Towards the Identification of the Possible Security Issues of the 464XLAT IPv6 Transition Technology

Ameen Al-Azzawi, Gábor Lencse
Department of Networked Systems and Services
Budapest University of Technology and Economics
Budapest, Hungary
Email: alazzawi@hit.bme.hu, lencse@hit.bme.hu

***Abstract*—This paper focuses on one of the most prominent IPv6 transition technologies named 464XLAT. The aim is to analyze the security threats that this technology might face. We carry out the threat analysis by applying the STRIDE method, which stands for Spoofing, Tampering, Repudiation, Information Disclosure and Elevation of Privilege. STRIDE uses the DFD (Data Flow Diagram) as a basis for its analysis. We have analyzed the structure of 464XLAT then applied the STRIDE method on it and came up with interesting results regarding its security vulnerabilities and we have narrowed down the most common attacks that might have an effect on its deployment.**

***Keywords*—464XLAT; DNS; IPv4aaS; IPv6; STRIDE; Translation.**

I. Introduction

In [1], we have overviewed the five most prominent IPv6 transition technologies for sustaining IPv4 service, while using only IPv6 in the Internet Service Provider access and core networks. One possible solution could be the combination of DNS64 [2] NAT64 [3]. However, this technology has its own drawbacks in terms of not supporting IPv4 literals and IPv4 only applications like Skype, Netflix, etc., [4]. This issue of DNS64+NAT64 has been solved by 464XLAT [5] with its double translation mechanism. However, its application may involve various security vulnerabilities. Therefore, it is essential to analyze the security threats that might affect this promising technology. The main focus of this paper is to highlight security threats facing the network infrastructure as a result of deploying 464XLAT within the network topology.

According to [5], 464XLAT in general is very quick to deploy and has minimal IPv4 resource requirements & maximum IPv4 efficiency. Moreover, 464XLAT employs traffic engineering and capacity planning without the indirection or obfuscation of a tunnel [5].

In Section II, we discuss the operation of 464XLAT and its structure, section III is about operation of the STRIDE method, its elements and how it works, section IV is about 464XLAT security revealed by applying STRIDE on it, while in section V, we mention some previous publications regarding 464XLAT / NAT64 security threats and in section VI, we summarize and conclude the whole value and results that this paper came up with, where we prove the efficiency of the STRIDE approach and it also shows how vulnerable some parts of 464XLAT are, and eventually categorize the main threats that 464XLAT is liable to.

II. The Operation of 464XLAT

The main structure of 464XLAT, as shown in Fig. 1, is divided in two sides; CLAT & PLAT.

A. CLAT (customer-side translator)

CLAT algorithmically translates 1:1 private IPv4 addresses to global IPv6 addresses and vice versa [5]. It acts as IPv6 router, DNS proxy and DHCP server for local client as well.

Normally, CLAT must know its own prefix and PLAT side prefix in order to use it as destination for its outgoing packets [5].

B. PLAT (provider-side translator)

It translates N:1 global IPv6 addresses with the previously set CLAT prefix to public IPv4 addresses and vice versa [5], it actually implements a stateful NAT64 gateway as described in RFC 6146 [3].

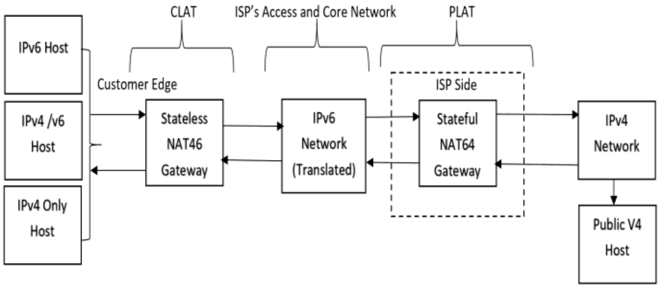

Fig. 1. Overview of 464XLAT Architecture

We give an easy introduction to understand to the operation of 464XLAT by Fig. 2. The client in the bottom left hand side corner of the figure (using private IPv4 address 192.168.1.2) wants to connect to the server in the top left had side corner (using public IPv4 address 198.51.100.1).

The prefix at CLAT side is 2001:8db:aaaa::/96, whereas the prefix at PLAT side is 2001:8db:1234::/96.

CLAT translates the IPv4 packet into an IPv6 packet, in which the source address will be 2001:db8:aaaa::192.168.1.2, and the destination address will be 2001:db8:1234::198.51.100.1.

At the PLAT side, the 2001:db8:1234::/96 prefix is discovered in the destination address, and an IPv4 packet is built using the embedded 198.51.100.1 IPv4 address as destination address, and the source IPv4 address is chosen from the pool of 192.0.2.1-192.0.2.100 (this time it happened to be 192.0.2.1). Source port is also replaced, when needed, and the connection is registered into the state table of the NAT64 translator to be able to perform the stateful translation in the reverse direction, too. (Please refer to RFC 6146 [3] for further details of the stateful NAT64 translation.)

```
+----------------------------+
|        IPv4 server         |
|       [198.51.100.1]       |          IP packet header
+----------------------------+   +--------------------------------+
              ^                  | Destination IP address         |
              |                  | [198.51.100.1]                 |
              |                  | Source IP address              |
              |                  | [192.0.2.1]                    |
+----------------------------+   +--------------------------------+
|            PLAT            |                   ^
| IPv4 pool address          |                   |
| [192.0.2.1 - 192.0.2.100]  |                   |
| PLAT-side XLATE IPv6 prefix|                   |
| [2001:db8:1234::/96]       |                   |
+----------------------------+   +--------------------------------+
              ^                  | Destination IP address         |
              |                  | [2001:db8:1234::198.51.100.1]  |
              |                  | Source IP address              |
              |                  | [2001:db8:aaaa::192.168.1.2]   |
+----------------------------+   +--------------------------------+
|            CLAT            |                   ^
| PLAT-side XLATE IPv6 prefix|                   |
| [2001:db8:1234::/96]       |                   |
| CLAT-side XLATE IPv6 prefix|                   |
| [2001:db8:aaaa::/96]       |                   |
+----------------------------+   +--------------------------------+
              ^                  | Destination IP address         |
              |                  | [198.51.100.1]                 |
              |                  | Source IP address              |
              |                  | [192.168.1.2]                  |
+----------------------------+   +--------------------------------+
|        IPv4 client         |
|      [192.168.1.2/24]      |
+----------------------------+
```

Fig. 2. 464XLAT Packet Processing [5]

Besides double translation, there are two other possible scenarios. If both the client and the server have IPv6 addresses, then there is no translation at all, but native IPv6 is used. If the client has an IPv6 address, but the server has only and IPv4 address, then there are two possible modes of operation:

- If DNS64 is configured, then the DNS64 server returns an IPv4-embedded IPv6 address, and only a single translation happens at the PLAT. (This is the DNS64 + NAT64 solution.).
- If no DNS64 is configured, then the client uses IPv4 and double translation happens as described above.

III. The Operation of STRIDE

STRIDE stands for Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service and Elevation of Privilege [7]. These are general threats that any network device/node might be susceptible to.

A. *Spoofing:* an attacker tries to hide his real identity so he claims to be someone else by changing his real source IP address in order to gain an unauthorized access to some sensitive information or service [7].

B. *Tampering:* the process of changing the content of data flow on its way to the destination, for example, the attacker might alter the packet destination to a malicious server [7]. It can also be achieved using the SDR (Software-Defined-Radio) [7], which made the need to buy an expensive hardware tampering equipment obsolete. SDR unit can be used to tamper with the wireless protocol.

C. *Repudiation:* it is the claim of not doing an act, while he actually did, like ATM money withdrawal or DNS resolution request [8]. This threat often appears on the business layer (above network layer in TCP/IP or above application layer such as HTTP/HTML).

D. *Information Disclosure:* an attacker gets sensitive information, which could be used in various ways, e.g. it might help him in hacking, like TTL value of the packet or learns who’s talking to whom by monitoring DNS traffic [8].

E. *Denial of Service:* The attacker can flood a system with illegitimate requests to prevent it from servicing legitimate ones, for example, it can flood a DNS server with huge number of useless queries to prevent legitimate queries from getting a response [8].

F. *Elevation of Privilege:* bypassing the authority matrix of specific organization, like getting root permission on a specific server [7].

The STRIDE method uses the DFD (Data Flow Diagram) of the investigated system in order to examine the critical areas within the system, so it comes up with total security analysis using the four types of elements of the DFD (Data Flows, Data Stores, Processes and Interactors).

Data flow models usually applied on network & architecture systems rather than software products, but they can be applied on both [7]. STRIDE has different approaches regarding threat models:

- Assets-centered threat model: anything the attacker wants to access, control or damage. According to [9], assets-centered threat model is being conducted using 4 approaches: DREAD, Trike, OCTAVE and PASTA. For instance, OCTAVE, which stands for Operationally Threat Asset and Vulnerability Evaluation, is a robust approach but its rather complicated, it takes considerable time to learn and get familiar with its process. Furthermore, its documentation is voluminous [9].

TABLE I. VULNERABILITIES OF DIFFERENT DFD ELEMENTS [8]

DFD Element	Spoofing	Tampering	Repudiation	Information Disclosure	Denial of Service	Elevation of Privilege
Data Flows		✓		✓	✓	
Data Stores		✓		✓	✓	
Processes	✓	✓	✓	✓	✓	✓
Interactors	✓		✓			

- Attacker-centered threat model: it is based on knowing the attacker, his motivations and skills. It is useful but hard to implement [7].
- Software-centric threat model: focuses on software being built and the deployed systems, it's the best approach for threat modeling [7], because it supposes that software developers are the best people to understand the software they are developing, which makes the software an ideal starting point to trigger the threat modeling process.

In general, the best models are diagrams that help participants understand the software and find threats against it. Each element of the DFD has its own security threats as explained in Table I. It means each element is susceptible to some threats while not susceptible to others [7].

IV. SECURITY ISSUES OF 464XLAT

We presented DFD of 464XLAT in a previous paper [10]. Nevertheless, we made some slight changes on the DFD and after applying the STRIDE method on the DFD diagram of 464XLAT in Fig.3, some security threats are visible at the points (1-11), which represent the threat possibilities within the DFD diagram. In this section, we carefully examine all the elements of the DFD for all possible threats & attacks in details. (Please see the summary of vulnerabilities in Table II.)

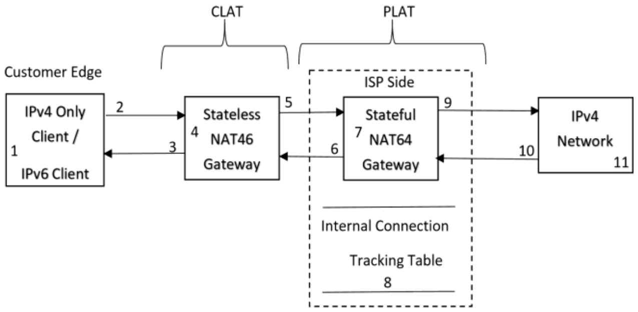

Fig. 3. DFD for the threat Analysis of 464XLAT

A. IPV4 / IPV6 Client

1) Spoofing:

- Spoofing the client's IP address could be used go get unauthorized access to the NAT46 gateway.
- DoS (Denial of Service) attack against the CLAT might be possible due to spoofed client's IP address by flooding the CLAT with high traffic.

2) Repudiation: the client might deny the request he made in the first place.

B. Data flow from IPv4 only client to NAT46

1) Tampering: it can be used as an attack against the domain name or changing the IP address of packet destination, which might be used to direct the packet towards fraudulent server, this kind of attack is also called FoS (Failure of Service) because it prevents the real client from receiving an answer to its real query [8].

2) Information Disclosure: an attacker might be interested in knowing the browsing habits of the requester, and the packet itself might contain some sensitive information sent by the client himself [8].

3) Denial of Service: flooding the gateway with unwanted requests to prevent the real query from getting an answer.

C. Data flow from NAT46 gateway to the client

1) Tampering: an attack against the client, for example sending misleading information at application level or breaking the connecting sending a RST at TCP level.

2) Information Disclosure: an attacker getting access to sensitive data.

3) Denial of Service: sending high number of forged replies to the client to prevent if form processing the genuine ones.

D. NAT46 Gateway (CLAT)

1) Spoofing: in this case, spoofing means unauthorized user controls the gateway and translate the private IPv4 to the wrong IPv6 and send the packet to different destination.

2) Tampering: an attacker tampered with the data within the gateway itself by which might result in e.g. returning the wrong IPv6 address [8].

3) Repudiation: after spoofing the CLAT, an attacker might deny sending a packet that was actually sent by the CLAT himself while hiding his own identity. Logging is the key here, if the database administrator is not fully trusted, then a system in another privilege domain has to be installed.

4) Information Disclosure: an attacker might make use of the browsing data and queries made by the requester in order to hack the main requester later on.

5) Denial of Service: it could be by an attacker spoofing an IP of legitimate user flooding the CLAT with huge number of requests, see section IV.B.3.

6) Elevation of Privilege: it happens when an attacker gain access to a service he shouldn't be getting in the first place. However, it mainly happens due to inside job [8] and the attacker might gain the right of admin or root to whatever he likes later on.

E. Data flow from NAT46 to NAT64 gateway

1) Tampering: the packet destination IP might be altered while it's on its way to NAT64 gateway.

2) Information Disclosure: see section IV.B.2.

3) Denial of Service: after spoofing the NAT46, attacker might send numerous useless packets to the NAT64 gateway.

F. Data flow from NAT64 to NAT46 gateway

1) Tampering: see section IV.C.1.

2) Information Disclosure: it is possible that an attacker might access the packet details on its way back to NAT46 gateway and extract sensitive information out of it.

3) Denial of Service: flooding the NAT46 gateway with unwanted packets to prevent it from translating the genuine traffic.

G. NAT64 Gateway (PLAT)

1) Spoofing: an attacker might take control (spoof) the gateway and do many malicious activities with it, see section IV.D.1.

2) Tampering: an attacker might change the content of packet details withing the gateway, see section IV.D.2.

3) Repudiation: see section IV.D.3.

4) Information Disclosure: see section IV.D.4.

5) Denial of Service: DoS attack might come in a way that affect the NAT64 Gateway (PLAT), such as Exhaustion of source port and public IPv4 address pool, which is an issue since the gateway uses 63K[1] number of source ports per public IPv4 address. An enhanced algorithm presented by [11] helps in tackling this issue in details.

6) Elevation of Privilege: one of the elevation problems is called buffer overflow attack [12], which could happen if a device like NAT64 getting inputs from both sides and that might affect its memory storage units.

H. Internal connection tracking table

Potential attackers have no direct access to it, they can influence its content in indirect ways only.

1) Denial of Service: The attacker may initiate fake connections (either using his real IPv6 address or fake ones) and thus achieve the insertion of fake entries into the connection tracking table. The high number of fake entries may slow down the operation of the NAT64 gateway or even prevent legitimate users from establishing further connections, when the table is full. If PLAT applies a connection limit per source IPv6 address, then the attacker may exhaust the available number of connections for legitimate users by spoofing there IPv6 addresses, when initiating fake connections spoofing their IPv6 addresses.

I. Data flow from PLAT to IPv4 Server

1) Tampering: attacker might change the source IP address of the packet so the IPv4 server will not know, who sent the packet in the first place.

2) Information Disclosure: see Section IV.B.2

3) Denial of Service: an attacker might spoof the IP and flood the IPv4 server with plenty of undesired requests.

J. IPV4 server / IPv4 network

1) Spoofing: Source IP address might be spoofed, see section IV.A.1.

2) Repudiation: denying of sending a request is viable in this case, *see section IV.A.2.*

K. Data flow from IPV4 server / IPv4 network to PLAT

1) Tampering: the attacker might send TCP-RST packets to erase the mapped entries within the NAT64 gateway.

2) Information Disclosure: it is possible that an attacker might access the packet details on its way back to NAT64 gateway and extract sensitive information out of it, like TTL value or browsing habits [8].

3) Denial of Service: flooding the *NAT64* gateway with unwanted requests to prevent it from translating the real traffic.

L. Summary of the results

To summarize the attacks or the vulnerabilities within 464XLAT structure, we concluded the following threats:

A. Spoofing of NAT46 or NAT64 gateways results in altering packets destination or returning the wrong IP address to the requester.

B. DoS: denying access of a legitimate user to his authorized traffic and obstructing the function of NAT46 & NAT64 gateways.

C. FoS: preventing the real client from receiving an answer to its real query.

D. Leaking of confidential information like IP address, TTL value and browsing habits.

E. Tampering with NAT64 tracking table: loosing of mapped entries.

F. Privileges level altering: getting root privilege will increase the inside job attack very often.

G. Buffer overflow attack in case of NAT64, which affects the storage (connection tracking table) entries and might erase them accidently.

[1] Similarly to NAT devices, NAT64 gateways usually use source port numbers from the range of 1024 – 65536.

V. Related Work

Very few papers have been published regarding our topic. However, [13] has focused on the IPv6 security issues as far as cellular networks concern and it came up with different categories of possible attacks. They demonstrated three different DoS attacks on NAT64 block targeting features that only exist in IPv6 cellular networks:

A. *NAT overflow attack:* According to [13], most of service providers tend to drop the source address of a spoofed packet and replace it with a public IPv4 address. Therefore, a host can send & receive packets using single private IPv4 address assigned by NAT.

   As a result, the maximum of external mapping for single targeted service is 65,535. Meanwhile, in IPv6 cellular networks, a device can utilize 2^{64} IPv6 addresses. So, if a device creates mapping on NAT64 using all the 2^{64} IPv6 addresses, the result will be 65,535 * 2^{64} mappings, which can lead to overload for NAT64 [13]. It also showed that NAT64 gateway will stop the mapping process for any incoming request after 1500 entries (depending on the preset value) within its tracking table (if the requester is sending from the same IP address targeting the same service) and sends back TCP-RST packet back to the requester as response for the TCP-SYN packet. However, this policy of NAT64 can be exploited as DoS attack [13].

B. *NAT wiping attack*: The targeted victim in this case is the mapping entry itself. NAT64 uses the N:1 mapping criterion. If an adversary targets the external IPv4 of NAT64 gateway, N hosts are sharing the same external IPv4 address will be liable to DoS attack. The adversary will send malicious TCP-RST packets to wipe out the target mappings within the NAT64. As a result, the mapped users to the very same external IPv4 address will be denied access to their service.
   To do so, the attacker needs to know the TCP 5-tuple of the targeted service (Protocol, Destination IP address, port number, External IP address of NAT64 and External port number of NAT64).

C. NAT Bricking attack: it's type of DoS attack which also exploits the N:1 mapping algorithm adopted by NAT64. Basically, the adversary can send huge number of requests using the external IPv4 address(es) of the NAT64 gateway [13]. However, big vendors (google, YouTube, etc.) have IP blocking approach if it exceeds specific number of requests per minute. Nevertheless, [13] has done an experiment to target Google scholar [14] website, which is an IPv4 based site. So, the IPv6 cellular host sends 150 requests per minute to trigger CAPTCHA request. Every time CAPTCHA request emerges, adversary source IP address is being changed by turning the airplane mode on and off, this process was repeated 1000 times. Finally, the NAT bricking attack was able to trigger CAPTCHA request for a total of 631 external IPv4 from Google Scholar, including one of the victim's external IP address [13].

Moreover, [15] has explained that the majority of the transition technologies use some form of NAT, NAT44, NAT64, NAT46, etc. and how it is a myth that NAT is putting the user inside this secured box of protective shield from the outside attackers, the sequence of communications below explains how vulnerable the NAT client could be:

1- Attacker attracts the victim towards specific website.
2- Victim clicks on the malicious URL and enters the page.
3- The page has a hidden form connecting to http:// attacker.com:6667 (IRC port).
4- The victim submits the form without his consent.
5- An HTTP connection is created to the (fake) IRC server.
6- The form as well has hidden value which sends: "OPEN DCC CHAT PORT" .
7- Router sees an "IRC connection" then open a port back through the NAT.
8- The attacker now has an open path to the network.

The very same process could have been applied using FTP NAT helper if not IRC.
According to [15], todays preferred transition technologies are 6rd, DS-Lite and 464XLAT, while risk Mitigation Strategies can be summarized as follow:

1- Minimizing the need for SP-NAT (Service-Provider NAT).
2- The more IPv6 established sessions, the less you rely on SP-NAT and all the security issues associated with that.
3- Search for a transition plan that uses native IPv6 such as 464XLAT & DS-Lite.

VI. Plans for future research

Currently we are working on building a test bed (an isolated environment) to check if the most important 464XLAT implementations actually have the discovered, theoretically existing potential vulnerabilities.

Our next step will be to seek mitigation for the most serious vulnerabilities.

We consider the performance of the different 464XLAT implementations important also from security point of view, because in some cases "high performance can be a kind of mitigation of DoS attacks" [8], and some of our team members deal with the performance analysis of IPv6 transition technologies.

VII. Conclusion

We have presented a threat analysis for one of the most prominent IPv6 transition technologies (464XLAT) using the STRIDE method in order to find its potential vulnerabilities.

STRIDE approach of dividing the 464XLAT structure into sperate elements and analyze them one by one proved to be effective and easy to navigate. Moreover, 464XLAT structure showed that it has a lot of threats & attacks possibilities all over its infrastructure. One of the main possible attacks is DoS attack and tampering with NAT64 internal connection tracking table. We have found out that both sides of 464XLT (CLAT & PLAT) have potential security vulnerabilities.

For the next step and future research, we will be working on eliminating or minimizing the effect of those threats one by one.

TABLE II. SUMMARY OF 464XLAT THREATS

DFD Element	***Threat***	***Possible attacks***
1	Spoofing & Repudiation	DoS attack against the CLAT
2, 3	Tampering, Information Disclosure and Denial of Service	FoS, collecting unauthorized information, DoS
4	All STRIDE Elements	FoS, DoS and unauthorized access,
5, 6	Tampering, Information Disclosure and Denial of Service	FoS, collecting unauthorized information, DoS
7	All STRIDE Elements	FoS, DoS and unauthorized access,
8	Only indirect attacks	Tampering with Connection Tracking Table; DOS attack (exhaustion of connection tracking table, slowing down look up speed)
9, 10	Tampering, Information Disclosure and Denial of Service	FoS, collecting unauthorized information, DoS
11	Spoofing & Repudiation	DoS attack against the PLAT

REFERENCES

[1] A. Al-Azzawi, "Towards the Security Analysis of the Five Most Prominent IPv4aaS Technologies", Acta Technica Jaurinensis, in press, DOI: 10.14513/actatechjaur.v13.n2.530

[2] M. Bagnulo, A Sullivan, P. Matthews and I. Beijnum, “DNS64: DNS extensions for network address translation from IPv6 clients to IPv4 servers”, IETF RFC 6147, 2011, doi:10.17487/RFC6147.

[3] M. Bagnulo, P. Matthews and I. Beijnum, “Stateful NAT64: Network address and protocol translation from IPv6 clients to IPv4 servers”, IETF RFC 6146, 2011, doi:10.17487/RFC6146.

[4] S. Répás, T. Hajas, G. Lencse, “Application compatibility of the NAT64 IPv6 transition technology,” in Proc. 37th International Conference on Telecommunications and Signal Processing, Berlin, 2014, pp. 49–55, doi:10.1109/TSP.2015.7296383.

[5] M. Mawatari, M. Kawashima, C. Byrne, “464XLAT: Combination of stateful and stateless translation, IETF RFC 6877 (2013).

[6] M. Bagnulo, P. Matthews, I. Beijnum, “Stateful NAT64: Network address and protocol translation from IPv6 clients to IPv4 servers”, IETF RFC 6146 , 2011, doi:10.17487/RFC6146.

[7] A. Shostack, “Threat modeling: Designing for security”, 1st Edition, Wiley, Indiana, 2014.

[8] G. Lencse and Y. Kadobayashi, “Methodology for the identification of potential security issues of different IPv6 transition technologies: Threat analysis of DNS64 and stateful NAT64”, Computers & Security, vol. 77, no.1, pp. 397-411, 2018, doi: 10.1016/j.cose.2018.04.012.

[9] L. O. Nweke and S. D. Wolthusen, “A Review of asset-centric threat modelling approaches” International Journal of Advanced Computer Science and Applications (IJACSA), 11 (2), 2020. http://dx.doi.org/10.14569/IJACSA.2020.0110201

[10] A. Al-Azzawi, “Plans for the security analysis of IPv4aaS technologies”, in Proc.14th International Symposium on Applied Informatics and Related Areas, University of Obuda, Székesfehérvár, Hungary, 2019, pp. 101–105.

[11] M. S. Ferdous, F. Chowdhury, J. C. Acharjee, “An extended algorithm to enhance the performance of the current NAPT”, in Proc. International Conference on Information and Communication Technology, Dhaka, Bangladesh, 2007, pp. 315–318, doi:10.1109/ICICT.2007.375401.

[12] A. D. Keromytis, “Buffer overflow attacks”, in H. C. A. van Tilborg, S. Jajodia (Eds.), Encyclopedia of Cryptography and Security, Springer, Boston, 2011, pp. 174–177, doi:10.1007/ 978-1-4419-5906-5-502.

[13] Hong, H., Choi, H., Kim, D., Kim, H., Hong, B., Noh, J., & Kim, Y. (2017), “When cellular networks met IPv6: Security problems of middleboxes in IPv6 cellular networks”, IEEE European Symposium on Security and Privacy (EuroS&P), 2017, doi:10.1109/eurosp.2017.34.

[14] Google, Google Scholar. https://scholar.google.com.

[15] Itu.int. (2016). “IPv6 Security Mechanisms”, [online] Available at: https://www.itu.int/en/ITU-D/Regional-Presence/AsiaPacific/Documents/s11-ipv6-securingtransitionmechanisms.pdf [Accessed 5 Feb. 2020].