WCTF 2020 Challenge: Spaceless Spacing

Connor Nelson
Shellphish

Initial Step: What is the challenge?

Connect to 3.137.154.50:4242

We know there is a challenge on some port.

This is all we know.

Initial Step: What is the challenge speaking?

```
$ nc 3.137.154.50 4242 | hexdump -C

random_data

00000000  00 00 12 04 00 00 00 00  00 00 03 00 00 00 80 00  |................|

00000010  04 00 01 00 00 00 05 00  ff ff ff 00 00 04 08 00  |................|
```

Initial Step: What is the challenge speaking?

```
$ nc 3.137.154.50 4242 | hexdump -C

random_data

00000000  00 00 12 04 00 00 00 00  00 00 03 00 00 00 80 00  |................|

00000010  04 00 01 00 00 00 05 00  ff ff ff 00 00 04 08 00  |................|
```

Random binary data, what is this?

Google!

Initial Step: What is the challenge speaking?

```
$ nc 3.137.154.50 4242 | hexdump -C

random_data

00000000  00 00 12 04 00 00 00 00  00 00 03 00 00 00 80 00  |................|

00000010  04 00 01 00 00 00 05 00  ff ff ff 00 00 04 08 00  |................|
```

Google reveals this is HTTP/2.0

Initial Step: How do we speak to it?

```
$ curl 3.137.154.50:4242/

curl: (1) Received HTTP/0.9 when not allowed
```

Initial Step: How do we speak to it?

```
$ curl --http2 3.137.154.50:4242/

curl: (1) Received HTTP/0.9 when not allowed
```

Initial Step: How do we speak to it?

```
$ curl --http2-prior-knowledge 3.137.154.50:4242/

#!/usr/bin/env python


import os

import time


from flask import Flask

...
```

The Challenge

```
@app.route("/<secret>")
def check_secret(secret):
    if len(secret) != len(SECRET):
        return "SPACELESS SPACING!"
    for a, b in zip(secret, SECRET):
        if a == " ":
            continue
        elif a != b:
            return "INCORRECT!"
        else:
            time.sleep(PLANCK_TIME)
    if " " in secret:
        return "INCORRECT!"
    return "CORRECT!"
```

The Challenge: Test

```
$ curl --http2-prior-knowledge 3.137.154.50:4242/test

SPACELESS SPACING!%
```

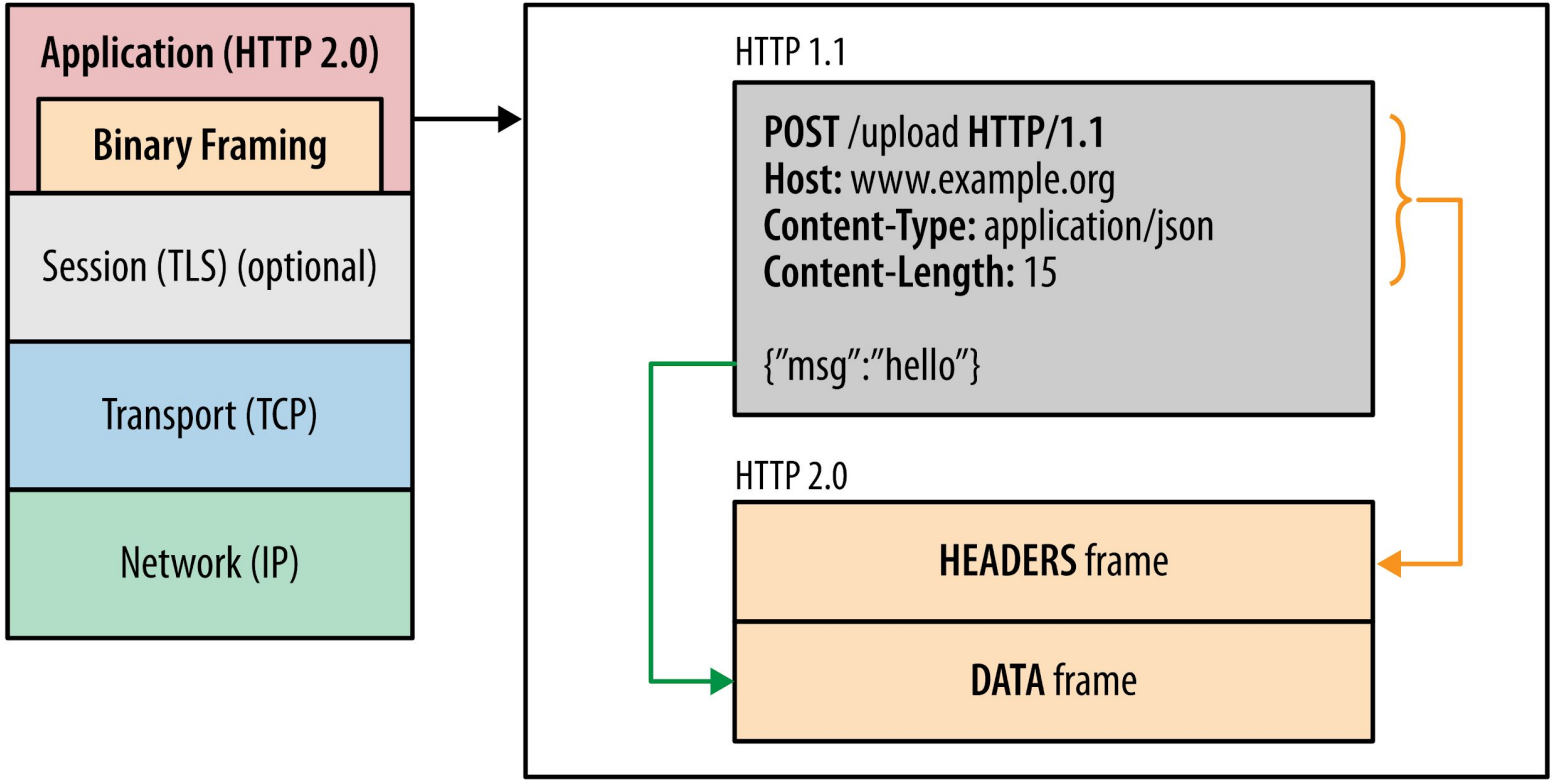
The Challenge: Bruteforce Flag Size

```
$ curl --http2-prior-knowledge 3.137.154.50:4242/a
SPACELESS SPACING!%
$ curl --http2-prior-knowledge 3.137.154.50:4242/ab
SPACELESS SPACING!%
$ curl --http2-prior-knowledge 3.137.154.50:4242/abc
SPACELESS SPACING!%
$ curl --http2-prior-knowledge 3.137.154.50:4242/abcd
SPACELESS SPACING!%
$ curl --http2-prior-knowledge 3.137.154.50:4242/abcde
SPACELESS SPACING!%
$ curl --http2-prior-knowledge 3.137.154.50:4242/abcdef
SPACELESS SPACING!%
$ curl --http2-prior-knowledge 3.137.154.50:4242/abcdefg
SPACELESS SPACING!%
$ curl --http2-prior-knowledge 3.137.154.50:4242/abcdefgh
SPACELESS SPACING!%
$ curl --http2-prior-knowledge 3.137.154.50:4242/abcdefghi
SPACELESS SPACING!%
$ curl --http2-prior-knowledge 3.137.154.50:4242/abcdefghij
INCORRECT!%
```

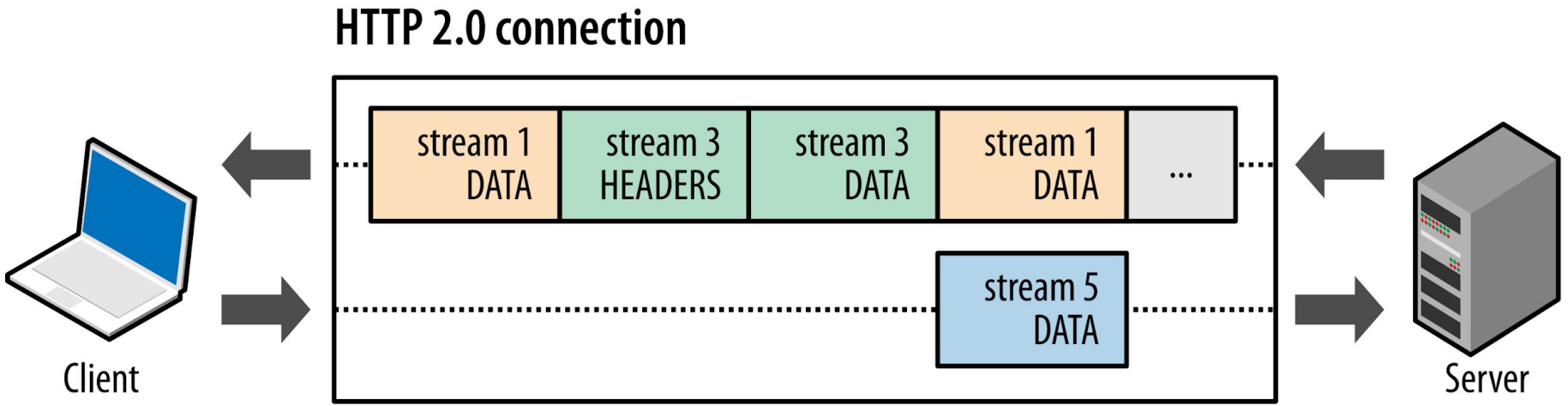
The Challenge

```
@app.route("/<secret>")
def check_secret(secret):
    if len(secret) != len(SECRET):
        return "SPACELESS SPACING!"
    for a, b in zip(secret, SECRET):
        if a == " ":
            continue
        elif a != b:
            return "INCORRECT!"
        else:
            time.sleep(PLANCK_TIME)
    if " " in secret:
        return "INCORRECT!"
    return "CORRECT!"
```

HTTP/2.0


Application (HTTP 2.0)
Binary Framing
Session (TLS) (optional)
Transport (TCP)
Network (IP)
HTTP 1.1
POST /upload HTTP/1.1
Host: www.example.org
Content-Type: application/json
Content-Length: 15
{"msg":"hello"}
HTTP 2.0
HEADERS frame
DATA frame

HTTP/2.0: Multiplexing



The Challenge

```
@app.route("/<secret>")
def check_secret(secret):
    if len(secret) != len(SECRET):
        return "SPACELESS SPACING!"
    for a, b in zip(secret, SECRET):
        if a == " ":
            continue
        elif a != b:
            return "INCORRECT!"
        else:
            time.sleep(PLANCK_TIME)
    if " " in secret:
        return "INCORRECT!"
    return "CORRECT!"
```

Timeless Timing Attacks

Timeless Timing Attacks: Exploiting Concurrency to Leak Secrets over Remote Connections

Tom Van Goethem, *imec-DistriNet, KU Leuven;* Christina Pöpper, *New York University Abu Dhabi;* Wouter Joosen, *imec-DistriNet, KU Leuven;* Mathy Vanhoef, *New York University Abu Dhabi*

https://www.usenix.org/conference/usenixsecurity20/presentation/van-goethem

This paper is included in the Proceedings of the 29th USENIX Security Symposium.

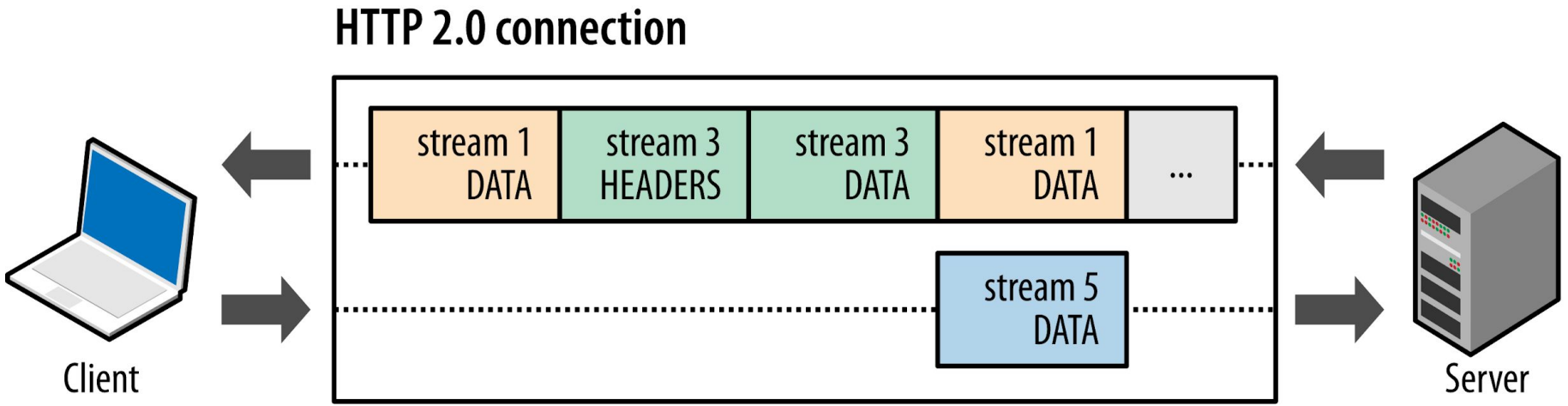
August 12–14, 2020

978-1-939133-17-5

The Challenge

```
@app.route("/<secret>")
def check_secret(secret):
    if len(secret) != len(SECRET):
        return "SPACELESS SPACING!"
    for a, b in zip(secret, SECRET):
        if a == " ":
            continue
        elif a != b:
            return "INCORRECT!"
        else:
            time.sleep(PLANCK_TIME)
    if " " in secret:
        return "INCORRECT!"
    return "CORRECT!"
```

HTTP/2.0: Multiplexing



The Solution

Make several simultaneous requests for:

- Some known wrong character (*)
- Some guessed character (A)

Measure frequency of which came back first vs second

The Solution

Make several simultaneous requests for:

- Some known wrong character (*)
- Some guessed character (B)

Measure frequency of which came back first vs second

The Solution

Make several simultaneous requests for:

- Some known wrong character (*)
- Some guessed character (C)

Measure frequency of which came back first vs second