SLAAC Attack Detection Mechanism

Nazrool Omar[*], Selvakumar Manickam[**]

[*, **]National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia, Penang, Malaysia

Email: [*]no13_com055@student.usm.my, [**]selva@nav6.usm.my

ORCID ID: 0000-0003-0778-2343, 0000-0003-4378-1954

Research Paper Received: 07.09.2019 Revised: 22.12.2019 Accepted: 02.02.2020

Abstract- Attacks against Neighbour Discovery Protocol (NDP) is a major security issue in Internet Protocol Version 6 (IPv6). It demands security expert attention because the availability of attacking toolkits has amplified the risk of NDP attack in IPv6 network. Stateless Address Autoconfiguration (SLAAC) attack is a type of NDP attack exploited by attacker to launch MiTM and DoS attack. Researcher have proposed IPSec, Secure NDP (SeND), SAVI, RA-Guard, Trust-ND and other methods but have not been implemented widely due to enormous resources requirement for cryptographic process and alteration of original NDP. This paper proposes a detection mechanism named SADetection to detect SLAAC attack. SADetection incorporated enhanced ongoing packet verification and authentication mechanism. SADetection has been implemented in testbed and has detected three (3) variants of SLAAC attack which are attack using ICMPv6 packet, using fragment packet and using packet with extension header. SADetection has been found to be lightweight, platform-independence and interoperable. SADetection does not alter original NDP thus resource practical to SLAAC attack.

Keywords: IPv6, Network Discovery Protocol, Testbed, Network Security, SLAAC Attack

1. Introduction

Internet Protocol version 6 (IPv6) is proposed to eliminate inefficiencies of IPv4. The RFC8200 - Internet Protocol, Version 6 (IPv6) Specification has proposed new addressing capabilities; header format simplification; support for extensions and options; flow labeling capability; and authentication and privacy capabilities [1]. Many new protocols have been introduced in IPv6 and one of them is Neighbor Discovery Protocol (NDP).

As specified by RFC 4861, NDP provides link layer address resolution feature like Address Resolution Protocol (ARP) in IPv4. NDP is also required by Stateless Address Auto-configuration (SLAAC) to configure self-generated IP address, Duplicate Address Detection (DAD) to avoid IP duplication, Neighbor Unreachable Detection (NUD) to detect neighbor unreachability and Redirect to redirect network traffic [2]. NDP has remarkably simplified network deployment and improved performance of IPv6. Unfortunately, NDP vulnerabilities have been discovered that lead to misuse of routing headers, ICMPv6 and fragmentation.

NDP vulnerabilities have been exploited to attack features in IPv6 including Stateless Address Autoconfiguration (SLAAC) process [5]. SLAAC inherits NDP vulnerabilities that exposes IPv6 host to information leak, denial of service and credential stealing. SLAAC attack can incapacitate IPv6 network if not properly mitigated. Therefore, the main motivation of this paper is to complement and strengthen the security of SLAAC by proposing enhanced detection mechanism named SADetection. Existing detection mechanism can only examine RA message in normal ICMPv6 packet header. Hidden RA message in fragment packet or in packet with extension header remained undetected due to assumption that it is impractical to inspect the whole packet to search for RA message. Even though RFC6980 [13] proposes that NDP packet must not be fragmented, it is still not been fully implemented in Microsoft and Linux operating system. Existing security safeguards such as SEUI-64, Trust-ND, SSAS, SAVA, TRDP and SeND are complicated to

implement and not available by default in operating system. Most of the safeguards alter currently implemented NDP by adding new processes, messages and options. Cryptography configuration and process require huge computing resources and meticulous management. It could trigger Denial of Service (DoS) attack to NDP if not properly managed. SADetection offers simplicity, scalability and manageability because it does not modify original NDP.

2. Overview of SLAAC

SLAAC is the ability of IPv6 nodes to create and configure IPv6 address [3]. It has simplified IP address assignment and configuration in IPv6. There are two (2) main processes in SLAAC which are Router Solicitation (RS) and Router Advertisement (RA). When an IPv6 interface is activated, it sends out RS message to request RA message from router. As reply, router advertises multicast RA message that can be used by all multicast-enabled hosts in the network.

SLAAC is performed by acquiring 64-bits network prefix from RA message and appending the prefix to 64 bits interface identifier (IID) to form 128-bits IPv6 address. Figure 1 depicts SLAAC process. IP address created using SLAAC is stateless, but it can be used to communicate with global IPv6 network.

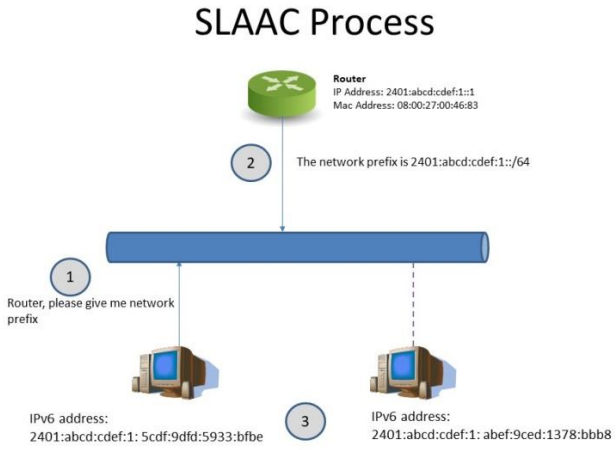


Fig. 1. SLAAC process

3. Related Works

Previous researchers has proposed many security safeguards to secure SLAAC using authentication and cryptography, deploying security monitoring tools and applying closed network policy. Some of the security safeguards are developed as prevention mechanism and some as detection mechanism. The proposed safeguards can protect from attacks and exploitation but suffer some limitations. Shah, Anbar, Al-Ani, and Al-Ani [14] have proposed an entropy-based technique combined with the adaptive threshold algorithm to detect RA flooding attack. The technique selects entropy from the features of NDP packet and dynamically adapting threshold to build flooding detection rules. The proposed technique may detect DoS RA flooding attack but deriving entropy solely based on IP address randomness as attack indicator can be unreliable and increases false alarm possibility.

Abdullah [15] proposes SEUI-64 bits addressing strategy that produces unpredictable Interface Identifier (IID). Instead of using host's own MAC address as identifier, SEUI -64 bits algorithm uses router's first three (3) bytes of MAC address for first connected host and applies permutation to the bytes for subsequent connected hosts in the network. To make up 64 bits IID, next 2 bytes is assigned by network administrator and last 3 bytes is randomly generated by interface. This strategy offers unpredictable IID but can only protect from reconnaissance attack. It also can be

problematic to keep track correct order of subsequent connected hosts in big and scalable network.

Massamba and Cheikh [16] secure SLAAC in home IPv6 network by controlling router decision to relay or reject RA based on a trusted table. The trusted table contains information of trusted RA packet learnt through type-length-value (TLV) and NDP message shared among routers. Routers can only relay or sent RA that exists in the trusted table. This solution cannot protect network from SLAAC attack launched from internal network because information of trusted RA is gathered only from Internet Service Provider (ISP) and domain router.

Praptodiyono et al. [7] has proposed Trust-ND to secure SLAAC by introducing Trust Options. The Trust Options are appended to NDP message so that receiver can perform trust calculation using Message Authentication Data from the Trust Options. Trust-ND can prevent SLAAC attack launched using ICMPv6 but not attack launched using hidden RA. Trust-ND amends original NDP by introducing new message options and process for IPv6 host and router. There are concern on high processing resource and security of SHA-1 used in the algorithm vulnerable to hash collision attack.

Snort IPv6 Plugin is proposed by Schutte [18] is signature-based detection system to detect NDP attack. If there is new RA message, the plugin will verify the message using pre-defined NDP attack signature. IPv6 Snort Plugin can detect SLAAC attack launched using ICMPv6 packet but cannot detect hidden RA message in fragment packet and packet with extension header. It is resource intensive because must be implemented in Snort IDS. SeND employs Authorization Delegation Discovery (ADD) [11] to mitigate SLAAC security vulnerability. SeND's ADD only allows authorized router to advertise RA. Attacker who is not authorized will not be trusted by hosts. SeND ADD has the capability to prevent SLAAC attack. Unfortunately, SeND's ADD has modified original NDP extensively, requires many configurations for every new host and compatibility issue with operating system.

Source Address Validation Improvement (SAVI) [9] enforces source IP address validation by router or authentication device before packet is forwarded to destination. SAVI prevents SLAAC attack by implementing local subnet source address validation. However, SAVI implementation consumes high processing resource because it validates every packet in the network. The requirement to configure all legitimate source IP addresses or prefixes will complicate network administration when new router needs to be introduced.

RA Guard is detection mechanism implemented in network switch [22]. It only allows RA to be sent from a dedicated network switch port. RA Guard only detects and prevents SLAAC attack launched using ICMPv6. Unfortunately, RA Guard can be evaded if attacker conceals RA message in fragment packet or in packet with extension header. RA Guard does not consume high computing resource because it only processes RA packet received from dedicated switch port. However, if new legitimate router needs to be introduced, RA Guard and network switch must be reconfigured.

4. Proposed Detection Mechanism

This research proposes a detection mechanism, named SLAAC attack Detection Mechanism (SADetection) to protect IPv6 network from SLAAC attack. SADetection is designed to fulfill three (3) main requirements which are; reduce complexity, prevent exploitation of packet with extension header; and provide distributed protection at network level. SADetection is incorporated with enhanced ongoing packet verification and authentication algorithm and optimized with anomaly profiling and detection rulesets to detect attack launched SLAAC attack efficiently.

There are four (4) modules in the SADetection which are generic verification (GV) handler, RA

handler, Fragment Handler and Extension Handler as showed in Fig. 2. Attack detection is started by GV handler. It verifies each packet regardless of type to check source IP address, source MAC address and network prefix of inspected packet. If the packet authenticity is not yet verified, the verification will be continued by specific module whether RA handler, Fragment handler or Extension handler according to type of the inspected packet. Database tables are used to facilitate data packet storage and retrieval. There are Packet Table, RA Table, Fragment Table, Extension Table, Authentication Table, Log Table and Signature Table. Packets that have been verified will be deleted from Packet Table, RA Table, Fragment Table and Extension Table in two and half (2.5) hour interval time to prevent high storage utilization.

4.1 Generic Verification (GV) Handler

First verification is done by Generic Verification (GV) Handler. It looks up Authentication Table to check if the inspected packet comes from legitimate machine. If source IP address and source MAC address of the inspected packet exist in Authentication Table, it considered legitimate, and no alert is raised. If it is not the case, Log Table will be checked. Source MAC address of the inspected packet will be compared with MAC address in the Log Table. If the source MAC address matches any MAC address in Log Table, the packet will be considered as attack. When attack is detected earlier by GV Handler, it can cut down unnecessary verification by other handlers. Figure 3 shows flow chart of GV Handler.

Attack packet may not be detected by GV Handler if attacker's IP address and MAC address are new and not yet verified by SADetection. In this case, the packet will be forwarded to next verification process based the type of the packet. If the packet is ICMPv6 RA packet, it will be forwarded to RA handler. If the packet is Fragment packet, it will be forwarded to Fragment handler and to Extension Handler if the packet is packet with extension header.

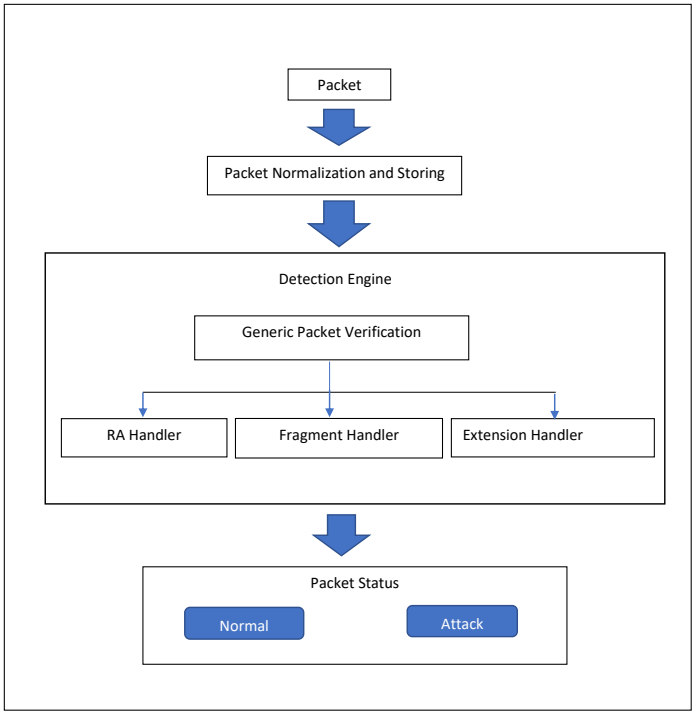


Fig. 2. Detection mechanism of SADetection

4.2 RA Handler

RA Handler continues verification process for ICMPv6 RA packet if GV Handler cannot determine the authenticity of the packet. There are two (2) functions in RA Handler which are Verify_MACAddress and Verify_All. The Verify_MACAddress function is invoked if source IP address and network prefix of the inspected packet exist in Authentication Table but the source MAC address is not matched. Meanwhile, the Verify_All function will be invoked if source IP address, source MAC address and network prefix of the inspected packet do not exist at all in Authentication Table. Figure 4, 5 and 6 show flow charts of RA Handler.

The Verify_MACAddress function compares timestamp of the inspected RA first packet with timestamp of legitimate RA last packet that has same IP address and network prefix. Comparison is done by analyzing RA packets from Packet Table. If the inspected RA first packet is found before or concurrently with legitimate RA packet last packet, then the inspected RA is considered as attack because it trying to spoof legitimate RA. Alert will be raised, and information of attack packet will be updated in Log table. If the inspected RA first packet is found after legitimate RA last packet, then the inspected RA is considered legitimate. This situation happens during replacement of router's port when RA with new MAC address will start after existing RA is suppressed. In this case, Authentication Table will be updated with information of new legitimate RA. The Verify_All function does the same timestamp comparison between the inspected RA first packet and legitimate RA last packet. It will retrieve information of legitimate RA from Authentication Table. If there are more than one record, it will loop verification process for each record in Authentication Table to compare the timestamp. If the inspected RA first packet is found before or concurrent with legitimate RA last packet, then the inspected RA packet is considered an attack and will be logged in Log table. If it is found after legitimate RA last packet, the number of inspected RA packets is counted within a time frame. The time frame is time from the inspected RA packet first appeared in the network until current timestamp. Unit of measurement for the time frame duration is in seconds. The calculated time frame duration will be divided by number of packets exist in the time frame to measure interval time of inspected RA packet between one another.

The time interval will be compared with pre-determined threshold. The pre-determined threshold value is 3 second. RFC 4861 [2], regarding specification of NDP message has stated the minimum time allowed between sending unsolicited RA advertisement must be no less than 3 seconds. If the time interval of inspected RA packet is shorter than 3 seconds, then it is considered as attack. If the inspected RA packet passes the pre-determined threshold value, it will be verified by attack signature. Attack signature defines attack packet as packet that contains Hop Limit equals to 255, Current Hop Limit equals to 255, Default Router Preference sets to High, Router Lifetime equals to 2048 and Retransmit Timer equals to 0.

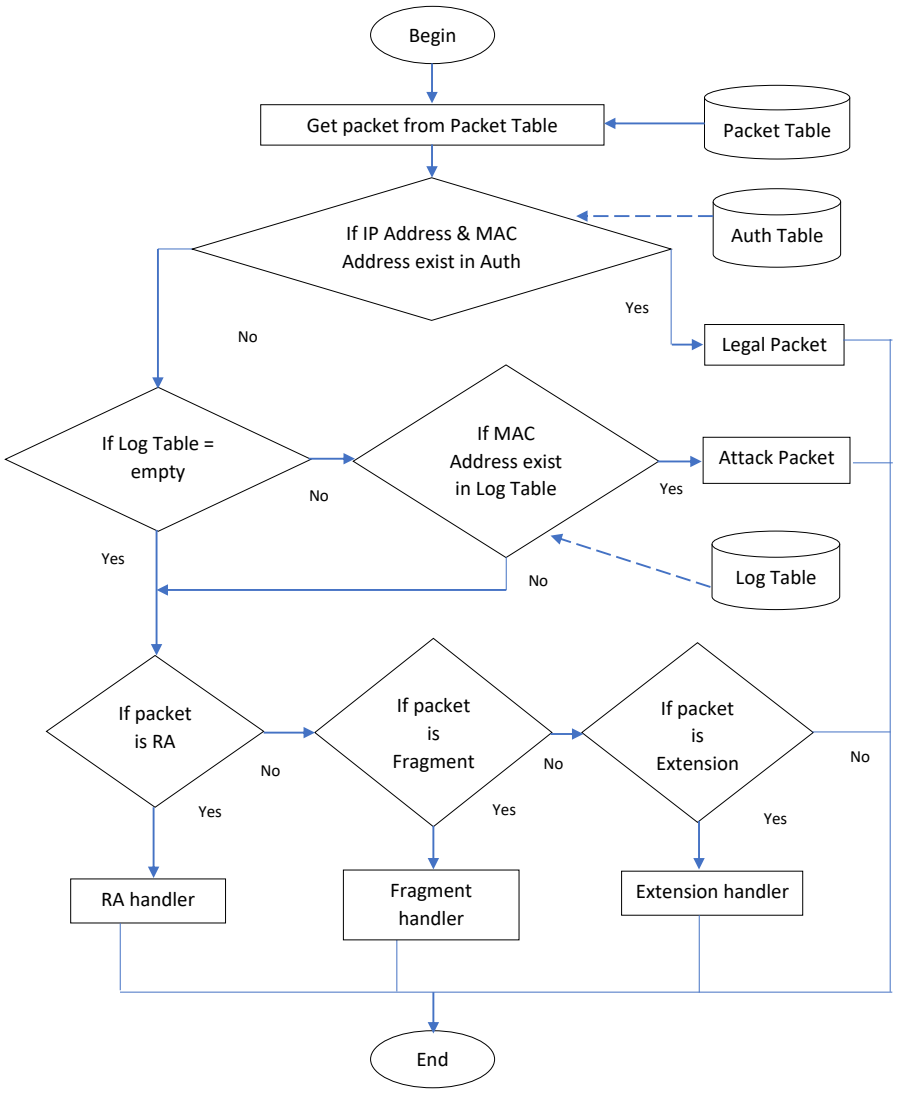


Fig. 3. Flow chart of GV Handler

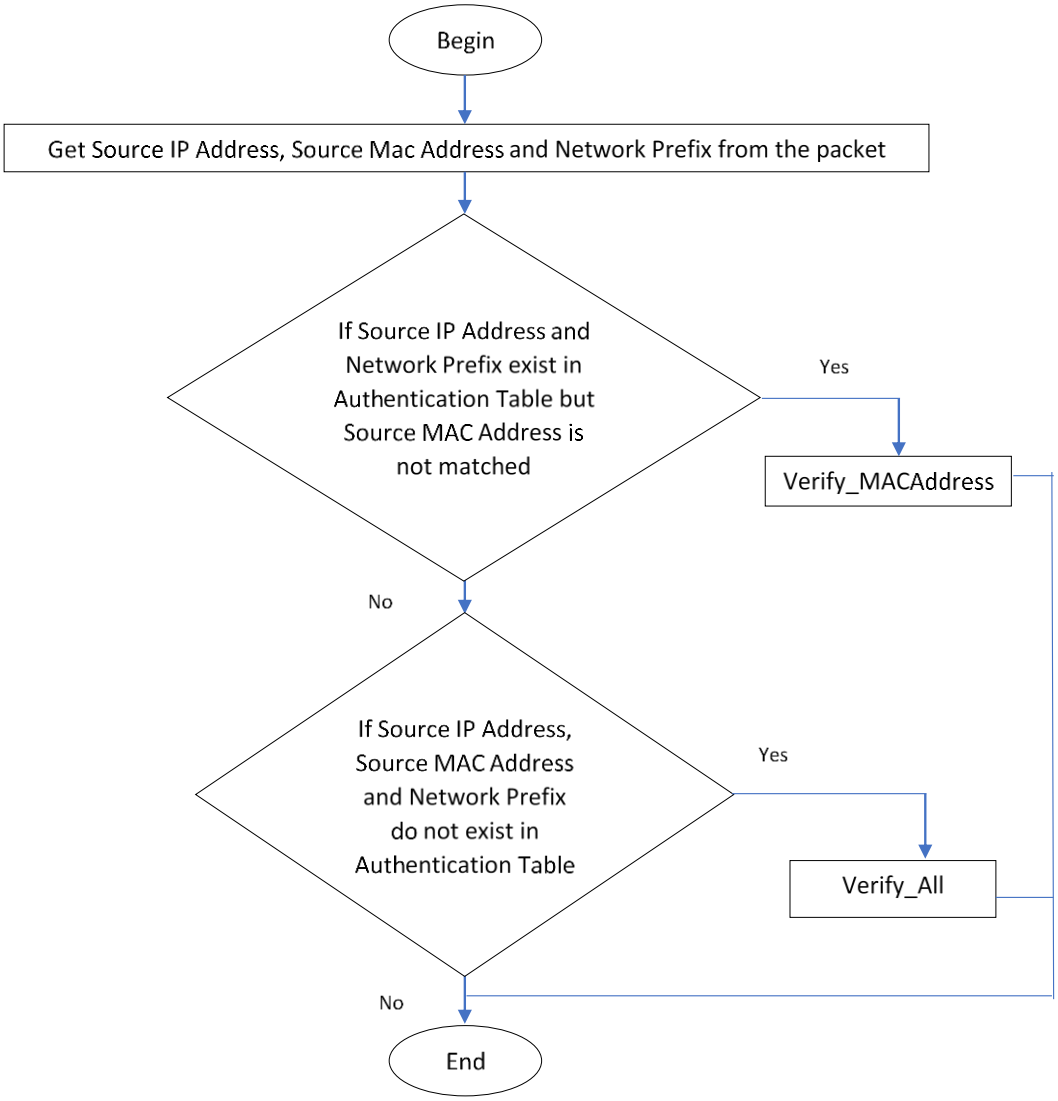


Fig. 4. Flow chart of RA Handler

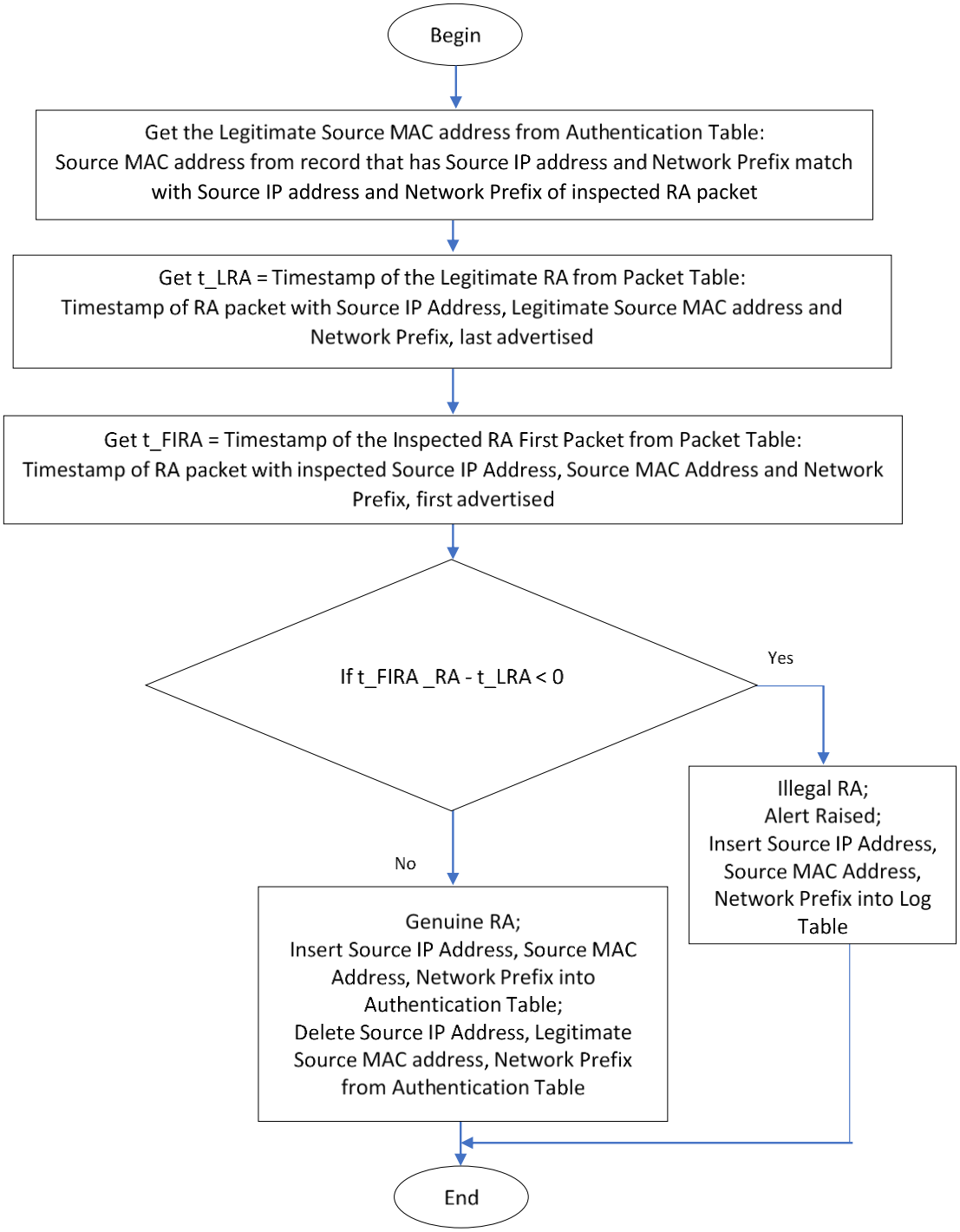


Fig. 5. Flow chart of RA Handler – Verify_MACAddress

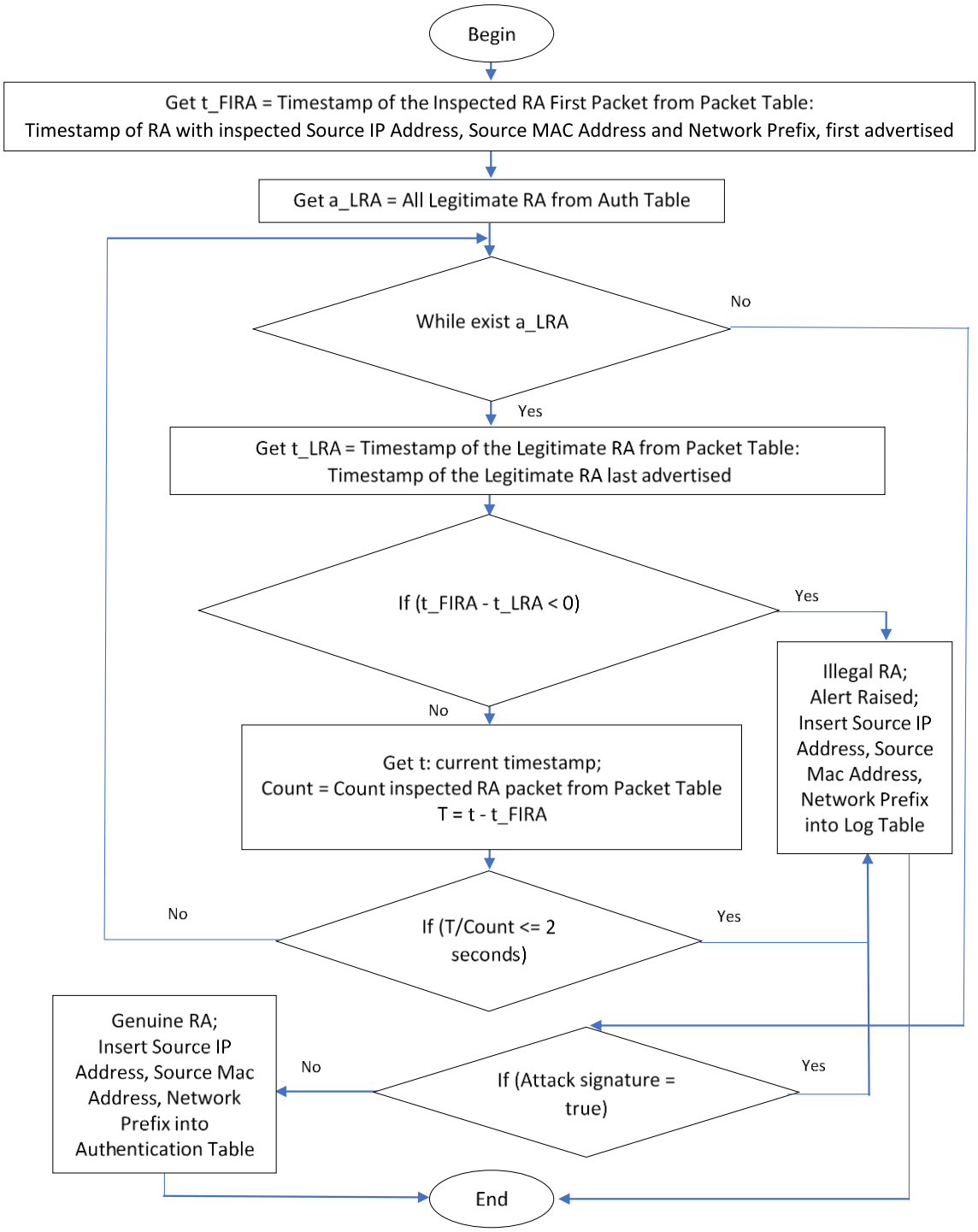


Fig. 6. Flow chart of RA Handler – Verify_All

4.3 Fragment Handler

Fragment Handler verifies first fragmented packet of every fragment packets. First fragmented packet is identified by value 0 in 'fragment offset' field. Fragment Handler detects hidden RA message by locating ICMPv6 type 134 header in the packet. Figure 7 shows flow chart of Fragment Handler. Fragment Handler will check 'next header' field in fragment header. If 'next header' field indicates value 59 which means "no next header", the fragment packet is considered normal and verification process will exit. If 'next header' field indicates value 58 which means next header is ICMPv6 header, the type of the ICMPv6 will be checked. If the type is 134 then the packet is considered as attack because it is an attempt to hide RA message in the fragment packet.

If 'next header' field indicates value 44, it means next extension header is another fragment packet. It is considered as attack because fragment packet that extends another fragment extension is anomalous packet behavior. IPv6 does not specifies double fragmentation which means a fragmented packet must not be fragmented to smaller packet again. If 'next header' field indicates value 60, it means next header is Destination Options header. Fragment Handler needs to scan through packet payload. Although entire packet payload is scanned, Fragment Handler only checks the 'next header' field that present in extension header or upper layer header exist in the packet.

Fragment Handler utilizes 'payload length' field to determine starting point for scanning. The 'payload length' is the length of the packet which counted starting from first extension or upper-layer header until end of packet including fragment header itself. The fragment header must be excluded from scanning. Thus, scanning kick off point is computed by subtracting 71 which is the fragment header length, from 'payload length' value. Starting from the kick off point, Fragment Handler check 'next header' of every subsequent extension and upper-layer header until "no next header" value is found, or scanning have reached end of packet.

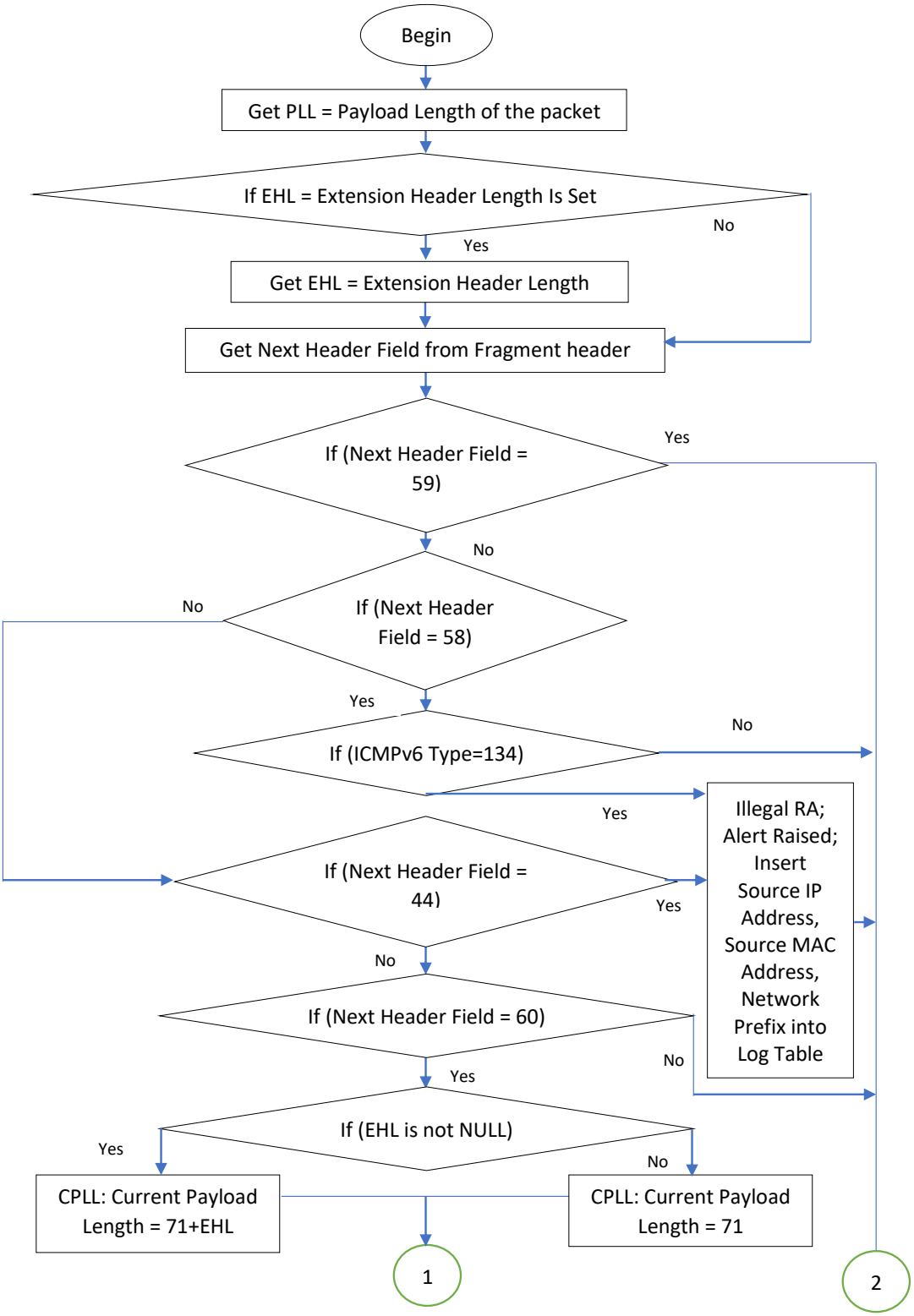
4.4 Extension Handler

Extension Handler verifies packet with Hop-By-Hop Options extension header, packet with Routing extension header and packet with Destination Options extension header. It checks 'next header' field of the extension header to determine the packet's status. Figure 8 shows flow chart of Extension Handler.

If the value of 'next header' is 59 which means "no next header", then the packet is legitimate and will exit verification process. If the value of 'next header' is 58 which means ICMPv6 header, Extension Handler will check the type of the ICMPv6 message. The packet is considered as attack if the ICMPv6 type is 134.

If the value of 'next header' is 44 which means fragment packet, Extension Handler will check Fragment Offset value. If the value is 0, Extension handler will pass the packet to Fragment Handler for fragment verification. If Fragment Offset value is other than 0, the packet will be not verified and will be discarded.

If the 'next header' field indicates value other than 58, 59 and 44, Extension Handler module will recursively call itself to verify subsequent extension until "no next header" or ICMPv6 type 134 is found.


Begin
Get PLL = Payload Length of the packet
If EHL = Extension Header Length Is Set
No
Yes
Get EHL = Extension Header Length
Get Next Header Field from Fragment header
If (Next Header Field = 59)
Yes
No
If (Next Header Field = 58)
No
Yes
If (ICMPv6 Type=134)
No
Yes
Illegal RA; Alert Raised; Insert Source IP Address, Source MAC Address, Network Prefix into Log Table
If (Next Header Field = 44)
Yes
No
If (Next Header Field = 60)
No
Yes
If (EHL is not NULL)
Yes
No
CPLL: Current Payload Length = 71+EHL
CPLL: Current Payload Length = 71
1
2

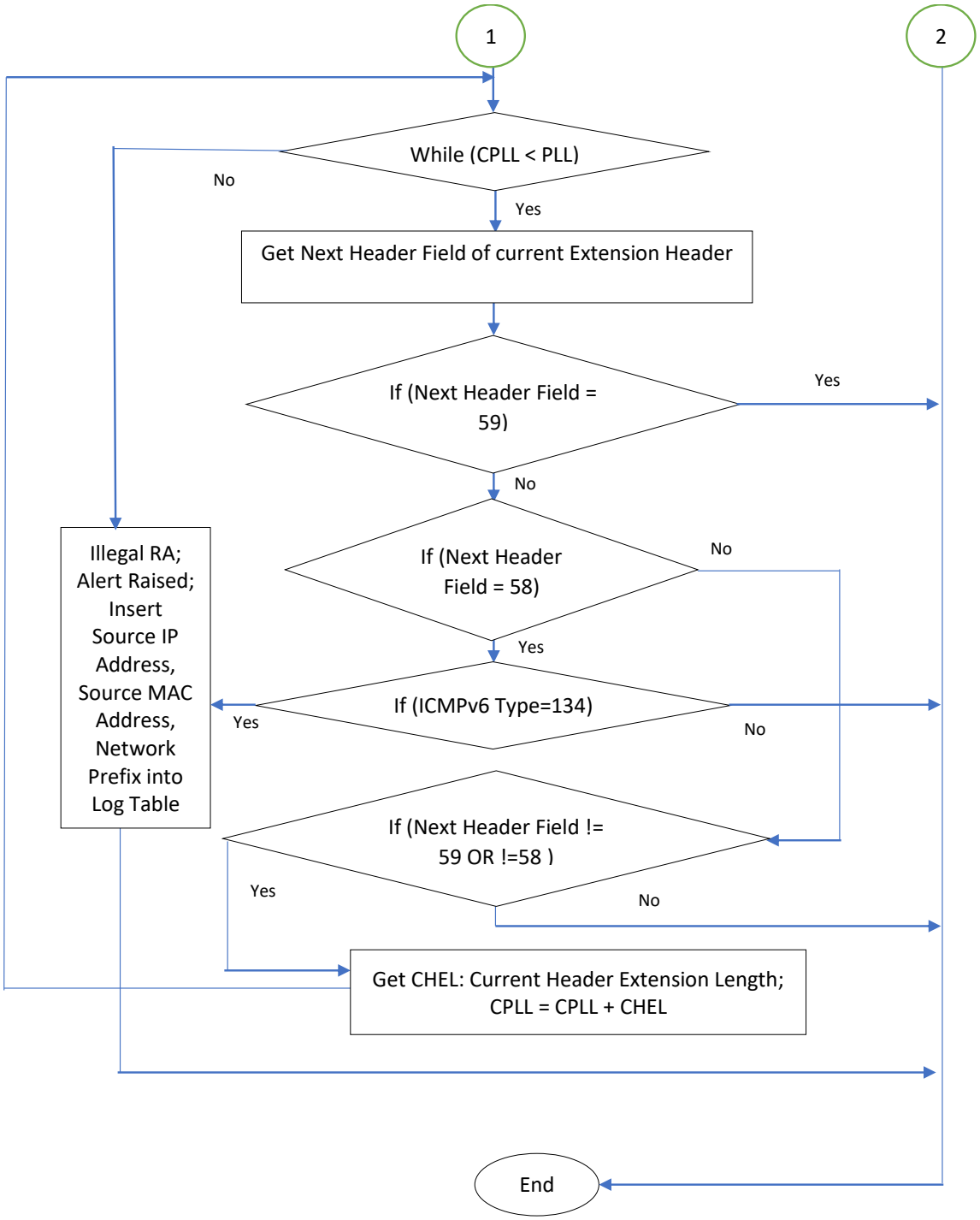


Fig. 7. Flow chart of Fragment Handler

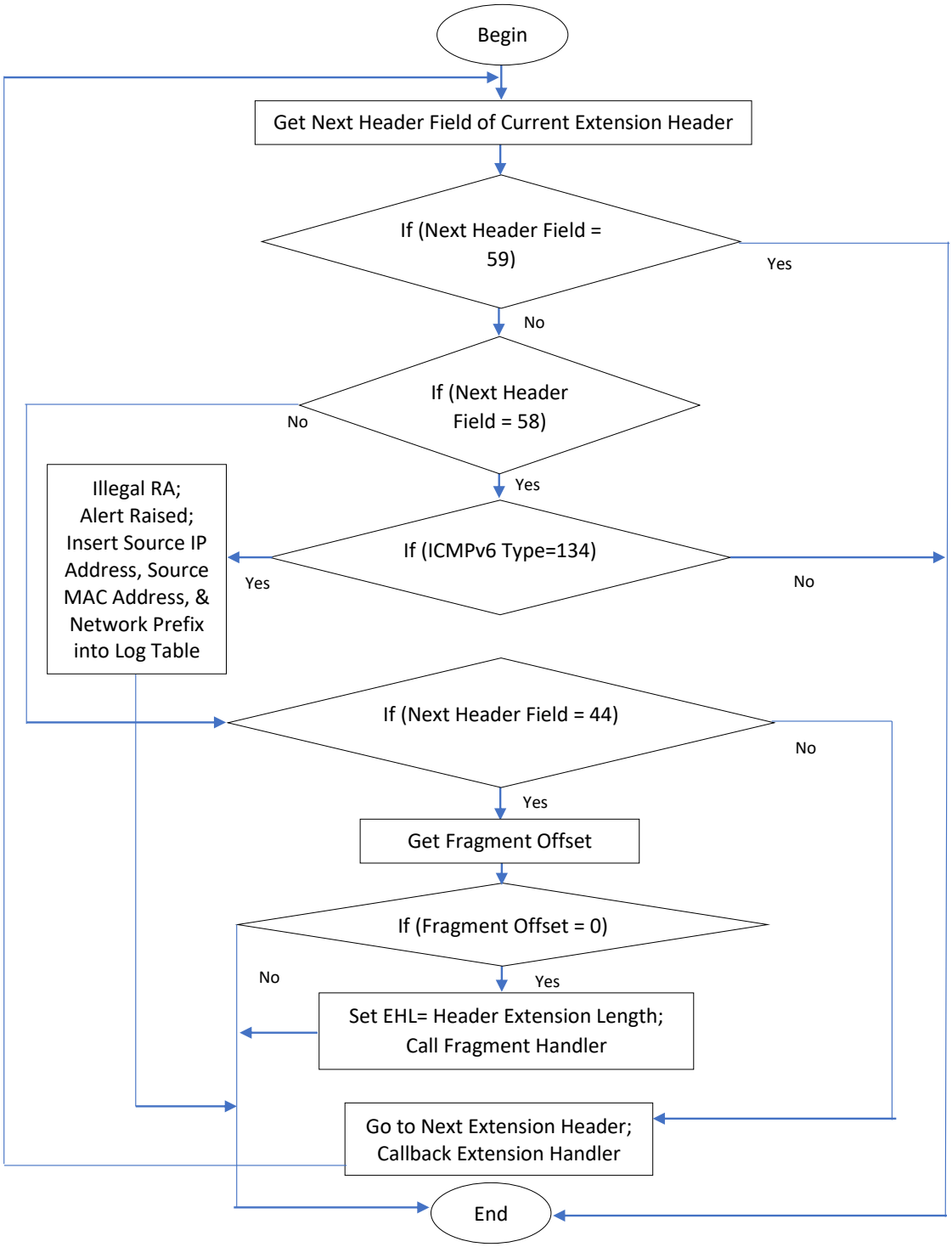


Fig. 8. Flow chart of Extension Handler

5. Implementation of SADetection

SADetection is developed as monitoring server that continuously listens to network traffics and has been implemented in test-bed environment. SADetection resides in the same network segment with victim and attacker. It is connected to a mirror port of the network switch so that it can sniff all network traffics from all connected machines. Figure 9 shows network diagram of SADetection during SLAAC attack scenario.

The test-bed simulates three (3) scenarios which are scenario when SADetection is running but SLAAC attack is not launched; scenario when SADetection is not running but SLAAC attack is launched; and scenario when SADetection is running during SLAAC attack. All three (3) scenarios are simulated in same network architecture, configuration and setting. For each scenario, test-bed is run for five (5) hours. Figure 10, 11 and 12 depict all three (3) scenarios.

fake_router26 tool from THC-IPv6 toolkit is used to launch the attack. The tool sends unsolicited RA message periodically to all hosts in the network. Three (3) variants of SLAAC attack will be simulated in both scenario two (2) and scenario three (3). The first variant is are attack using ICMPv6 type 134 packet. Attack using ICMPv6 type 134 packet is a common SLAAC attack. Attacker uses this packet when attacking network that has no NDP security safeguard. Second variant is attack using fragment packet. The third is attack using packet with Hop-by-Hop Option extension header.

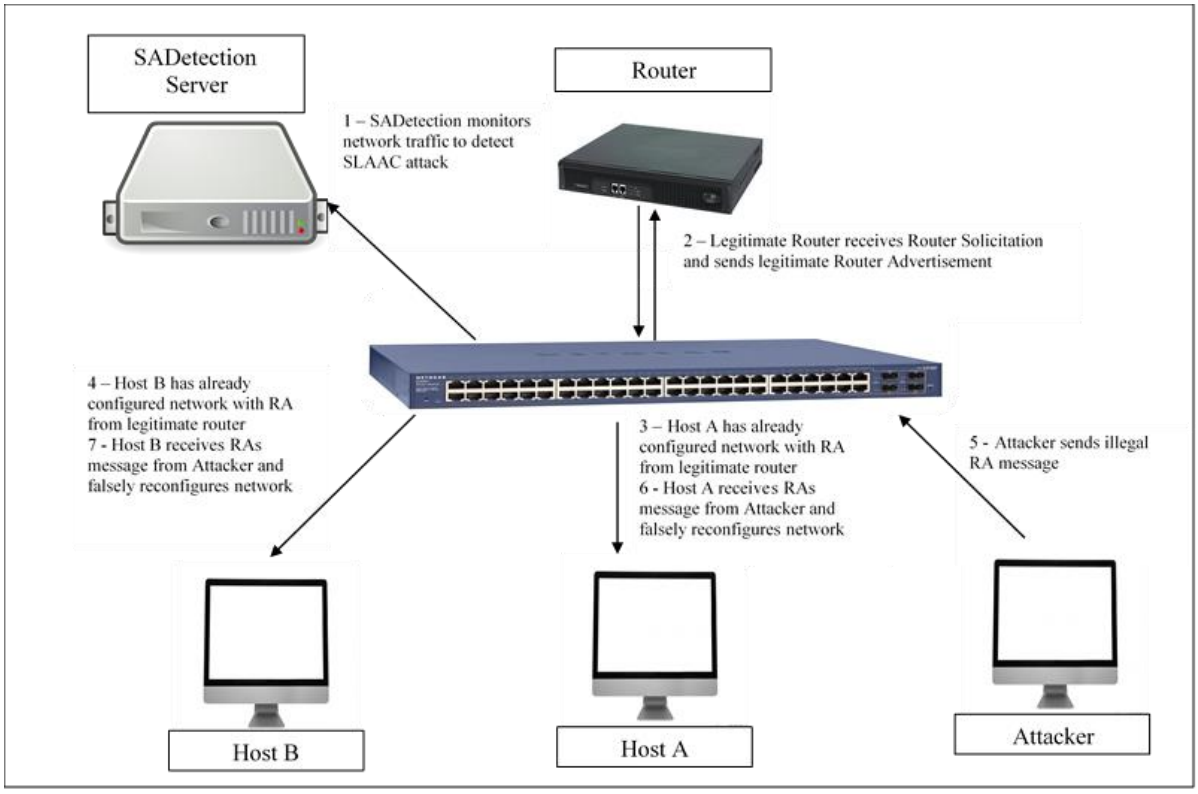


Fig. 9. SADetection network design

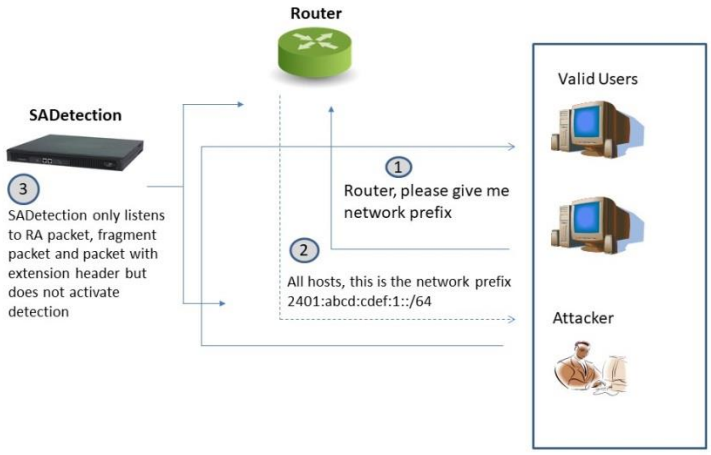


Fig. 10. SADetection is running but SLAAC attack is not launched

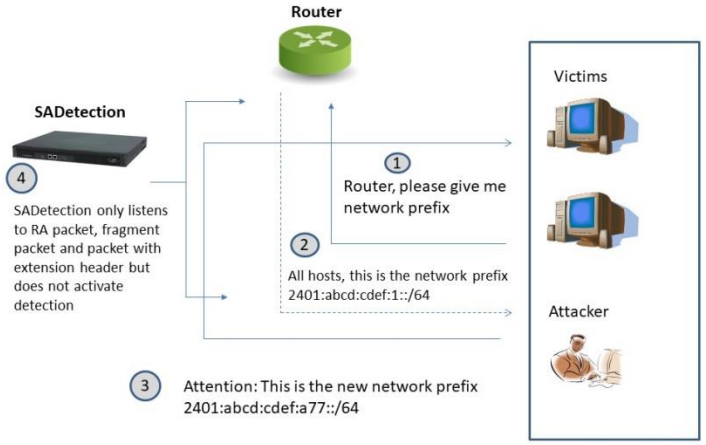


Fig. 11. SADetection is not running but SLAAC attack is launched

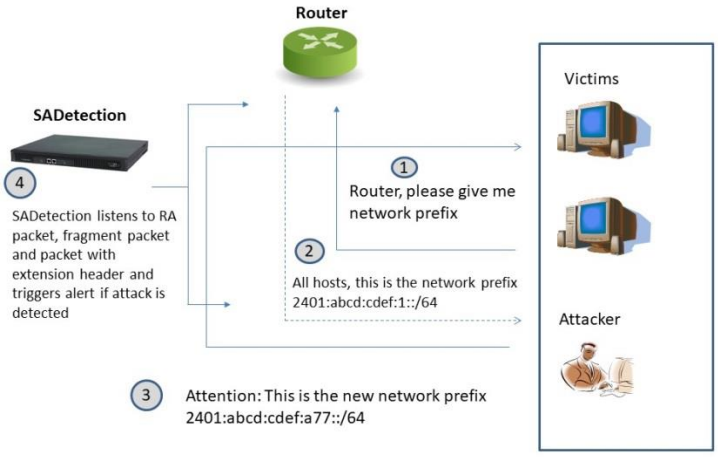


Fig. 12. SADetection is running during SLAAC attack

6. Results and Discussions

Based on result of implementation, it is found that SADetection algorithm runs linear time complexity. Since each packet is only verified once, the algorithm time complexity depends linearly on number of packets to be verified. The algorithm does not have overhead verification process because it applies lightweight anomaly profile and detection rulesets to maintain detection capability at the same time to reduce processing time and network resources. SADetection has detected all three (3) SLAAC attack variants. It demonstrates the capability of SADetection to prevent exploitation of packet with extension header. Once attack is detected, SADetection alert mechanism enables administrator to identify and neutralize attack from the network to protect other hosts. It shows the capability of SADetection to provide distributed protection at network level without involvement of host which can save host's processing and network resources. Simulation of scenario three (3) is repeated three (3) times to measure detection performance. Table 1, 2 and 3 show data packets analysis of the simulations. SADetection has correctly detected 97% of attack packets in first simulation, 98% of the attack packets in second simulation and 99% of attack packets in third simulation. In average, SADetection has achieved 98% detection accuracy rate as showed in Table 4. The 2% false alarm (false negative) happened at beginning of simulation session due to numbers of packets is not yet reach the attack threshold value. There is no false positive which means none of normal packet is wrongly classified as attack packet.

Table 1. Data packets analysis of first simulation

No.	Packet type	Numbers of Packet	Number of Packet Classified as Attack
1.	Attack packet using ICMPv6	871	830
2.	Attack packet using Fragmentation	770	750
3.	Attack packet using Extension Header	697	678
	Total	2338	2258

Table 2. Data packets analysis of second simulation

No.	Packet type	Numbers of Packet	Number of Packet Classified as Attack
1.	Attack packet using ICMPv6	1311	1290
2.	Attack packet using Fragmentation	1086	1057
3.	Attack packet using Extension Header	975	965
	Total	3372	3312

Table 3. Data packets analysis of third simulation

No.	Packet type	Numbers of Packet	Number of Packet Classified as Attack
1.	Attack packet using ICMPv6	2301	2259
2.	Attack packet using Fragmentation	1974	1969
3.	Attack packet using Extension Header	1511	1497
	Total	5786	5725

Table 4. Analysis of attacks detection by percentage

No.	Packet type	Session 1	Session 2	Session 3	Average
1.	Attack packet using ICMPv6	95%	98%	98%	97%
2.	Attack packet using Fragmentation	97%	97%	99%	98%
3.	Attack packet using Extension Header	97%	99%	99%	98%
	Total	97%	98%	99%	98%

The detection accuracy analysis has showed that SADetection is an effective detection mechanism. RA Guard and SAVI also can produce high detection rate, but false positive can be an issue if new legitimate RA information is not reconfigured in the setting. SADetection on the other hand is already designed to overcome and reduce false positive by verifying every new RA packet so that unauthenticated legitimate RA packet is not automatically blocked.

CPU utilization in term of processing time is compared between SADetection and other mechanisms. It is done to demonstrate processing time overhead that incurred by SADetection and other mechanisms in processing RA message with standard RA processing time as the baseline. The comparison is done both at sender and receiver side. Host is considered as sender and SADetection server is considered as receiver. The comparison is showed in Table 5 and Table 6.

Table 5. CPU utilization at host/sender

No.	Mechanism	Processing Time (miliseconds)	Overhead
1.	Standard RA (baseline)	1.146	-
2.	Trust-ND	15.25	14.104
3.	SeND	75.97	74.824
4.	SADetection	1.146	-

Table 6. CPU utilization at server/receiver

No.	Mechanism	Processing Time (miliseconds)	Overhead
1.	Standard RA (baseline)	1.69	-
2.	Trust-ND	15.377	13.687
3.	SeND	75.97	74.28
4.	SADetection	7.942	6.252

Processing time at host side is constant because SADetection does not modify standard RA message during implementation. Meanwhile, Trust-ND and SeND have incurred 14.104 and 74.824 processing time overhead respectively. At server side, SADetection only increases 6.252 processing time compared to 13.687 by Trust-ND and 74.28 by SeND. SADetection has showed that its processing time is better than Trust-ND and SeND. Network efficiency is measured from consumption of network bandwidth. Since bandwidth requirement depends on the length of packet, bandwidth consumption is computed from the length of RA packet in kilobits (KB). Network bandwidth consumption of SADetection is compared with SeND and Trust-ND to demonstrate network efficiency as showed in Table 7. SADetection does not modify or encrypt RA packets. Thus, in SADetection, standard plaintext RA packet is used which is 0.832 KB in length. The size of modified RA packet in Trust-ND and SeND are 3.072 KB and 35.328 KB respectively. SADetection has showed that the bandwidth consumption is better than Trust-ND and SeND.

Table 7. Bandwidth consumption analysis

No.	Mechanism	Bandwidth Consumption (kilobits)
1.	Standard RA (baseline)	0.832
2.	Trust-ND	3.072
3.	SeND	35.328
4.	SADetection	0.832

7. Conclusions

Selecting the best safeguard for SLAAC attack is a vital responsibility for security administrator. This paper has proposed and justified SADetection as security safeguard to protect IPv6 network from SLAAC attack. SADetection offers exceptional detection capability and security features. SADetection is detection mechanism using active monitoring and verification has detected SLAAC attack launched using ICMPv6 packet and launched using hidden RA message in packet with extension header at high accuracy rate. The detection mechanism of SADetection does not amend and conflicts with original NDP in Windows and Linux operating system.

SADetection very lightweight detection mechanism and can be deployed rapidly in wired or wireless as dedicated monitoring server. SADetection can be deployed in any IPv6 network without changes to network topology, configuration or policies. As centralized detection server, SADetection has avoided unnecessary utilization of computing resource for detection duty at host machine. SADetection also does not require configuration and extra process in monitored hosts.

There are potential future works that can be extended from this paper to further strengthen IPv6 security. Future works can be extended from SADetection itself or can be expanded from different point of security and technology view. Future works can be; extending active ongoing verification and authentication mechanism to detect other NDP attack; enhancing SADetection detection algorithm with smart or intelligent algorithm; improving SADetection by introducing automatic prevention feature; and improving SLAAC by redesigning router advertisement mechanism.

References

[1] S. Deering and R. Hinden. RFC 8200 Internet Protocol, Version 6 (IPv6) Specification. *RFC standard.* Internet Engineering Task Force (IETF). http://www.ietf.org/rfc/rfc8200.txt, 2017.

[2] T. Narten, E. Nordmark, W. Simpson and H. Soliman. RFC 4861 Neighbor Discovery for IP version 6 (IPv6). *RFC standard.* Internet Engineering Task Force (IETF). http://www.ietf.org/rfc/rfc4861.txt, 2007.

[3] S. Thomson, T. Narten and T. Jinmei. RFC 4862 IPv6 Stateless Address Autoconfiguration. *RFC standard.* Internet Engineering Task Force (IETF). http://www.ietf.org/rfc/rfc4862.txt, 2007.

[4] A. Cooper, F. Gont, and D. Thaler. RFC 7721 Security and Privacy Considerations for IPv6 Address Generation Mechanisms. *RFC standard.* Internet Engineering Task Force (IETF). http://www.ietf.org/rfc/rfc7721.txt, 2016.

[5] P. Nikander, J. Kempf and E. Nordmark. RFC 3756 IPv6 Neighbor Discovery (ND) Trust Models and Threats. *RFC standard.* Internet Engineering Task Force (IETF). http://www.ietf.org/rfc/rfc3756.txt, 2004.

[6] S. U. Rehman and S. Manickam. "Improved Mechanism to Prevent Denial of Service Attack in IPv6 Duplicate Address Detection Process". *International Journal of Advanced Computer Science and Applications (IJACSA)*, Vol. 8, No. 2, 2017.

[7] S. Praptodiyono, R. K. Murugesan, I. H. Hasbullah, C. Y. Wey, M. M. Kadhum and A. Osman. "Security Mechanism for IPv6 Stateless Address Autoconfiguration". *International Conference on Automation, Cognitive Science, Optics, Micro Electro-Mechanical System, and Information Technology (ICACOMIT)*, pp. 31-36, 2015.

[8] H. Rafiee and C. Meinel. "SSAS: A simple secure addressing scheme for IPv6 autoconfiguration".

Eleventh Annual Conference on Privacy, Security and Trust, pp. 275-282, 2013.

[9] D. McPherson, F. Baker and J. Halpern. RFC 6959 Source Address Validation Improvement (SAVI) Threat Scope. *RFC standard.* Internet Engineering Task Force (IETF). http://www.ietf.org/rfc/rfc6959.txt, 2013.

[10] J. Zhang, J. Liu, Z. Xu, J. Li and X. Ye. "TRDP: a Trusted Router Discovery Protocol". *International Symposium on Communications and Information Technologies*, pp 660-665, 2007.

[11] E. J. Arkko, J. Kempf, B. Zill, and P. Nikander. RFC 3971 SEcure Neighbor Discovery (SeND). *RFC standard.* Internet Engineering Task Force (IETF). http://www.ietf.org/rfc/rfc3971.txt, 2005.

[12] F. Gont. RFC 7113 Implementation Advice for IPv6 Router Advertisement Guard (RA-Guard). *RFC standard.* Internet Engineering Task Force (IETF). http://www.ietf.org/rfc/rfc7113.txt, 2014.

[13] F. Gont. RFC 6980 Security Implications of IPv6 Fragmentation with IPv6 Neighbor Discovery. *RFC standard.* Internet Engineering Task Force (IETF). http://www.ietf.org/rfc/rfc6980.txt, 2013.

[14] S. I. Shah, M. Anbar, A. Al-Ani and A. Al-Ani. "Hybridizing Entropy Based Mechanism with Adaptive Threshold Algorithm to Detect RA Flooding Attack in IPv6 Networks". *International Conference on Computational Science and Technology 2018 (ICCST2018)*, 2019.

[15] S. A. Abdullah. "SEUI-64 bits an IPv6 Addressing Strategy to Mitigate Reconnaissance Attacks". *Engineering Science and Technology, an International Journal*, Volume 22, Issue 2, pp 667-672, 2018.

[16] S. Y. Massamba and S. A. R. R. Cheikh. "Securisation of an IPv6 Address Obtaining with SLAAC in Home Networks*". OALib. 05*, pp 1-12, 2018.

[17] Y. Lu, M. Wang and P. Huang. "An SDN-Based Authentication Mechanism for Securing Neighbor Discovery Protocol in IPv6". *Security and Communication Networks*, pp 1-9, 2017.

[18] M. Schutte. IPv6 Plugin for the Snort Intrusion Detection System. *Technical report.* IPv6 Intrusion Detection System. http://www.idsv6.de, 2014.

[19] J. N. Goel and B. Mehtre. "Dynamic IPv6 Activation Based Defense for IPv6 router advertisement flooding (DoS) attack". *IEEE International Conference on Computational Intelligence and Computing Research*, pp. 1-5, 2014.

[20] F. A. Barbhuiya, S. Biswas and S. Nandi. "Detection of Neighbor Solicitation and Advertisement Spoofing in IPv6 Neighbor Discovery Protocol". *The 4th international conference on Security of information and networks (SIN '11)*, pp 111-118, 2011.

[21] G. Bansal, N. Kumar, S. Nandi and S. Biswas. "Detection of NDP Based Attacks Using MLD". *The 5th International Conference on Security of Information and Networks (SIN '12)*, pp 163-167, 2012.

[22] E. Levy-Abegnoli, G. Van de Velde, C. Popoviciu and J. Mohacsi. RFC 6105 IPv6 Router Advertisement Guard. *RFC standard.* Internet Engineering Task Force (IETF). http://www.ietf.org/rfc/rfc6105.txt, 2011.

[23] K. Scarfone and P. Mell. Guide to Intrusion Detection and Prevention Systems (IDPS). *Technical report*. The National Institute of Standards and Technology (NIST). https://www.nist.gov/publications/guide-intrusion-detection-and-prevention-systems-idps, 2007.