# QUICForge

## Client-side Request Forgery in QUIC

Yuri Gbur and Florian Tschorsch




SEC Consult
an atos company

# whoami


## Yuri Gbur

- MSc in Computer Science at Technische Universität (TU) Berlin
- Security Consultant and Researcher at SEC Consult
- Berlin, Germany

yuri.gbur@posteo.com
y.gbur@sec-consult.com
@yukonsec

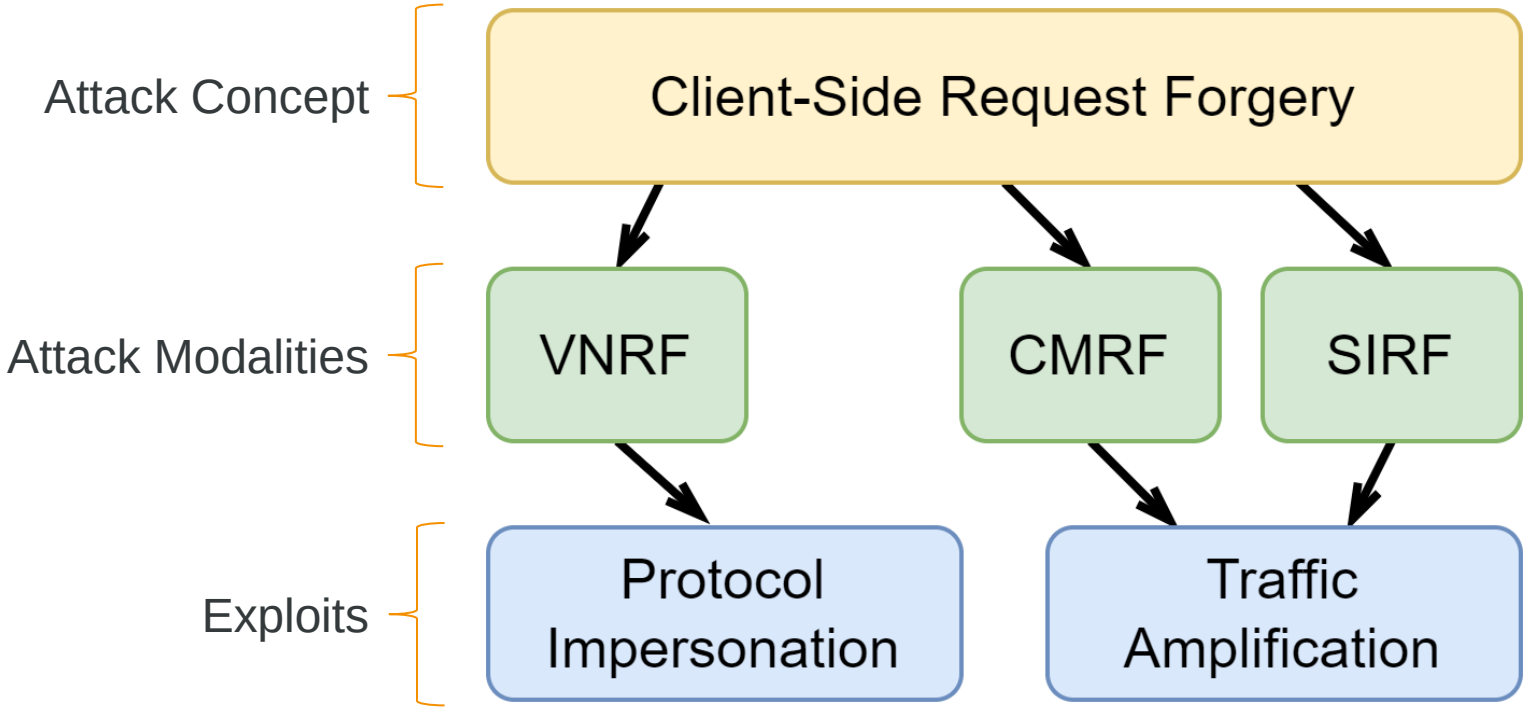
## Florian Tschorsch

- Professor for Computer Science at Technische Universität (TU) Berlin
- Chair for Distributed Security Infrastructure (DSI)
- Berlin, Germany

florian.tschorsch@tu-berlin.de
@flotschorsch

# Research Goals

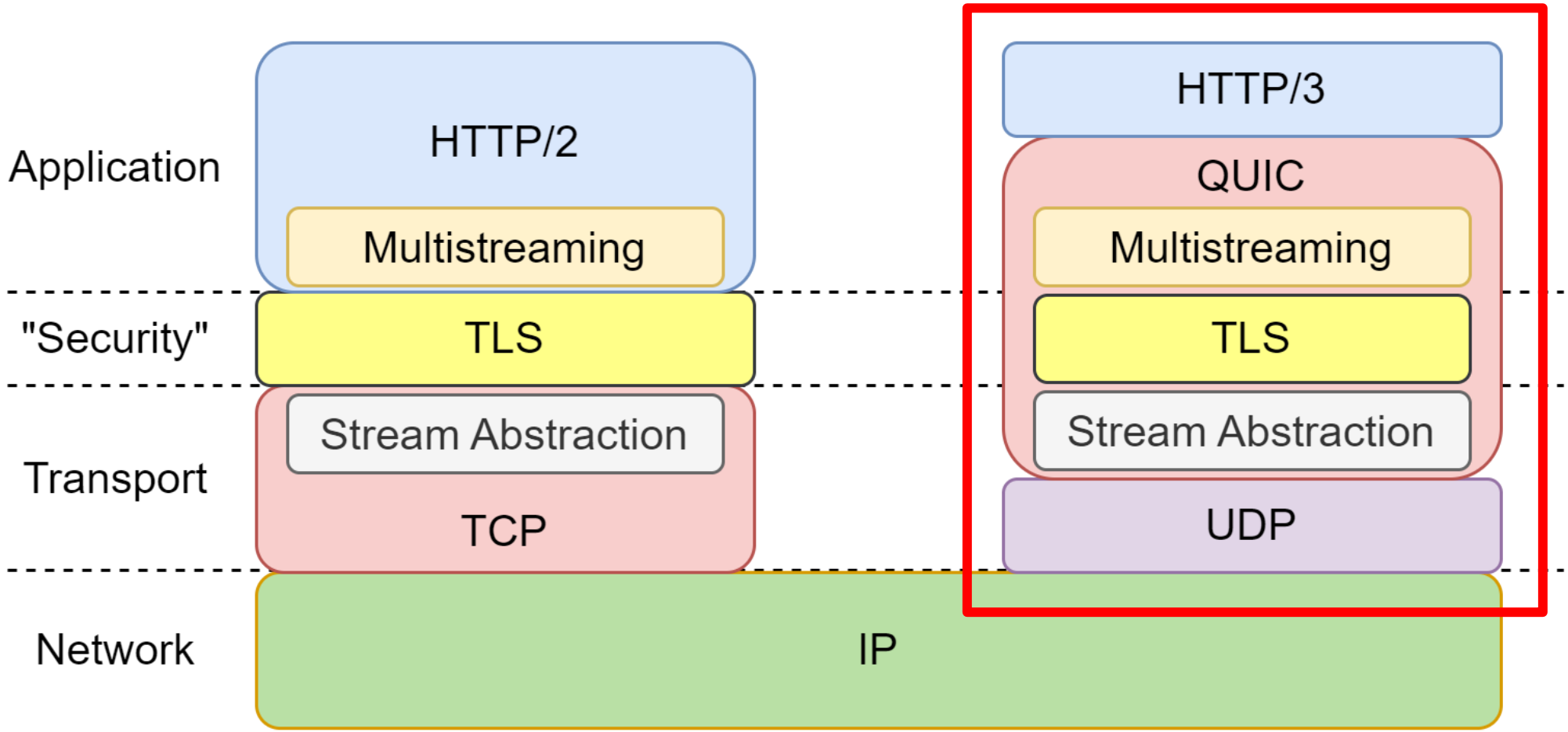
# Attack Modalities VS Exploits



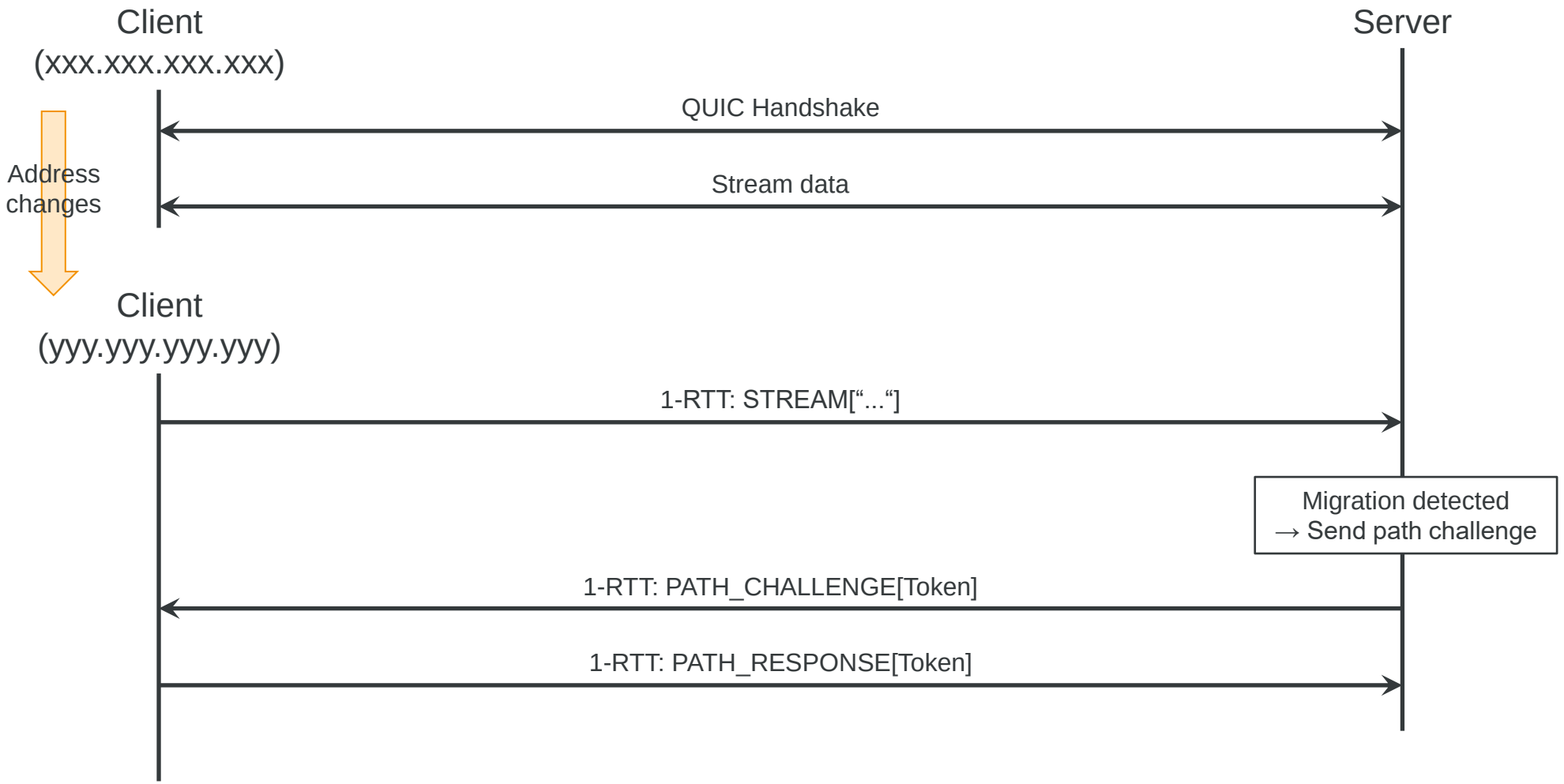

- Version Negotiation Request Forgery (VNRF)
- Connection Migration Request Forgery (CMRF)
- Server Initial Request Forgery (SIRF)

# QUIC(K) Background

# HTTP/2 VS HTTP/3

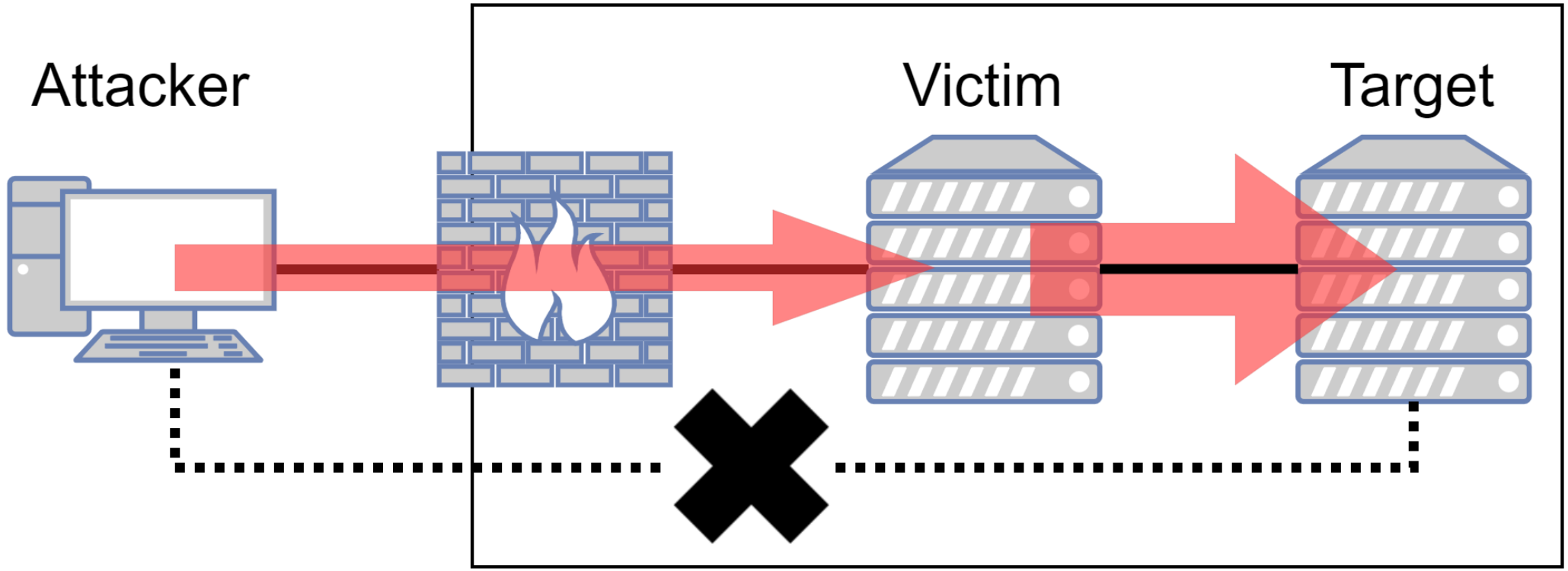


# Connection Migration



# Request Forgery

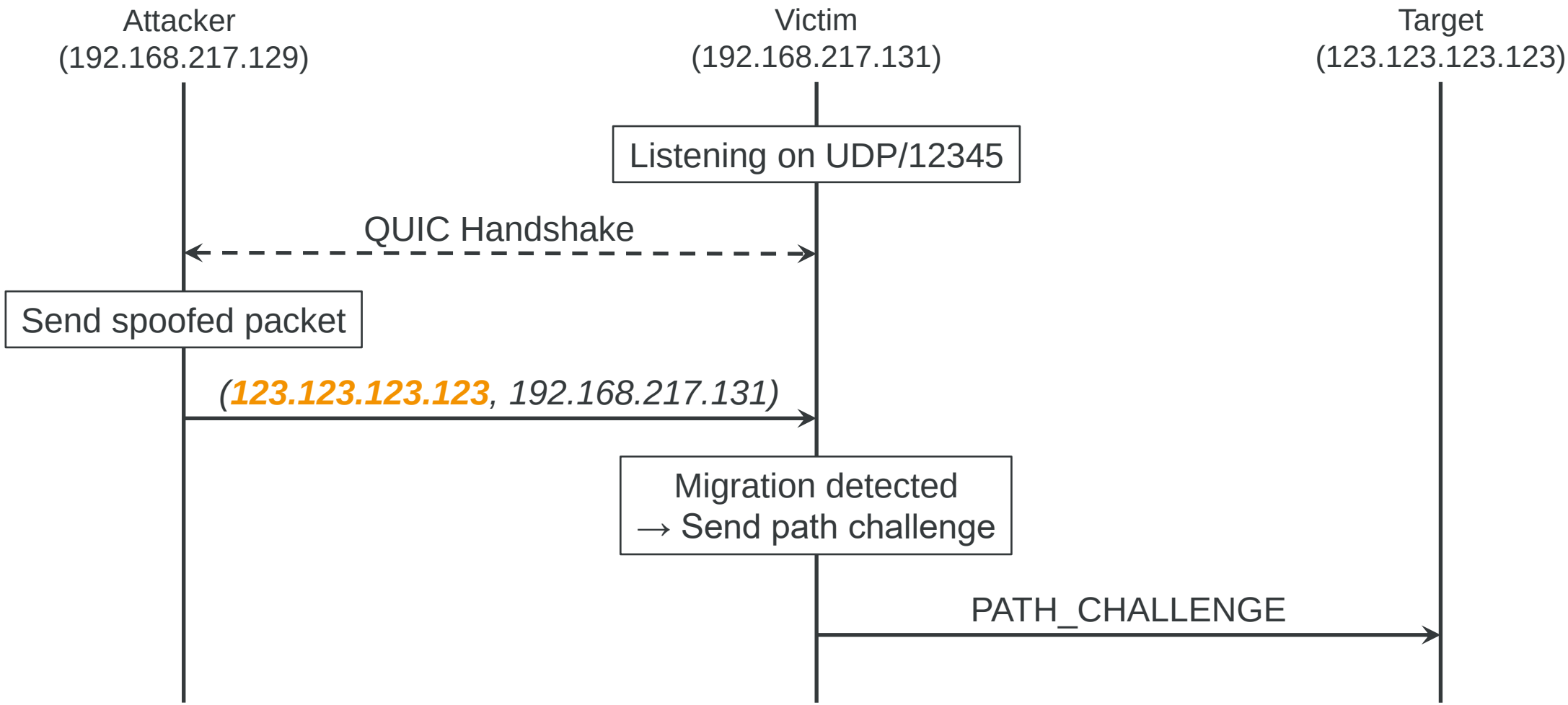
# Client-side Request Forgery



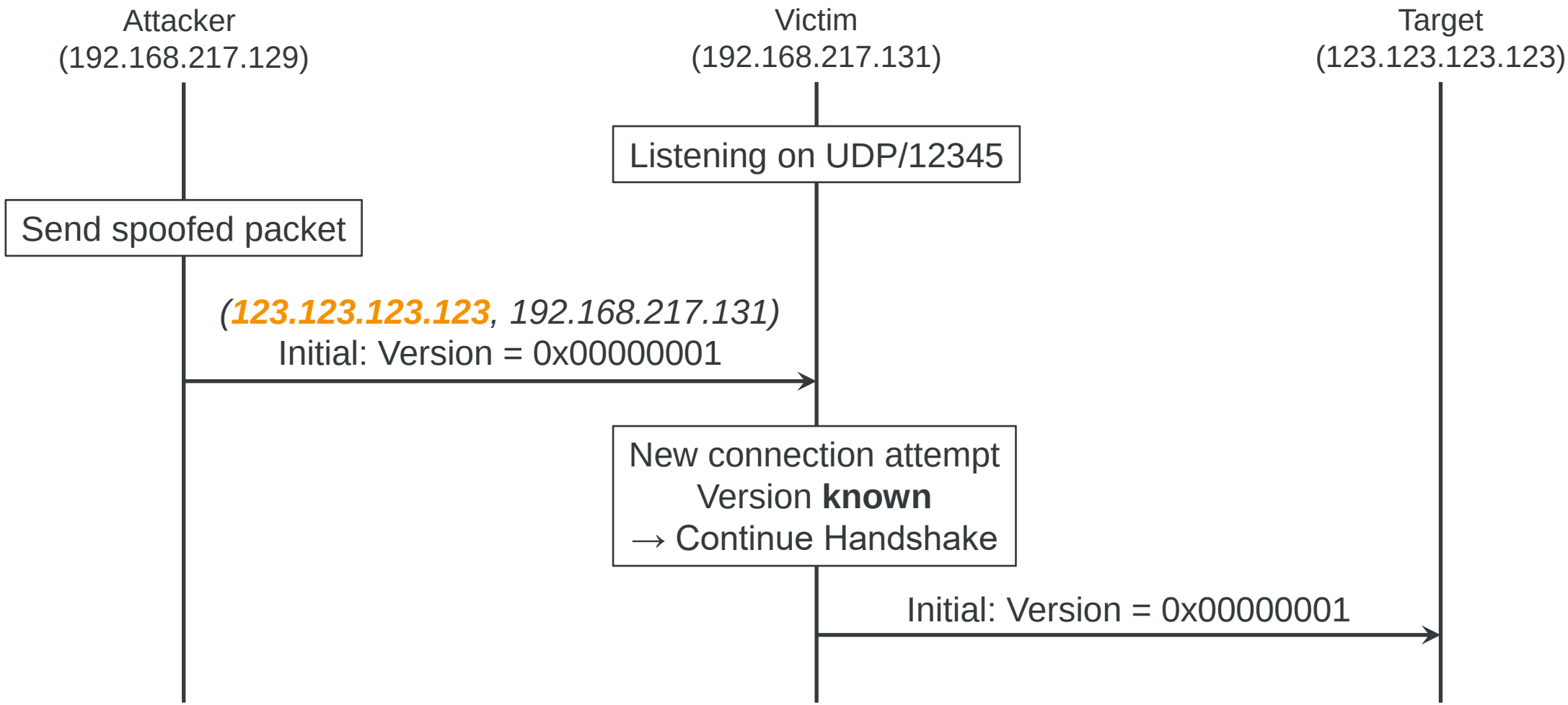

Bypassing Network Restrictions

Utilizing Victim Resources

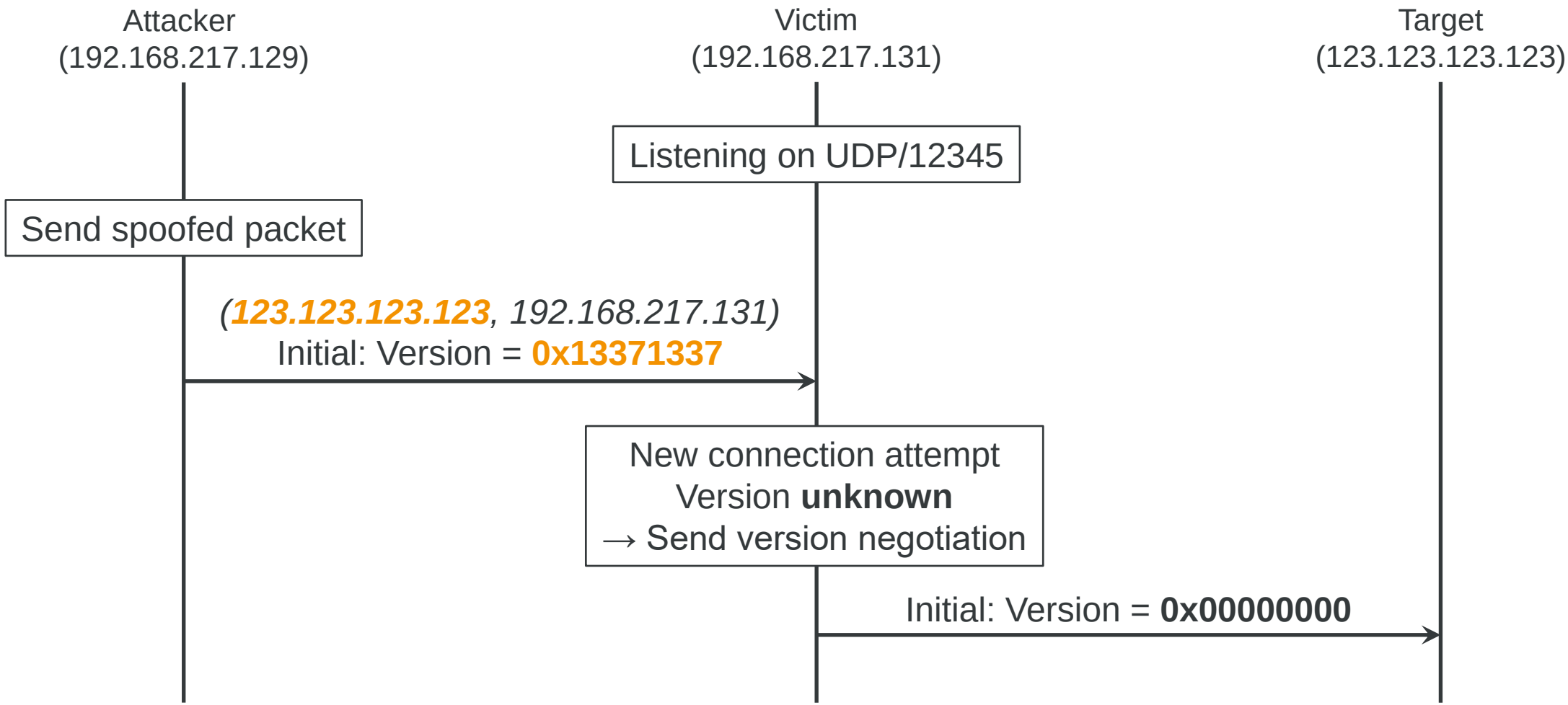
# Connection Migration Request Forgery (CMRF)



# Server Initial Request Forgery (SIRF)

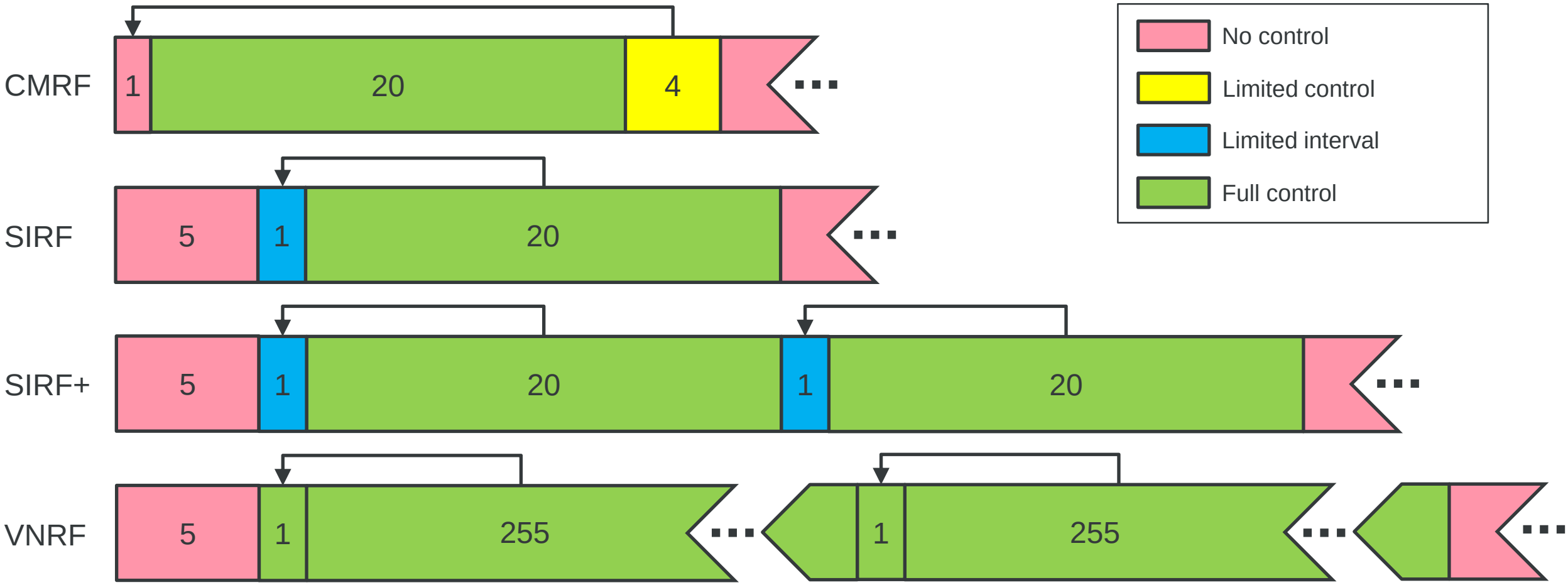


# Version Negotiation Request Forgery (VNRF)

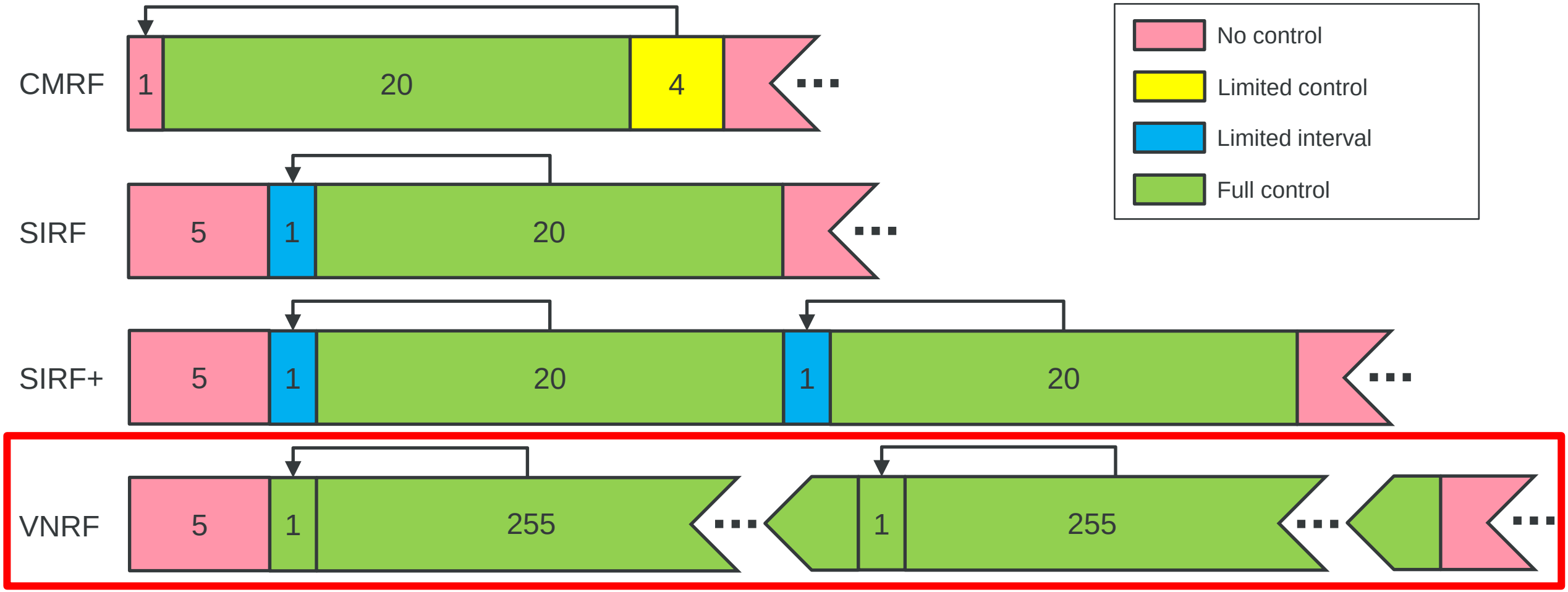


# Protocol Impersonation

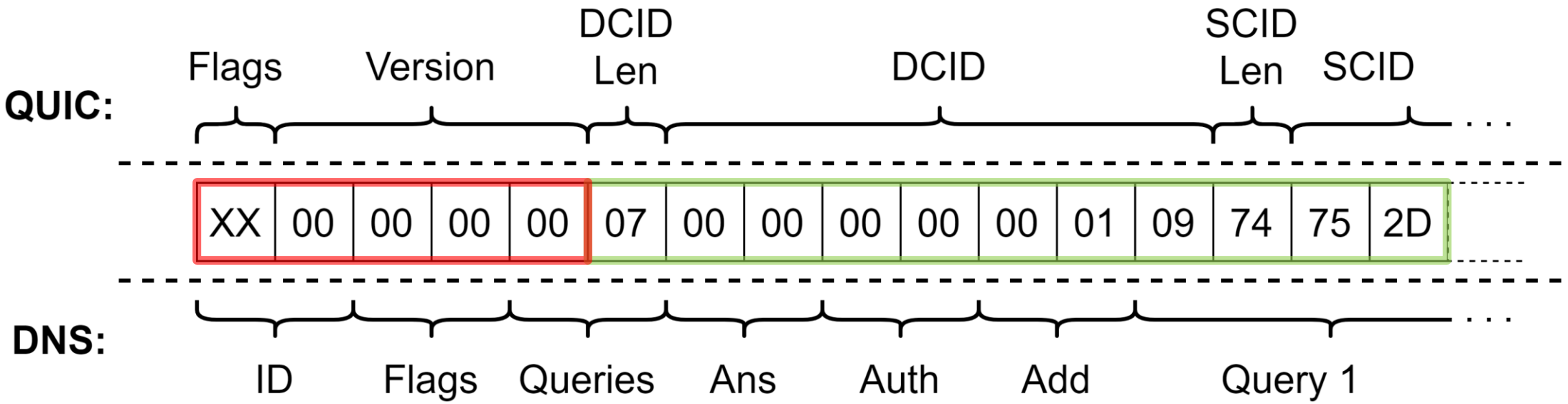
# Controllable Bytes for Protocol Impersonation



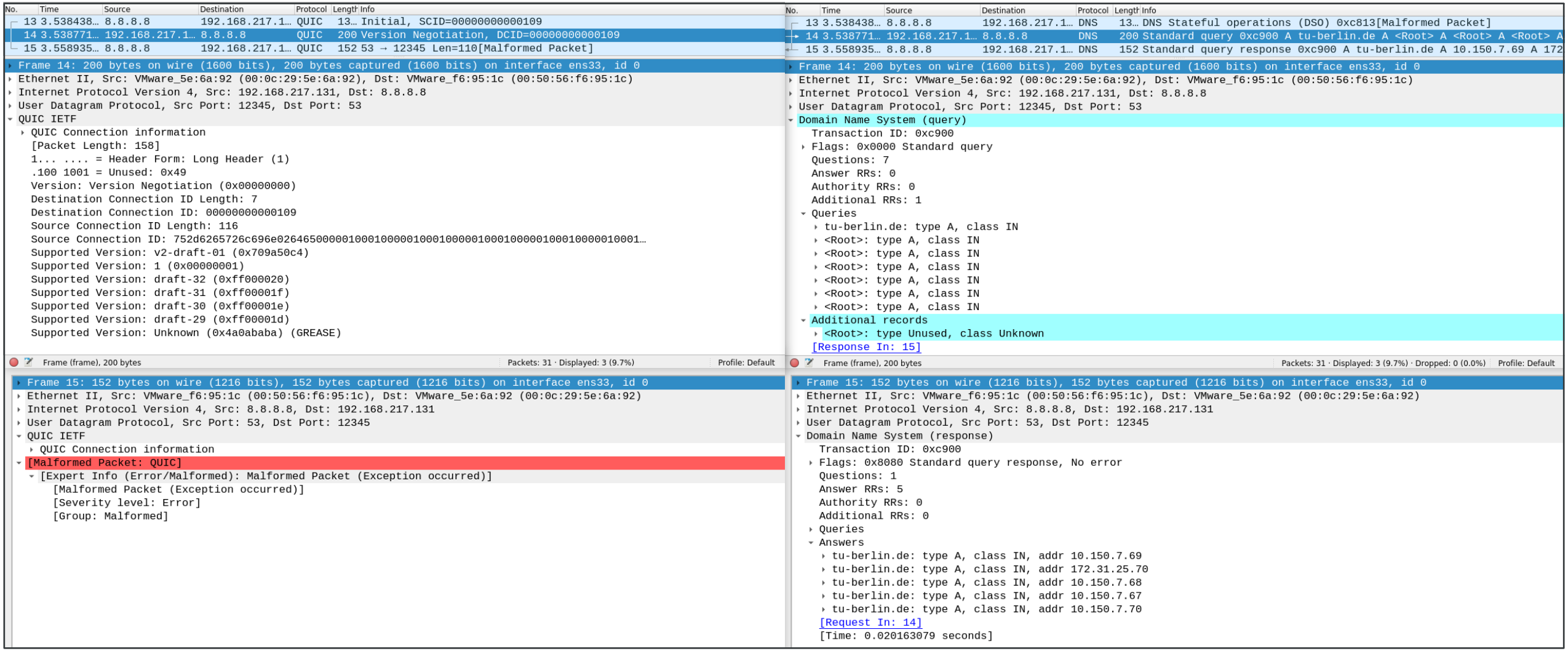
# Controllable Bytes for Protocol Impersonation



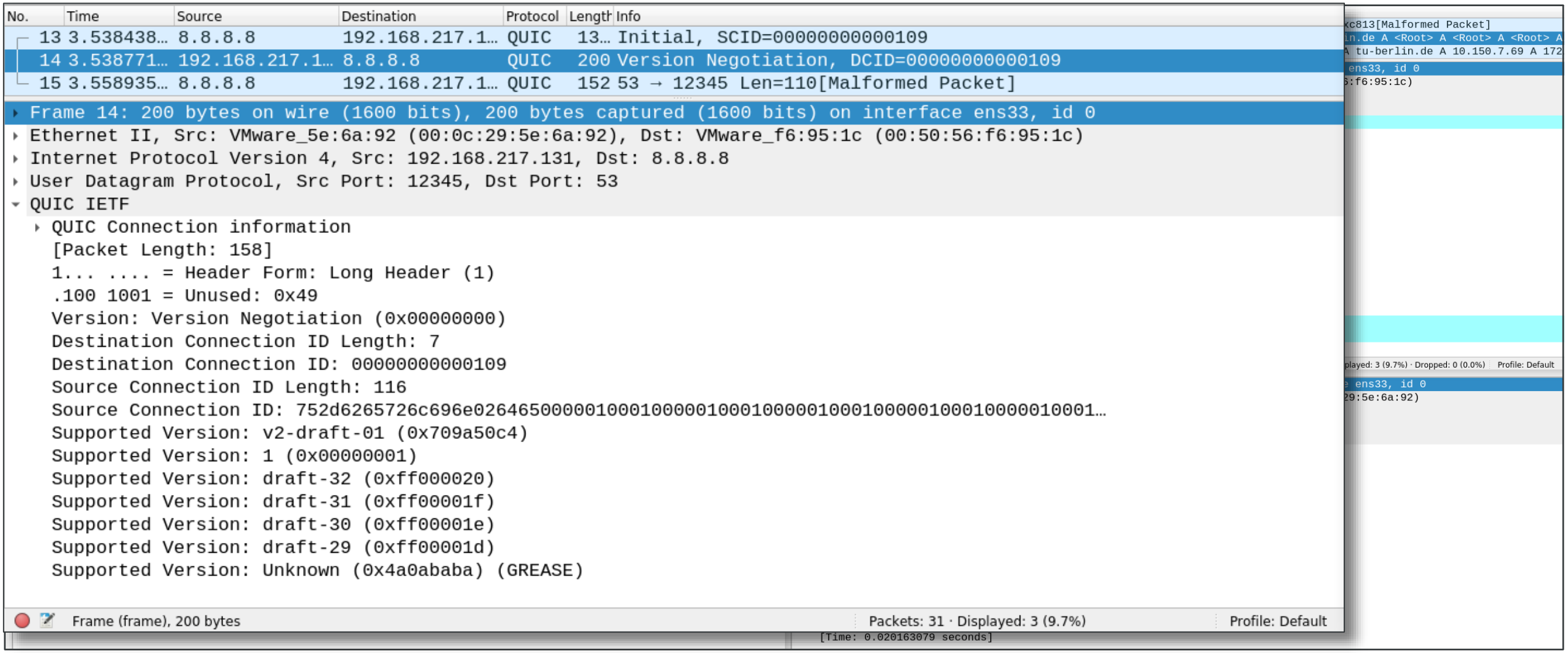
# Impersonating DNS Requests with VNRF



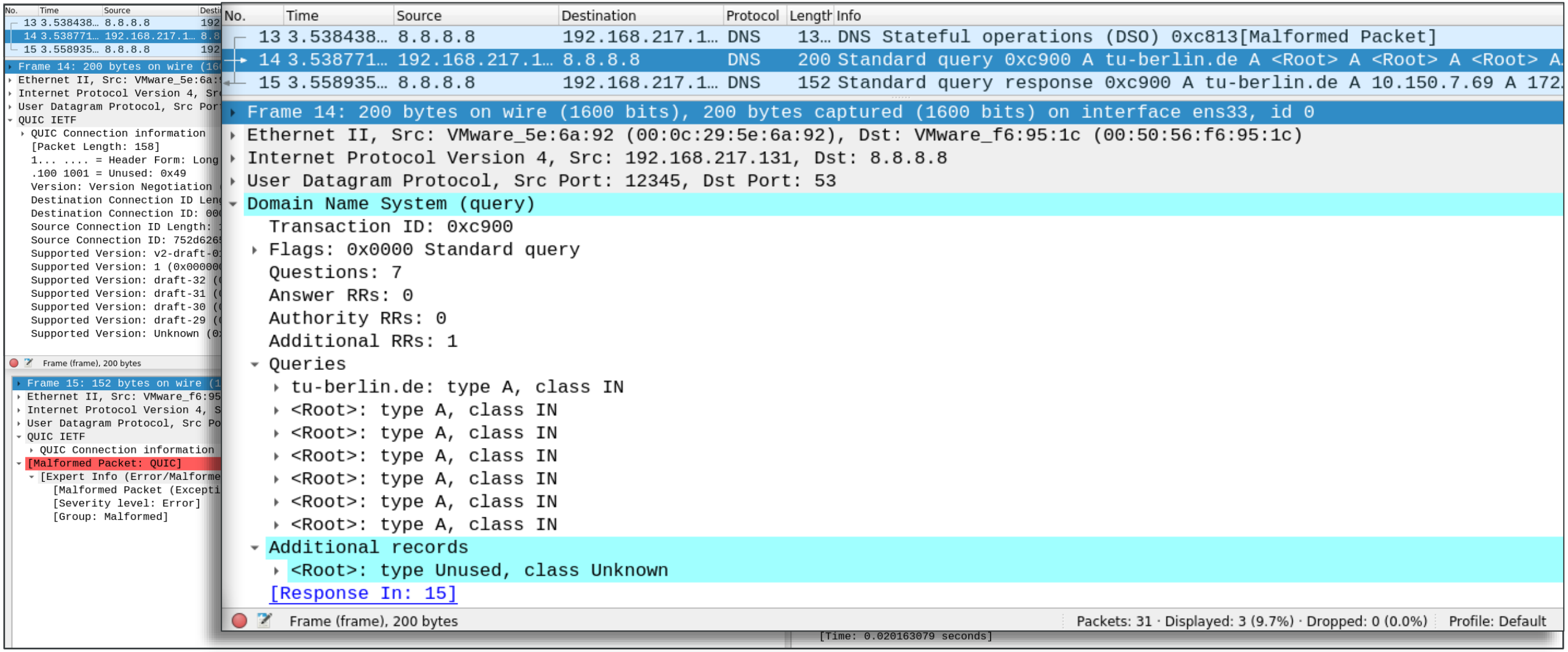
# Impersonating DNS Requests with VNRF (cont‘d)



# Impersonating DNS Requests with VNRF (cont‘d)



# Impersonating DNS Requests with VNRF (cont‘d)



# Mitigation

## CID Reflection

- A server always choses a fresh SCID, also for version negotiation

### Hashing

- A „seed“ for a CID still chosen by the client
- The server uses a hash of the seed as DCID
- An attacker would need to calculate the inverse to create a meaningful payload

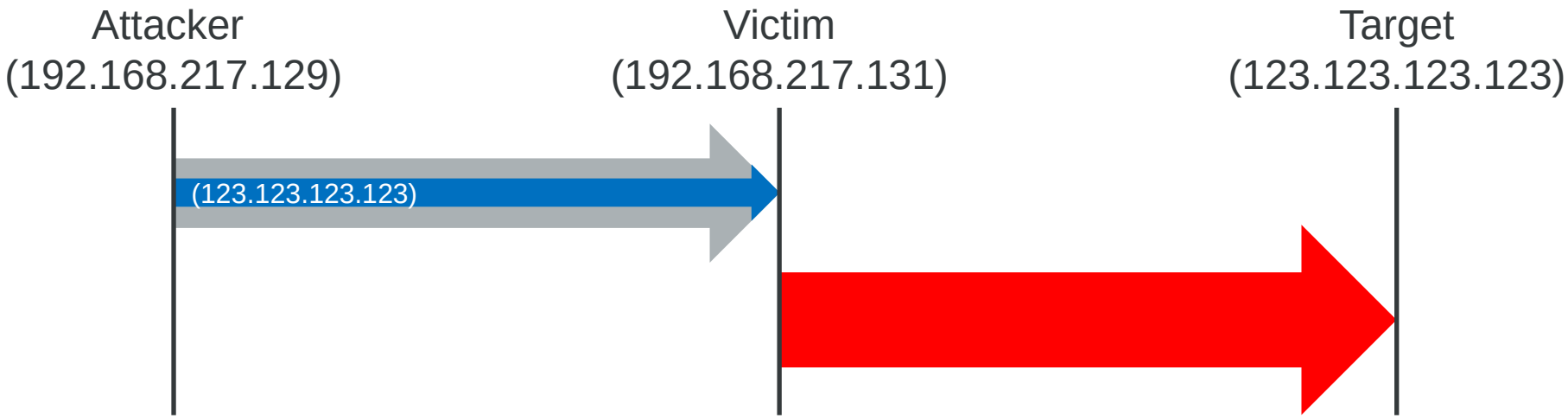
### Masking

- QUIC headers get an additional field containing a masking value
- The masking value is randomly generated by the server
- The entire remaining header is XORed
- Client maintains control over DCID but payloads will appear „random“

# Traffic Amplification

# Path amplification VS Bandwidth Amplification

$$PAF = \frac{\#\ Bytes\ from\ victim\ to\ target}{\#\ Bytes\ from\ attacker\ to\ victim\ with\ spoofed\ address}$$




$$BAF = \frac{\#\ Bytes\ from\ victim\ to\ target}{\#\ Bytes\ from\ attacker\ to\ victim}$$

# Amplification Pitfalls

***"[…] not send more than three times the amount of data received on any unvalidated path."***

**Minimum path requirements**

- „QUIC must not be used if the network path cannot support 1200 bytes datagrams“
- Ensured through padding of initial packets and path challenges
- Small packets on new paths are an issue
- ***Server should send two separate path validations***

**Unbalanced handshake sizes**

- Server initial packets are larger than client intial packets
- ***Server initial packets should never be larger than 3*1200 bytes***

# Amplification Pitfalls

## Reliability

- No „typical“ reliability in path challenges, server can send multiple challenges
  - Initial bursts
  - Re-send with timeout
- ***Multiple challenges definitely surpass amplification limits***

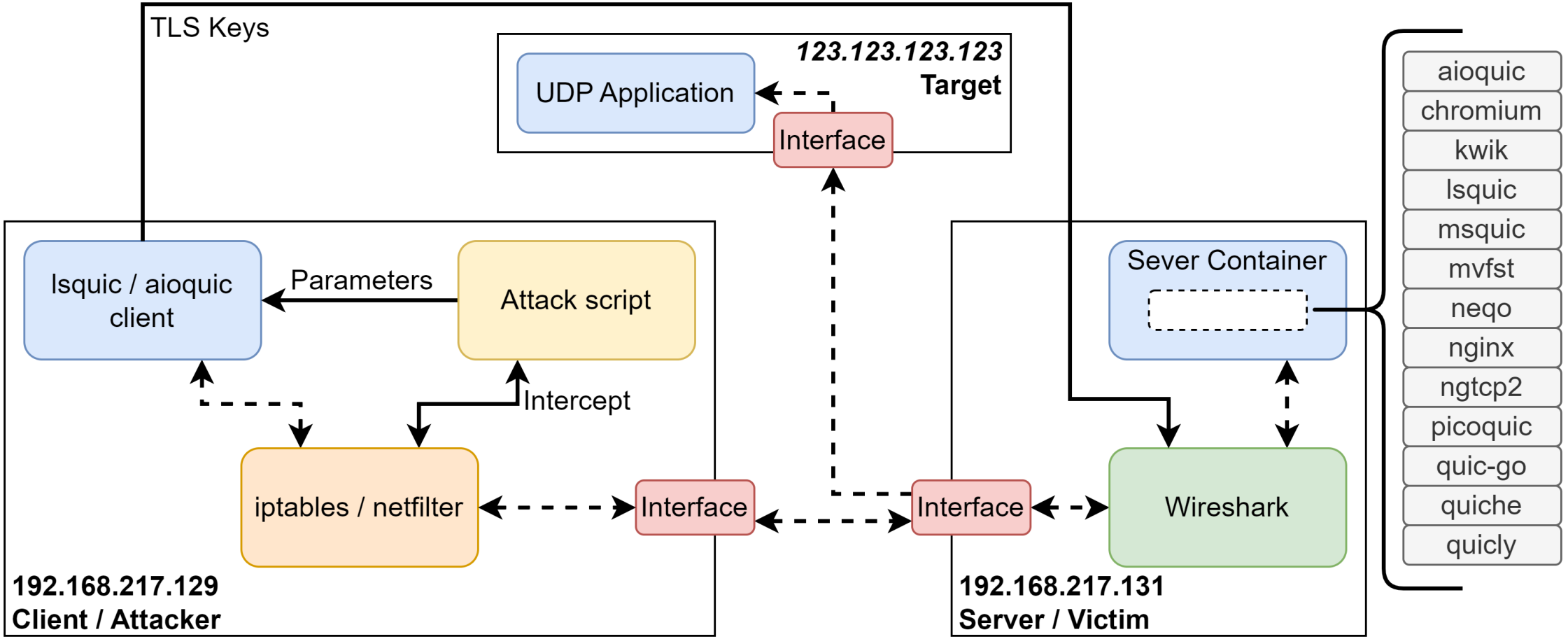
- Normal packets are re-sent if the acknowledgment is not received
- ***Server should not re-send server initial packets***
  - Retries for the initial messages have to be handled by the client.

# Mitigation

**RTFM**

# Evaluation

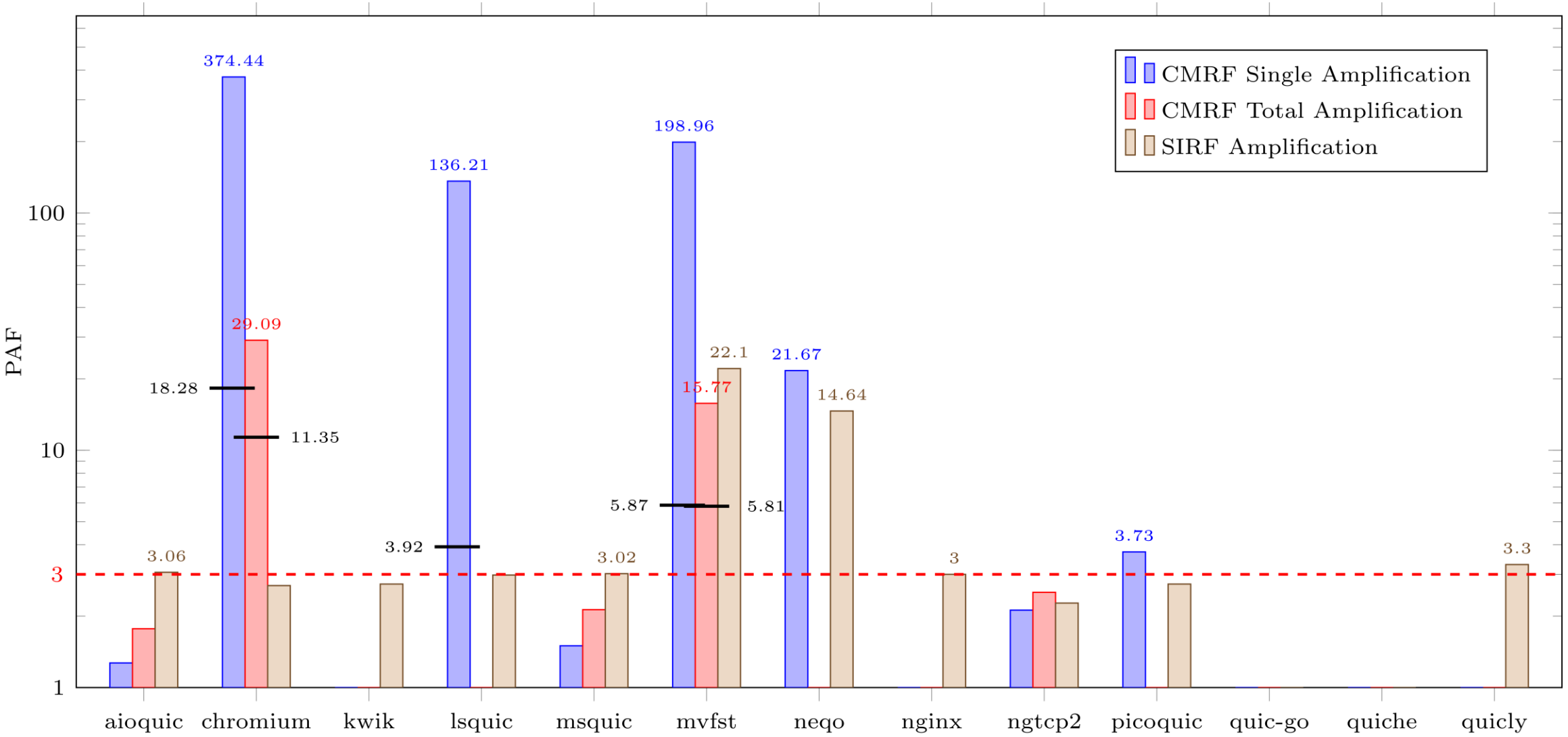
# Evaluation Setup



# Testing Implementations

| | CMRF | | | | SIRF | | | VNRF | |
|---|---|---|---|---|---|---|---|---|---|
| | Vuln. | Pad. | New CID | PAF > 3 | Vuln. | PAF > 3 | Ref. CID | Vuln. | CID>20 |
| aioquic | ✓ | ✗ | ✗ | ✗ | ✓ | ✓ | ✗ | ✓ | ✗ |
| chromium | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ |
| kwik | ✗ | - | - | - | ✓ | ✗ | ✗ | ✓ | ✓ |
| lsquic | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ |
| msquic | ✓ | ✗ | ✓ | ✗ | ✓ | ✓ | ✗ | ✓ | ✓ |
| mvfst | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ |
| neqo | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ |
| nginx | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ |
| ngtcp2 | ✓ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✓ | ✓ |
| picoquic | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ |
| quic-go | ✗ | - | - | - | (✓) | ✗ | ✗ | ✓ | ✓ |
| quiche | ✗ | - | - | - | (✓) | ✗ | ✗ | ✓ | ✓ |
| quicly | ✗ | - | - | - | ✓ | ✓ | ✗ | ✓ | ✓ |
| **Total** | **9** | **2** | **5** | **6** | **11**(13) | **6** | **0** | **13** | **9** |

# Amplification



# Conclusion

# Conclusion

- ***All 13 implementations were „vulnerable“ to at least one technique***

- Amplification protection is covered in the specification
- ***Yet, significant PAF values uncovered***
  - 374.44 for CMRF
  - 22.1 for SIRF
- Vendors were notified and implemented

- ***Protocol impersonation is possible with VNRF***
- Currently no built-in protection mechanism
  - changes to the specification should be considered

# Thanks!

**Blogpost with additional technical details:**

**https://r.sec-consult.com/quic**

**Thanks for listening!**