black hat BRIEFINGS

black hat BRIEFINGS

AUGUST 6-7, 2025

MANDALAY BAY / LAS VEGAS

Amplify and Annihilate: Discovering and Exploiting Vulnerable Tunnelling Hosts

Angelos Beitis, Mathy Vanhoef

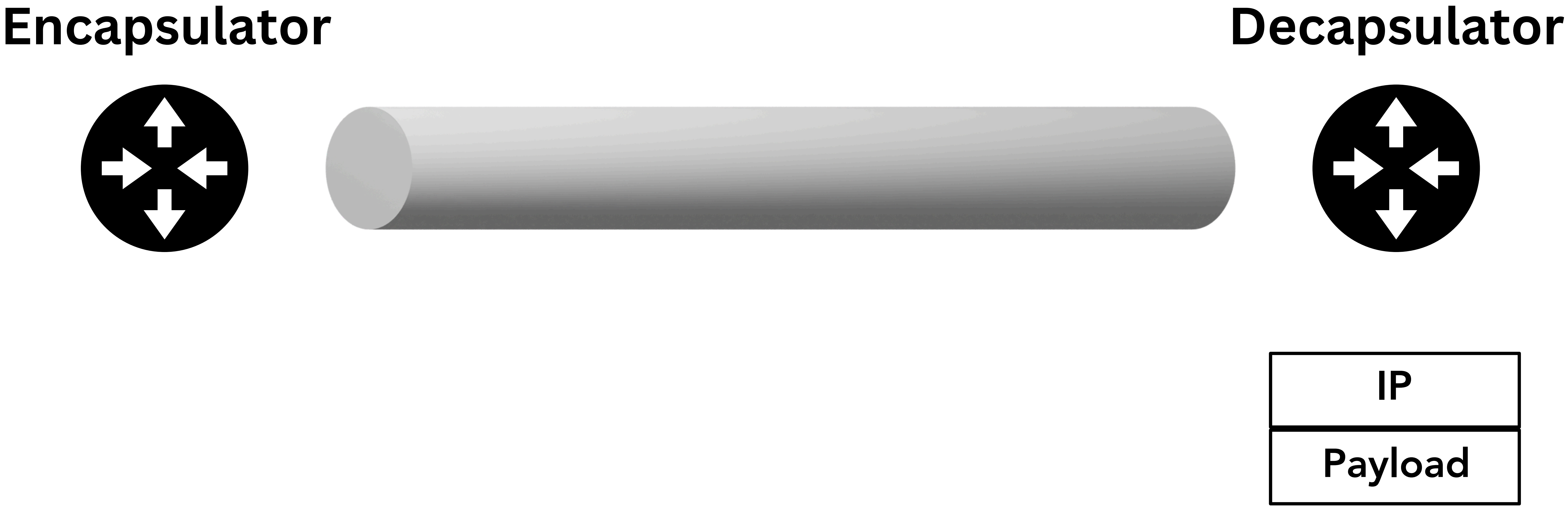
KU LEUVEN DistriNet

Introduction


Encapsulator
Decapsulator
IP
Payload


Encapsulator
Decapsulator
IP
Payload


Encapsulator
Decapsulator
IP
Payload


Encapsulator
Decapsulator
IP
Payload

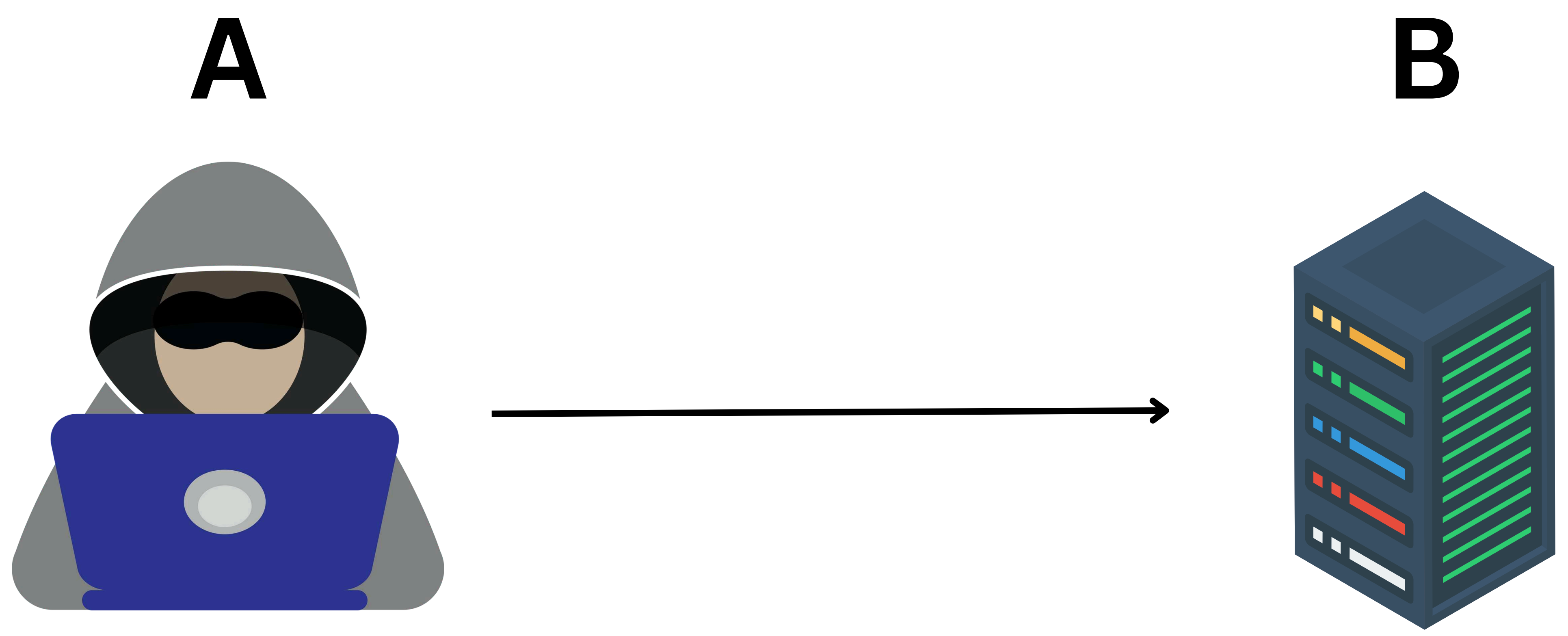
Authentication
Encryption
Data Integrity

CVE-2020-10136

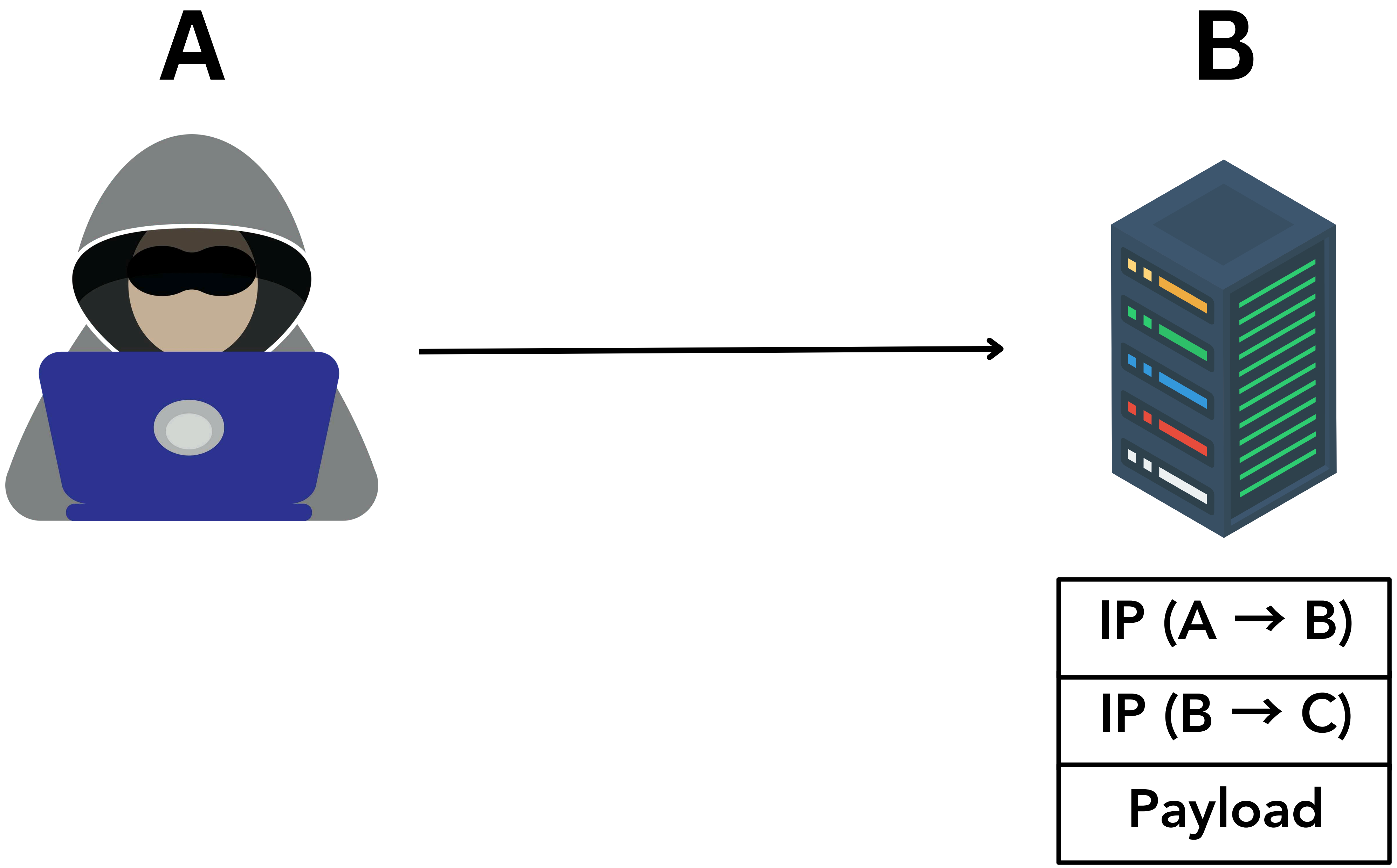
CVE-2020-10136

Reported by: Yannay Livneh

***IP-in-IP** protocol specifies IP Encapsulation within IP standard (RFC 2003, STD 1) that decapsulate and route IP-in-IP traffic **is vulnerable to spoofing, access-control bypass and other unexpected behavior** due to the **lack of validation** to verify network packets **before decapsulation and routing**.*



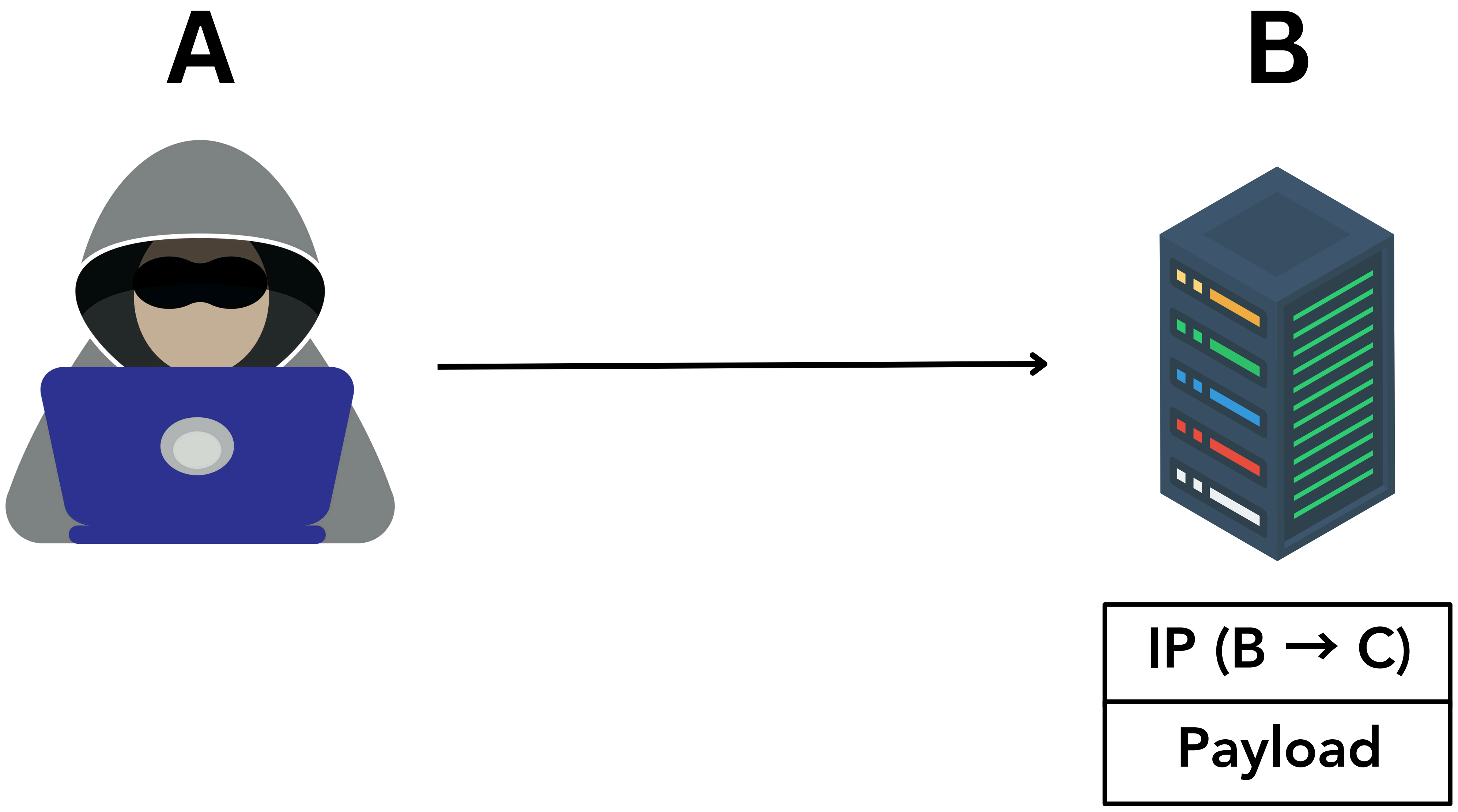
IP (A → B)
IP (B → C)
Payload

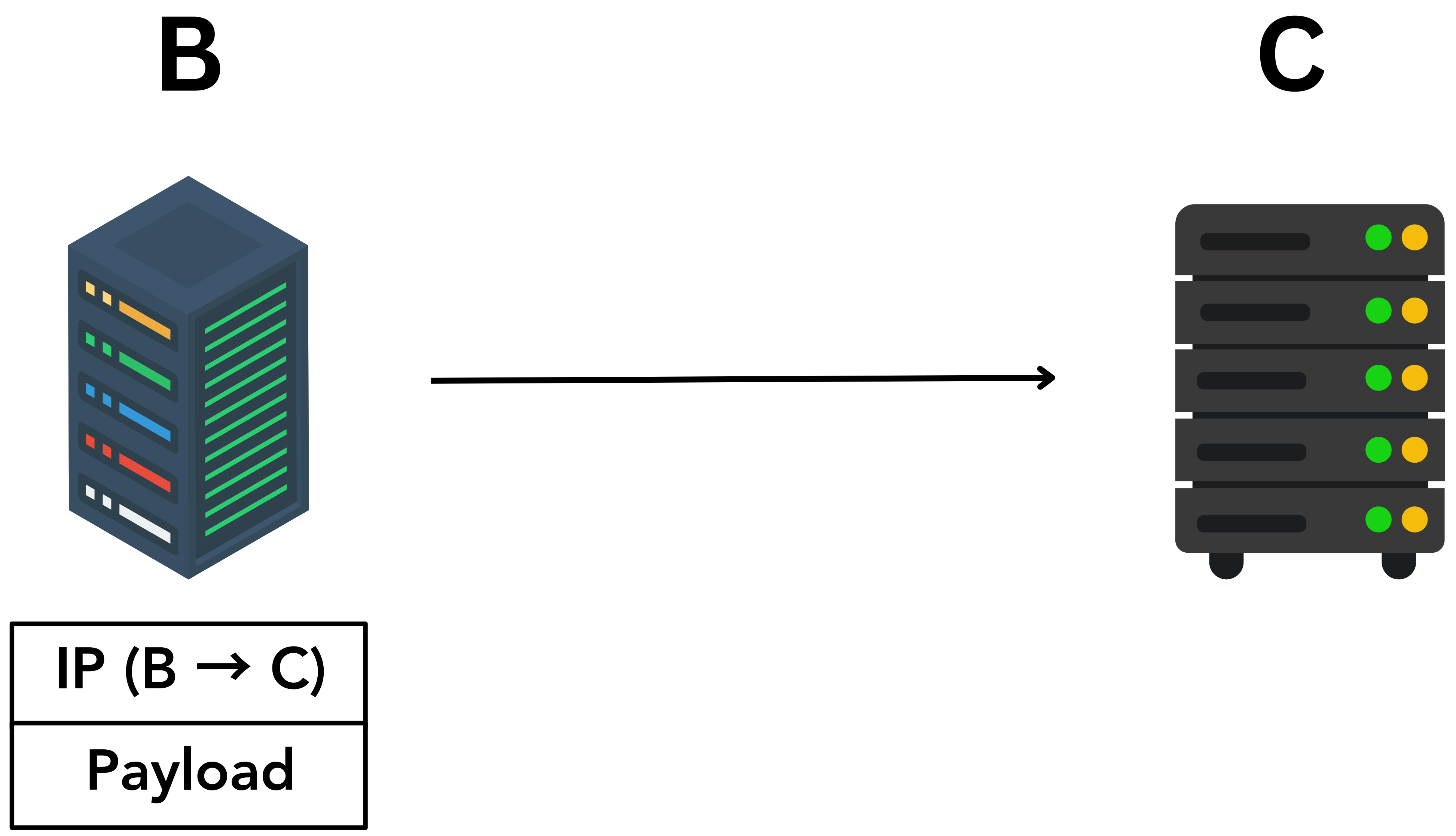

A
B
IP (A → B)
IP (B → C)
Payload

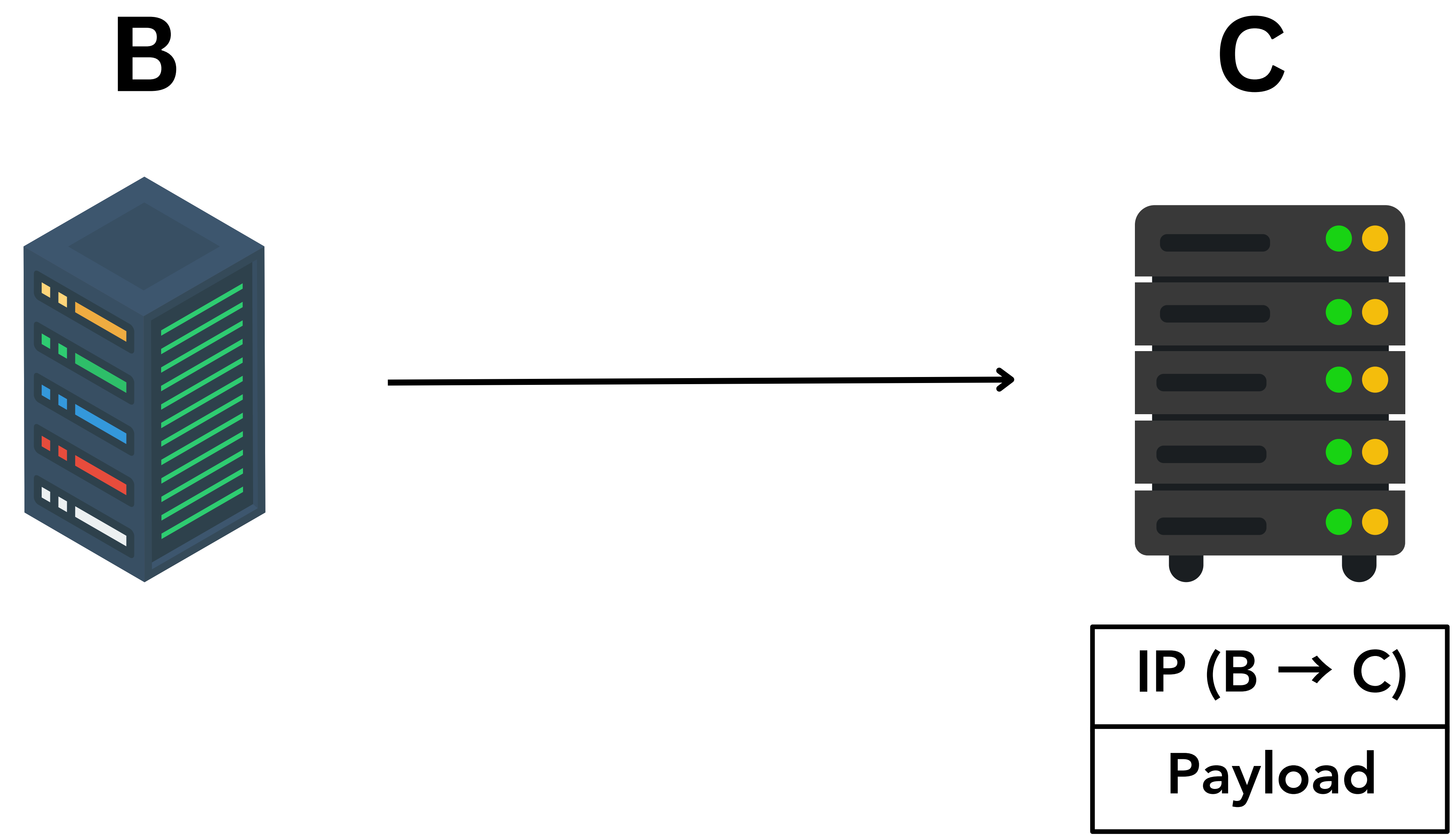
A

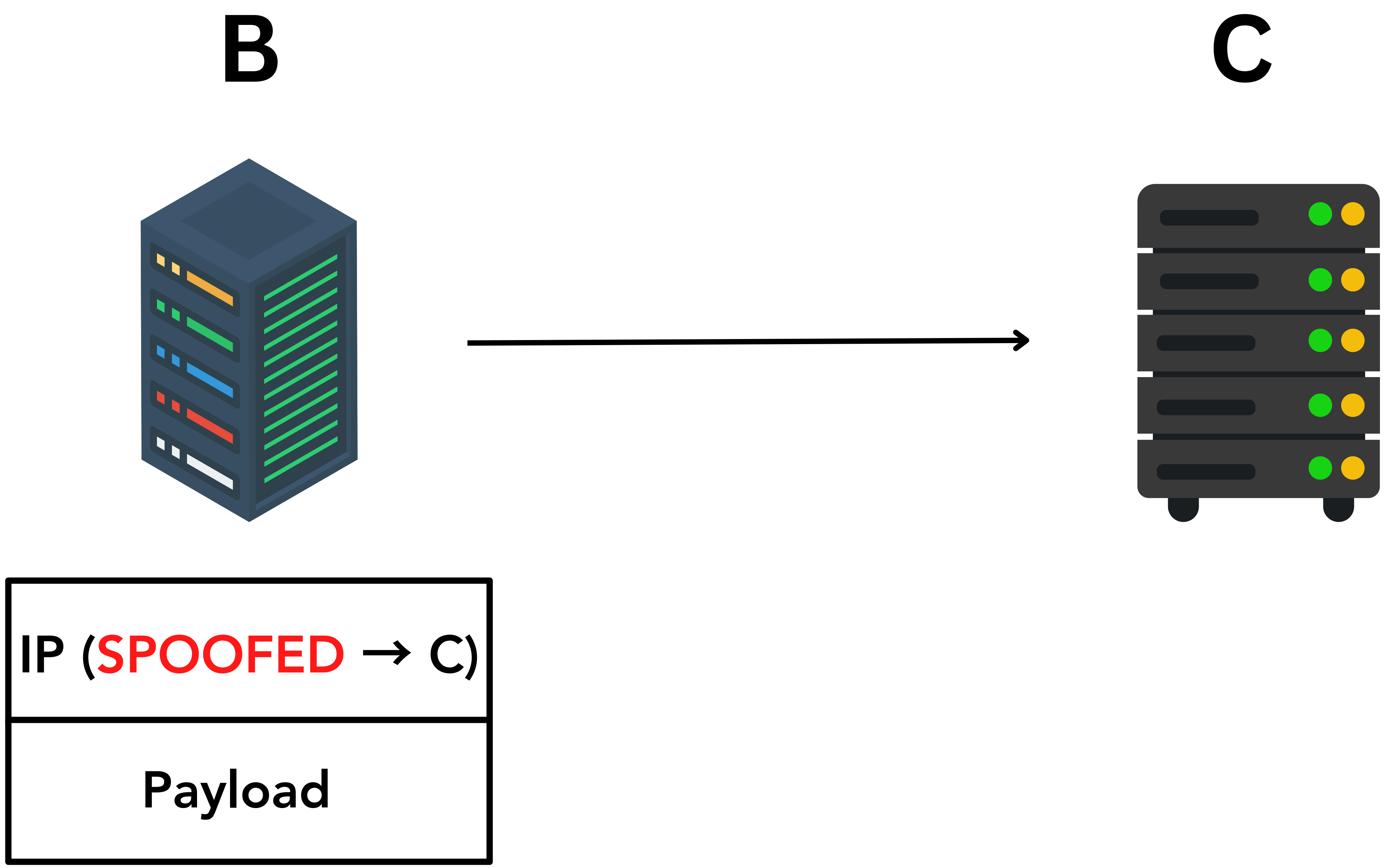
B

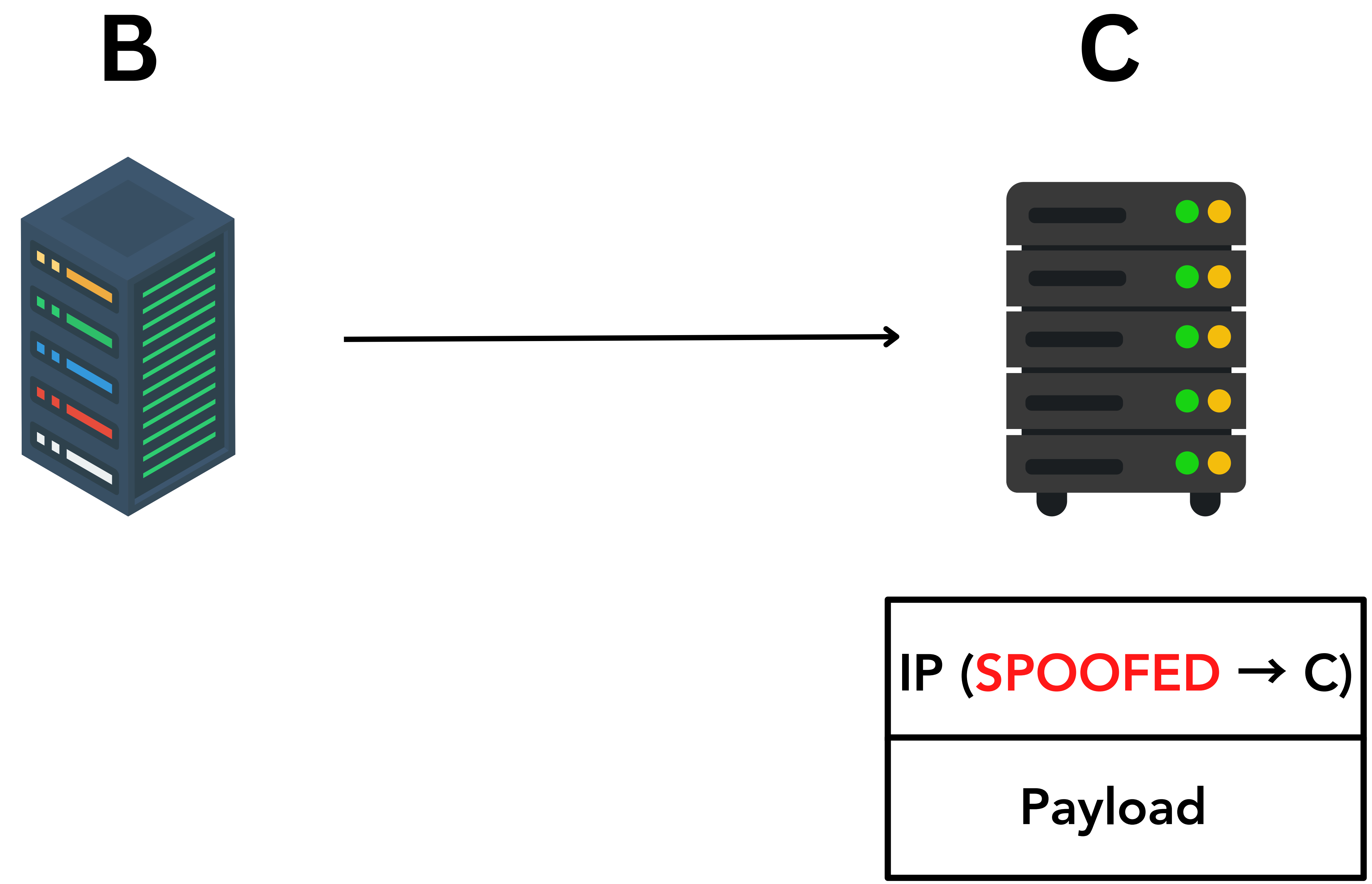
IP (A → B)
IP (B → C)
Payload


A
B
IP (B → C)
Payload


B
C
IP (B → C)
Payload

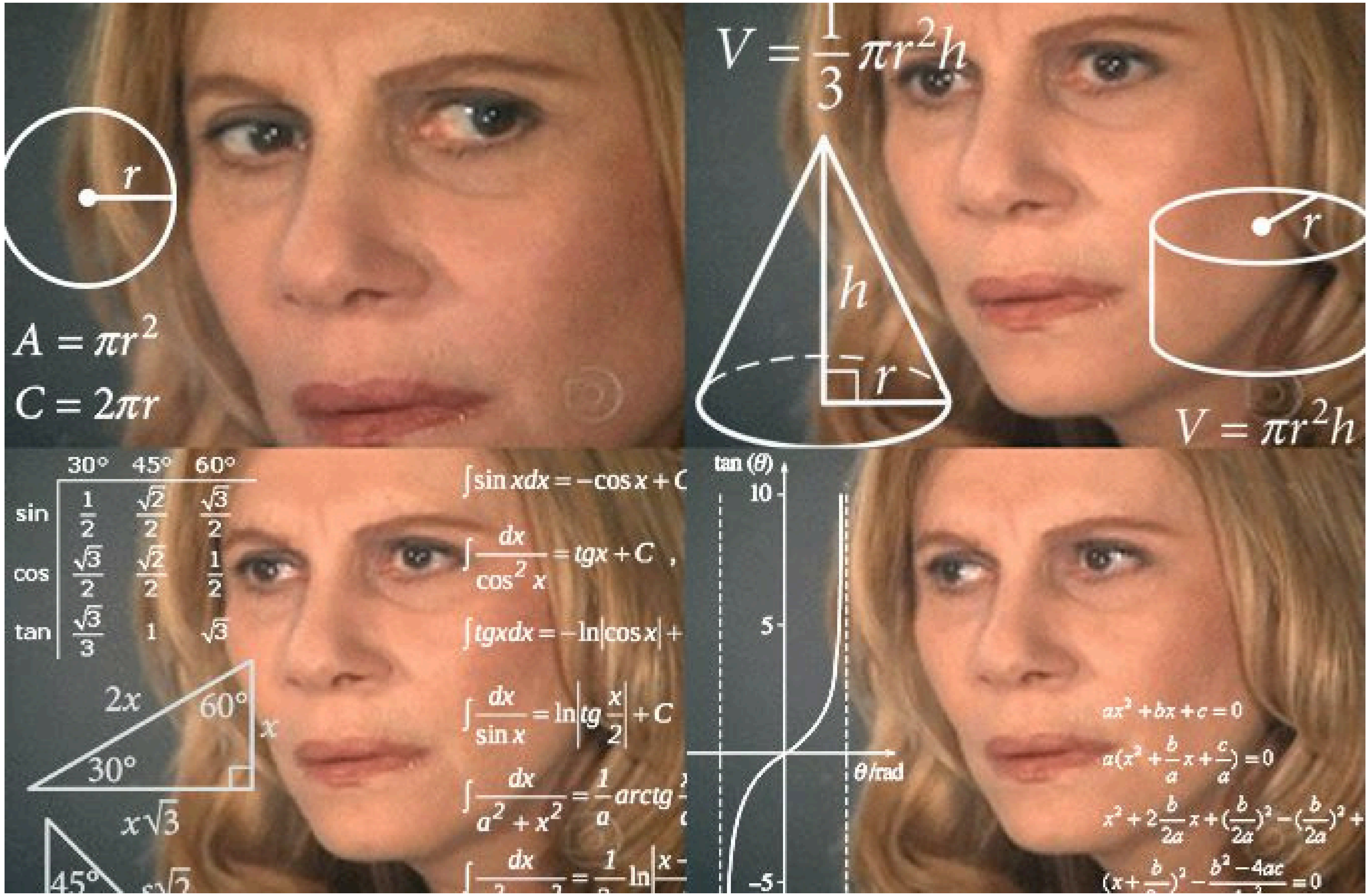

B
C
IP (B → C)
Payload


B
C
IP (SPOOFED → C)
Payload


B
C
IP (SPOOFED → C)
Payload

2020: 150k vulnerable

Cisco NX-OS Software vulnerable by default

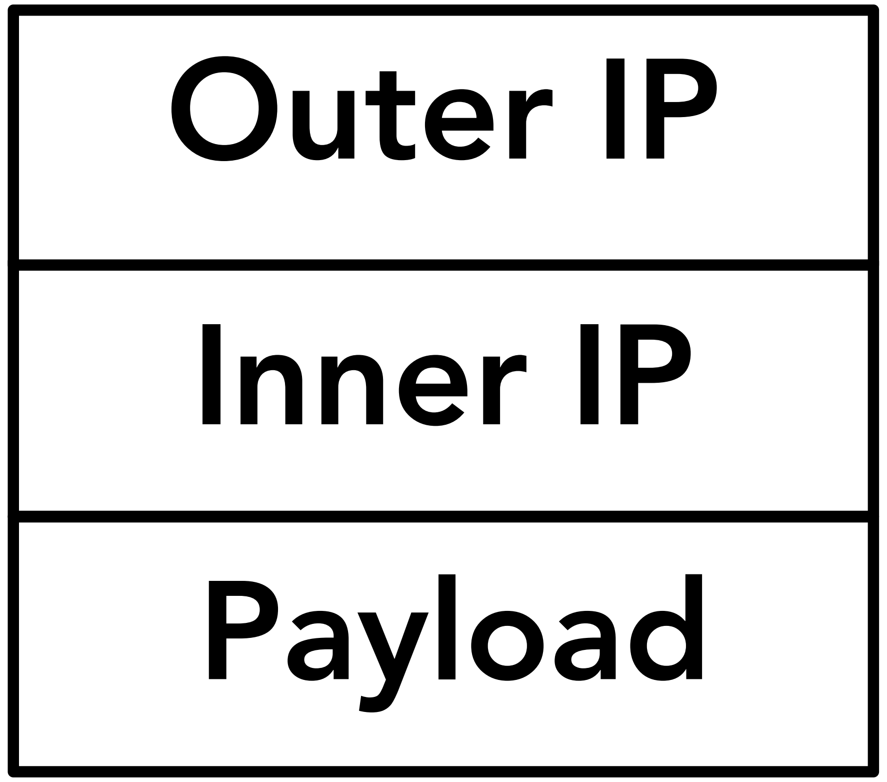

A = πr²
C = 2πr
r
V = 1/3 πr²h
h
r
r
V = πr²h
30° 45° 60°
sin
cos
tan
2x
60°
x
30°
x√3
tan (θ)
10
5
-5
θ/rad
ax² + bx + c = 0

The Culprits

The Culprits

IPIP GRE 6in4 4in6

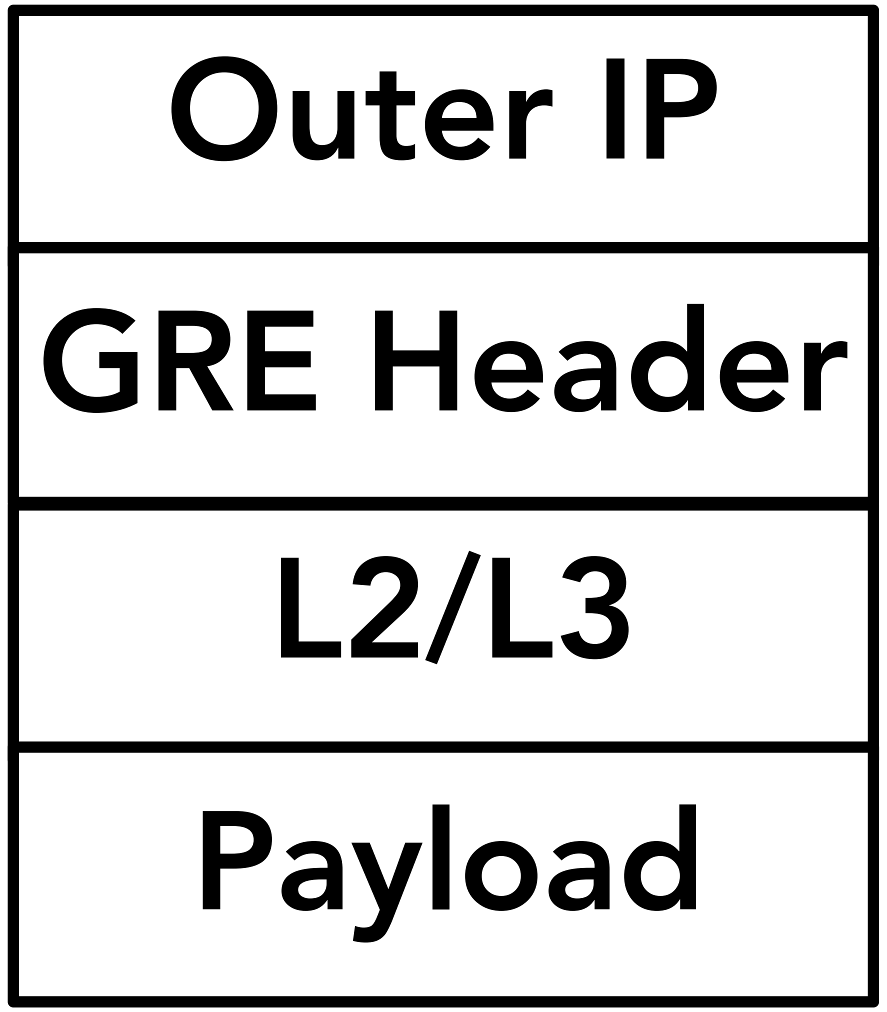
IPIP



- No encryption
- No authentication

IPIP GRE 6in4 4in6

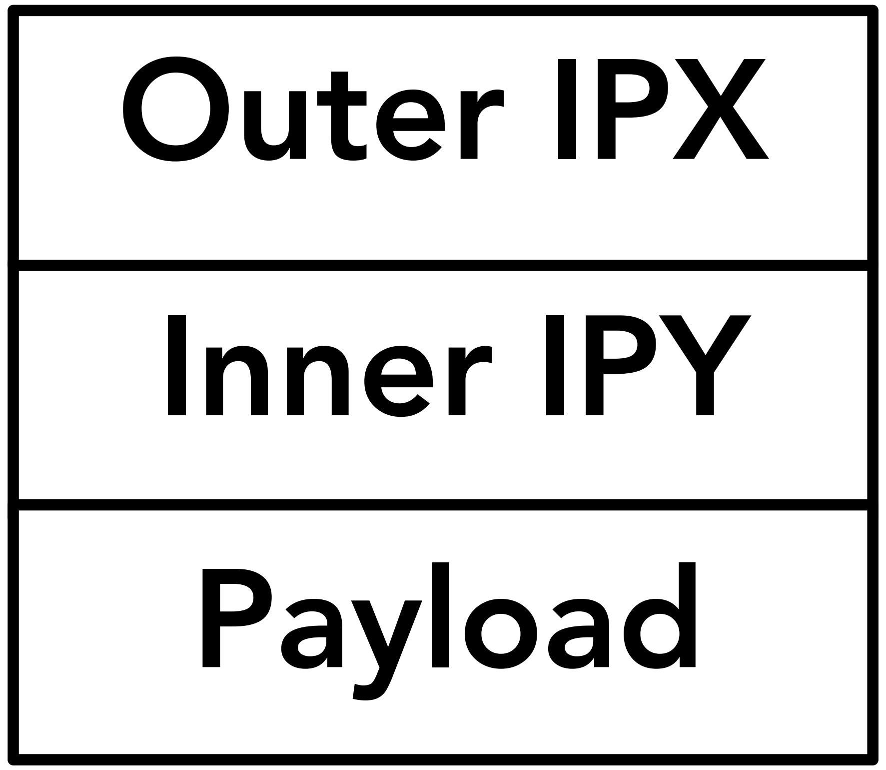
GRE



- **Adds GRE header**
- **Can encapsulate L2/L3**
- **4-byte optional key field**

IPIP GRE 6in4 4in6

6in4 4in6



- Transition to IPv6
- IPX packet in IPY networks

On Linux (GRE Example)

```
sysctl -w net.ipv4.ip_forward=1
modprobe gre
ip tunnel add gre10 mode gre local a.b.c.d remote x.y.g.f.e
ip link set gre10 up
ip addr add 192.168.5.1/24 dev gre10
```

On Linux (GRE Example)

```
sysctl -w net.ipv4.ip_forward=1
modprobe gre
ip tunnel add gre10 mode gre local a.b.c.d remote x.y.g.f.e
ip link set gre10 up
ip addr add 192.168.5.1/24 dev gre10
```

Accept packets from any source

On Linux (GRE Example)

```
sysctl -w net.ipv4.conf.all.accept_local=1
sysctl -w net.ipv4.ip_forward=1
modprobe gre
ip tunnel add gre10 mode gre local a.b.c.d ~~remote x.y.g.f.e~~
ip link set gre10 up
ip addr add 192.168.5.1/24 dev gre10
```

Accept packets with local source addresses

Anonymise DoS Attacks

Give access to private networks

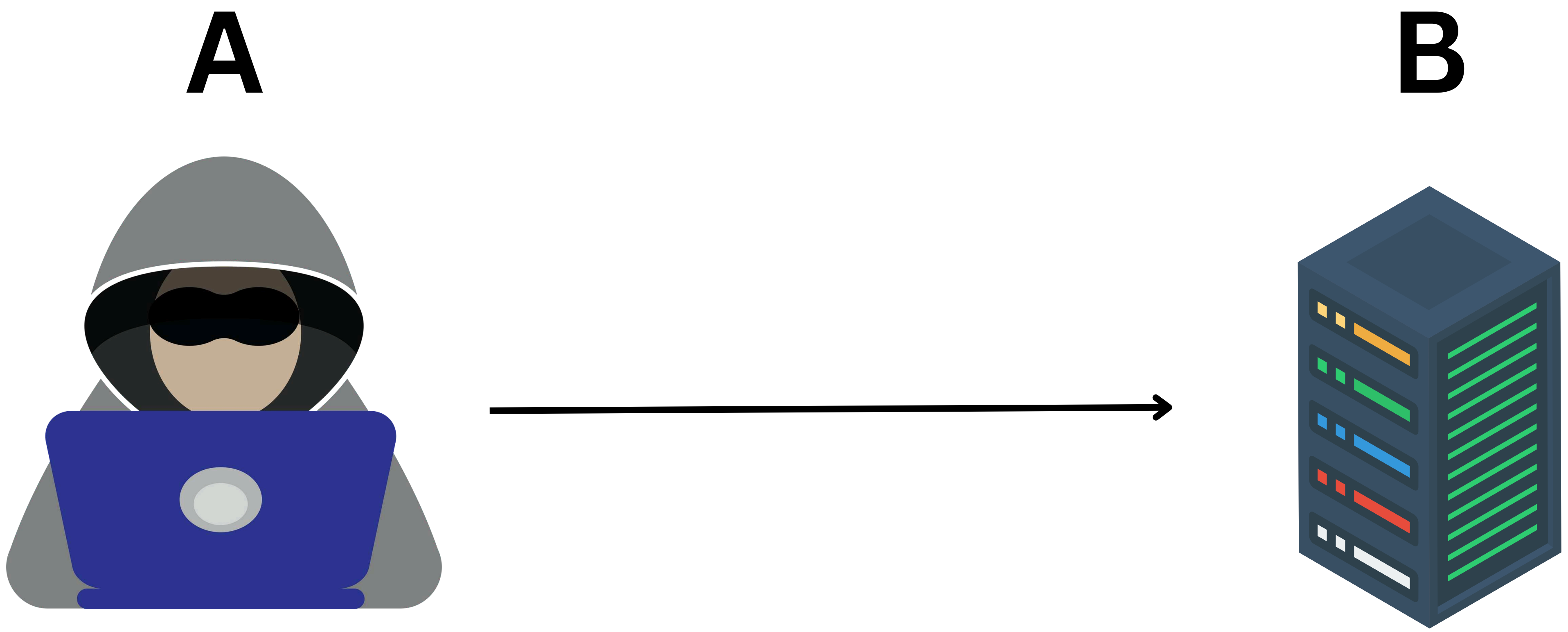
Amplify traffic

Scanning Methods

The Scanning Process

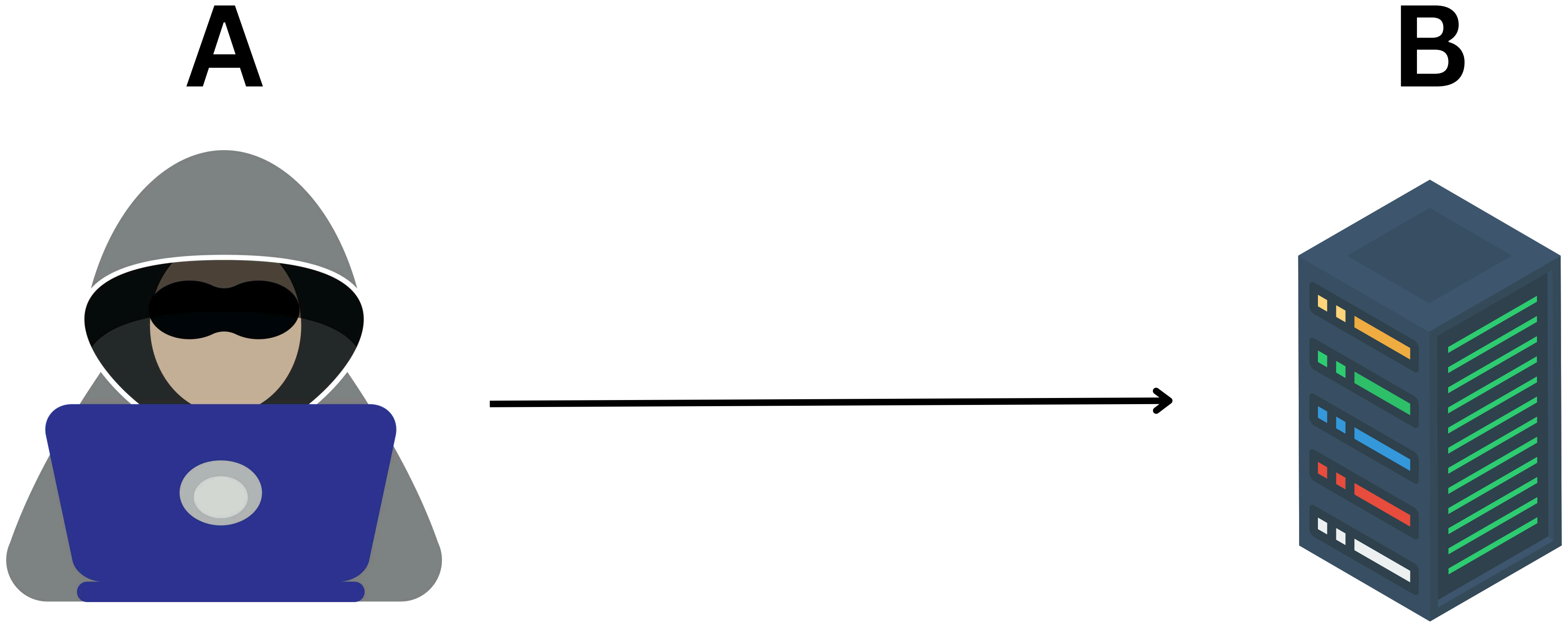
- Extended ZMap
- 26 scans
- Many scan subtypes (see paper)
- For v6 variants → IPv6 Hitlist addresses

Standard Scan



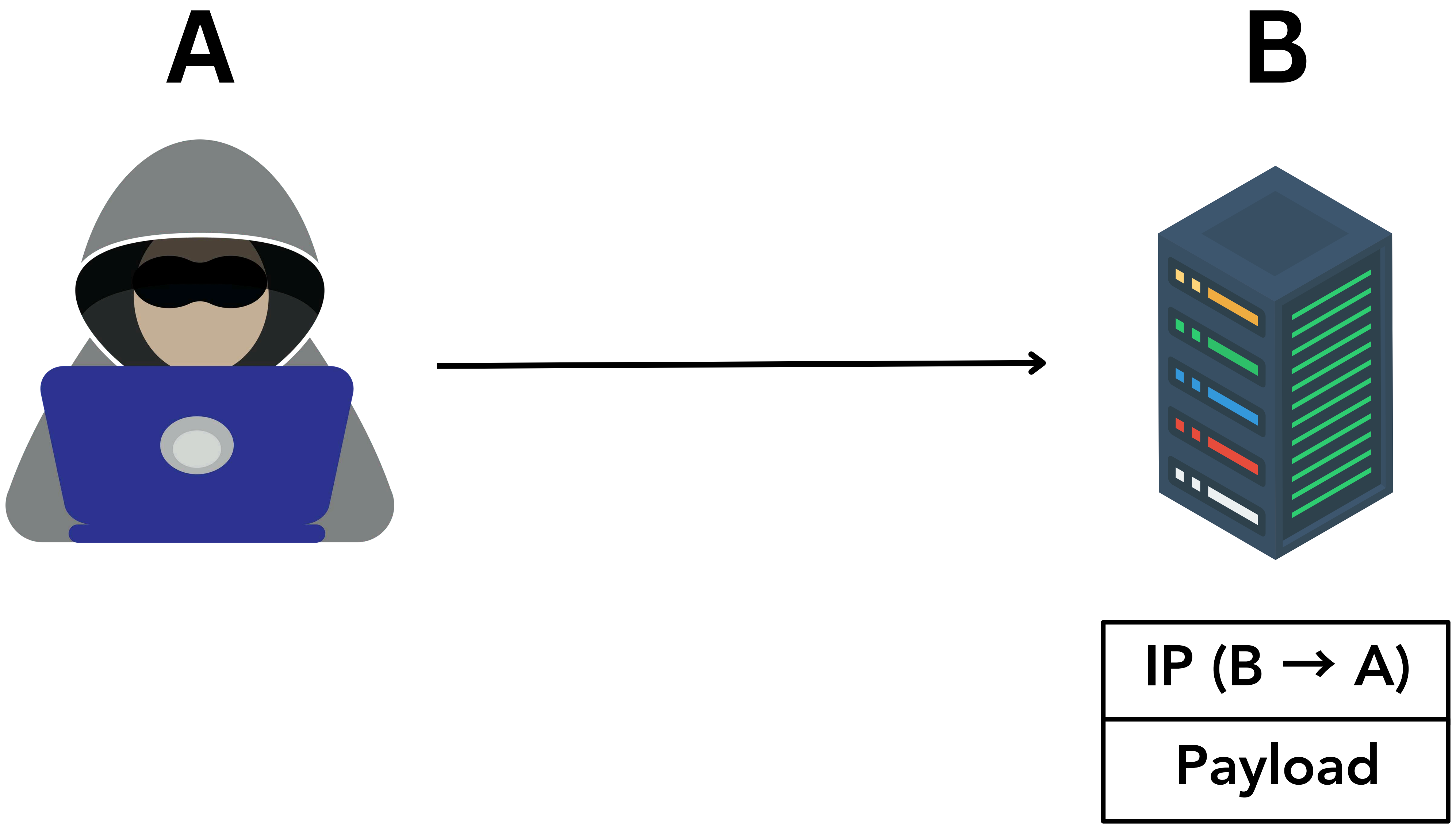
IP (A → B)
GRE Header
IP (B → A)
Payload

Standard Scan

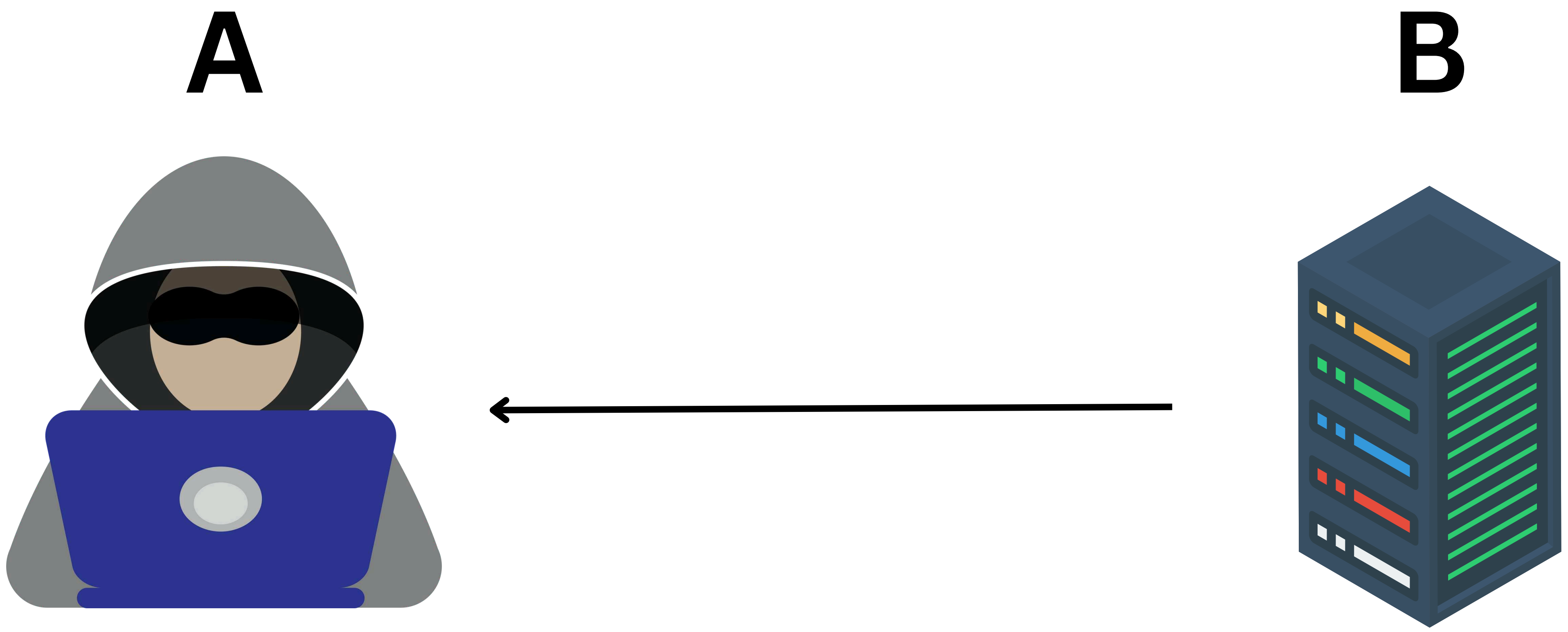


IP (A → B)
GRE Header
IP (B → A)
Payload

Standard Scan



Standard Scan

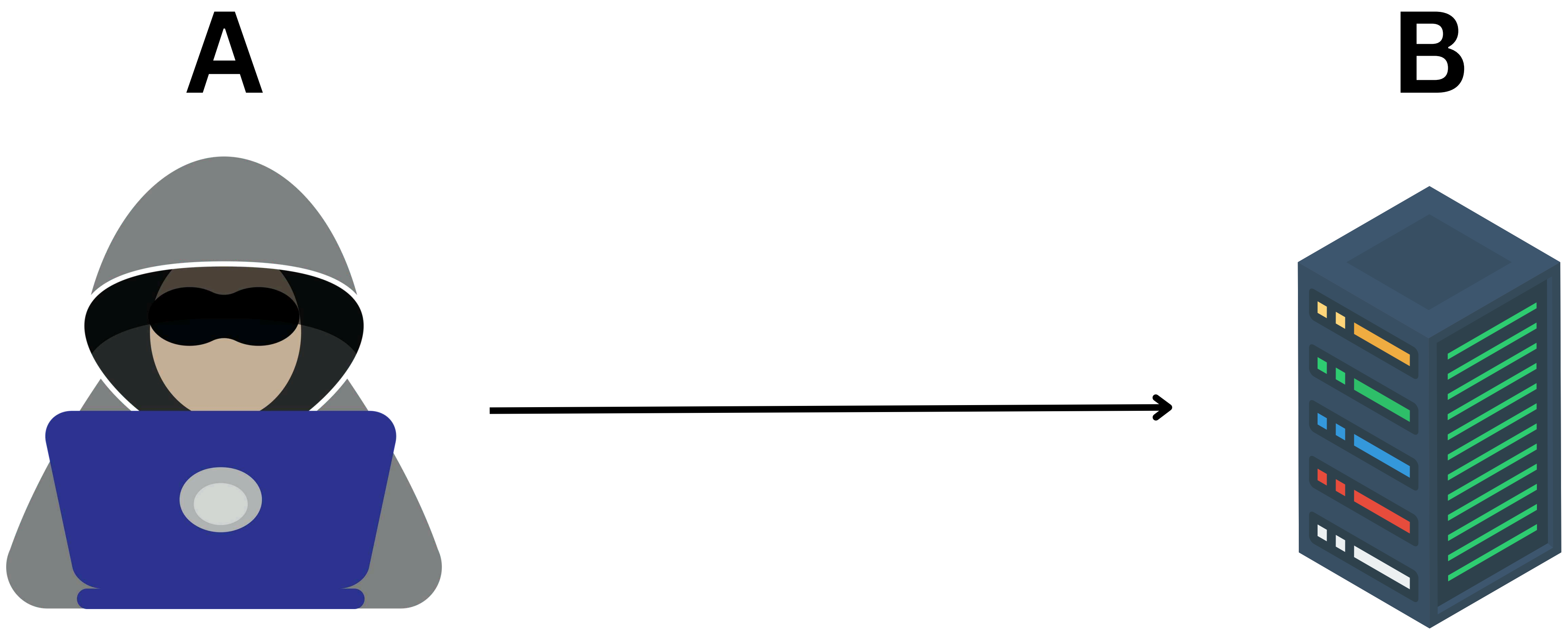


IP (B → A)
Payload

Standard Scan

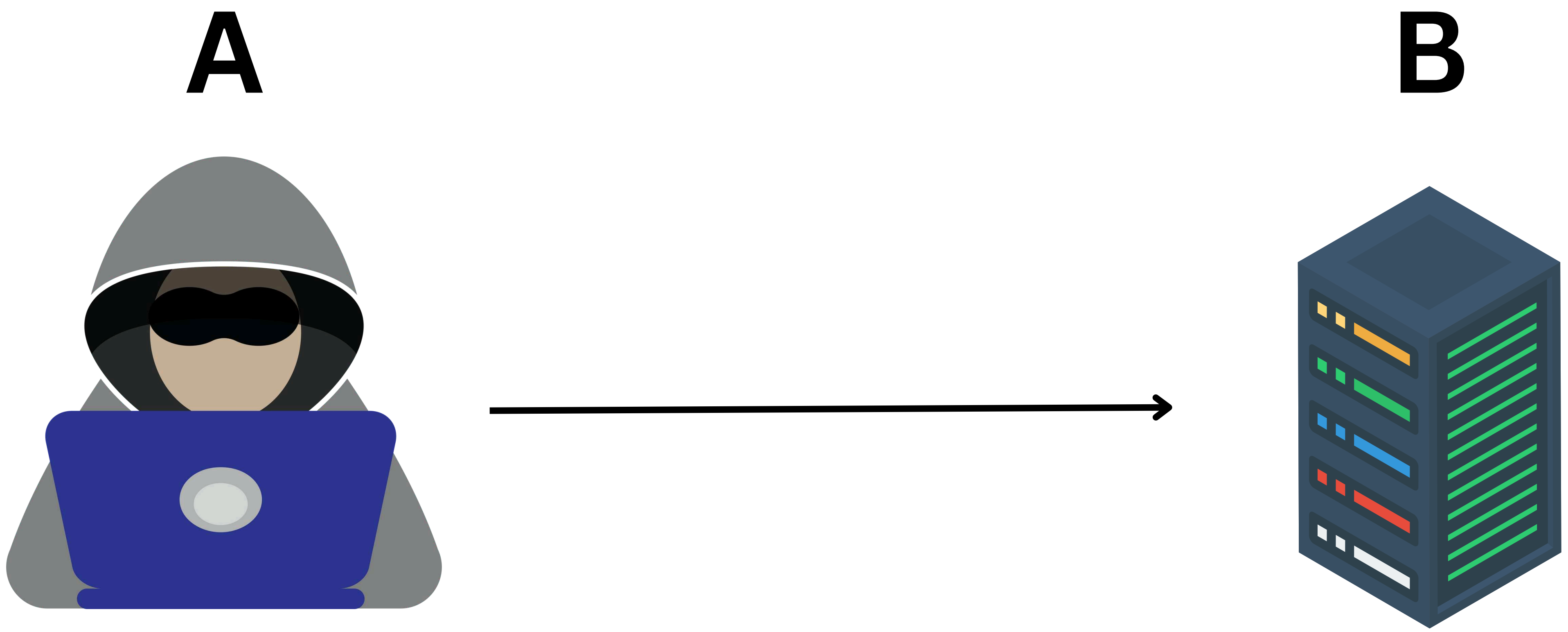
- Decapsulate from any source
- Forward traffic to destination
- Can spoof (spoof scans)

ICMP Echo Scan



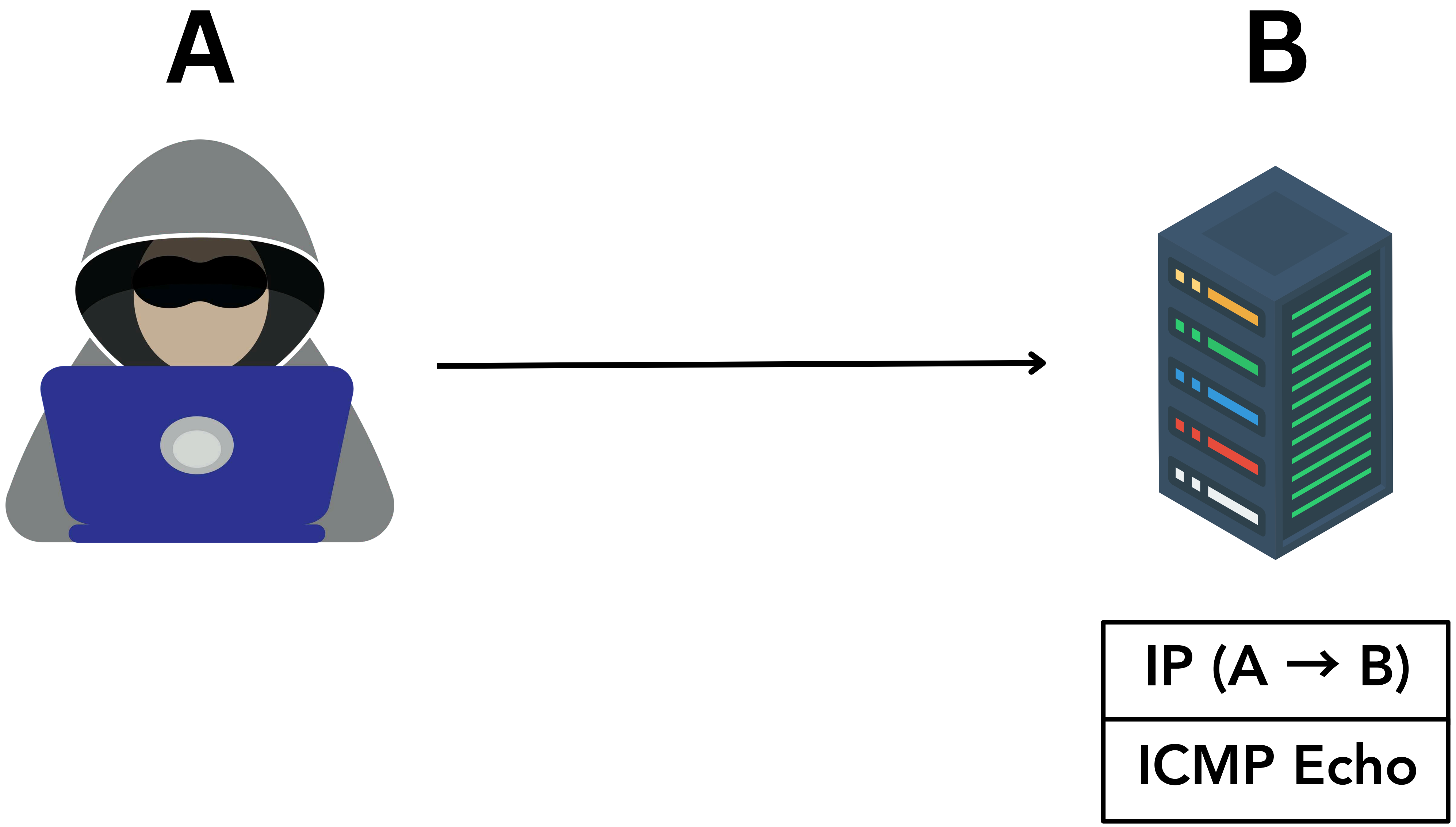
IP (A → B)
GRE Header
IP (A → B)
ICMP Echo

ICMP Echo Scan

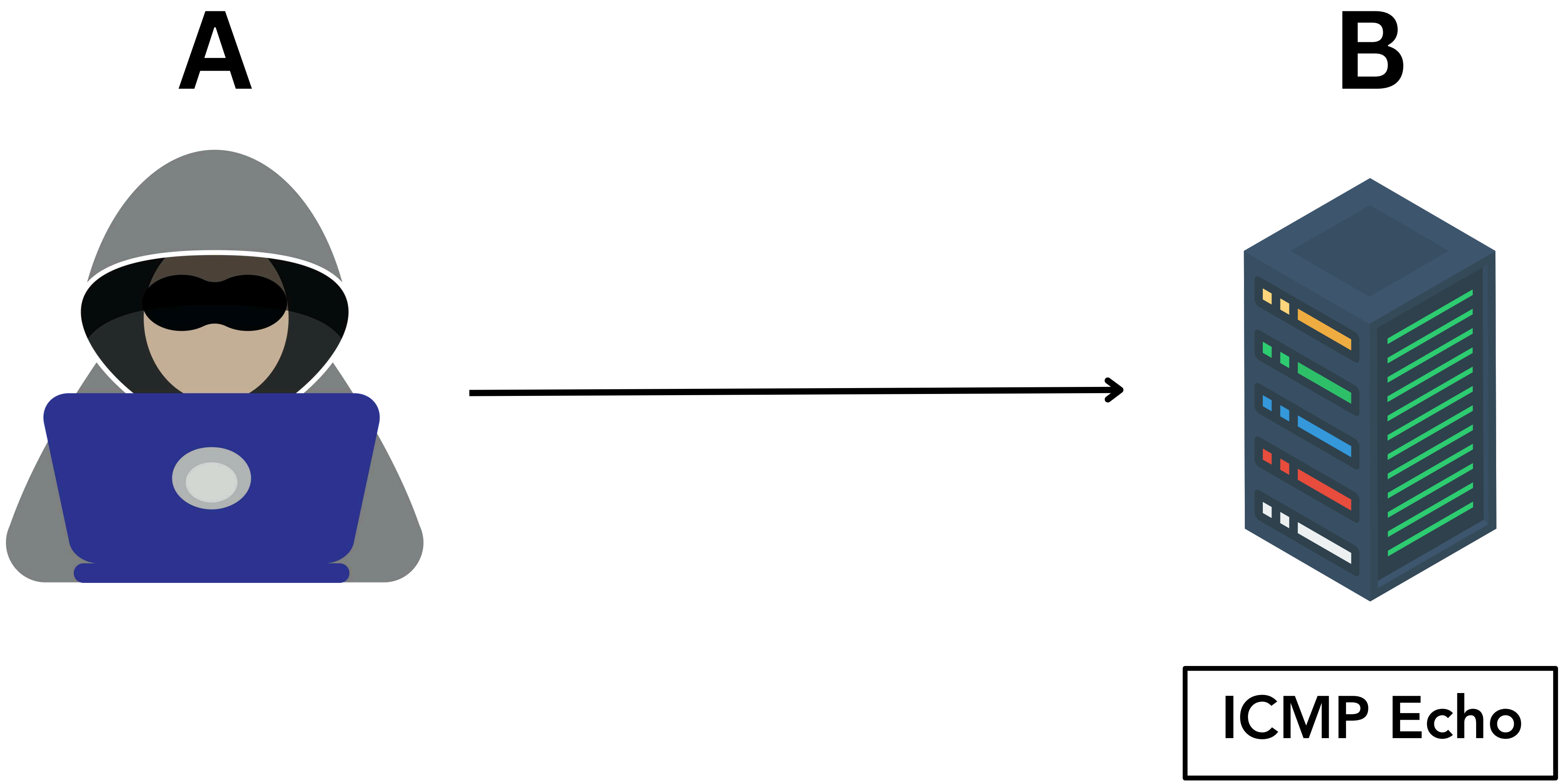


IP (A → B)
GRE Header
IP (A → B)
ICMP Echo

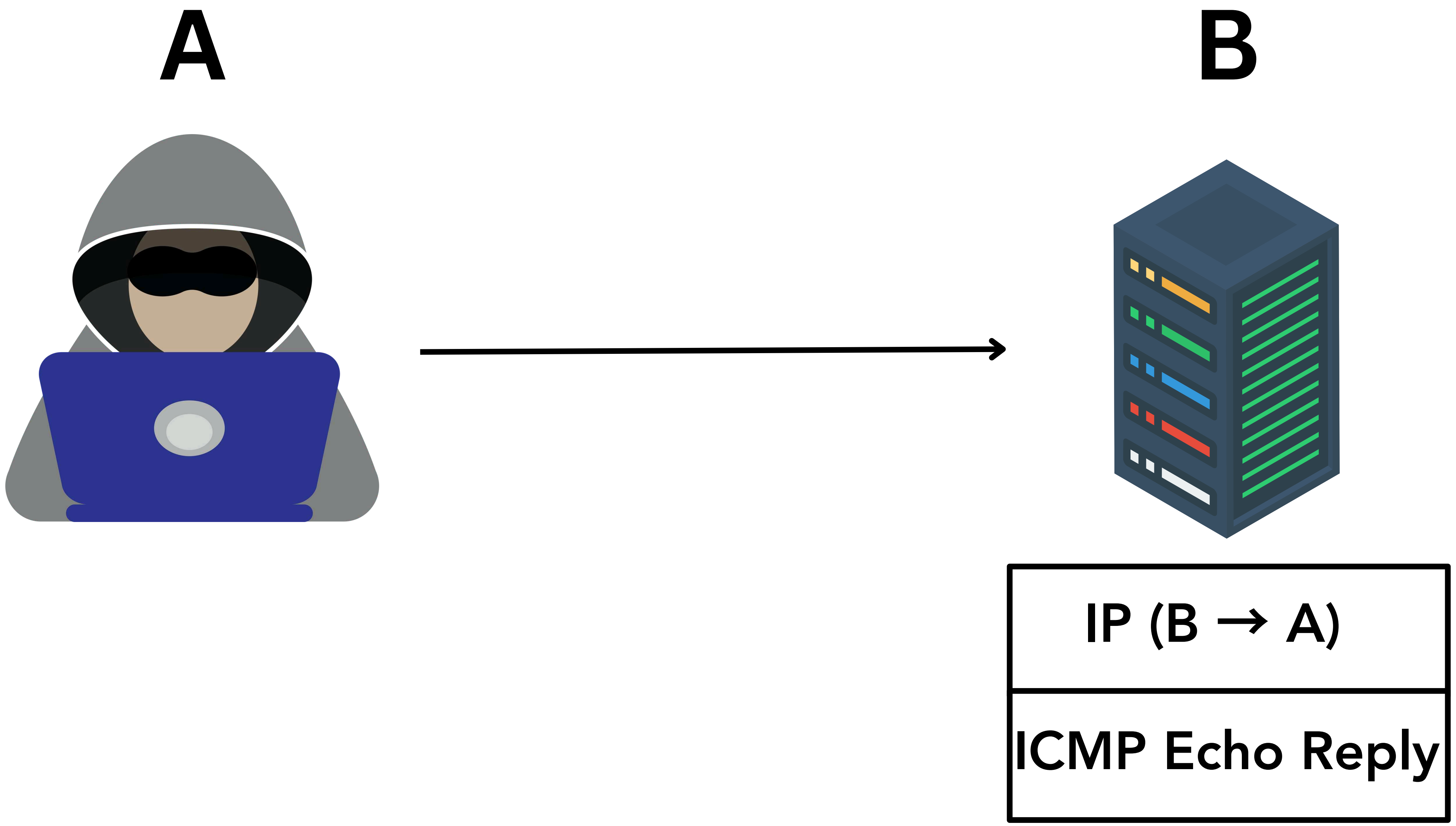
ICMP Echo Scan



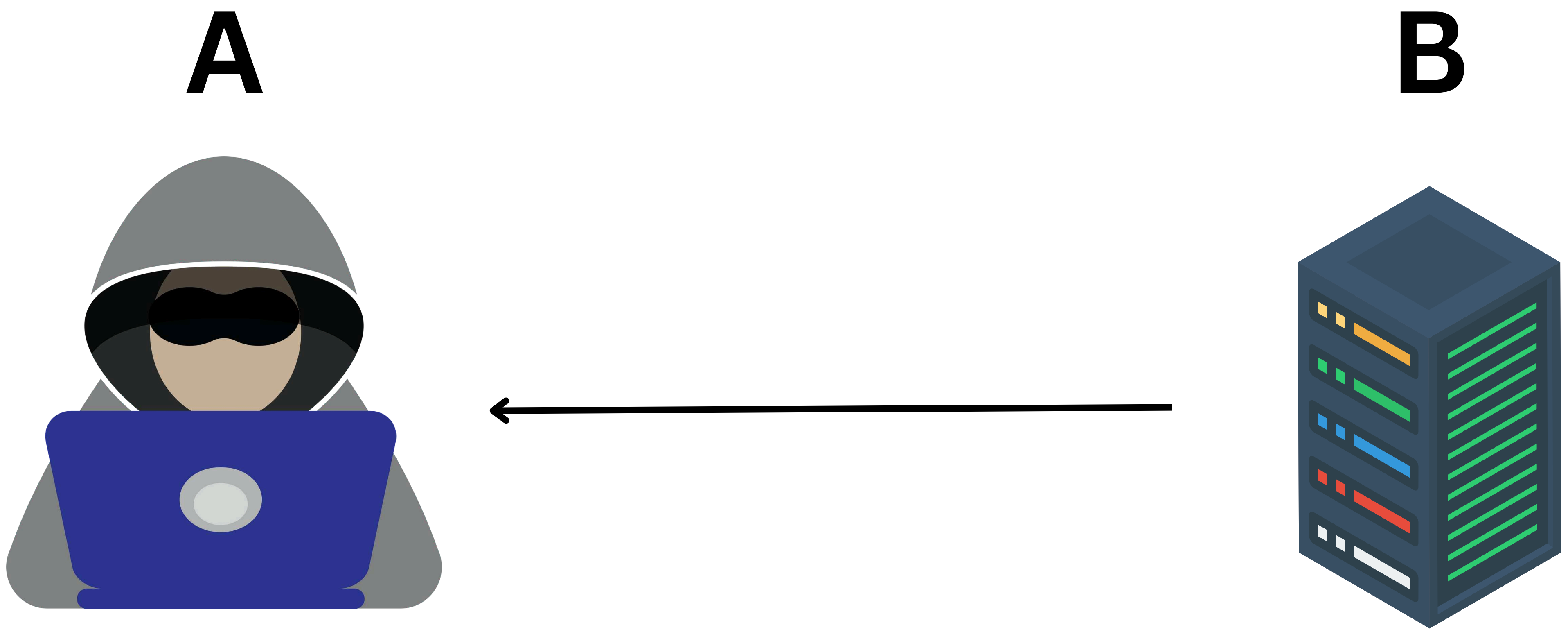
ICMP Echo Scan



ICMP Echo Scan



ICMP Echo Scan

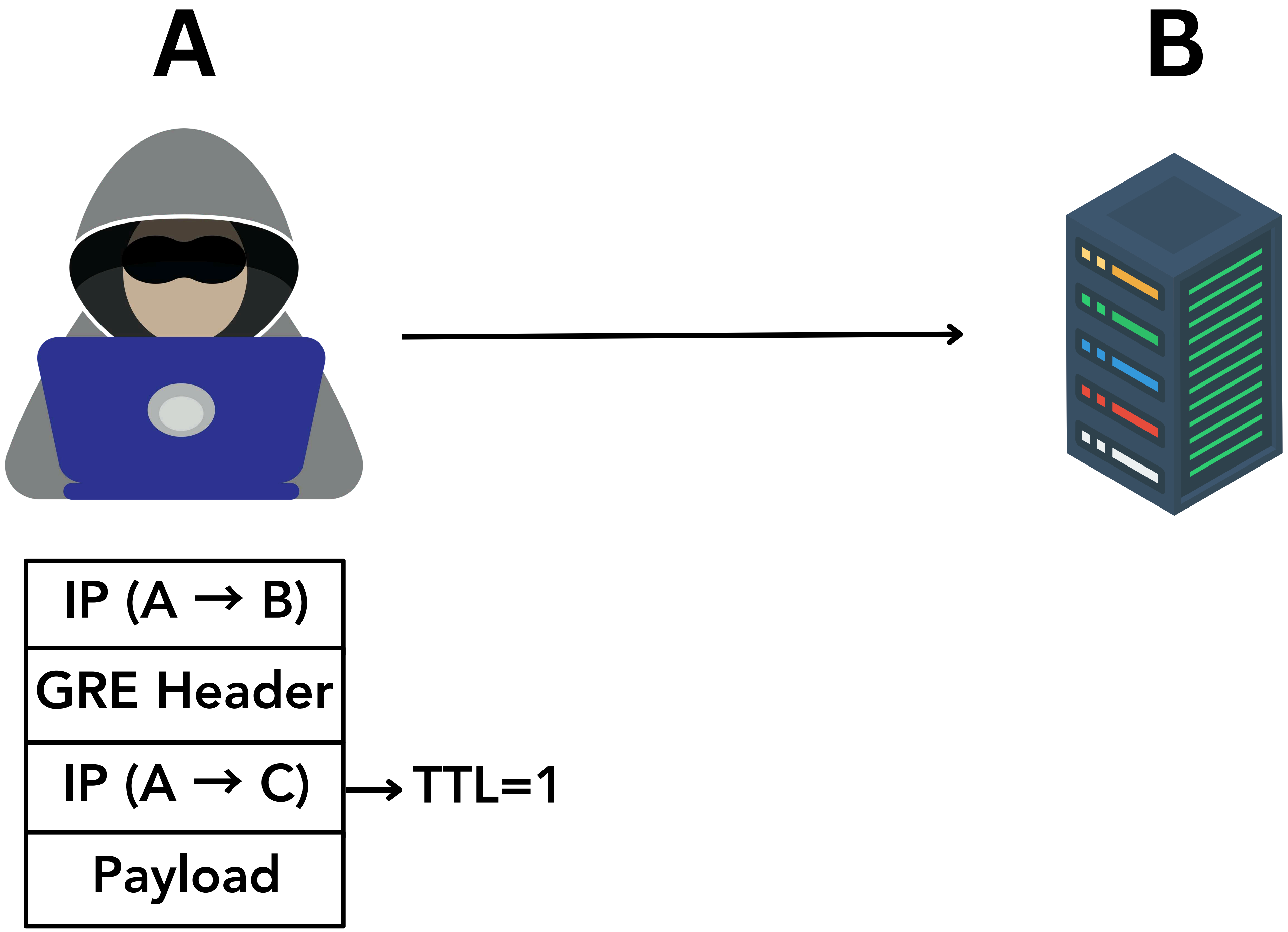


IP (B → A)
ICMP Echo Reply

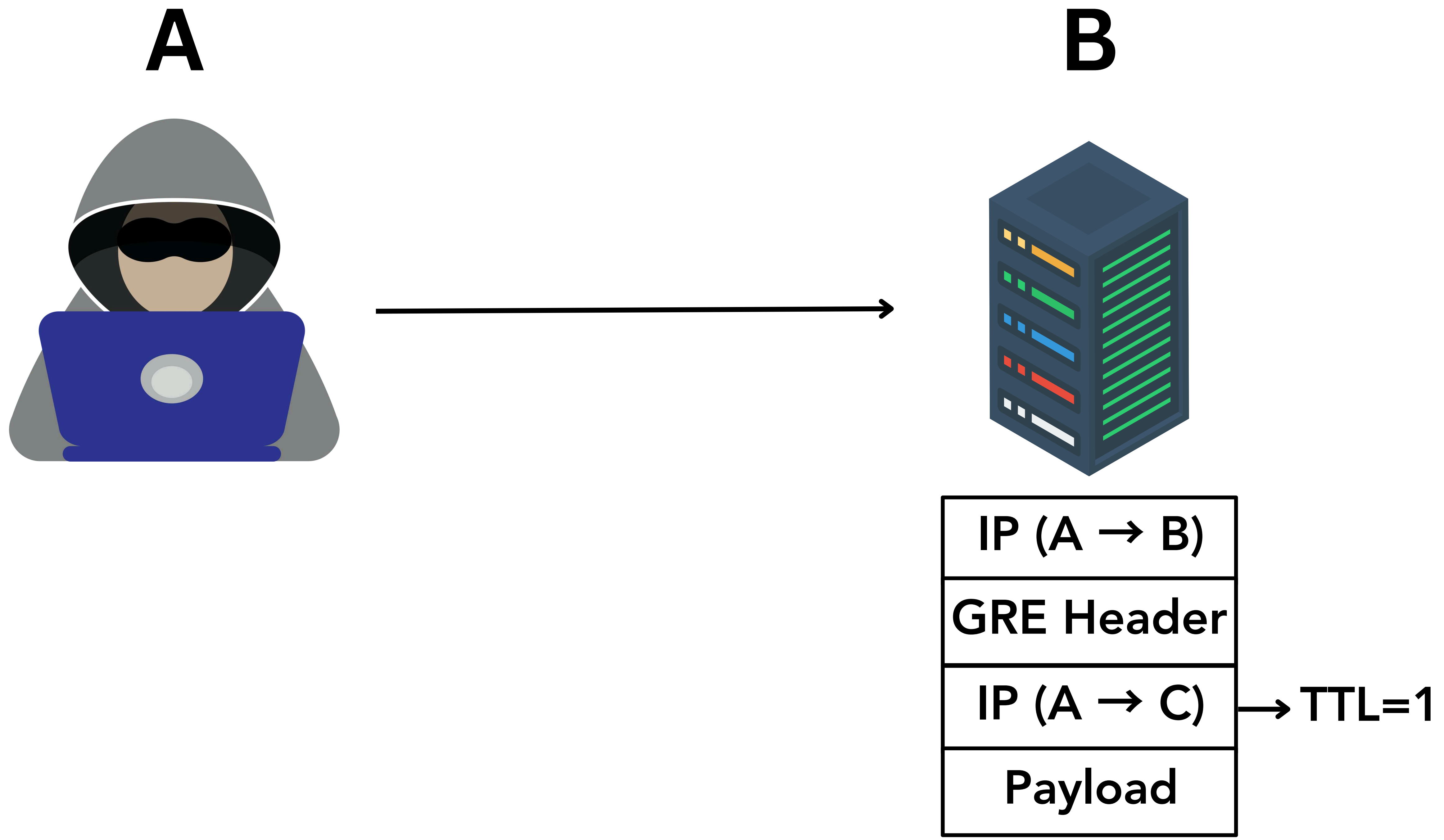
ICMP Echo Scan

- **Decapsulate from any source**
- **May not forward (to anyone)**

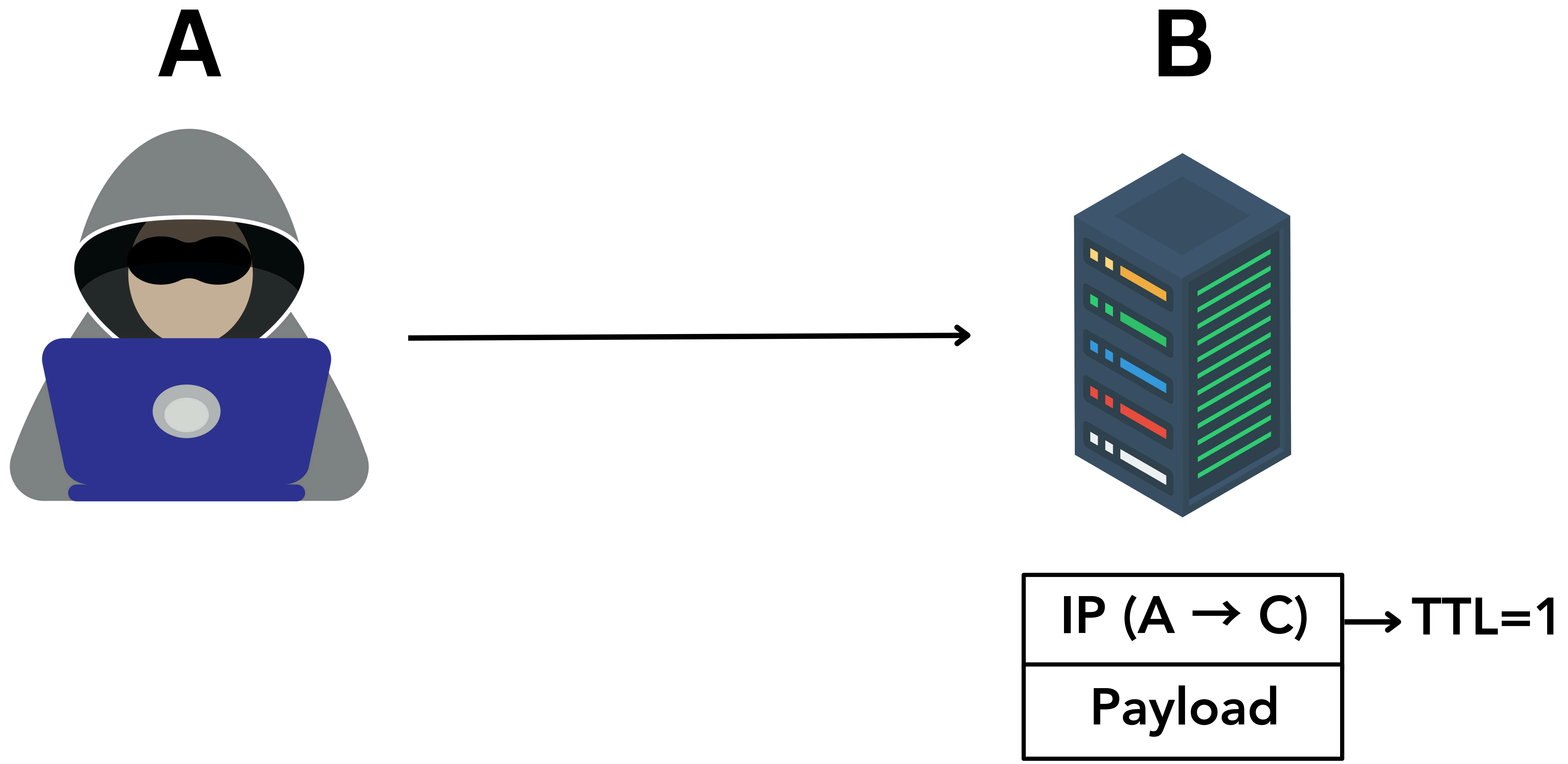
TTL Expired Scan



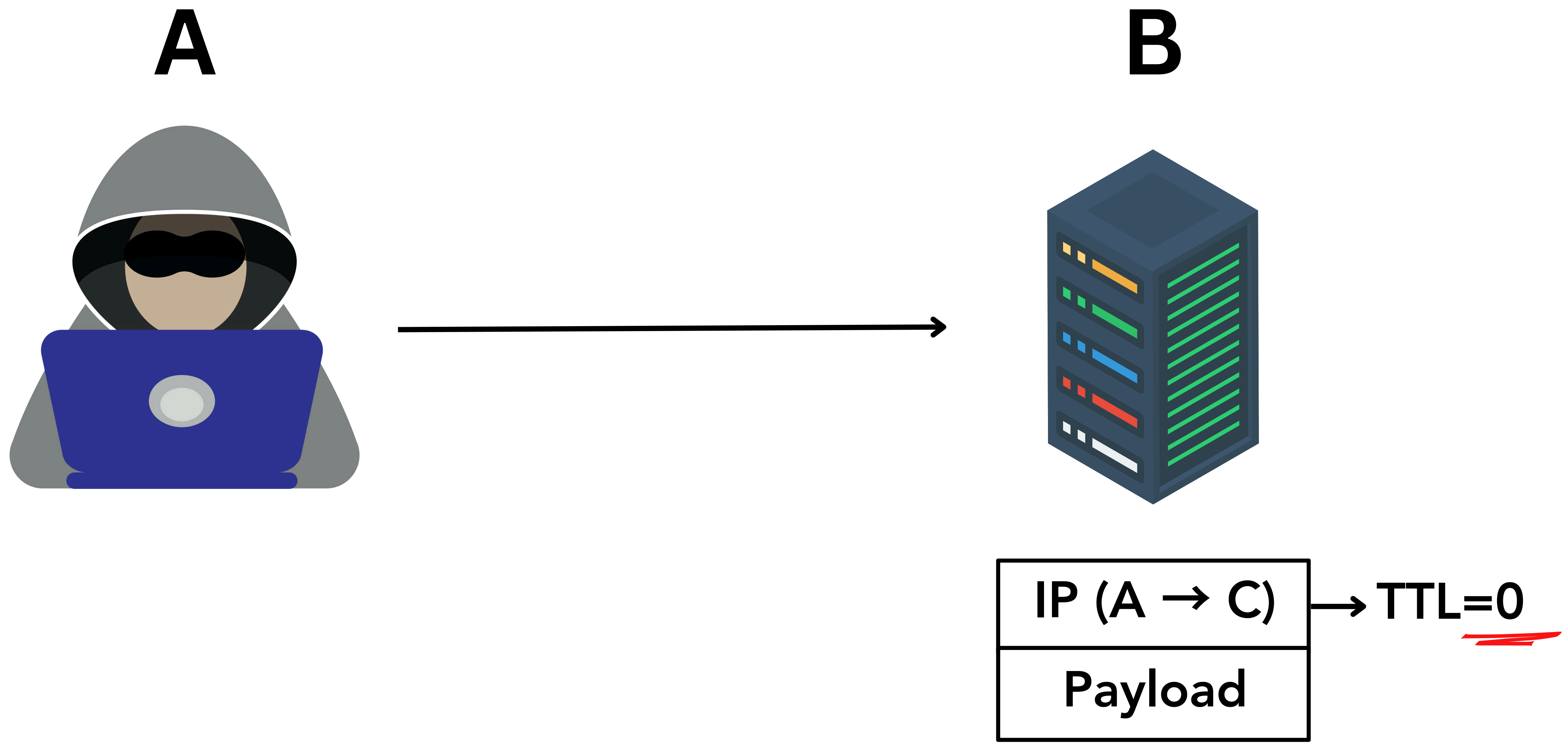
TTL Expired Scan



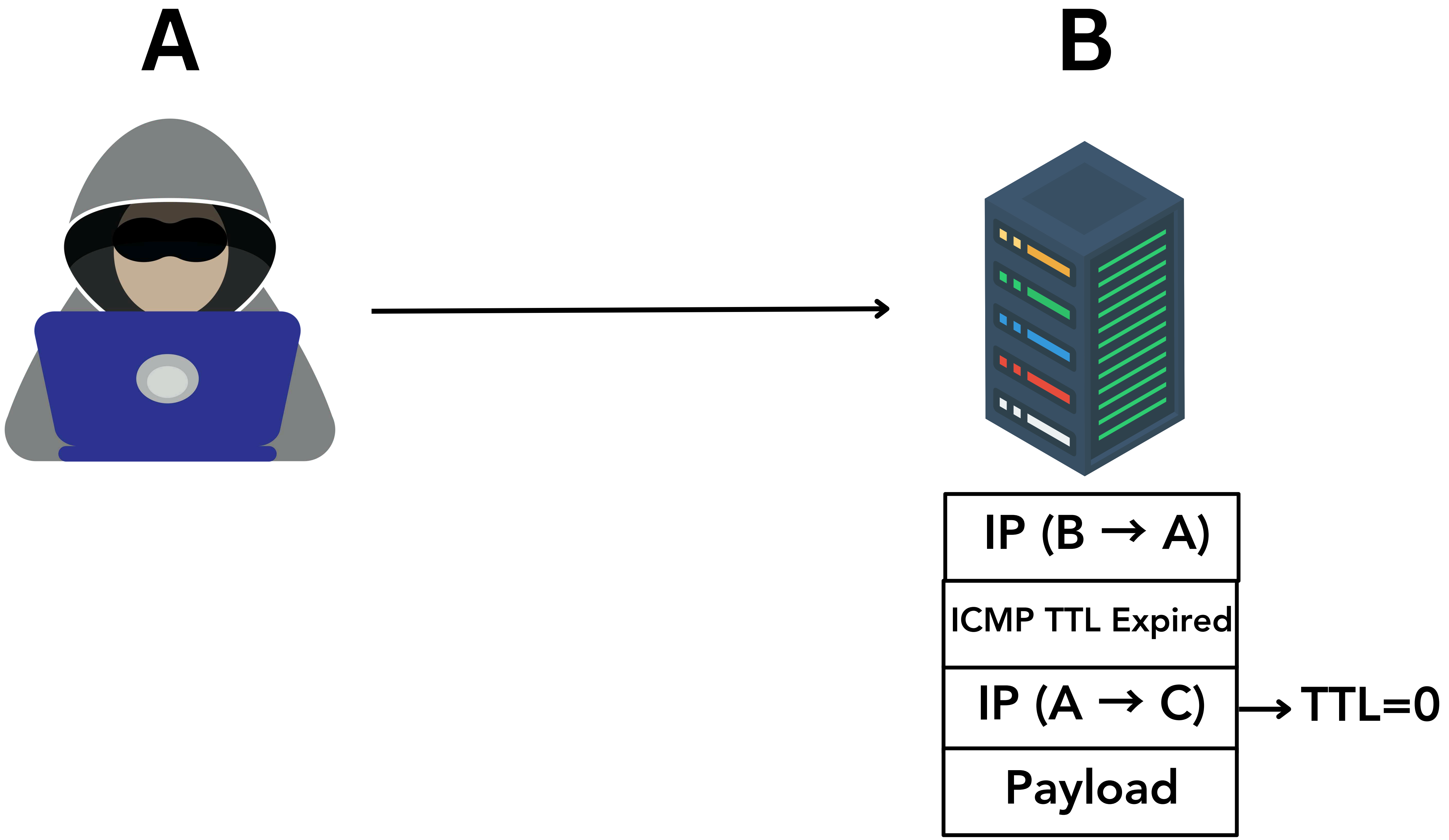
TTL Expired Scan



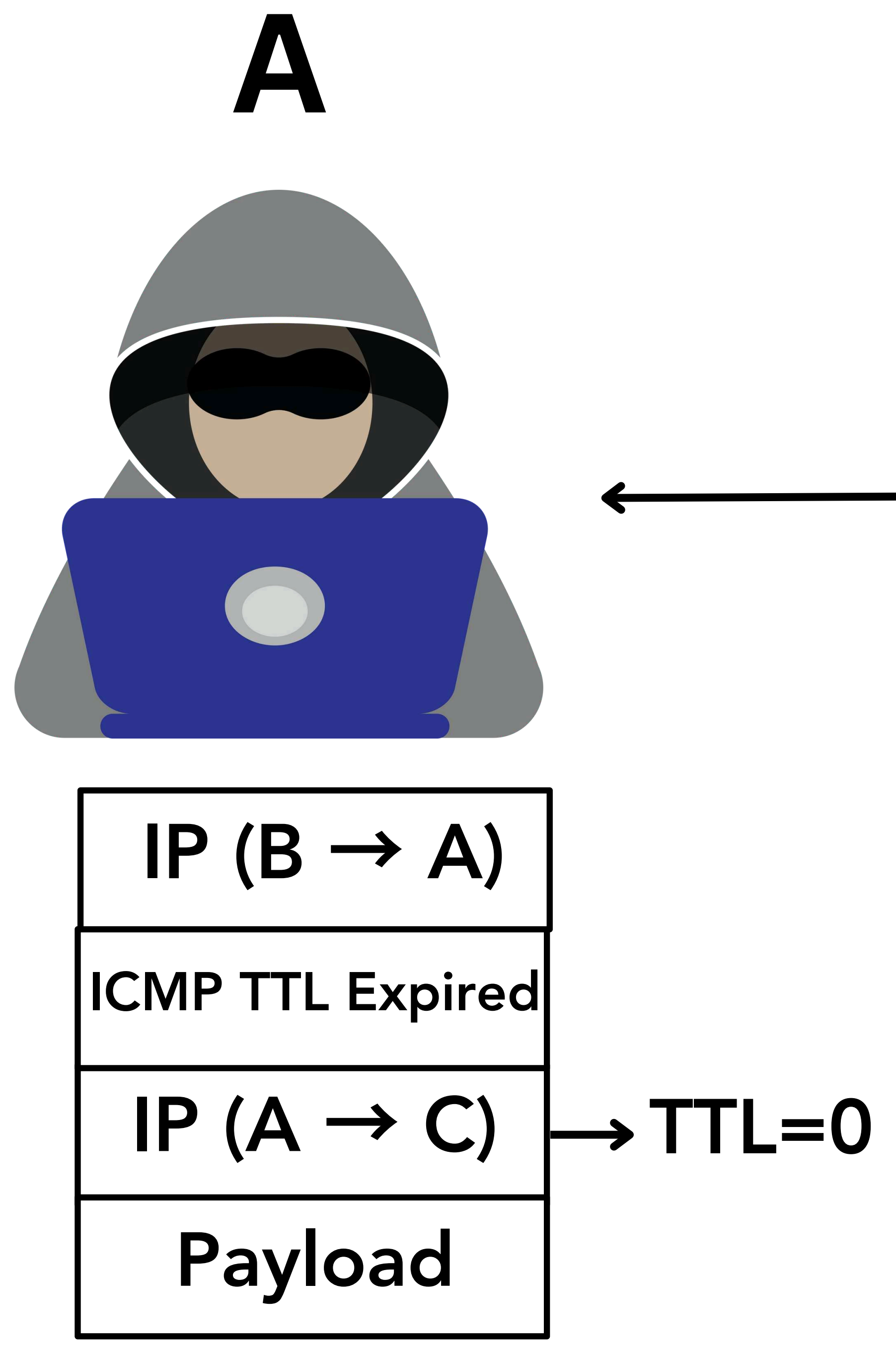
TTL Expired Scan



TTL Expired Scan



TTL Expired Scan



TTL Expired Scan

- Decapsulate from any source
- May not forward (to anyone)
- May block ping responses
- Behind NAT

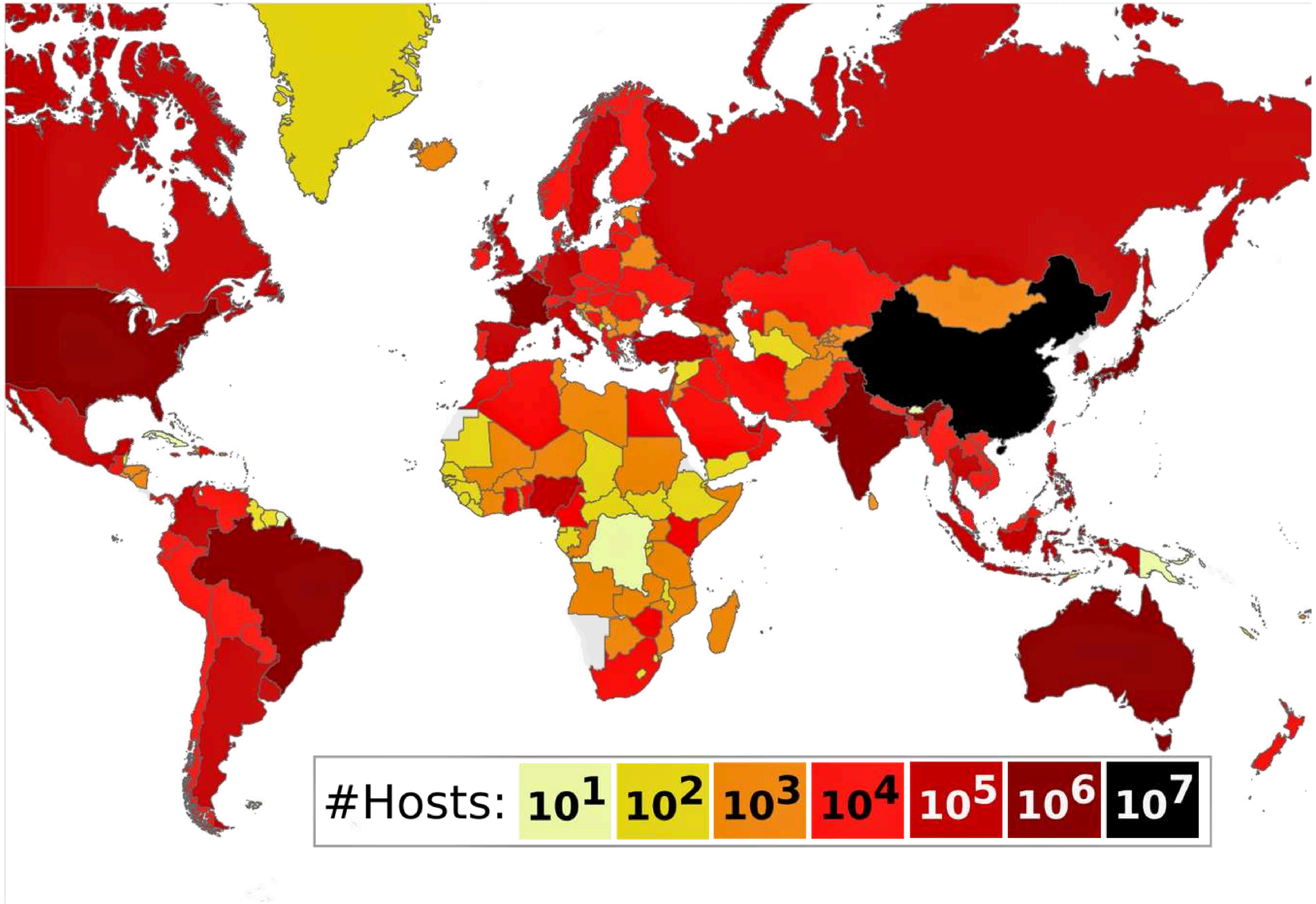
Results

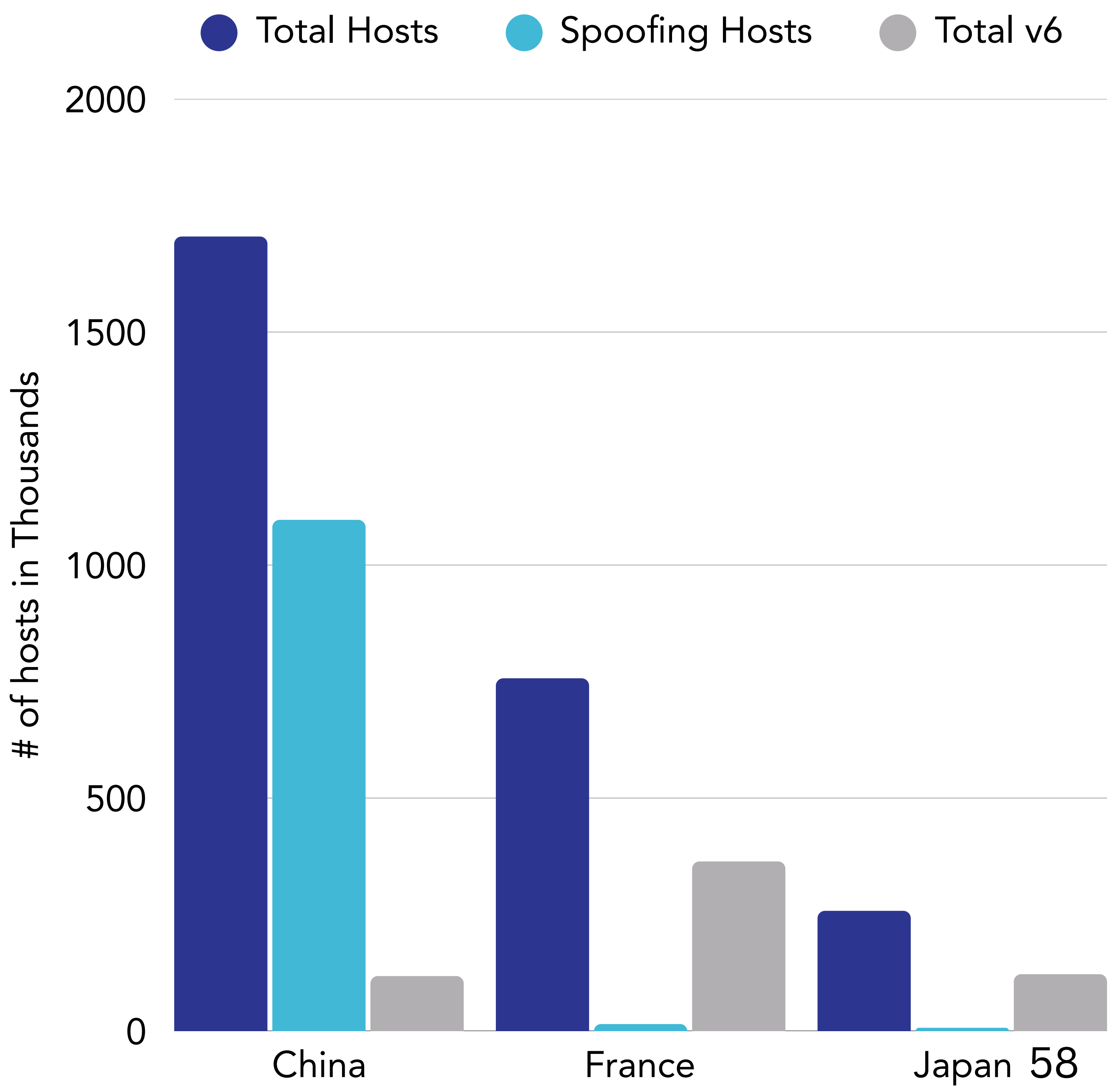
4,263,193

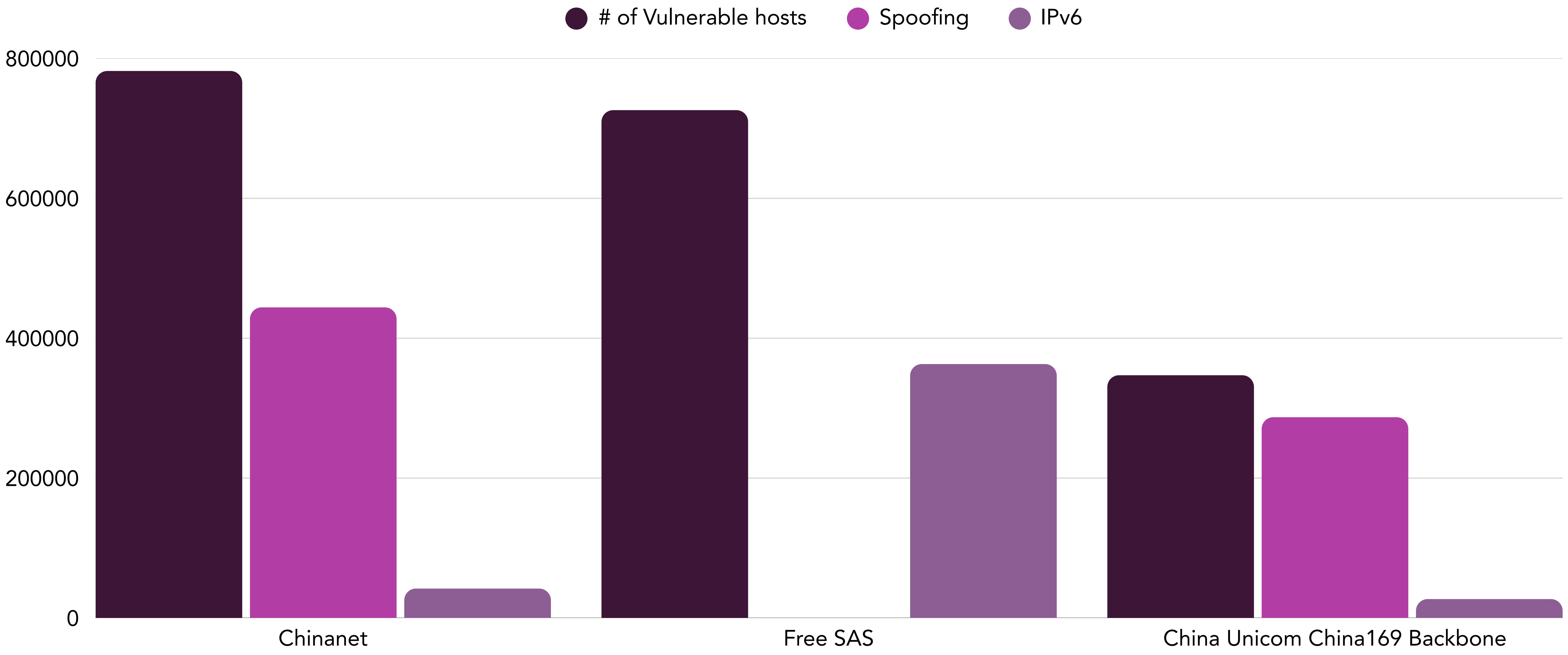
11,027 ASs

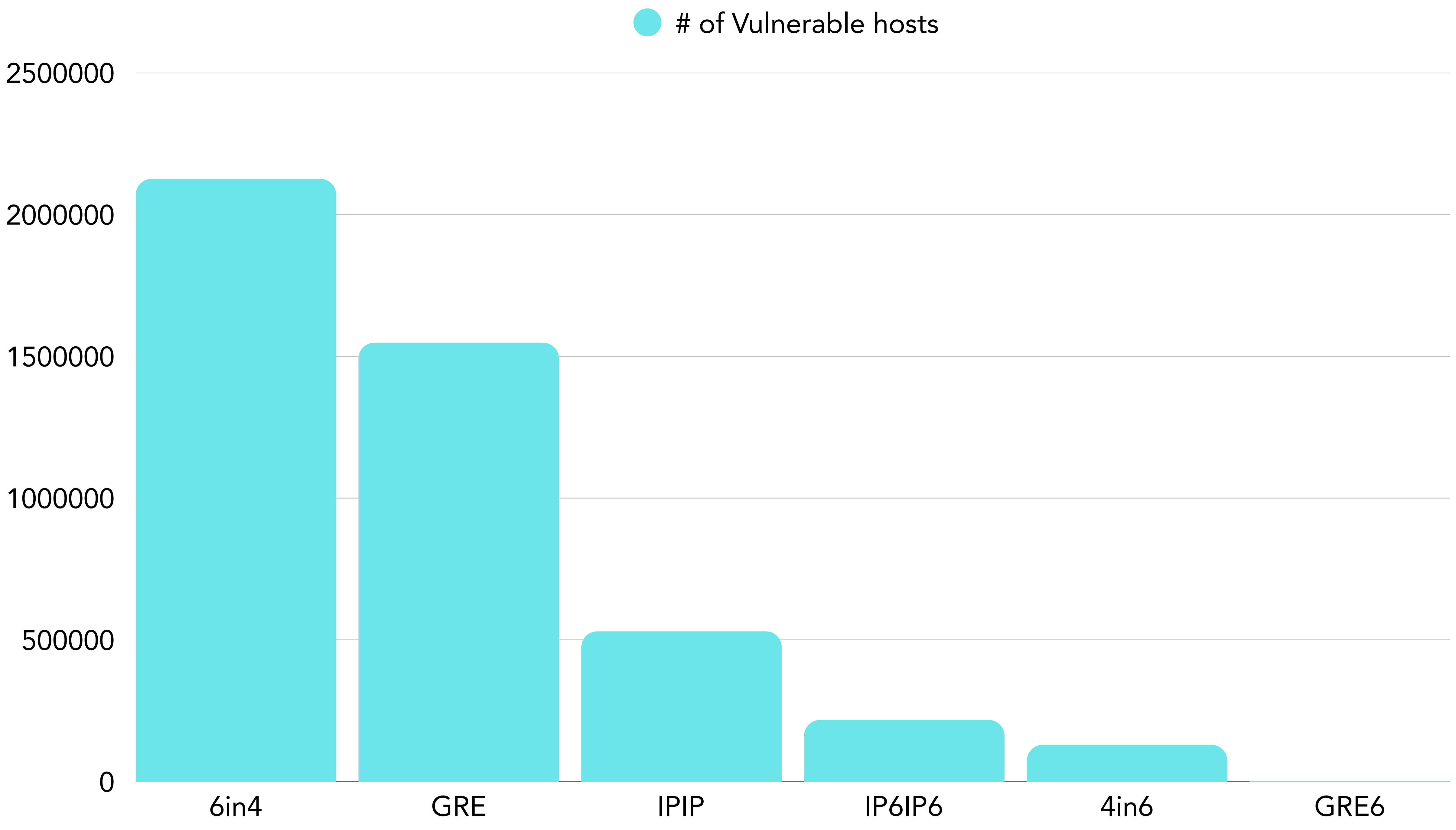
1,858,892

4,276 ASs


#Hosts: 10^1 10^2 10^3 10^4 10^5 10^6 10^7


Total Hosts
Spoofing Hosts
Total v6
# of hosts in Thousands
2000
1500
1000
500
0
China
France
Japan


# of Vulnerable hosts
Spoofing
IPv6
800000
600000
400000
200000
0
Chinanet
Free SAS
China Unicom China169 Backbone



# of Vulnerable hosts
2500000
2000000
1500000
1000000
500000
0
6in4
GRE
IPIP
IP6IP6
4in6
GRE6

Open ports

- IPIP & IP6IP6 → HTTP(s)
- GRE → BGP, GTP-C, GTP-U
- 6in4 → NTP, SNMP

Domains

- IPIP & IP6IP6 → Mainly CDNs
- GRE & 6in4 → ISP domains

The Disclosure

Disclosed to CERT/CC

CVE-2024-7596, CVE-2024-7595, CVE-2025-23018, CVE-2025-23019

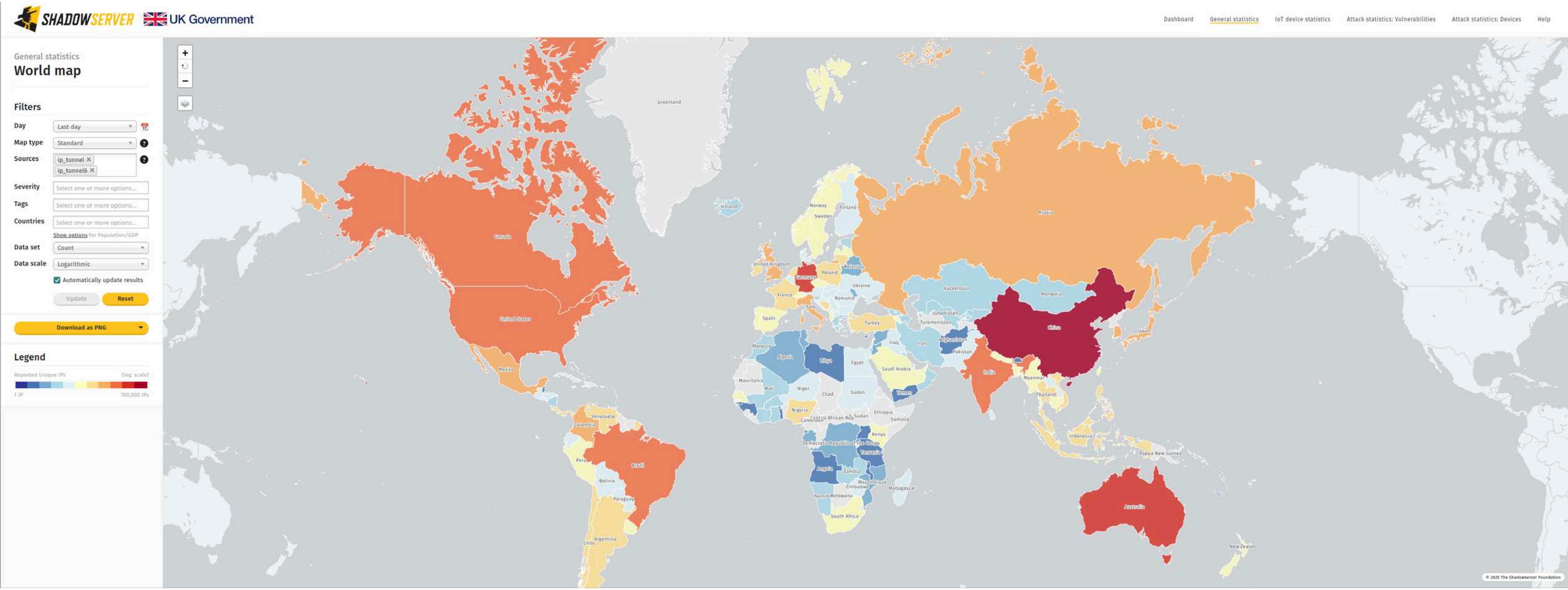
Individually contacted major vulnerable organisations

SHADOWSERVER

Non-profit organization

Scanned with subset of scans

Vulnerable parties registered were notified


SHADOWSERVER
UK Government
Dashboard
General statistics
IoT device statistics
Attack statistics: Vulnerabilities
Attack statistics: Devices
Help
General statistics
World map
Filters
Day
Last day
Map type
Standard
Sources
ip_tunnel
ip_tunnel6
Severity
Select one or more options...
Tags
Select one or more options...
Countries
Select one or more options...
Show options for Population/GDP
Data set
Count
Data scale
Logarithmic
Automatically update results
Update
Reset
Download as PNG
Legend
Reported Unique IPs
(log. scale)
1 IP
700,000 IPs
Greenland
Canada
United States
Mexico
Venezuela
Colombia
Brazil
Peru
Bolivia
Paraguay
Chile
Argentina
Iceland
Norway
Sweden
Finland
United Kingdom
Germany
Poland
Belarus
Ukraine
France
Italy
Romania
Spain
Turkey
Russia
Kazakhstan
Mongolia
Uzbekistan
Turkmenistan
Afghanistan
Pakistan
Iran
Iraq
China
Japan
India
Myanmar
Thailand
Morocco
Algeria
Libya
Egypt
Saudi Arabia
Yemen
Mauritania
Mali
Niger
Chad
Sudan
Nigeria
Ethiopia
Somalia
Kenya
Tanzania
Angola
Zambia
Mozambique
Zimbabwe
Botswana
South Africa
Madagascar
Indonesia
Papua New Guinea
Australia
New Zealand
© 2025 The Shadowserver Foundation

The Good News

Major ISP fixed the issue

Over 700k hosts were patched

Many organisations got in contact

The Bad News

Some didn’t reply ☹

Emails you want to see

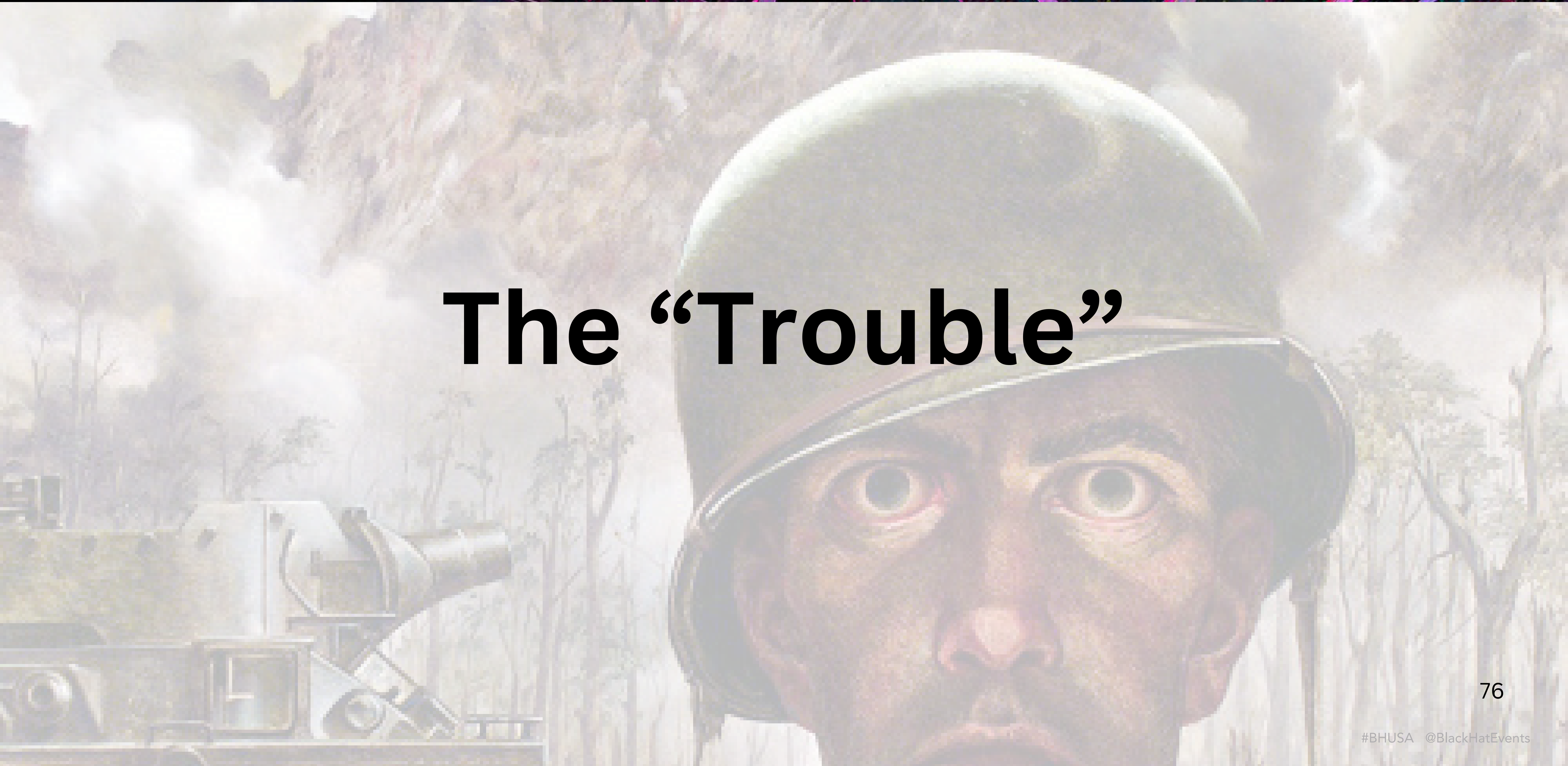
"We can confirm that there is a potential flaw on 6in4 setup…We have forwarded your study to our network and router development teams, and they agree a fix needs to be implemented…"

"…one of our hosts is affected … We would very much appreciate a current copy of your tunneltester scripts…"

Emails you DON'T want to see

" … investigating an issue involving thousands of customer…devices rebooting at 1 AM yesterday."

"I ask you to avoid scanning the following ip address ranges : …
since this is causing some trouble for some devices in our network"



The “Trouble”

O-421E-B ONT Devices

- Optical Network Terminal
- Placed at each residence
- EOL 2018
- Still in use by ISPs...

O-421E-B ONT Devices

1: Receive encapsulated packet

O-421E-B ONT Devices

1: Receive encapsulated packet

2:

O-421E-B ONT Devices

1: Receive encapsulated packet

2:

O-421E-B Replacement

Nov. 2024: 1010/3980

May 2025: 1047/3980

June 2025: 1190/3980

July 2025: 1328/3980

Amplification Attacks

Ping-Pong

Loop traffic between two hosts

Ping-Pong

- Contruct packet with many headers
- Header *X* : *A* → *B* then *X+1:* *B* → *A*

Ping-Pong Attack

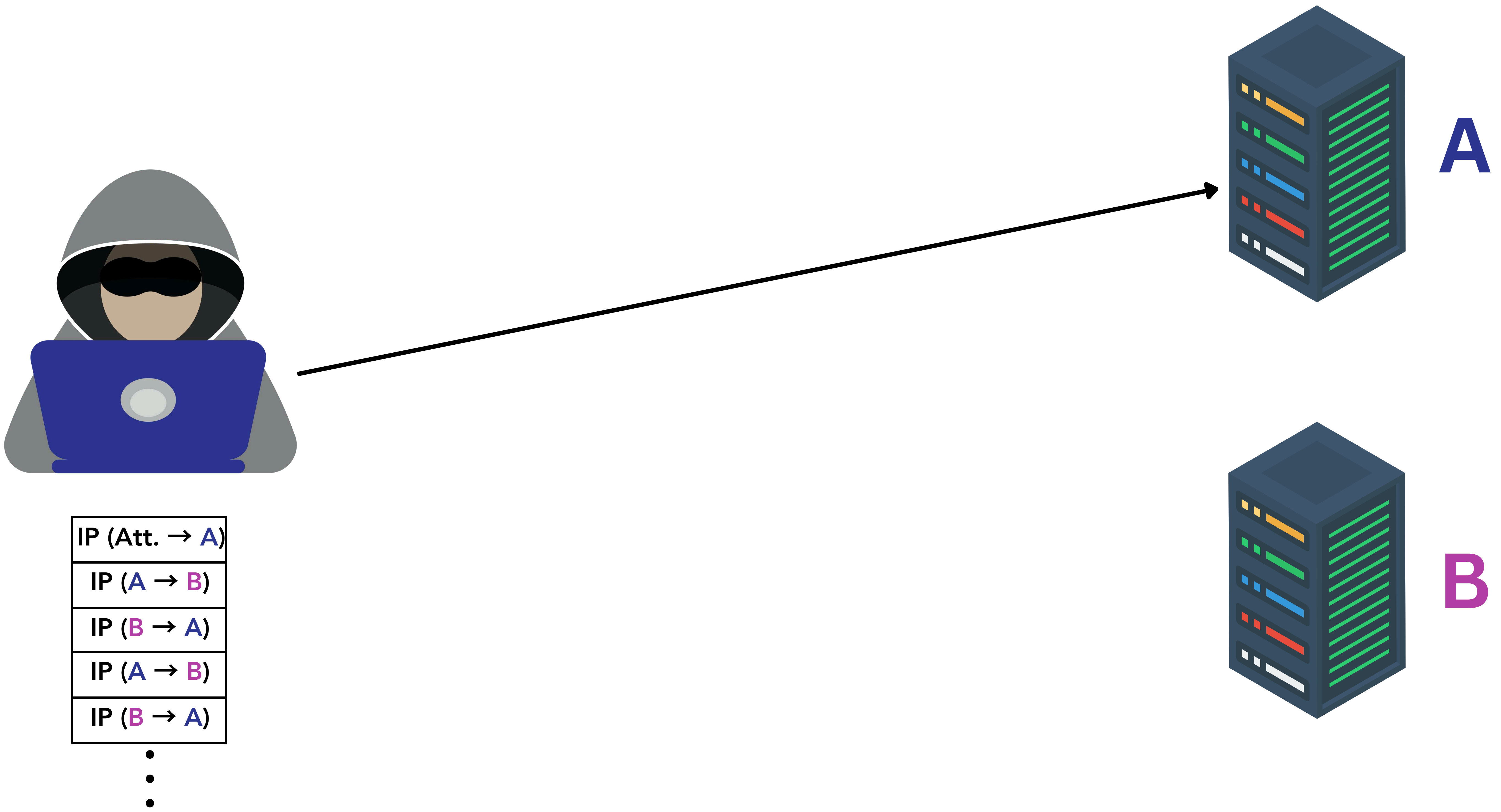
IP (Att. → A)
IP (A → B)
IP (B → A)
IP (A → B)
IP (B → A)

⋮

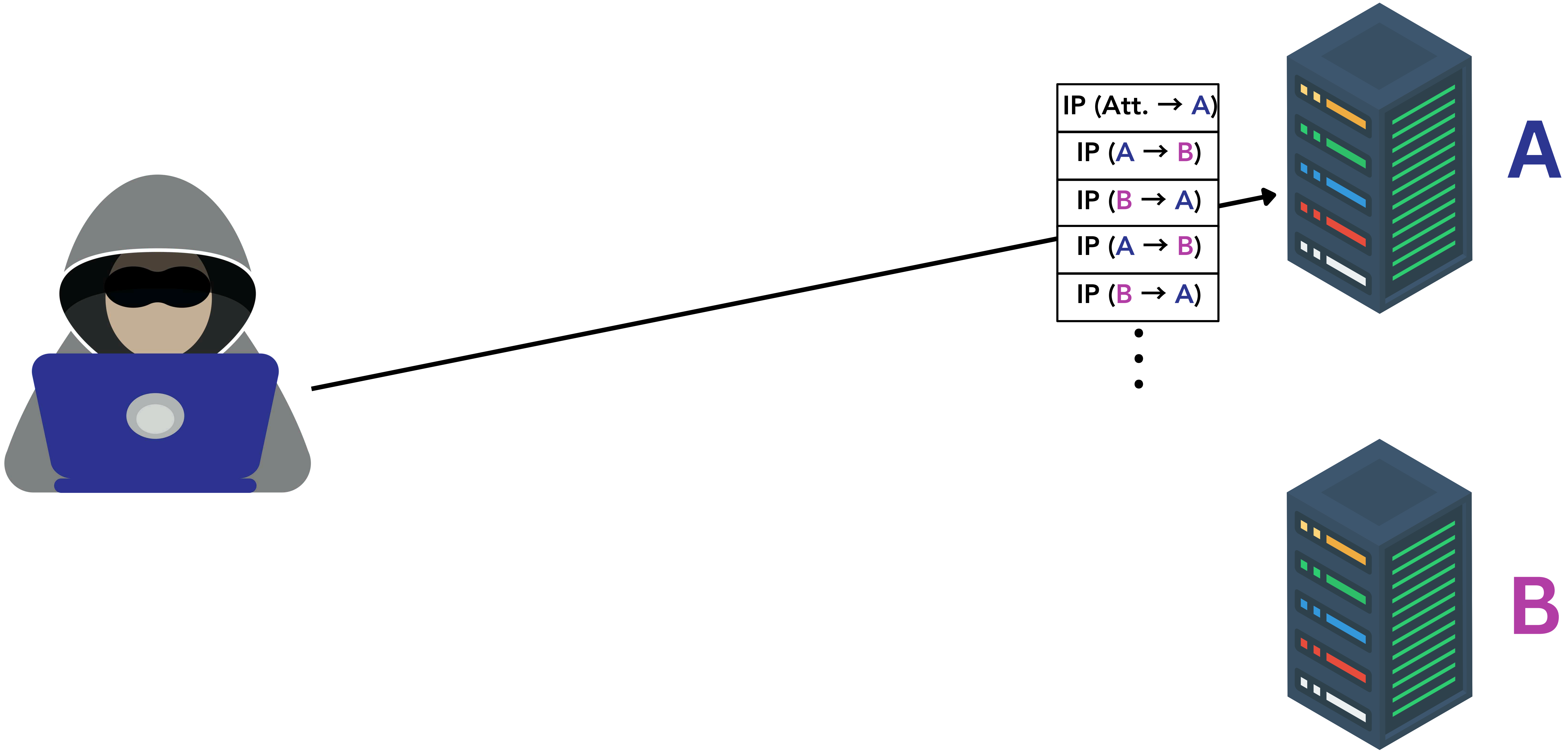
A

B

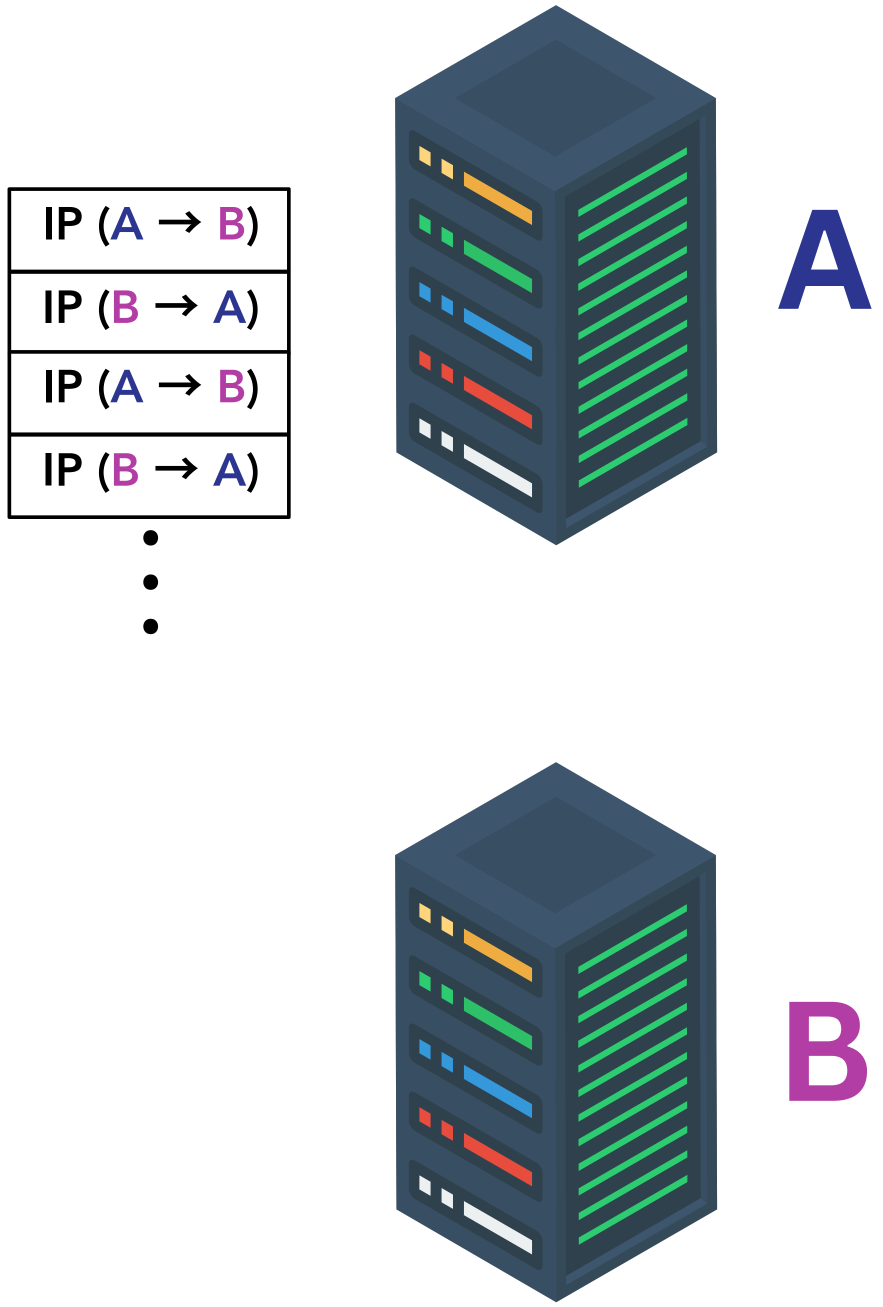
Ping-Pong Attack



Ping-Pong Attack

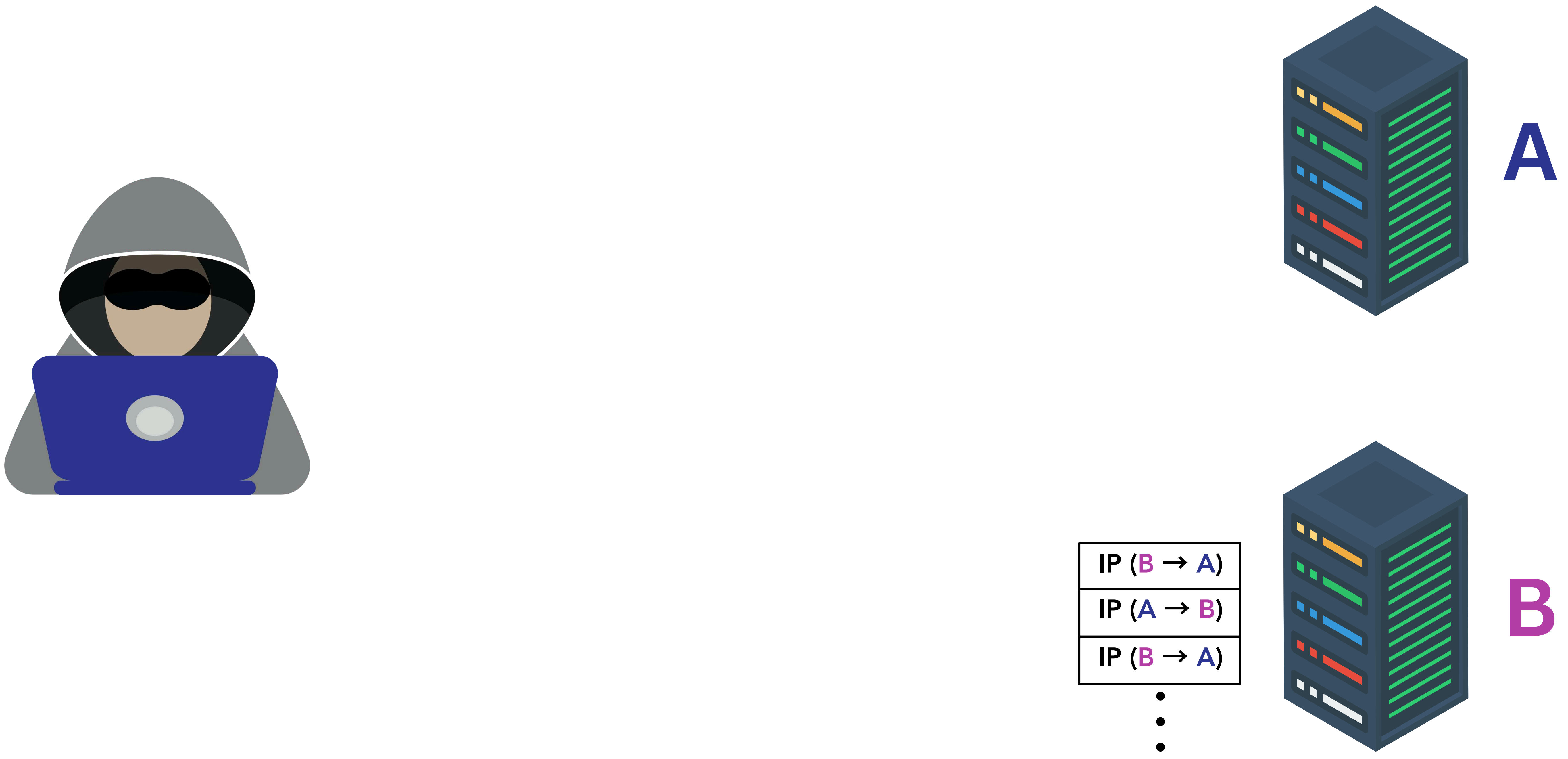


Ping-Pong Attack



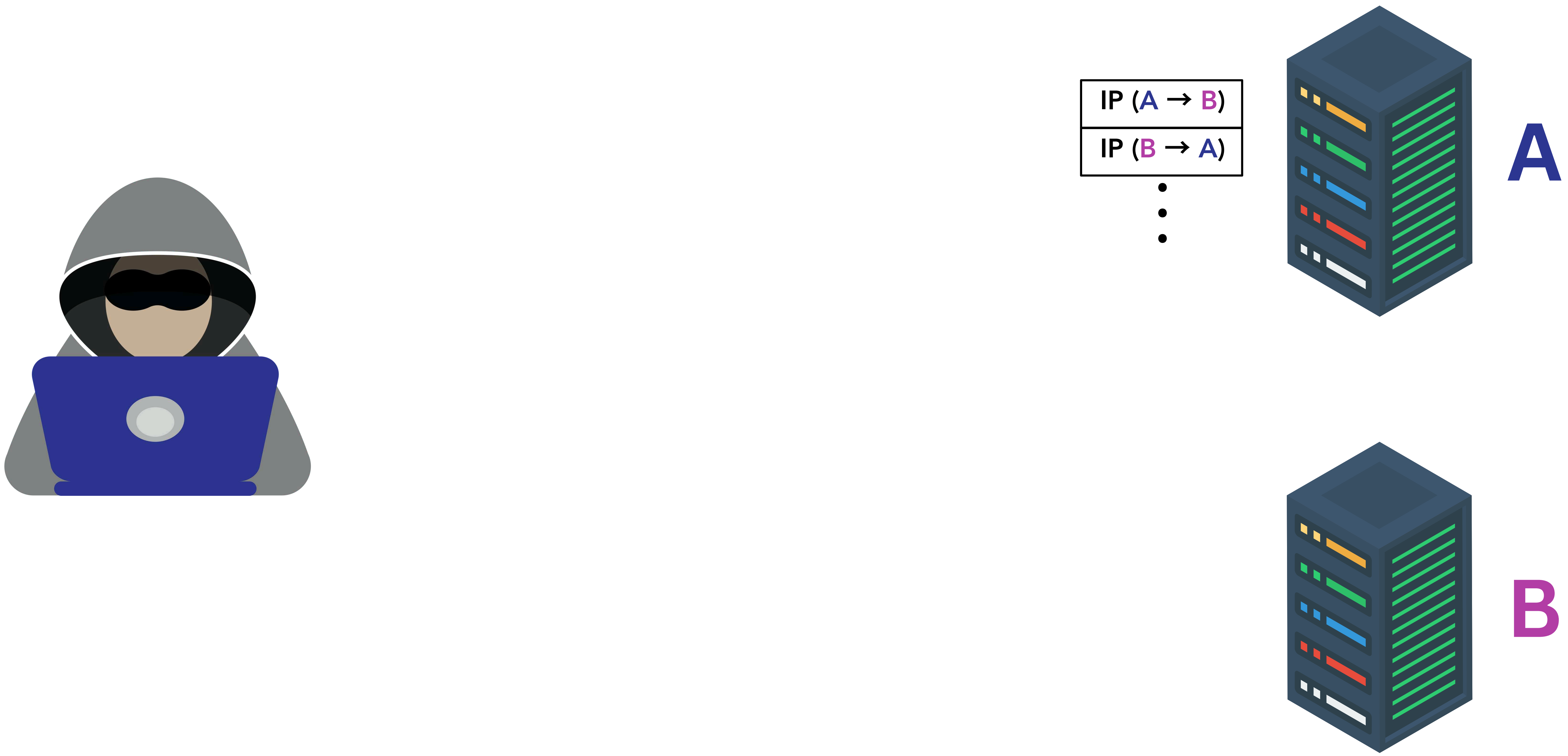
Ping-Pong Attack

Ping-Pong Attack



Ping-Pong Attack

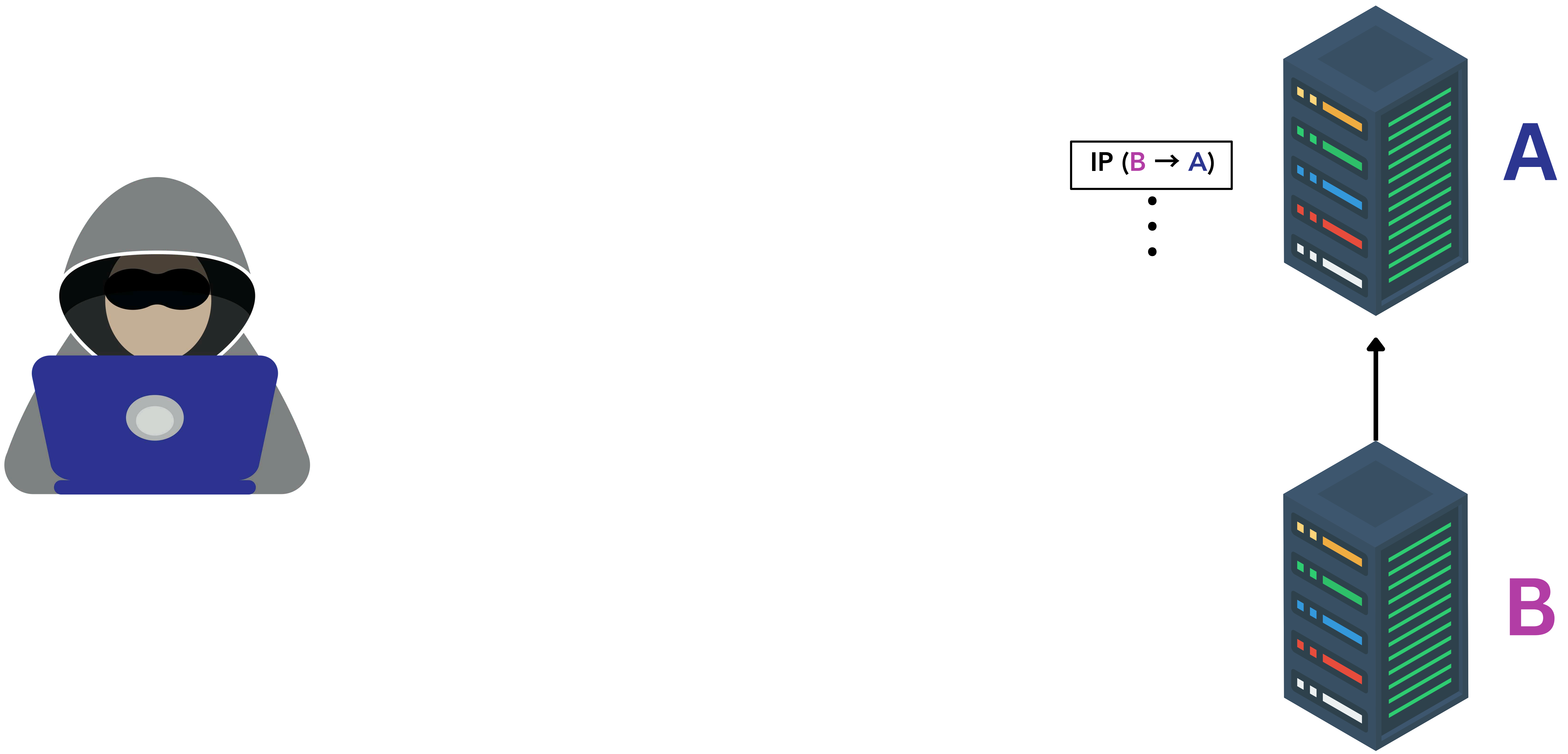
Ping-Pong Attack



Ping-Pong Attack

Ping-Pong Attack

Ping-Pong Attack



Ping-Pong Attack

Ping-Pong Attack
Amplification

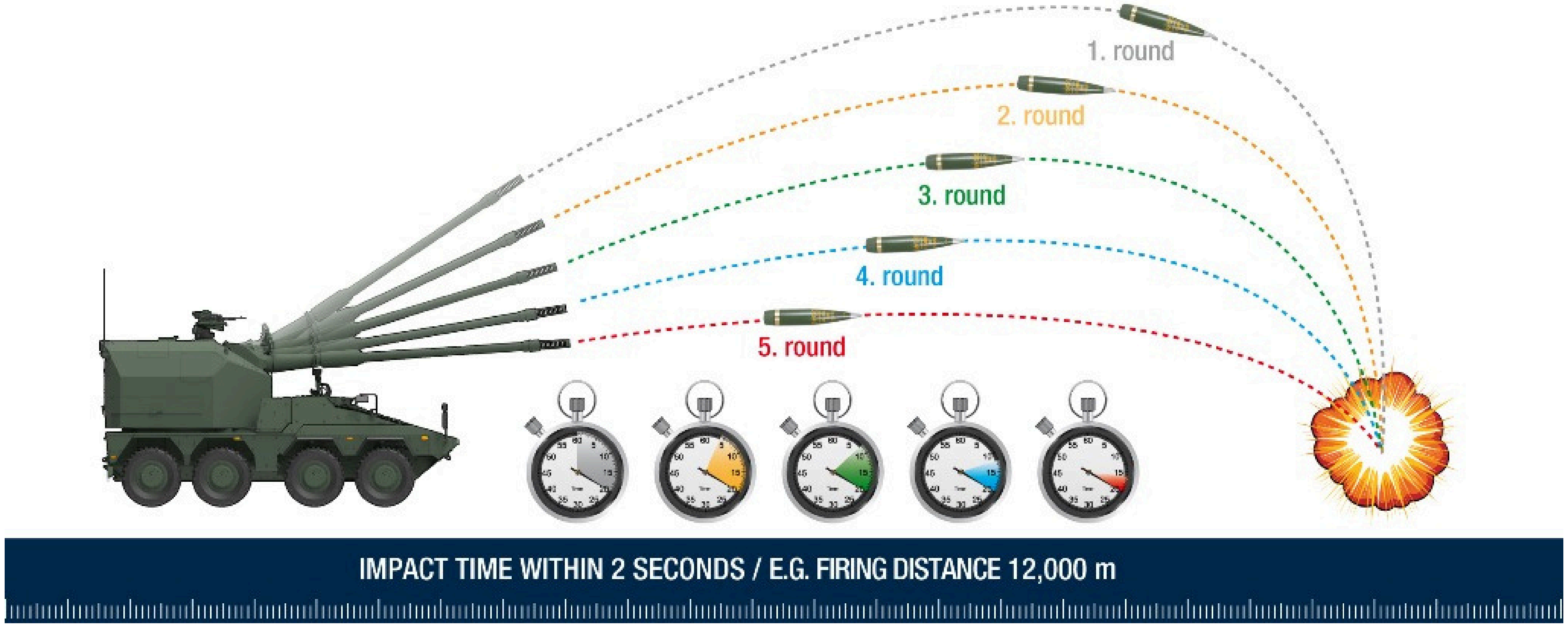
- $a \approx \frac{headers}{2}$
- **MTU = 1500 → x37.5**
- **Can be combined with other amplification attacks**

Tunneled-Temporal Lensing (TuTL)

Concentrate packets in time

Create pulsing effect at victim

Tunneled-Temporal Lensing (TuTL)



1) Collect latencies

2) Construct paths

3) Schedule & Send traffic

1) Collect latencies

1) Collect latencies

1) Collect latencies: Attacker to Host

1) Collect latencies: Attacker to Host

1) Collect latencies: Attacker to Host

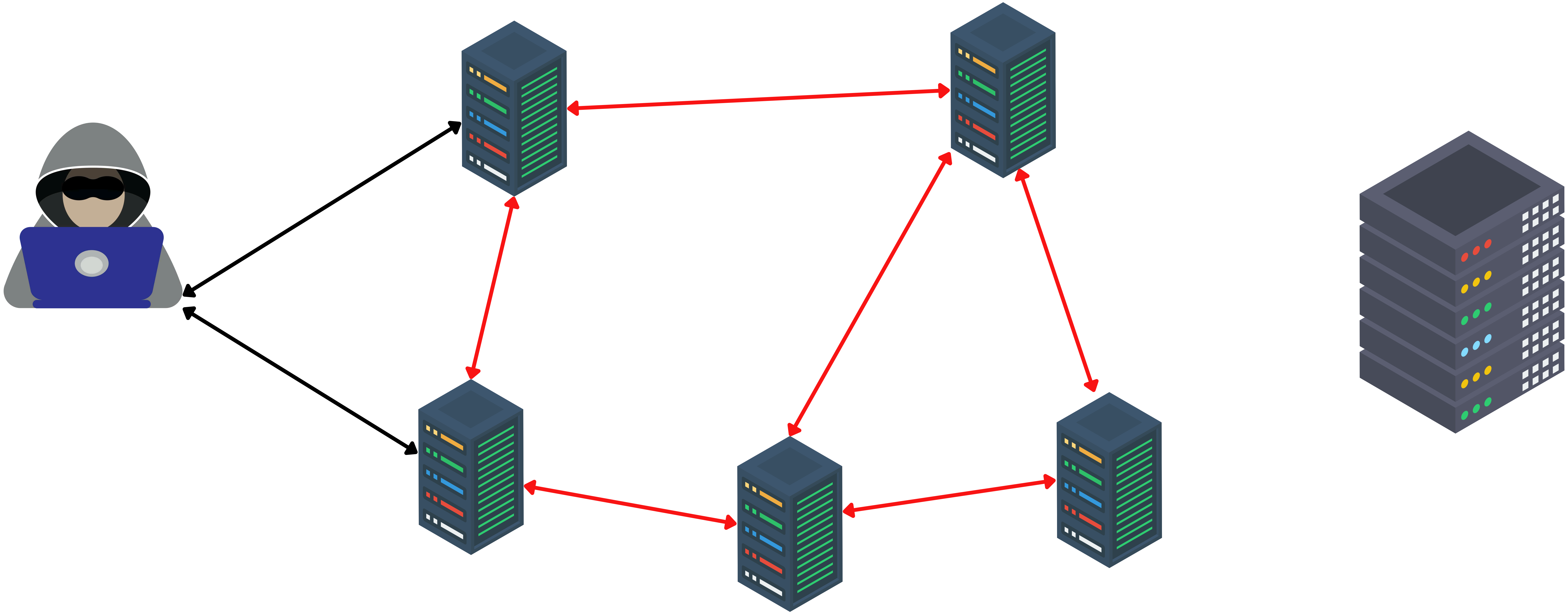
1) Collect latencies: Attacker to Host

1) Collect latencies: Attacker to Host

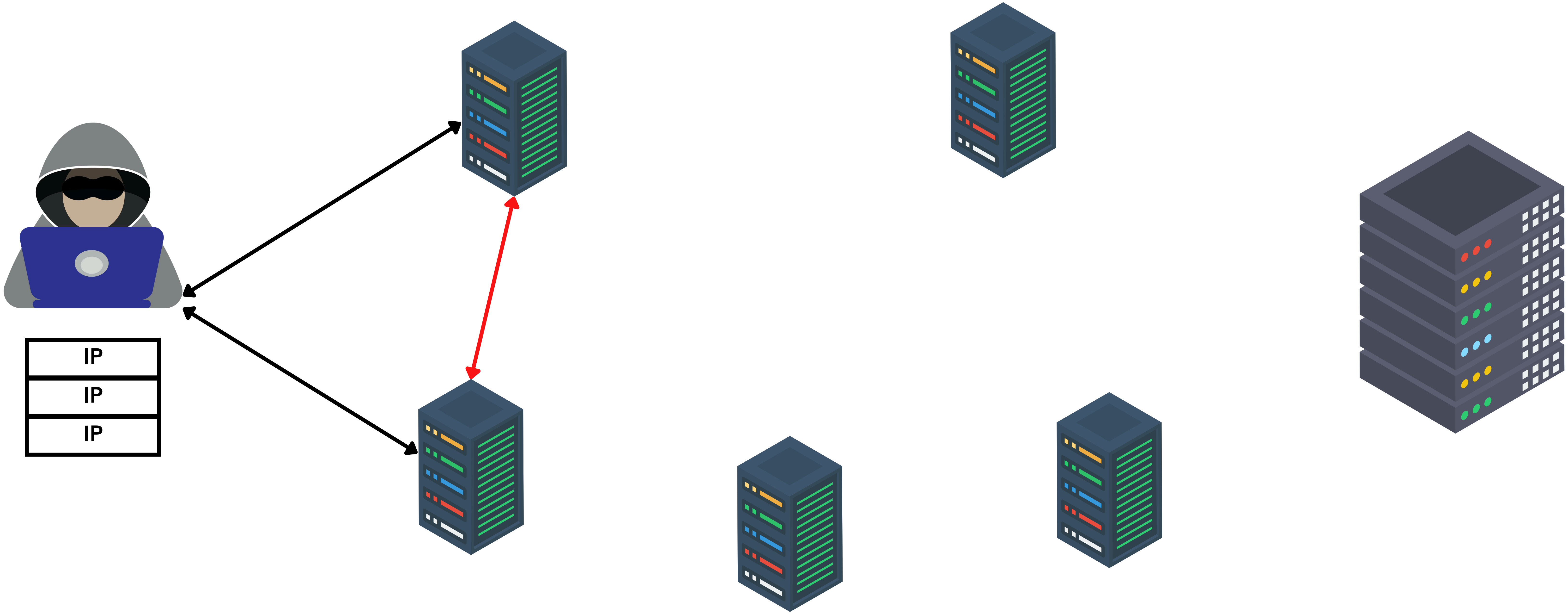
1) Collect latencies: Attacker to Host

$$l1 = \frac{RTT\,(attacker \rightarrow h1 \rightarrow attacker)}{2}$$

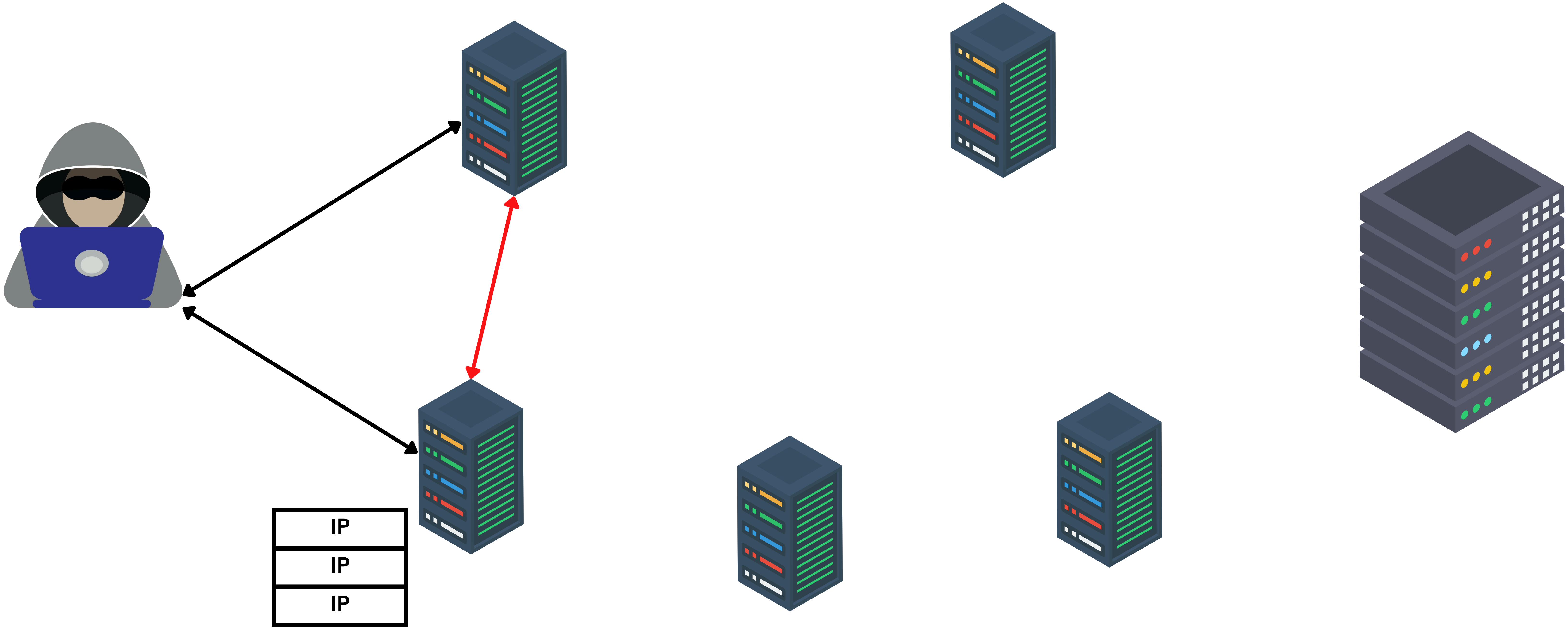
1) Collect latencies: Host to Host



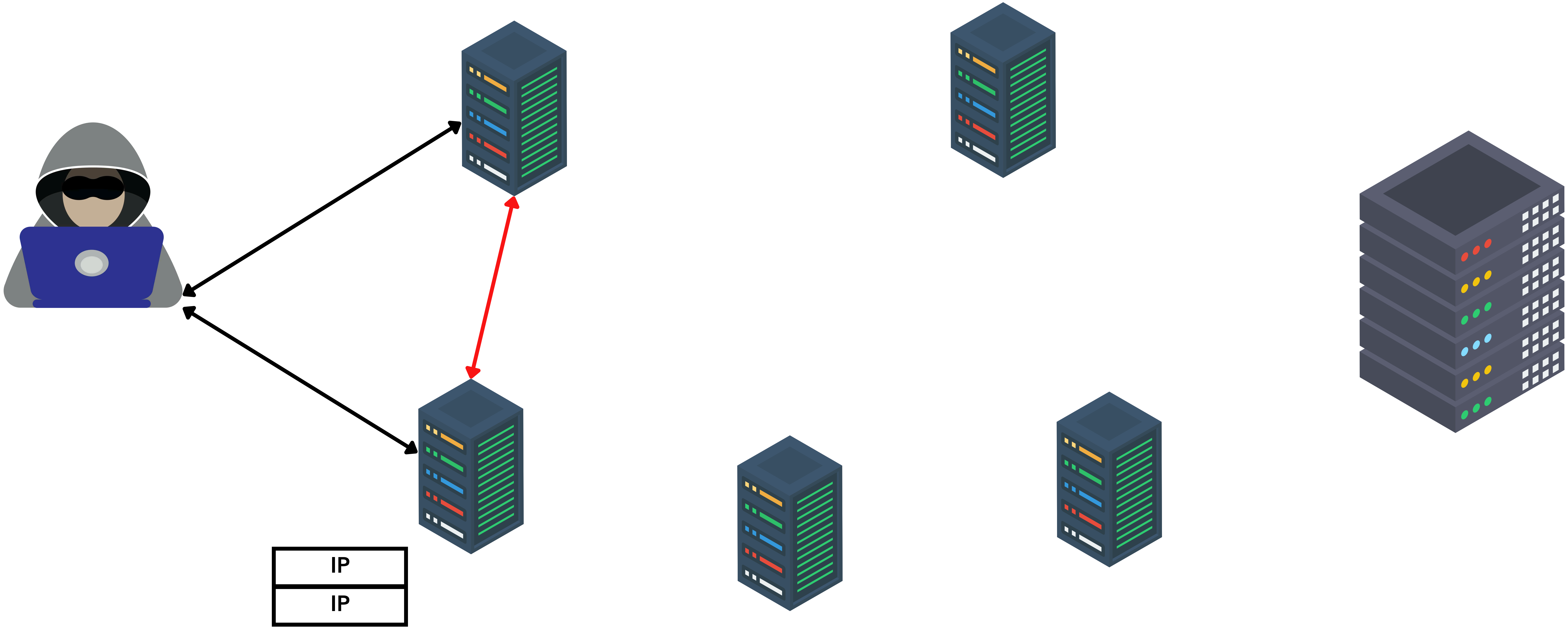
1) Collect latencies: Host to Host



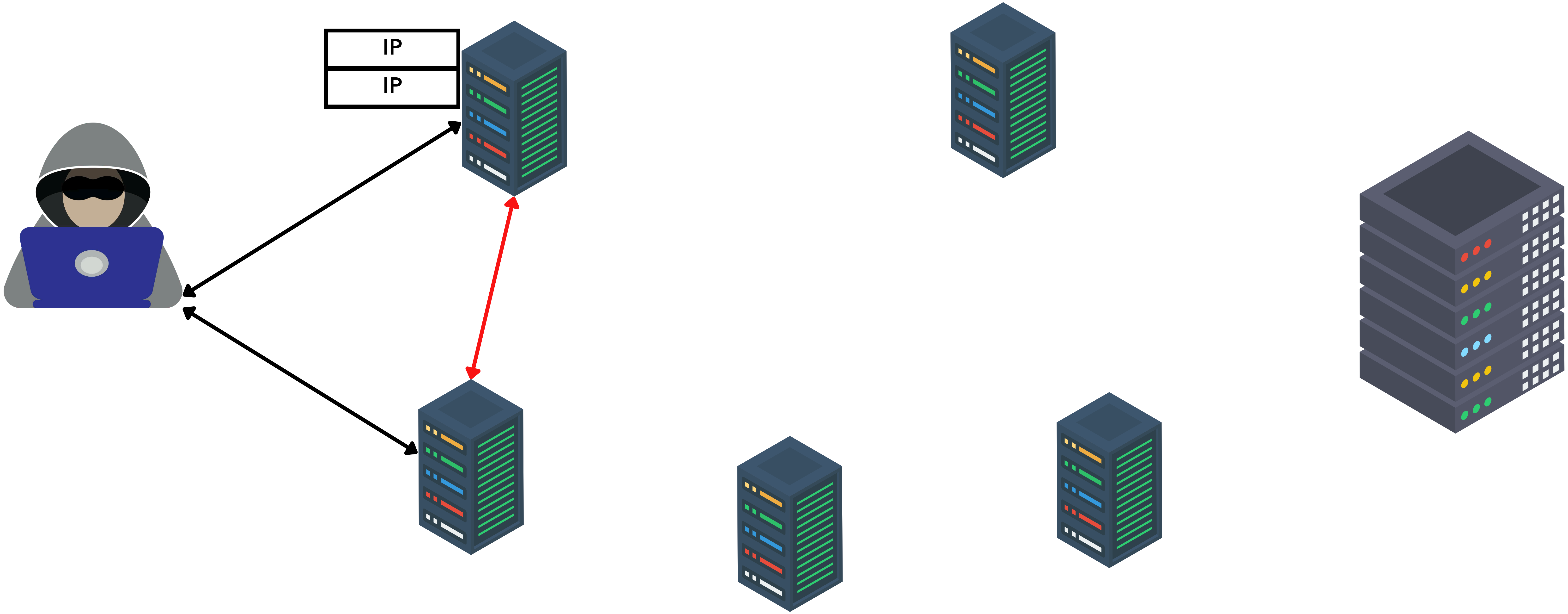
1) Collect latencies: Host to Host



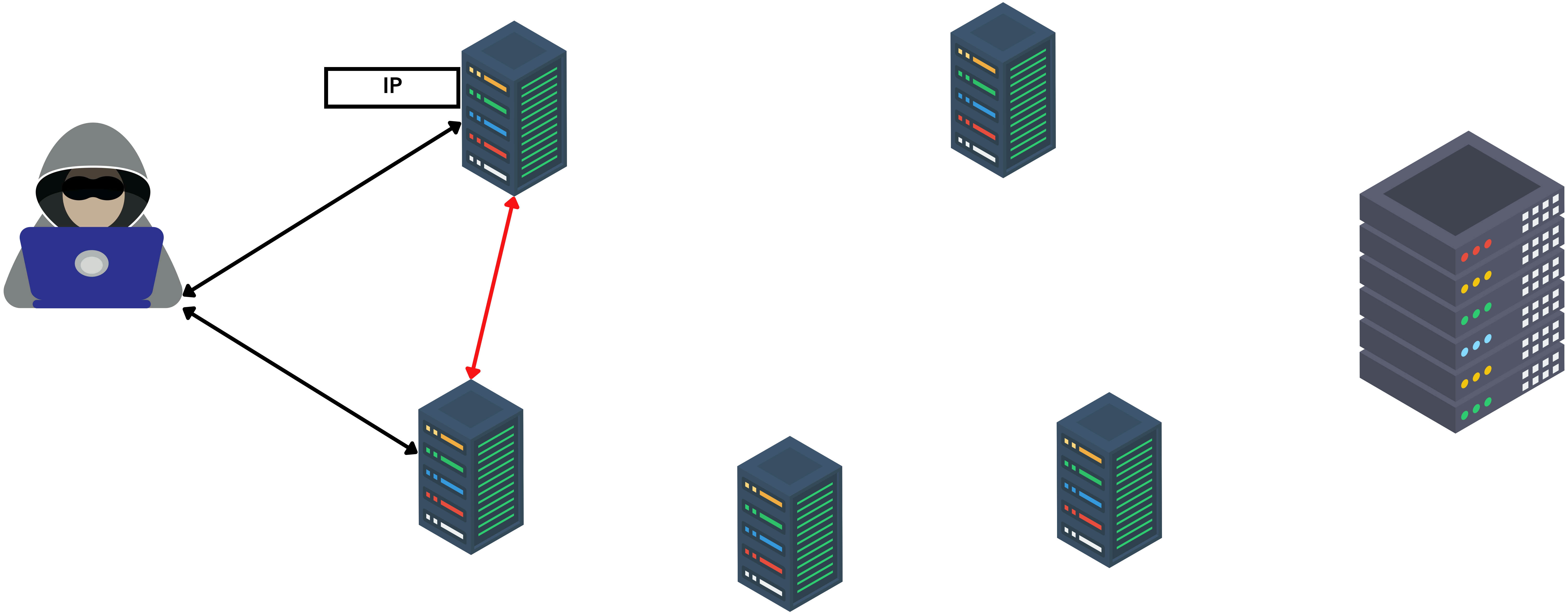
1) Collect latencies: Host to Host



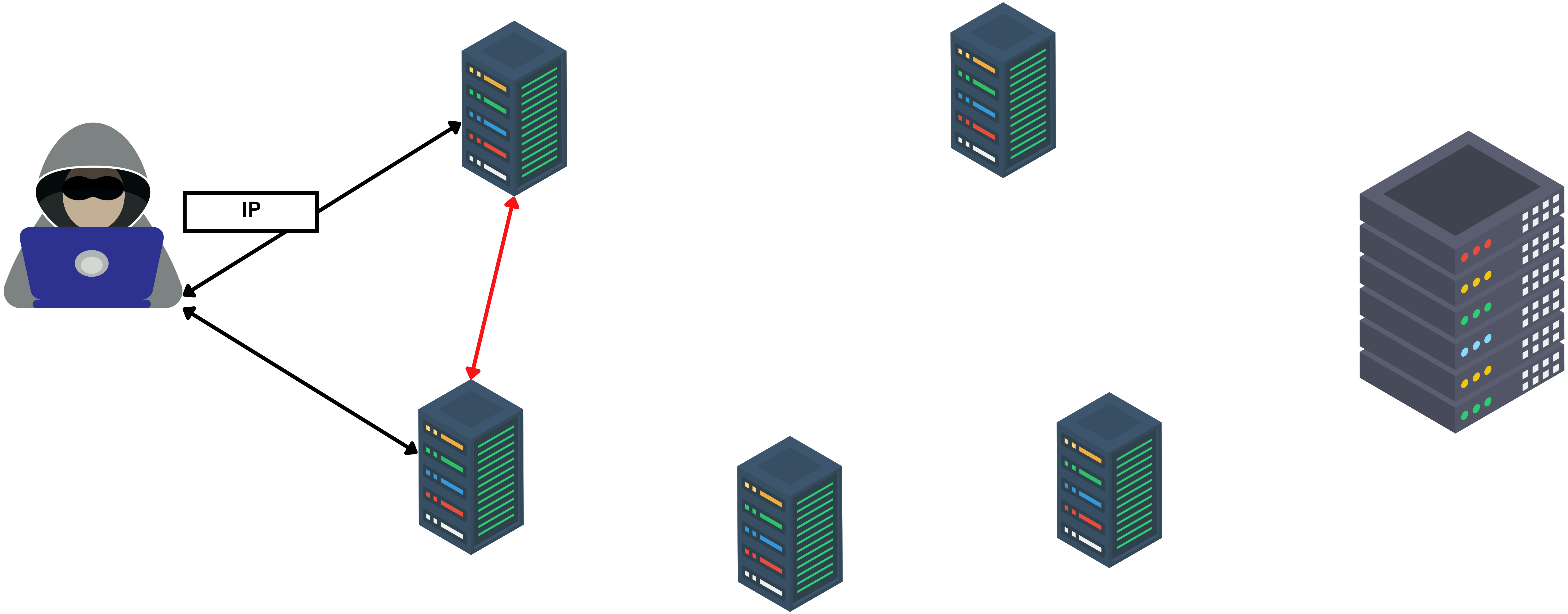
1) Collect latencies: Host to Host



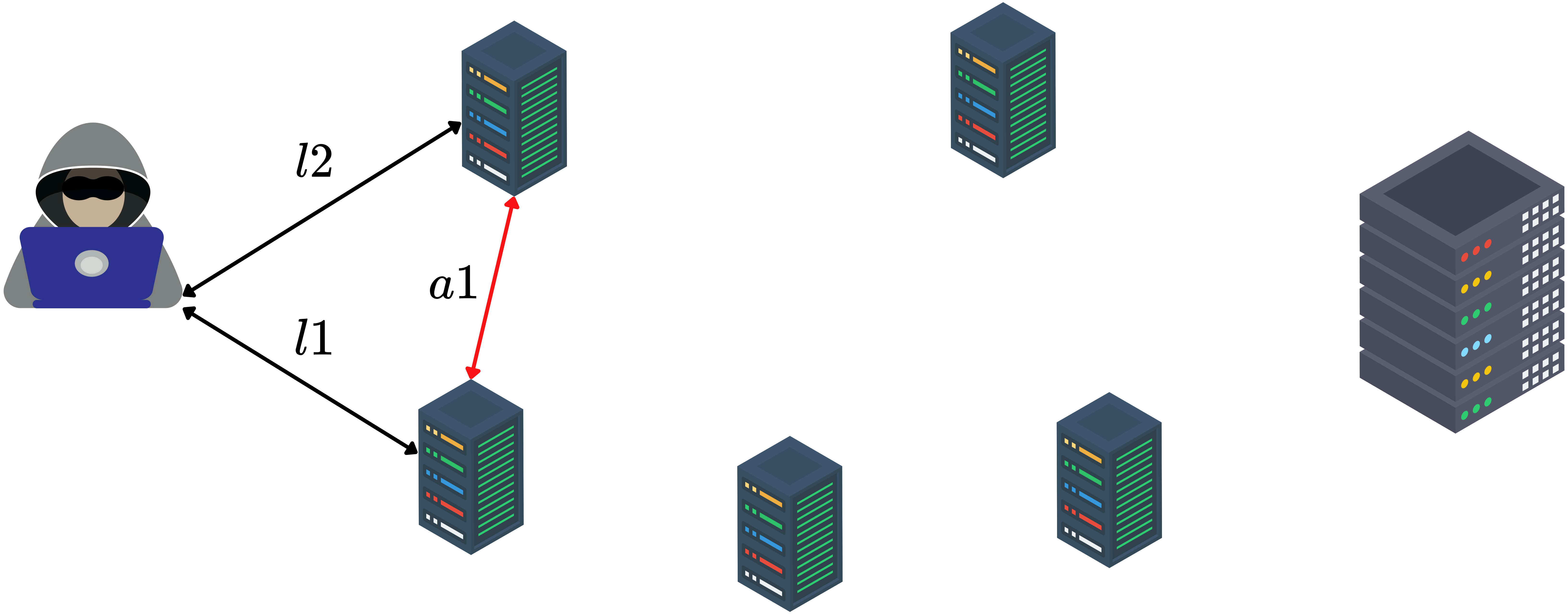
1) Collect latencies: Host to Host



1) Collect latencies: Host to Host

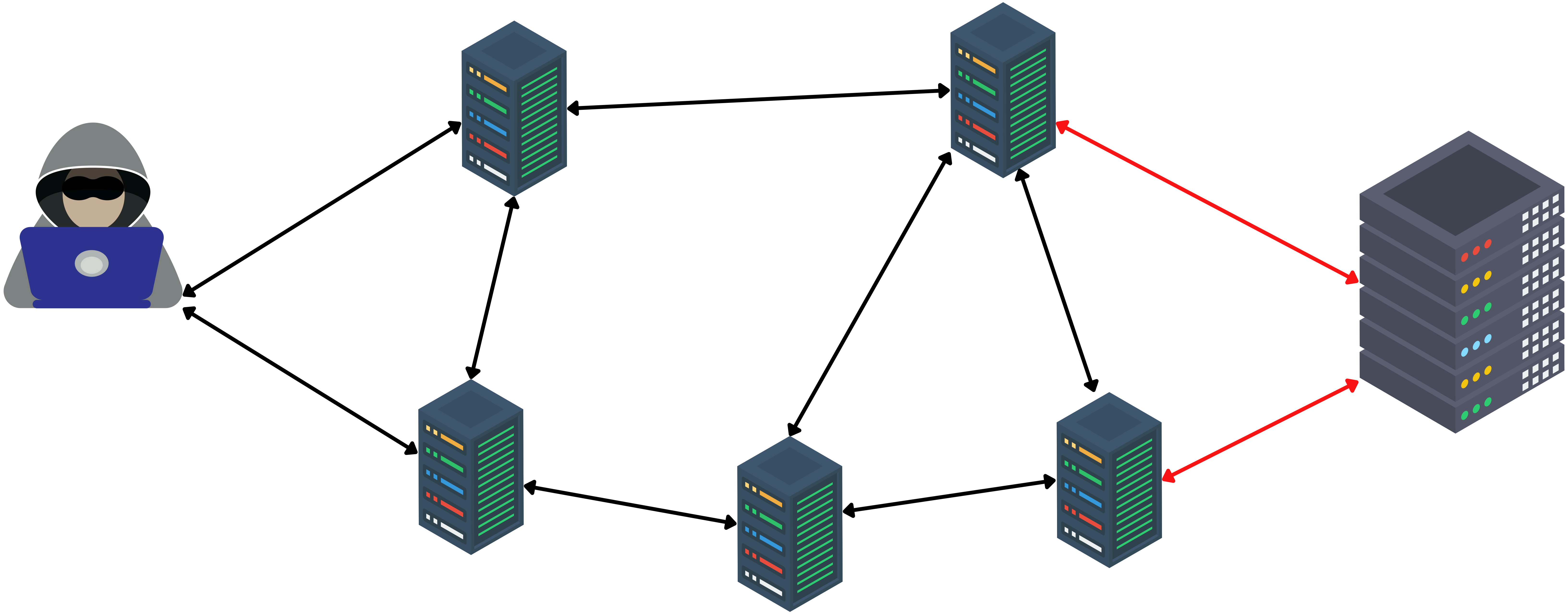


1) Collect latencies: Host to Host

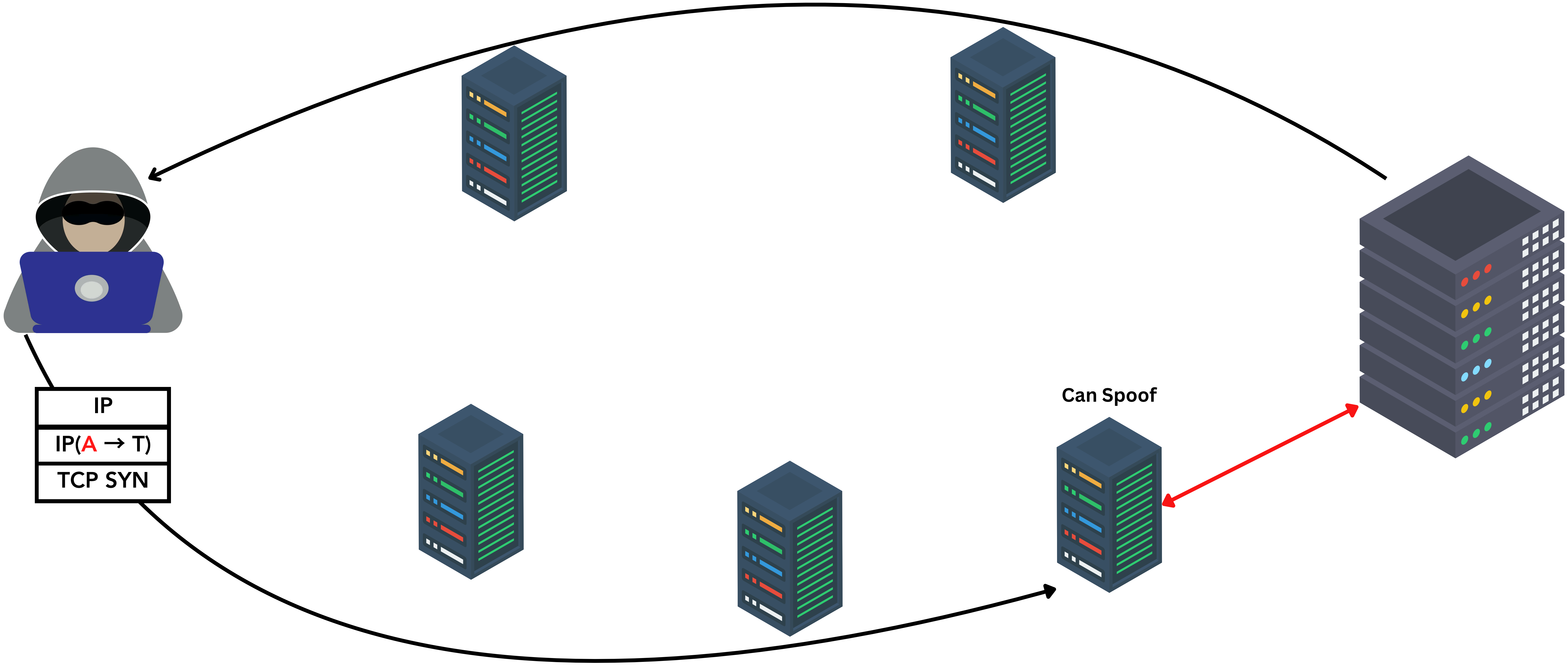


$$a1 = RTT\,(attacker \rightarrow h1 \rightarrow h2 \rightarrow attacker) - l1 - l2$$

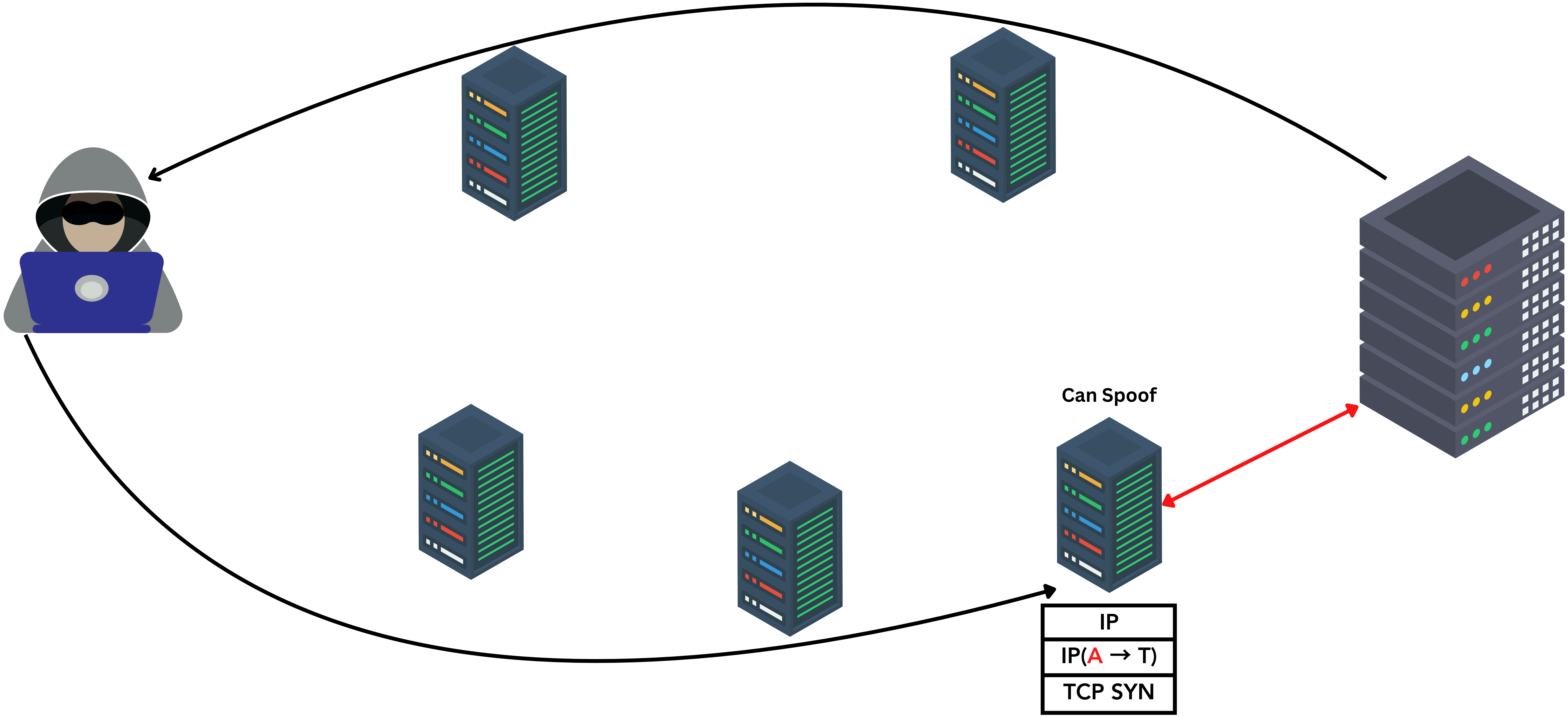
1) Collect latencies: Host to Victim



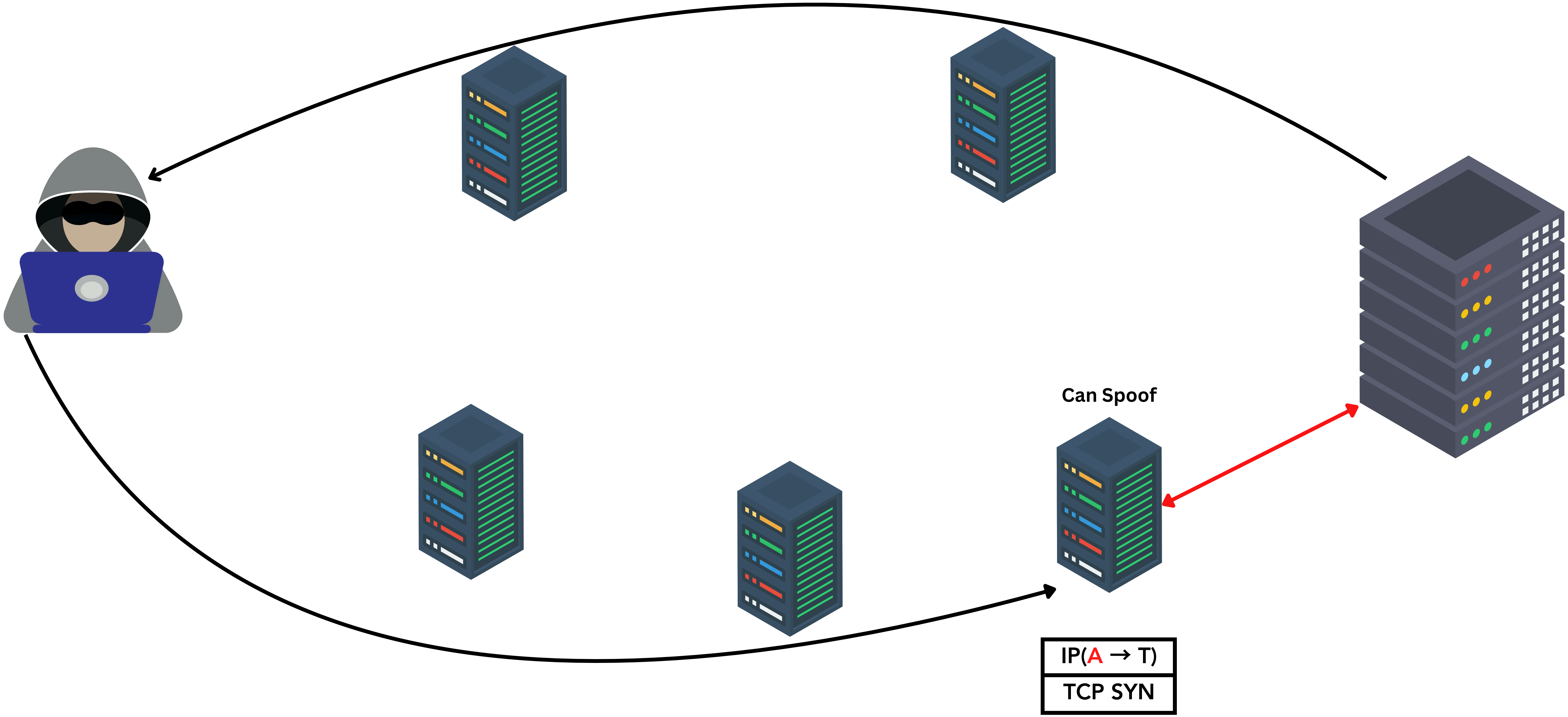
1) Collect latencies: Host to Victim



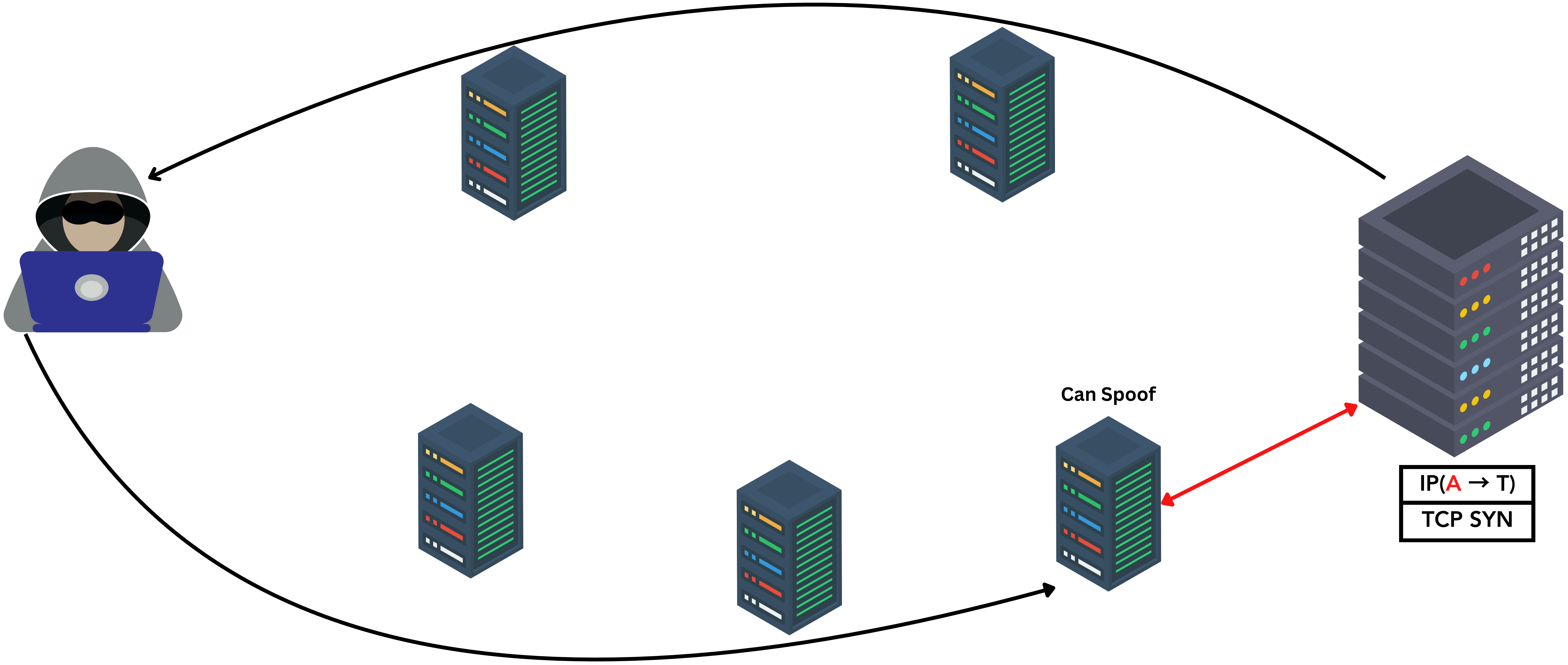
1) Collect latencies: Host to Victim



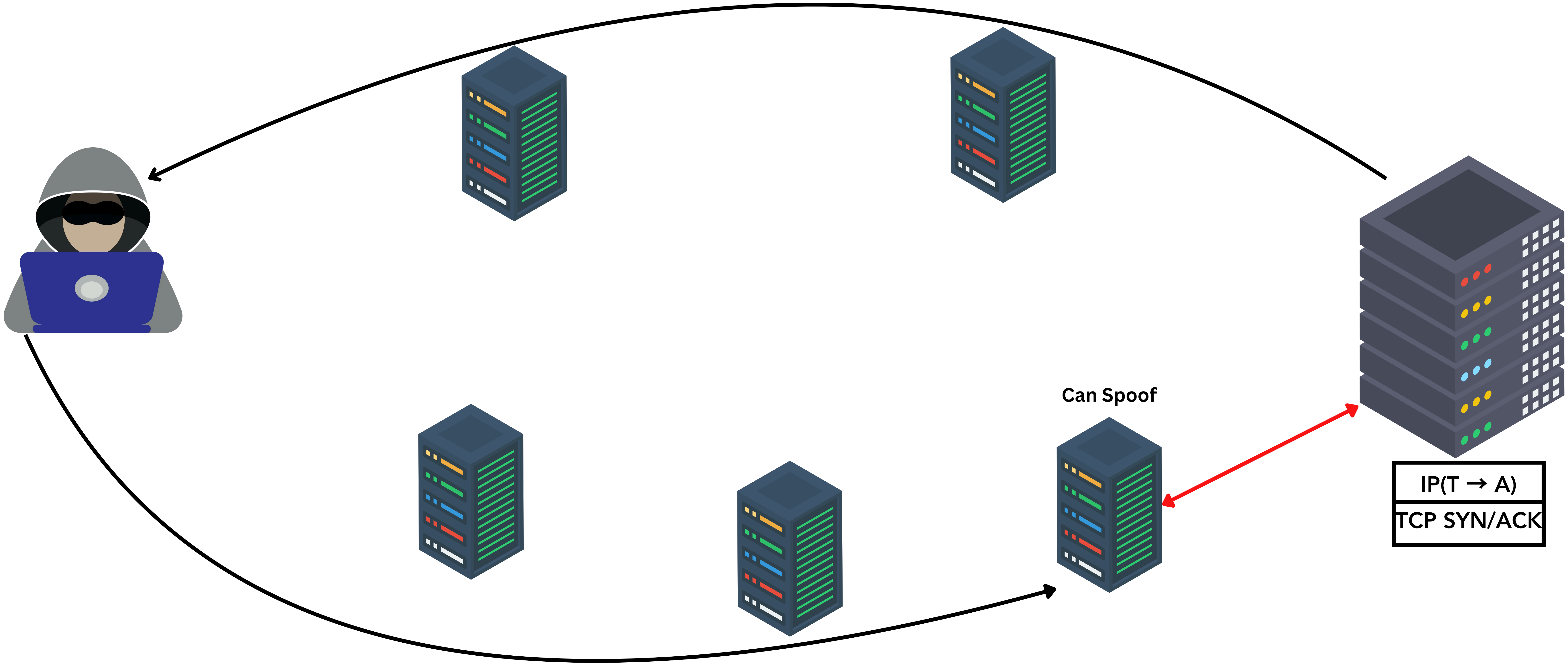
1) Collect latencies: Host to Victim



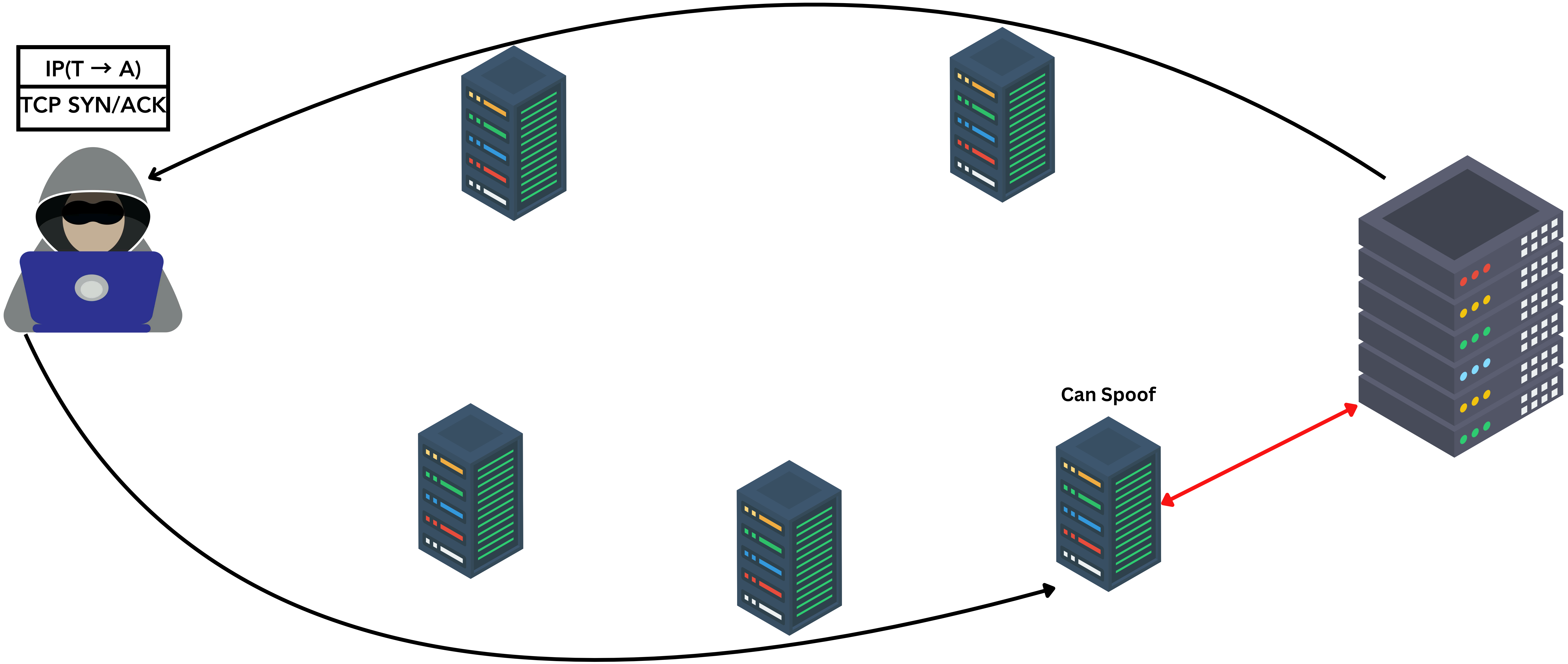
1) Collect latencies: Host to Victim



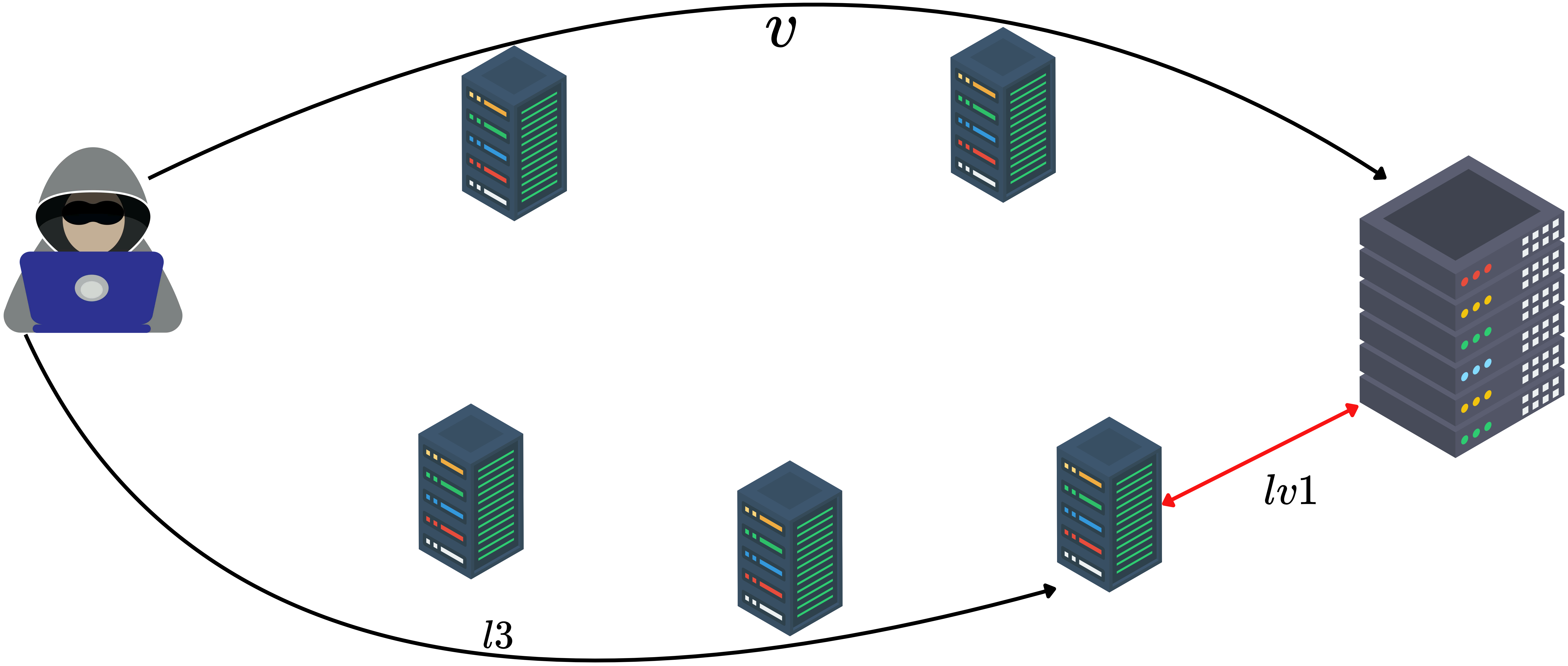
1) Collect latencies: Host to Victim



1) Collect latencies: Host to Victim



1) Collect latencies: Host to Victim



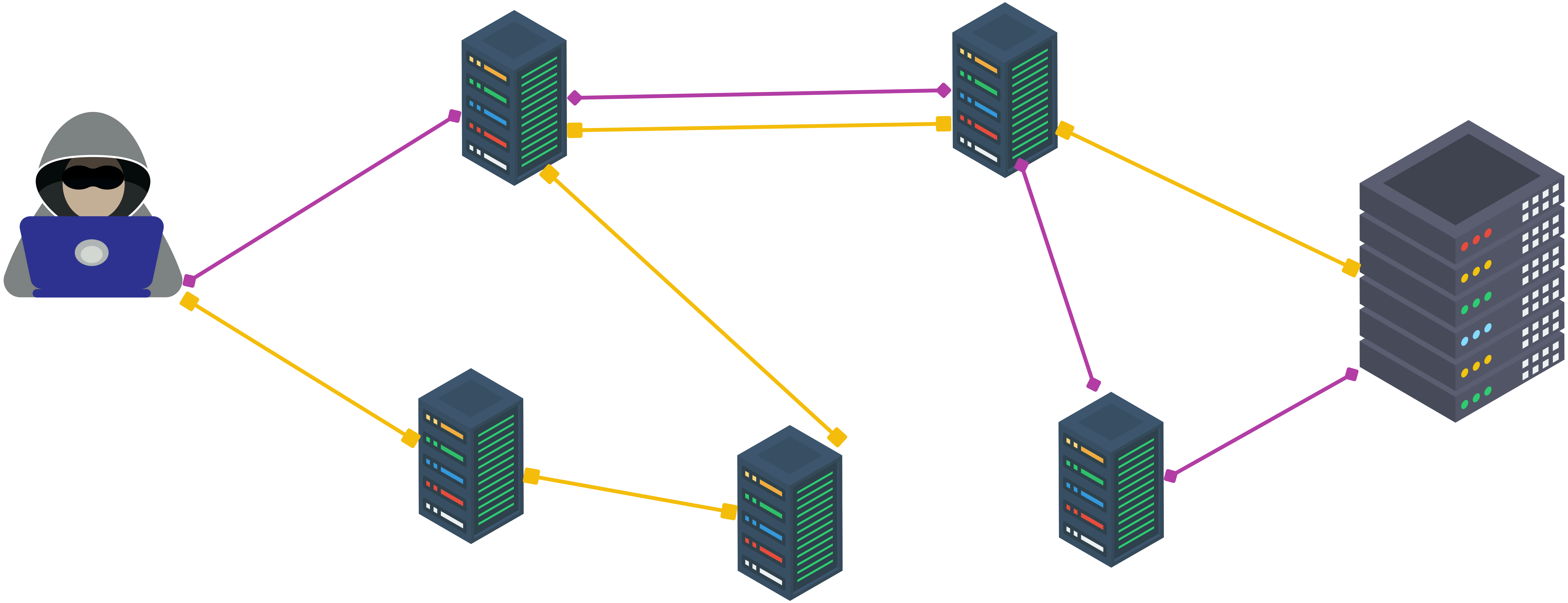
$$lv1 = RTT\,(attacker \rightarrow h3 \rightarrow victim) - l3 - v$$

2) Construct paths

2) Construct Paths

- Attacker → a1,a2.... → Victim
- Estimate total path latencies

2) Identify Paths

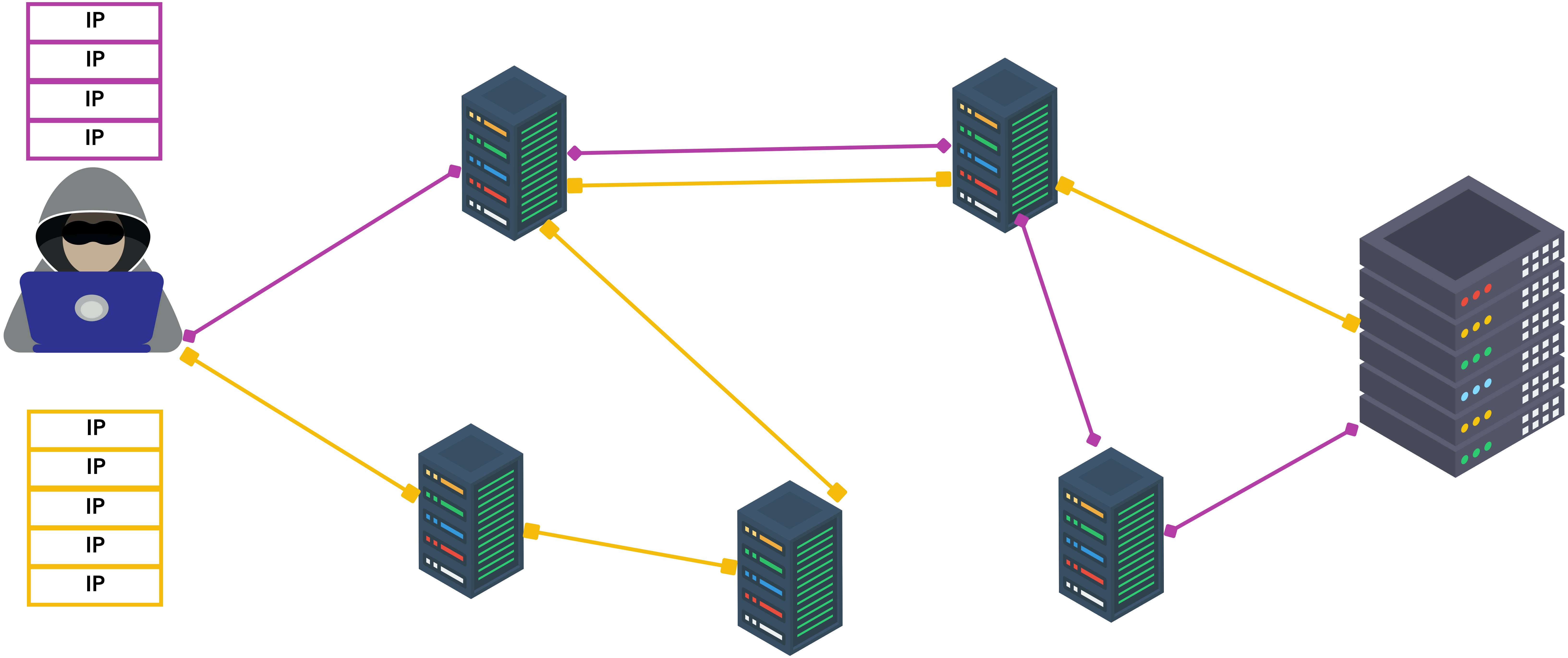


3) Schedule & Send

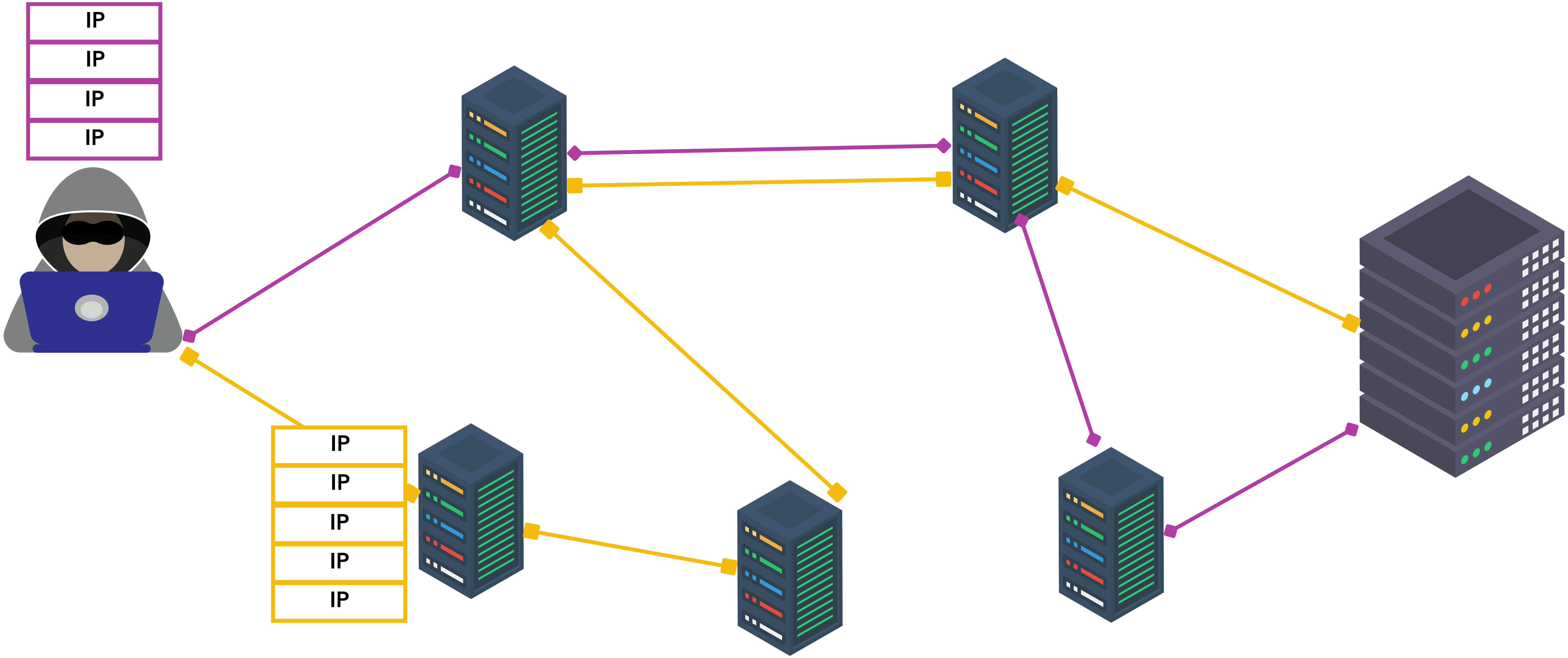
3) Schedule & Send

- Sort paths
- Estimate wait time between paths
- Send-Switch-Send

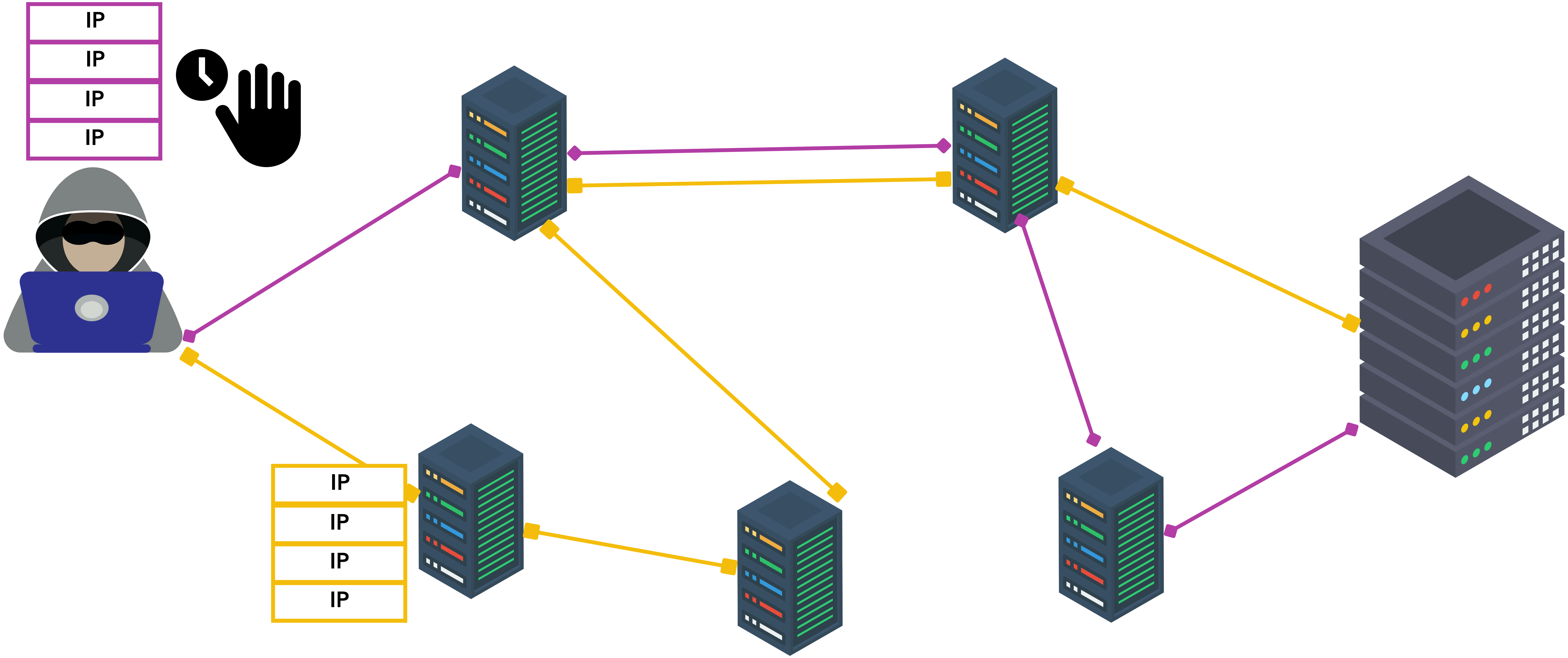
3) Schedule & Send



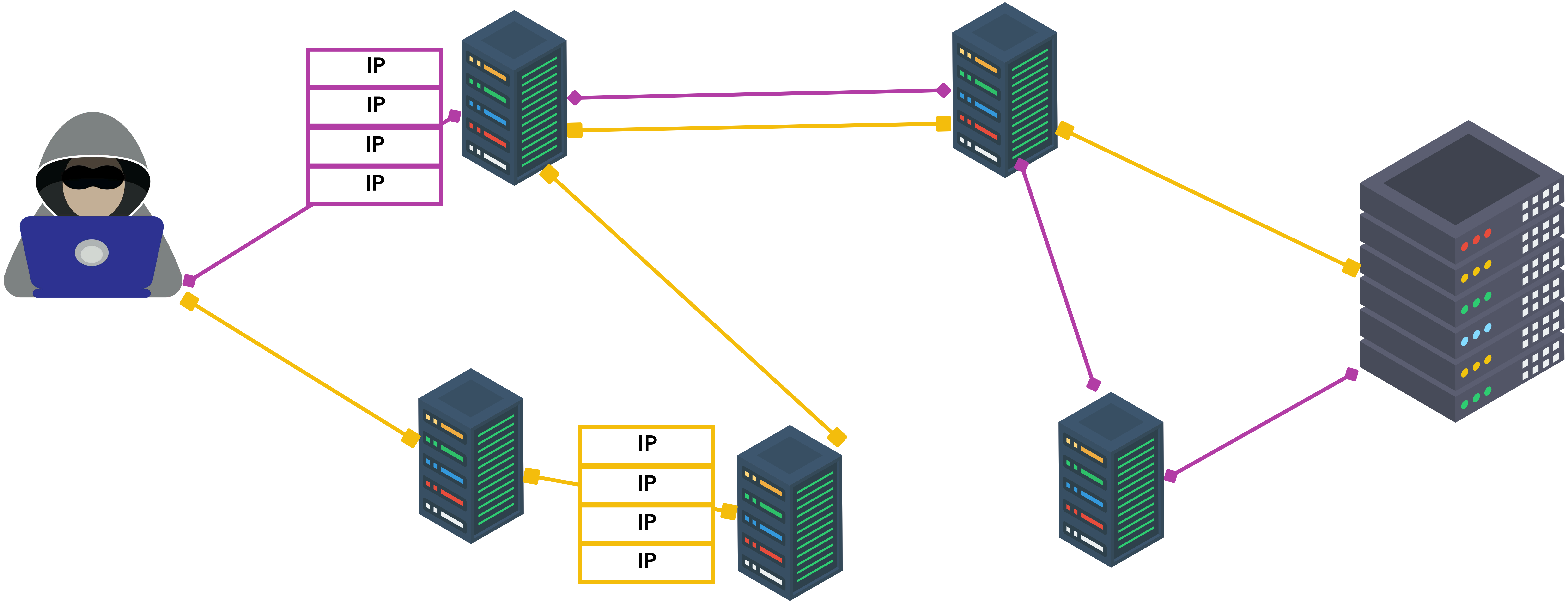
3) Schedule & Send



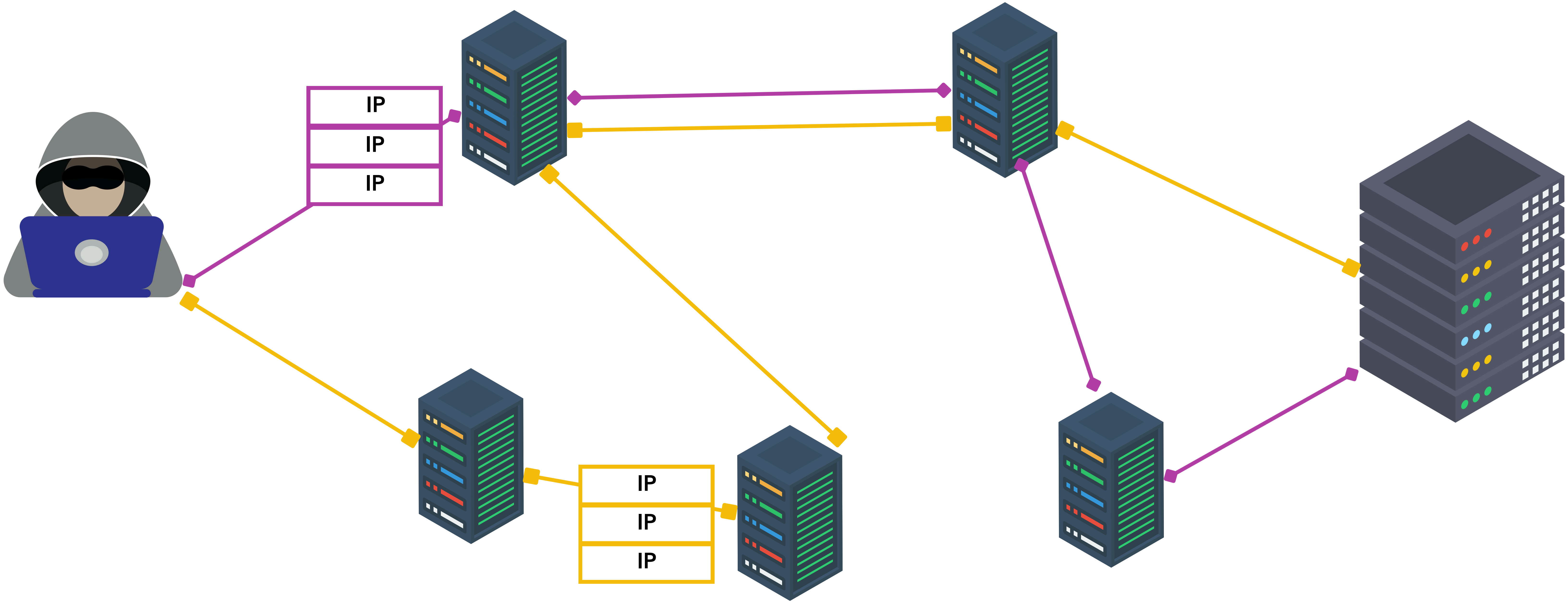
3) Schedule & Send



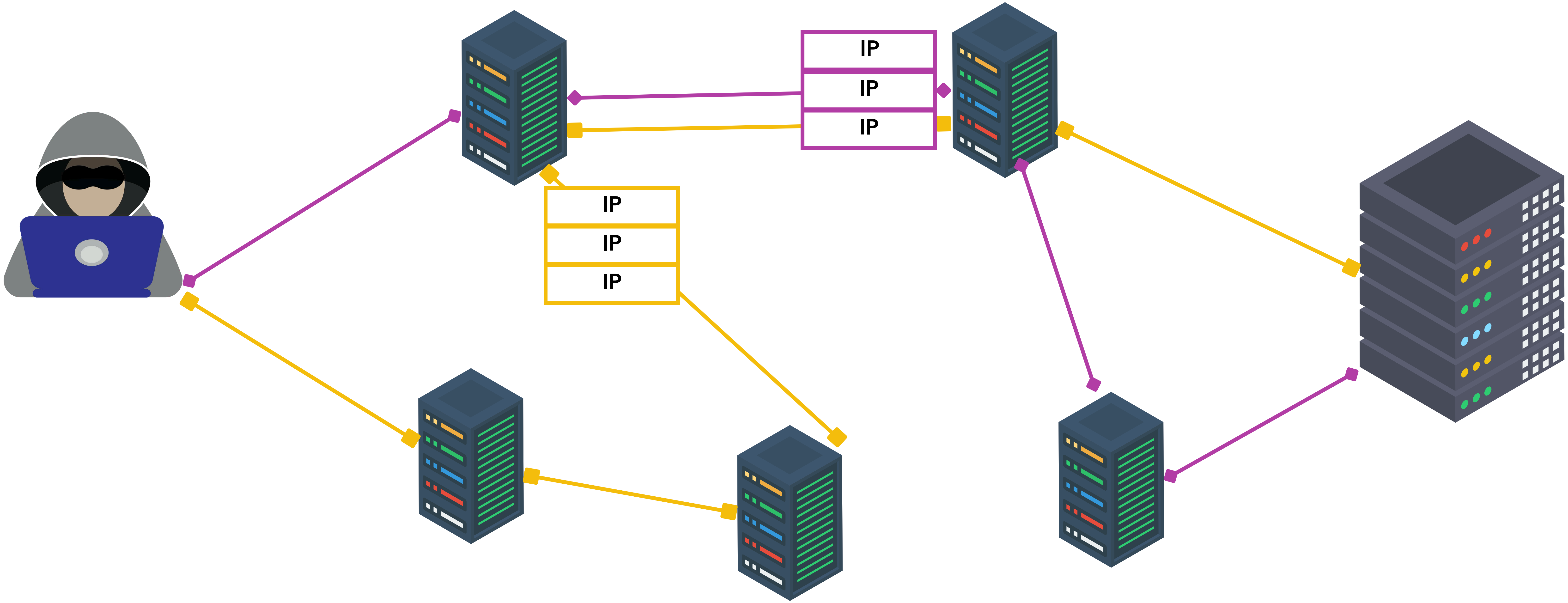
3) Schedule & Send



3) Schedule & Send

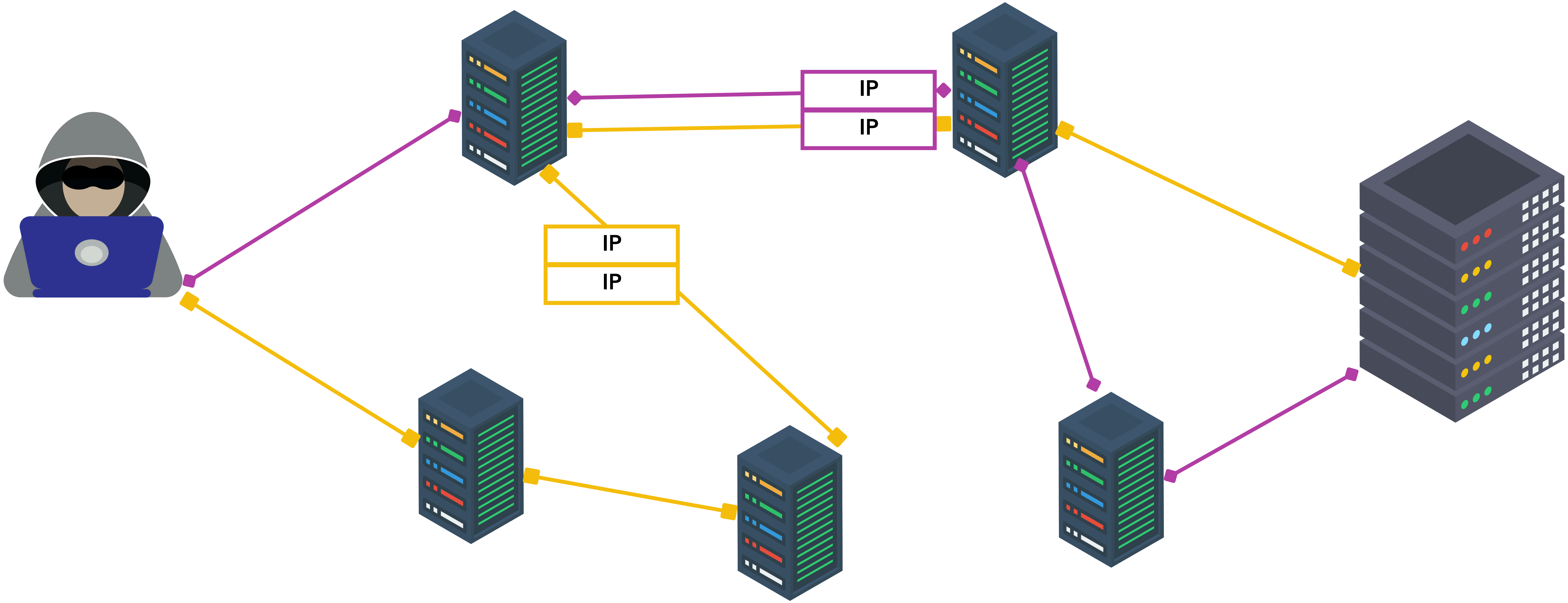


3) Schedule & Send

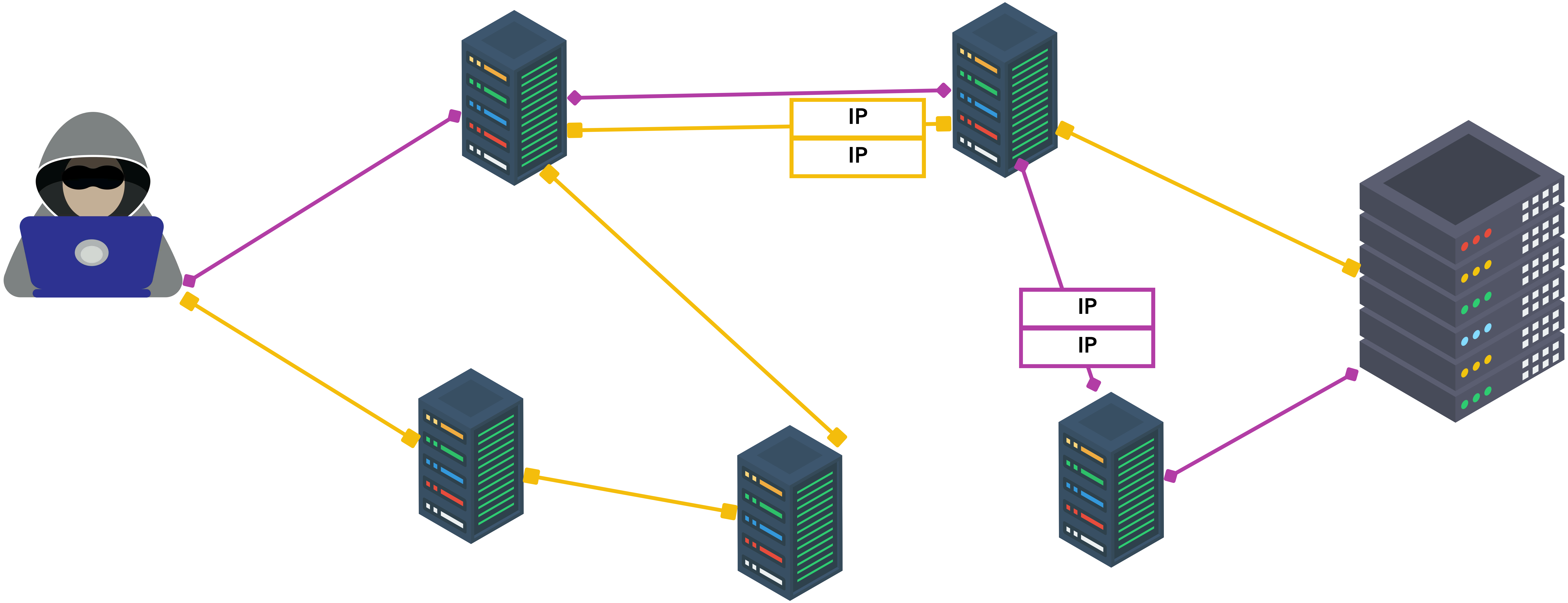


black hat BRIEFINGS

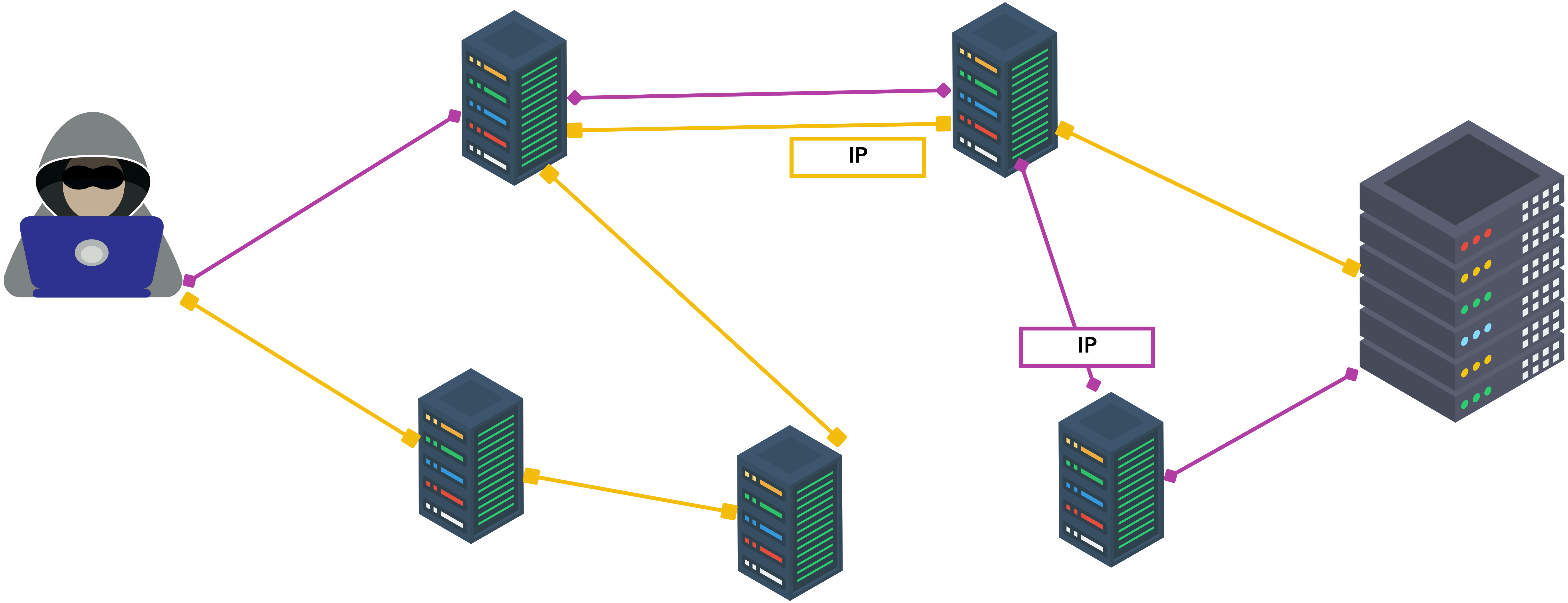
3) Schedule & Send



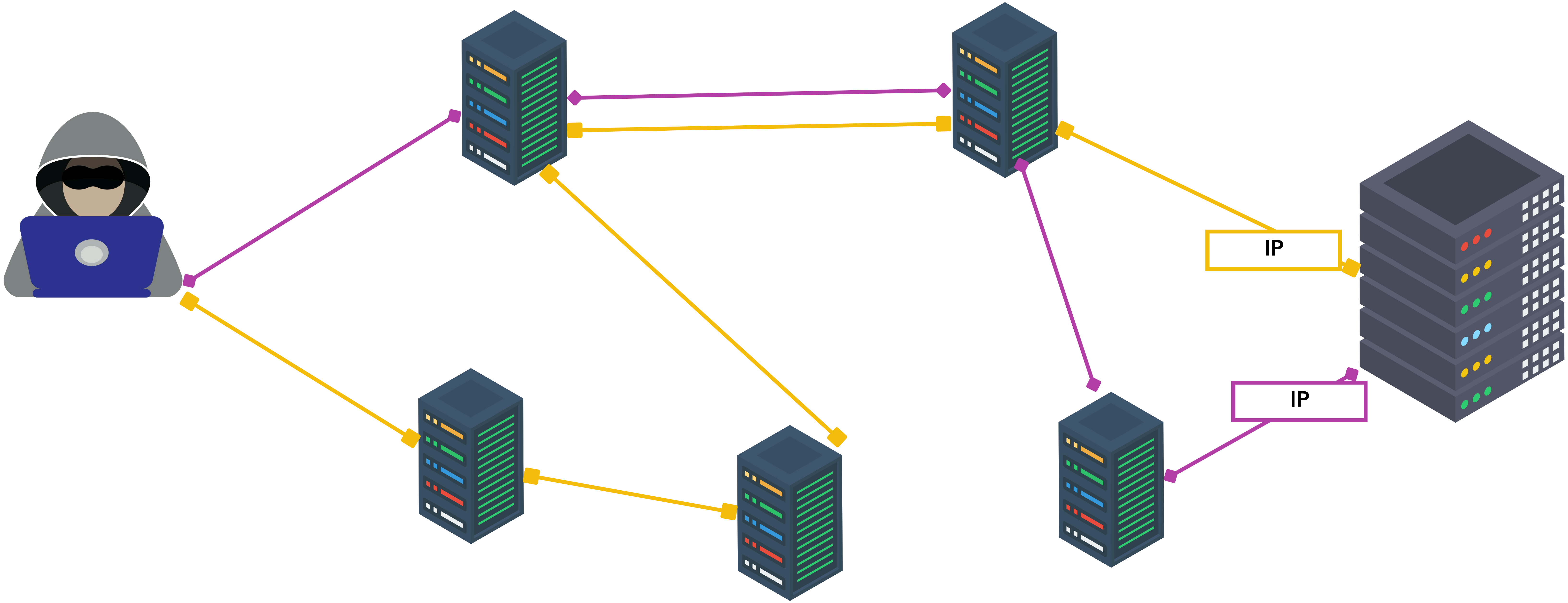
3) Schedule & Send



3) Schedule & Send

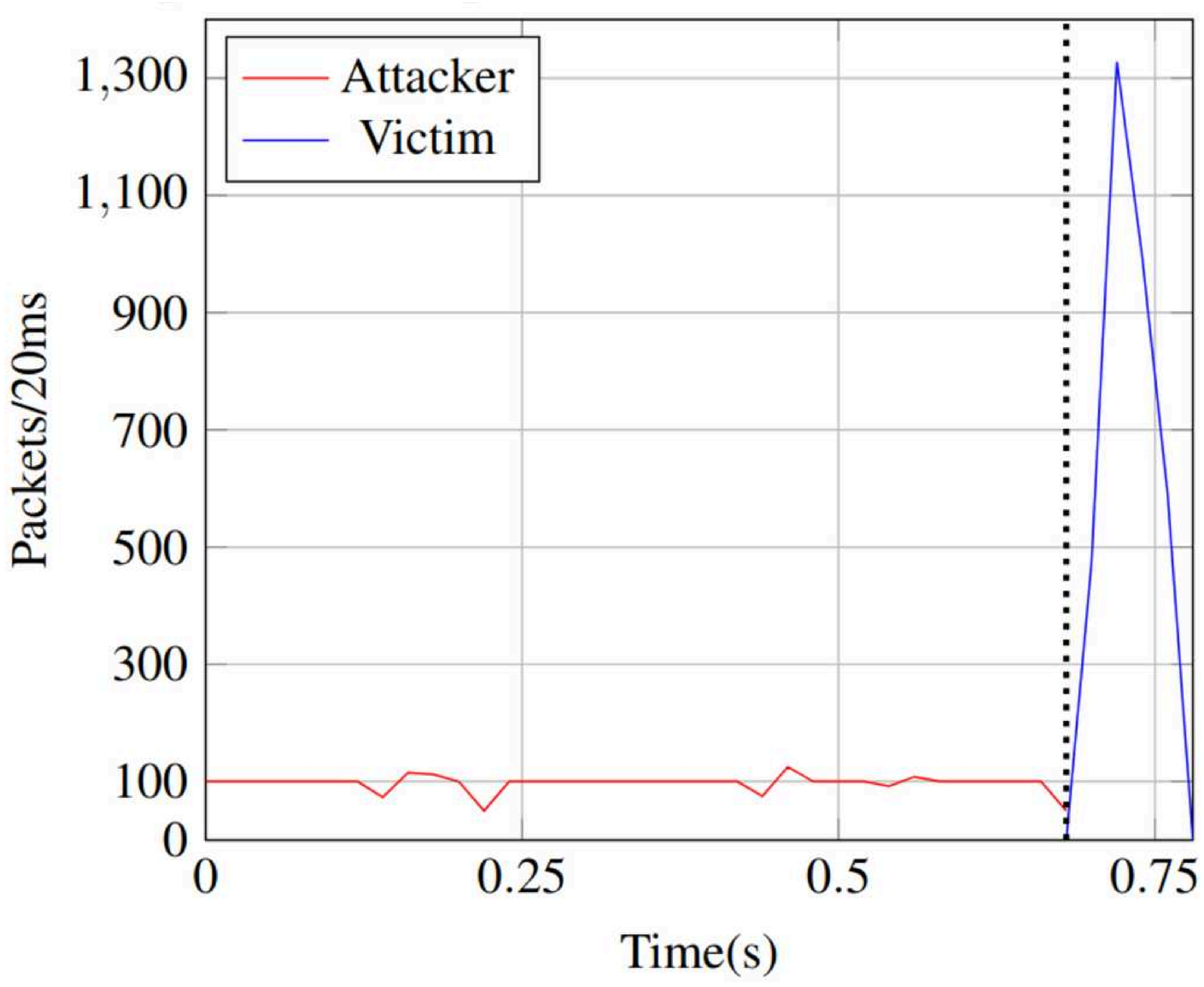


3) Schedule & Send



Tunneled-Temporal Lensing (TuTL)

- **5 hosts**
- **33 paths**
- **Avg. amplification: x16**



Mitigations

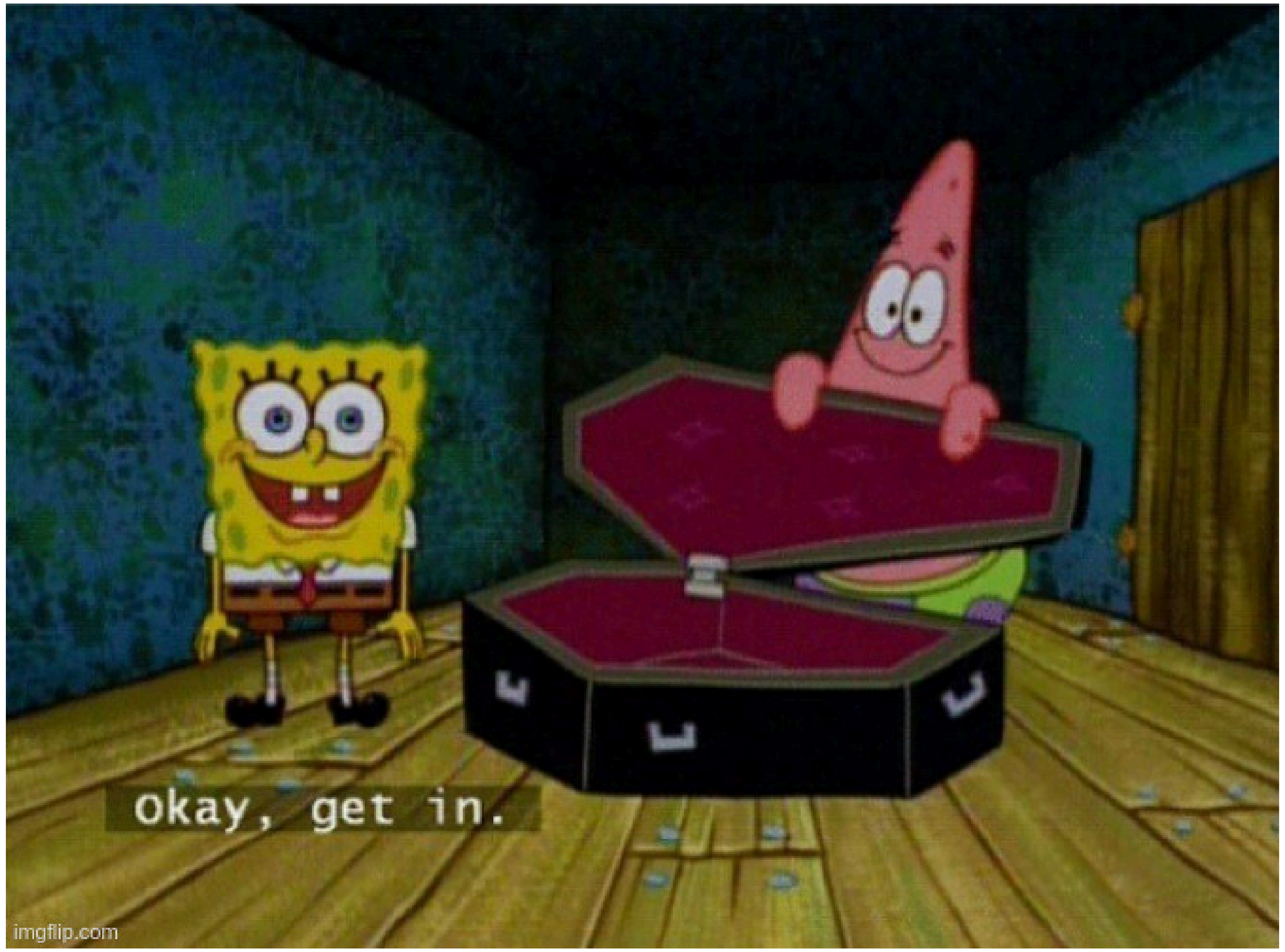
Network Defences

- **Source IP filtering (BCP38)**
- **Drop suspicious packets (DPI)**

Host Defences

- Authentication & Encryption (e.g., IPSec)
- Drop tunneled packets from untrusted sources
- Security Concerns with IP Tunneling- RFC 6169

Bonus

EOL = 

Conclusion

Black Hat Sound Bytes:

- **IP Spoofing became accessible (1.8 million)**
- **Open tunnels are still a thing (4 million)**
- **New (D)DoS variants are possible**

Thank you!

Haunted by Legacy: Discovering and Exploiting Vulnerable Tunnelling Hosts

Angelos Beitis
DistriNet, KU Leuven
angelos.beitis@kuleuven.be

Mathy Vanhoef
DistriNet, KU Leuven
Mathy.Vanhoef@kuleuven.be

Abstract

This paper studies the prevalence and security impact of open tunnelling hosts on the Internet. These hosts accept legacy or modern tunnelling traffic from any source. We first scan the Internet for vulnerable IPv4 and IPv6 hosts, using 7 different scan methods, revealing more than 4 million vulnerable hosts which accept unauthenticated IP in IP (IPIP), Generic Routing Encapsulation (GRE), IPv4 in IPv6 (4in6), or IPv6 in IPv4 (6in4) traffic. These hosts can be abused as one-way proxies, can enable an adversary to spoof the source address of packets, or can permit access to an organization's private network. The discovered hosts also facilitate new Denial-of-service (DoS) attacks. Two new DoS attacks amplify traffic: one concentrates traffic in time, and another loops packets between vulnerable hosts, resulting in an amplification factor of at least 16 and 75, respectively. Additionally, we present an Economic Denial of Sustainability (EDoS) attack, where the outgoing bandwidth of a host is drained. Finally, we discuss countermeasures and hope our findings will motivate people to better secure tunnelling hosts.

1 Introduction

Tunnelling protocols are an essential backbone of the Internet, with examples being the IP in IP (IPIP) and the Generic Routing Encapsulation (GRE) protocol [37, 49]. These protocols can link disconnected networks and form Virtual Private Networks (VPNs). One limitation of these protocols is that they do not authenticate or encrypt traffic. Instead, to secure these protocols, they must be combined with Internet Protocol Security (IPsec) [53].

Unfortunately, tunnelling protocols are often used without extra security, allowing an on-path or even off-path attacker to inject traffic into the tunnel [38, 64]. More troubling is how recently Yannay has demonstrated that misconfigured IPv4 hosts may accept unauthenticated IPIP tunnelling traffic from any source [39]. He also showed that an adversary can abuse these vulnerable hosts to spoof IPv4 addresses. Although this was an excellent discovery, several questions remain unanswered. In particular, it is unclear: (1) whether IPv6 hosts can also be vulnerable; (2) how to best scan for vulnerable hosts; (3) whether other tunnelling protocols can be vulnerable; (4) what the security implications of vulnerable tunnelling hosts are; and (5) what some practical defences are.

To answer these questions, we systematically analyse tunnelling protocols, including GRE, IP6IP6, Generic UDP Encapsulation (GUE), IPv4 in IPv6 (4in6), and IPv6 in IPv4 (6in4), and devise 7 different scanning methods to detect hosts that accept unencrypted tunnelling traffic from any source. We use these methods to scan the Internet for vulnerable IPv4 and IPv6 hosts. This reveals that all studied tunnelling protocols can be vulnerable: in total, we detected 3,527,565 vulnerable IPv4 hosts and 735,628 vulnerable IPv6 hosts.

Even more problematic is that many vulnerable hosts can be abused to spoof the source address of packets. This is significant since the increased adoption of source address filtering was thought to make spoofing harder for ordinary Internet users [6]. Vulnerable hosts may also allow access to an organisation's private network, where carefully constructed tunnelling packets can reveal details about the victim's (private) network, facilitating subsequent attacks. Additionally, we present an administrative Denial-of-service (DoS) attack, where traffic is spoofed to trigger incorrect abuse reports towards the victim.

We also investigate how vulnerable hosts can facilitate new DoS attacks. We uncover two new amplification DoS attacks, called *Tunnelled-Temporal Lensing (TuTL)* and the *Ping-Pong* attack, with an amplification factor of at least 16 and 75, respectively. Additionally, we present an Economic Denial of Sustainability (EDoS) attack, where the outgoing bandwidth of a host is consumed to incur unexpected financial costs or lead to service disruptions.

To summarise, our main contributions are:

- We scan the Internet for vulnerable hosts that accept tunnelling packets from any source (Section 3 and 4).
- We present a new administrative DoS attack (Section 4).
- We introduce three tunnelling-based DoS attacks: Ping-

TunnelTester script: https://github.com/vanhoefm/tunneltester