View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by UUM Repository

Journal of ICT, 14, 2015, pp: 77–93

PROTECTING HOME AGENT CLIENT FROM IPv6 ROUTING HEADER VULNERABILITY IN MIXED IP NETWORKS

[1]Bassam Naji Al-Tamimi, [2]Mohamed Shenify & [3]Rahmat Budiarto
[1]Taibah University, Almadinah Almonawarah, Saudi Arabia
[2&3]Albaha Uniersity, Albaha, Saudi Arabia

btamimi@taibahu.edu.sa; maalshenify@bu.edu.sa; rahmat@bu.edu.sa

ABSTRACT

Mixed IPv4/IPv6 networks will continue to use mobility support over tunneling mechanisms for a long period of time until the establishment of IPv6 end-to-end connectivity. Encapsulating IPv6 traffic within IPv4 increases the level of hiding internal contents. Thus, mobility in mixed IPv4/IPv6 networks introduces new security vulnerabilities. One of the most critical vulnerabilities associated with the IPv6 protocol is the routing header that potentially may be used by attackers to bypass the network security devices. This paper proposes an algorithm (V6HAPA) for protecting home agent clients from the routing header vulnerability, considering that the home agents reside behind an IPv4 Network Address Translation (NAT) router. The experimental results show that the V6HAPA provides enough confidence to protect the home agent clients from attackers.

Keywords: Mobile IP, IPv4/IPv6 coexistence, IPv6 security, IPv6 routing header.

INTRODUCTION

Wireless technologies are being more widely applied today across the world in various fields of sciences and industries. This trend is triggered by the needs of a huge number of users and devices in utilizing the Internet communication services. It is expected to continue unabatedly accompanied with the need of having numerous amounts of Internet Protocol (IP) addresses. The increasing demand of IP addresses has been exposing the shortage of the current IPv4 address space. The next generation IPv6 protocol is developed by the Internet Engineering Task Force (IETF) to overcome the shortage of the IP addresses. Nonetheless, it is not possible to migrate from IPv4 to IPv6 in the near future

due to the fact that many users and networks are still using IPv4 (Barker, 2013). Thus, the transition should be done gradually, as such, IPv6 will coexist with its precedence version IPv4 for a long period of time until the Internet is fully migrated to IPv6 (Hong, 2013; Bi, Deng, Xu, Shi, & Hu, 2013). Various transition mechanisms have been defined such as dual-stack, translation, and tunneling to support the interoperability between mixed IP networks (Amoss & Minoli, 2008).

Mobile Internet Protocol Overview

Mobile Internet Protocol (MIP) is the most well-known macro mobility scheme that solves the problem of node mobility by redirecting the traffic for a mobile node (MN) to its current location. MIP is an open standard protocol designed by the IETF to allow users to move from one network to another while maintains their own permanent IP addresses (Perkins, 1996).

Routing system in IP networks is based on fixed IP addresses and analogous to a postal letter delivery system (Taylor, Waung & Banan, 1997). Once the MN moves away from its home network, it is no longer reachable by using normal IP routing. The MN asks its home post office to forward the mail to its new attached network through the local post office there. Thus, when the MN leaves its home network and moves to another network, it uses the same IP address while moving over a different network. Therefore, MIP ensures that a moving individual can maintain its communication without sessions or connections being dropped.

MIP which is based on the Internet Protocol (IP) is more scalable for the Internet and offers wide connectivity for users, whether they are moving within their home network or traveling away from home crossing multiple Local Area Networks (LANs) configured with different IP address systems. Mobility support and its solutions are described in details in (Perkins, 2002; Arkko, Perkins & Johnson, 2011).

Security of MIP has always been of a high concern in any internetworking environment. Moreover, it is significance to be implemented in dual stack IPv4/IPv6 networks, since there is no compatibility between both protocols (Ahmadi, 2012).

SECURITY CONCERNS IN MIXED IP NETWORKS

Throughout the transition period, the movement of the MNs among networks that configured with different IP protocols is inevitable (Lee, Jung, Lee, Lee &

Han, 2013). As a result, many researchers have shown interests in proposing new systems that address the issue of IPv4 and IPv6 coexistence with mobility support.

Several studies have investigated security concerns and implications of MIP such as (Convery & Miller, 2004; Durdağı & Buldu, 2010; La Polla, Martinelli, & Sgandurra, 2013; Zagar, Grgic, & Rimac-Drlje, 2007). However, the security concerns of both MIP protocols (i.e., MIPv4 and MIPv6) have been considered separately since their designing period, and a little attention has been given to these protocols in mixed IPv4/IPv6 environment.

Shanmugaraja and Chandrasekar (2012) discuss some security issues of IPv4 and IPv6 as well as analyze different security threats that may emerge due to implementation of various transition mechanisms. The most critical vulnerability related to IPv6 extension headers was identified in a research work by Savola (2002). This vulnerability can occur due to the exploitation of the IPv6 RH feature which has been demonstrated and analyzed in many recent studies (Durdağı & Buldu, 2010; Frankel, Graveman, Pearce, & Rooks, 2010; Karthikeyan & Prittopaul, 2013). According to the IPv6 specification (Deering & Hinden, 1998), all nodes that support IPv6 must be able to process IPv6 RHs. On the other hand, such vulnerability can be used by attackers to bypass network security through avoiding access control lists on destination addresses (Biondi & Ebalard, 2007). In this concern, Abley, Savola, & Neville-Neil (2007) suggest that the firewall policy must block forwarding packets with type 0 RHs (RH0) and permit other types of RHs to pass through. Blocking all IPv6 packets containing RHs is, however, not a worthy solution as this could have serious implications for the IPv6 future development. Recently, most of firewall policies block all packets containing RH0. In addition, the default firewall configuration prevents the forwarding of IPv6 traffic with RH0.

Moreover, IPv6 packets channeled through Teredo tunnel is not subjected to a deep traffic inspection process, because the packets are encapsulated into IPv4 packets (Krishnan, Hoagland, & Thaler, 2011). Therefore, Teredo tunnel allows inbound access from the Internet to the devices located behind NAT router. As a consequence, the use of IPv6 RH in different IP networks may increase the vulnerability of IPv4 networks. It is due to the fact that IPv4 security devices, such as firewalls, are not able to inspect IPv6 header encapsulated in IPv4 packet. Thus, proposing new security mechanisms that consider the vulnerability of IPv6 RH are crucial to secure the communications between different IP networks.

Vulnerability of Using IPv6 Routing Header

The RH functionality which is originally provided by IPv6 can be used to list one or more intermediate nodes to be visited on the way to a packet's destination. At the same time, it can be exploited by the attackers to bypass the traffic filtering mechanism and generate a Denial of Service (DoS) attack (Abley et al., 2007; Convery & Miller, 2004; Wadhwa & Khari, 2011) .

Having done packet inception using Man-In-The-Middle technique, an attacker can exploit the RH in order to generate malicious packets which are performed through specifying a victim node's IP address in the RH. These kinds of packets will be routed through a public accessible IP address (e.g., network server) and some intermediate nodes to be finally delivered to the victim host. Certainly, the malicious packets will be going through a checking process in the server of the intended network. However, without checking whether the IP addresses are trusted or not the server forwards these packets based on the IP addresses specified in the RH. Thus, the malicious packets will reach the victim host without breaking any of security policies as illustrated in Figure 1.

This vulnerability results in such case: with every new attachment of an MN to an IPv6 network, all the clients of its Home Agent (HA) would become susceptible to attacks. Consequently, all packets which are received and passed through the HA must be subjected to a deep inspection process.

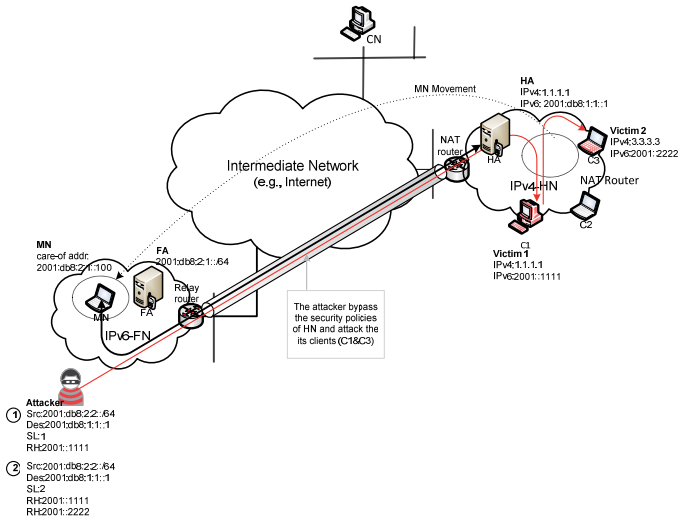

Figure 1. Routing header attack

IPV6 HOME AGENT PROTECTION ALGORITHM (V6HAPA)

This paper proposes an algorithm (V6HAPA) that works once an IPv6 address has been activated on the HA or on one of its clients. The main purpose of implementing this algorithm is to deny packets with spoofed addresses accessing the home network, and to allow the trustable packets to reach its correct destinations.

Once an IPv6 packet arrives at HA, it is subjected to the rules of filtration process. In case of a packet does not match the rules, then the algorithm discards the packet. Otherwise, the packet is checked regarding the value of the next header (NH). If this value is not equal 43 (i.e., the following header is not RH), then the algorithm accepts the packet. If the value is equal 43 then, the type of IPv6 RH is checked either it is 0 type or not. If a type 0 of IPv6 RH exists, then the algorithm checks the value of IPv6 RH Segment Left (RH_{SL}) whether it is 0 or not. If it is equal to 0 then accept the packet. Otherwise, check the currently first destination RH (Dst_{RH}). As a result, the packet is discarded if its address does not match the assigned filtration rules (i.e., the Dst_{RH} is not found in the Authorized_$IPv6_{RH}$ list). In case of matching, the current Dst_{RH} is interchangeable with the address of the next one, and at the same time, the value of RH_{SL} is decremented by one. The interchangeability and matching processes are repeated until the RH_{SL} value becomes zero.

On the other hand, if the type of IPv6 RH is not equal 0, then the algorithm checks whether the RH type is not equal 2 and then discards the packet. Otherwise, the algorithm checks the RH_{SL}. In case the value of RH_{SL} is greater than 1, the algorithm simply discards the packet. Otherwise, the embedded Dst_{RH} IP address inside the packet is checked and if it is found in the $IPv6_{CoA}$–cache table, then the algorithm accepts the packet. If the IPv6 Dst_{RH} IP address is not found in the list, the packet would be silently discarded. In case the RH_{SL} is equal 0, the algorithm accepts the packet. Figure 2 presents the details of the V6HAPA.

The filtration process of the packets containing RHs is basically based on two essential lists:

a) The Authorized_$IPv6_{RH}$ list, is mainly concerned with the packets of RH0. Packets of this type will be denied unless all the destination RH addresses are matched with the entries inside the list.
b) The IPv6CoA_cache is concerned with the packets of RH2. The V6HAPA algorithm accepts these packets as long as the embedded RH destination IP addresses of the received packet matches the IP addresses in this list. In case of having a packet with RHSL greater than zero, then it is silently discarded.

V6HAPA Algorithm

```
1:       // Packet (pkt) arrival & decapsulation process
2:      // Matching pkt with the filtration rules
3:   for each pkt do the following:
4:       if   src_IP does not match the filtering rules then
5:           discard  pkt
6:       // Check the IPv6 Next Header (NH) whether consists of RH
     or not
7:       else if   the NH value ≠ 43 then
8:           accept pkt
9:       // Check the type of routing header (RH)
10:      else if   the RH = type 0 then
11:          temp = RH_SL
12:          while (RH_SL > 0){
13:          // Check the first Destination Routing Header (Dst_RH)
14:              if the Dst_RH does not match with the filtering rules
     then
15:                   discard  pkt
16:                   exit
17:              else{
18:               // Perform replacement process
19:                    swap (( temp – RH_SL ) + 1) , current Dst_RH)
20:                    RH_SL -- // the value RHSL is decreased by 1
21:          } // End of while
22:          accept pkt
23:          Exit
24:      else if the RH ≠ type 2
25:          // In case the type of RH ≠ "type 2"
26:          discard  pkt
27:          Exit
28:         // Check the value of SL
29:      Else
30:           If RH_SL = 0
31:                accept pkt
32:           else if RH_SL = 1
33:              // Check the IPv6 Dst_RH with the IPv6_CoA cache
34:                   If the Dst_RH is matched then
35:                        accept pkt
36:                        Exit
37:                   else discard  pkt
38:                        Exit
39:           Else
40:                discard  pkt
41:                exit
```

Figure 2. V6HAPA Algorithm

PERFORMANCE EVALUATION

In order to evaluate the functionality of the V6HAPA algorithm, two metrics are considered: (a) the performance and (b) the accuracy. The equations that calculate the accuracy of preventing suspicious IPv6 packets with RH are given in Equation (1) and Equation (2) (Osareh & Shadgar, 2008).

$$Accuracy = \left(\frac{TP+TN}{TP+TN+FP+FN}\right) * 100\% \quad (1)$$

$$False\ Positive\ rate\ = \left(\frac{FP}{(FP+TN)}\right) * 100\% \quad (2)$$

The detected suspicious and non-suspicious packets in V6HAPA algorithm can be categorized as follows:

- *True Positive (TP)*: the situation in which the actual attack is detected as an attack.
- *True Negative (TN)*: the situation in which the actual normal is detected as a normal.
- *False Positive (FP)*: the situation in which the actual normal is detected as attack.
- *False Negative (FN)*: the situation in which the actual attack is detected as normal.

In this paper, the false positive is defined as the situation in which the actual normal packet is detected as an attack. False positive occurres because the proposed algorithm rejects all the suspicious packets (i.e., malicious and normal packets) carrying unmatched IPv6 routing header addresses.

The filtration process (matched/unmatched) of the packets containing routing header is configured by the network administrator. Possibly, some of the rejected packets by V6HAPA were not harmful and also do not intend to attack the network. However, the V6HAPA rejects all packets (malicious and normal packets) carrying routing header due to IPv6 routing header addresses in those rejected packets do not match the assigned filtration rules in the Authorized_IPv6RH list. Hence, some of the normal packets carrying IPv6 routing header will be rejected. As a result, the accuracy was not 100% and false positives are recorded during the experiment

Experimental Testbed Topology

The testbed topology has three components as follows (See Figure 3).

a) *IPv6 traffic emulator* is designed to generate realistic IPv6 packets including RH0 and RH2 that are used to evaluate the performance of the V6HAPA algorithm. We can emulate several CNs simultaneously send packet to a MN stay behind NAT router
b) *V6HAPA module in home network* implemented in the HA. This module receives the packets sent through the NAT router and then processes these packets according to the V6HAPA algorithm.

c) *HA clients* act as MNs moved in into IPv4 only network with NAT. This HA clients are having connection with outsiders and obtaining IPv6 addresses from Teredo server.

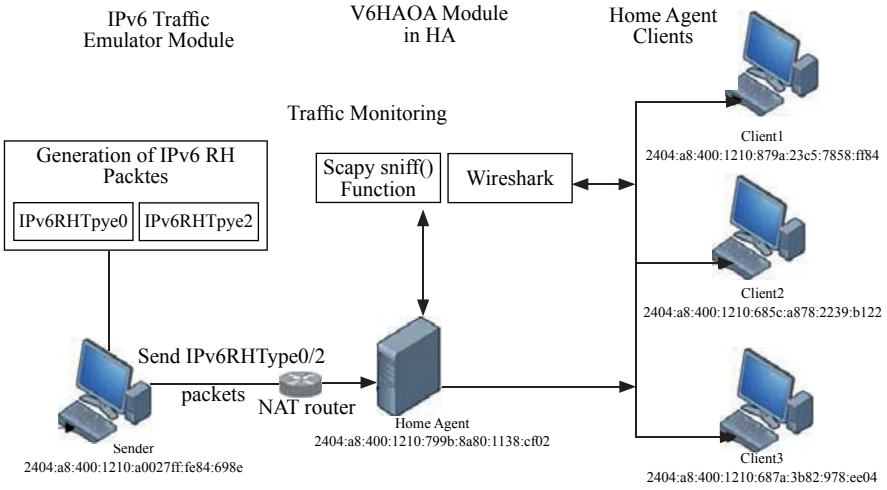

Figure 3. Testbed Topology

Table 1 shows the hardware specifications and the configuration settings for the undertaking experiment.

Table 1

Hardware and Configuration Settings and Specifications

Installation & configuration settings	Sender	Receiver	Clients
Operating System	Linux Fedora 13	Linux Fedora 13	Windows 7
PC manufacturer	Acer® PC	Acer® PC	Dell® PC
Processor	Intel® Core™ 2 CPU, E4500 @ 2.20GHz	Intel® Core™ 2 CPU, E4500 @ 2.20GHz	Intel® Core™ 2 CPU, E4500 @ 2.20GHz
RAM	2 GB	2 GB	4 GB
Implementation	Scapy 2.2.0 , Python	C programming language	Configuration
Traffic Monitoring tool	Scapy sniffing function	Wireshark, Scapy sniffing function	Wireshark , Scapy sniffing function

(Continued)

Installation & configuration settings	Sender	Receiver	Clients
IPv6 Address	2404:a8:400:1210:a00:27 ff:fe84:698e	2404:a8:400:1210:799b:8 a80:1138:ef02	C1. 2404:a8:400:1210:8 79a:23c5:7858:ff84 C2. 2404:a8:400:1210:6 85c:a878:2239:bf22 C3. 2404:a8:400:1210:6 87a:3b82:978:ee04

The Experiments

The experiments exhibit the ability of the V6HAPA algorithm in protecting the networks from the attackers who have exploited the IPv6 RH function as well as demonstrate the performance in term of processing time and accuracy. For all experiments, the IPv6 RH has been classified into 5 categories: RH0 [5], RH0 [10], RH0 [15], RH0 [20] and RH0 [24]. The number in the brackets indicate the number of IPv6 destination addressess inluded in the RH. This classification is designed to be consistent with the number of the IPv6 destination addresses in each RH (i.e., the number of IP destination addresses in each RH could be at most 24 IP addresses).

The Scenarios

First, we consider the experiments for evaluating the performance (in term of packet filtering time) of the V6HAPA algorithm to detect suspicious packets containing RH0 and its accuracy in filtering such packets without disturbing normal packets containing the same headers. To this end, the Abley et al., (2007) suggestions are employed. However, this trend has raised another implication in the accuracy aspect. It may prohibit the forwarding of normal packets containing RH0.

Scenario 1: multiple CNs send IPv6 packets containing 50% normal packets and 50% suspicious packets. Table 2 shows the details regarding the RH type and size. For each RH type we conducted 10 runs in the experiment, starting with 500 packets up to 5000 packets with 500 packets increament. Thus, totally we experiment with overall of 27500 packets.

Scenario 2: Five CNs are emulated to craft and send simultaneously 5000 IPv6 packets to the HA. Three CNs send IPv6 packets containing RH0, while the rest send packets without RH0. According to Field (2009) the majority of observations should be at least 60% of the population as a normal packets. Hence, in this paper 70% normal packets (i.e., packets without RH0) and 30%

malicious (packets that include RH0) are considered to be the representative of the majority of the packets. The packets which include RH0 are distributed as follows:

Table 2

Traffic Data Type and Size for Scenario

No	RH type	RH size
Test 1	RH0[5]	88 bytes
Test 2	RH0[10]	168 bytes
Test 3	RH0[15]	248 bytes
Test 4	RH0[20]	328 bytes
Test 5	RH0[24]	392 bytes

(1) 20% of the packets have matched IP destination addresses with the identified list (Authorized_IPv6$_{RH}$), and (2) 10% of those packets have unmatched IP destination addresses (i.e., Suspicious packets) in the RH. The unmatched packets are divided into 7% malicious packets and 3% normal packets (see Table 3).

We conduct an experiment for this scenario, and the results have been subsequently used to calculate the accuracy of the V6HAPA algorithm in terms of preventing the HA from RH0 vulnerability using Equation (1) and Equation (2).

Table 3

Traffic Data Classification for Scenario 2

No	Classification	RH type	No. of packets	Percentage of packets %
1	Normal packets	without RH	3500 pkts	70%
2	Packets containing RH0 (matched packet)	RH0[5]	1000 pkts	20%
3	Suspicious packets containing RH0 (malicious packet)	RH0[5]	350 pkts	7% (out of 10%)
4	Suspicious packets containing RH0 (normal packet)	RH0[5]	150 pkts	3% (out of 10%)

Secondly, we consider the experiment for evaluating the performance and accuracy of the V6HAPA algorithm to detect suspicious packets containing RH2.

Scenario 3: Five CNs are emulated to send IPv6 packets to HA clients through NAT and V6HAPA module in HA. The CNs are divided into three sets. The first set has two nodes which are intended to generate and send suspicious packets with RH2 (containing unregistered IPv6 destination address). The second set contains two nodes that generate and send packets without RH2. The last set represents an authorized CN which intends to generate packets containing RH2 with valid IPv6 destination address. The generated packets sent by the authorized CN are specified with only one RH destination IP address per packet. The embedded IP addresses within the RH2 must be matched with the home address of the MN that has already stored in the $IPv6_{CoA}$–cache. Further details of the data experiment for this scenario are given in Table 4. Total number of packets is 5000.

Table 4

Traffic Data Classification for Scenario 3

No	Classification	Sender	RH type	No. of packets	packets %
1	Normal packets	Set 2	without RH	3500 pkts	70%
2	Packets containing RH2 (only 1 matched RH Dest IP Address)	Set 3	RH2	1000 pkts	20%
3	Suspicious packets containing 1 unmatched RH Dest IP address (normal packets)	Set 1	RH2	100 pkts	2% (out of 10%)
4	Suspicious packets containing 1 unmatched RH Dest IP address (malicious packet)	Set 1	RH2	100 pkts	2% (out of 10%)
5	Packets containing more than 1 matched RH Dest IP address RH2	Set 1	RH2[2]-RH2[24]	100 pkts	2% (out of 10%)
6	Suspicious packets containing more than 1 unmatched RH Dest IP address RH2 (normal packets)	Set 1	RH2[2]-RH2[24]	100 pkts	2% (out of 10%)
7	Suspicious packets containing more than 1 unmatched RH Dest IP address RH2 (malicious packet)	Set 1	RH2[2]-RH2[24]	100 pkts	2% (out of 10%)

EXPERIMENTAL RESULTS AND DISCUSSION

Figure 4 shows the filter processing time in seconds versus the number of RH IPv6 addresses and the number of IPv6 packets sent (using the data in Scenario 1). For 1000 IPv6 packets containing up to 24 IPv6 destination addresses (i.e., RH0 [24]), the processing time is 29 seconds. For 5000 IPv6 packets of the same RH0 classification, the processing time increases to 139 seconds. It can be noticed that there is a significant increase in the packet filtering time when the number of packets increases.

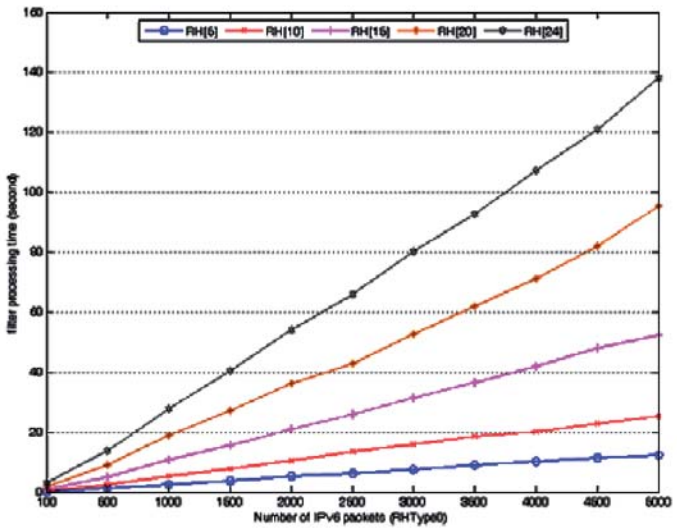

Figure 4. Filter Processing Time vs. Number of IPv6 Packets with RH0 [5]-[24]

Figure 5 shows the filtration processing time consumed in filtering 5000 packets according to the settings of the Scenario 2. From Figure 5, it can be noticed that there are three levels of the aggregation points indicated by three arrows. From bottom to top, the first arrow refers to the time that is consumed in filtering around 3500 packets without RHs. The second arrow refers to the time that is consumed in filtering around 500 suspicious packets with RHs. The last arrow refers to the time that is spent in filtering around 1000 matched packets with RHs. It is obvious that the filtration time for the suspicious packets is less than the time consumed by the RH matched packets. However, the time of filtration for packets without RHs is less than the time consumed by the suspicious packets. The rationale behind this observation is that the filtration process for matched packets continues until the last RH IP

address while in case of unmatched packets the filtering process stops when at least one of those IP addresses does not match with the IP addresses in the list. Hence, it can be concluded that this algorithm performs better considering the time required for filtering the unmatched packets.

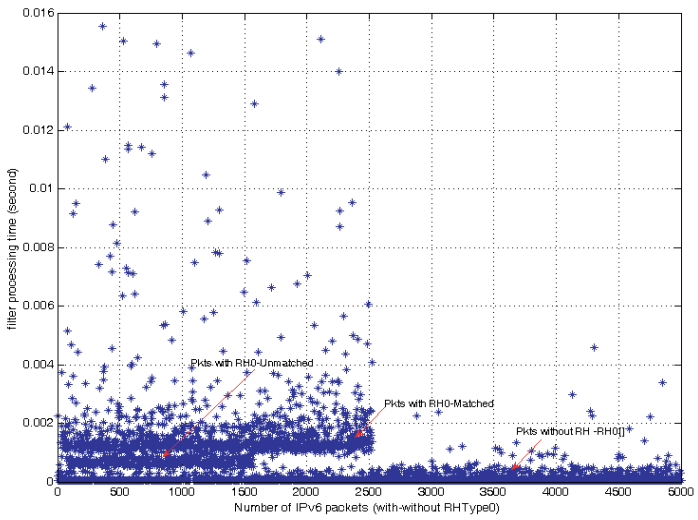

Figure 5. Three Levels of Aggregation Points Represent Different Types of RH0

Experimental results of Scenario 3 is shown in Figure 6. The results show that the V6HAPA algorithm has the highest accuracy compared to that recorded with the existing firewall policies (configured at the IPv4 only network in Figure 2). In false positive rates, the total number of normal packets, involving RHs, which are recognized as suspicious packets, is taken into consideration. From Figure 6, it can be clearly seen that the false positive rates will increase even more if the number of suspicions packets increases. Also, the results indicate that the V6HAPA algorithm is more accurate than the existing firewall policies.

The experimental results for the time spent in filtering packets containing RH2 for V6HAPA algorithm are shown in Figure 7 and Figure 8. These two figures also show the filtering time per packet for the same scenario without security consideration. The V6HAPA algorithm affects the network performance in terms of delay caused by the filtering process. However, the V6HAPA algorithm provides a notably greater effect compared to the standard tunneling, because

the V6HAPA algorithm discards any packet containing multi-hop RH2 and the RH2 unmatched packets. As a consequence the accuracy of the V6HAPA in filtering packets consisting RH2 is only about 94%.

Figure 6. Accuracy of Filtering Packets Containing RH0

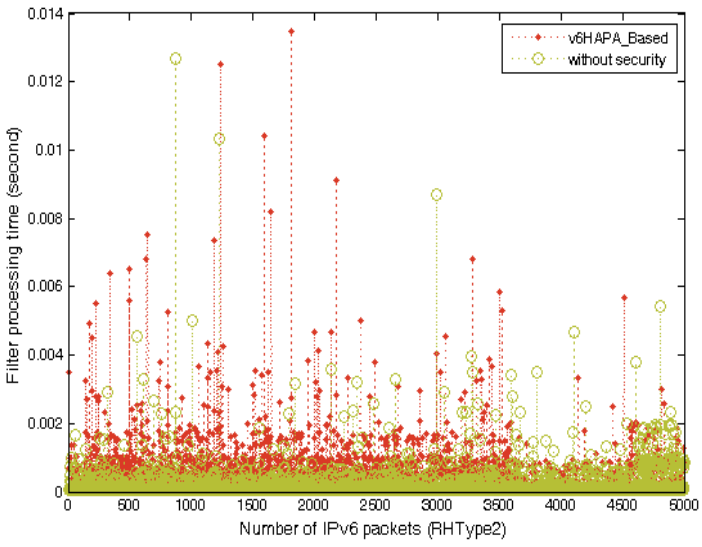

Figure 7. Filter Processing Time vs. Number of IPv6 Packets Containing RH2

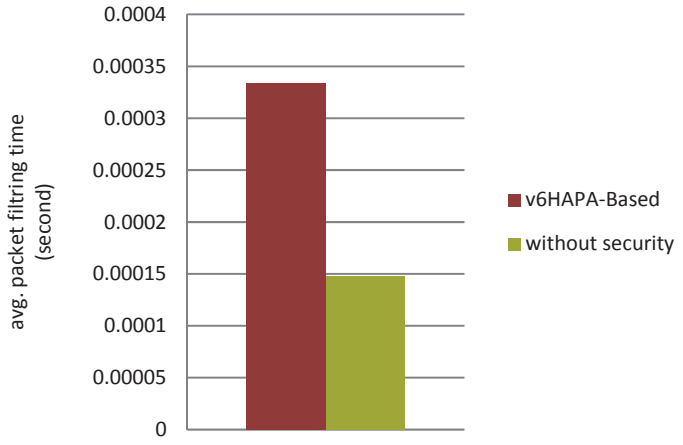

Figure 8. The average RH2 packet filtering time for V6HAPA algorithm

CONCLUSION

Mobile IP security always has a high concern in any internetworking environment. It also has special advantages to be implemented in mixed environments of MIPv4 and MIPv6. Since there is no compatibility between both protocols, the security concern is considered in this paper. V6HAPA, an algorithm to secure the HA clients from the attackers who can exploit the IPv6 RH through bypassing the security filtration policies is introduced.

Experimental results using a testbed, demonstrate the ability of the proposed algorithm to detect suspicious packets containing RH0 or RH2 and its accuracy in filtering such packets without disturbing normal packets containing the same headers. In spite of higher filtering consumption time, the V6HAPA algorithm has a high performance and accuracy in preventing attackers from bypassing security filtration policies.

As for future work we consider to extend the capability of V6HAPA in handling multi hop routing header type 2 (RH2).

REFERENCES

Abley, J., Savola, P., & Neville-Neil, G. (2007). *Deprecation of type 0 routing headers in IPv6*. IETF Internet standard, RFC 5095. Retrieved from http://www.ietf.org/rfc/rfc5095.txt.

Ahmadi, S. M. (2012). Analysis towards mobile IPv4 and mobile IPv6 in computer networks. *International Journal of Intelligent Systems and Applications (IJISA), 4*(4), 33-51.

Amoss, J., & Minoli, D. (2008). *Handbook of IPv4 to IPv6 transition: Methodologies for institutional and corporate networks*. Boca Raton, FL: Auerbach Publications.

Arkko, J., Perkins, C., & Johnson, D. (2011). *Mobility support in IPv6.* IETF Internet standard, RFC 6275. Retrieved from http://ietf.org/rfc/rfc6275.txt.

Barker, K. (2013). The security implications of IPv6. *Network Security, 6*, 5-9.

Bi, J., Deng, H., Xu, M., Shi, F., & Hu, G. (2013). *A General framework of source address validation and traceback for IPv4/IPv6 transition scenarios*. IETF Internet standard Draft. Retrieved from http://tools.ietf.org/html/draft-xu-savi-transition-03.txt.

Biondi, P., & Ebalard, A. (2007). *IPv6 routing header security*. Retrieved from SECDEV website http://www.secdev.org/conf/ipv6_rh_security-csw07.pdf.

Convery, S., & Miller, D. (2004, May). *IPv6 and IPv4 threat comparison and best-practice evaluation (V1. 0)*. [PowerPoint slides]. Paper presented at the 17th NANOG conference. Retrieved from http://seanconvery.com/v6-v4-threats.pdf

Deering, S., & Hinden, R. (1998). *Internet Protocol, Version 6 (IPv6) Specification*. IETF Internet standard, RFC 2460. Retrieved from http://ietf.org/rfc/ rfc2460.txt.

Durdağı, E., & Buldu, A. (2010). IPv4/IPv6 Security and Threat Comparisons. *Procedia-Social and Behavioral Sciences, 2*(2), 5285-5291.

Field, A. (2009). *Discovering statistics using SPSS* (3rd ed.). London: Sage Publications.

Frankel, S., Graveman, R., Pearce, J., & Rooks, M. (2010). *Guidelines for the Secure Deployment of IPv6*. NIST Special Publication, 800(119). Retrieved from http://csrc.nist.gov/sp800-119.pdf.

Hong, L. X. (2013). The research of network transitional technology from IPv4 to IPv6. *Proceedings of the 4th International Conference on Digital Manufacturing and Automation (ICDMA)*, pp.1507-1509, IEEE Publisher.

Karthikeyan, V., & Prittopaul, P. (2013). A survey on vulnerability of type 0 routing header in IPv6. *International Journal of Computer Science and Management Research, 2*(2), 1671-1676.

Krishnan, S., Hoagland, J., & Thaler, D. (2011). *Security Concerns with IP Tunneling*. IETF Internet standard, RFC 6169. Retrieved from http://ietf.org/rfc/ rfc6169.txt.

La Polla, M., Martinelli, F., & Sgandurra, D. (2013). A survey on security for mobile devices. *IEEE Communications Surveys & Tutorials, 15*(1), 446-471.

Lee, K.-H., Jung, H.-K., Lee, H.-W., Lee, S.-K., & Han, Y.-H. (2013). A network-based IP mobility management scheme with IPv4/IPv6 Dual Stack Support. In H-K Jung et. al. (Eds.), *Future Information Communication Technology and Applications: ICFICE 2013.* Lecture Notes in Electrical Engineering (pp. 199-216), Springer.

Osareh, A., & Shadgar, B. (2008). Intrusion Detection in computer networks based on machine learning algorithms. *International Journal of Computer Science and Network Security, 8*(11), 15-23.

Perkins, C. (1996). *IP Mobility Suppor*t. IETF Internet standard, RFC 2002. Retrieved from http://ietf.org/rfc/rfc2002.

Perkins, C. (2002). *IP mobility support for IPv4*. IETF Internet standard, RFC 3344. Retrieved from http://ietf.org/rfc/rfc3344.

Savola, P. (2002). *Security of IPv6 routing header and home address options*. IETF Technical report. Retrieved from http://tools.ietf.org/html/draft-savola-ipv6-rh-ha-security-00.txt.

Shanmugaraja, P., & Chandrasekar, S. (2012). Accessible methods to mitigate security attacks on IPv4 to IPv6 transitions. *European Journal of Scientific Research, 77*(2), 165-173.

Taylor, M. S., Waung, W., & Banan, M. (1997). *Internetwork mobility: The CDPD approach*. New Jersey: Prentice Hall.

Wadhwa, M., & Khari, M. (2011). Security holes in contrast to the new features emerging in the next generation protocol. *International Journal of Computer Applications, 20*(3), 35-39.

Zagar, D., Grgic, K., & Rimac-Drlje, S. (2007). Security aspects in IPv6 networks-implementation and testing. *Computers & Electrical Engineering, 33*(5-6), 425-437.

.