Exploiting Sequence Number Leakage: TCP Hijacking in NAT-Enabled Wi-Fi Networks

Yuxiang Yang, Xuewei Feng, Qi Li, Kun Sun, Ziqiang Wang, Ke Xu

Overview

- Threat Model
- Background
- Attack Procedure
- Empirical Study
- Disclosure and Mitigation
- Conclusion

Threat Model

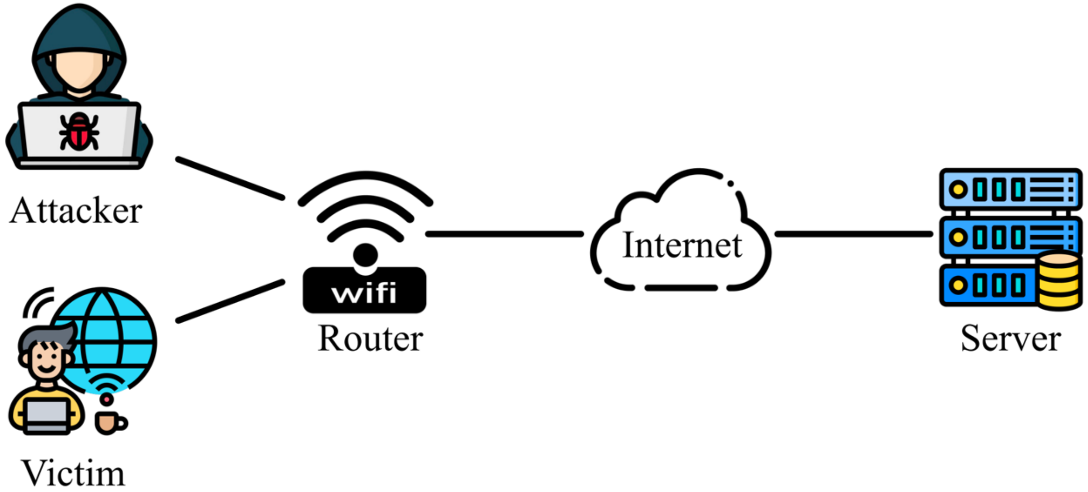
Threat Model

✐**Consists of:**

- An arbitrary remote **server**
- A **router**, providing Wi-Fi
- A victim **client** who connected to Wi-Fi
- An off-path **attacker** who can access the same Wi-Fi

✐**The attack can be used towards:**

- TCP **connection termination** attack
- TCP **packets hijacking** attack
- Malicious **data injection** attack



Background

History of TCP Hijacking Attacks

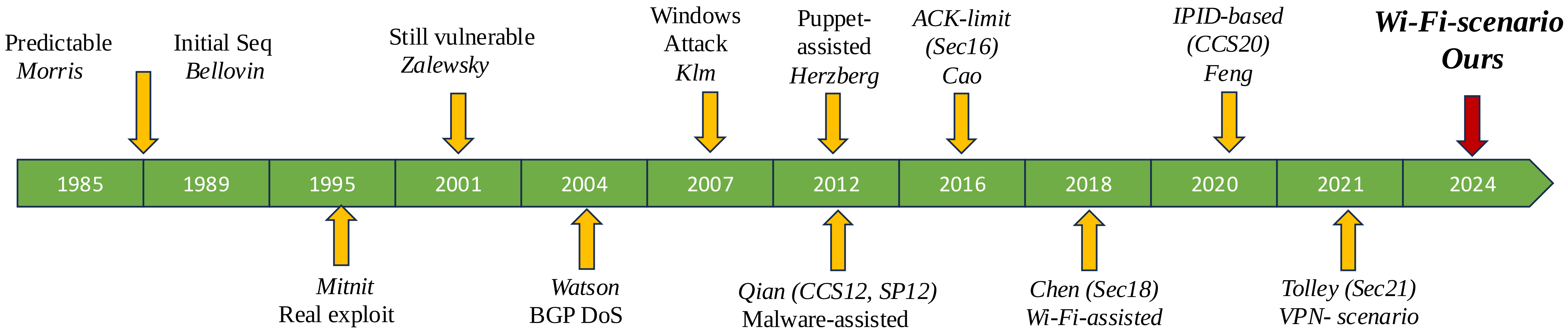
✎ Given a target server, we already know:

- Src IP address: client's public IP (the same as the attacker)
- Dst IP address: server IP
- Dst Port number: service at server (e.g. 80)

✎ We want to know:

- **Src Port number**
- **SEQ number**
- **ACK number**

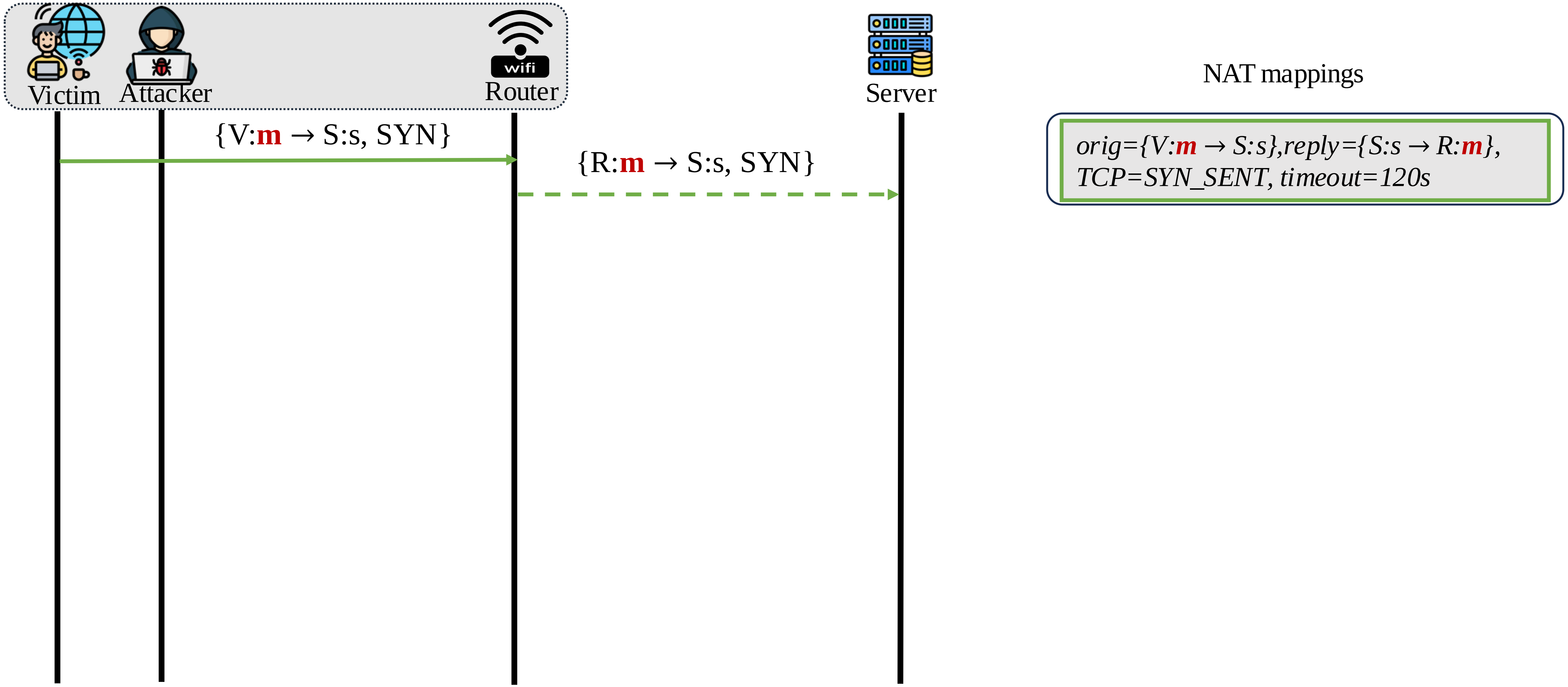
```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Acknowledgment Number                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Data |           |U|A|P|R|S|F|                               |
| Offset| Reserved  |R|C|S|S|Y|I|            Window             |
|       |           |G|K|H|T|N|N|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |         Urgent Pointer        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             data                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```



NAT and Port Allocation Strategies

✐NAT Port Allocation Method:

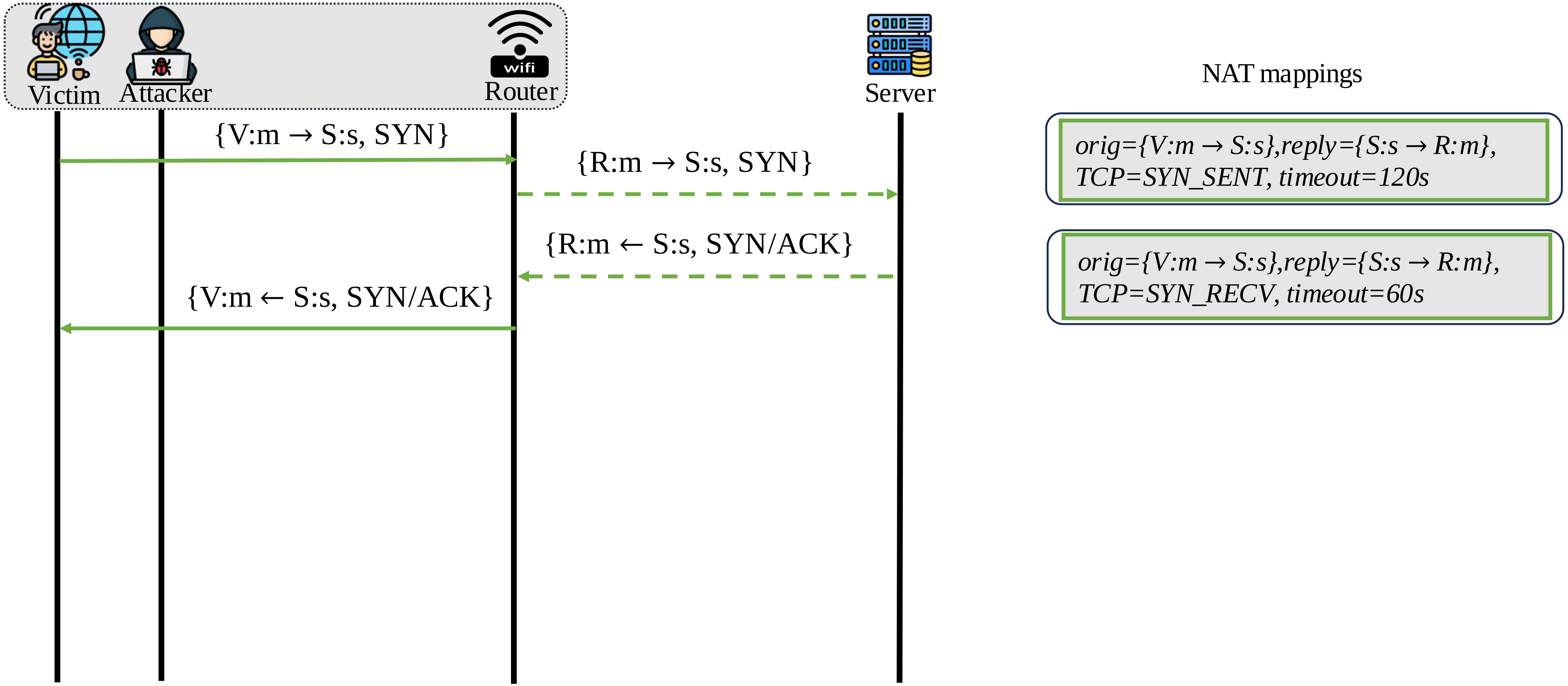
(1) random allocation; (2) per-destination sequential; (3) **port preserving allocation**



NAT and Port Allocation Strategies

✐NAT Port Allocation Method:

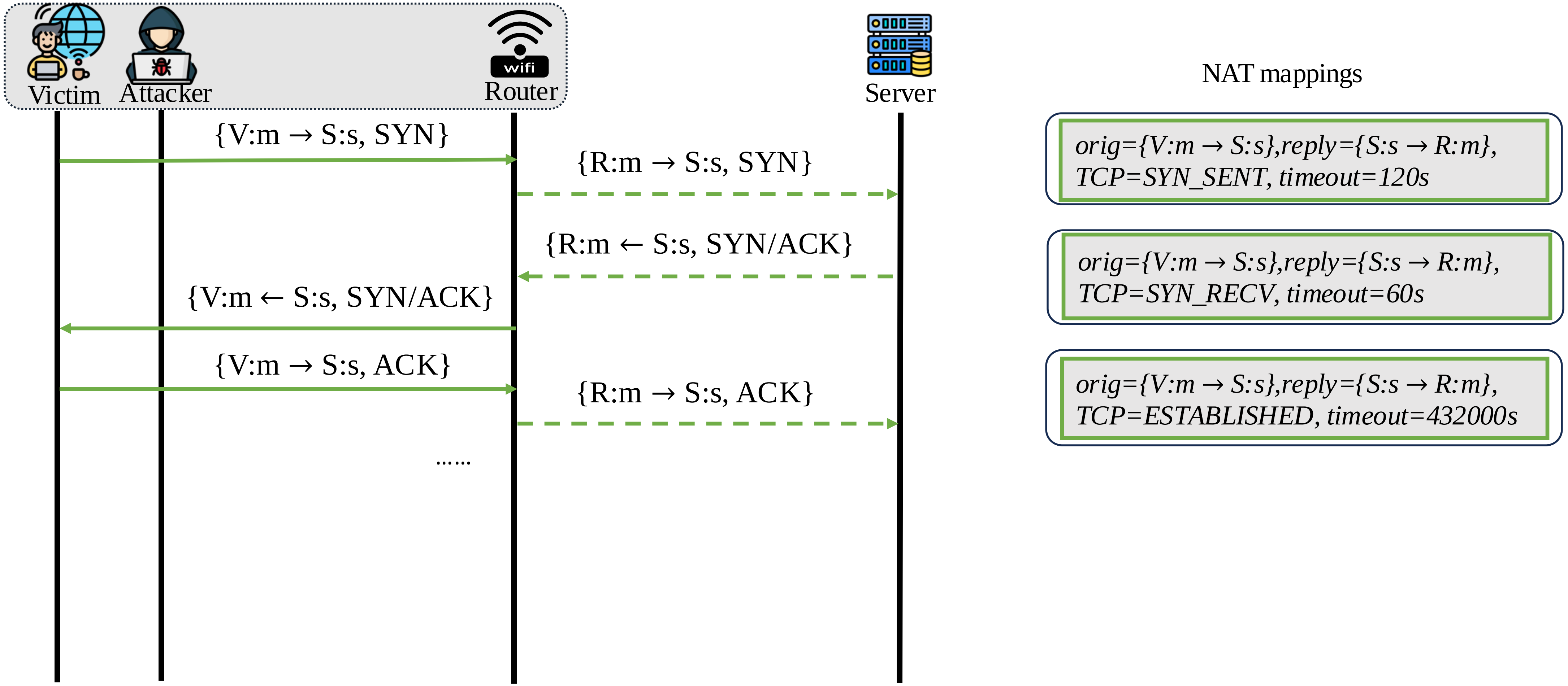
(1) random allocation; (2) per-destination sequential; (3) **port preserving allocation**



NAT and Port Allocation Strategies

NAT Port Allocation Method:

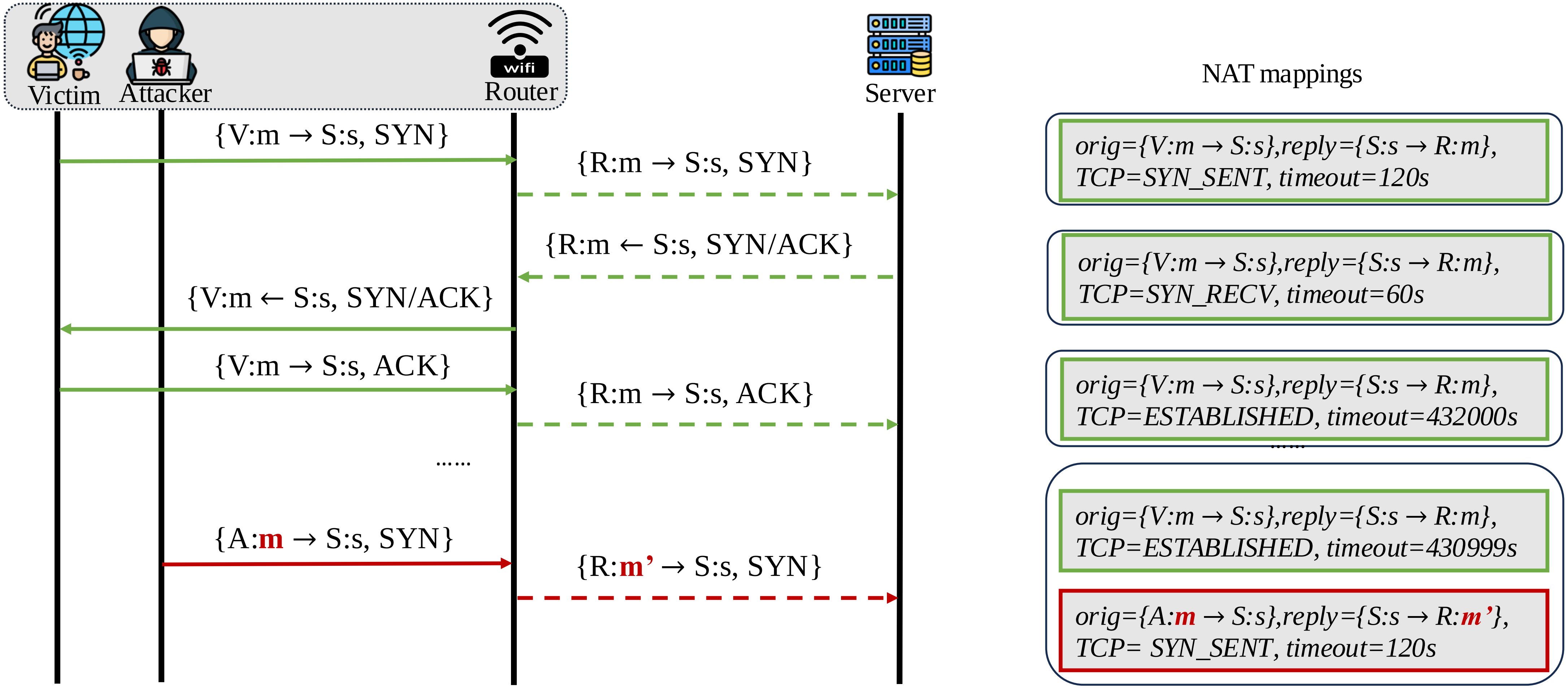
(1) random allocation; (2) per-destination sequential; (3) **port preserving allocation**



NAT and Port Allocation Strategies

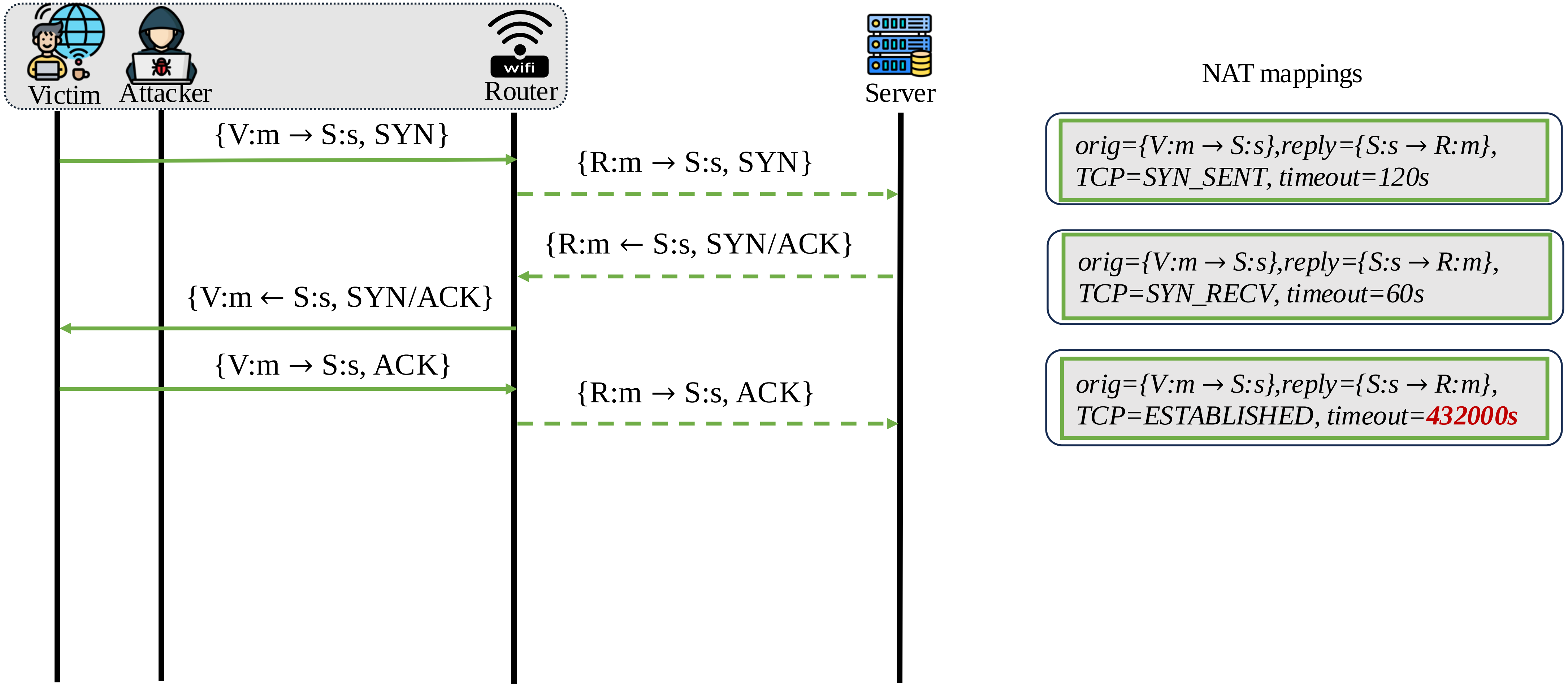
NAT Port Allocation Method:

(1) random allocation; (2) per-destination sequential; (3) **port preserving allocation**



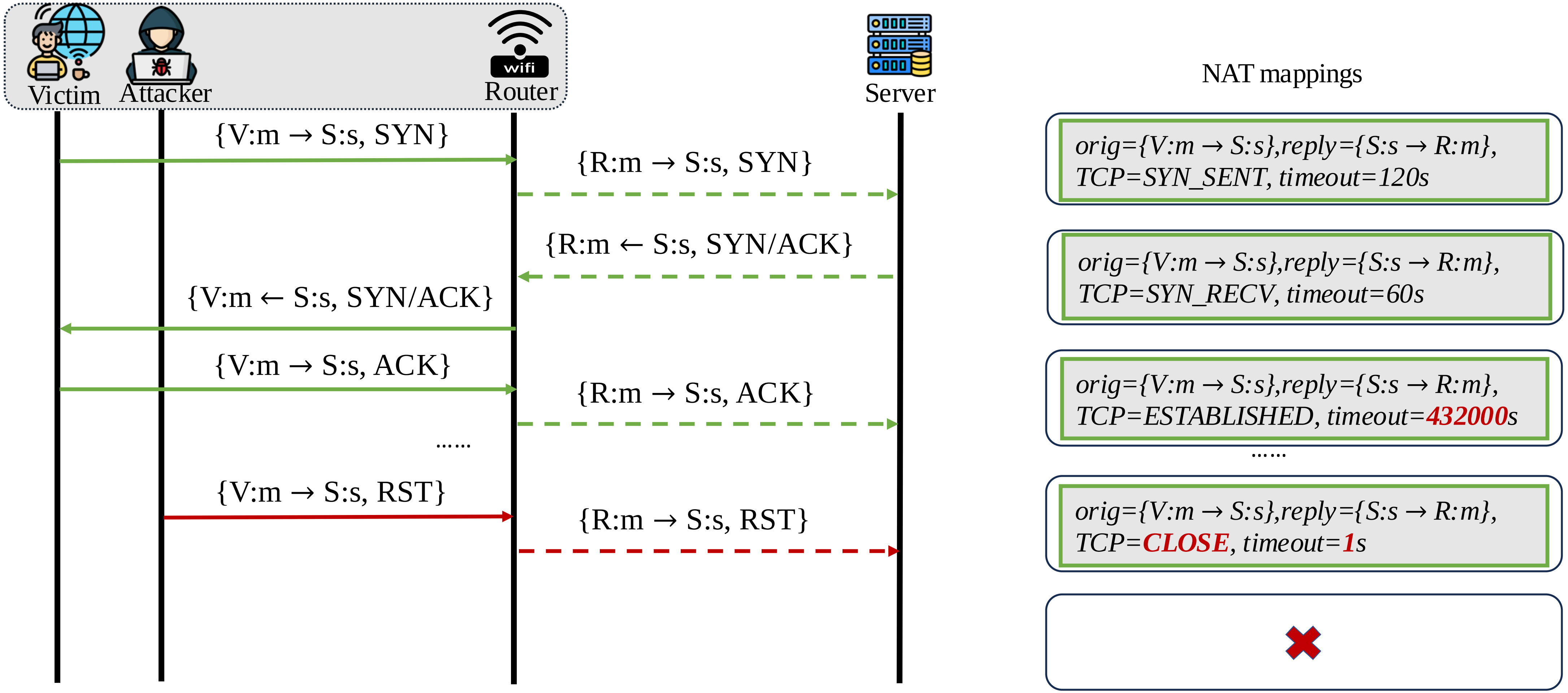
TCP Window Tracking in Routers

Due to many reasons, router will not track the TCP window of the connection, and thus it **will not check the sequence** and acknowledgment **numbers** of TCP packets strictly.



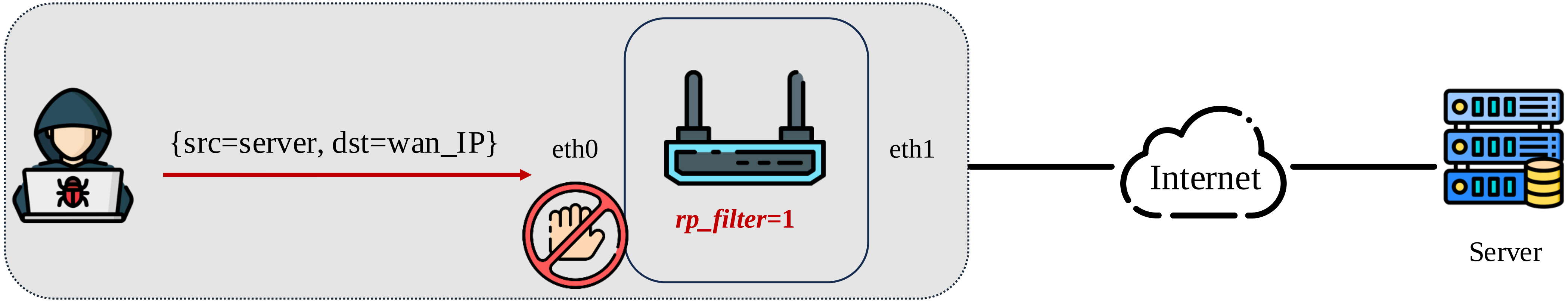
TCP Window Tracking in Routers

Due to many reasons, router will not track the TCP window of the connection, and thus it **will not check the sequence** and acknowledgment **numbers** of TCP packets strictly.



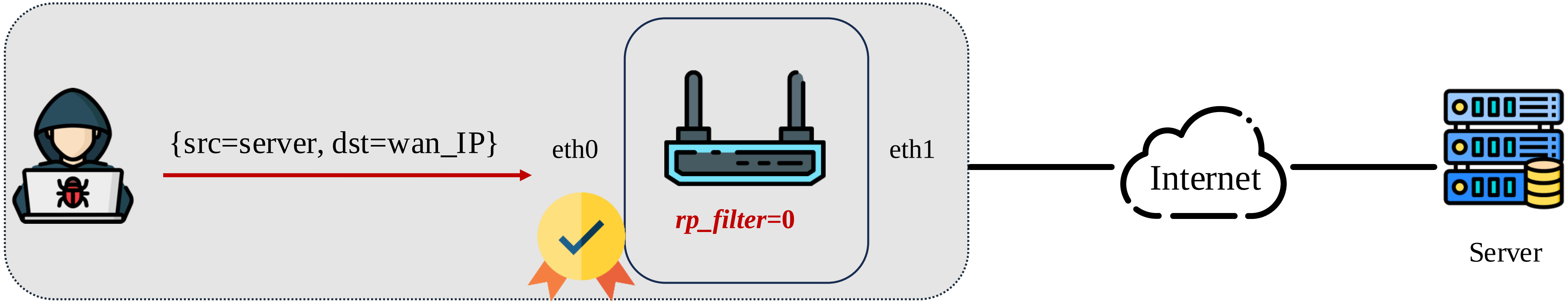
Reverse Path Validation

- **Proposed by RFC 3704, to prevent IP spoofing attacks**
  - verifies inbound traffic by checking whether **the source IP address** can **be routed back via the interface** on which packets are received against the routing table.
- **Controlled by the *rp_filter* kernel variable.**
  - 0: disabled;
  - 1: Strict Mode;
  - 2: Loose Mode



Reverse Path Validation

- **Proposed by RFC 3704, to prevent IP spoofing attacks**
  - verifies inbound traffic by checking whether **the source IP address** can **be routed back via the interface** on which packets are received against the routing table.
- **Controlled by the *rp_filter* kernel variable.**
  - 0: disabled; • 1: Strict Mode; • 2: Loose Mode

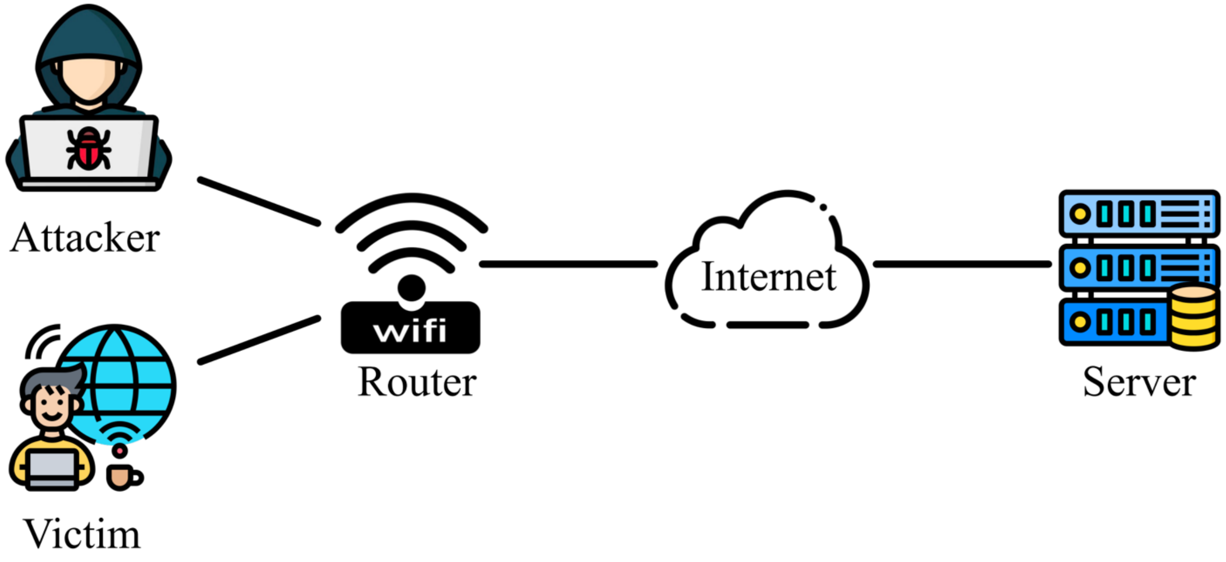


ATTACK PROCEDURE

Attack Overview

Attack Steps:

- Step 1: Probing the Wi-Fi Network (to get the **router's external IP** address)
- Step 2: Making Inferences about Active Connections (to infer the **source port** number)
- Step 3: Hijacking Active Connections (to get the **SEQ and ACK** numbers)

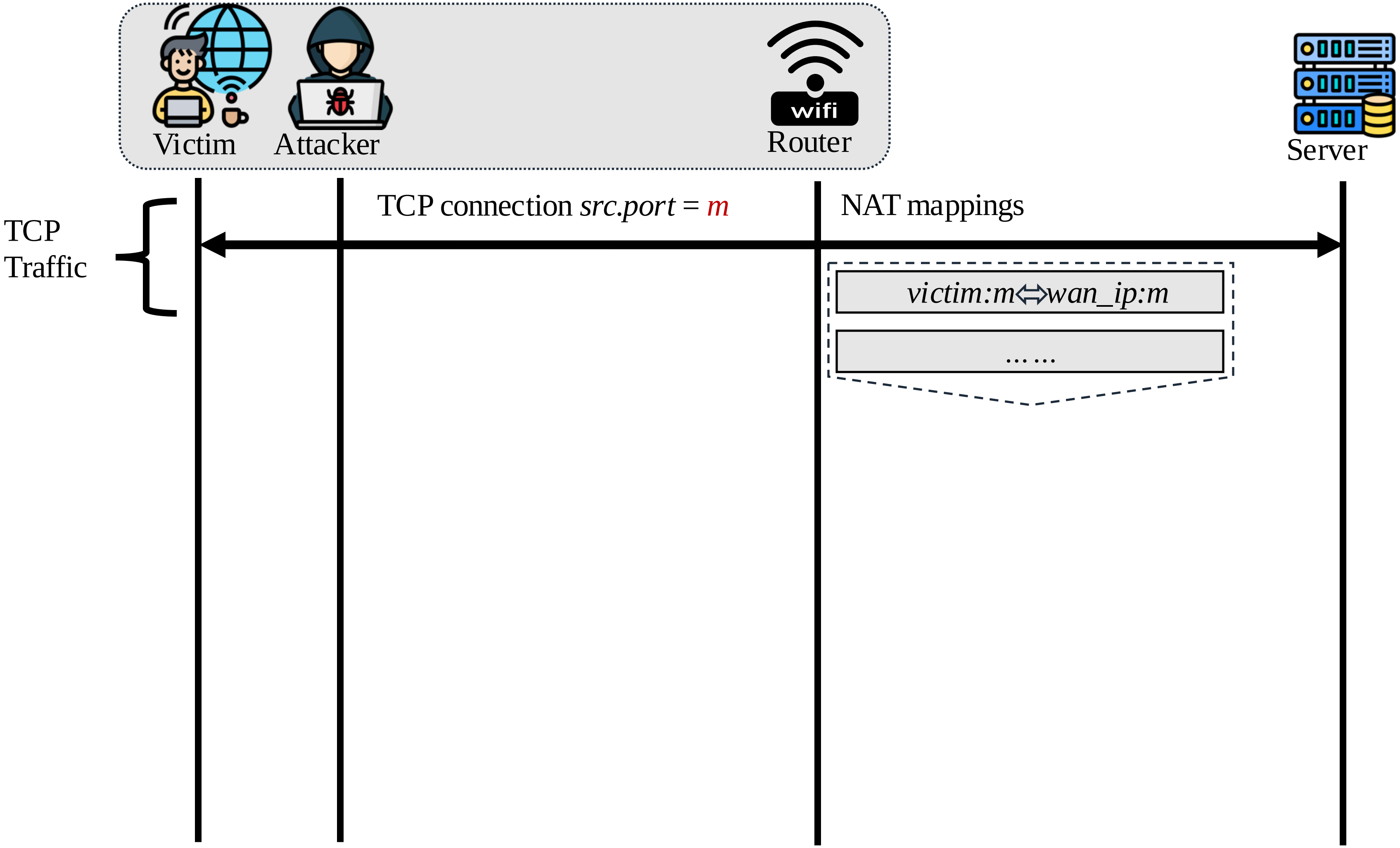


Probing the Network

- **Identifying the status of AP isolation in the network**
  - Nmap, MacStealer
- **Probing the external IP address of the router**
  - TraceRoute and Ping with RECORD_ROUTE Option.
  - Scan and access IPs via web browsers.

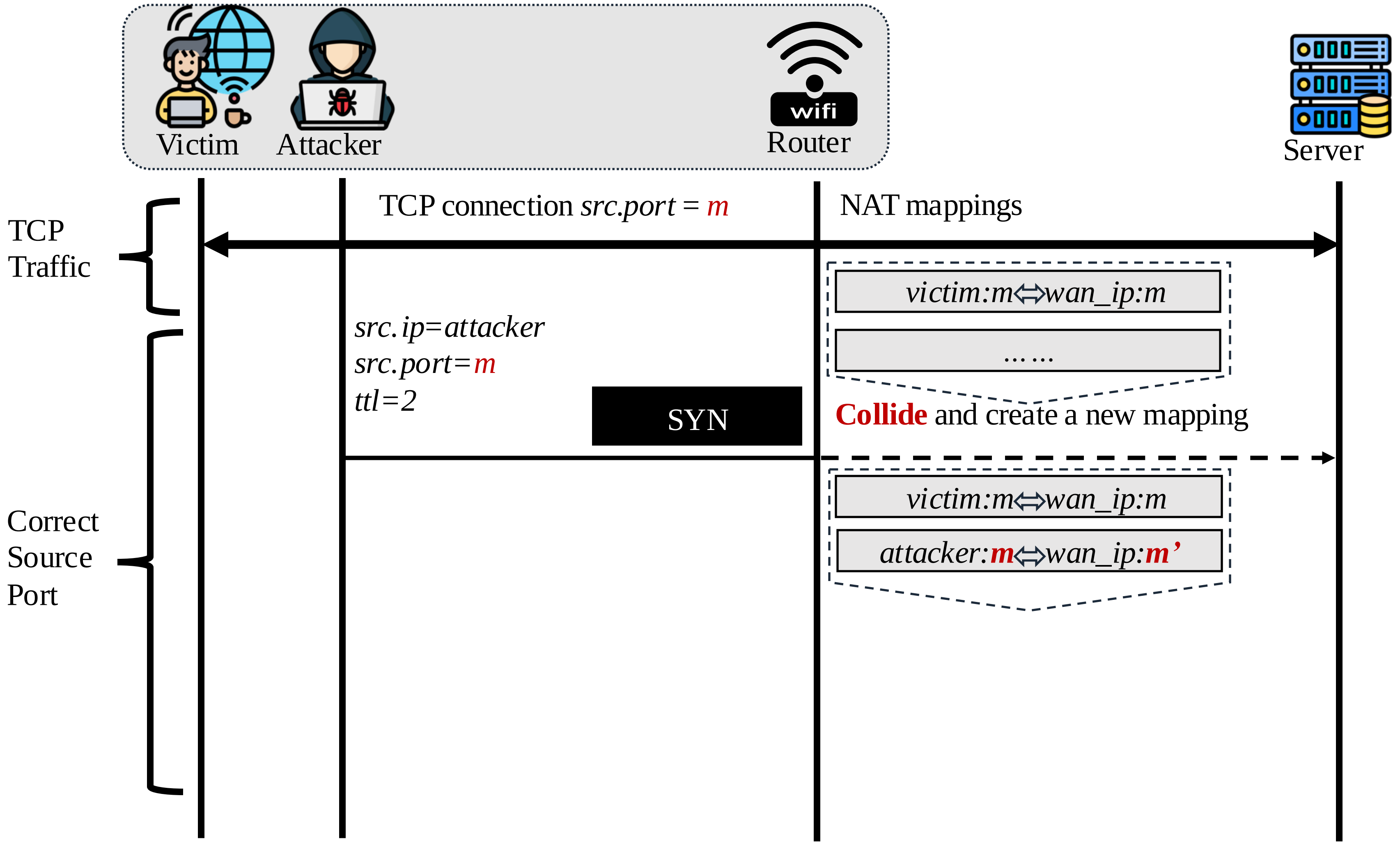
```
# parallels @ ubuntu-linux-22-04-desktop in ~/Desktop [10:10:09]
$ traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  8A7770.lan (10.254.0.1)  30.586 ms  31.126 ms  31.238 ms
 2  100.64.0.1 (100.64.0.1)  103.118 ms  103.133 ms  103.576 ms
 3  14.148.21.29 (14.148.21.29)  103.552 ms  103.530 ms  103.648 ms^C

# parallels @ ubuntu-linux-22-04-desktop in ~/Desktop [10:10:16] C:130
$ ping -R 100.64.0.1
PING 100.64.0.1 (100.64.0.1) 56(124) bytes of data.
64 bytes from 100.64.0.1: icmp_seq=1 ttl=254 time=54.7 ms
RR:     10.254.205.199
        100.64.129.73
        100.64.0.1
        10.254.0.1
        10.254.205.199
```

Making Inferences about Active Connections



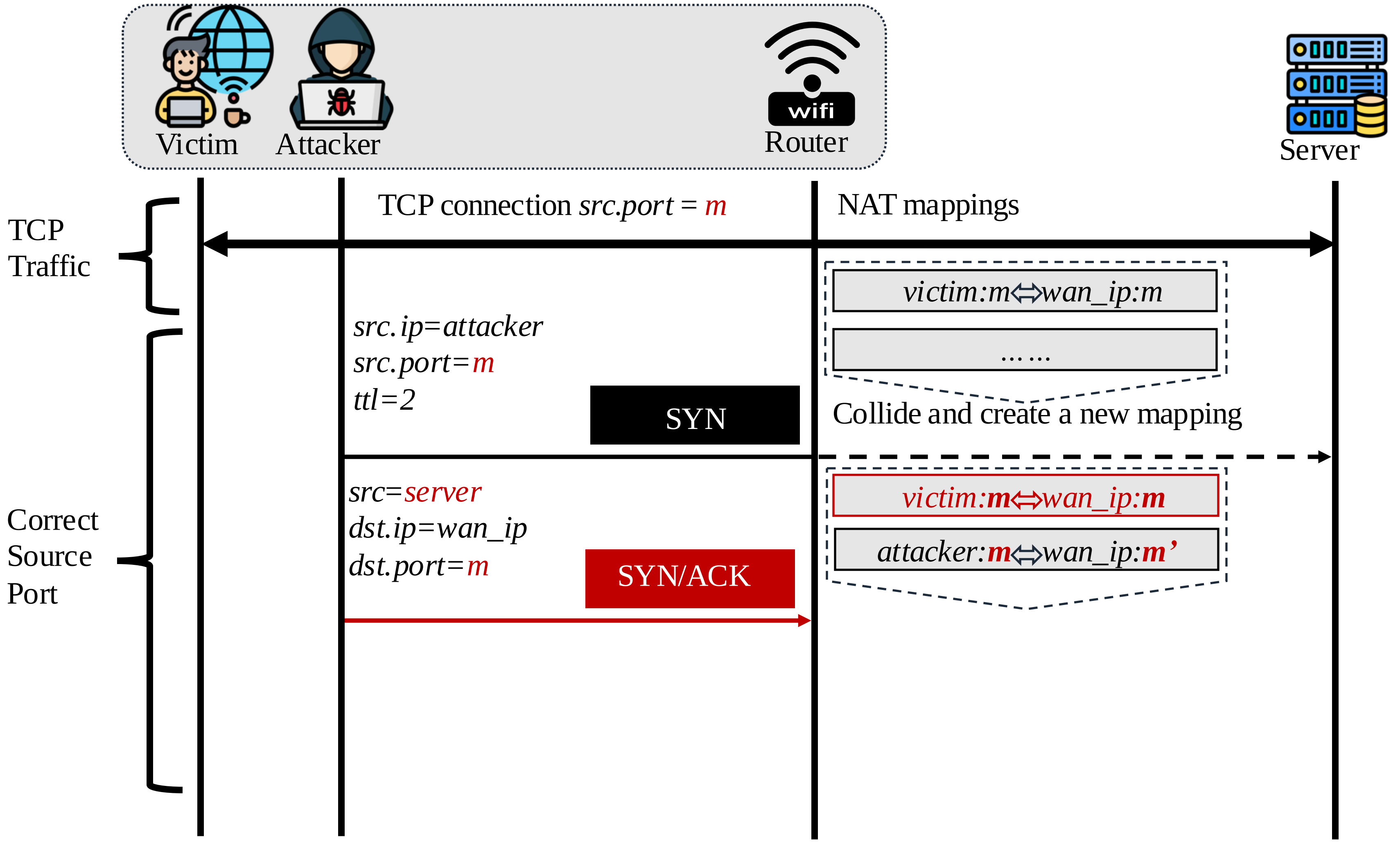
Guess when the source port **has been used** by the victim.

Making Inferences about Active Connections



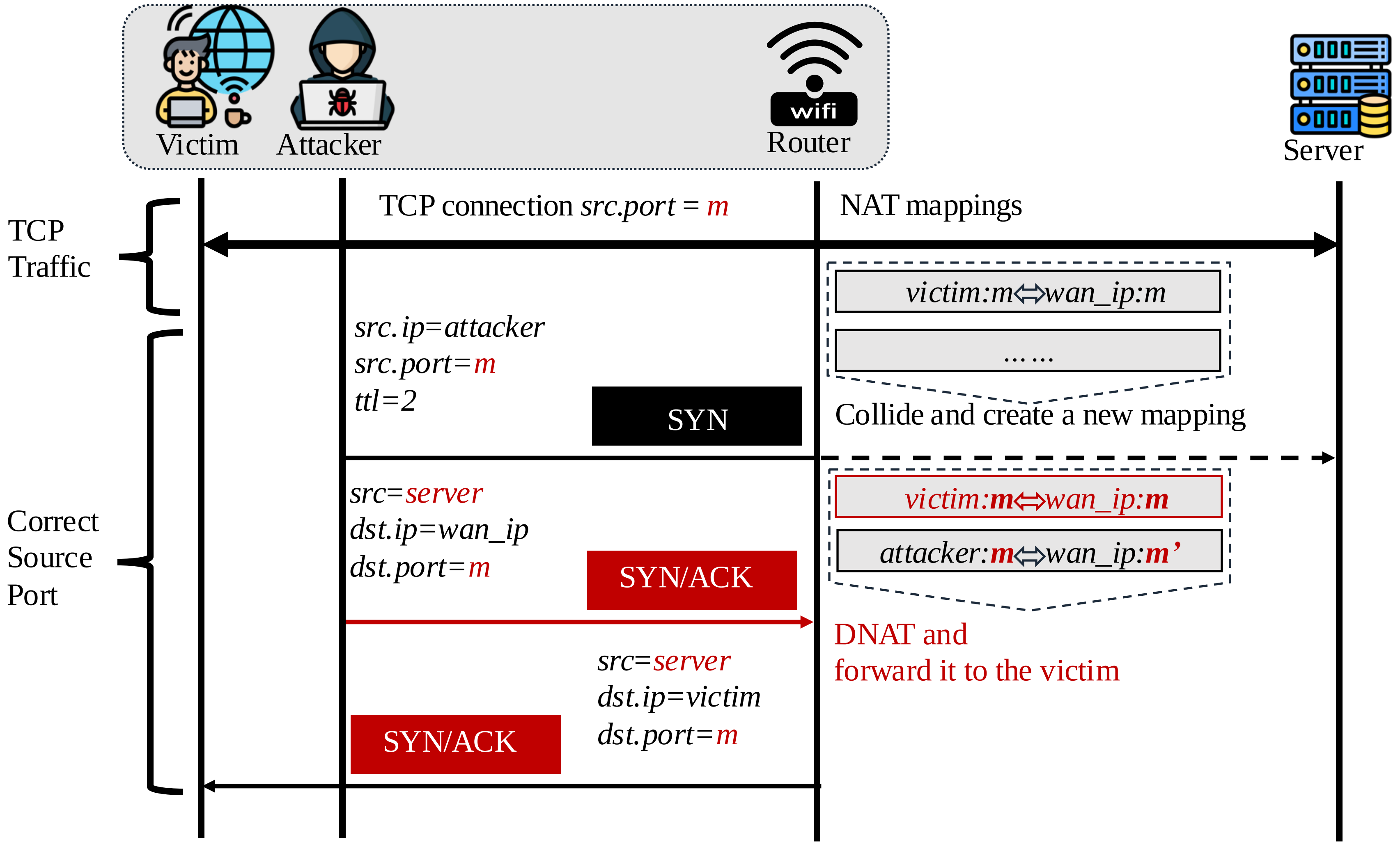
Guess when the source port **has been used** by the victim.

Making Inferences about Active Connections



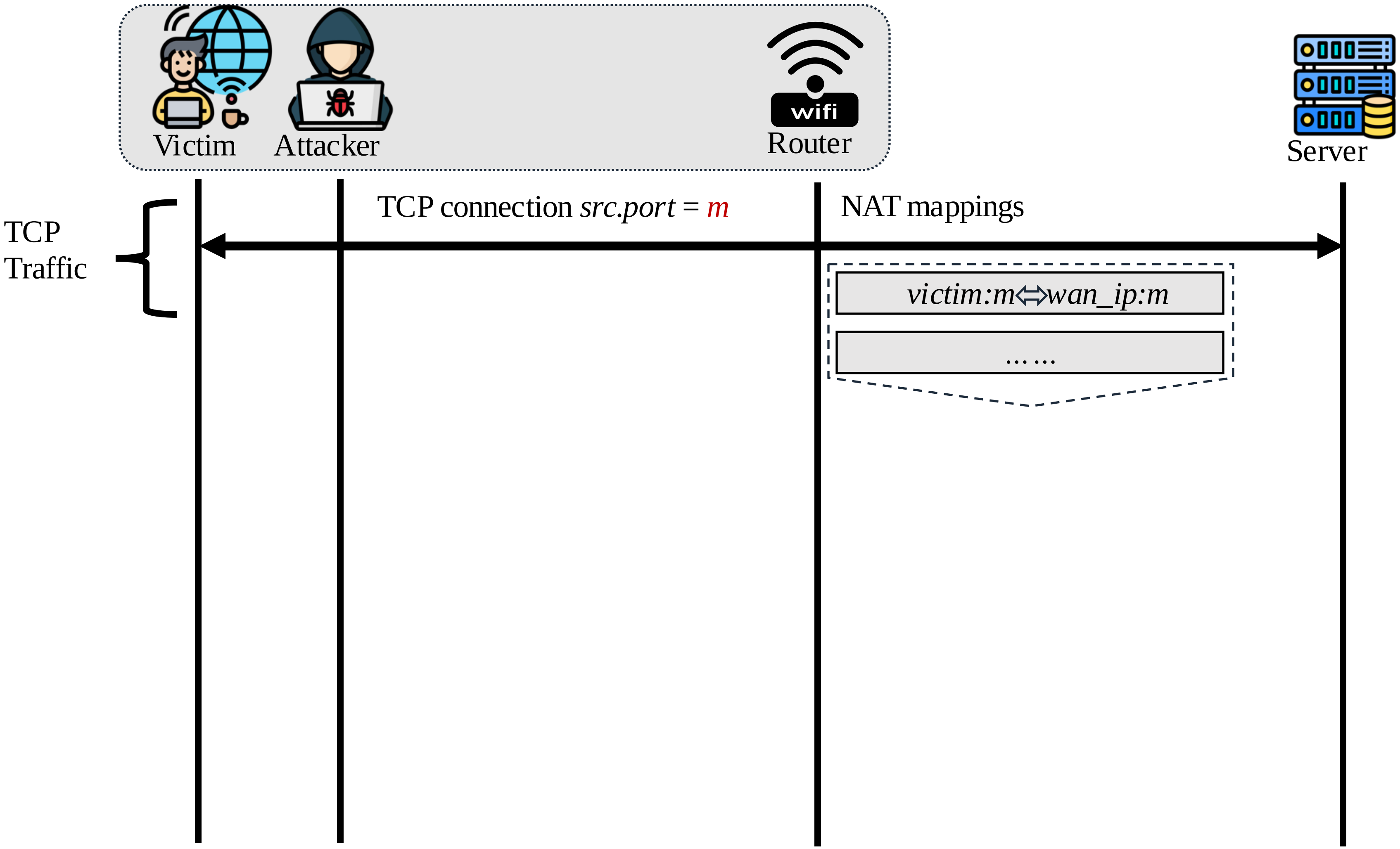
Guess when the source port **has been used** by the victim.

Making Inferences about Active Connections



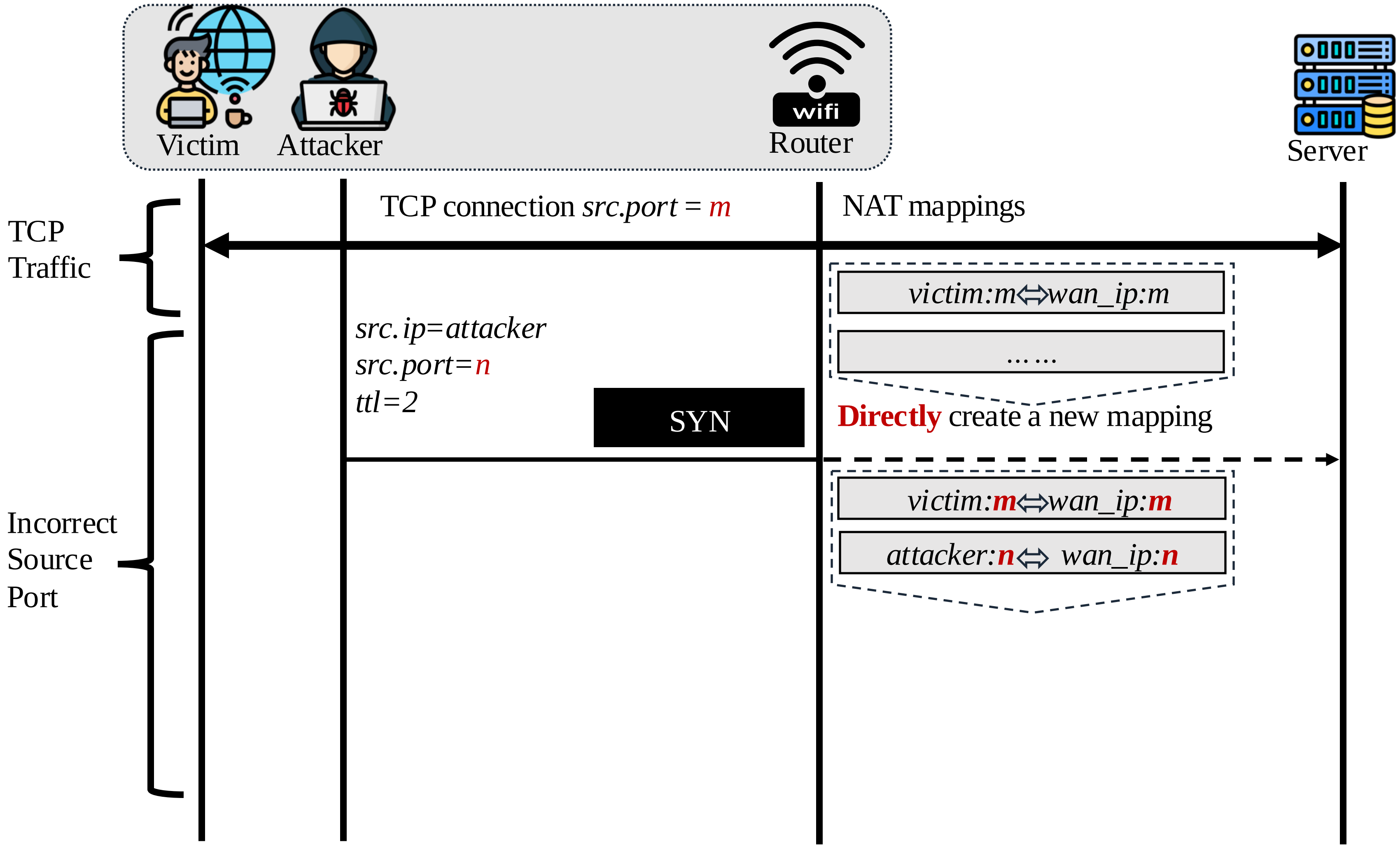
Guess when the source port **has been used** by the victim.

Making Inferences about Active Connections



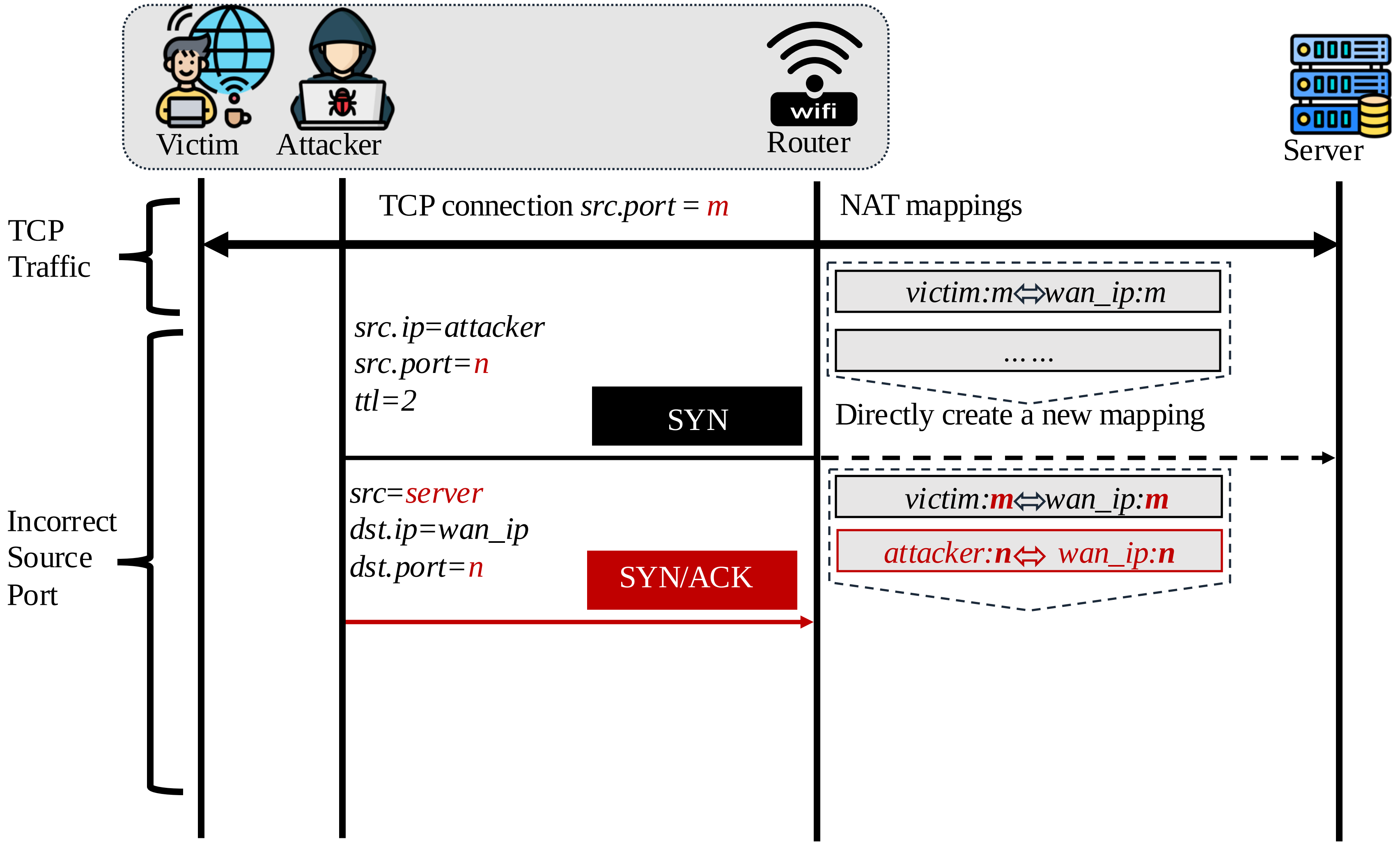
Guess when the source port **has not been used** by the victim.

Making Inferences about Active Connections



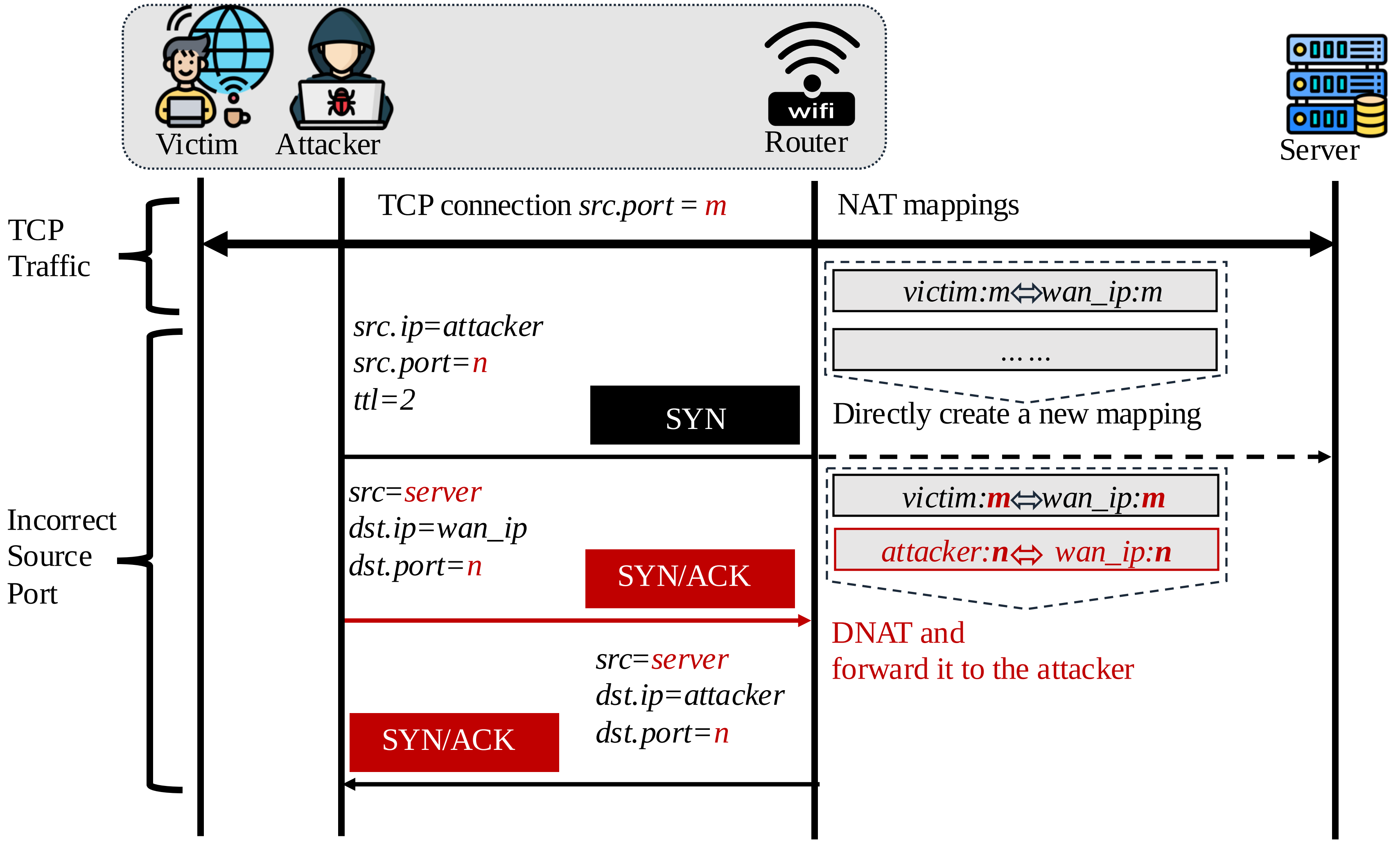
Guess when the source port **has not been used** by the victim.

Making Inferences about Active Connections



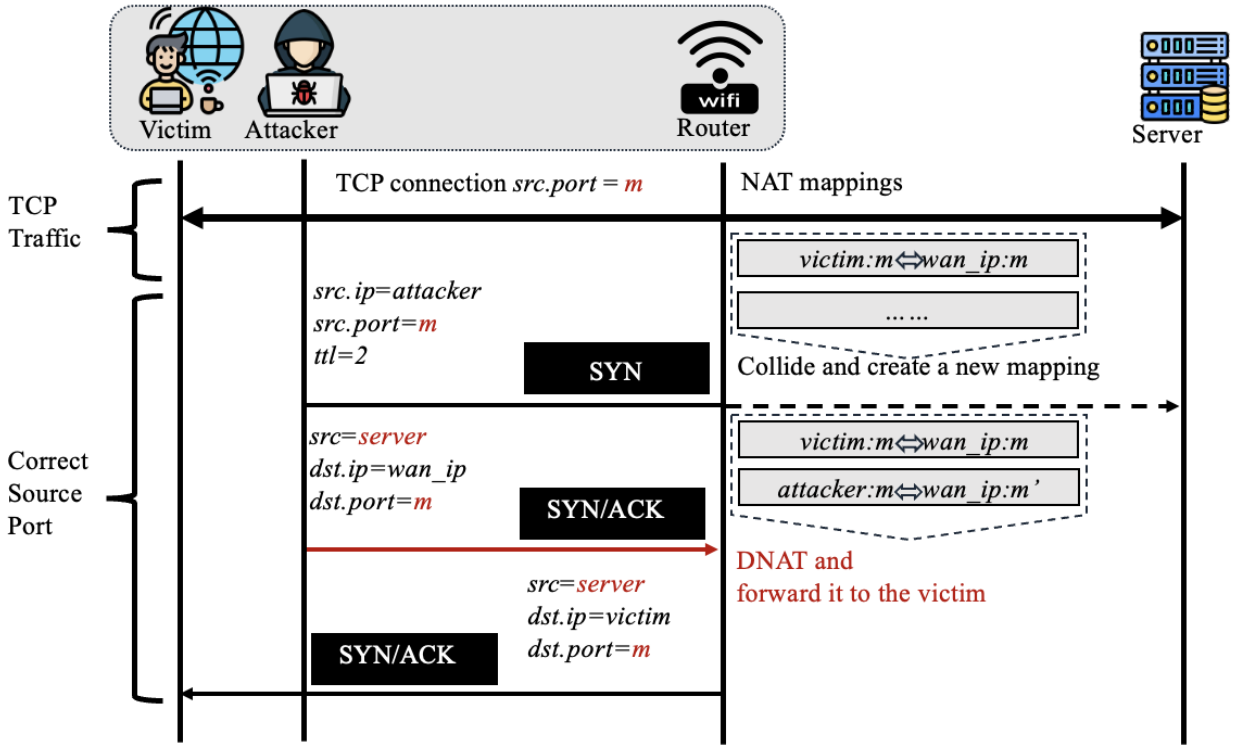
Guess when the source port **has not been used** by the victim.

Making Inferences about Active Connections

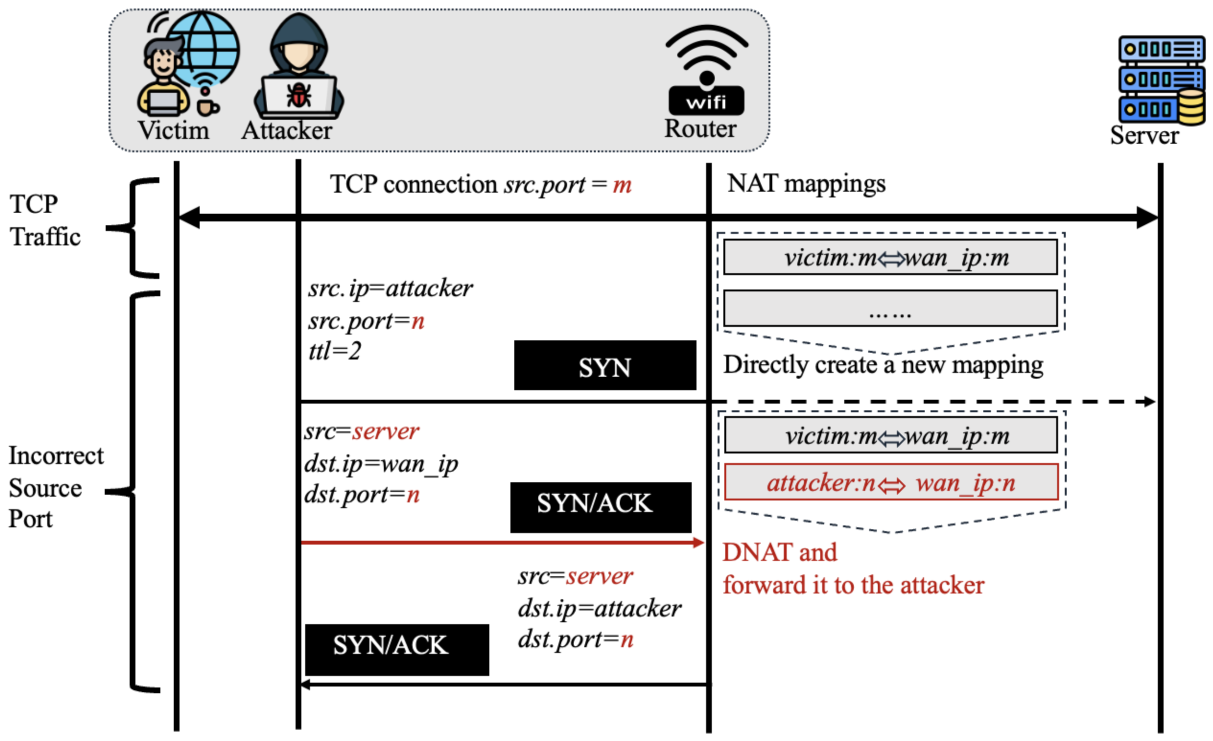


Guess when the source port **has not been used** by the victim.

Making Inferences about Active Connections

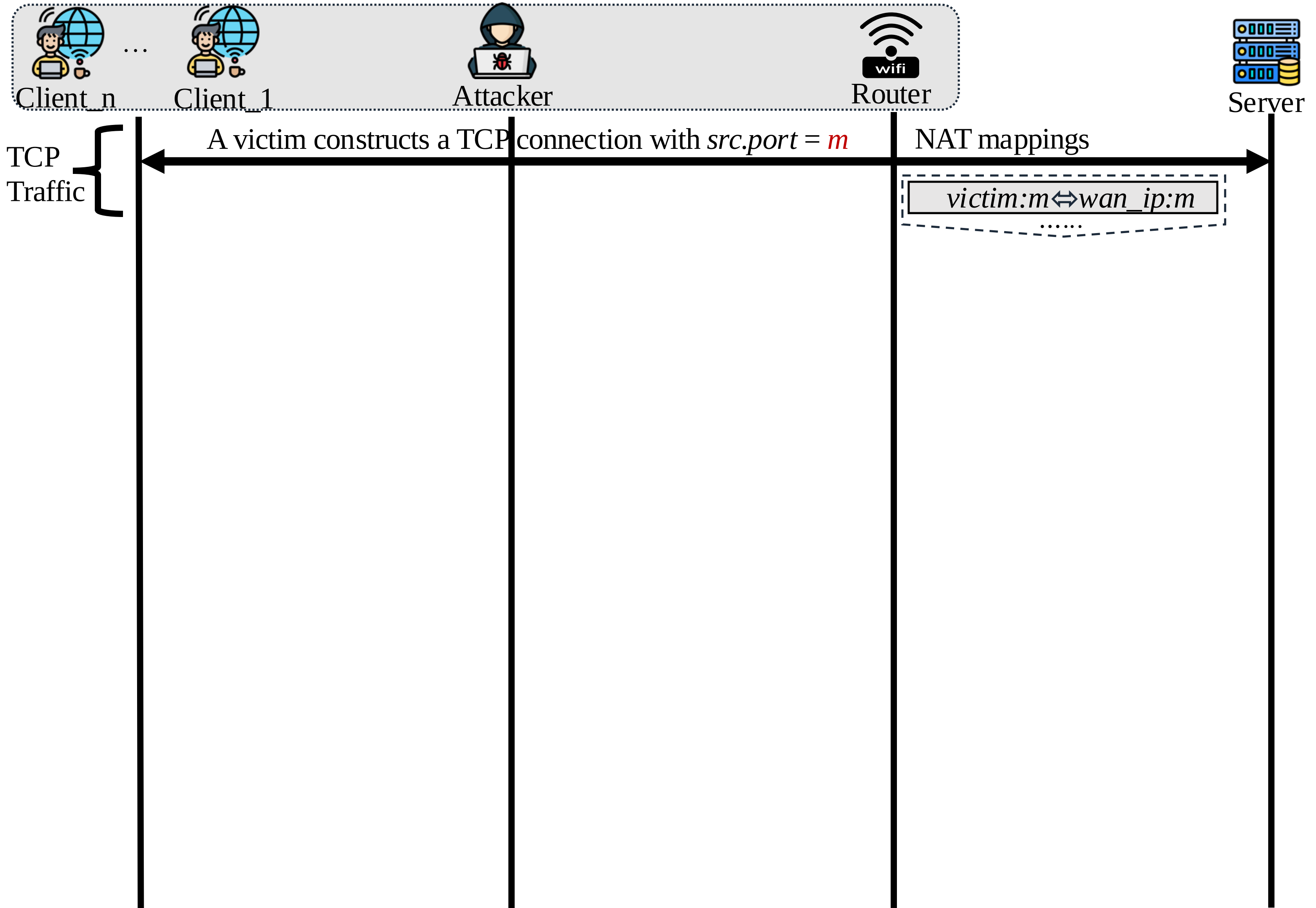


Guess when the source port is **used**

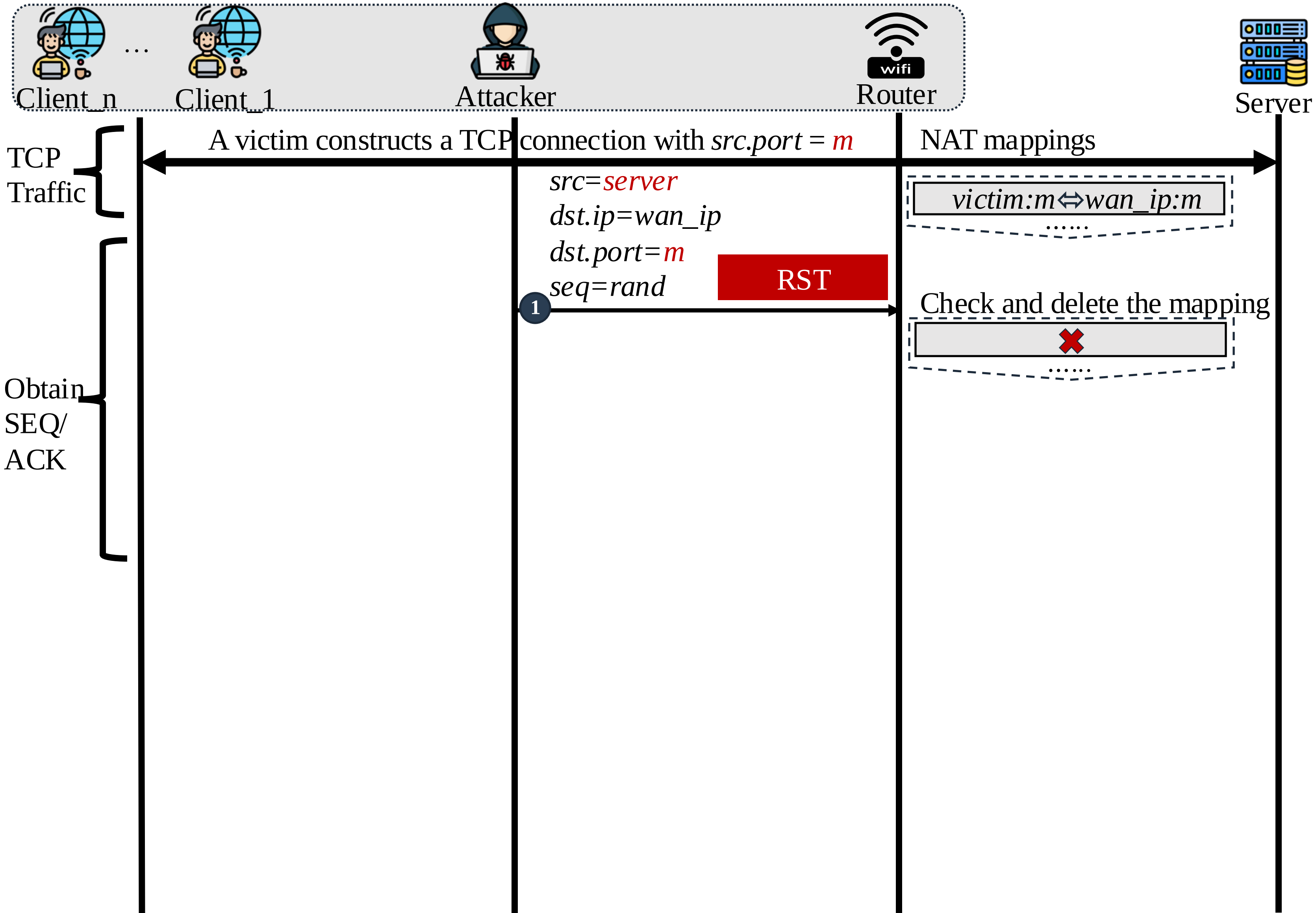


Guess when the source port is **not used**

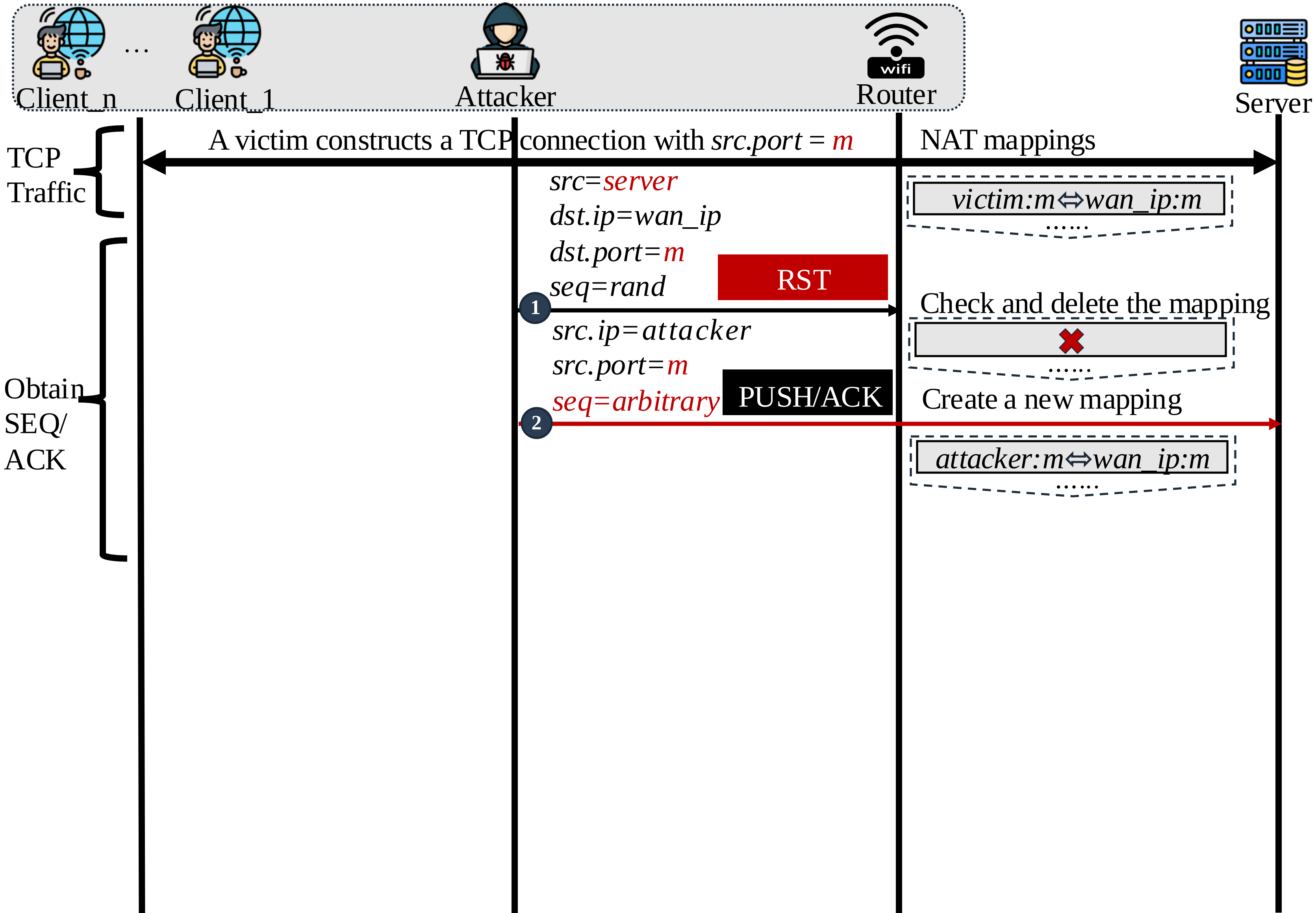
Hijacking Active Connections



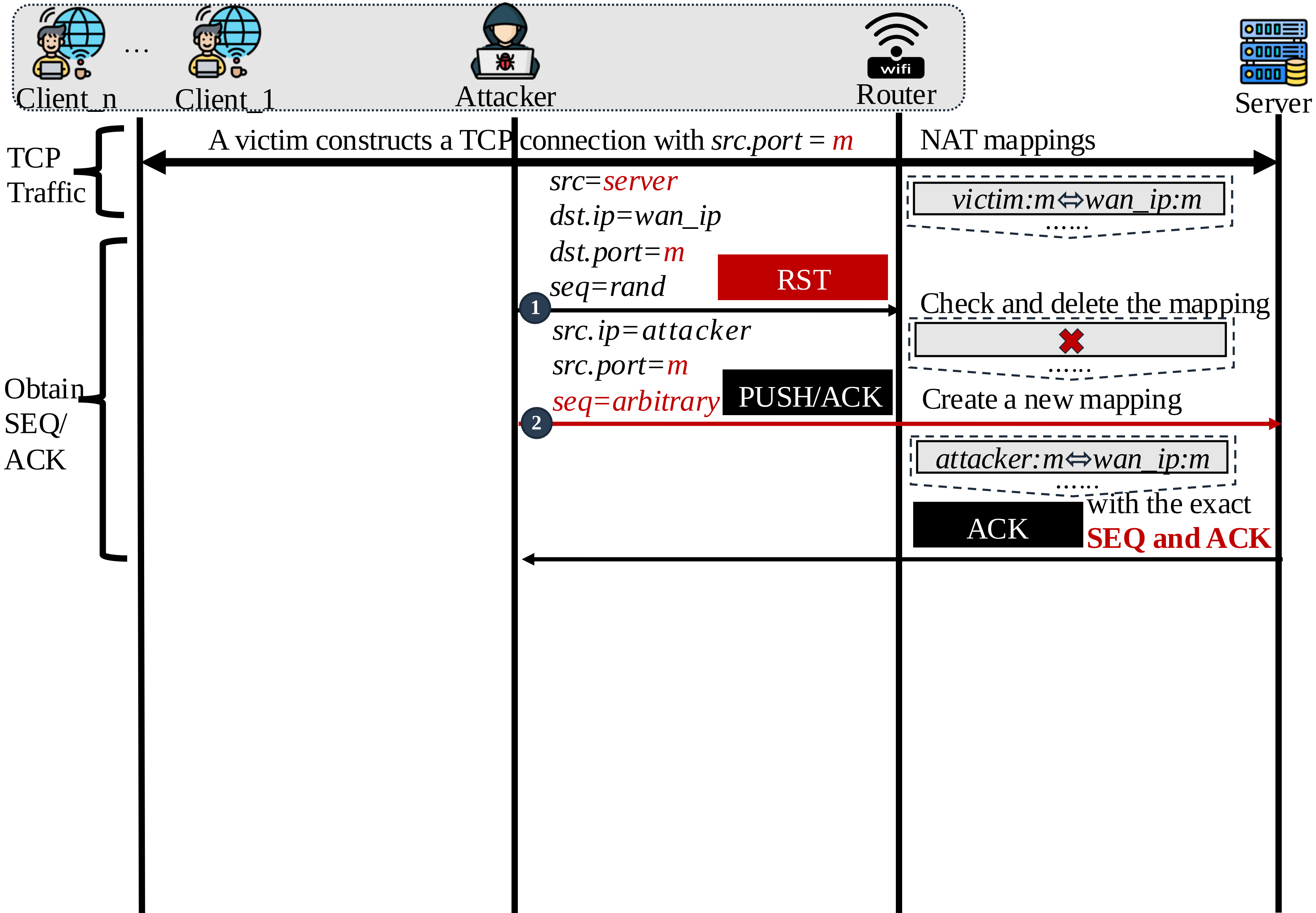
Hijacking Active Connections



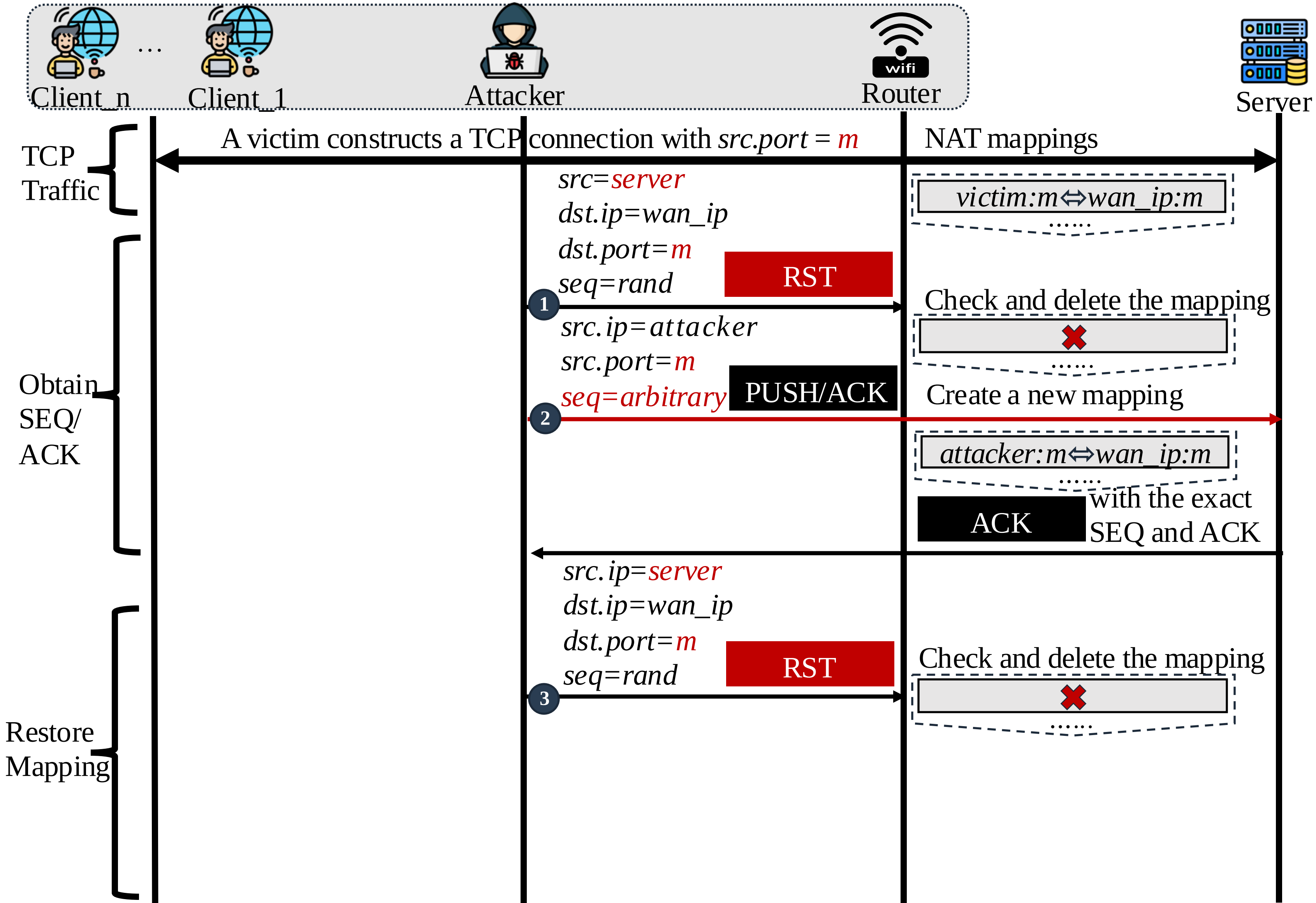
Hijacking Active Connections



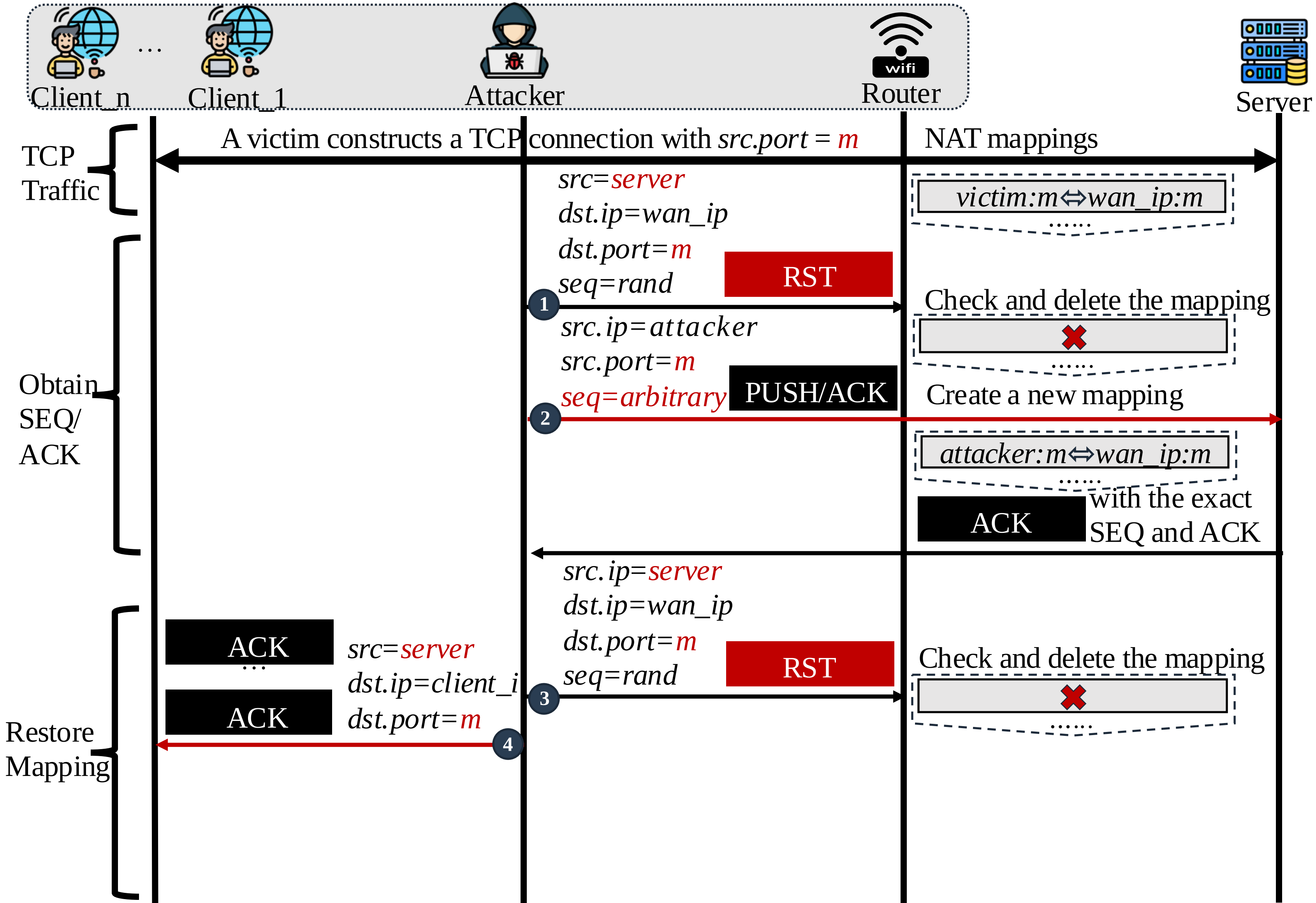
Hijacking Active Connections



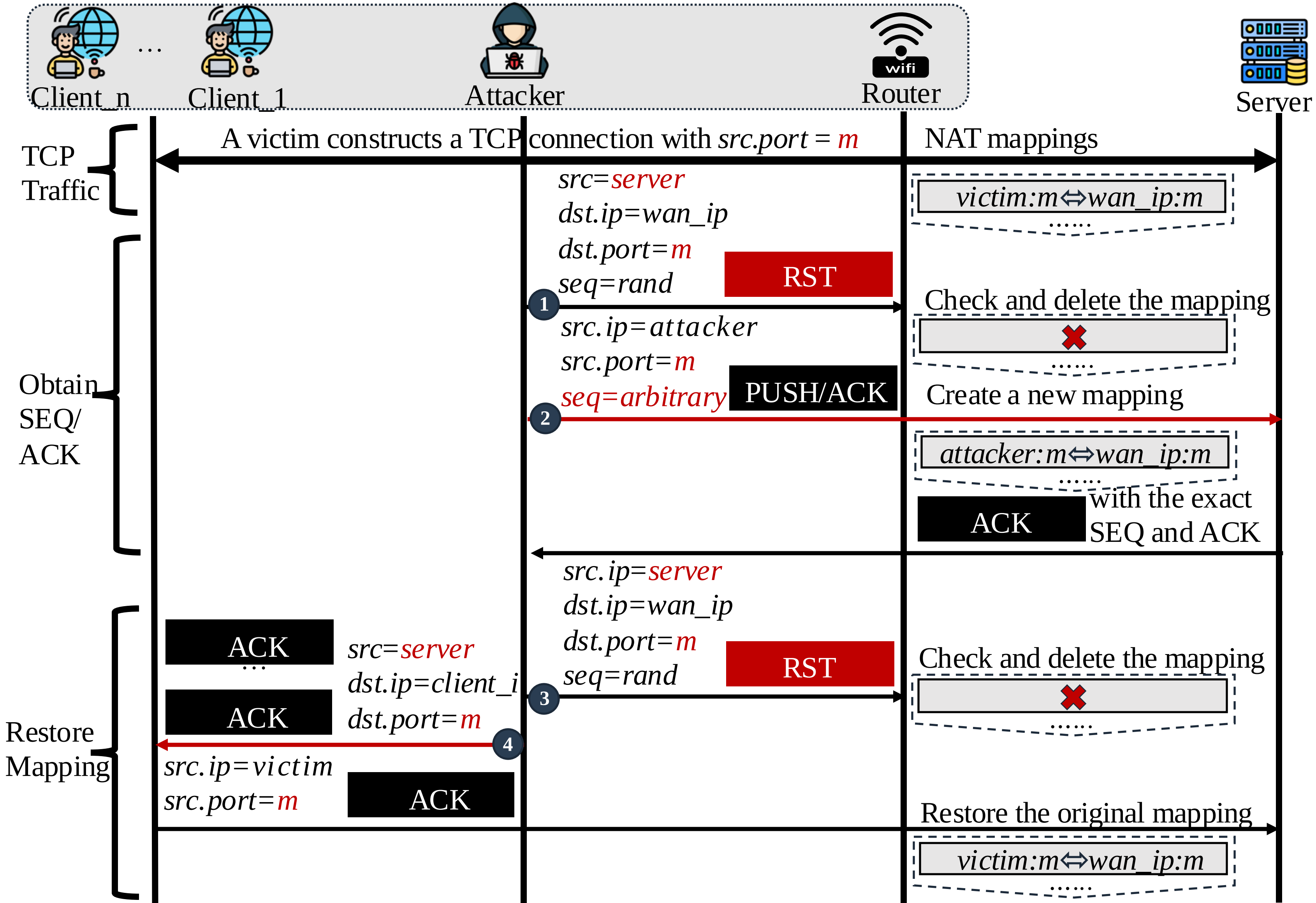
Hijacking Active Connections



Hijacking Active Connections



Hijacking Active Connections



Empirical Study

Analysis of Routers

✏ **We perform tests on 67 mainstream router models (acting as the gateway to provide Internet services) from 30 vendors.**

- 360, Aruba, ASUS, Amazon, Cisco Meraki
- China Mobile, Comfast, D-Link, GL.iNet
- Google, H3C, Huawei, IP-COM, iKuai
- JdCloud, Linksys, Mercury, Netgear, Netcore
- Ruijie, Skyworth, Tenda, TP-Link, Ubiquiti
- Volans, Wavlink, WiMaster, Xiaomi, and ZTE

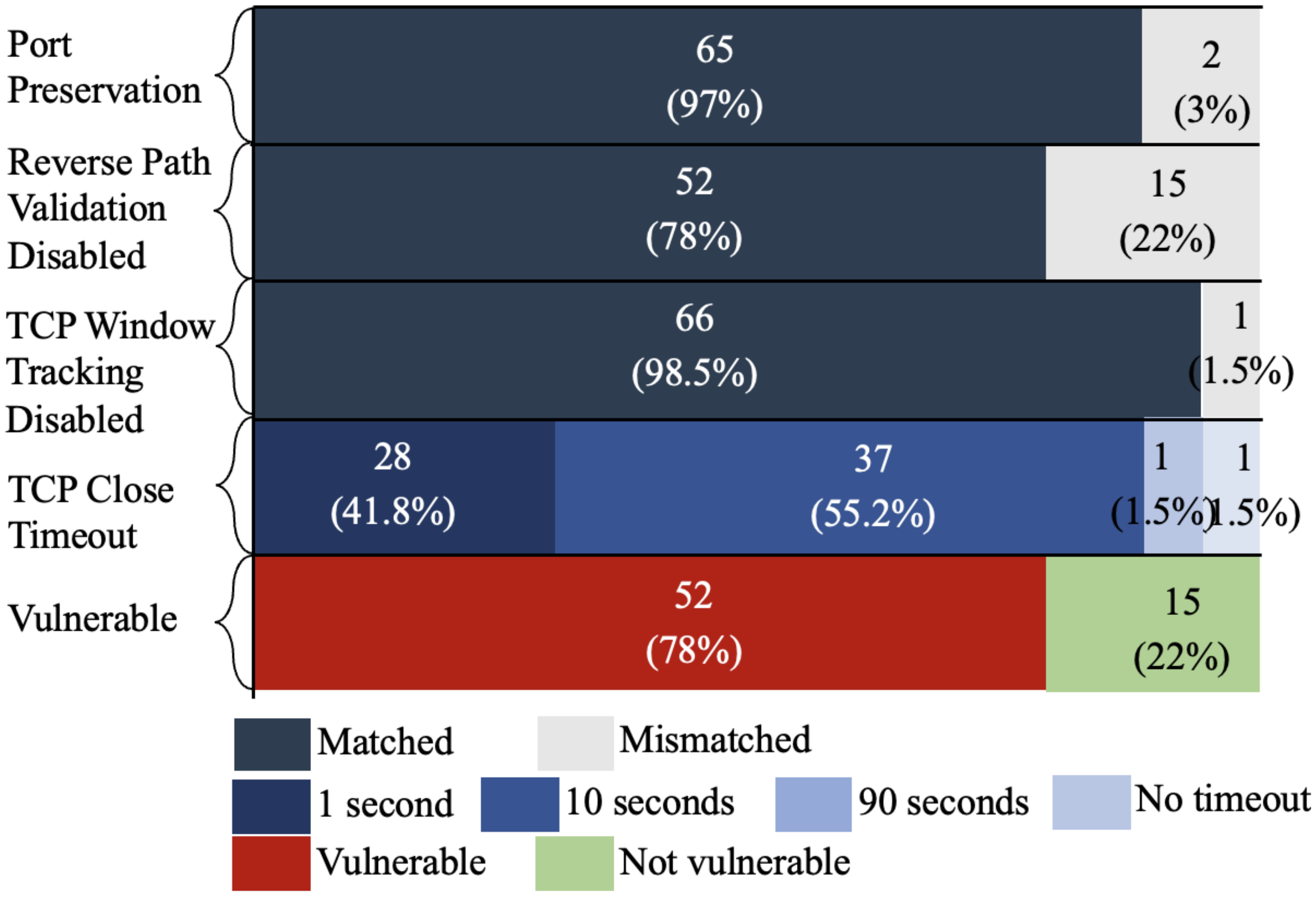
No.	Router Model	Vendor	OS	Generation	Port Preservation	Reverse-path Validation Disabled	TCP Window Tracking Disabled	TCP Close Timeout (second)	Vulnerable
1	TL-XDR6020	TP-Link	Linux-based	Wi-Fi 6	✔	✔	✔	1	✔
2	TL-WDR7620	TP-Link	Vxworks-based	Wi-Fi 5	✔	✘	✔	1	✘
3	AX3 Pro	Huawei	EMUI (Linux-based)	Wi-Fi 6	✔	✔	✔	10	✔
4	AR6140E-9G-2AC*	Huawei	VRP (Linux-based)	-	✘	✘	✔	10	✘
5	V6G	360	360OS(Linux-based)	Wi-Fi 6	✔	✔	✔	1	✔
6	Magic R365	H3C	Comware(Linux-based)	Wi-Fi 5	✔	✔	✔	10	✔
7	W30E	Tenda	Linux-based	Wi-Fi 6	✔	✔	✔	1	✔
8	RAX1800Z	China Mobile	AOS(Linux-based)	Wi-Fi 6	✔	✔	✔	10	✔
9	X32 Pro	Ruijie	RGOS(Linux-based)	Wi-Fi 6	✔	✔	✔	1	✔
10	Redmi RA81	Xiaomi	MiWiFi(Linux-based)	Wi-Fi 6	✔	✔	✔	1	✔
11	MW300R	Mercury	Vxworks-based	Wi-Fi 4	✔	✘	✔	1	✘
12	X30G	Mercury	Linux-based	Wi-Fi 6	✔	✔	✔	1	✔
13	RAX50	Netgear	DumaOS(Linux-based)	Wi-Fi 6	✔	✘	✔	10	✘
14	RT-AX89X	ASUS	AsusWrt(Linux-based)	Wi-Fi 6	✔	✘	✔	10	✘
15	E9450	Linksys	Linux-based	Wi-Fi 6	✔	✔	✔	10	✔
16	QUANTUM D2G	Wavlink	Linux-based	Wi-Fi 5	✔	✔	✔	10	✔
17	CF-616AC	Comfast	OrangeOS(Linux-based)	Wi-Fi 5	✔	✔	✔	10	✔
18	DI-7003GV2*	D-Link	Linux-based	-	✔	✔	✔	1	✔
19	AX3000	ZTE	ZXR10ROS(Linux-based)	Wi-Fi 6	✔	✘	✔	10	✘
20	M80*	IP-COM	Linux-based	-	✔	✔	✔	1	✔
21	SK-WR6640X	Skyworth	Linux-based	Wi-Fi 6	✔	✔	✔	10	✔
22	VE5200G*	Volans	Linux-based	-	✔	✔	✔	1	✔
23	NBR1009GPE	Netcore	NOS(Linux-based)	-	✔	✔	✔	1	✔
24	Wimaster*	Wimaster	Linux-based	-	✔	✔	✔	10	✔
25	IK-Enterprise*	iKuai	iKuaiOS(Linux-based)	-	✔	✔	✔	10	✔
26	Instant On AP22	Aruba	ArubaOS(Linux-based)	Wi-Fi 6	✔	✘	✔	10	✘
27	EdgeRouter X*	Ubiquiti	Linux-based	-	✔	✔	✔	10	✔
28	AX1800	JdCloud	Linux-based	Wi-Fi 6	✔	✔	✔	10	✔
29	Cisco Meraki 64*	Cisco Meraki	Linux-based	-	✔	✘	✘	-	✘
30	eero pro	Amazon	Linux-based	Wi-Fi 5	✔	✔	✔	10	✔
31	Google Wi-Fi	Google	ChromeOS(Linux-based)	Wi-Fi 5	✔	✔	✔	10	✔
32	GL-MT3000	GL.iNet	Linux-based	Wi-Fi 6	✔	✔	✔	10	✔
33	pfSense 2.7.0*	pfSense	FreeBSD-based	-	✘	✘	✔	90	✘

✔means that the router is satisfied with the condition, and ✘means that the router is dissatisfied with the condition.
✔means that the router is vulnerable to our attack, and ✘means that the router is immune to our attack.
* means that the model is an enterprise router which does not support Wi-Fi by itself and needs to work together with wireless access points.

Test Results of the Router Models.

Analysis of Routers

- **52 of the 67 tested routers are vulnerable.**
- **15 models are immune to the attack as they do not fulfill all of the conditions.**
- **The vulnerable routers are from 24 of the 30 vendors.**



Attack Evaluation

- ✎ We conduct thorough experiments of the attack in 93 various Wi-Fi networks.
- ✎ We take three case studies of attacks on SSH, FTP, and HTTP applications and measuring the time cost and success rate of each attack.

No.	Network Mode	SSID	Router Vendor	Wi-Fi Generation	WPA2/3 Enterprise/Personal	Attack Result	Time Cost (s)	Success Rate
1	Enterprise mode	Campus 1	Huawei	Wi-Fi 6	WPA2-Enterprise	SSH DoS	15.43	18/20
2	Enterprise mode	Campus 2	TP-Link	Wi-Fi 4	WPA2-Enterprise	FTP Hijacking	10.32	18/20
3	Enterprise mode	Campus 3	H3C	Wi-Fi 6	WPA2-Enterprise	HTTP Injection	48.87	15/20
4	Enterprise mode	Enterprise 1	TP-Link	Wi-Fi 6	WPA2-Enterprise	SSH DoS	11.56	16/20
5	Enterprise mode	Enterprise 2	TP-Link	Wi-Fi 5	WPA2-Enterprise	FTP Hijacking	11.43	18/20
6	Enterprise mode	Enterprise 3	Netcore	Wi-Fi 6	WPA2-Enterprise	HTTP Injection	87.20	15/20
7	Enterprise mode	Office building 1	TP-Link	Wi-Fi 5	WPA2-Enterprise	SSH DoS	9.56	18/20
8	Enterprise mode	Office building 2	iKuai	Wi-Fi 6	WPA2-Enterprise	FTP Hijacking	21.46	17/20
9	Enterprise mode	Office building 3	Mercury	Wi-Fi 6	WPA2-Enterprise	HTTP Injection	31.14	15/20
10	Enterprise mode	Hotel 1	Netcore	Wi-Fi 5	WPA2-Enterprise	SSH DoS	15.75	18/20
11	Enterprise mode	Hotel 2	D-Link	Wi-Fi 6	WPA2-Enterprise	FTP Hijacking	9.45	19/20
12	Enterprise mode	Hotel 2	iKuai	Wi-Fi 6	WPA2-Enterprise	HTTP Injection	71.32	16/20
13	Home mode	Restaurant 1	TP-Link	Wi-Fi 5	WPA2-Personal	SSH DoS	8.95	17/20
14	Home mode	Restaurant 2	Comfast	Wi-Fi 5	WPA2-Personal	FTP Hijacking	21.56	18/20
15	Home mode	Restaurant 3	Skyworth	Wi-Fi 6	WPA2-Personal	HTTP Injection	62.35	13/20
16	Home mode	Coffee shop 1	Mercury	Wi-Fi 4	WPA2-Personal	SSH DoS	8.98	17/20
17	Home mode	Coffee shop 2	TP-Link	Wi-Fi 4	WPA2-Personal	FTP Hijacking	9.29	18/20
18	Home mode	Coffee shop 3	Wavlink	Wi-Fi 5	WPA2-Personal	HTTP Injection	45.22	13/20
19	Home mode	Shopping mall 1	Tenda	Wi-Fi 6	WPA3-Personal	SSH DoS	24.23	18/20
20	Home mode	Shopping mall 2	TP-Link	Wi-Fi 4	WPA2-Personal	FTP Hijacking	11.44	19/20
21	Home mode	Shopping mall 3	Huawei	Wi-Fi 6	WPA3-Personal	HTTP Injection	78.44	15/20
22	Home mode	Bookstore 1	360	Wi-Fi 5	WPA2-Personal	SSH DoS	19.45	18/20
23	Home mode	Bookstore 2	Xiaomi	Wi-Fi 6	WPA3-Personal	FTP Hijacking	10.61	18/20
24	Home mode	Bookstore 3	H3C	Wi-Fi 6	WPA3-Personal	HTTP Injection	56.12	14/20
25	Home mode	Experience store 1	Xiaomi	Wi-Fi 6	WPA3-Personal	SSH DoS	16.97	17/20
26	Home mode	Experience store 2	Huawei	Wi-Fi 6	WPA3-Personal	FTP Hijacking	23.98	18/20
27	Home mode	Experience store 3	Xiaomi	Wi-Fi 5	WPA2-Personal	HTTP Injection	52.14	16/20
28	Home mode	Cinema 1	Ruijie	Wi-Fi 5	WPA2-Personal	SSH DoS	8.89	19/20
29	Home mode	Cinema 2	Mercury	Wi-Fi 6	WPA3-Personal	FTP Hijacking	11.31	18/20
30	Home mode	Cinema 2	Huawei	Wi-Fi 6	WPA3-Personal	HTTP Injection	54.26	16/20

Experimental Results of TCP Attacks in the Wi-Fi Networks.

Attack Evaluation

SSH DoS Attack:

- Success rate: 87.4%
- Attack time: ~17.5s

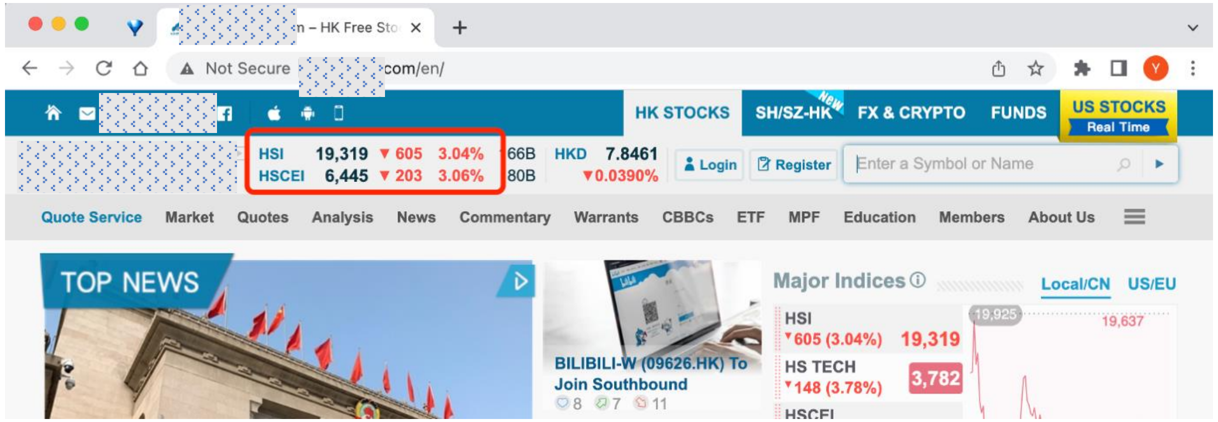
FTP Hijacking Attack:

- Success rate: 82.6%
- Attack time: ~19.4s

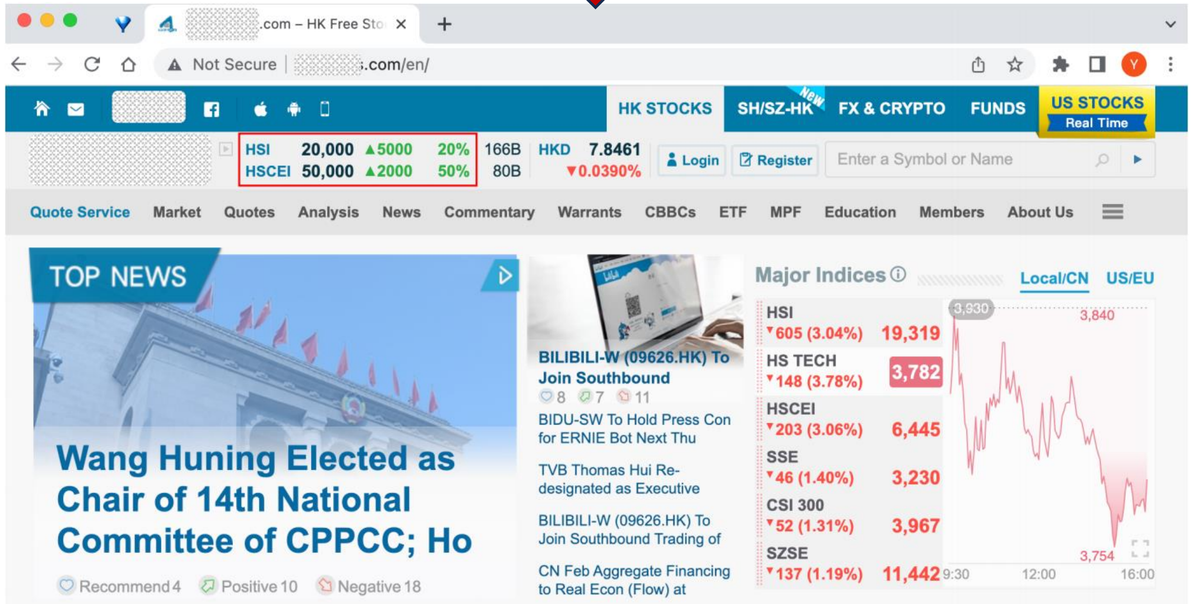
HTTP Injection Attack:

- Success rate: 76.1%
- Attack time: ~54.5s

Attack Type	Inferring Port(s)	Getting SEQ/ACK(s)	Finishing Attacking(s)	Total Time(s)	BW (pkts)	Success Rate
SSH DoS	8.1	8.4	1.0	17.5	4000	87.4%
FTP Hijacking	9.1	9.2	1.1	19.4	4000	82.6%
HTTP Injection	9.4	15.2	29.9	54.5	4000	76.1%



After attack



Disclosure and Mitigation

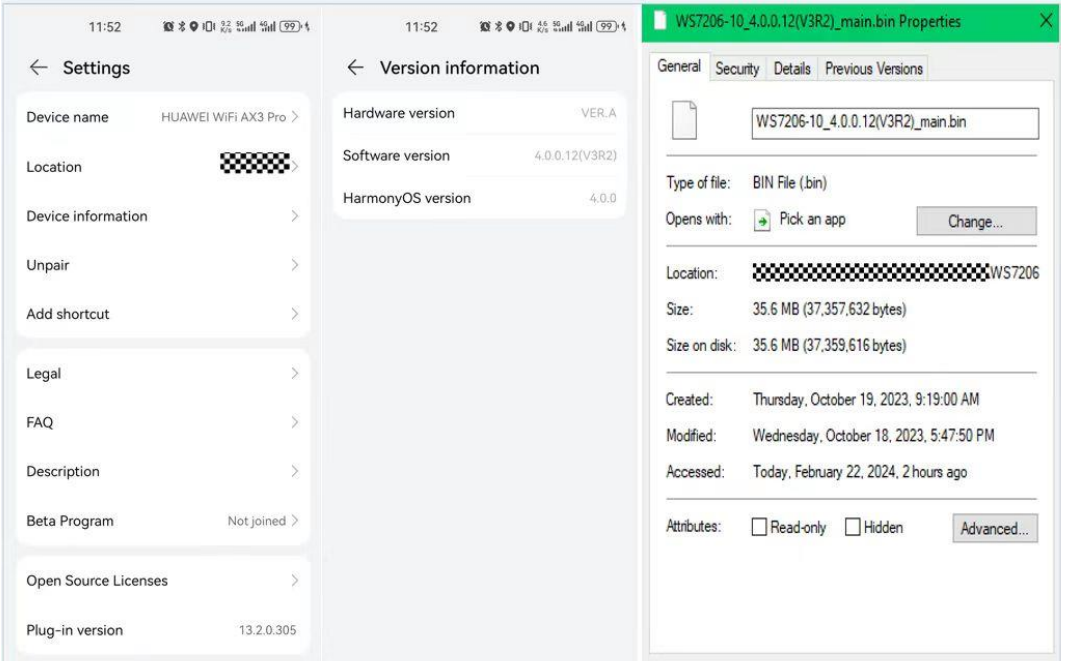
Disclosure and Mitigation

Ethical disclosure:

- Acknowledgment from the **OpenWrt** community and **7 router vendors** (i.e., TP-Link, Huawei, Xiaomi, 360, Mercury, Ubiquiti, and Linksys)
- Some vendors have released patches to fix this vulnerability, e.g., **OpenWrt, Huawei …**
- **10 CVEs** (from CVE-2023-30305 to CVE-2023-30314)

Mitigation :

- Take random port allocation method
- Enable reverse path validation
- Enable TCP window tracking



The Patch from Huawei Developers

Conclusion

✐ Uncovered a new NAT vulnerability in Wi-Fi networks to attack TCP connections.

✐ Performed large-scale measurements of routers and experiments in real networks.

✐ Suggested defense countermeasures and some of them have been adopted.

Questions?