OXFORD

The Computer Journal, 2024, 1–14
https://doi.org/10.1093/comjnl/bxae059
Original Article

Security analysis of the MAP-T IPv6 transition technology

Ameen Al-Azzawi* and Gábor Lencse

Department of Networked Systems and Services, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, Műegyetem rkp. 3, Budapest 1111, Hungary

*Corresponding author. Department of Networked Systems and Services, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, Műegyetem rkp. 3, Budapest 1111, Hungary. E-mail: alazzawi@hit.bme.hu

Abstract

In this paper, we focus on one of the most prominent IPv6 transition technologies, namely Mapping of Address and Port using Translation (MAP-T), and we give attention to Mapping of Address and Port with Encapsulation (MAP-E) as well. We emphasize the uniqueness of MAP-T and MAP-E, and we discuss the differences between those two technologies, including their topology, functionality, and security vulnerabilities. We apply a threat modeling technique, Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege (STRIDE), to assess potential vulnerabilities in the MAP-T infrastructure. Furthermore, we build a testbed for MAP-T using the open-source software, Jool, and we conduct testing on the translation process capabilities of Jool and its port allocation per subscriber. Finally, we present various attacking scenarios against the main routers of MAP-T, such as IP address spoofing, information disclosure, and source port exhaustion, and we propose mitigation methods for several attacks.

1. INTRODUCTION

The growth of the Internet and the proliferation of connected devices have led to a rapid depletion of the available IPv4 address space. As a result, there was a growing need for a new version of the Internet Protocol that could provide a much larger address space, which led to the development and deployment of IPv6. However, transitioning from IPv4 to IPv6 is a complex and challenging process that requires careful planning, significant investment and coordination among multiple stakeholders.

One of the major obstacles in adopting IPv6 is its coexistence with the legacy IPv4 infrastructure and applications, which poses interoperability and compatibility challenges. Several transition technologies have been proposed to address these challenges, including Dual Stack (deploying IPv4 and IPv6 stacks on the same device), Tunneling, and Translation method [1]. These technologies allow for the coexistence of IPv4 and IPv6 networks during the transition period, ensuring a smooth and seamless migration from the former to the latter. Several research works were carried out to narrow down the list of promising technologies that can be deployed to ensure reliable communication among two devices having different IP versions or between two devices having the same IP version, while the network between them is based on another IP version [2]. In 2013, a new translation technology called Combination of Stateful and Stateless Translation (464XLAT) [3] was presented by T-Mobile in the USA. The technology is based on a double translation mechanism using two translation routers, a Customer-side translator (CLAT) and a Provider-side translator (PLAT). In previous research, the 464XLAT technology was identified as one of the most promising technologies on the market. Subsequently, our previous research focused on conducting a comprehensive security analysis of 464XLAT using the STRIDE method and building a testbed using Debian virtual machines [4].

We found out that 464XLAT is liable to several attacking possibilities such as Denial of Service (DoS) attacks.

Moreover, we have previously surveyed the five most prominent IPv6 transition technologies (464XLAT, DS-Lite, lw4o6, and MAP-T/MAP-E) [2]. Later, we examined the security threats of DS-Lite [5] and lw4o6 [6].

All of that has led us to the next translation technology named MAP-T [7]. The technology itself is part of two coupled technologies, MAP-T and MAP-E.

The essential difference is that in MAP-E, the IPv4 packet is encapsulated within an IPv6 packet, whereas in MAP-T, the IPv4 packet is translated into an IPv6 packet. While MAP-E is based on encapsulation, we decided to investigate more translation methods. As a result, we have focused mainly on MAP-T in this paper.

The main target of this paper is to analyze the potential security threats to MAP-T infrastructure, expose them, and find methods to mitigate them. We plan to fulfill our objectives as follows:

- applying the STRIDE threat modeling technique to the Data-Flow Diagram (DFD) of a MAP-T system to discover all potential security vulnerabilities;
- building a testbed for MAP-T using reliable open-source software to test its capabilities and translation process;
- executing various attacking scenarios against MAP-T main routers to test their resilience, strength and security vulnerabilities;
- eventually proposing practical mitigation methods for such attacks.

The rest of the paper is organized as follows. Section 2 presents the background of MAP-E and MAP-T and their operation. Section 3 discusses the related work that has been carried out

Received: November 29, 2023. **Revised:** April 3, 2024. **Accepted:** June 21, 2024

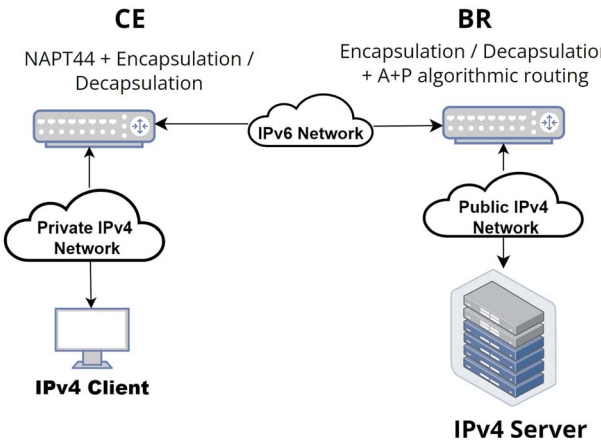

Figure 1. MAP-E topology.

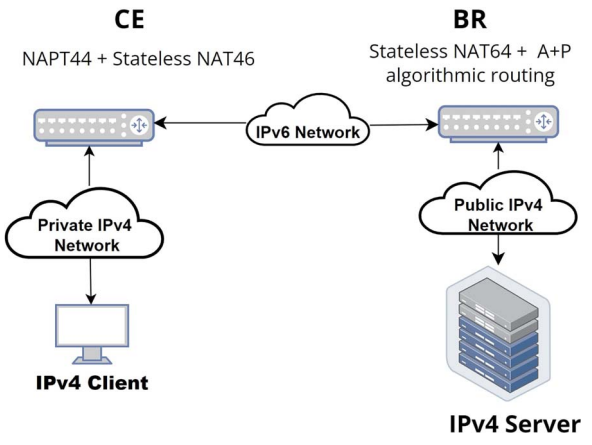

Figure 2. MAP-T topology.

before in a similar domain. Section 4 introduces the security analysis of both MAP-E and MAP-T architecture based on the STRIDE threat modeling technique. Section 5 presents the possible implementation scenarios for MAP-T infrastructure using the Jool software, and explains our MAP-T testbed, its components, and the required resources to build it. Section 6 summarizes the attacking possibilities against MAP-T infrastructure (such as spoofing and source port exhaustion), categorizes their severity, and proposes a mitigation method for each of them. In Section 7, we discuss our findings and correlate them with our previous research results to find a pattern of common vulnerabilities among several IPv6 transition technologies. In Section 8, we outline directions for future research opportunities regarding the IPv6 transition technologies.

Finally, in Section 9, we summarize the main contributions of the paper, its novelty, and the lessons learned.

2. MAP-E/MAP-T OPERATION

In Figs 1 and 2, we illustrate the main differences between MAP-E and MAP-T topologies and the core of their functionalities, which are summarized as follows:

- Both technologies consist of two main components, the Customer Edge (CE) router and the Border Relay (BR) router, and each one of them has its own purpose and functionality.
- Both technologies have a shared feature that they critically depend on, which is called 'Port Mapping'. This feature allocates a certain range of User Datagram Protocol (UDP)/Transmission Control Protocol (TCP) ports to certain CE routers so that applications of the user (IPv4) client can communicate with the IPv4 server. The port sets are previously provisioned in the CE and BR routers.
- The benefit of port mapping is the ability for multiple CE routers to share the same public IPv4 address, given that they use unique port sets. The port set is selected through a parameter called Port Set Identifier (PSID) [8].
- Both technologies adopt the MAP-domain deployment, which represents one or multiple CE and BR devices connected through a virtual link [8]. The Internet Service Providers (ISPs) can deploy more than one MAP domain as needed.
- Both MAP-T and MAP-E configure one or more MAP-rule(s), which are sets of parameters that regulate the communication between the MAP nodes (CE and BR) or among the CEs themselves, especially when it comes to IPv4/IPv6 prefixes and port sets. Those rules are unique among the MAP domains such as Basic mapping rule (BMR), Forwarding Mapping Rule (FMR), and Default Mapping Rule (DMR).

2.1. MAP-E operation

Figure 1 shows a simple topology of MAP-E, which consists of four elements: IPv4 client, CE, BR, and IPv4 server. MAP-E was originally presented in RFC-7597 [8]. We explain below the packet journey throughout the MAP-E infrastructure and what are the CE and BR roles here:

- The IPv4 client sends a UDP/TCP packet to the IPv4 server.
- The CE router receives the packet and performs the Network Address and Port Translation (NAPT44) process. The CE router then adds an entry to its NAPT44 binding table (Table 1). Subsequently, the CE router encapsulates the translated IPv4 packet into an IPv6 packet, which is commonly referred to as a '4in6 packet'. This encapsulated packet contains the original IPv4 packet within an IPv6 packet. The 4in6 packet is then forwarded to the BR router.
- The BR router receives the 4in6 packet, decapsulates the IPv6 packet to reveal the original IPv4 packet, selects a best-matching MAP domain rule (based on the prefixes and port set), and then forwards the packet to the IPv4 server.
- The IPv4 server receives the packet, and replies with CE's public IPv4 address as a destination address.
- The BR router receives the packet, encapsulates it in an IPv6 packet and forwards it to the CE through the IPv6 interface.
- The CE router receives the packet back, decapsulates the public IPv4 address, and looks for a matching entry from Table 1. When a matching entry is found, the source IPv4 address and the source port number will be translated back according to the found entry, and the packet will be forwarded to the IPv4 client.
- Finally, the IPv4 client receives the answer of the IPv4 server.

2.2. MAP-T operation

MAP-T was presented in RFC-7599 [7], its infrastructure does not differ much from MAP-E. As shown in Fig. 2, the main difference is that MAP-T uses translation instead of encapsulation on its MAP nodes (CE and BR routers). Below is the packet journey throughout the MAP-T infrastructure:

- The IPv4 client sends a UDP / TCP packet to the IPv4 server.
- The CE router receives the packet, performs NAPT44 translation (just like in MAP-E), and then adds the translation entry to its NAPT binding table (Table 1). The CE router then

Table 1. CE router NAPT table

Src. IP nddress	Src. Port number	External IP address	Temporary port number	Dest. IP address	Dest. port number	Transport protocol
10.0.0.2	5000	203.0.113.1	1050	192.0.2.2	80	TCP

performs a stateless NAT46 translation process by translating the IPv4 packet into an IPv6 packet using the pre-configured prefix (end-user IPv6 prefix) and forwards the translated packet to the BR.

- The BR router undertakes the following actions upon receiving the IPv6 packet:
  - Derives the source and destination port and translates the IPv6 header to IPv4 one.
  - Forwards the resulting IPv4 packet to its final destination (the IPv4 server in our example).

  The BR router is already provisioned with the public IPv4 address and the allocated port range for each CE within the MAP domain.
- The IPv4 server receives the IPv4 packet and replies with CE's public IPv4 address as the destination address.
- The BR router undertakes the following actions upon receiving the IPv4 packet:
  - Computes the IPv6 source address from the IPv4 source address and BR's IPv6 prefix.
  - Computes the IPv6 destination using the FMR rule based on the IPv4 destination address and destination port number.
  - Forms an IPv6 packet and forwards it to the CE router.
- The CE router receives the IPv6 Packet and performs several processes on the packet, such as pre-routing, filtering, and NAPT44 translation as explained below:
  - The CE router performs a stateless NAT64 translation on the IPv6 packet to an IPv4 packet based on the configured BMR.
  - The resulting IPv4 packet will be forwarded to the NAPT44 function when a matching entry in Table 1 is found, the destination IPv4 address and destination port will be translated and the packet will be forwarded to the IPv4 client.
  - Finally, the IPv4 client receives the answer from the IPv4 server.

2.3. IPv4 destination address and MAP-Domain

When considering the packet traversal in the direction of IPv4 client ⇒ CE ⇒ BR ⇒ IPv4 server, the value of the IPv4 destination address plays a significant role in the translation process within the CE router, which has two scenarios:

- If the IPv4 destination address is within the MAP domain:
  - The source IPv6 address is derived via the BMR.
  - The destination IPv6 address will be synthesized using the FMR.
- If the IPv4 destination address is outside the MAP domain:
  - The source IPv6 address will be the MAP CE's IPv6 address.
  - The destination IPv6 address will be synthesized using the Default Mapping Rule. For instance, if 192.0.2.2 (0xc0000202 in hexadecimal) is the destination IPv4 address and the DMR IPv6 prefix is 64:ff9b::/96, then the IPv4 embedded IPv6 destination address will be translated to 64:ff9b::c000:202.

2.4. Port mapping

As we described previously, MAP-E and MAP-T assign specific port set for every CE router by applying a criterion illustrated in RFC-7597 [8] Section 5.1, where PSID can be calculated as follows:

- Port set size = 10 ⇒ 2^{10} = 1024 ports per set.
- PSID length = 6 ⇒ number of port sets: 2^6 = 64, it is also called 'the sharing ratio' [8].
- PSID = 1 ⇒ allocated ports range = [1024-2047].

Below is a scenario for port number allocation within multiple CE routers that share the same public IPv4 address while having different port sets:

As the total number of source port numbers is 2^{16}=65 536, and if we assume that the sharing ratio is 64 (see above), then the size of each port set is 1024 (65 536 / 64) ports. We still need to define the PSID value per CE, to assign a port set for each CE router:

- PSID = 0 ⇒ allocated ports = [0–1023].
- PSID = 1 ⇒ allocated ports = [1024–2047].
- PSID = 8 ⇒ allocated ports = [8192–9215].

In conclusion, with PSID length = 6 (sharing ratio is 64), we can support 64 unique CEs, each of them having 1024 source port numbers. Additionally, if we choose PSID = 1, the range of ports that will be assigned will amount to [1024–2047].

However, it is recommended to exclude the first set as it is reserved for the well-known ports [0– 1023], which leaves us with 63 subscribers (CPEs) in this scenario.

3. RELATED RESEARCH

In [9], the performance and scalability of the 464XLAT and MAP-T IPv6 transition technologies were tested and compared. As for MAP-T, the authors built a testbed on Debian-based machines, where they used the Jool open-source software [10] to implement CE and BR routers. The authors tested the scalability of the performance of CE and BR routers while adding more Central Processing Unit (CPU) cores and monitoring the performance. They found out that the BR router scales better than the CE router, where a bottleneck is obvious at the CE side [9]. However, the authors' focus was merely on performance and not on security, which is our current focus area. Another test environment for MAP-T infrastructure was built by several researchers in Brazil [11]. The research aimed to test the connectivity of some applications using the MAP-T translation mechanism. As for MAP-T implementation, they used software developed by Cernet Center [12]. They used Fedora Linux-based virtual machines for the CE and BR routers. In addition, the host machine (IPv4 client) used three different machines (Linux Kubuntu, Windows 7 and Windows XP) to gather as many results as possible. Unfortunately, this testbed

also did not inspect nor analyse the security threats that MAP-T might face.

A MAP-T tester was designed by Al-Hamadani [13], where the author presented his progress toward building the world's first RFC-8219 compliant tester for MAP-T to record some measurements such as throughput and latency. For his MAP-T testbed, the author used the Jool software [10] to implement the CE and BR routers.

Furthermore, another experiment was performed by Georgescu [14], where the authors built a penetration testbed for MAP-T and conducted multiple attacking scenarios such as Address Resolution Protocol (ARP) Cache Poisoning, Neighbour Advertisement Flooding, and Traffic Analysis.

Another study conducted by [15] introduced the IPv6 Network Evaluation testbed (IPv6NET), aiming to gather feasibility insights for devising a scenario-driven IPv6 transition strategy. It outlined the structure of the IPv6NET testing approach and presented empirical findings for a designated network scenario. The empirical data encompasses network performance metrics (latency, throughput, packet loss) and operational indicators (configuration, troubleshooting, application support). One of the tested technologies was MAP-T, where the authors tested the targeted technologies under two environments: closed (isolated testbeds) and open (with internet access).

The authors of [15] recognized the absence of scalability assessment in their previous work and subsequently extended their research to propose a method for quantifying scalability [16]. This article represents an initial endeavor to benchmark the load scalability of IPv6 transition technologies. The methodology introduces a metric termed 'network performance degradation', which quantifies the percentile degradation of network performance aspects as scalability increases. This metric is deemed significant as it impacts various IPv6 transition scenarios irrespective of their scale. The primary contributions of this article lie in its detailed benchmarking methodology and empirical scalability results. These results can directly aid network operators in navigating the IPv6 transition process. Through the proposed methodology, the authors successfully benchmarked the load scalability of two open-source IPv6 transition implementations, namely asamap and tiny-map-e, encompassing various IPv6 transition technologies such as MAP-E and MAP-T.

Despite the valuable insights provided by the aforementioned studies on MAP-T deployment, scalability, and performance evaluation, there remains a notable gap in the literature regarding the security analysis of MAP-T. While studies such as those conducted by [13] and [16] have contributed to understanding of the functionality of MAP-T and performance metrics, there is a distinct lack of research focusing on identifying and addressing potential security vulnerabilities inherent in the MAP-T protocol. Security is a critical aspect of any networking technology, particularly in the context of transition mechanisms where new attack vectors may emerge.

4. SECURITY ANALYSIS

There are multiple methods to conduct security analysis and threat modeling, such as Common Attack Patterns Enumeration and Classification (CAPEC) and STRIDE. In a nutshell, CAPEC is a tool that helps developers to think like hackers, while it provides the basic knowledge of attack patterns [17]. CAPEC can list attack patterns and allow researchers and developers to identify weaknesses in a system, which can be exploited through an attack.

However, we have selected STRIDE to be our threat modeling tool due to its wide coverage of several topologies and most importantly its reputation, as it has been used by various research works to analyze potential security threats for multiple network infrastructures.

4.1. STRIDE methodology

The STRIDE method was explained by Shostack [18], where he summarized all the possible attacks that any network system might be susceptible to. These types of attacks are briefly explained as follows:

- Spoofing: the act of an adversary to claim that he is someone else to perform any malicious activities.
- Tampering: the possibility of altering the actual content of the exchanged data.
- Repudiation: the act of denying the responsibility of a certain act (most probably malicious one), like sending a specific packet.
- Information Disclosure: accessing highly confidential information.
- Denial of Service: one of the most implemented attacks worldwide and it is based on overwhelming the target with a huge number of queries that are not useful to the targeted server while blocking the legitimate queries from being processed at all.
- Elevation of Privileges: the access of a certain user to a higher level sensitive data, which he was not supposed to reach. This could be by granting himself root user privilege and accessing confidential documents.

The type of vulnerability depends on what is being done with the data (processing, storing, etc..). Table 2 shows those vulnerabilities accordingly.

4.2. Applying STRIDE on MAP-T

According to [18], to inspect the vulnerabilities of a system, a DFD of the system's topology needs to be built to identify the threat points one by one. As shown in Fig. 3, we identified 11 points of vulnerabilities, which we intend to inspect in the next section.

4.3. MAP-T attacking possibilities

4.3.1. *IPv4 client*

1) Spoofing: an adversary spoofs the IP address of the IPv4 client and starts sending a huge number of packets toward the CE router to fully utilize its processing power.
2) Repudiation: an adversary denies that he sent a specific packet from the IPv4 client machine. As a result, the attacker might send a harmful packet (echo request, DNS resolution request, etc.) to the CE router and then deny his responsibility.

4.3.2. *Data flow from the IPv4 client to the CE Router*

1) Tampering: an attacker sniffs the communication channel and tampers (alters) the content of the exchanged data like destination IP address, Time to Live (TTL) value, etc. Changing the destination IP address would diverge the packet to a malicious server and cause an Failure of Service (FoS) attack [1], which means a legitimate user does not get a response to his request.
2) Information Disclosure: an adversary gets access to confidential information such as online banking login credentials, user browsing habits, etc.

Table 2. Vulnerability of different DFD elements to different threats [1]

	Spoofing	Tampering	Repudiation	Information Disclosure	Denial of Service	Elevation of Privilege
Data Flow		✓		✓	✓	
Data Stores		✓		✓	✓	
Processes	✓	✓	✓	✓	✓	✓
Interactors	✓		✓			

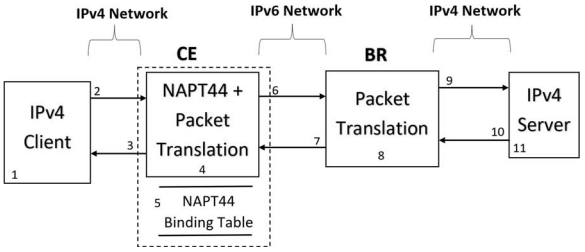

Figure 3. Data flow diagram of MAP-T.

3) Denial of Service: an attacker sends a huge number of useless queries to the CE router to overwhelm it and prevent legitimate packets from being processed.

4.3.3. *Data flow from the CE router to the IPv4 client*

1) Tampering: in this direction, the victim is the IPv4 client himself, while the attacker performs Man-in-the-Middle (man-in-the-middle attack (MITM)) and alters the content of the sent packets [18]. Another potential risk is an injection attack, where the attacker injects malicious packets into an existing network flow [19]. Moreover, an adversary might inject a TCP RST signal into the client, causing the termination of the already-established TCP connection.
2) Information Disclosure: an attacker gets unauthorized access to sensitive information from the packets headed from the CE router toward the client.
3) Denial of Service: an adversary flooding the IPv4 client with an excessive amount of requests with the intention to overwhelm it and deprive it of the ability to process any more packets.

4.3.4. *The CE router*

1) Spoofing: an attacker's machine spoofs the source IP address of the CE router and starts communicating with other network elements of MAP-T infrastructure such as the IPv4 client or the BR router. This action endangers every machine that talks to him (the attacker) by exposing sensitive information that is meant to be sent to the legitimate CE router. On the other hand, ARP cache poisoning attack is also possible here, where an attacker can intercept network traffic between the CE router and BR router (for instance). The attacker can impersonate the BR router by sending fake ARP messages to the CE router, causing it to update its ARP cache with the attacker's Media Access Control (MAC) address instead of the correct MAC address of the BR router. Once the attacker has successfully poisoned the ARP cache of the target device, he can intercept and modify network traffic, steal sensitive information or launch other attacks [20].
2) Tampering: simply modifying the actual content of the packet's information such as the destination IPv6 address and redirecting the packet toward a dangerous recipient instead of the BR router. Moreover, a change in the source port will change the entry in the NAPT table (see Table 1) and result in forwarding the returned packet to a malicious server instead of the original requester.
3) Repudiation: an attacker spoofs the source IPv6 address of the CE router and performs all sorts of illegal activities, then denies the fact that he did so because it was not his IP address, but the actual CE's IP address.
4) Information Disclosure: an attacker having unauthorized access to confidential data within the CE router such as the whole routing table of the CE router or the payload's content of some packets.
5) Denial of Service: an attacker flooding the CE router with a huge number of useless packets and putting it out of service for at least several seconds (depending on the computation power of the CE router) [18].
6) Elevation of Privileges: an adversary gets highly privileged permissions inside the CE router such as root permission, which makes it easier for the attacker to perform his malicious activities. Those kinds of attacks happen mostly due to an inside job [1].

4.3.5. *The NAPT44 binding table of the CE router*

1) Tampering: an attacker alters the NAPT44 table (see Table 1), such as the public IPv4 address, port number, etc. Such an act will lead to a wrong translation process by the CE router and eventually re-direct the traffic to a malicious server.
2) Denial of Service: an attacker inundates the CE router with superfluous packets. This flood of packets results in the addition of unnecessary entries to the NAPT44 table, eventually overwhelming it and causing legitimate entries to be dropped.

4.3.6. *Data flow from the CE router to the BR Router*

1) Tampering: an attacker sniffs the communication channel and performs an MITM attack, where the attacker alters the content of the transmitted data, causing packet's redirection [18], please refer to Section 4.3.2.1.
2) Information Disclosure: when the attacker sniffs the channel, he exposes the sensitive/confidential information and puts both CE and BR routers at risk. Please refer to Section 4.3.2.2.
3) Denial of Service: an attacker floods the BR router with an excessive stream of packets to overwhelm its processing power [18], please refer to Section 4.3.2.3.

4.3.7. *Data flow from the BR Router to the CE router*

1) Tampering: an attacker intercepts the flowing data and alters the packet content such as IP addresses and port numbers, which shifts the traffic to a malicious recipient [18], please refer to Section 4.3.2.1.

Table 3. Summary of the potential vulnerabilities of MAP-T

Attack name	Intricacy of performing the attack	Intricacy of mitigation	Attack impact (severity)
DoS	Easy	Difficult	Critical
Man-in-the-Middle (MITM)	Average	Difficult	High
Information Disclosure	Average	Average	Medium
Source IP address Spoofing	Easy	Difficult	Critical
Source Port exhaustion	Average	Average	Medium
TCP RST Signal	Easy	Easy	Low
TCP SYN Flood	Easy	Average	High
Packet's Payload Tampering	Average	Difficult	High
ARP Poisoning	Average	Difficult	High

2) Information Disclosure: an attacker gets access to the exchanged data between CE and BR routers, which might contain confidential information such as online banking credentials.
3) Denial of Service: an attacker gets access to the BR router and sends a vast number of packets toward the CE router to overwhelm its processing power, please refer to Section 4.3.2.3.

4.3.8. *The BR router*

1) Spoofing: an attacker impersonates the BR router by spoofing its source IP address and starts communicating with the CE router or the IPv4 server, which endangers all of the involved machines with the possibility of sharing sensitive data with the attacker [18]. Another risk is the ARP Cache Poisoning attack, where the attacker sends a falsified ARP message to the BR router to associate the attacker's MAC address with the spoofed IP address of another host (IPv6 address of CE for example) [20].
2) Tampering: the attacker might alter the data that is being processed within the BR router itself such as the mapped IP addresses, port numbers, DMR and FMR rules, etc. Tampering with such data will lead to altering the IPv6 destination address and eventually forwarding the packet to the malicious server.
3) Repudiation: when the attacker spoofs the source IP address of the BR router, he can send harmful packets to a recipient and then claim he has not done so because the packet's source IP address does not belong to him, but to the BR router.
4) Information Disclosure: an attacker gets access to the BR router and therefore exposes confidential data such as the packet's payload, that might contain confidential information.
5) Denial of Service: an attacker executes the notorious process of a DoS attack, which involves flooding the BR router with a large number of packets to exhaust its CPU and prevent the BR router from processing incoming packets from the CE router or the IPv4 server for a specific amount of time.
6) Elevation of Privileges: an attacker getting high privilege access inside the BR router such as read and write permission, which will allow him to run harmful scripts such as altering the routing table.

4.3.9. *Data flow from BR router toward IPv4 Server*

1) Tampering: an attacker alters the content of the exchanged data. This could involve injecting malicious code, and commands or manipulating the data in a way that leads to unintended consequences or security vulnerabilities. Please refer to Section 4.3.3.1.
2) Information Disclosure: an attacker exposes the content of the exchange packets' payload, such as online banking credentials, which the IPv4 client originally sent. Please refer to Section 4.3.3.2.
3) Denial of Service: an attacker exhausts the IPv4 server with an extreme rate of packets to overwhelm it and prevent it from replying to the legitimate request of the BR router, and, therefore, to the original request of the IPv4 client. Please refer to Section 4.3.3.3.

4.3.10. *Data flow from IPv4 server toward BR router*

1) Tampering: an attacker creates fraudulent data packets masquerading as legitimate requests by forging source addresses, manipulating headers, or injecting malicious code within the packets, please refer to Section 4.3.2.1.
2) Information Disclosure: an attacker eavesdrops on the exchanged data and extracts sensitive information, such as a token sent by the IPv4 server, which can be used to access a particular service, please refer to Section 4.3.2.2.
3) Denial of Service: an attacker overwhelms the BR router with a flood of requests to make it unavailable or unresponsive to legitimate requests from both the IPv4 server and the CE router, please refer to Section 4.3.2.3.

4.3.11. *IPv4 server*

1) Spoofing: an attacker sends a packet with a spoofed source IP address (IPv4 server IPv4 address) and communicates with the BR router, which endangers the BR router by trusting and exchanging sensitive information with a malicious machine. Please refer to Section 4.3.1.1.
2) Repudiation: after spoofing the source IP address of the IPv4 server, an attacker sends the harmful packet to the BR router, then denies his responsibility for such an act, because the logged source IP address doesn't belong to him but to the IPv4 server, please refer to Section 4.3.1.2.

After conducting a thorough analysis of potential security threats in MAP-T, we evaluated the severity of each identified threat. Table 3 categorizes those threats according to three key parameters: Intricacy of Performing the Attack, Intricacy of Mitigation, and Attack Impact (Severity). Furthermore, Table 3 provides a structured classification of threats based on the complexity involved in executing the attack, the level of complexity in mitigating it, and the severity of its impact on the targeted

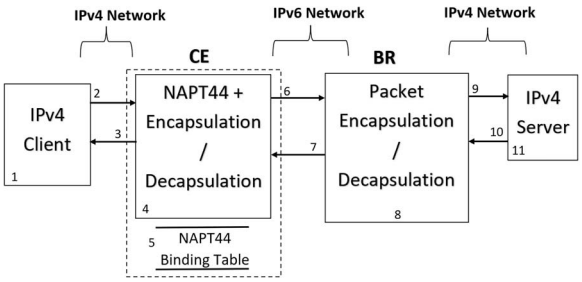

Figure 4. Data flow diagram of MAP-E.

system or data. The 'Intricacy of Performing the Attack' column denotes the difficulty or complexity an attacker faces in executing a specific threat. This factor considers the technical expertise, resources, and effort required to launch the attack successfully.

Conversely, the 'Intricacy of Mitigation' column assesses the level of complexity involved in detecting and mitigating the identified threats. It accounts for the challenges and intricacies faced by defenders in identifying and effectively neutralizing these threats once they occur.

Lastly, the 'Attack Impact (Severity)' column highlights the potential severity of each threat's impact on the targeted system or data. This categorization considers the potential harm caused by the attack, encompassing factors such as data integrity compromise, system availability disruption, and confidentiality breaches. The categorization was based on two factors: estimation based on the sophistication of the attack, and assessment based on our own experience in attacking and mitigations. For example, the DoS attack was easy to implement and difficult to mitigate, which is why we categorized it as Critical. On the other hand, spoofing attacks using Scapy software [21] were medium-level to build (based on our experience) and difficult to mitigate (we were not able to mitigate it). That's why we also categorized it as medium-level. The same process applies to the rest of the attacks.

4.4. Applying STRIDE on MAP-E

As shown in Fig. 4, MAP-E infrastructure also has 11 attacking possibilities, and it has almost identical DFD to MAP-T, which makes the potential security threats (to some extent) similar to the ones that MAP-T faces. However, unlike MAP-T, MAP-E adapts the packet's encapsulation, which will cause relatively different sorts of attacking possibilities. We have summarized the main possible attacks against MAP-E infrastructure:

- DoS: an attacker sends a large amount of traffic to the BR router to overwhelm it and prevent it from processing legitimate traffic, which disrupts network connectivity and services. DoS can also be harmful to the CE router as it overflows the NAPT44 table with too many useless entries and erases the legitimate ones, see Table 1
- Spoofing: an attacker can spoof the IPv6 source address of the CE router and craft an IPv4-embedded IPv6 packet to bypass access controls or launch a reflection or amplification attack against the BR router and vice versa [22].
- MITM: an attacker can intercept traffic between the CE and BR routers, potentially allowing them to eavesdrop on sensitive information or modify traffic in transit.
- Unauthorized access: an attacker can gain unauthorized access to the CE router and modify or extract sensitive information such as the content of the NAPT44 table (see Table 1). Such data data-altering attacks could lead to wrong encapsulation and forwarding of the IPv4-embedded IPv6 packet to the wrong recipient.

5. MAP-T IMPLEMENTATION

5.1. Jool implementation

In the current research, we focus on the MAP-T implementation and leave MAP-E for future research opportunities. For example, Jool [10], which is an open-source IPv4/IPv6 translator, supports MAP-T and other solutions, but not MAP-E. On the other hand, VPP [23] does support (in theory) both MAP-E and MAP-T. However, it proved to be complicated to configure. As a result, we have chosen Jool as a candidate for our MAP-T implementation.

As shown in Fig. 5, MAP-T consists of two main routers (CE and BR), and each one of them acts differently. The CE router consists of two separate name spaces (napt and global). The 'napt' namespace represents the communication between the IPv4 client and the CE router. In contrast, the 'global' namespace oversees the communication between the CE router and the outside world (the BR router). On the CE side, Jool applies the NAT64 well-known prefix (64:ff9b::/96) to translate the IPv4 address into the IPv6 address by appending the IPv4 address to the end of the IPv6 address. For example, if the IPv4 destination address (for packet heading from IPv4 client toward IPv4 server) is 192.0.2.2, it will be translated into 64:ff9b::c000:202. Both namespaces are connected through static route Linux commands:

```
ip netns exec napt ip route add default via
192.168.0.1
ip route add 64:ff9b::/96 via 2001:db8:6::1
ip route add 203.0.113.8/32 via 192.168.0.2
```

The full bash script configuring the CE router is available through the '`CE.sh`' script in our GitHub repository [24].

As for the BR, Fig. 5 shows that it has only one name-space and IPv6 address-equipped interface (ens34), where it receives the translated packet from the CE router and translates it back to IPv4 and forwards it to its final destination (IPv4 server in our example).

For BR to perform such actions, the Jool implementation applies another set of Linux commands:

```
ip route add 2001:db8:ce:11b::/64 via 2001:db8:6::
11b /sbin/modprobe jool_mapt
jool_mapt instance add "BR" –netfilter –dmr 64:ff9b
::/96
jool_mapt -i "CE 11b" global update end-user-ipv6-
prefix \  2001:db8:ce:11b::/64
```

The full bash script to configure the BR router is also available through the '`BR.sh`' script in our GitHub repository [24]. Moreover, the installation of the Jool software itself was made possible through an automated script '`jool.yml`' in the same repository.

5.2. MAP-T testbed

To build our testbed, we used a 'P' series node of NICT StarBED, Japan [25], which is a Dell PowerEdge 430 server that has the following details: two 2.1GHz Intel Xeon E5-2683v4 CPUs with 16 cores each, 348GB 2400MHz DDR4 SDRAM. On this server, we have installed a Windows 10 Pro operating system.

As shown in Fig. 5, our testbed consists mainly of four machines created with Linux-based Debian-10 Virtual machines built on top of VMware Workstation player virtualization software. Every machine has 8-GB RAM, 6 CPU cores, and 20-GB HDD. The

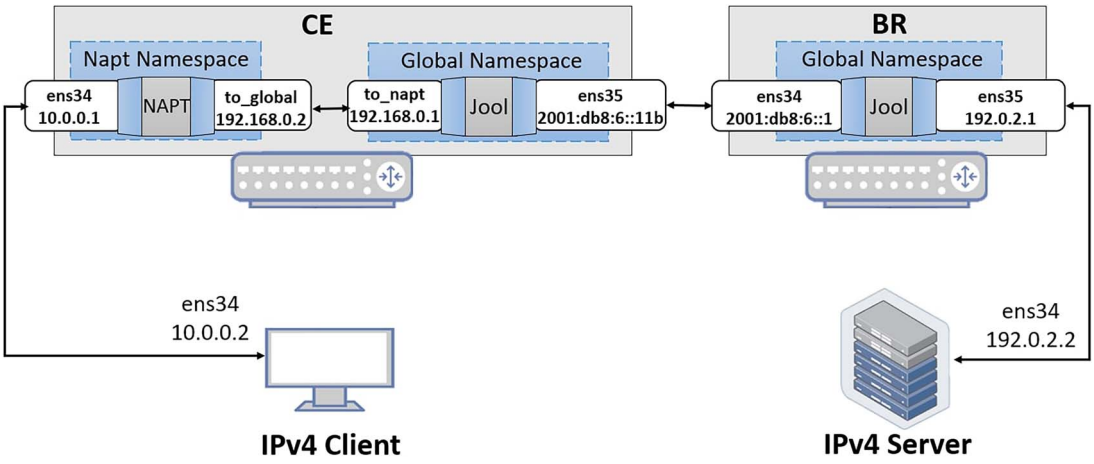

Figure 5. MAP-T implementation using the Jool software.

Table 4. MAP-T packet tacketranslation packetrocess on CE and BR

Packet capture at IPv4 Client `ens34`
`1 0.000000000 10.0.0.2 > 192.0.2.2 UDP 42 Source port: 5000 Destination port: 80`
Packet capture at CE `to_napt`
`1 0.000000000 203.0.113.8 > 192.0.2.2 UDP 42 55398 → 80 Len=0`
Packet capture at CE `ens35`
`1 0.000000000 2001:db8:ce:11b:0:cb00:7108:1b > 64:ff9b::c000:202 UDP 62 55398 → 80 Len=0`
Packet capture at BR `ens35`
`1 0.000000000 203.0.113.8 > 192.0.2.2 UDP 42 55398 → 80 Len=0`

full and detailed configurations were written and explained in an automated manner in our GitHub repository [24].

We have used an open-source IT automation tool called Ansible [26] to automate the installation process. Our automation script inspects the host machine, determines what Linux distribution it has, installs all of the dependencies, deploys the Jool software, and runs all necessary routing commands.

In the repository, separate automation scripts were written and customized to configure the IPv4 client, IPv4 server, CE, and BR router separately [24].

5.3. Results

While sending a crafted UDP packet from the IPv4 client (with a source port of 5000) to the IPv4 server (see Fig. 5), we monitored the traffic with the `tshark` command at different locations. As shown in Table 4, the NAPT44 + NAT46 translation process on the CE router's side and the stateless NAT64 on the BR router's side are quite obvious. UDP Packets were successfully translated by the CE and the BR routers as configured and no packet loss was reported.

As shown in Table 4, packets stemming from ens35 of the CE router (CE ⇒ BR) have the following source IPv6 address: `2001:db8:ce:11b:0:cb00:7108:1b`.

The above IP address consists of two separate segments:

- end-User-IPv6 Prefix (2001:db8:ce:11b), as configured in CE and BR,
- IPv6 Interface Identifier (0:cb00:7108:1b).

As illustrated in Fig. 6, the IPv6 Interface identifier consists of three main parts (padding, IPv4 address and PSID). As a result, applying the same structure on our IPv6 Interface Identifier (0:cb00:7108:1b) shows us the following:

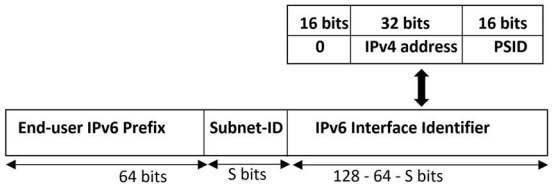

Figure 6. Derivation of MAP-T IPv6 address.

- 0: padding on the left side.
- cb00:7108: hexadecimal version of the public IPv4 address (203.0.113.8), which is the allocated public IPv4 address for the CE router.
- 1b: the PSID value, which is '27' in decimal and corresponds to ports [57, 343-55, 296] that we assigned to our CE router.

The PSID value is derived and calculated based on the PSID length and the given Embedded Address (EA) bits, which should be pre-configured. In our Testbed implementation, we passed PSID length of 2048 and '13' to EA bits, which led to PSID = 27.

The derivation process of PSID value out of EA bits and PSID length is explained in RFC-7597, Section 5.2 [8]

6. ATTACKING SCENARIOS AND MITIGATION

As we summarized the list of attacks in Table 3, we have implemented some of those attacks against multiple machines, especially the CE and BR routers.

6.1. UDP packet spoofing

As shown in Fig. 7, we spoofed the source IPv6 address of the outgoing traffic (CE ⇒ BR). To achieve that, we used a powerful

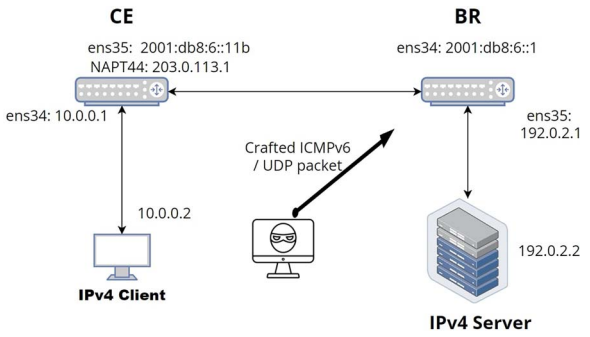

Figure 7. ICMPv6/UDP spoofing attacks.

interactive packet manipulation program called Scapy [21], which has multiple use cases such as sniffing and spoofing. The attack was based on a Python script that we wrote (`udp-spoofer.py`) [24]. The script sniffs the communication channel between CE and BR, looks for UDP traffic (CE ⇒ BR), grabs the source and destination port number, and then crafts a new UDP packet with the same details (IP addresses, MAC addresses, and port numbers) and sends the crafted packet to the BR router. However, the script alters the payload of the packet with a random text to imitate a real-life attack scenario.

The importance of the attack lies in the fact that the crafted UDP packet was received and processed by the BR router and then forwarded to the IPv4 server.

The BR router was deceived and processed a malicious packet as if it came from a legitimate CE machine. Such action endangers the BR router and the IPv4 server by receiving a stream of data from an attacker while believing it is a trustworthy one.

6.2. ICMPv6 packet spoofing

A similar procedure was repeated for the ICMP packet: sniffing, monitoring, and sending a similar and crafted ICMP packet, while we monitored the ICMP traffic this time.

It is worth mentioning that ICMP packets do not have a port field in their headers. On the other hand, the CE router encompasses the NAPT44 function that assigns ports to the outgoing packets. As a result, MAP-T has developed a workaround for ICMP packets [7], where it replaces the ICMP packet's ID field with a port number (out of the assigned pool). That's how it filters the ICMP packets against the allocated ports in CE and BR.

As shown in Fig. 7, we managed to spoof the source IPv6 address of the outgoing traffic from CE toward the BR router and send a crafted ICMPv6 packet. This was made possible by another Python script that we wrote for such an attack (`icmpv6-spoofer.py`) [24]. The script also sniffs the communication channel between CE and BR, searches for ICMP traffic, grabs the ICMPv6 packet, which was sent originally by the IPv4 client, and saves the ICMP ID field as a variable. Eventually, the script crafts a new ICMPv6 packet with the same saved details (using the same ICMPv6 ID value) of the found ICMPv6 packet and then sends it accordingly to the BR router.

The attack went through with the below steps: Attacker ⇒ BR ⇒ IPv4 server ⇒ BR ⇒ CE ⇒ IPv4 client. In conclusion, the CE and BR routers processed a fake crafted ICMPv6 packet, while the IPv4 client received a reply to a packet that he hadn't sent.

This type of attack has a similar effect as the UDP spoofing, but the impact of ports assigning with UDP packets is clearer since every UDP packet includes a source port in its header, unlike ICMP packet, where ports are assigned as ID fields value in the ICMP layer of the packet.

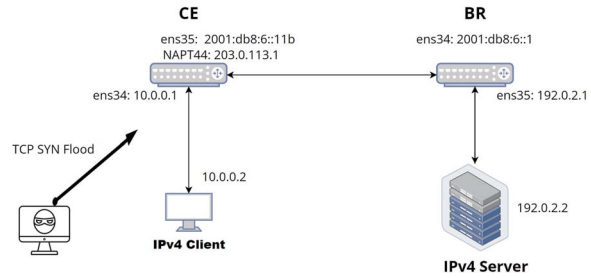

Figure 8. DoS attack using `hping3` package.

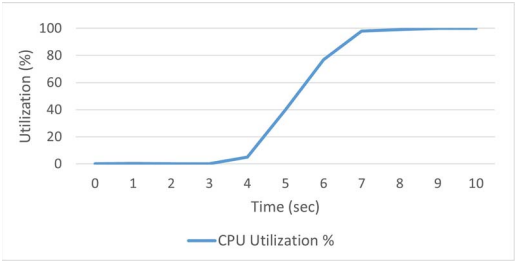

Figure 9. CPU utilization for CE machine.

6.3. DoS attack

We carried out a DoS attack to overwhelm the CE / BR machines and exhaust their commutation power. We achieved our goal by applying two methods (`hping3` package and Scapy crafting script):

6.3.1. *hping3 method*

As illustrated in Fig. 8, we used the `hping3` package, where we sent a flood of TCP SYN packets from the IPv4 client to the IPv4 server using the command as follows:

```
hping3 -S –flood -V -p 80 192.0.2.2
```

While the attack was up and running, we tried reaching the IPv4 server from the IPv4 client side, we recorded a 70% packet loss, which proved that our attack caused more than two-thirds of the legitimate packets to be drooped.

Furthermore, we monitored the CPU utilization of the CE machine before and after the attack. In Fig. 9, we show that the CPU utilization was very low until we started our attack in the third second.

We kept the attacking script running till the tenth second when the CPU was fully utilized.

6.3.2. *Packet crafting method*

In the second method for a DoS attack, we used our Scapy script (`tcp-sync-dos.py`) [24]. As illustrated in Fig. 10, the script sniffs the traffic between CE and BR machines looking for TCP traffic. When the attacker finds a TCP packet headed in this direction: CE ⇒ BR, it saves the source port as a variable, then crafts a similar TCP SYN packet with the same source port number and eventually sends the crafted packet again to the BR machine as if it came from the CE.

The attack went through with the below steps: Attacker ⇒ BR ⇒ IPv4 server ⇒ BR ⇒ CE ⇒ IPv4 client.

However, the script sends an extreme rate of TCP SYN packets (10 000 packets per second), which results in a DoS attack against the BR machine by consuming a large amount of its computation power. The rate of the packets can be modified in the script, assuming that the attacker's machine is able to perform such an attack in terms of its physical resources.

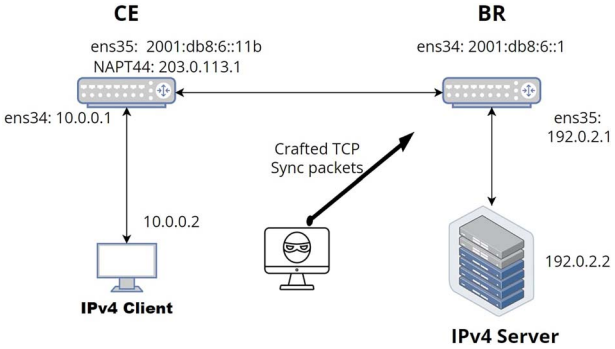

Figure 10. DoS attack using Scapy script (`tcp-sync-dos.py`) [24].

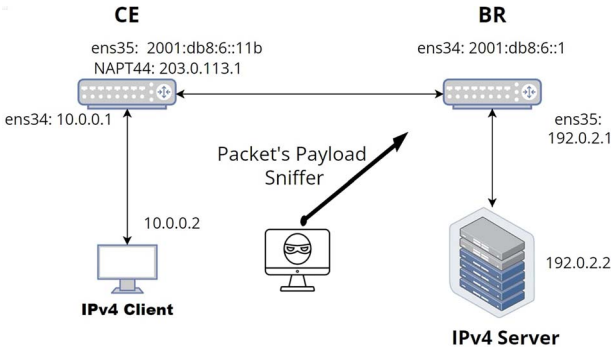

Figure 11. Information Disclosure attack using Scapy script.

The attack was relatively successful as it caused 11% packet loss when the IPv4 client tried to ping the IPv4 server during the attack period.

Moreover, we repeated the same attack with a different script (`tcp-sync-dos-random-sport.py`) [24], which is identical to the original script but it applies a random TCP source port number. We found out that all of the crafted TCP SYN packets were dropped by the BR machine because they hadn't used a port from the pre-defined ports range. As a result, we conclude that the source port number has to be extracted from the allocated port range for the attack to succeed.

The second DoS method can also be considered a spoofing attack, as it spoofs the source IPv6 address of the CE machine and communicates with the BR accordingly.

6.4. Information disclosure attack

As shown in Fig. 11, we performed an information disclosure attack using a specific Python script that leverages the Scapy module (`info-disclosure.py`) [24], where we monitored the communication channel between CE and BR machines. The attack was successful. Our script detected the traffic (TCP, UDP, or ICMP), then applied a specific Python function (depending on the detected traffic type) and printed out the payload of the TCP or UDP packet. In the case of ICMP traffic, the script printed out the content of the data field of the ICMPv6 layer.

6.5. MITM attack

To perform such an attack, we wrote a Python script (`mitm-attack.py`) [24], which does the following:

- sniffs the communication channel between CE and BR machines, looks for UDP traffic headed in this direction: CE ⇒ BR,
- when matched traffic is found, the attacker machine saves the packet details (IP addresses, MAC addresses, source port number, etc.),
- crafts a new packet with the same details, but with different UDP payload content,
- sends the newly crafted packet to the BR, claiming to be the CE machine.

The attack went through and the BR router processed the crafted packet (which has the wrong payload) as if it came from the CE machine, which constitutes MITM Attack.

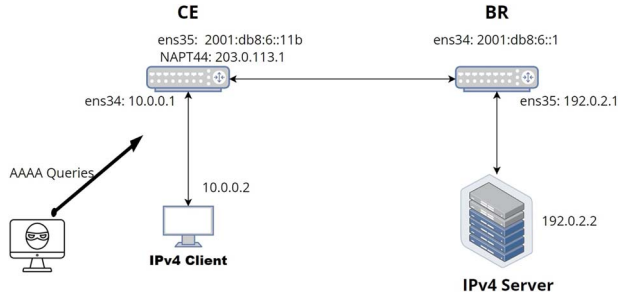

Figure 12. Source port exhaustion attack using AAAA DNS Queries.

6.6. Source port exhaustion

The port numbers are stored in 16-bit format, which means that the maximum number of UDP ports equals 65 536 [0-65,535]. This number applies to TCP and UDP ports. By design, every CE router has a built-in pool of assigned port numbers. However, it is recommended not to assign any of the reserved and well-known port numbers [0–1023]. In our testbed configuration, the CE router is pre-assigned with 2048 port numbers [55,296-57,343].

As a result, our attack's goal is to exhaust this pool for the CE router to stop functioning to the level that it will not be able to process any incoming packets anymore.

As shown in Fig. 12, we sent an excessive amount of 'AAAA" record queries from the attacker toward the IPv4 server. To implement such an attack, we have used a tool called `dns64perf++` [27]. It was designed to be used as a testing tool for measuring the performance of DNS64 servers. As it can send DNS queries at a high rate, we used it as an attacking tool. In the meantime, the assigned ports for CE in our testbed were 2048 ports [55,296-57,343]. After installing the `dns64perf++` tool, we applied the below command on the attacker's machine:

```
./dns64perf++v4 192.0.2.2 53 0.0.0.0/
5 60000 1 1 60000 \  400000 0.1
```

There are multiple parameters within the command, and we will provide a detailed explanation of each parameter as follows:

- 192.0.2.2: IPv4 address of the DNS server, here: destination IP address;
- 53: port number of the DNS server, here: destination port number;
- 0.0.0.0/5: the subnet for generating a label for the DNS queries, here: redundant;
- 60 000: total number of queries to send;
- 1: the burst size;
- 1: the number of threads;
- 60 000: number of ports per thread;
- 400 000: the delay between queries in nanoseconds, which means, we send 2500 queries/s;

No.	Time	Src. IP	Dst. IP	Protocol	Src. port	Dst. port
2053	1.213747489	2001:db8:ce:11b:0:cb00:7108:1b	64:ff9b::c000:202	DNS	55356	53
2054	1.214886657	2001:db8:ce:11b:0:cb00:7108:1b	64:ff9b::c000:202	DNS	55357	53
2055	1.228824597	2001:db8:ce:11b:0:cb00:7108:1b	64:ff9b::c000:202	DNS	56367	53
2064	29.998550310	2001:db8:ce:11b:0:cb00:7108:1b	64:ff9b::c000:202	DNS	57152	53

Figure 13. Wireshark capture on CE ens35.

- 0.1: the timeout (specifying the maximum waiting time for a response), here: redundant.

After sending an excessive amount of 'AAAA' queries (2500 per second) from the attacker toward the IPv4 server, we managed to exhaust the pool of source port numbers in less than 2 seconds. As a result, we stopped the CE router from further processing incoming packets for several seconds. Figure 13 shows the wire-shark capture on the CE router (ens35 interface), where it illustrates that in 1.22 seconds (the second column), we managed to exhaust the pool of allocated ports, where the packet flow stops. Only after the 30th-second (which is the default timeout for UDP connection), did the flow continue because the ports were re-assignable by then.

6.7. Mitigation methods

6.7.1. *ICMP and UDP source address spoofing mitigation*

The crafted packets are generated via Scapy in a sophisticated manner that makes them look almost identical to the original ones. Therefore, it is technically very hard for the BR and CE routers to realize that the received packets are IPv6 address packets with spoofed source addresses, especially when the crafted packets have identical IP and MAC addresses to the original ones.

As a result, a sophisticated tool could be deployed to act as an Intrusion Detection System (IDS) or an Intrusion Prevention System (IPS) to intercept malicious packets, filter them and eventually drop them. For instance, Snort [28], which is an open-source software package, that can be deployed and act as IDS and IPS as well.

Snort, with its deep inspection method, can detect the spoofed ICMPv6 packets by examining the packet headers and looking for anomalies that are indicative of spoofing. In particular, Snort can use the following methods to detect spoofed ICMPv6 packets:

- IP address matching: Snort can compare the source IP address of the ICMPv6 packet against the known IP address of the CE router. If the source IP address does not match, Snort can trigger an alert indicating that the packet is likely spoofed.
- TTL value checking: Snort can check the TTL value in the packet header to determine if the packet has traveled a realistic distance from its source. If the TTL value is too low or too high, Snort can infer that the packet has been spoofed.
- Protocol anomalies: Snort can look for other protocol-level anomalies in the packet header that suggest spoofing.

We have installed and configured Snort on the BR router to detect and prevent spoofing attacks by leveraging the Snort rule, which we inserted in a file we named 'icmp.rules', which has the following line configured:

```
alert icmp any any -> any any (msg:"Possible
Spoofed \  Address Detected"; ip6_saddr:!<CE-IPv6-
prefix>::/64; \  icmpv6_type:echo-request; sid:1;
rev:1;)
```

Snort looks at a packet that originated from a machine that resides outside the network, if such a packet has a source IPv6 address that belongs to the internal network of the BR ens34 interface, it will be tagged as a malicious packet. Below is the command that we used to run snort to monitor the traffic and detect the filtered ICMP packets:

```
snort -c $snort_path/etc/snort/snort.lua -R icmp.
rules \  -Q -daq afpacket -i "ens34:ens35"
```

However, the mitigation using Snort did not work in this scenario. The reason behind that is that Snort is not able to detect inside jobs, where the attacker is within the network.

For more information regarding Snort installation and configuration, please refer to our GitHub repository, where we installed and configured Snort on Debian 10 machine [29]. It is worth mentioning that such software packages might cause lower performance [30].

6.7.2. *DoS mitigation*

In general, DoS can be mitigated by having a list of IP addresses of potential malicious adversaries who have committed such acts before and blocking those IP addresses. However, some attackers have the ability to hide their own identities and spoof an innocent machine's IP address. Therefore, the rate-limiting [31] technique can be deployed on the CE router to control/limit the traffic of packets per second that passes through the Network Interface Card (NIC).

As a result, we wrote a shell script '`rate-limiting.sh`' [24], which we executed on the CE machine to set a severe rate limit of 10 packets/second (for testing purposes). It did prevent CPU exhaustion on the CE machine while under DoS attack.

The script is composed of two fundamental rules: the first permits the processing and forwarding of incoming TCP SYN packets that occur at a rate of 10 packets per second, whereas the second rule discards all TCP SYN packets:

```
/sbin/ip netns exec napt iptables -A FORWARD
-p tcp \   -syn -m limit -limit 10/s -j ACCEPT
/sbin/ip netns exec napt iptables -A FORWARD
-p tcp \   -syn -j DROP
```

6.7.3. *Information disclosure mitigation*

As presented in Fig. 11, we were able to expose the payload of the UDP / TCP packet. Therefore, we wrote a Python script to mitigate such an attacking scenario [24]. The script leverages a cryptographic Python library called 'Fernet', which provides a straightforward and secure way to encrypt and decrypt data [33]. It is specifically designed for symmetric encryption, which means the same key is used for both encryption and decryption processes.

We repeated the attack of Section 6.4, which sniffs the communication channel between the CE and the BR router and prints the payload content of the packet. However, this time we send a TCP packet with an encrypted payload from the IPv4 client toward the IPv4 server. As a result, the attacker was only able to see the encrypted value of the payload, which is useless in that regard.

The full Python script for the mitigation method can be found under the name of '`tcp-crafter-enc.py`' in our GitHub repository [24].

Table 5. Implemented attacks against MAP-T infrastructure

Attack name	Complexity of executing	Complexity of mitigation	Mitigation method	Severity
DoS	Easy	Difficult	Rate-Limiting	Critical
Man-in-the-Middle (MITM)	Average	Difficult	VPN / MFA [32]	Critical
Information Disclosure	Average	Average	Payload Encryption	High
ICMP Spoofing	Average	Difficult	IDS / IPS [28]	Medium
UDP Spoofing	Average	Difficult	IDS / IPS [28]	Medium
Source Port exhaustion [27]	Medium	Difficult	Traffic-Control	High

6.7.4. *MITM attack mitigation*

Unfortunately, the mitigation of MITM attacks is also dependent on installing an IDS/IPS tool such as SNORT, which was also discussed in the mitigation of ICMP/UDP source IP address Spoofing. Please refer to Section 6.7.1. Other methods can be applied to mitigate MITM attacks by enhancing the security of communication and endpoints, such as establishing a secure Virtual Private Network (VPN) connection between the CE and BR routers or enabling Multi-Factor Authentication (MFA).

MFA is an essential security measure that augments the login process by necessitating not only the usage of a username and password but also an additional form of verification. This supplementary verification can take various forms, such as the input of a Personal Identification Number (PIN) or the utilization of a unique code transmitted via Short Messaging Service (SMS) to a designated mobile device [32]. nochmal Personal Identification Number (PIN).

6.7.5. *UDP source port exhaustion mitigation*

To address this issue, we plan to add more ports. However, this method alone may not be enough to fully mitigate the attack, as the attacking script could still exhaust the new, larger pool of ports within seconds. In addition, adding more ports will not be a practical approach, because there will not be enough ports to allocate for other CEs.

Therefore, to drop DNS queries that have extremely high packet rates, we have utilized the Traffic Control (tc) command to shape the traffic on the ingress interface. Here is a snippet from the script that we run on the CE router to limit DNS queries to 10 per second per IP address and drop any queries that exceed that limit:

```
/sbin/ip netns exec napt tc qdisc add dev to_global\
 root handle 1: htb default 1

/sbin/ip netns exec napt tc class add dev to_global\
 parent 1: classid 1:1 htb rate 1gbit

/sbin/ip netns exec napt tc class add dev to_global\
parent 1:1 classid 1:10 htb rate 10kbit ceil 10kbit

/sbin/ip netns exec napt tc filter add dev to_global\
parent 1:0 protocol ip prio 2 u32 match ip protocol
17 \  0xff match ip dport 53 0xffff match u32 0 0 \
flowid 1:10 action drop
```

The script does the following:

- creates a hierarchical token bucket (HTB) qdisc on the 'to_global' interface with a default class (1:1) and a child class(1:10) for DNS traffic,
- limits the maximum bandwidth for the parent class to 1Gbps,
- limits the maximum bandwidth for the child class to 10 kilobits per second (htb rate 10kbit),
- finally, `tc filter` command applies a filter on the ingress interface to drop packets that exceed the configured limits.

As a result, after running the same attack of Section 6.6, the attacker was not able to exhaust the pool of allocated ports within 30 seconds (UDP connection timeout). However, we had to configure a radical rate limit of 10 packets/second. Therefore, it is a trade-off between protecting your infrastructure from DoS/source port exhaustion attacks and causing high latency and service disruption.

The full mitigation shell script can be found under the name of '`traffic-controller.sh`' in our GitHub repository [24].

After implementing our attacking scenarios and presenting the corresponding mitigation methods, we summarize them in Table 5, which includes the applied/proposed attack mitigation techniques. Additionally, we classify the severity of the attacks based on the complexity of execution and mitigation.

7. DISCUSSION

As for the security threats that we uncovered in our attacking scenarios, our findings were not programming errors in the Jool MAP-T implementation, but general aspects of the technology. Therefore, we have published all of the applied scripts on GitHub and decided to disclose the vulnerabilities in this journal paper rather than notifying the developers.

This research work has been a continuous effort since our previous publications, where we have surveyed and tested several IPv4-as-a-Services (IPv4aas) technologies such as 464XLAT, DS-Lite, lw4o6 and finally MAP-T. We have found that some attacks are viable at all of these technologies such as the DoS attack that aims to exhaust some resources of the technology. It can be either filling the state table with useless connections or it can also be the wasting of its pool of source port numbers and/or public IP addresses. Therefore, all those technologies have similar vulnerabilities, however, at different locations (Internet Service Provider (ISP) side or customer side).

Some technologies are labeled as stateless, but they are indeed stateful somewhere within their infrastructure. For example, lw4o6, MAP-T and MAP-E are labeled as stateless technologies, but they are stateful at the customer edge side. As a result, DoS attacks on the customer side are more likely in this case, where the effect of such attacks will endanger only the given customer.

On the other hand, a DoS attack against a technology that is stateful at its ISP side such as DS-Lite or 464XLAT will affect all subscribers and have wider damage and downtime in the network.

8. PLANS FOR FUTURE RESEARCH

Since our current paper focused on MAP-T, we plan to conduct a comprehensive security analysis that covers MAP-E and its security analysis and summarizes the potential threats that the

MAP-E infrastructure faces. Eventually, we plan to build a testbed for MAP-E using VPP open-source software [23]. As a result, we will be able to compare MAP-E and MAP-T implementation testbeds and their attacking scenarios.

9. CONCLUSION

In this paper, we highlighted the importance of the MAP-T IPv6 transition technology, especially its translation method, and showcased its practicality. We have shown that the Jool software can be applied as a solution for deploying MAP-T and providing customers with IPv4aas, particularly for ISPs operating with an IPv6-only infrastructure. However, it is important to acknowledge that this solution exhibits several security vulnerabilities. Through the utilization of the STRIDE threat modeling technique, we identified various potential attacks, including DoS, IP address spoofing, information disclosure, and ARP Cache Poisoning, which pose risks to the MAP-T infrastructure, especially the CE and BR routers.

Furthermore, our testing platform was subjected to different types of attacks, such as IP address spoofing of ICMP/ TCP/UDP packets, TCP SYN DoS and source port exhaustion. While some of these attacks were successfully mitigated, others were not. Specifically, the Jool-based CE and BR routers were found to be inadequate in preventing or detecting a crafted packet with a spoofed source IP address, necessitating the deployment of an additional layer of firewall or IDS/IPS software alongside the Jool.

LIST OF ACRONYMS

464XLAT, Combination of Stateful and Stateless Translation; ARP, Address Resolution Protocol; BMR, Basic mapping rule; BR, Border Relay; CAPEC, Common Attack Patterns Enumeration and Classification; CE, Customer Edge; CLAT, Customer-side translator; CPU, Central Processing Unit; DFD, Data-Flow Diagram; DMR, Default Mapping Rule; DoS, Denial of Service; EA, Embedded Address; FMR, Forwarding Mapping Rule; FoS, Failure of Service; IDS, Intrusion Detection System; ISP, Internet Service Provider; ISPs, Internet Service Providers; IPS, Intrusion Prevention System; IPv4aas, IPv4-as-a-Services; MAC, Media Access Control; MAP-E, Mapping of Address and Port with Encapsulation; MAP-T, Mapping of Address and Port using Translation; MFA, Multi-Factor Authentication; MITM, man-in-the-middle attack; NAPT44, Network Address and Port Translation; NIC, Network Interface Card; PIN, Personal Identification Number; PLAT, Provider-side translator; PSID, Port Set Identifier; SMS, Short Messaging Service; STRIDE, Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege; tc, Traffic Control; TCP, Transmission Control Protocol; TTL, Time to Live; UDP, User Datagram Protocol; VPN, Virtual Private Network

ACKNOWLEDGEMENT

The authors thank Professor Youki Kadobayashi, Laboratory for Cyber Resilience, Nara Institute of Science and Technology, Japan, and Bertalan Kovács, Budapest University of Technology and Economics, for reading and commenting on the manuscript.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

FUNDING

This work was not supported by any organization.

DATA AVAILABILITY

The data underlying this article are available in [MAP-T-builder] at [https://github.com/ameen-mcmxc/MAP-T-builder], and can be accessed with [accessing the GitHub database publicly, no special account is needed].

REFERENCES

1. Lencse G, Kadobayashi Y. Methodology for the identification of potential security issues of different IPv6 transition technologies: threat analysis of DNS64 and stateful NAT64. *Comput Secur* 2018;**77**:397–411.
2. Al-Azzawi A. Towards the security analysis of the five most prominent IPv4aaS technologies. *Acta Technica Jaurinensis* 2020;**13**:85–98.
3. RFC 6877. *464XLAT: Combination of Stateful and Stateless Translation*. Wilmington, DE: IETF, 2013.
4. Al-Azzawi A, Lencse G. Identification of the possible security issues of the 464XLAT IPv6 transition technology. *Infocommun J* 2021;**13**:10–8.
5. Al-Azzawi A, Lencse G. Analysis of the security challenges facing the DS-Lite IPv6 transition technology. *Electronics* 2023;**12**:2335.
6. Al-Azzawi A, Lencse G. Lightweight 4over6 test-bed for security analysis. *Infocommun J* 2023;**15**:30–41.
7. RFC 7599. *Mapping of Address and Port using Translation (MAP-T)*. Wilmington, DE: IETF, 2015.
8. RFC 7597. *Mapping of Address and Port with Encapsulation (MAP-E)*. Wilmington, DE: IETF, 2015.
9. Lencse G, Nagy N. Towards the scalability comparison of the Jool implementation of the 464XLAT and of the MAP-T IPv4aaS technologies. *Int J Commun Syst* 2022;**35**:e5354.
10. ITESM. Jool, Open Source IPv4/IPv6 Translator. [online]. Available at: https://www.jool.mx/en/index.html.
11. RFC 7703. *Experience with Testing of Mapping of Address and Port Using Translation (MAP-T)*. Wilmington, DE: IETF, 2015.
12. Center C. Open Source MAP Implementation. [online]. Available at: https://github.com/cernet/MAP.
13. Al-hamadani A, Lencse G. Towards implementing a software tester for benchmarking MAP-T devices. *Infocommun J* 2022;**14**:45–54.
14. Georgescu M, Hazeyama H, Okuda T. et al. The STRIDE towards IPv6: a comprehensive threat model for IPv6 transition technologies. In: *2nd International Conference on Information Systems Security and Privacy*, Rome, Italy, February. p. 243–254. Setúbal: SCITEPRESS, 2016.
15. Georgescu M, Hazeyama H, Kadobayashi Y. et al. Empirical analysis of IPv6 transition technologies using the IPv6 network evaluation testbed. *EAI Endorsed Trans Ind Netw Syst* 2015;**2**:e1–1.
16. Georgescu M, Hazeyama H, Okuda T. et al. Benchmarking the load scalability of IPv6 transition technologies: a black-box analysis. In: *IEEE Symposium on Computers and Communication (ISCC)*, Larnaca, Cyprus, July. p. 329–334. Piscataway, NJ: IEEE, 2015.
17. of Homeland Security UD. Common Attack Pattern Enumeration and Classification (CAPEC). [online]. Available at: https://capec.mitre.org/.
18. Shostack A. Threat Modeling: Designing for Security. Hoboken, NJ: John Wiley & Sons, 2014.

19. Kristiyanto Y, Ernastuti E. Analysis of Deauthentication attack on IEEE 802.11 connectivity based on IoT technology using external penetration test. *CommIT J* 2020;**14**:45–51.
20. Abad CL, Bonilla RI. An analysis on the schemes for detecting and preventing ARP cache poisoning attacks. In: *27th International Conference on Distributed Computing Systems Workshops (ICDCSW'07)*, Toronto, ON, Canada, June, p. 60–60. Piscataway, NJ: IEEE, 2007.
21. Biondi P. Scapy, Packet Manipulation Program. [online]. Available at: https://scapy.net/index.
22. Gupta BB, Dahiya A, Upneja C. et al. A comprehensive survey on DDoS attacks and recent Defense mechanisms. In: *Handbook of Research on Intrusion Detection Systems*. Hershey, PA: IGI Global, 2020;**13**:186–218.
23. FDio 'VPP Software'. [online]. Available at: https://github.com/FDio/vpp.
24. Al-Azzawi A. MAP-T Jool Testbed Installation [online]. available: https://github.com/ameen-mcmxc/MAP-T-builder.
25. Making a Synthesis Emulation in IoT ERA Possible, Starbed5 Project. StarBED5 Project website. [online]. Available at: https://starbed.nict.go.jp/en/equipment/.
26. Hat R. Ansible, Open Source IT Automation Tool. [online]. Available at: https://www.ansible.com/.
27. Bakai D. DNS64perf++ Measurement Tool. [online]. Available at: https://github.com/bakaid/dns64perfpp.
28. Sourcefire I. Snort 3, Open Source Intrusion Detection and Prevention System. [online]. Available at: https://snort.org/.
29. Al-Azzawi A. Snort Installation and Configuration using Ansible Automation Software. [online]. Available at: https://github.com/ameen-mcmxc/SNORT-Automation.
30. Park W, Ahn S. Performance comparison and detection analysis in Snort and Suricata environment. *Wirel Pers Commun* 2017;**94**:241–52.
31. Noormohammadpour M, Raghavendra CS. Datacenter traffic control: understanding techniques and Tradeoffs. *IEEE Commun Surv Tutor* 2018;**20**:1492–525.
32. Ometov A, Bezzateev S, Mäkitalo N. et al. Multi-factor authentication: a survey. *Cryptography* 2018;**2**:1.
33. Jain A, De P, Enhancing database security for facial recognition using Fernet encryption approach. In: *International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India, December, p. 748–753. Piscataway, NJ: IEEE, 2021.