TCP Hijacking in NAT-enabled Networks

Capstone Project Presentation

Suraj Sharma (1120231904)

Supervisor: Prof. Mahavir Jhawar

Department of Computer Science
Ashoka University

December 19, 2025

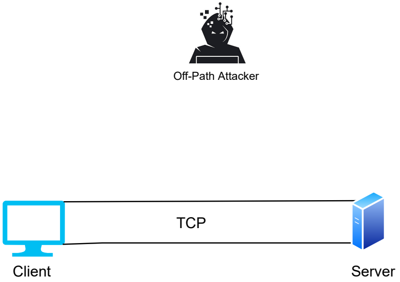
Problem Statement: TCP hijacking

Off-path TCP hijacking is a type of network attack in which an attacker can:

- Terminate a TCP connection between endpoints.
- Inject malicious data into the traffic.
- Reroute data from the legitimate endpoint to itself.

Recent work in the TCP hijacking:

- ReDAN: An Empirical Study on Remote DoS Attacks against NAT Networks https://www.ndss-symposium.org/wp-content/uploads/2025-972-paper.pdf
- Off-Path TCP Hijacking Attack to NAT-Enabled Wi-Fi Networks https://ieeexplore.ieee.org/document/11076087
- Off-Path TCP Hijacking in Wi-Fi Networks: A Packet-Size Side Channel Attack https://www.ndss-symposium.org/wp-content/uploads/2025-305-paper.pdf



Problem Statement

To implement and reproduce the recently published **remote off-path DDoS attack against clients behind NAT networks** communicating over TCP with an external server described by [FYL$^+$25], which was presented at the Network and Distributed System Security (NDSS) Symposium 2024.

ReDAN: An Empirical Study on Remote DoS Attacks against NAT Networks

Xuewei Feng*, Yuxiang Yang*, Qi Li*†, Xingxiang Zhan†, Kun Sun‡, Ziqiang Wang§, Ao Wang§, Ganqiu Du¶, Ke Xu*†✉
*Tsinghua University, †Zhongguancun Laboratory, ‡CSIS, George Mason University
§Southeast University, ¶China Software Testing Center
fengxw06@126.com, yangyx22@mails.tsinghua.edu.cn, qli01@tsinghua.edu.cn, zhansingsong@gmail.com, ksun3@gmu.edu, {ziqiangwang, wangao}@seu.edu.cn, duganqiu@cstc.org.cn, xuke@tsinghua.edu.cn

1

***Abstract*—In this paper, we conduct an empirical study on remote DoS attacks targeting NAT networks (ReDAN, short for Remote DoS Attacks targeting NAT). We show that Internet attackers operating outside local NAT networks possess the capability to remotely identify a NAT device and subsequently terminate TCP connections initiated from the identified NAT device to external servers. Our attack involves two steps. First, we identify NAT devices on the Internet by exploiting inadequacies in the Path MTU Discovery (PMTUD) mechanism within NAT specifications. This deficiency creates a fundamental side channel that allows Internet attackers to distinguish if a public IPv4 address serves a NAT device or a separate IP host, aiding in the identification of target NAT devices. Second, we launch a remote DoS attack to terminate TCP connections on the identified NAT devices. While recent NAT implementations may include protective measures, such as packet legitimacy validation to prevent malicious manipulations on NAT mappings, we discover that these safeguards are not widely adopted in real world. Consequently, attackers can send crafted packets to deceive NAT devices into erroneously removing innocent TCP connection mappings, thereby disrupting the NATed clients to access remote TCP servers. Our experimental results reveal widespread security vulnerabilities in existing NAT devices. After testing 8 types of router firmware and 30 commercial NAT devices from 14 vendors, we identify vulnerabilities in 6 firmware types and 29 NAT devices that allow off-path removal of TCP connection mappings. Moreover, our measurements reveal a stark reality: 166 out of 180 (over 92%) tested real-world NAT networks, comprising 90 4G LTE/5G networks, 60 public Wi-Fi networks, and 30 cloud VPS networks, are susceptible to exploitation. We responsibly disclosed the vulnerabilities to affected vendors and received a significant number of acknowledgments. Finally, we propose our countermeasures against the identified DoS attack.**

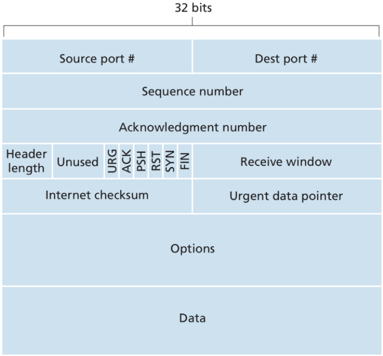
VPS networks, public Wi-Fi networks, and IoT networks, to condense multiple local private addresses into a public one. According to CAIDA's investigations, more than 23% Autonomous Systems (ASes) use NAT to conserve public IPv4 addresses and the proportion keeps increasing [24]. Moreover, it is widely believed that NAT offers enhanced security [42], [33], [35], [2], since NAT serves as an added security measure for private networks by concealing the actual IP addresses of internal hosts. This prevents direct exposure of the internal hosts to Internet attackers.

In this paper, we undertake a comprehensive empirical study to demonstrate that real-world NAT implementations may exhibit vulnerabilities, which can be exploited by off-path attackers on the Internet to pose a substantial threat to end-to-end communication connectivity. Particularly, by exploiting these vulnerabilities in various NAT devices (e.g., NAT gateways in public Wi-Fi networks or PDN gateways/UPF devices in 4G LTE/5G networks), we demonstrate that off-path attackers operating outside local NAT networks can launch remote DoS attacks against the NAT network (i.e., the network segment linked to the Internet through the NAT device) to cut off TCP connections initiated by the NATed clients to an external server. This identified DoS attack can occur even when the internal NATed clients have a robust TCP/IP implementation and are free from DoS vulnerabilities. Our attack consists of two main steps, namely, i) identifying NAT devices on the Internet and ii) remotely severing TCP connections on the NAT devices.

[1] https://www.ndss-symposium.org/wp-content/uploads/2025-972-paper.pdf

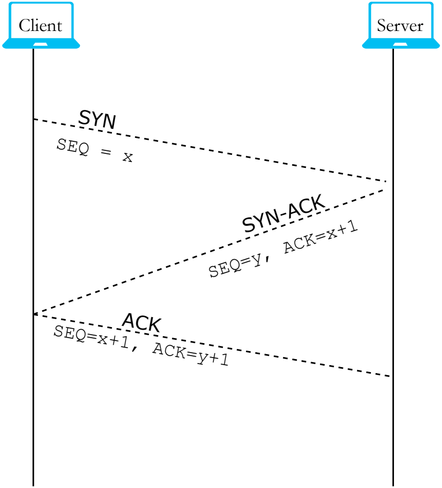
Outline

1. Introduction
2. Background
3. Threat Model
4. Experimental Setup
5. Experiments & Result
6. Discussion & Conclusion
7. Future Work



- TCP is a Layer 4 (transport layer) protocol.
- Provides full-duplex, reliable, and in-order data exchange between end hosts.
- Used for process-to-process communication.

TCP: Three-Way Handshake



- A three-way handshake is the process of establishing a TCP connection.
- It occurs before the endpoints can exchange data.

TCP: Connection Closing

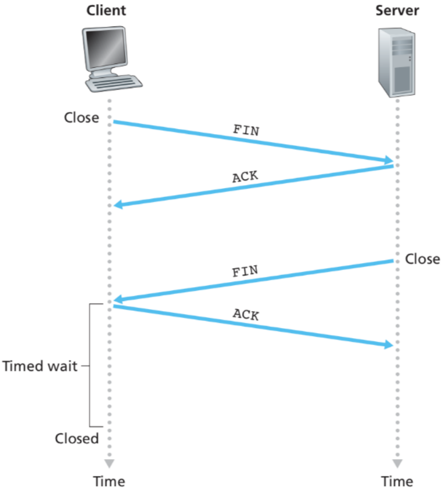


Figure: Graceful Connection Closing

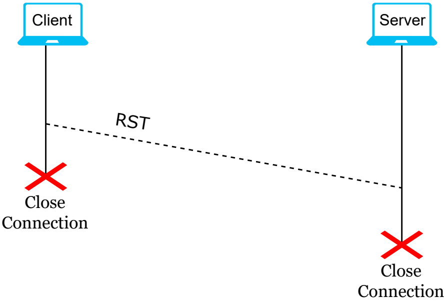
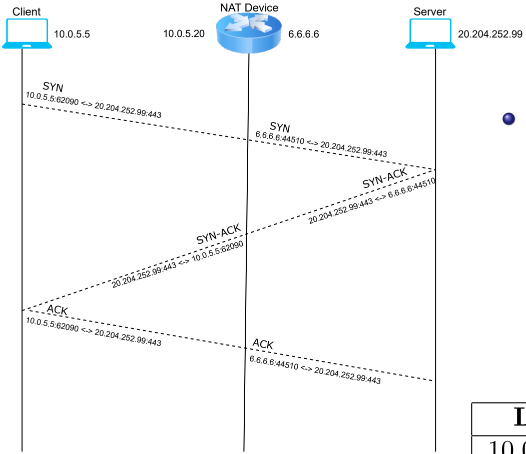


Figure: Abrupt Connection Closing

NAT (Network Address Translation)



- A technique that allows multiple clients with private IPs to use one or a limited set of public IPs to access the Internet.
- Two important features:
  - It was introduced to reduce the exhaustion of IPv4 addresses.
  - Perceived as increasing security by hiding the internal topology and blocking unsolicited inbound connection requests.

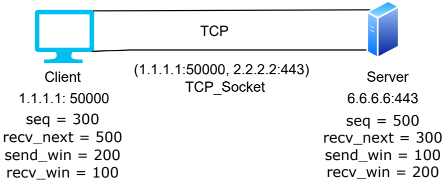
LAN Side	**WAN Side**
10.0.5.5 : 62090	6.6.6.6 : 44510

Outline

1. Introduction
2. Background
3. Threat Model
4. Experimental Setup
5. Experiments & Result
6. Discussion & Conclusion
7. Future Work

Background: TCP

- A TCP connection is identified by a 4-tuple: [`Source IP, Source Port, Destination IP, Destination Port`].
- This tuple allows hosts and intermediate devices, including NATs and firewalls, to demultiplex concurrent TCP flows.
- TCP uses a 32-bit sequence number space, where each transmitted byte is labeled with a sequence number and acknowledged cumulatively upon successful receipt.
- An attacker must guess a valid 4-tuple and a valid in-window sequence number to hijack a TCP connection.



Background: TCP Hijacking Attacks

- Earlier attacks exploited predictable Initial Sequence Number (ISN) generation methods, such as timing-based or global counters, making it easier to guess valid in-window sequence numbers.
- Common techniques:
  - RST injection
  - SYN injection

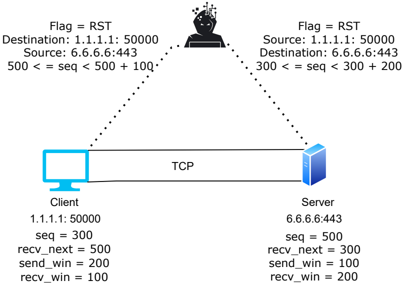
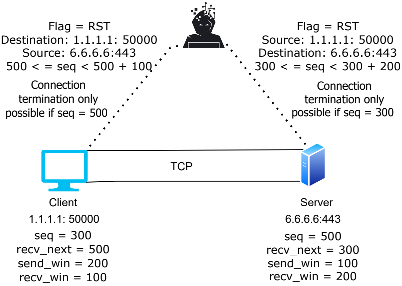


Figure: Classic RST Injection Attack

Background: TCP Hijacking Attack

- Modern systems use cryptographically secure ISN generation and ephemeral source port randomization.
- RFC 5961 [RSD10] mandates that a RST packet must match the exact next expected sequence number to close a connection. In-window RSTs result in a challenge ACK instead.
- The high entropy of connection identifiers ($\approx 2^{32+16}$) makes brute-force attacks impractical without auxiliary side channels.

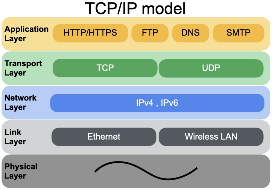


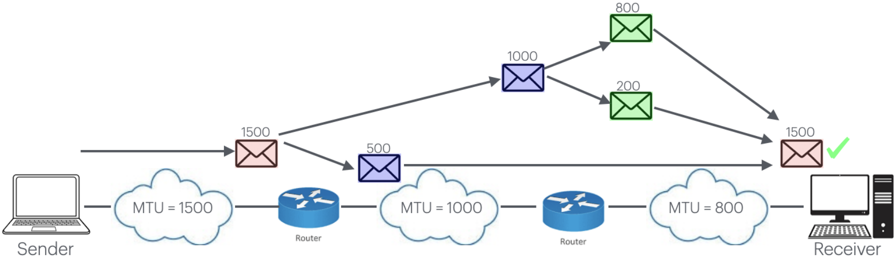
Background: NAT Working

- NATs rewrite source or destination IP addresses and port numbers and maintain per-flow state in a translation table.
- NATs inspect TCP control flags like `SYN`, `ACK`, `FIN`, and `RST` to infer connection state.
- Real-world implementations often skip sequence number validation for inbound RST packets due to performance considerations.
- An attacker who can guess a valid 4-tuple can send a spoofed RST packet to prematurely terminate a TCP connection.

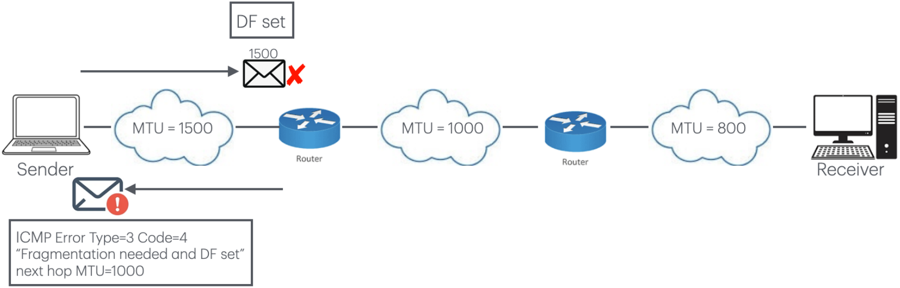
Path MTU discovery (PMTUD)

PMTUD is designed to prevents IP fragmentation by dynamically determining the maximum packet size supported along a network path.

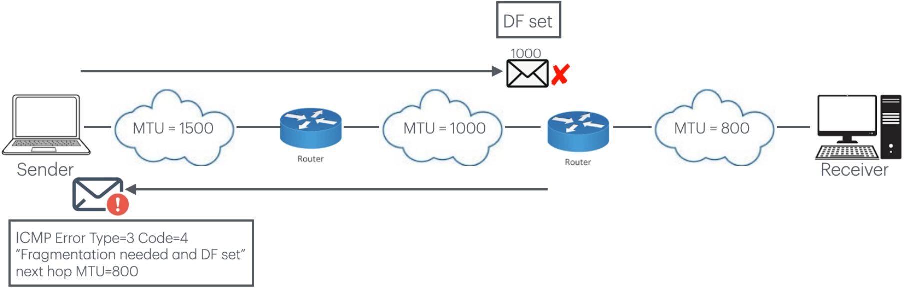




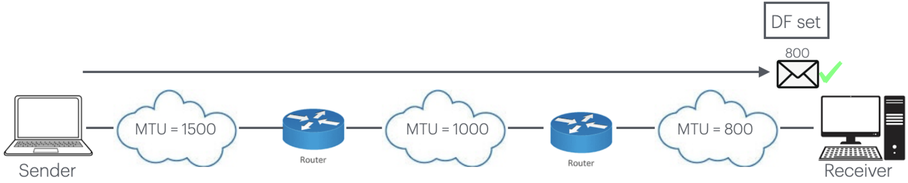
Path MTU discovery (PMTUD)



Path MTU discovery (PMTUD)



Path MTU discovery (PMTUD)



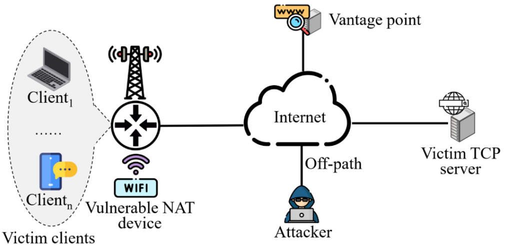
Outline

1. Introduction
2. Background
3. Threat Model
4. Experimental Setup
5. Experiments & Result
6. Discussion & Conclusion
7. Future Work

Attack overview

An off-path attacker can remotely identify a NAT device and terminate TCP connections initiated to a server from that device

1. Identifying whether a client is behind the NAT
2. If in NAT, performing a remote DDoS attack to terminate a TCP connunication between a client and remote server.



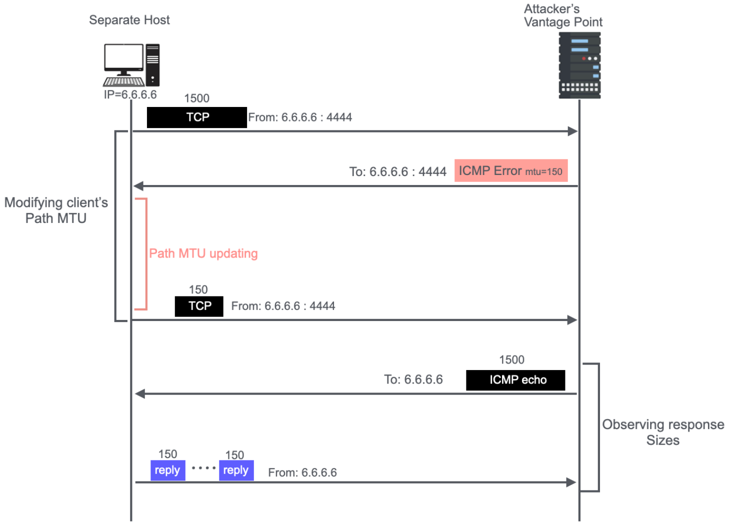
[2]

[2] https://www.ndss-symposium.org/wp-content/uploads/7A-s0972-Yang.pdf

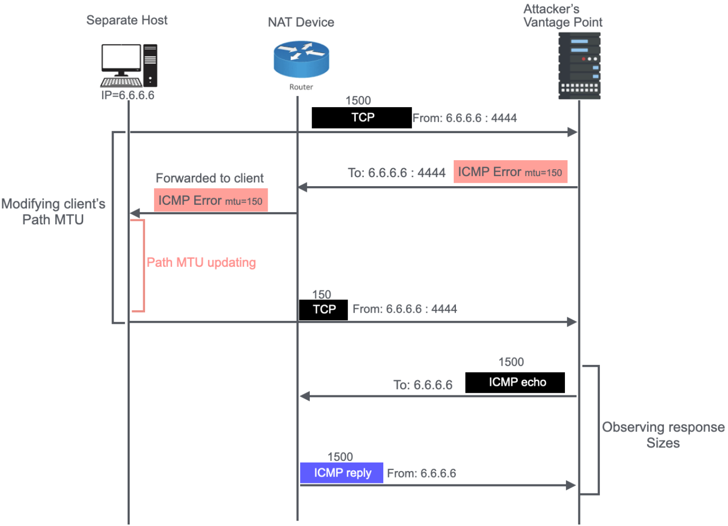
Identifying NAT Device

- Determine if a target host is a saperate host on the internet or a host behind the NAT by exploiting a side channel vulnerability in real world NAT implementation.
- Two step process:
    1. Changing Client's Path MTU
    2. Oberseving the size of subsquent TCP packet to the vantage point

Identifying NAT device: Client is a saperate host



Identifying NAT device: Client is a behind the NAT

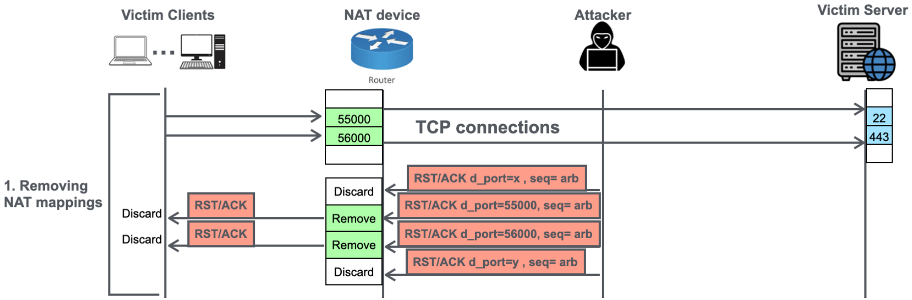


Terminating a TCP connection

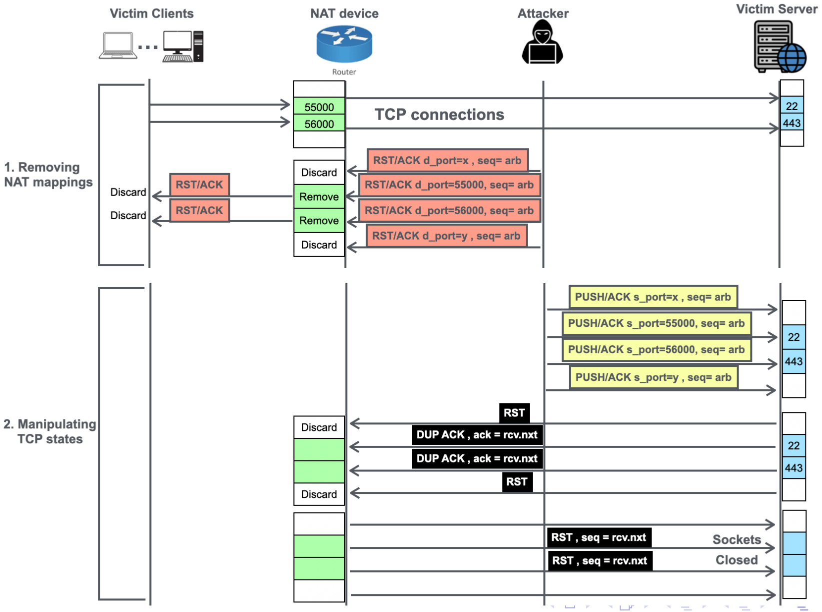
After identifying if a host is behind the NAT, the attacker perform the attack in two stage to terminate the TCP connection.

1. Removing NAT mappig
2. Manipulating TCP state
3. Terminating TCP connection

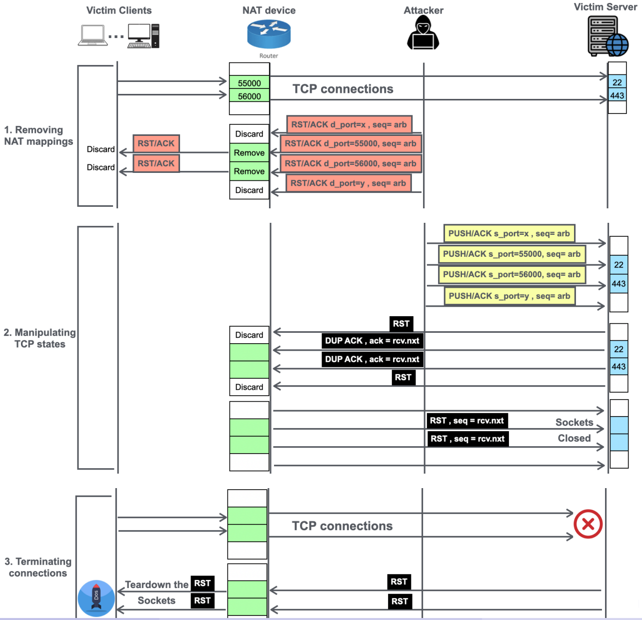
Performing the DDoS attack



Performing the DDoS attack



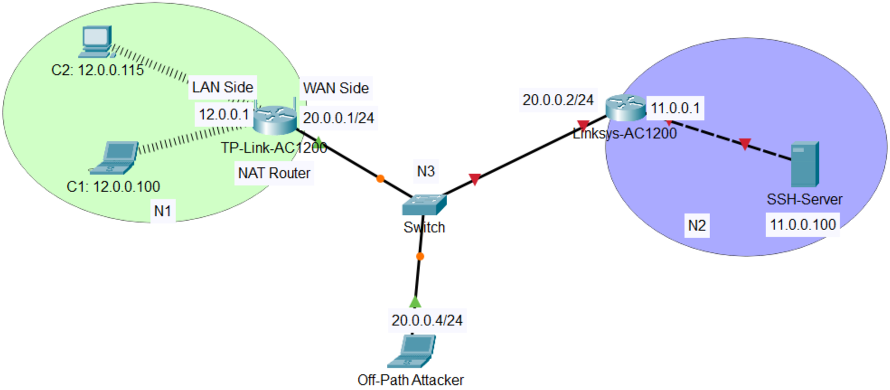
Performing the DDoS attack



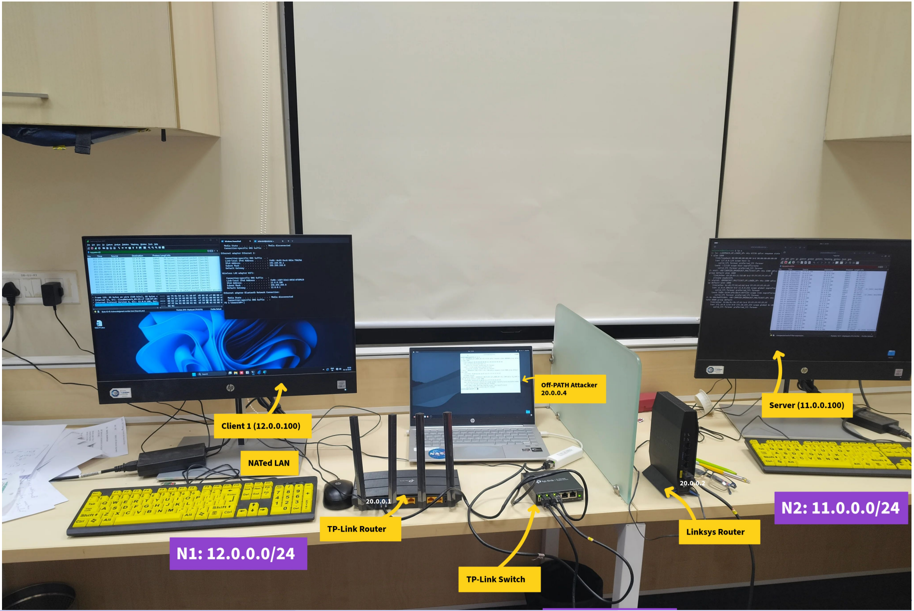
Outline

1. Introduction
2. Background
3. Threat Model
4. Experimental Setup
5. Experiments & Result
6. Discussion & Conclusion
7. Future Work

Experimental Setup: Logical View

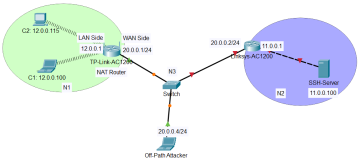


Experimental Setup: Physical Setup



Outline

1. Introduction
2. Background
3. Threat Model
4. Experimental Setup
5. Experiments & Result
6. Discussion & Conclusion
7. Future Work

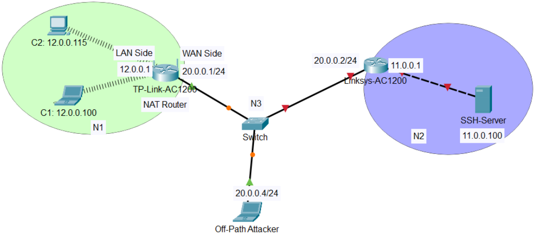


1. SSH server at `11.0.0.100`
2. C1 (12.0.0.100) and C2 (12.0.0.115) established TCP connections to the server on ports 57605 and 45510, respectively.

LAN Side	**WAN Side**
12.0.0.100:57605	20.0.0.1:57605
12.0.0.115:45510	20.0.0.1:45510

Table: NAT Mapping Table

Experiments: Server Side



From the server's perspective:

Socket #	**Source**	**Destination**
S1	20.0.0.1:57605	11.0.0.100:22
S2	20.0.0.1:45510	11.0.0.100:22

Table: Sockets at Server Side for the TCP connections

The off-path attacker performs the attack in two steps:

1. Impersonate the server and send spoofed RST-ACK packets to the public IP of the NAT device.
2. Impersonate the NAT device and send spoofed PSH-ACK packets to the server.

Experiments: Attack Stage 1

The attacker sends crafted RST-ACK packets to the public IP of the NAT device.

Source	Server (11.0.0.100)
Destination	NAT device (20.0.0.1)
Source Port	22
Dest. Port	32768–65535
SEQ	Arbitrary
ACK	Arbitrary
Flag	`RST-ACK`

Experiments: Clearing NAT Mapping

Initial NAT mappings were:

LAN Side	**WAN Side**	Destination	Status
12.0.0.100:57605	20.0.0.1:57605	11.0.0.100:22	Established
12.0.0.115:45510	20.0.0.1:45510	11.0.0.100:22	Established

Once the NAT router receives the RST packets with destination ports 57605 and 45510, it sets the status to `Closing`.

LAN Side	**WAN Side**	Destination	Status
12.0.0.100:57605	20.0.0.1:57605	11.0.0.100:22	Closing
12.0.0.115:45510	20.0.0.1:45510	11.0.0.100:22	Closing

After the timeout period:

LAN Side	**WAN Side**	Destination	Status

Experiments: Attack Stage 2

The attacker sends crafted PSH-ACK packets to the server.

Source	NAT device (20.0.0.1)
Destination	Server (11.0.0.100)
Source Port	32768–65535
Dest. Port	22
SEQ	Arbitrary
ACK	Arbitrary
Flag	`PSH-ACK`

Experiments: Step 2 — Manipulating TCP State

- The server receives the PSH-ACK packets:

Socket #	Source	Destination
S1	20.0.0.1:57605	11.0.0.100:22
S2	20.0.0.1:45510	11.0.0.100:22

Table: Sockets at Server Side for the TCP connections

- For ports 57605 and 45510, it sends duplicate ACKs (DUP-ACKs) to the NAT device [APB09].
- The NAT device, having no mapping, should typically discard the packet and issue an RST. In our case, it does not issue an RST and silently discards the packet (RFC9293).
- After the mapping is cleared, any attempt to send other data packets from the client does not pass through the NAT router. This suggests that as soon as the NAT router receives the crafted TCP RST packet, it removes the NAT mapping after a timeout period. After that, any attempt to reestablish a new connection on the given ports fails.

Results

- The experiment was repeated across various client and server operating systems—namely Ubuntu 22.04, Ubuntu 24.04, Windows 10/11, and Android 16—and application protocols, namely SSH, HTTP, and FTP. In all cases, the NAT device consistently failed to validate TCP window parameters for incoming RST-ACK packets.
- After Stage 1 of the attack, when crafted RST-ACK packets were sent to the NAT, the corresponding NAT mappings were immediately removed. Consequently, all active TCP connections relying on those mappings were torn down. For example, HTTP clients had to initiate a new TCP connection using a different port.
- This suggests that NAT devices unconditionally remove the NAT mapping upon receiving an inbound RST packet without checking its legitimacy.

Outline

1 Introduction

2 Background

3 Threat Model

4 Experimental Setup

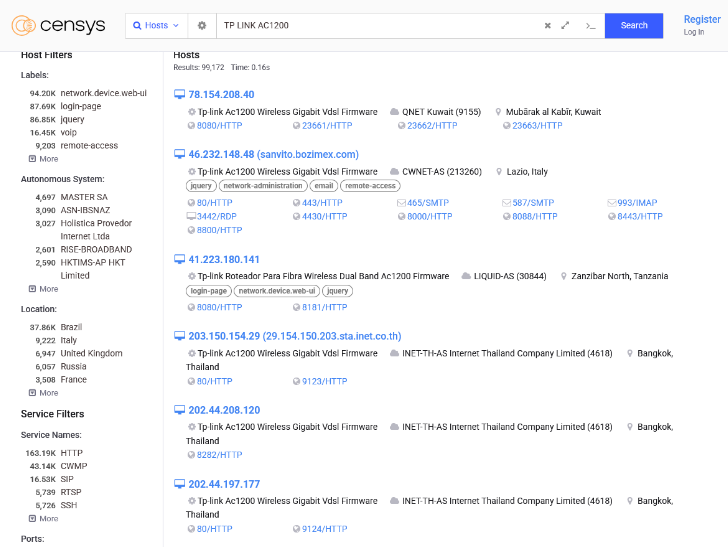
5 Experiments & Result

6 Discussion & Conclusion

7 Future Work

Discussion/Conclusion

- In this project, we explored the feasibility and execution of a remote off-path denial-of-service (DoS) attack against clients behind NAT devices by replicating the methodology proposed in the ReDAN study (feng et al.). By exploiting the vulnerability in NAT devices—that they do not perform TCP window validation for inbound RST packets—it was shown that an off-path attacker can terminate active TCP connections between internal clients and external servers without being directly on the communication path.
- The attack relies on two key observations.
  - NAT devices often fail to validate whether an inbound TCP RST segment falls within the acceptable receive window, a behavior that violates recommendations such as those in RFC 5961.
  - ICMP-based side channels, exploiting the vulnerability of NAT devices that do not correctly implement the Path MTU Discovery mechanism, allow remote adversaries to infer the presence of NAT.
- By exploiting these vulnerabilities in combination, an attacker can silently and selectively interfere with ongoing TCP connections.


censys
Hosts
TP LINK AC1200
Search
Register
Log In
Host Filters
Labels:
94.20K network.device.web-ui
87.69K login-page
86.85K jquery
16.45K voip
9,203 remote-access
More
Autonomous System:
4,697 MASTER SA
3,090 ASN-IBSNAZ
3,027 Holistica Provedor Internet Ltda
2,601 RISE-BROADBAND
2,590 HKTIMS-AP HKT Limited
More
Location:
37.86K Brazil
9,222 Italy
6,947 United Kingdom
6,057 Russia
3,508 France
More
Service Filters
Service Names:
163.19K HTTP
43.14K CWMP
16.53K SIP
5,739 RTSP
5,726 SSH
More
Ports:
Hosts
Results: 99,172 Time: 0.16s
78.154.208.40
Tp-link Ac1200 Wireless Gigabit Vdsl Firmware
QNET Kuwait (9155)
Mubārak al Kabīr, Kuwait
8080/HTTP
23661/HTTP
23662/HTTP
23663/HTTP
46.232.148.48 (sanvito.bozimex.com)
Tp-link Ac1200 Wireless Gigabit Vdsl Firmware
CWNET-AS (213260)
Lazio, Italy
jquery
network-administration
email
remote-access
80/HTTP
443/HTTP
465/SMTP
587/SMTP
993/IMAP
3442/RDP
4430/HTTP
8000/HTTP
8088/HTTP
8443/HTTP
8800/HTTP
41.223.180.141
Tp-link Roteador Para Fibra Wireless Dual Band Ac1200 Firmware
LIQUID-AS (30844)
Zanzibar North, Tanzania
login-page
network.device.web-ui
jquery
8080/HTTP
8181/HTTP
203.150.154.29 (29.154.150.203.sta.inet.co.th)
Tp-link Ac1200 Wireless Gigabit Vdsl Firmware
INET-TH-AS Internet Thailand Company Limited (4618)
Bangkok, Thailand
80/HTTP
9123/HTTP
202.44.208.120
Tp-link Ac1200 Wireless Gigabit Vdsl Firmware
INET-TH-AS Internet Thailand Company Limited (4618)
Bangkok, Thailand
8282/HTTP
202.44.197.177
Tp-link Ac1200 Wireless Gigabit Vdsl Firmware
INET-TH-AS Internet Thailand Company Limited (4618)
Bangkok, Thailand
80/HTTP
9124/HTTP

- Even though endpoint TCP stacks have adopted stronger sequence number randomization and RST validation mechanisms, intermediary devices such as NAT routers remain weak links in the security chain.
- One specific recommendation of this project is that NAT devices implement stricter checks on inbound RST packets, ensuring the sequence number falls within the receive win- dow.
- Insight: fixing flaws in middleboxes—such as enforcing strict TCP window tracking for inbound RST packets—might itself invite new problems. For instance, enabling full TCP validation within NATs could degrade throughput or expose additional side channels based on how the device reacts to crafted probes. Patching these devices retroactively is not a clean or scalable solution and may lead to more subtle forms of leakage and attack opportunities.

Way Forward: Security by Design

- TCP was not created with security in mind.
- NAT was a workaround to the exhaustion of IPv4.
- Fixing them post-hoc is not a reliable solution.
- Transitioning to IPv6 removes the need for NAT altogether, and adopting modern, cryptographically secure transport protocols like QUIC can offer end-to-end security while mitigating these structural weaknesses.

Outline

1 Introduction

2 Background

3 Threat Model

4 Experimental Setup

5 Experiments & Result

6 Discussion & Conclusion

7 Future Work

Future Work

- Port-allocation strategies based NAT vulnerabilities
- Defense mechanisms like reverse-path validation, and their effectiveness under various attack scenarios.
- Explore different methods of NAT identifications.

References I

M. Allman, V. Paxson, and E. Blanton, *TCP Congestion Control*, Request for Comments: 5681, September 2009, See Section 3.2.

Steven M. Bellovin, *Security problems in the tcp/ip protocol suite*, ACM SIGCOMM Computer Communication Review **19** (1989), no. 2, 32–48.

Henning Brauer, Max Laier, Daniel Hartmeier, and Christian Steinruecken, *Security improvements in openbsd tcp/ip stack*, `https://www.openbsd.org/papers/asiabsdcon05-tcpip/tcpip.pdf`, 2005.

Yue Cao, Zhiyun Qian, Zhongjie Wang, Tuan Dao, Srikanth V Krishnamurthy, and Lisa M Marvel, *Off-Path TCP exploits: Global rate limit considered dangerous*, 25th USENIX Security Symposium (USENIX Security 16), USENIX Association, 2016, pp. 209–225.

W. Eddy, *Transmission Control Protocol (TCP)*, Request for Comments: 9293, August 2022, Current TCP specification.

———, *Transmission Control Protocol (TCP)*, Request for Comments: 9293, August 2022, See Section 3.5.2 (Reset Generation).

Lars Eggert and Fernando Gont, *Recommendations for transport-protocol port randomization*, `https://datatracker.ietf.org/doc/html/rfc6056`, January 2011.

Xuewei Feng, Qi Li, Kun Sun, Chuanpu Fu, and Ke Xu, *Off-path tcp hijacking attacks via the side channel of downgraded ipid*, IEEE/ACM Transactions on Networking **30** (2022), no. 1, 409–422.

Xuewei Feng, Yuxiang Yang, Qi Li, Xingxiang Zhan, Kun Sun, Ziqiang Wang, Ao Wang, Ganqiu Du, and Ke Xu, *Redan: An empirical study on remote dos attacks against nat networks*, Proceedings of the 2025 Network and Distributed System Security Symposium (NDSS 2025), 2025.

S. Guha, K. Biswas, B. Ford, and S. Sivakumar, *NAT Behavioral Requirements for TCP*, Request for Comments: 5382, October 2008.

Jonathan Hart, *R7-2014-17: Nat-pmp implementation and configuration vulnerabilities*, Rapid7 Blog, October 2014.

Michio Honda, Yoshifumi Nishida, Costin Raiciu, Adam Greenhalgh, Mark Handley, and Hideyuki Tokuda, *Is it still possible to extend tcp?*, Proc. ACM Internet Measurement Conference (IMC), 2011, pp. 181–192.

Michio Honda, Yoshifumi Nishida, Costin Raiciu, Adam Greenhalgh, Mark Handley, and Hideyuki Tokuda, *Is it still possible to extend tcp?*, Proceedings of the ACM SIGCOMM (2011), 181–192.

J. Mogul and S. Deering, *Path MTU Discovery*, Request for Comments: 1191, November 1990.

J. McCann, S. Deering, and J. Mogul, *Path MTU Discovery for IP version 6*, Request for Comments: 1981, August 1996.

J. Postel, *Transmission Control Protocol*, Request for Comments: 793, September 1981, Original TCP specification.

R. Penno, D. Wing, M. Boucadair, A. Stoenescu, and T. Reddy, *Updates to Network Address Translation (NAT) Behavioral Requirements*, Request for Comments: 7857, April 2016.

Zhiyun Qian, Z. Morley Mao, and Ying Zhang, *Off-path tcp sequence number inference attack—how firewalls can turn into a security hole*, Proceedings of the 21st USENIX Security Symposium (USENIX Security '12) (Bellevue, WA, USA), USENIX Association, 2012, pp. 347–362.

References III

- Y. Rekhter, B. Moskowitz, D. Karrenberg, G. de Groot, and E. Lear, *Address Allocation for Private Internets*, Request for Comments: 1918, March 1996.
- A. Ramaiah, R. Stewart, and M. Dalal, *Improving TCP's Robustness to Blind In-Window Attacks*, Request for Comments: 5961, August 2010.
- P. Srisuresh and K. Egevang, *IP Network Address Translator (NAT) Terminology and Considerations*, Request for Comments: 2663, August 1999.
- P. Srisuresh and M. Holdrege, *Traditional IP Network Address Translator (Traditional NAT)*, Request for Comments: 3022, January 2001.
- Joe Touch, *Defending against sequence number attacks*, https://datatracker.ietf.org/doc/html/rfc6528, 2012, RFC 6528.
- Ziqiang Wang, Xuewei Feng, Qi Li, Kun Sun, Yuxiang Yang, Mengyuan Li, Ganqiu Du, Ke Xu, and Jianping Wu, *Off-path tcp hijacking in wi-fi networks: A packet-size side channel attack*, Proceedings of the 2025 Network and Distributed System Security Symposium (NDSS), Internet Society, 2025.
- Nicholas Weaver, Robin Sommer, and Vern Paxson, *Detecting forged tcp reset packets*, Proc. Network and Distributed System Security Symposium (NDSS), 2009.
- Yuxiang Yang, Xuewei Feng, Qi Li, Kun Sun, Ziqiang Wang, Ao Wang, and Ke Xu, *Off-Path TCP Hijacking Attack to NAT-Enabled Wi-Fi Networks*, IEEE Transactions on Networking (2025), 1–16, Early Access.

Acknowledgments

- Prof. Mahavir Jhawar
- Adityavir
- Mphasis Lab
- Nikhil Raj
- Countless cups of tea...

QnA

Questions? Suggestions?

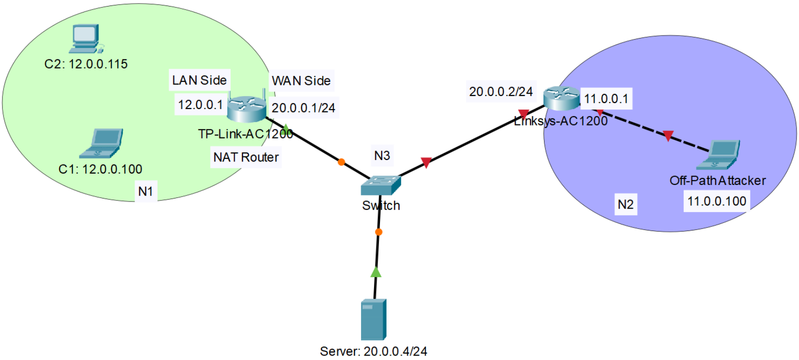


Figure: Alternative way of designing the network with off-path attacker