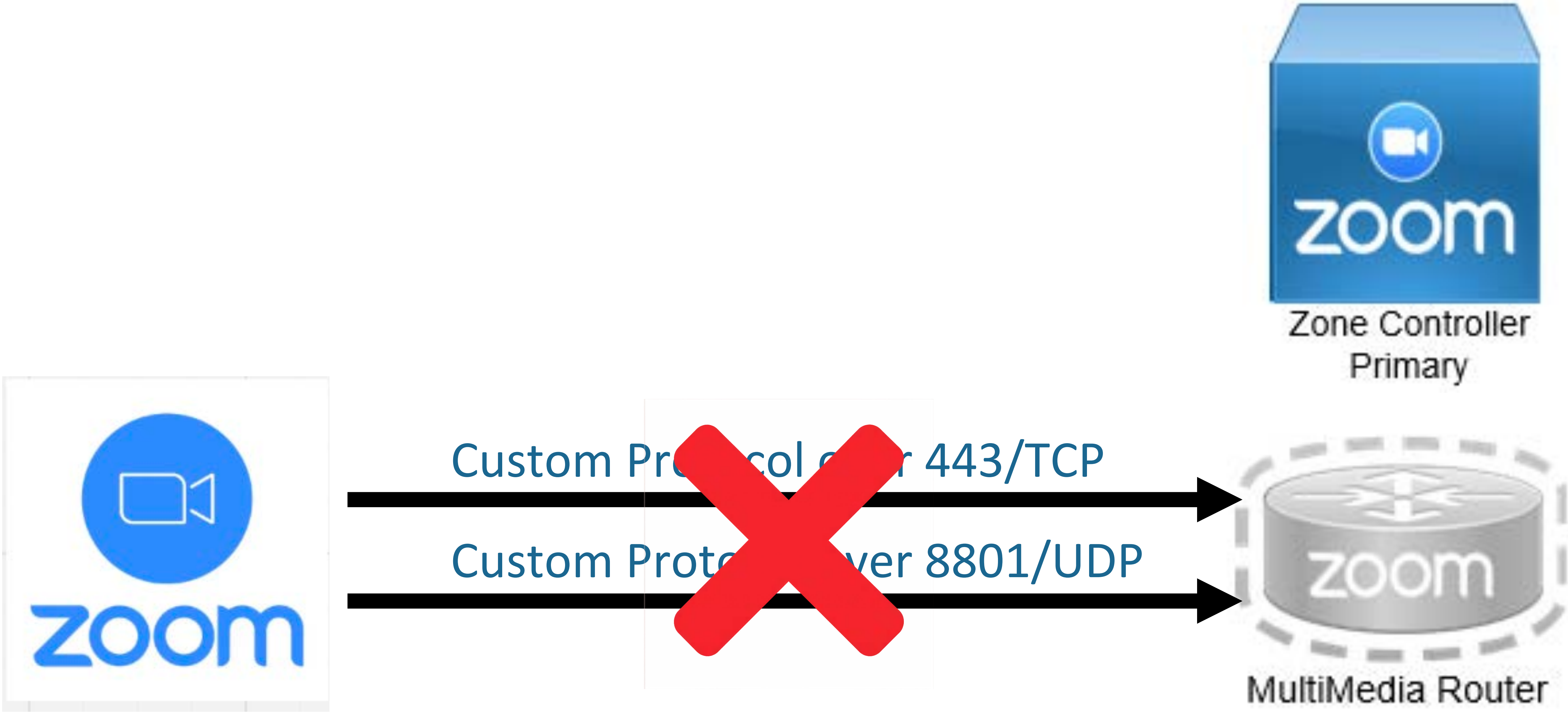


WebSockets over HTTPS on 443/TCP

# Example Zoom Desktop Egress Attempts



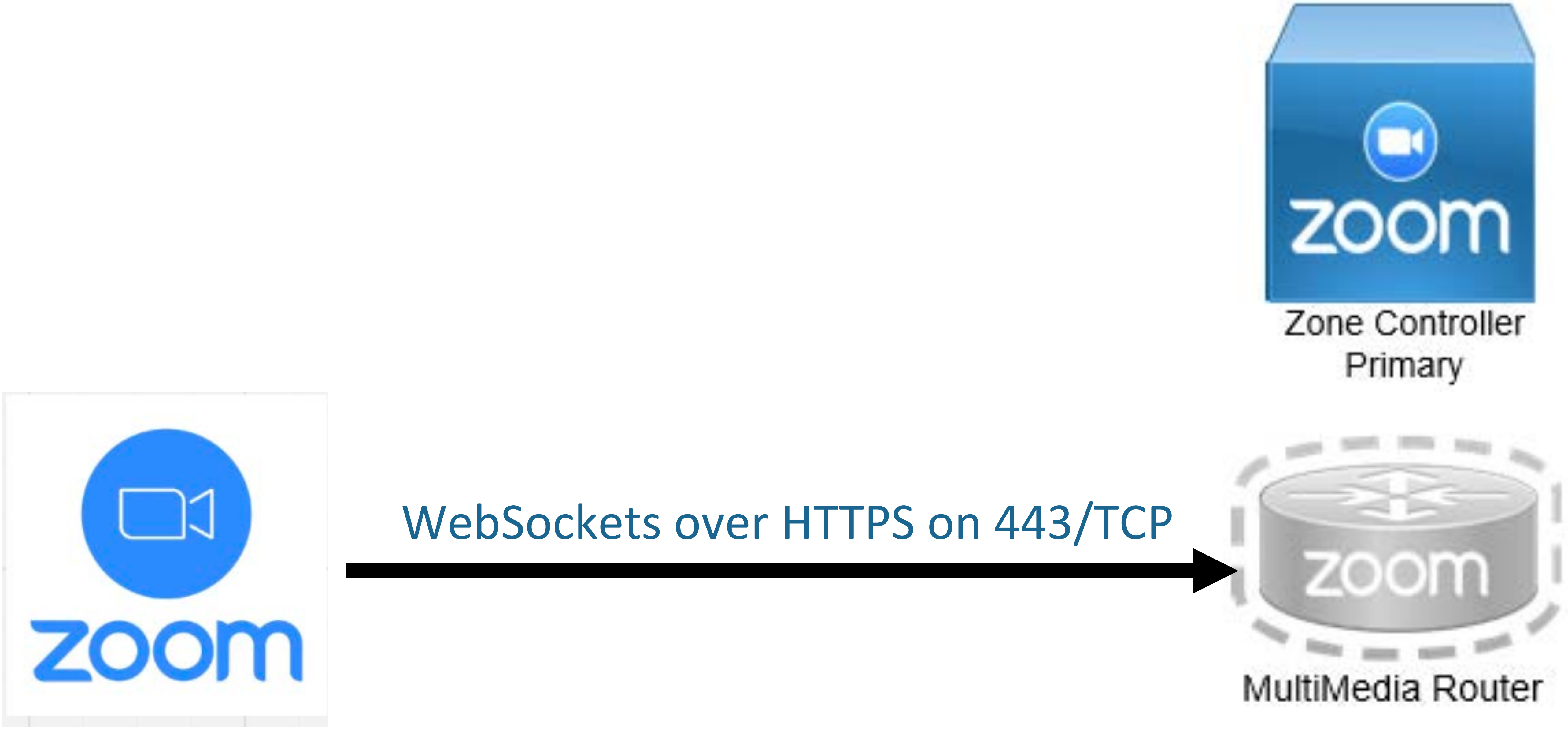
# Example Zoom Desktop Egress Attempts



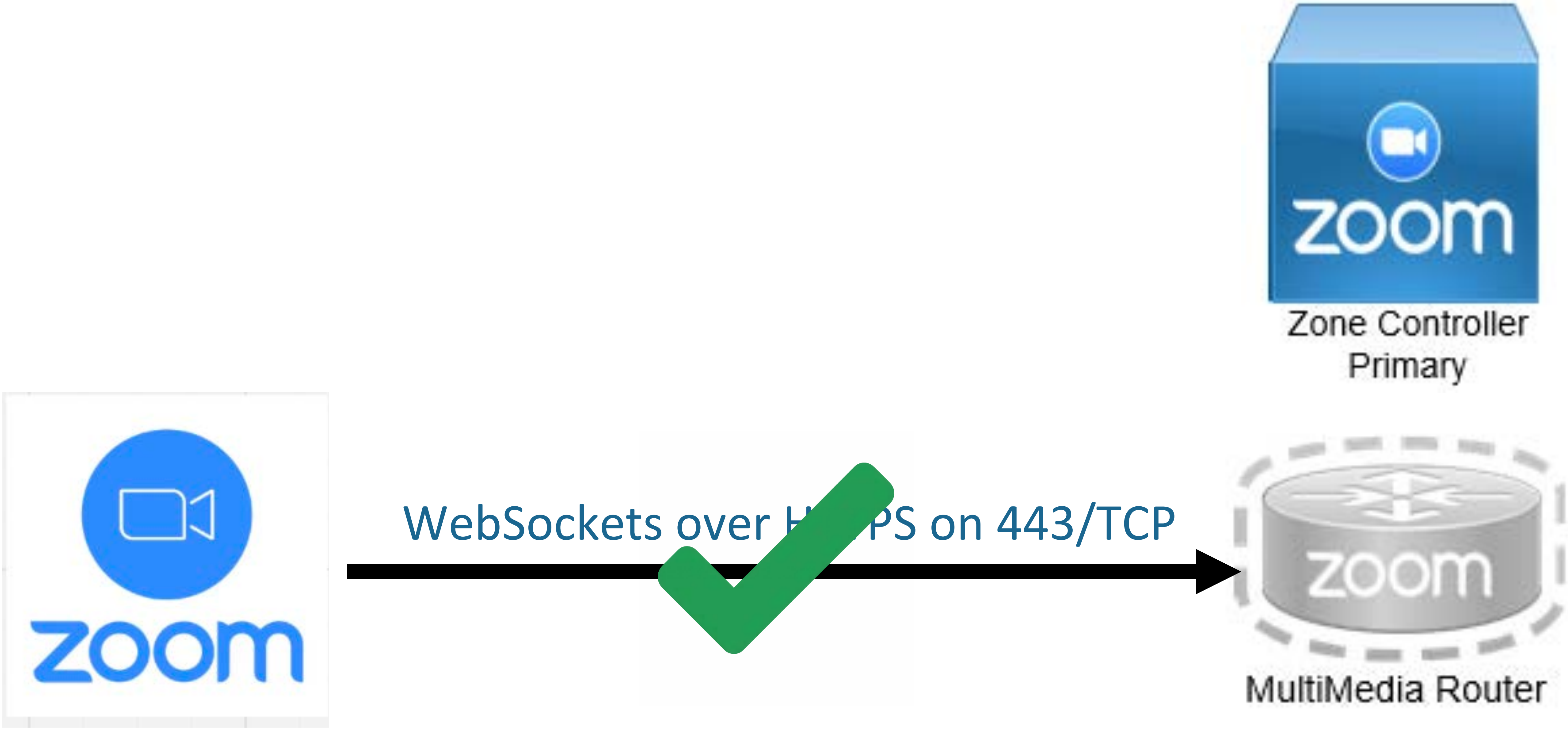
Custom Pr col r 443/TCP

Custom Prot ver 8801/UDP

# Example Zoom Desktop Egress Attempts

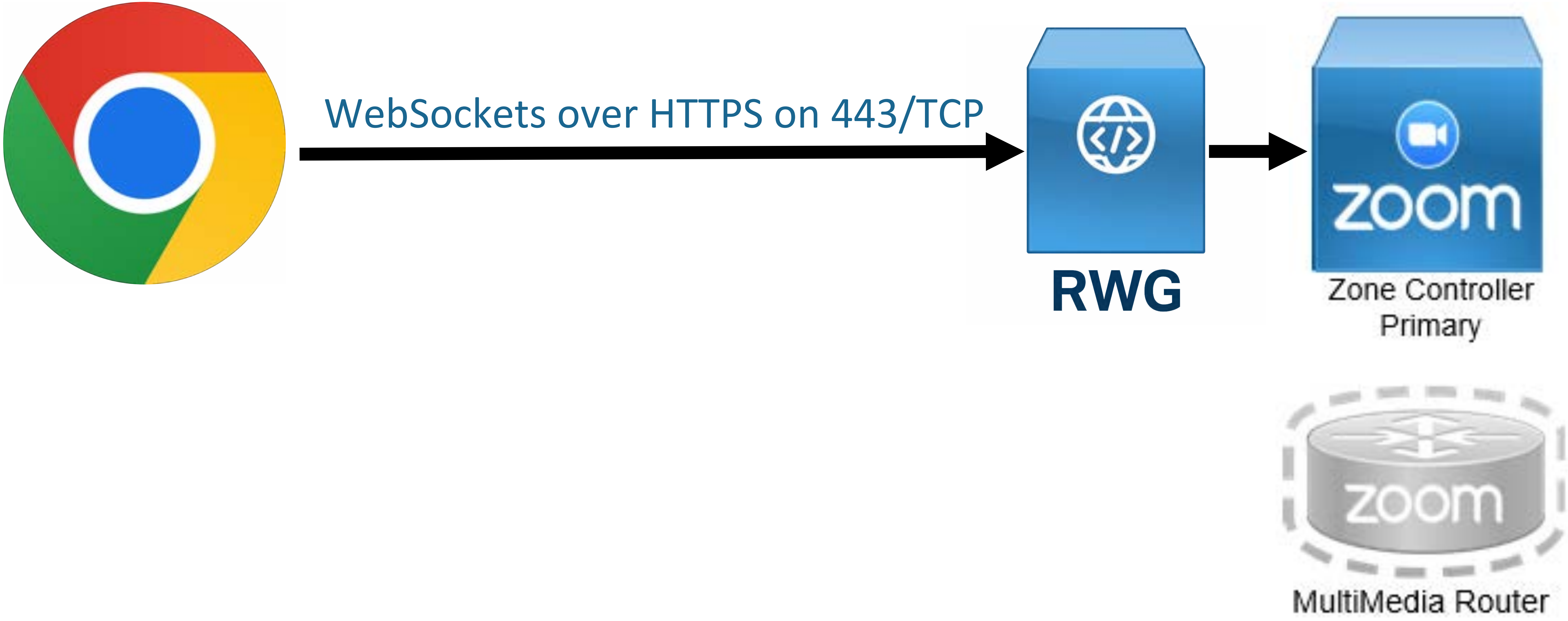


# Example Zoom Desktop Egress Attempts

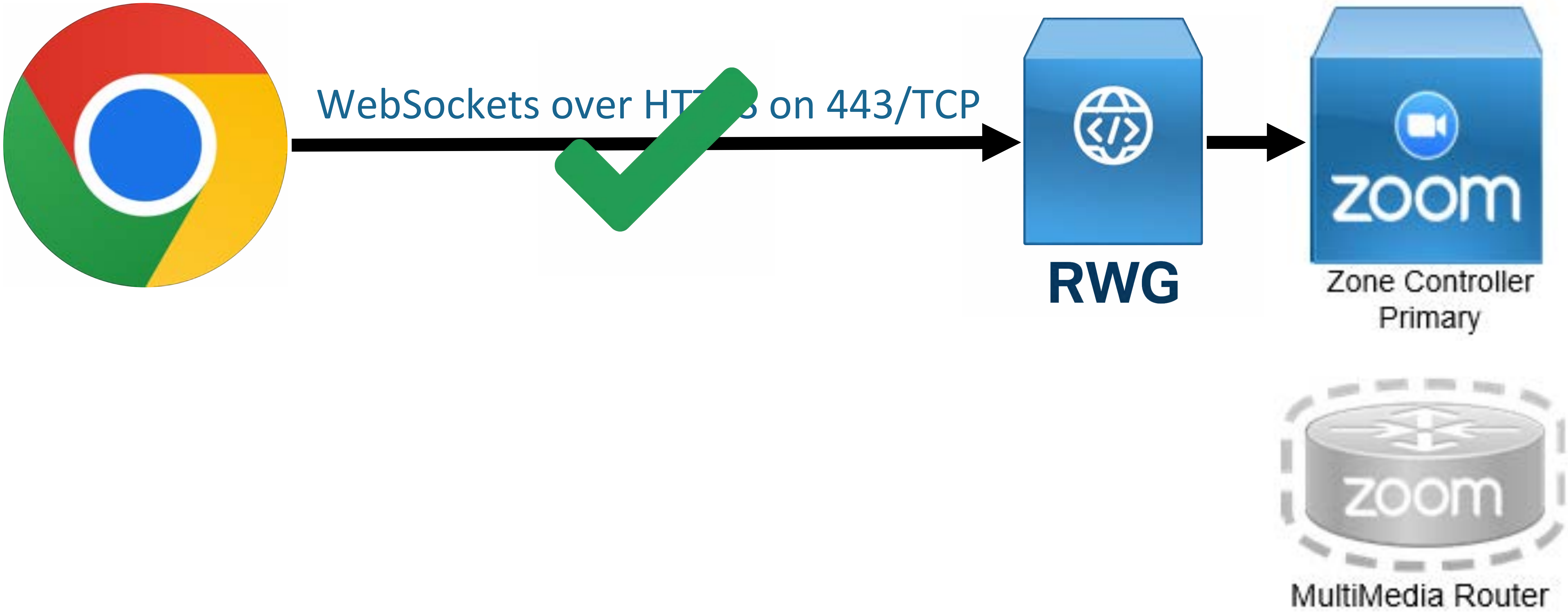


WebSockets over HTTPS on 443/TCP

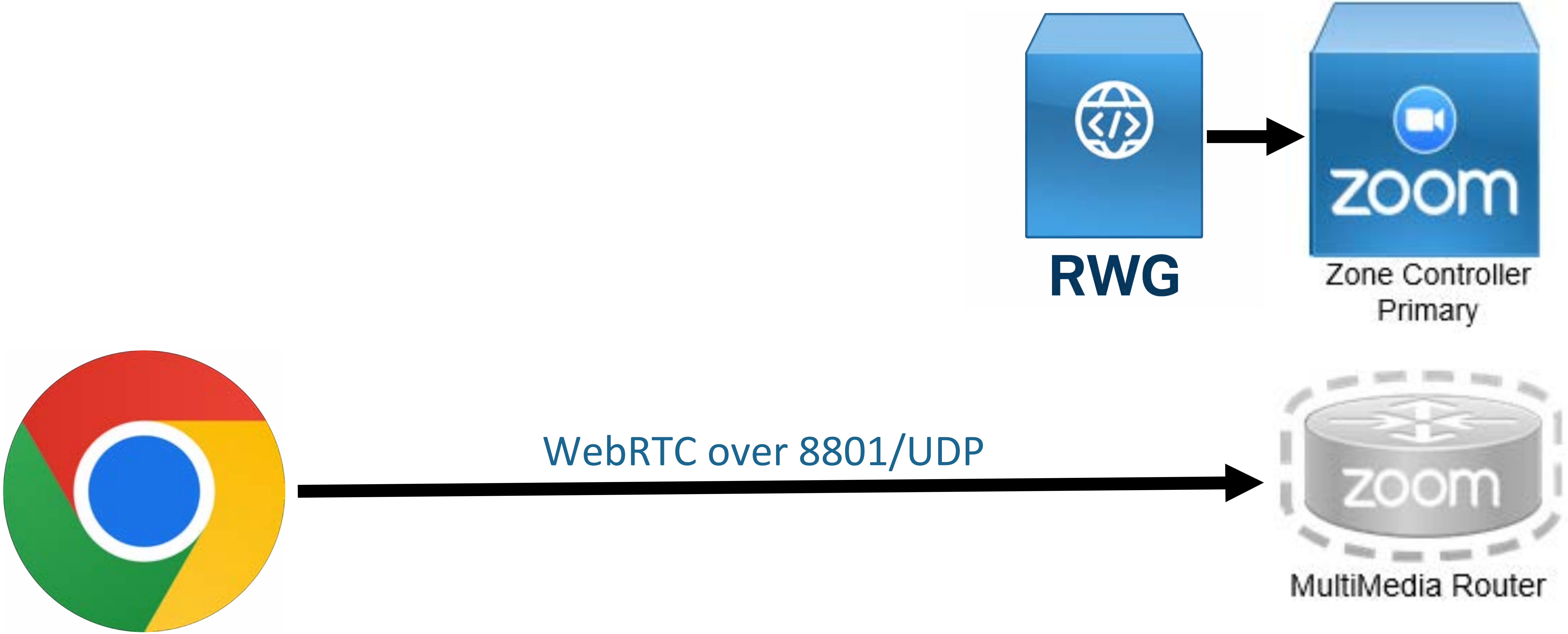
# Example Zoom Web Client Egress Attempts



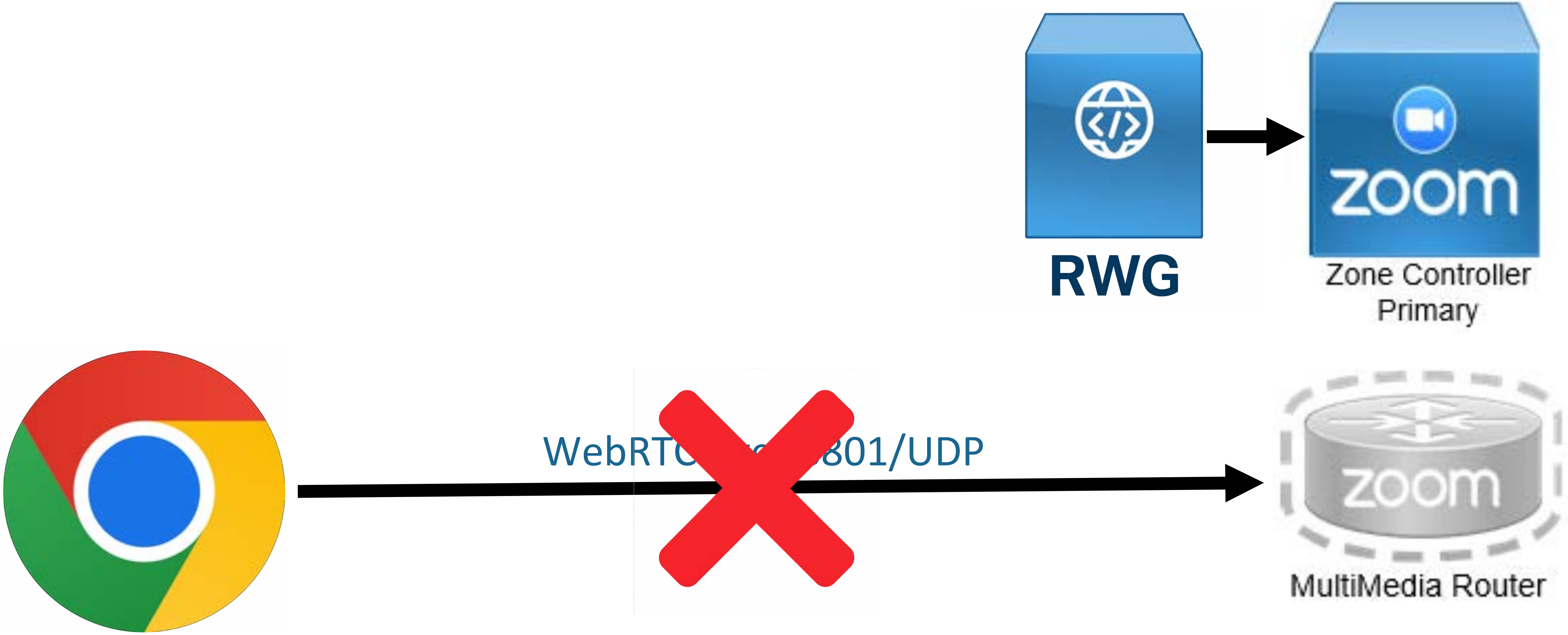
# Example Zoom Web Client Egress Attempts



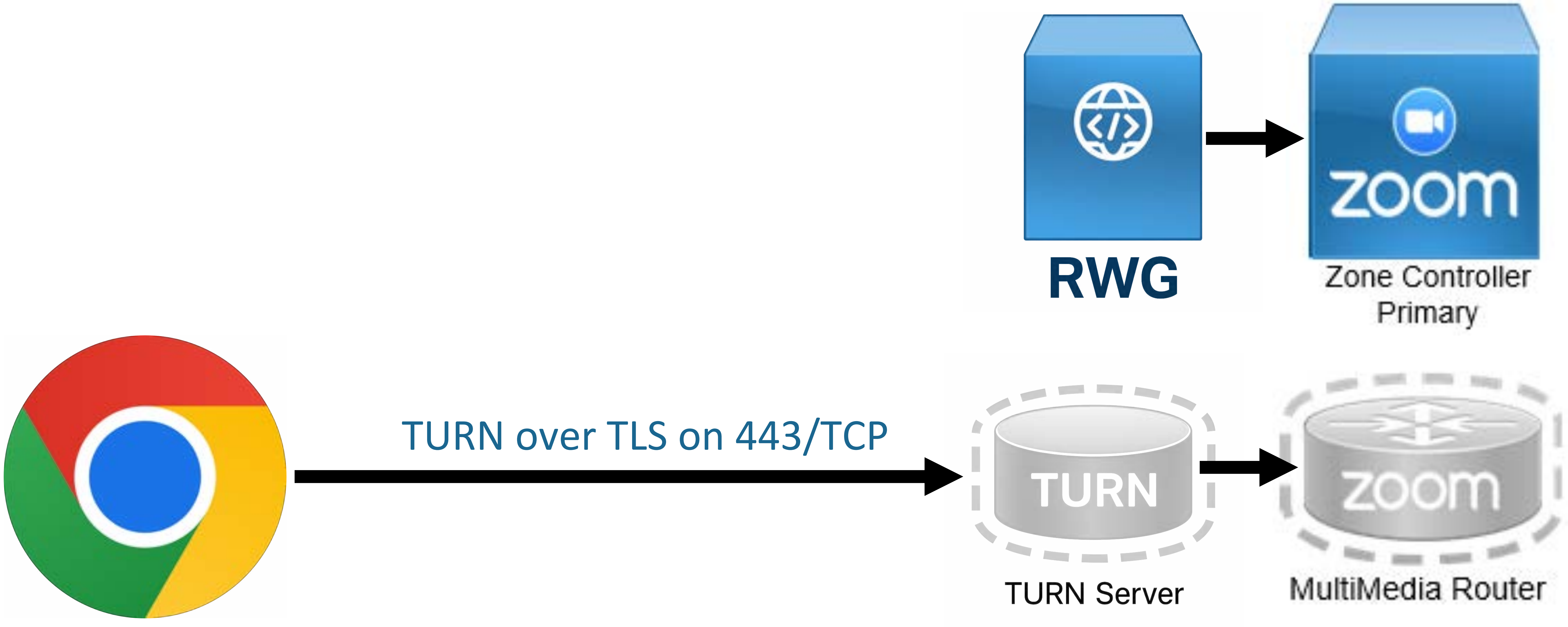
# Example Zoom Web Client Egress Attempts



# Example Zoom Web Client Egress Attempts

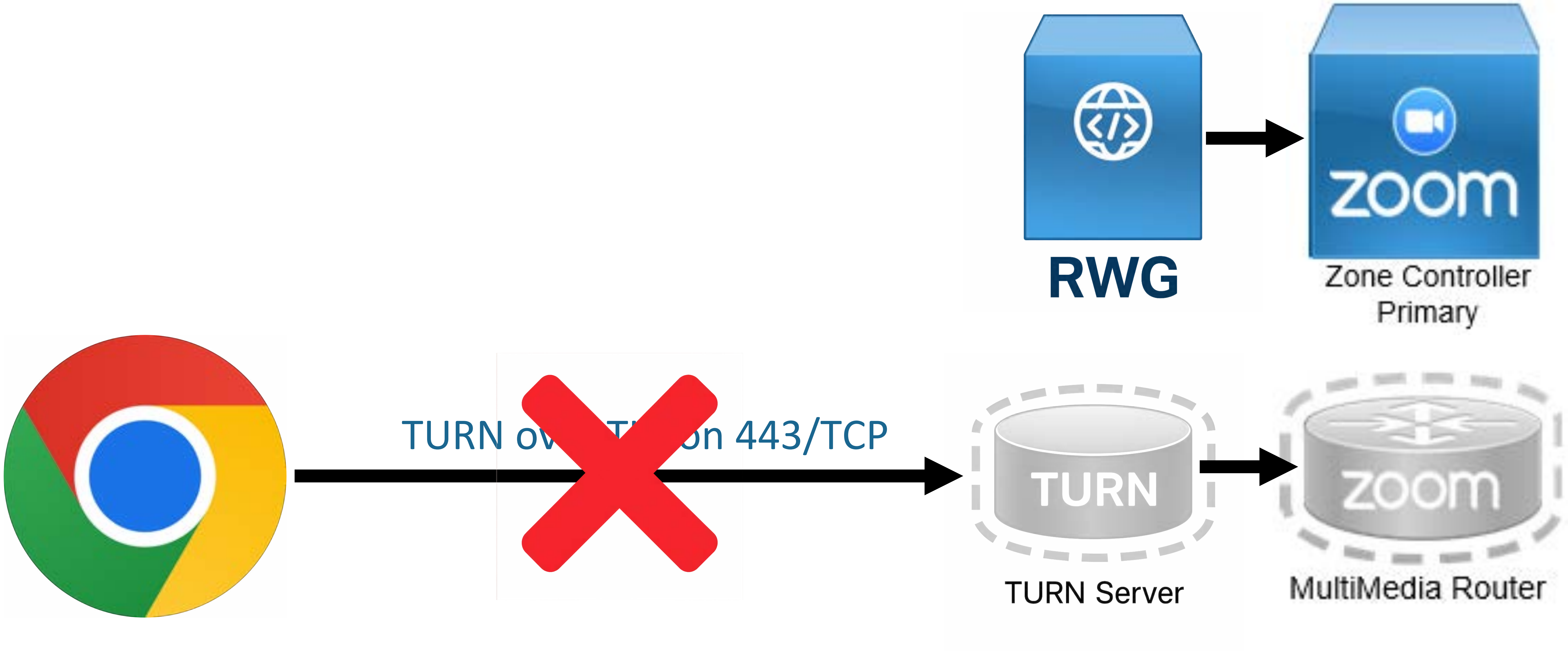


# Example Zoom Web Client Egress Attempts

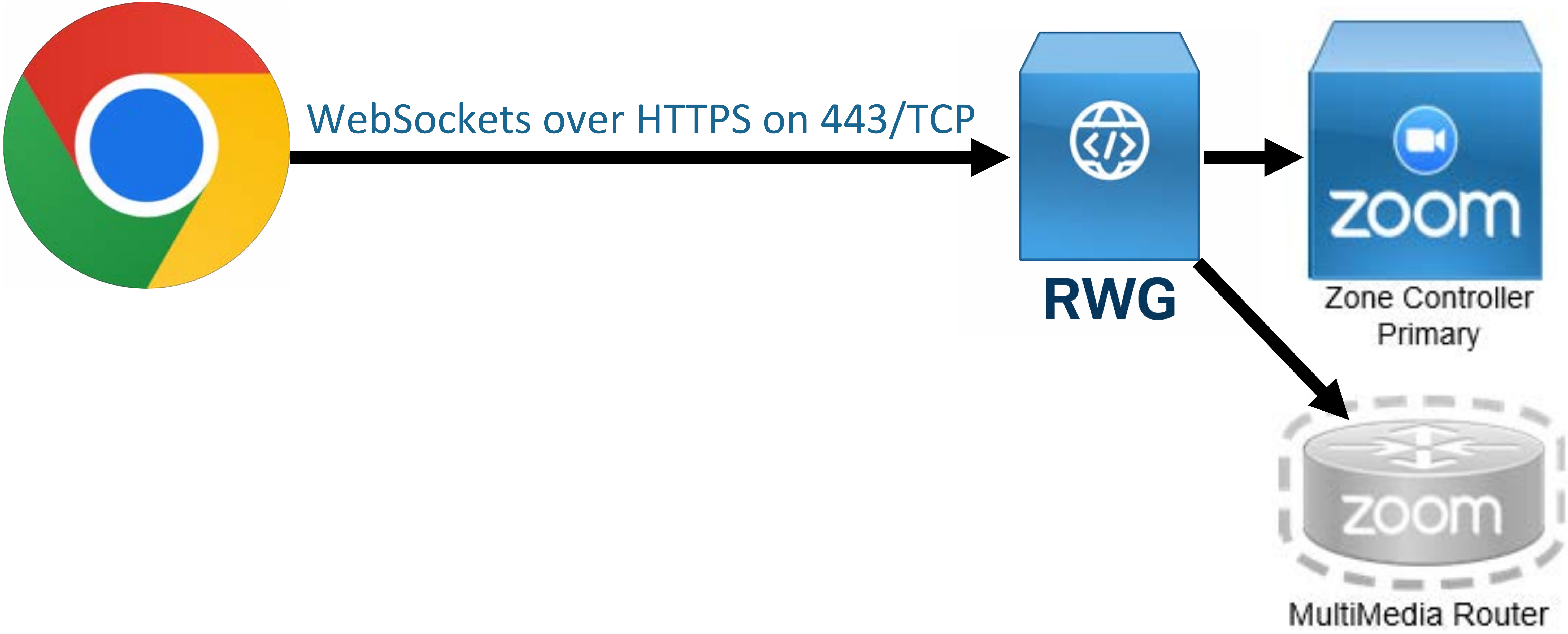


TURN over TLS on 443/TCP

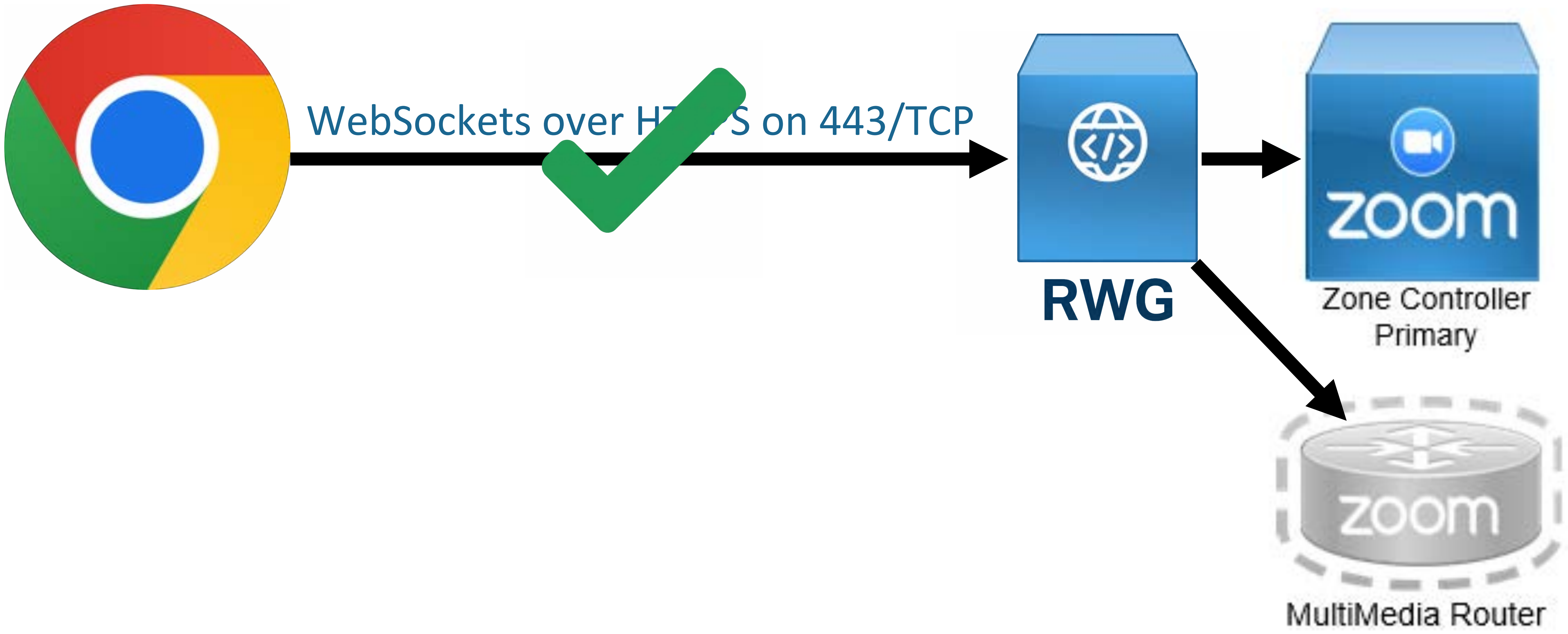
# Example Zoom Web Client Egress Attempts



# Example Zoom Web Client Egress Attempts



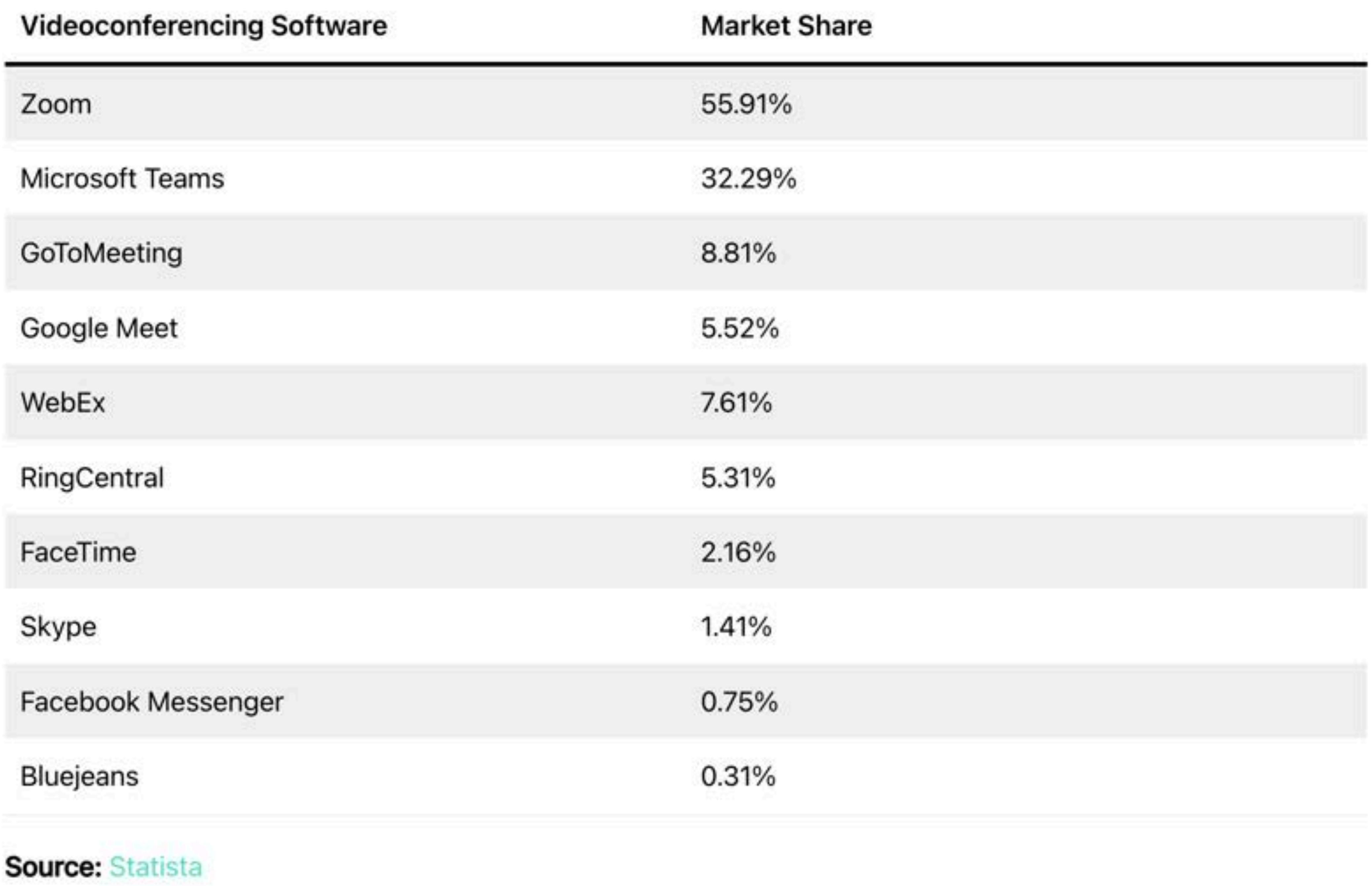
# Example Zoom Web Client Egress Attempts



# Highly Adaptable to Changing Environments

# Developing the Capability

# Analyzing Vendor Market Share

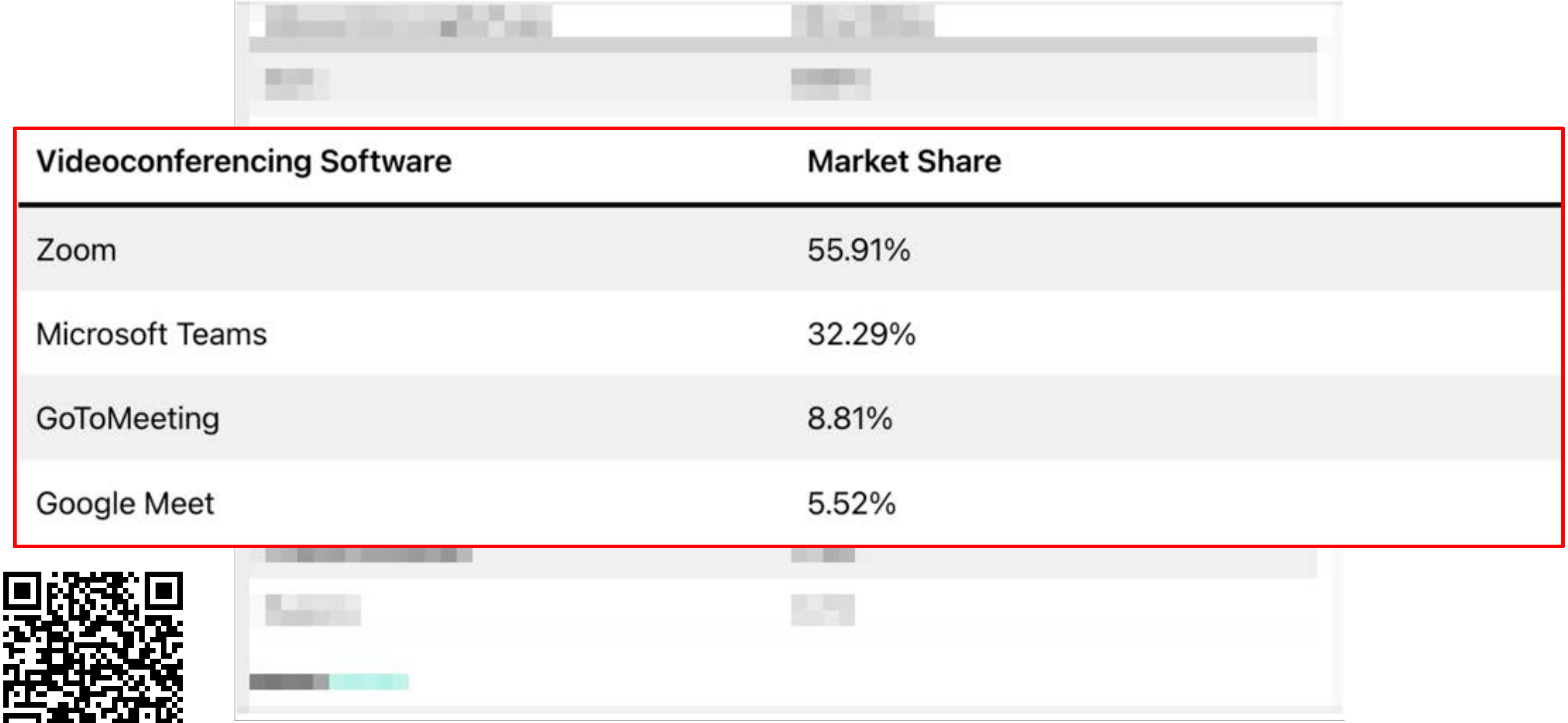


| Videoconferencing Software | Market Share |
|---|---|
| Zoom | 55.91% |
| Microsoft Teams | 32.29% |
| GoToMeeting | 8.81% |
| Google Meet | 5.52% |
| WebEx | 7.61% |
| RingCentral | 5.31% |
| FaceTime | 2.16% |
| Skype | 1.41% |
| Facebook Messenger | 0.75% |
| Bluejeans | 0.31% |

**Source:** Statista

https://www.demandsage.com/microsoft-teams-statistics/

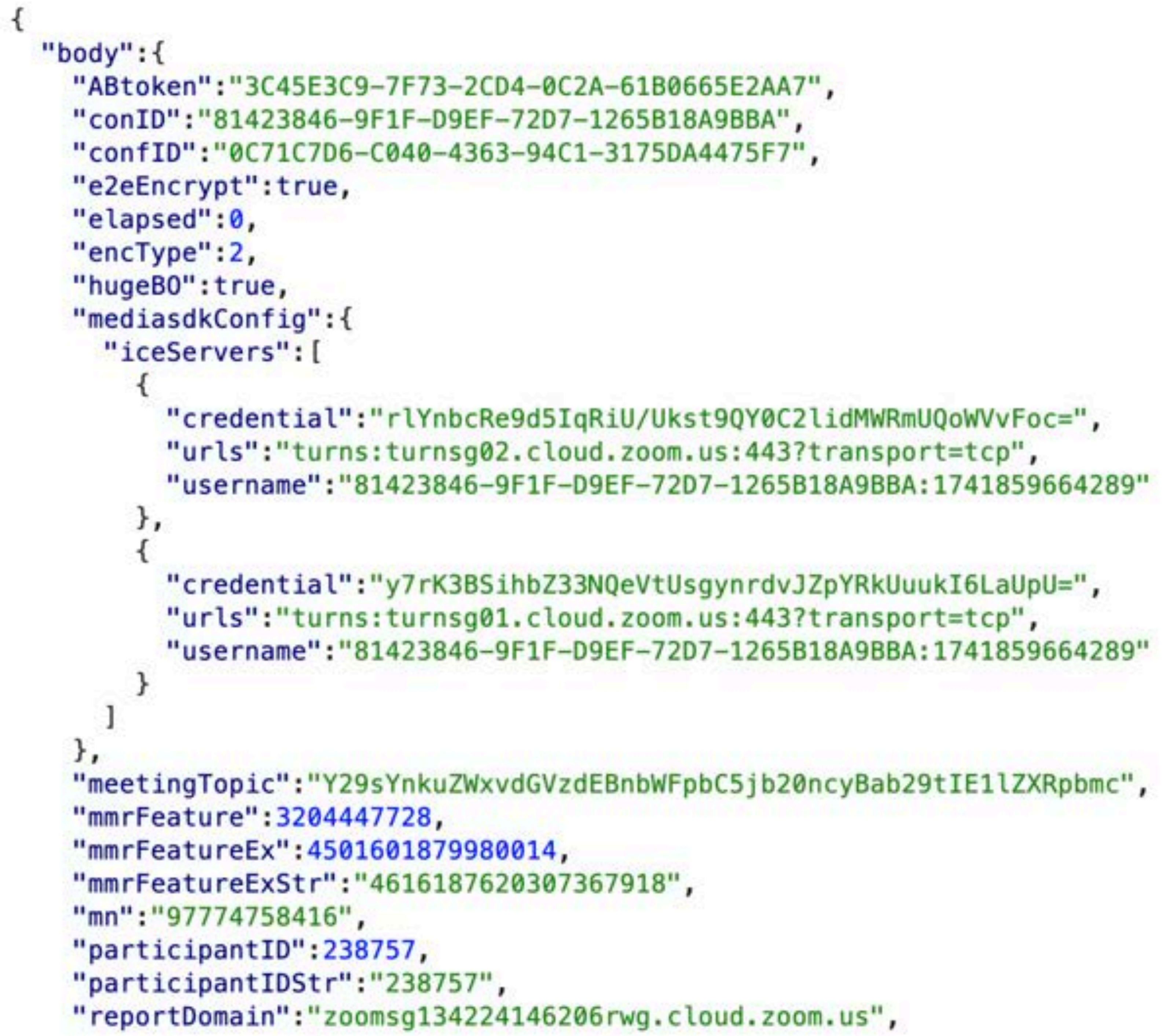
# Analyzing Vendor Market Share



| Videoconferencing Software | Market Share |
|---|---|
| Zoom | 55.91% |
| Microsoft Teams | 32.29% |
| GoToMeeting | 8.81% |
| Google Meet | 5.52% |

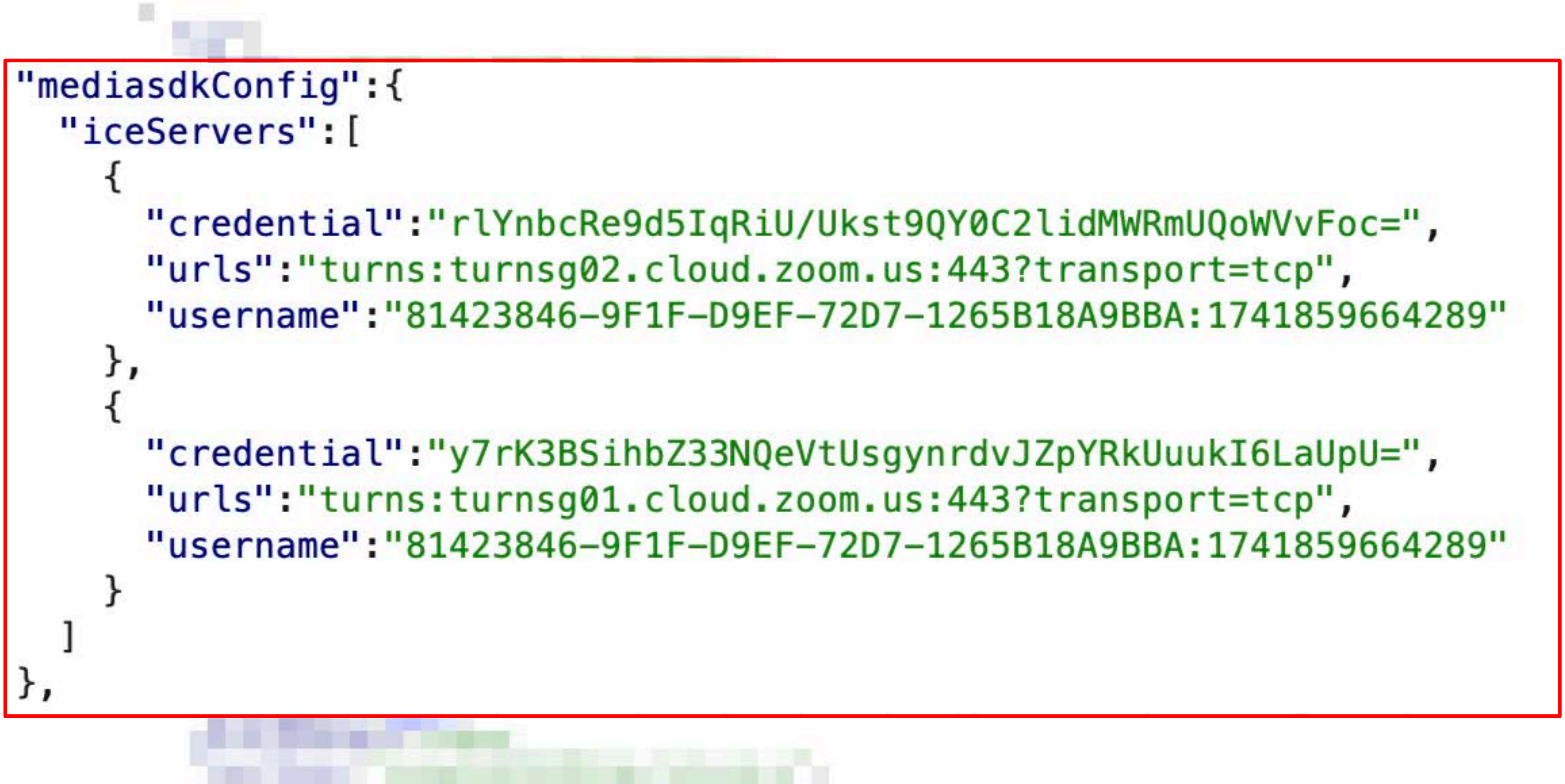
https://www.demandsage.com/microsoft-teams-statistics/

# Reverse Engineering Zoom



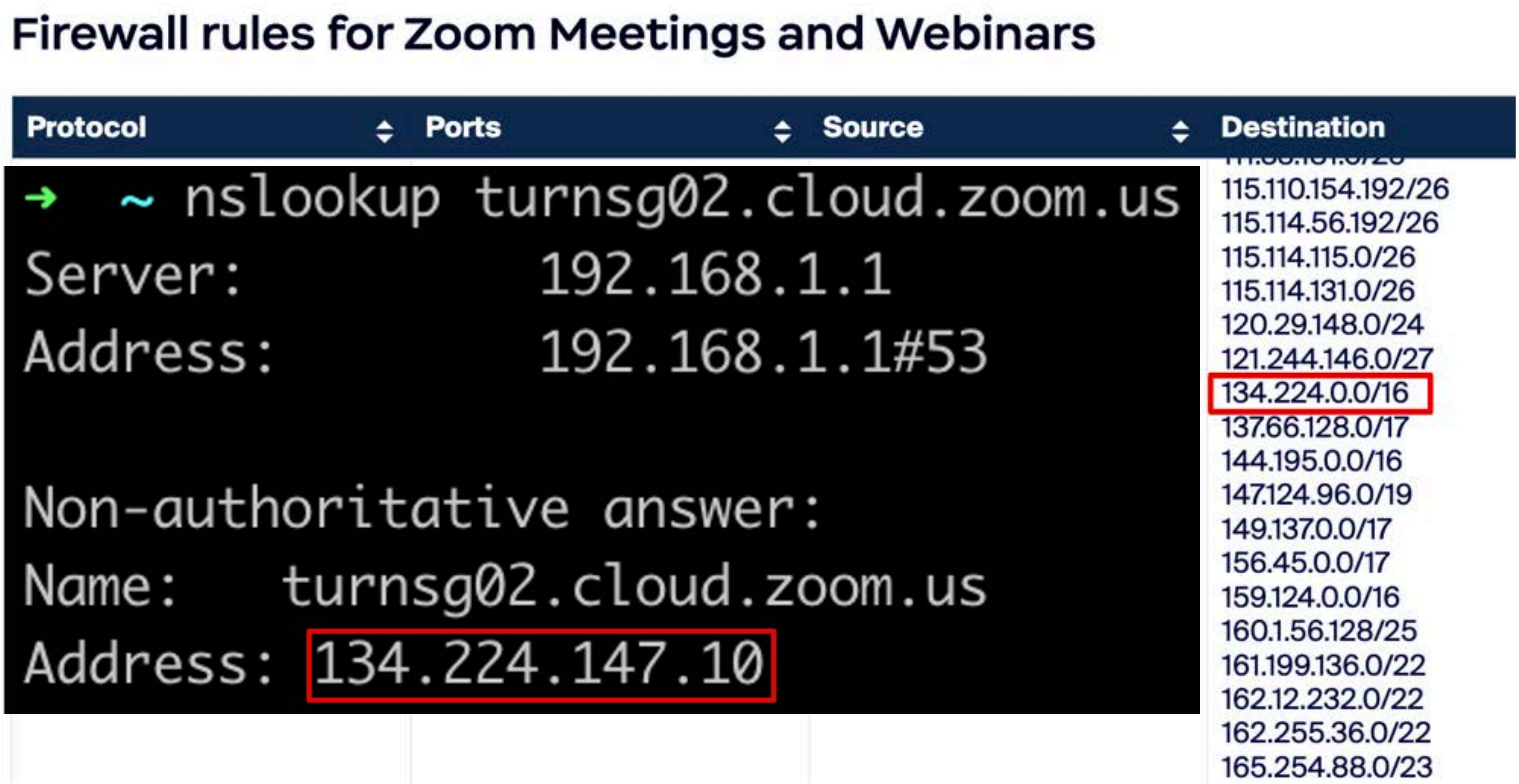
```
{
  "body":{
    "ABtoken":"3C45E3C9-7F73-2CD4-0C2A-61B0665E2AA7",
    "conID":"81423846-9F1F-D9EF-72D7-1265B18A9BBA",
    "confID":"0C71C7D6-C040-4363-94C1-3175DA4475F7",
    "e2eEncrypt":true,
    "elapsed":0,
    "encType":2,
    "hugeBO":true,
    "mediasdkConfig":{
      "iceServers":[
        {
          "credential":"rlYnbcRe9d5IqRiU/Ukst9QY0C2lidMWRmUQoWVvFoc=",
          "urls":"turns:turnsg02.cloud.zoom.us:443?transport=tcp",
          "username":"81423846-9F1F-D9EF-72D7-1265B18A9BBA:1741859664289"
        },
        {
          "credential":"y7rK3BSihbZ33NQeVtUsgynrdvJZpYRkUuukI6LaUpU=",
          "urls":"turns:turnsg01.cloud.zoom.us:443?transport=tcp",
          "username":"81423846-9F1F-D9EF-72D7-1265B18A9BBA:1741859664289"
        }
      ]
    },
    "meetingTopic":"Y29sYnkuZWxvdGVzdEBnbWFpbC5jb20ncyBab29tIE1lZXRpbmc",
    "mmrFeature":3204447728,
    "mmrFeatureEx":4501601879980014,
    "mmrFeatureExStr":"4616187620307367918",
    "mn":"97774758416",
    "participantID":238757,
    "participantIDStr":"238757",
    "reportDomain":"zoomsg134224146206rwg.cloud.zoom.us",
```

# Reverse Engineering Zoom



```
"mediasdkConfig":{
  "iceServers":[
    {
      "credential":"rlYnbcRe9d5IqRiU/Ukst9QY0C2lidMWRmUQoWVvFoc=",
      "urls":"turns:turnsg02.cloud.zoom.us:443?transport=tcp",
      "username":"81423846-9F1F-D9EF-72D7-1265B18A9BBA:1741859664289"
    },
    {
      "credential":"y7rK3BSihbZ33NQeVtUsgynrdvJZpYRkUuukI6LaUpU=",
      "urls":"turns:turnsg01.cloud.zoom.us:443?transport=tcp",
      "username":"81423846-9F1F-D9EF-72D7-1265B18A9BBA:1741859664289"
    }
  ]
},
```

# Zooming in on TURN



## Firewall rules for Zoom Meetings and Webinars

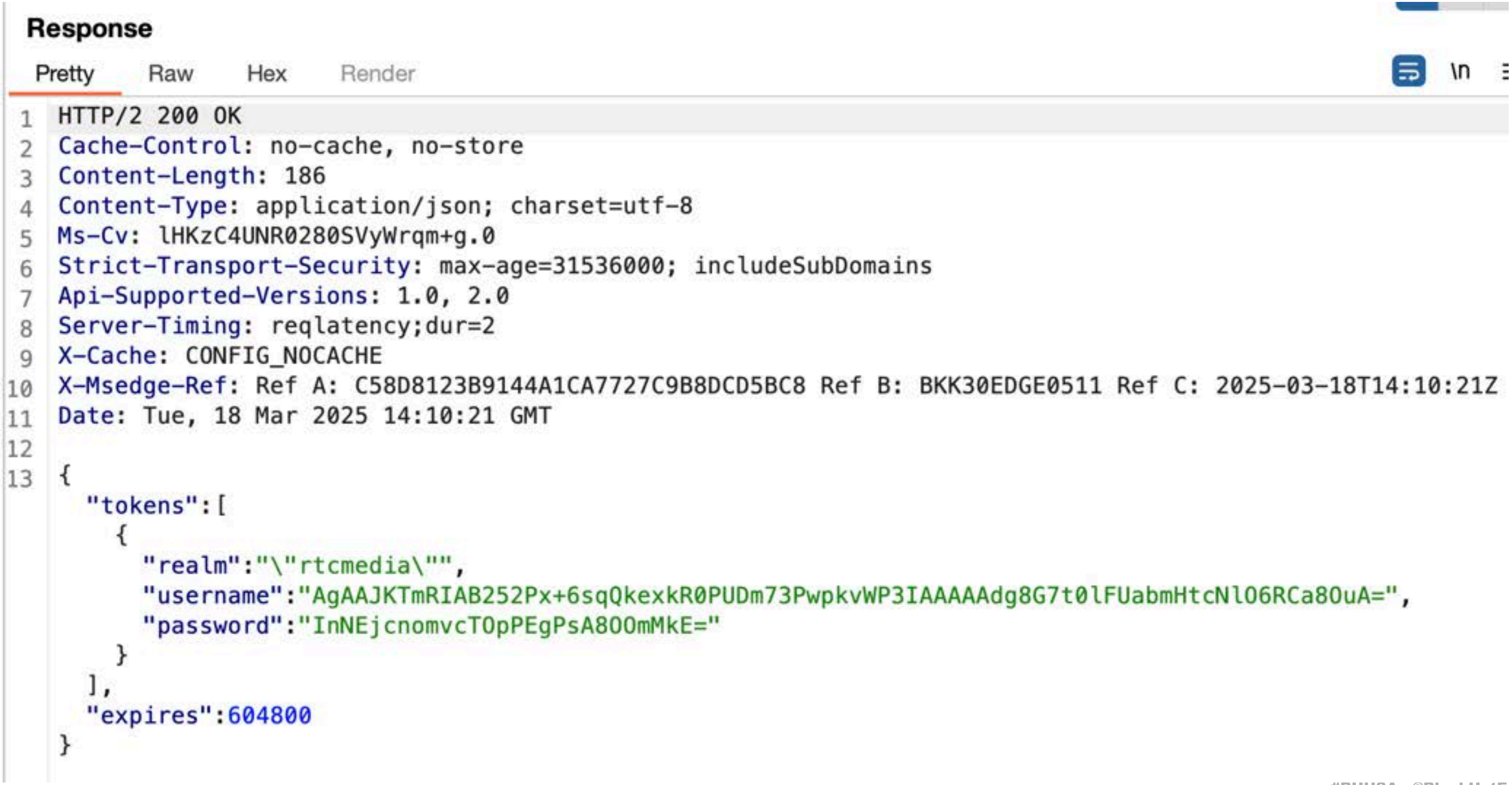
| Protocol | Ports | Source | Destination |
|---|---|---|---|
| | | | 115.110.154.192/26 |
| | | | 115.114.56.192/26 |
| | | | 115.114.115.0/26 |
| | | | 115.114.131.0/26 |
| | | | 120.29.148.0/24 |
| | | | 121.244.146.0/27 |
| | | | 134.224.0.0/16 |
| | | | 137.66.128.0/17 |
| | | | 144.195.0.0/16 |
| | | | 147.124.96.0/19 |
| | | | 149.137.0.0/17 |
| | | | 156.45.0.0/17 |
| | | | 159.124.0.0/16 |
| | | | 160.1.56.128/25 |
| | | | 161.199.136.0/22 |
| | | | 162.12.232.0/22 |
| | | | 162.255.36.0/22 |
| | | | 165.254.88.0/23 |

```
➜  ~ nslookup turnsg02.cloud.zoom.us
Server:         192.168.1.1
Address:        192.168.1.1#53

Non-authoritative answer:
Name:   turnsg02.cloud.zoom.us
Address: 134.224.147.10
```

https://support.zoom.com/hc/en/article?id=zm_kb&sysparm_article=KB0060548

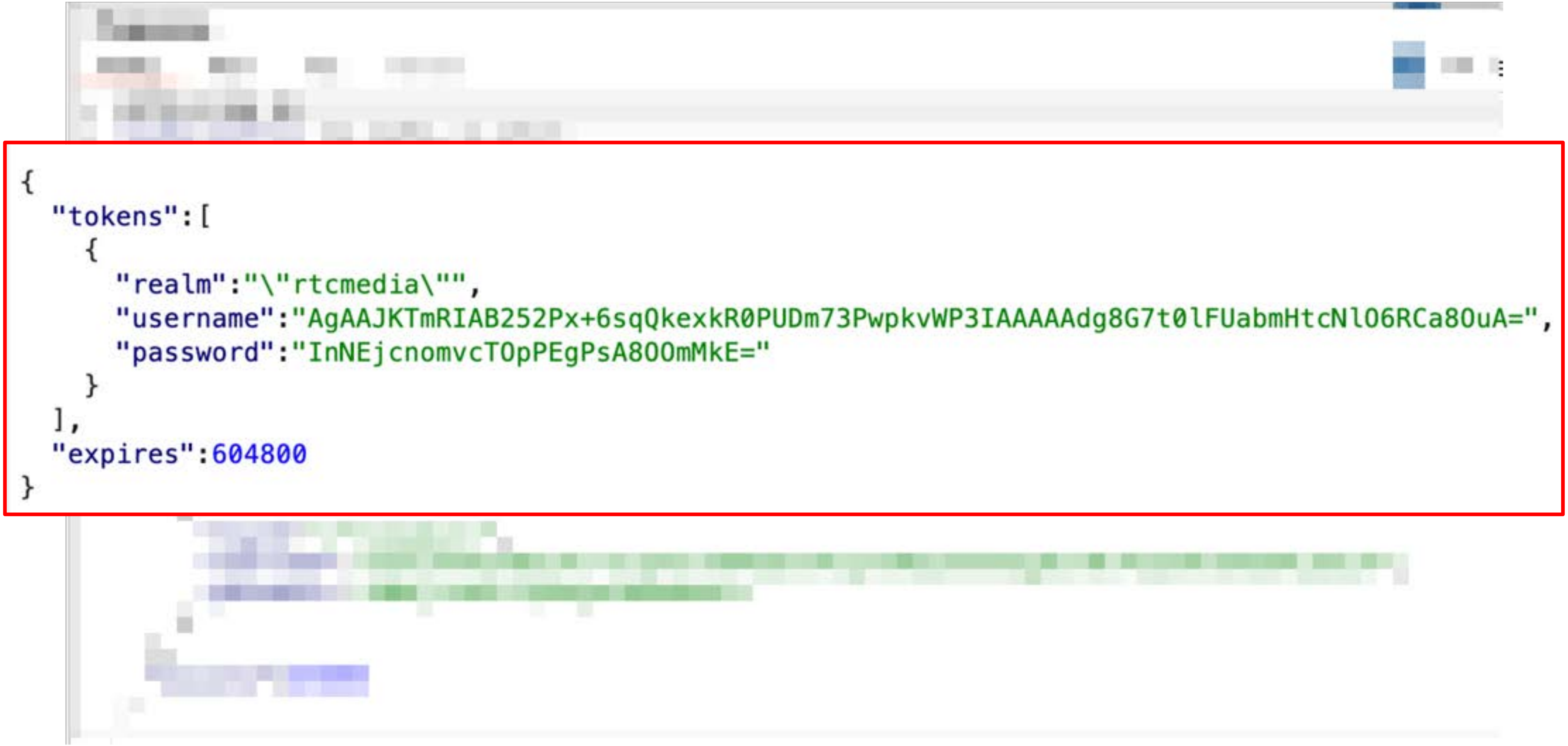
# Reverse Engineering Microsoft Teams



```
HTTP/2 200 OK
Cache-Control: no-cache, no-store
Content-Length: 186
Content-Type: application/json; charset=utf-8
Ms-Cv: lHKzC4UNR0280SVyWrqm+g.0
Strict-Transport-Security: max-age=31536000; includeSubDomains
Api-Supported-Versions: 1.0, 2.0
Server-Timing: reqlatency;dur=2
X-Cache: CONFIG_NOCACHE
X-Msedge-Ref: Ref A: C58D8123B9144A1CA7727C9B8DCD5BC8 Ref B: BKK30EDGE0511 Ref C: 2025-03-18T14:10:21Z
Date: Tue, 18 Mar 2025 14:10:21 GMT

{
  "tokens":[
    {
      "realm":"\"rtcmedia\"",
      "username":"AgAAJKTmRIAB252Px+6sqQkexkR0PUDm73PwpkvWP3IAAAAAdg8G7t0lFUabmHtcNl06RCa8OuA=",
      "password":"InNEjcnomvcTOpPEgPsA8OOmMkE="
    }
  ],
  "expires":604800
}
```

# Reverse Engineering Microsoft Teams

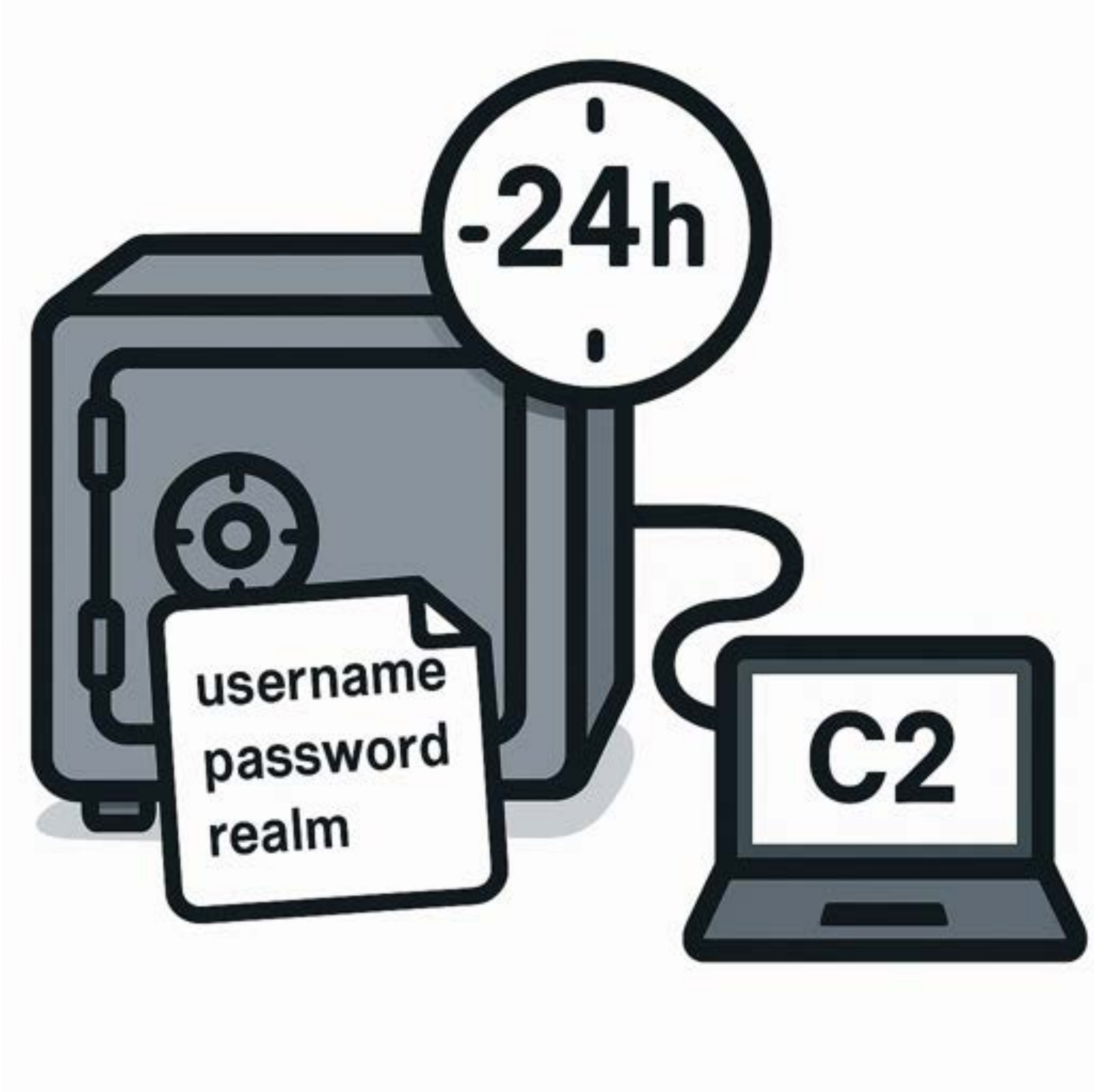


```
{
  "tokens":[
    {
      "realm":"\"rtcmedia\"",
      "username":"AgAAJKTmRIAB252Px+6sqQkexkR0PUDm73PwpkvWP3IAAAAdg8G7t0lFUabmHtcNlO6RCa8OuA=",
      "password":"InNEjcnomvcTOpPEgPsA8OOmMkE="
    }
  ],
  "expires":604800
}
```

[redacted]

# Observations Regarding TURN Credentials

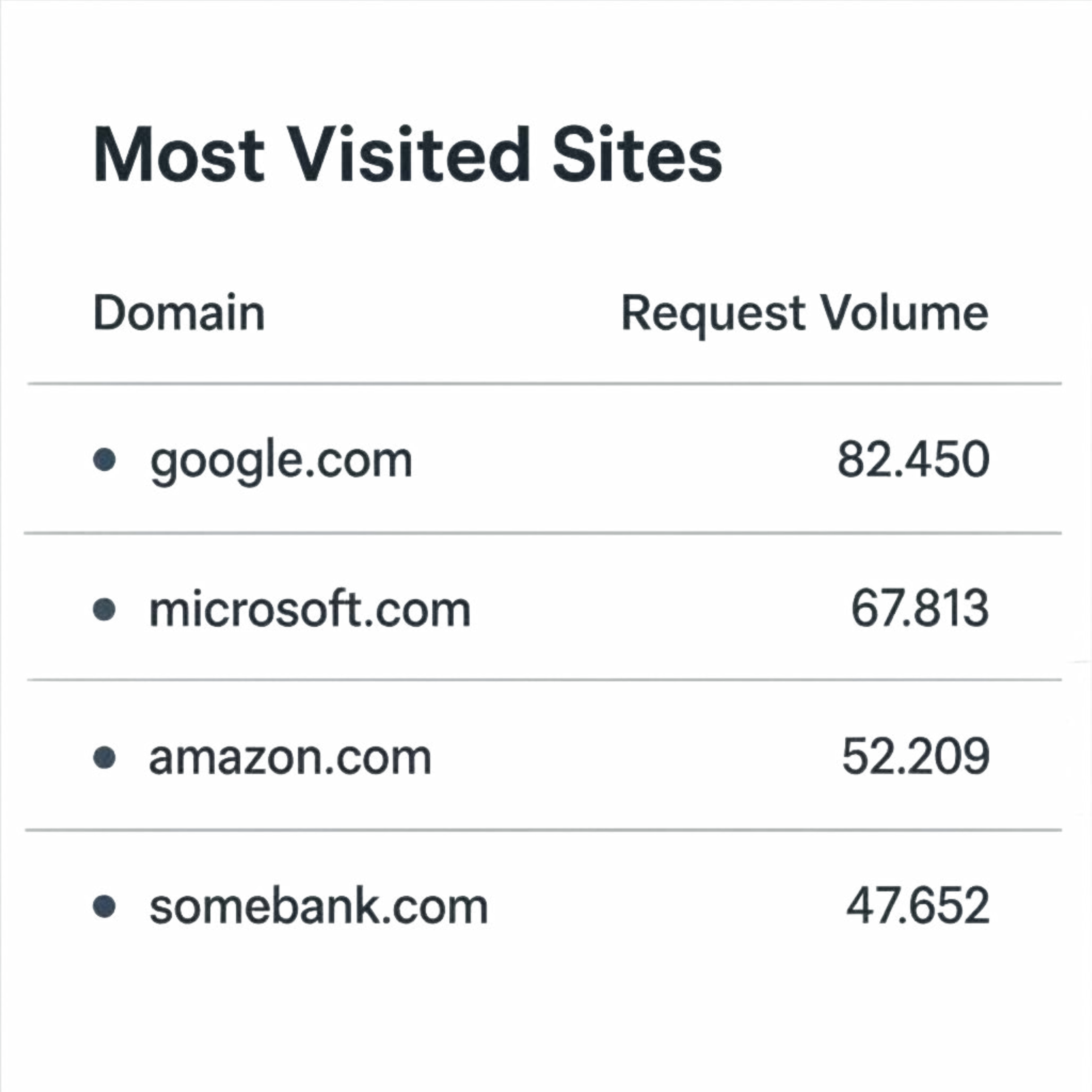
- Usually valid for a couple of days
- Complements an existing long-term channel
- Not tied to specific calls and credentials persist post-session
- Applies to common platforms like Zoom and Teams
- No install or meeting required on the victim side



# Building the Tool

# What do we want to build?

- A short-lived tunnel launched from an existing implant
- Used briefly and mimics activity like a video call
- Runs in parallel with long-term infrastructure
- Lightweight enough to avoid clogging that primary channel
- Disguised among high-traffic destinations (e.g., Zoom, Teams)



## Most Visited Sites

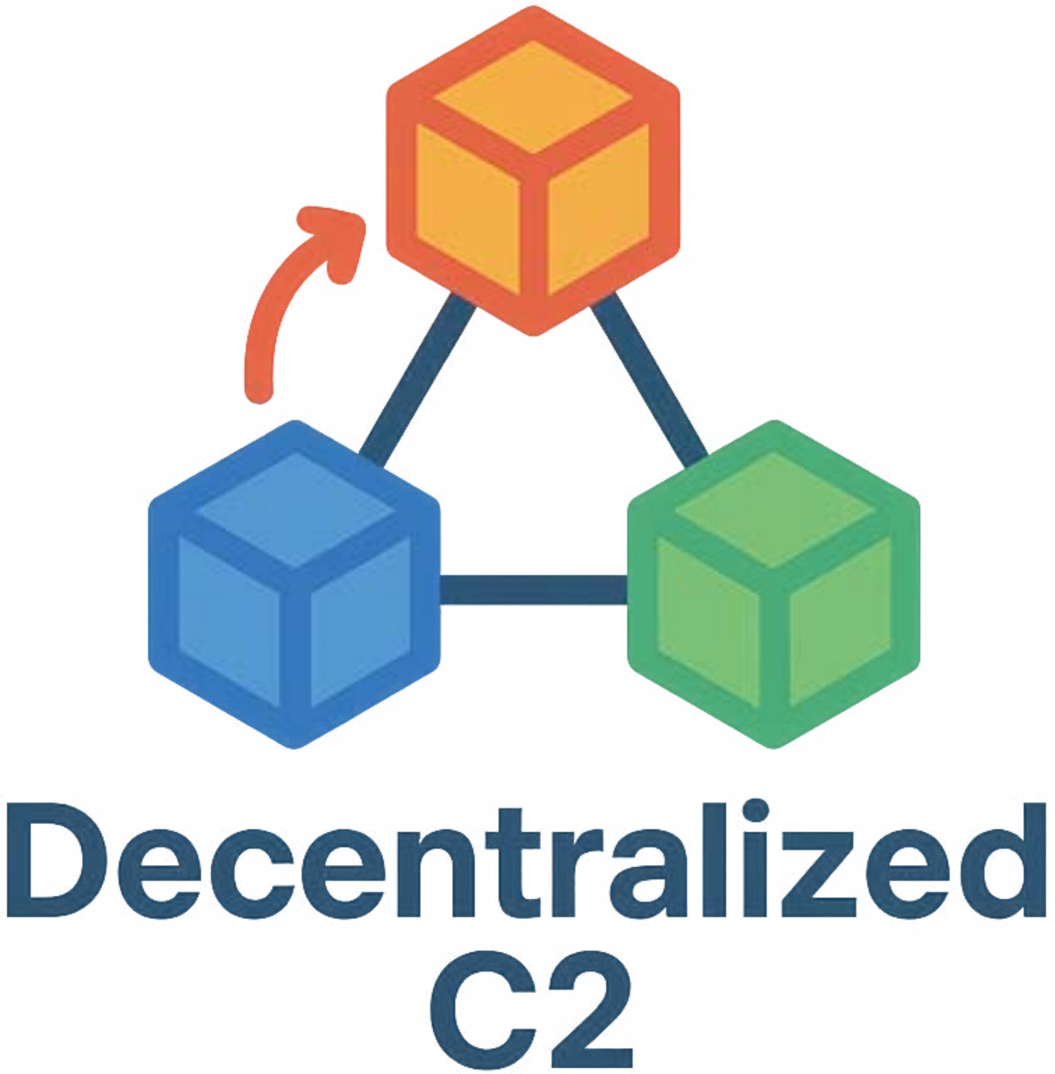
| Domain | Request Volume |
|---|---|
| ● google.com | 82.450 |
| ● microsoft.com | 67.813 |
| ● amazon.com | 52.209 |
| ● somebank.com | 47.652 |

# TURNt (TURN tunneler)

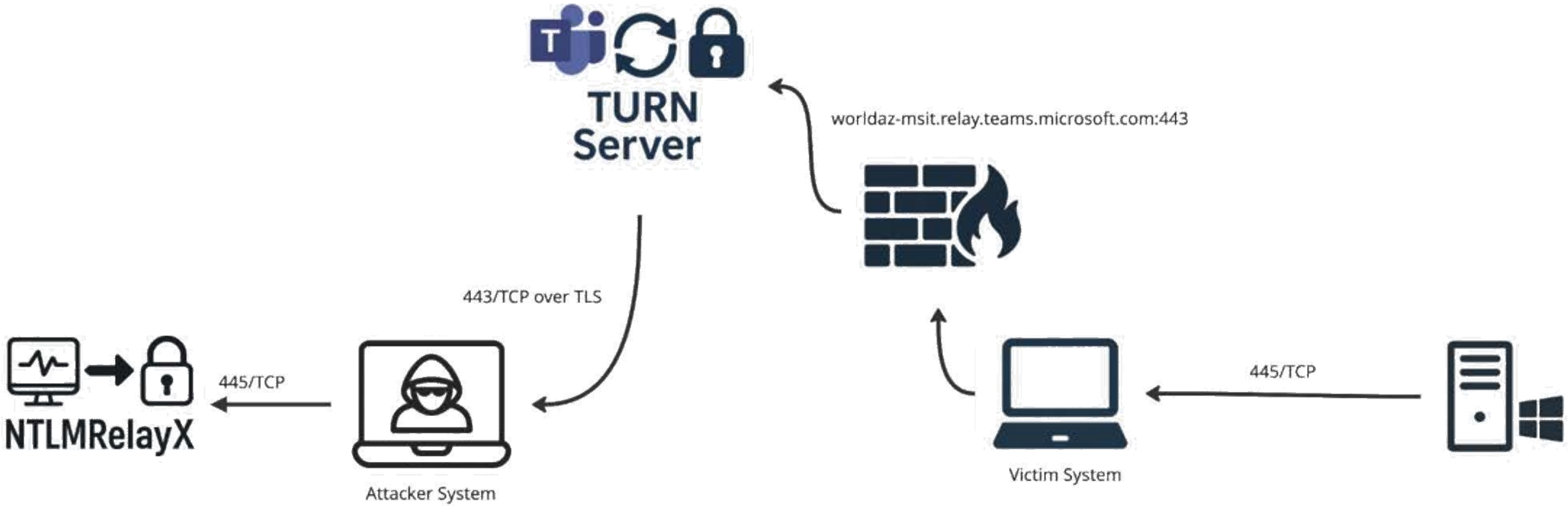
https://github.com/praetorian-inc/turnt

# Use-Cases and Capabilities

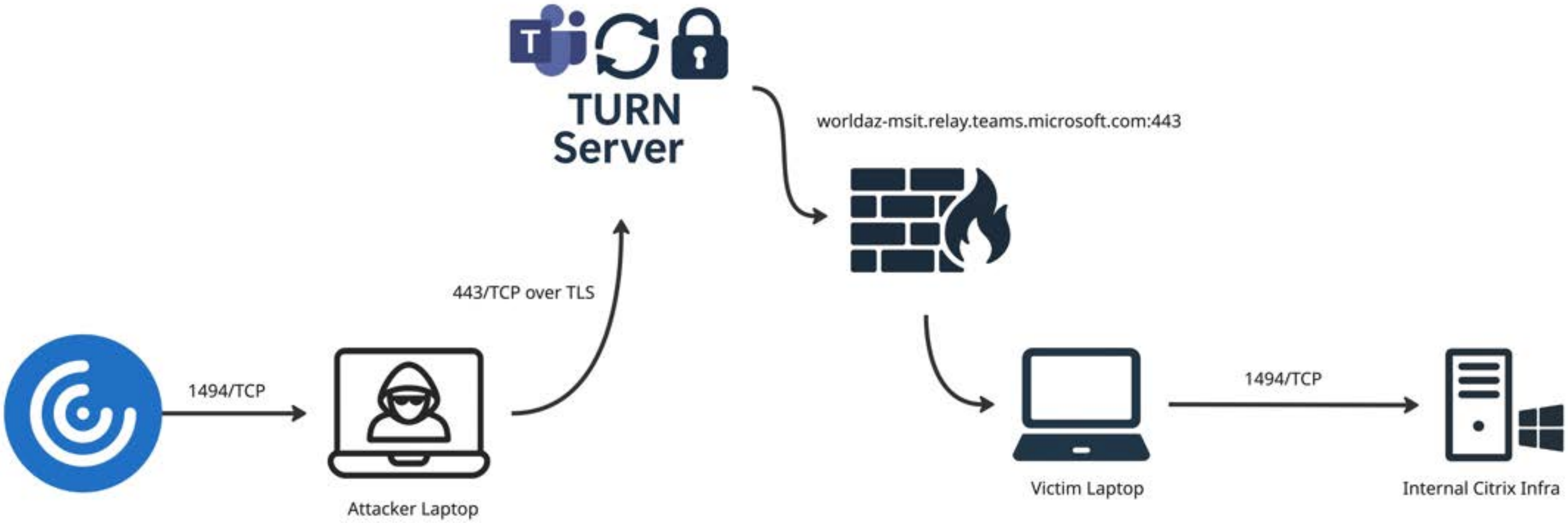
- Fast tunnel setup during assumed breach scenarios
- No need to provision infrastructure in advance
- Operates from operator laptop or disposable VDI
- Ideal for decentralized red team operations
- Lightweight, flexible, and serverless by design



# Remote Port Forwarding

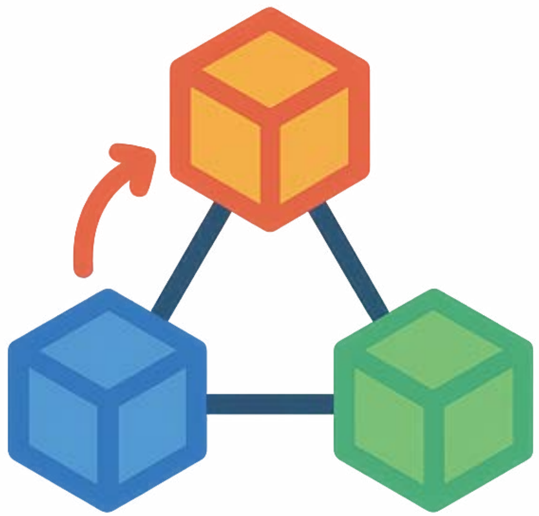


# Local Port Forwarding

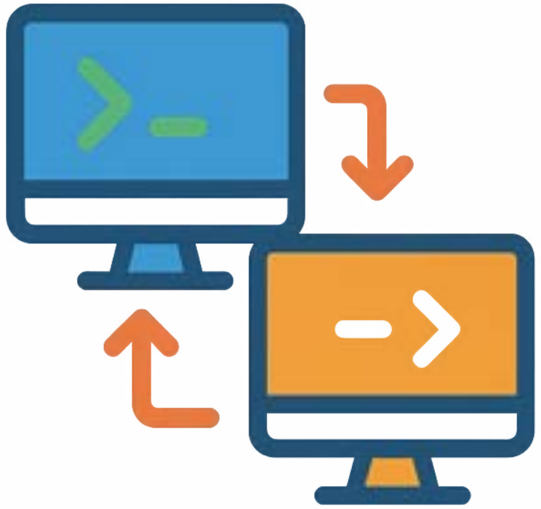


# Use-Cases and Capabilities

SOCKS Proxying



Decentralized C2

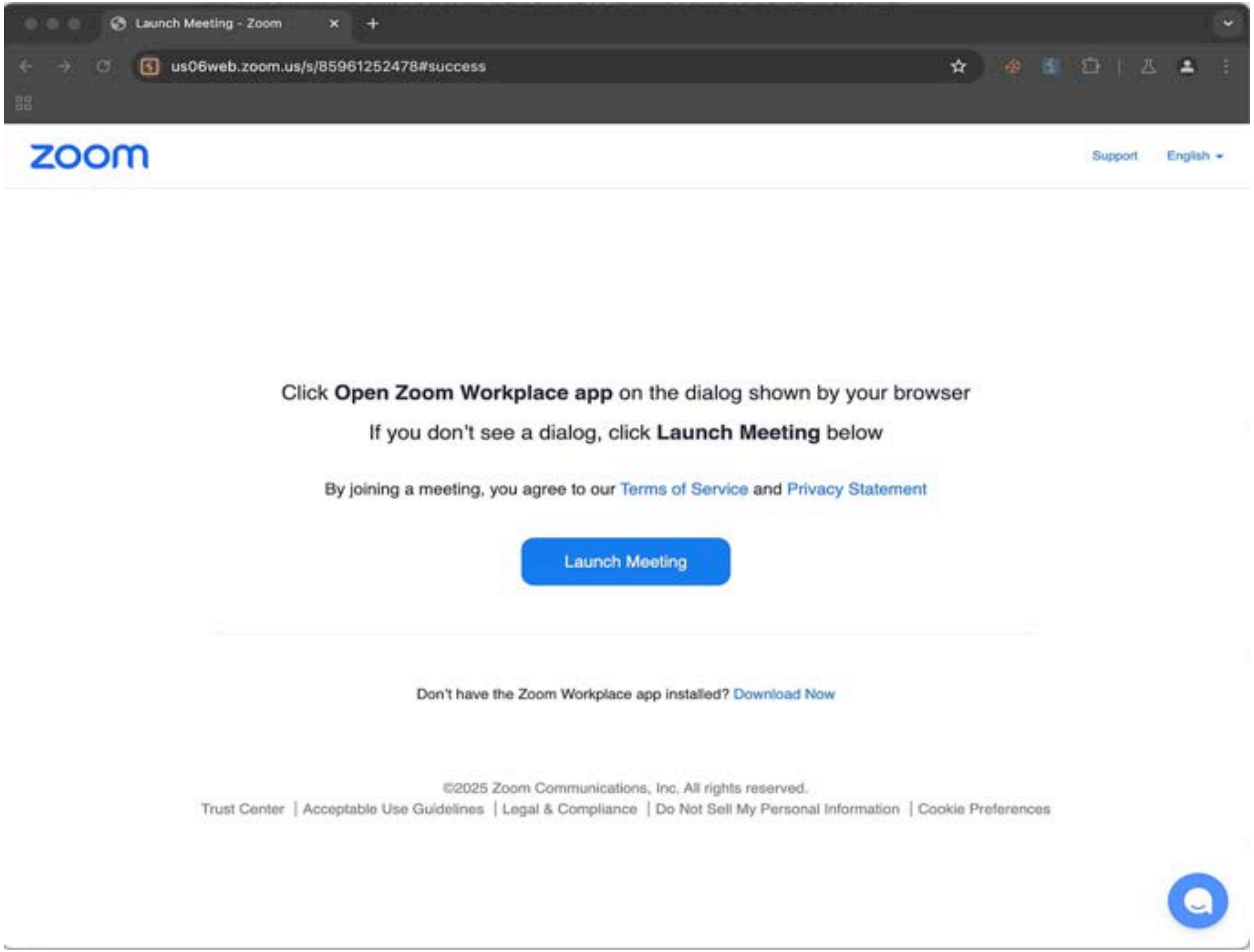


Local & Remote Port-Forwarding

# Zoom Demo Example Scenario

- Obtaining credentials from Zoom
- Victim doesn’t need to do anything
- Laptop is the operator laptop
- Example victim system is GCP virtual machine
- Demo downloading file through the channel

# Zoom Video Demo



Click **Open Zoom Workplace app** on the dialog shown by your browser

If you don't see a dialog, click **Launch Meeting** below

By joining a meeting, you agree to our Terms of Service and Privacy Statement

Launch Meeting

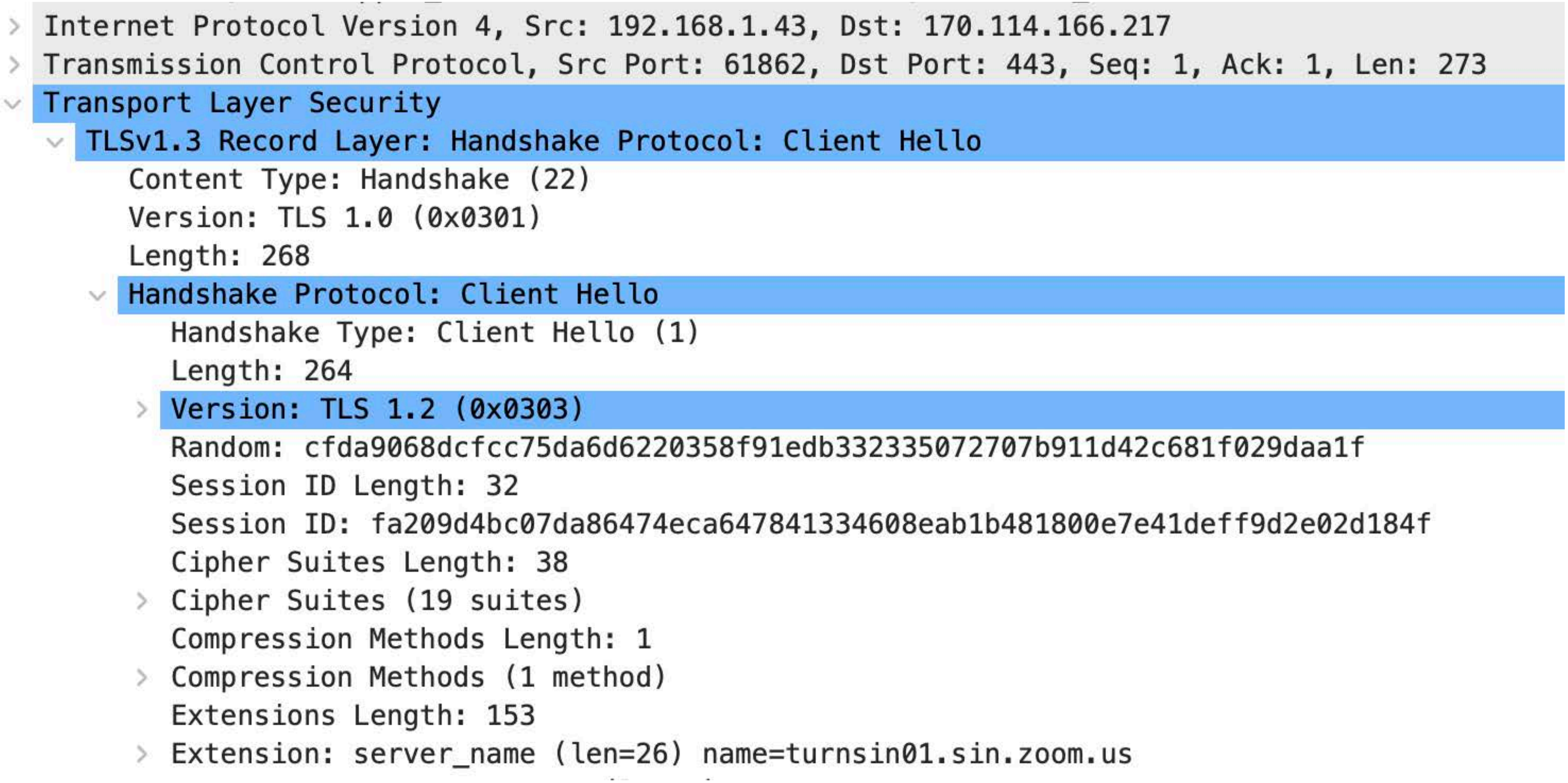
Don't have the Zoom Workplace app installed? Download Now

©2025 Zoom Communications, Inc. All rights reserved.

Trust Center | Acceptable Use Guidelines | Legal & Compliance | Do Not Sell My Personal Information | Cookie Preferences

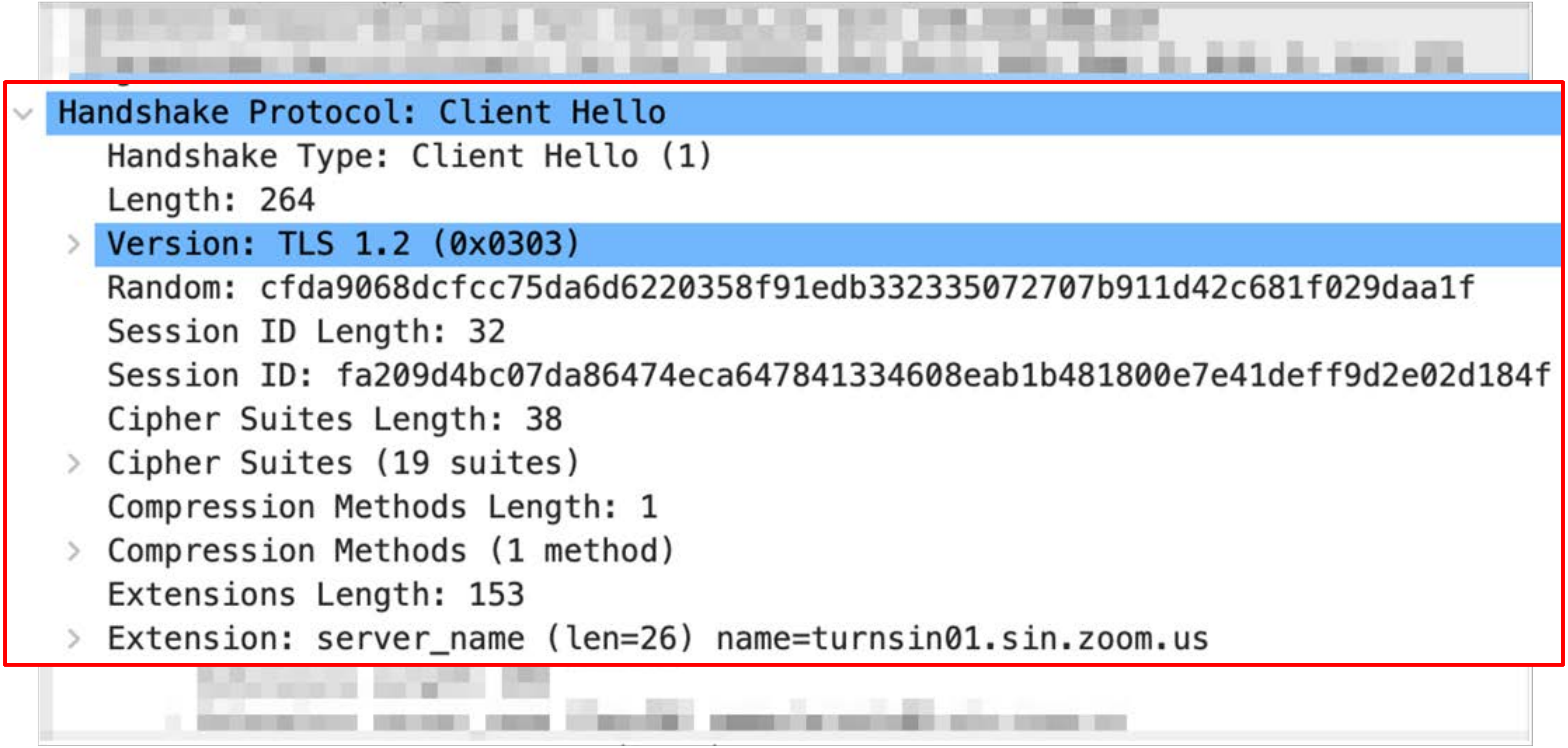
demo video

# Examining Wireshark Traffic (Zoom)



```
> Internet Protocol Version 4, Src: 192.168.1.43, Dst: 170.114.166.217
> Transmission Control Protocol, Src Port: 61862, Dst Port: 443, Seq: 1, Ack: 1, Len: 273
v Transport Layer Security
  v TLSv1.3 Record Layer: Handshake Protocol: Client Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 268
    v Handshake Protocol: Client Hello
        Handshake Type: Client Hello (1)
        Length: 264
      > Version: TLS 1.2 (0x0303)
        Random: cfda9068dcfcc75da6d6220358f91edb332335072707b911d42c681f029daa1f
        Session ID Length: 32
        Session ID: fa209d4bc07da86474eca647841334608eab1b481800e7e41deff9d2e02d184f
        Cipher Suites Length: 38
      > Cipher Suites (19 suites)
        Compression Methods Length: 1
      > Compression Methods (1 method)
        Extensions Length: 153
      > Extension: server_name (len=26) name=turnsin01.sin.zoom.us
```

# Examining Wireshark Traffic



[redacted]

```
⌄ Handshake Protocol: Client Hello
      Handshake Type: Client Hello (1)
      Length: 264
    › Version: TLS 1.2 (0x0303)
      Random: cfda9068dcfcc75da6d6220358f91edb332335072707b911d42c681f029daa1f
      Session ID Length: 32
      Session ID: fa209d4bc07da86474eca647841334608eab1b481800e7e41deff9d2e02d184f
      Cipher Suites Length: 38
    › Cipher Suites (19 suites)
      Compression Methods Length: 1
    › Compression Methods (1 method)
      Extensions Length: 153
    › Extension: server_name (len=26) name=turnsin01.sin.zoom.us
```

[redacted]

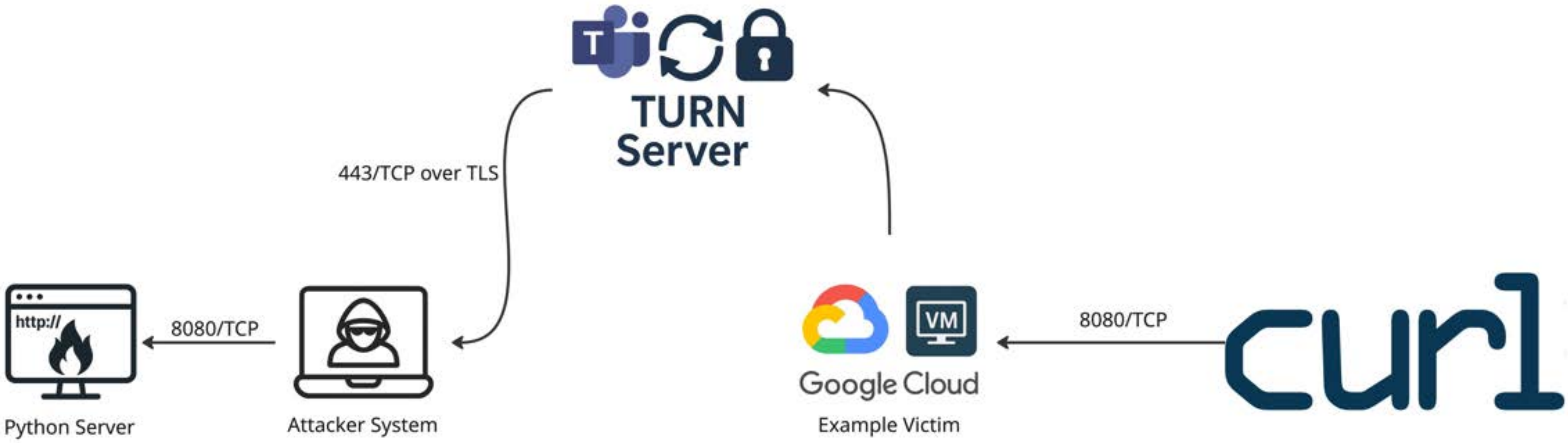
# Microsoft Teams Demo

- Show automated retrieval of TURN credentials from Microsoft
- Demonstrate a speed test showing a 100 MB file download
- Demonstrate remote port-forwarding capability
- Lab uses my local laptop and a demo virtual machine in GCP

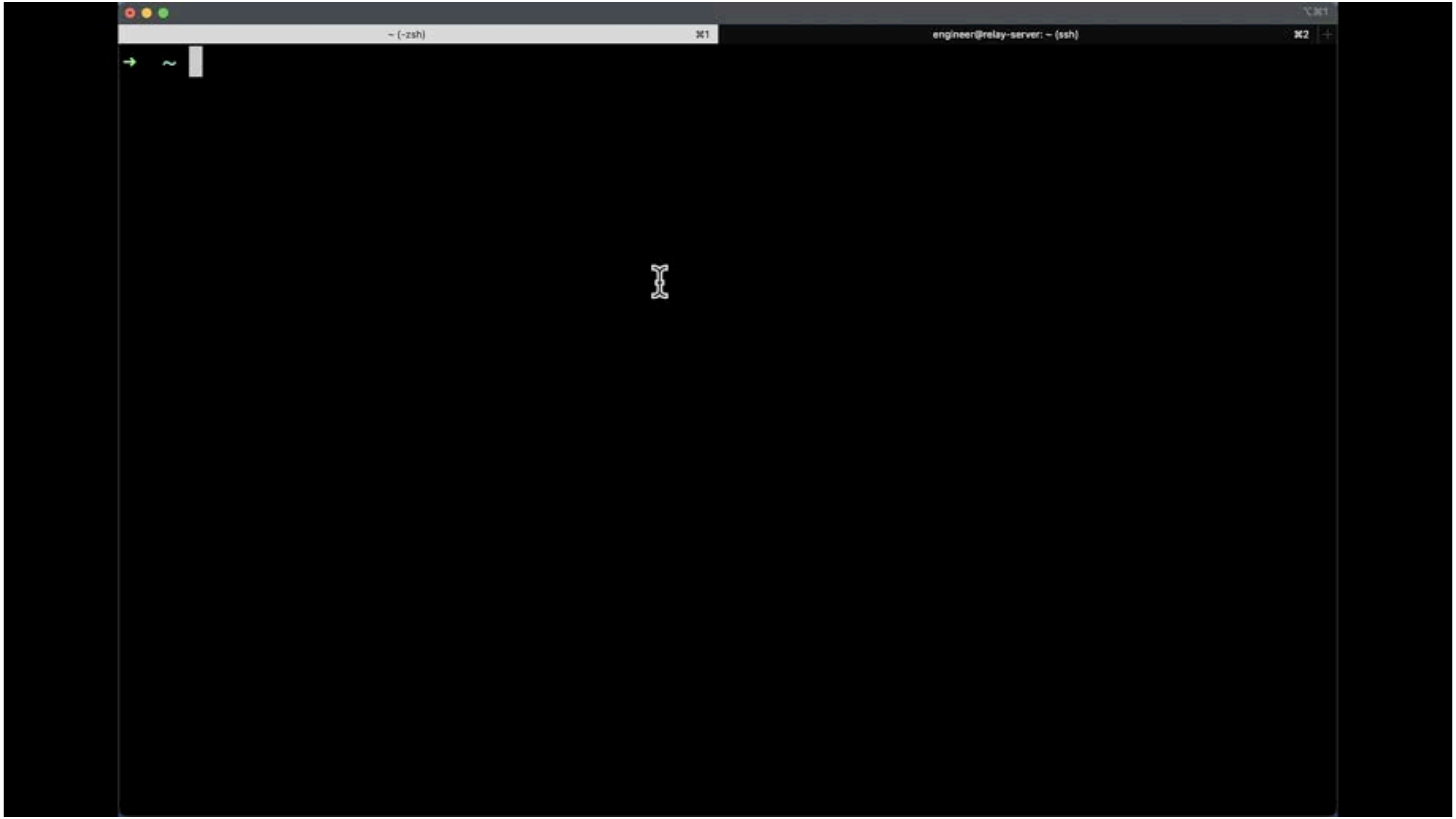
TURNt Credentials

TURNt Admin

# Microsoft Teams Demo



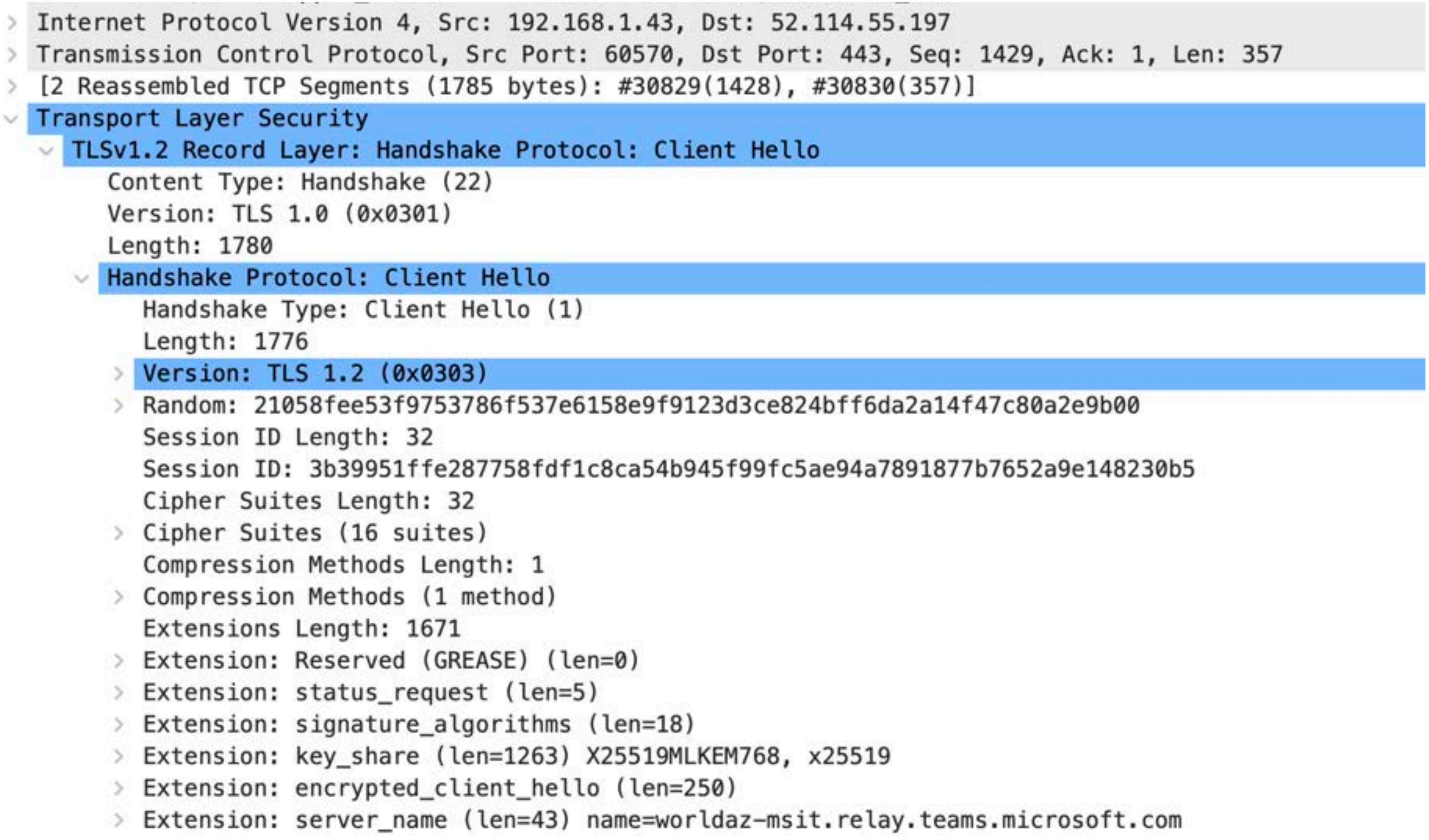
# Microsoft Teams Video Demo



```
➜  ~
```

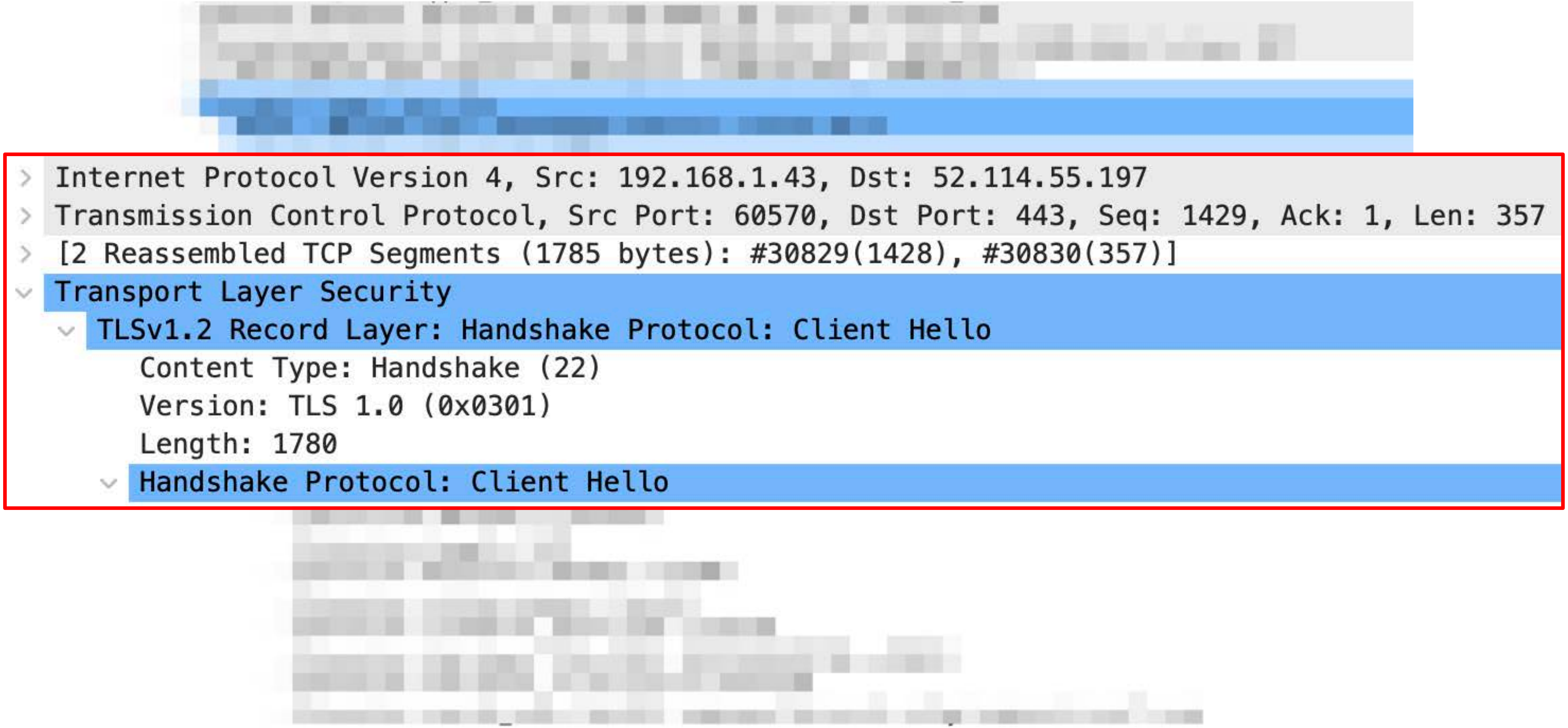
demo video

# Examining Wireshark Traffic



```
> Internet Protocol Version 4, Src: 192.168.1.43, Dst: 52.114.55.197
> Transmission Control Protocol, Src Port: 60570, Dst Port: 443, Seq: 1429, Ack: 1, Len: 357
> [2 Reassembled TCP Segments (1785 bytes): #30829(1428), #30830(357)]
v Transport Layer Security
  v TLSv1.2 Record Layer: Handshake Protocol: Client Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 1780
    v Handshake Protocol: Client Hello
        Handshake Type: Client Hello (1)
        Length: 1776
      > Version: TLS 1.2 (0x0303)
      > Random: 21058fee53f9753786f537e6158e9f9123d3ce824bff6da2a14f47c80a2e9b00
        Session ID Length: 32
        Session ID: 3b39951ffe287758fdf1c8ca54b945f99fc5ae94a7891877b7652a9e148230b5
        Cipher Suites Length: 32
      > Cipher Suites (16 suites)
        Compression Methods Length: 1
      > Compression Methods (1 method)
        Extensions Length: 1671
      > Extension: Reserved (GREASE) (len=0)
      > Extension: status_request (len=5)
      > Extension: signature_algorithms (len=18)
      > Extension: key_share (len=1263) X25519MLKEM768, x25519
      > Extension: encrypted_client_hello (len=250)
      > Extension: server_name (len=43) name=worldaz-msit.relay.teams.microsoft.com
```

# Examining Wireshark Traffic

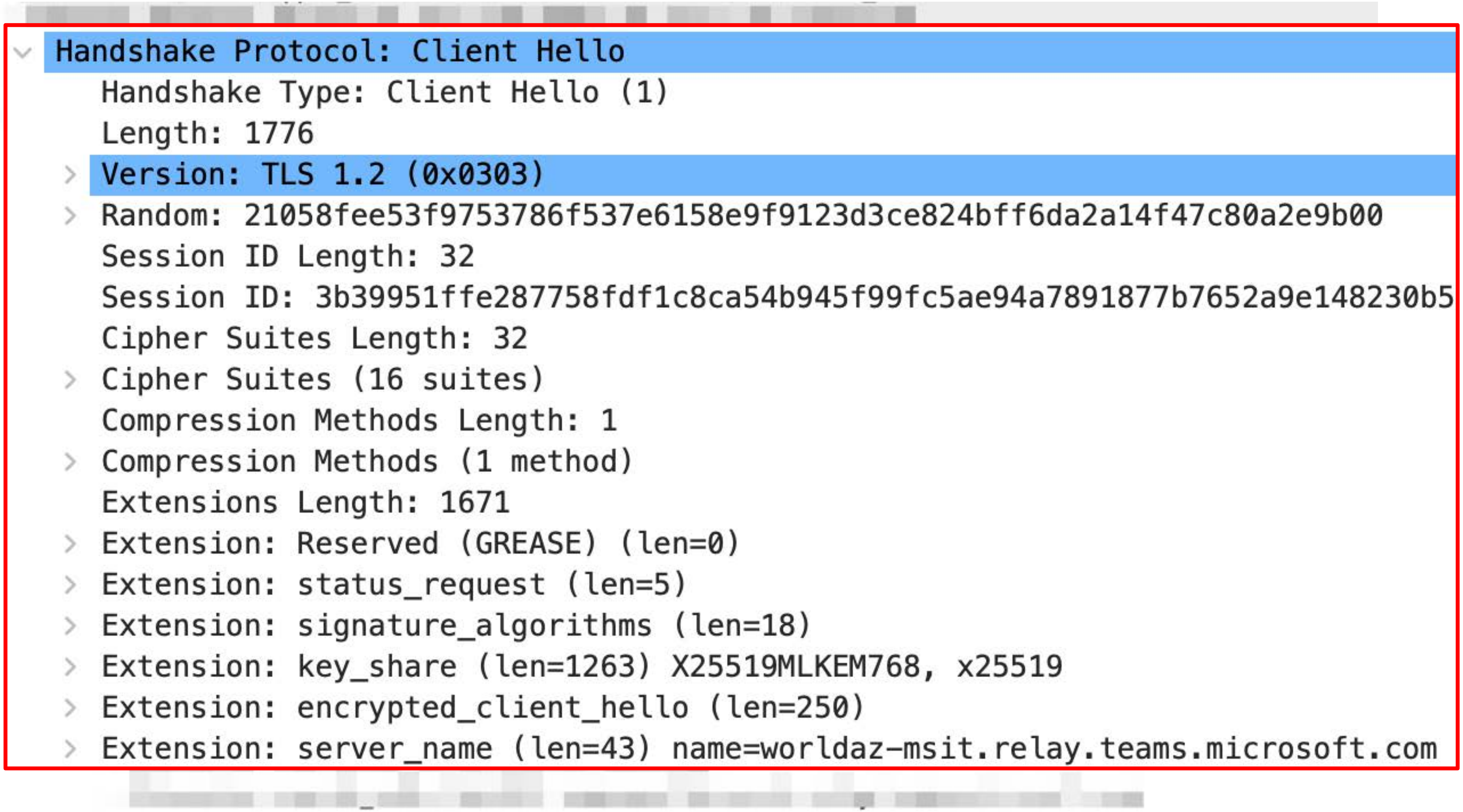


[redacted]

```
> Internet Protocol Version 4, Src: 192.168.1.43, Dst: 52.114.55.197
> Transmission Control Protocol, Src Port: 60570, Dst Port: 443, Seq: 1429, Ack: 1, Len: 357
> [2 Reassembled TCP Segments (1785 bytes): #30829(1428), #30830(357)]
v Transport Layer Security
  v TLSv1.2 Record Layer: Handshake Protocol: Client Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 1780
    v Handshake Protocol: Client Hello
```

[redacted]

# Examining Wireshark Traffic



[redacted]

```
∨ Handshake Protocol: Client Hello
      Handshake Type: Client Hello (1)
      Length: 1776
    > Version: TLS 1.2 (0x0303)
    > Random: 21058fee53f9753786f537e6158e9f9123d3ce824bff6da2a14f47c80a2e9b00
      Session ID Length: 32
      Session ID: 3b39951ffe287758fdf1c8ca54b945f99fc5ae94a7891877b7652a9e148230b5
      Cipher Suites Length: 32
    > Cipher Suites (16 suites)
      Compression Methods Length: 1
    > Compression Methods (1 method)
      Extensions Length: 1671
    > Extension: Reserved (GREASE) (len=0)
    > Extension: status_request (len=5)
    > Extension: signature_algorithms (len=18)
    > Extension: key_share (len=1263) X25519MLKEM768, x25519
    > Extension: encrypted_client_hello (len=250)
    > Extension: server_name (len=43) name=worldaz-msit.relay.teams.microsoft.com
```

[redacted]

# Conclusion

# Defensive Considerations

- Detection is hard
- Focus on other points in the kill chain
- Look for attacker tools proxied through the tunnel
- Low signal at network layer
- TURN creds can't be removed

# Things to Avoid

- Chasing weak signals like raw traffic volume
- Correlating process-to-destination traffic is noisy
- High effort, low return on detection accuracy
- Hard to distinguish legit conferencing from abuse

Process Correlation

Traffic Volume

# Canary Tokens

- “Read Teaming” targets credentials and shares
- Common targets: Slack, SharePoint, GitHub, Jira, etc.
- Targeting credentials and other sensitive data
- Canary tokens reveal enumeration early
- Simple, low-cost, and highly effective control

# Detecting Proxied Attacker Tooling

- Attackers proxy tools rather than run them locally
- Focus on offensive tool behavior not the channel
- Detect usage of tools like secretsdump.py or Impacket

# Future Work

- Other providers beyond Zoom/Teams also use TURN
- Opportunity for further mapping and validation
- Ideal entry-point project for new researchers
- Doesn’t require major tooling changes
- Expands applicability of the core method

# Future Work

- Current Go binaries weigh in around 2-3 MB
- Porting to C/C++ could reduce size under 1MB
- Smaller payloads improve operational stealth
- Better fit for constrained or ephemeral systems
- Helps with evasion and minimal footprint delivery

# Future Work

- Explore default settings in security appliances
- Identify vendor-based exclusions or allow-listing
- Check if IP ranges are auto-approved by default
- Investigate TLS inspection exemptions for key domains
- Assess how much trust these defaults embed

Dig into Security Tooling

# Takeaways and Questions

- Web conferencing solutions provide a compelling vector for covert short-term command and control channels
- TURNt is a new open-source tool that helps facilitate short-term C2 communication over the TURN protocol
- TURN provides a provider agnostic manner for tunneling traffic through potentially trusted web conferencing infrastructure

**Blog Post**

**Tool Release**

**LinkedIn**