Record Path Header for Triangle Routing Attacks in IPv6 Networks

SU-KIT TANG, KIN-YEUNG WONG
Macao Polytechnic Institute
Rua Luis Gonzaga Gomes, Macao SAR, CHINA
{sktang, kywong}@ipm.edu.mo

KAI-HAU YEUNG
City University of Hong Kong
Tat Chee Avenue, Kowloon, Hong Kong SAR, CHINA
eeayeung@cityu.edu.hk

Abstract: - Triangle routing is one of the serious attacks to the Internet infrastructure. It can be caused by malicious routers which misroute packets to wrong directions. This kind of attacks creates network problems such as network congestion, denial of service and network partition and results in degrade of network performance. This paper gives a comprehensive study on how the path analysis combats the triangle routing attacks. We discuss the method, implementation and limitation of path analysis to detect triangle routing in IPv4 network. We also discuss the implementation of path analysis in IPv6 by proposing a new extension header, called Record Path Header.

Key-Words: - IPv6, Infrastructure, Routing, Attack

1 Introduction

With the absence of the security mechanisms such as authentication or encryption in the TCP/IP protocol suite, various security algorithms for different security services (eg., confidentiality, authentication, integrity, etc) have been developed and being in use to protect sensitive information from unauthorized access.

Information transmitting all the way from the sender to the receiver is assumed to be safe based on the trust. All network devices in the transmission path are naturally trusted to perform their tasks properly. For example, routers deliver data to their destinations based on the best path determined by their routing protocols. Unfortunately, various kinds of attack may change the behavior of network devices and results in network congestions, throughput lowering and denial of service. These consequences seriously lower the availability of a network. Availability is an important issue that system resources being accessible and usable upon demand should be provided whenever users request them. Thus, network infrastructure security is highly recommended to be in place and stops any potential attacks to the network system.

Various kinds of attacks to the Internet infrastructure have been studied in [1][2][3][4]. One of the serious attacks is called Triangle Routing, which is caused by misrouting packets to wrong directions by malicious routers. It will create problems to a network such as network congestion, denial of service and network partition and will result in degrade of network performance. The objective of this paper is to introduce a method to detect the occurrence of triangle routing attack in a network by analyzing the data traveling path.

This paper will start with the definition of the triangle routing attack, followed by the impacts of the problem to a network. A solution to the detection of the misrouting problem is proposed and its implementation in IPv4 is discussed. However, the IPv4 solution is less practical as it has some limitations. We also have a comprehensive study on the triangle routing attacks in IPv6 network. The study shows that the IPv6 network suffers from the triangle routing attacks. We then propose a new extension header, called Record Path header, for the implementation of our path analysis detection method. This IPv6 solution works in a more flexible way and does not have the limitations occurred in IPv4.

2 Triangle Routing Attacks

In Internet infrastructure, router is vulnerable and likely to get attention of adversaries. The job of a router is simple: looks at the destination of a packet and routes packets to the next hop based on the best path calculated by its running routing algorithm. The best path calculation of a routing algorithm is based on the information it collects from other routers and the result is stored in a routing table. A

router looks up its routing table for best path when it routes packets.

A router could be compromised in some ways under the control of adversaries. Chakrabarti [2] has pointed out that a routing table poisoning attack may change the behavior of a router. Routing table under this attack may be modified or updated with incorrect routing information after exchanging of poisonous routing information. This poisonous information could be generated by injection of faulty routing information packets into the network. As a matter of fact, some popular routing protocols do not have any security protection in their message exchange. It will put the routers running that protocols in risk.

RIP, a popular distance-vector protocol, is an example of the protocol that does not employ any security mechanism on its message update [5]. OSPF, a popular link-state protocol, provides MD5 authentication option that requires every single router in a group to have a secret key preset. Update message is accepted if the hash generated from the receiving update message and the secret key matches [6]. This is also extended to RIP-2 [7]. It cannot stop rogue advertisements from neighbor routers. It increases the difficulty of router management and is more error-prone to misconfiguration [8]. The secret key is needed to be renewed periodically otherwise it is vulnerable to attacks for MD5.

As can be seen, routing protocols are vulnerable, and a malicious router could misroute packets to a wrong direction, instead of forwarding them to the best direction. This packet mistreatment may result in triangle routing, one of the serious network infrastructure attacks.

Fig.1 shows an example of triangle routing. In the example, all links have a unit cost except the link connecting routers R1 and R3 together. When packets sourced from R1 and targeted on R4, the shortest path should be R1-R2-R3-R4. This is the expected path in normal packet forwarding. However, R3 has been compromised and mishandles the packets by forwarding them back to R1 maliciously. In this case, a routing loop is formed and the packets will circulate until their time-to-live (TTL) expire. This looping not only overloads the routers but also causes extra network traffic. The problem becomes even more intractable if the malicious router only misroutes packets selectively (says only for selected networks or hosts

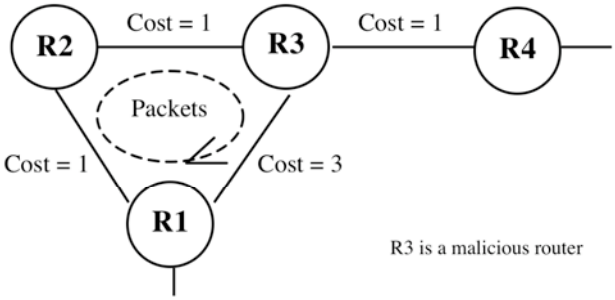


Figure 1. Illustration of Triangle Routing

at random time intervals), or if the number of routers involved in the routing loop is large.

2.1 Impact of Triangle Routing

Triangle routing attack causes different kinds of damages to the network operation.

2.1.1 Network congestion

Fig.2 illustrates how triangle routing causes network congestion in normal routing; R3 is transferring data at *T* bps to R4. This *T* bps of data is originated from R1 and targeted at somewhere behind R4. However, when R3 is malicious and forwards all packets for R4 back to R1 through their shared link (R1-R3), unexpected transfer will be sent to R1. As a result, R1 has to handle these extra packets from R3 and forwards them to R2 as its routing algorithm tells that R2 is the best way to go. This extra *T* bps traffic continues to loop through R1, R2 and R3 in the network until the TTLs of packets expire. The number of times packets need to loop around before expire would be a division of the TTL values *x* by the size of the loop *l*. The amount of traffic caused would be approximately $\frac{x}{l} \times T$ bps. In Fig.2, the loop size *l* is 3, the TTL value *x* is assumed on average 128 and the traffic rate *T* is 5 Kbps, then the total amount of traffic in one second would be 128 / 3 = 42.66 times heavier, resulting in 42.66 × 5 = 213.3 Kb. Incoming data accumulated with the looping data creates an avalanche effect that R1, which cannot handle this unexpected loading, will become a bottleneck in network and discard incoming packets, resulting in network congestion. Consequently, the throughput of victim routers is lowered.

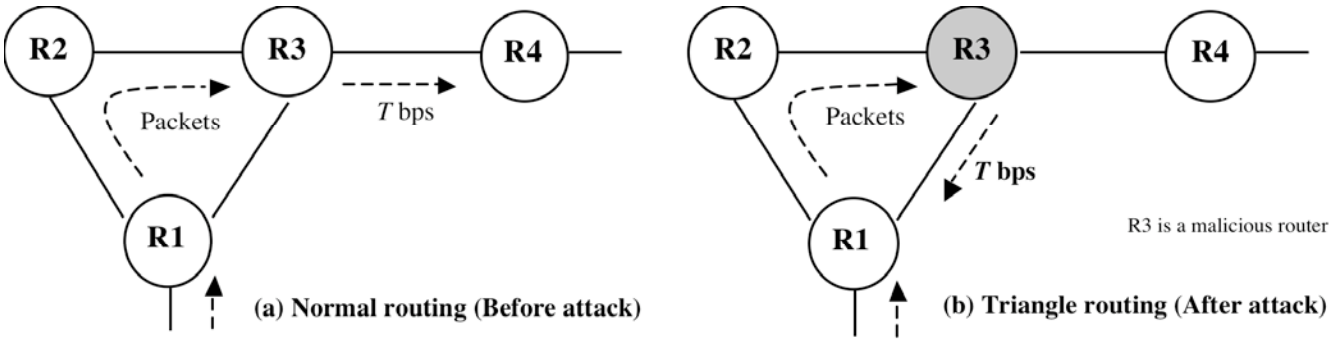


Figure 2. Network Congestion at R1 caused by Triangle Routing Attack

2.1.2 Denial of Service (DoS)

The network congestion caused by extra traffic can create denial of service attack to victim routers by disrupting their services in the network. For example, in Fig.2, R1 may not be capable of handling the unexpected load caused by triangle loop, hence start to discard incoming packets. Denial of service may occur in R1.

2.1.3 Network Partitioning

Another problem caused by triangle routing attack is that network will be partitioned into different parts if there is a malicious border router. No traffic is allowed to get through the malicious router, hiding up the area from the production network. The connectivity of the area behind R4 and availability of network services in the area depends on the degree of how R3 is malicious. In the worst case, R3 may block all ingoing traffic for the area.

3 Detection of Triangle Routing

In each IP packet, there is a time-to-live (TTL) value specifying the maximum number of hops it is allowed to pass in the Internet. Each router will decrement the value by one when a packet passes through it. If a packet's TTL value expires (becomes zero), the router simply disposes it and then sends an ICMP *time exceeded* packet back to the original sender of the expired packet. The triangle routing attack traps packets in a loop until their TTL values become zero.

3.1 Path Tracking

A solution proposed in this paper for the detection of triangle routing attack is by means of the analysis of the path packets have traveled. Packets fallen into a triangle loop will be reflected from their traveling path if repetition of passing through some particular routers occurs. A packet is then required to record down the nodes it has visited and returns its path to the sender when it expires. For example, the shortest path from R1 to R4 is R1-R2-R3-R4. However, the malicious router R3 redirects all packets for R4 to R2, forming a routing loop R1-R2-R3. The traveling path of packets shown when TTL expires will be …R1-R2-R3…R1-R2-R3…. If the path is returned, repetitions of pattern R1-R2-R3 can be found and triangle routing is identified.

We will first discuss how our solution in IPv4 returns the tracked path. We will also discuss the limitations that make our solution in IPv4 less practical. Following up will be a solution in IPv6 that overcomes those limitations occurred in IPv4.

3.2 Path Returning

To return the tracked path, the record routing feature defined in Options field of IPv4 header can be used. When the record-route option is set in a packet, the router will insert its IP address in the Options field of the packet.

The IPv4 Options field is an optional part of the IP header that instructs routers processing the packet to do some extra processing other than routing. It is primarily designed for network testing and debugging. One of the options that defined to facilitate this integral part of IP protocol is called

Record Routing. It reserves spaces in header for routers IP addresses. When a packet reaches its destination, a list of routers it has traveled is then shown in its header, thus the path is recorded.

Our proposed triangle routing detection can be achieved by using the ping program, a basic tool provided in most operating system. The ping program detects the availability of a machine by sending an ICMP echo request to the testing machine. The record-route enabled datagram travels along the network and arrives at the destination if the destination machine is available. The destination machine is then returned an ICMP echo reply to the sender by echoing the received datagram including the IP header. If the sender receives the reply, the list of routers the request datagram visited is obtained. In case of triangle routing, the echo request will never find the destination machine and loop until TTL expires. A router receiving the TTL expiring packet then generates the ICMP unreachable reply to the sender along with the path the echo request has traveled in the IP header as the path information stays in the IP header. Thus, sender can `detect the triangle routing attack in the network in the returned path list and identify the malicious router.

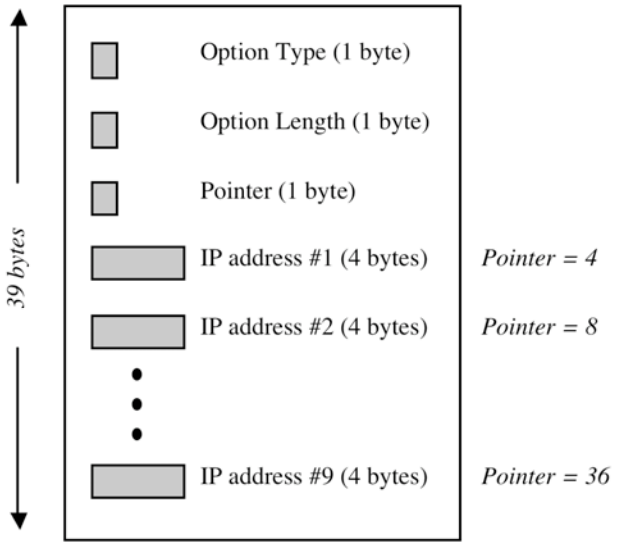


Figure 3. Options Field in IP Header

3.3 Limitations of Path Returning

The path tracking detects triangle routing problem with the built-in feature of IPv4. Though it is simple and immediately available in IPv4 networks, it suffers from a number of problems.

3.3.1 Security Issue

Security is an important issue that network administrators must consider in their network design. The record-route option in IP header shares the some Options field with another optional feature called source-route option. The source-route option allows packets to have its traveling path specified in advance before it is sent. This is a useful tool for path testing. However, this optional feature will create a security hole in network. A target host under the IP spoofing attack will reply to adversaries instead of the connection originator if a return path is specified. This allows adversaries to illegally access resources available to victim machines [9]. Bellovin also suggested two methods to defend against this attack: 1) Gateways inspect external packets and dispose those claimed to be local, to get into local network segment. It stops attack initiated externally. 2) Gateways may also determine if the routing list in header contains untrusted nodes before they let go. Nevertheless, many administrators found the methods complex and would prefer not providing this source routing service in the network to using the above methods. Therefore, the record routing feature is also disabled.

3.3.2 Length of Path

Our solution is based on the use of the Options field in IPv4 header. The IPv4 header includes this optional field item by extending its length to a number expressed in a 4-bit header length field in basic header, limiting the maximum header length to 15×4 bytes. The basic header takes off 20 bytes, leaving 40 bytes to the Options field. Within these 40 bytes, the record-route option is defined and indicated by the beginning 3 bytes. The remaining 37 bytes then allows only 9 IP addresses for router IPs. In other words, the maximum length of a path can have 9 router IP addresses only, limiting our solution to be effective in an autonomous system consisting 9 routers only. Therefore, this solution is suitable for use in the Intranet only in which 9 core routers are commonly enough.

3.3.3 Extra Router Loading

In addition to basic routing, a router will perform extra actions on the packets when it sees any option has been set in the Options field of the packets. Routers detect the option setting in header and decide which action they should take for the packets. As can be seen, extra computing resources is needed

to process the Options field. Packets being processed will also need to spend more time in routers. This increases the amount of time packets staying in network and creates an avalanche effect of delay. The delay would introduce a jitter problem to the network if it is designed for multimedia data transmission. Network performance is degraded to some degrees that network administrators would prefer to disable this extra processing function in their routers and optimize them for high efficient data delivery. Thus, no router IP address will be stored in a path and returned to the sender [10]. In this case, our proposed detection cannot be achieved. Therefore, the success of the triangle routing detection operation requires routers in the network to be Options field processing enabled.

3.3.4 Packet Discarding

The record-route option is vulnerable to packet discarding attack. An intermediate router simply discards Options-enabled packet as it passes through will disrupt the operation of the detection. The path will never come back.

As can be seen that the record routing feature in IPv4 implements our path analysis method but with some limitations. In the next section we will study the triangle routing attacks in IPv6 network. Our path analysis method in IPv6 detects triangle routing attacks and removes the limitations occurred in IPv4.

4 Triangle Routing in IPv6

IPv6 has a lot of improvement over IPv4. One of the significant upgrades is to provide security services to applications. Authentication of routing algorithms (RIP-2 or OSPF) has been removed and they rely on IPv6 to ensure integrity and authentication. Likely to the authentication in OSPF, a message digest in Authentication header is computed on a combination of the payload and a secret key for each message. Message is only accepted if the verification of the message digest with the right secret key on the receiving side is successful [11]. Unfortunately, it still suffers the same potential problems in IPv4 discussed in previous section. Thus, triangle routing attacks exist in IPv6 network.

IPv6 has a big change in organizing its functionalities in the IP header over IPv4. Our path analysis detection method works in a more flexible way. However, the record routing feature in IPv4 is not defined in IPv6 yet. Therefore, we will propose a new extension header in IPv6, called Record Path Header, for the implementation of path analysis method. This new header stores the path information of a packet as it travels and returns the stored information back to the initiator as the packet reaches its destination or expires.

4.1 Record Path Header

The Record Path header is used to reserve spaces in IPv6 packet for the IP addresses of its visited nodes. Each node processing this extension header will insert its IP addresses into the next available space. Once a packet arrives at its destination, all visited nodes in its traveling path will be obtained. Recalled that the record routing feature in IPv4 is not defined in IPv6, having the Record Path header will enable IPv6 with this specific feature. Fig.4 shows the format of the Record Path header

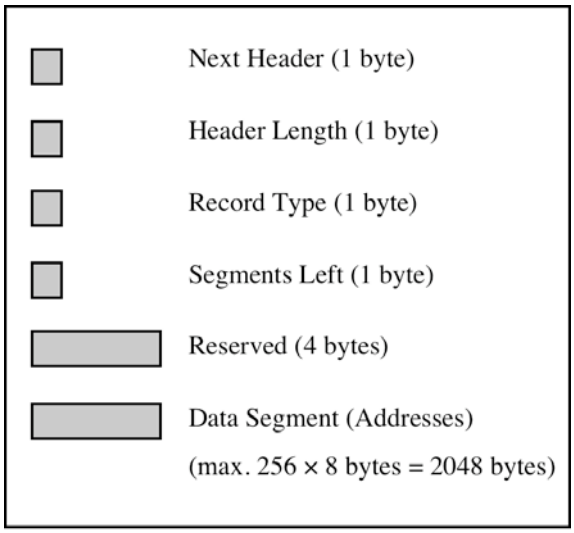


Figure 4. Record Path Header

Each field is described as follows.

Next Header – In IPv6, headers are indicated by the immediate preceding header in the Next Header field. This field identifies the type of header following the Record Path header. The Next Header value of Record Path header is proposed to be 136.

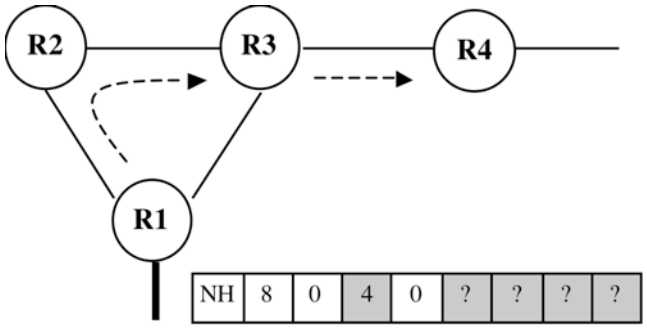
Header Length – This field shows the length of the header in a unit of 8-byte excluding the first 8 bytes, as defined in IPv6. The maximum size of the header is 2056 bytes.

Record Type – The use of the Record Path header is only defined to do record routing feature for the time being and this feature is indicated in this field by a value of zero.

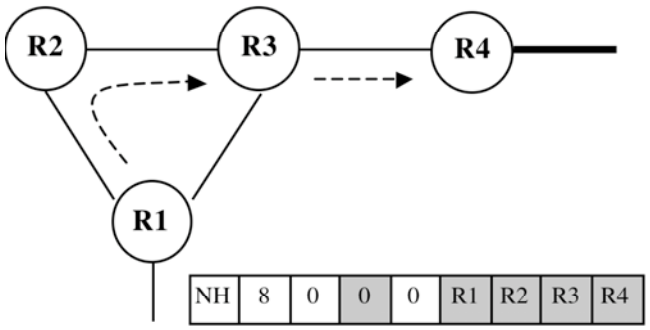
Segments Left – This field shows the number of router addresses can be inserted into this header. The data segment can accommodate at most 128 IPv6 addresses. The maximum value of this field is 128 and a value of zero indicates that no more IP address can be inserted into. Initially, the size of the header is defined by the originator of the packet.
Reserved – This field is not in use and is set to zero.
Data Segment – Router addresses are lined up here and the length of this field depends on the number of addresses expected to be inserted.

Fig.5 shows how the Record Path header works among four routers R1, R2, R3 and R4. A path showing that a packet is going through these routers will be recorded in the order R1-R2-R3-R4 by the Record Path header. The header, before entering R1, will be initiated as shown in Fig.5a. The header of length 8 shows that the packet is going to collect four IPv6 addresses. Therefore, a value of 4 is shown in Segments Left field and nothing is recorded yet. When the packet enters R1, R1 will insert its IP address into the first slot and decrement the Segment Left value by one. When the packet enters R2, R2 will do the same as R1 by inserting its IP address into the next available space and decrementing the Segment Left value by one. The packet will have all router IP addresses listed after it goes through all routers, as shown in Fig.5b. Table 1 shows the details of the Record Path header in Fig.5.

In IPv6, a single packet is able to have more than one extension header provided that the header order is maintained. To be compatible in IPv6 and working in harmony with other extensions headers, the Record Path header follows the header order. First, the Record Path header must exist after the base header as other headers do. Second, the Record Path header must exist before the fragment header, authentication header and encapsulating security header.



(a) Before entering R1



(b) After leaving R4

Figure 5. Record Path Header in a Packet

When a Record Path header is found in a packet, the processing node will look at the Record Type field. If the Record Type is zero, it determines the next available space for its IP address as there is no field to tell where the node's IP address goes to. Simple calculation on fields such as the Header

After leaving	NH	HL	RT	Seg	Rvd	Data Segment			
R1	-	8	0	3	0	R1	?	?	?
R2	-	8	0	2	0	R1	R2	?	?
R3	-	8	0	1	0	R1	R2	R3	?
R4	-	8	0	0	0	R1	R2	R3	R4
	1B	1B	1B	1B	4B	Header Length			

Table 1. Record Path Header Details

Length and Segment Left are needed. The algorithm below shows how a node finds the next available space for its IP address in the Record Path header.

The algorithm in Table 2 shows clearly that the processing continues if the node finds Record Type to be zero and IP address space is available (Segment Left is greater than zero only). Otherwise, it ignores the header. Let's take the router R2 in Fig.5 as an example. The first row in Table 1 is the Record Path header before entering R2. The calculation of the next available space is as follows.

- The total number of IP addresses allowed, T = (8 * 8) / 16 = 4
- The current IP position, C = 4 – 3 = 1
- The location in Data Segment, L = 1 * 16 = 16

The processing node then locates the position in Data Segment at byte 16 and inserts its IP address. After the decrement of Segment Left by one, the Record Path header processing is finished. The second row in Table 1 is the result of Record Path header after leaving R2.

4.1.1 Processing Rules

It is noteworthy that rules are needed to be followed during the processing of the Record Path header. The following error conditions may arise:

a) If the Record Type field is not recognized, the processing node simply ignores the header, forwards the packet as desired and sends an ICMP Parameter Problem message of code two to the originator of the packet.
b) If inconsistency between data in the fields occurs (Eg., the size of the Data segment field is not equal to the Header Size field), the processing node disposes the packet and returns an ICMP Parameter Problem message to the originator of the packet.
c) If the next link MTU size is too small for a packet, the processing node discards the packet and returns an ICMP Packet Too Big message to the originator of the packet.

4.1.2 Packet Size Issues

In IPv6, the minimum size of MTU of a link in Internet is required to be 1280 bytes [12]. It implies that the maximum packet size used on the path to the destination may be 1280 bytes only. The packet is then required to be restricted in size in order to meet the supported MTU. The Record Path header size is required to be limited and lesser IP addresses are collected in the header.

In our proposal, the triangle routing detection is operational on an ICMPv6 Echo Request packet. The recorded path in the header returned by an ICMPv6 packet (Echo Reply or Time Exceeded message) is analyzed for the occurrence of the routing problem. In this section, we will discuss how the Record Path header exists in an ICMPv6 packet over a link of minimum MTU size. Fig.6 shows the format of an ICMPv6 packet of size that meets the minimum MTU restriction.

The size of the ICMPv6 message as restricted by MTU is 1280 bytes. It is composed of some parts. The IPv6 Base header takes 40 bytes which is a minimum. The ICMPv6 message contains its header only which is eight bytes long. After the eight bytes of some fields in Record Path header, the packet has 1224 bytes left for the Data Segment of the Record Path header. As the size of IPv6 address is 16 bytes, the maximum number of node's IP address can be stored is approximately 76. Compared with the Record Path header of full size, around 60% of IP

```
If the Record Type = zero and the Segment Left > zero, then
 Calculate the total number of IP addresses is allowed:
    T = (Header Length * 8) / Size of an IP address
 Calculate the current IP position: C = (T – Segment Left)
 Calculate the location in the Data Segment: L = C * 16
 Insert the node's IP address into the Data Segment at L
 Decrement the Segment Left by one
Else
 Ignore the header
```

Table 2. Processing Algorithm for Record Path Header

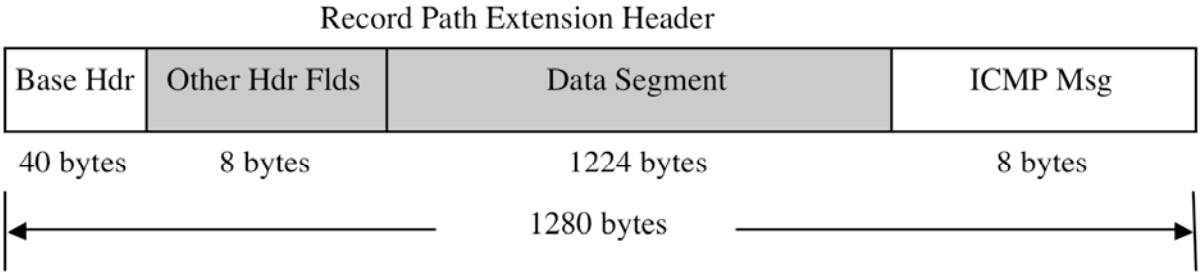


Figure 6. Record Path Header in IPv6 Packet

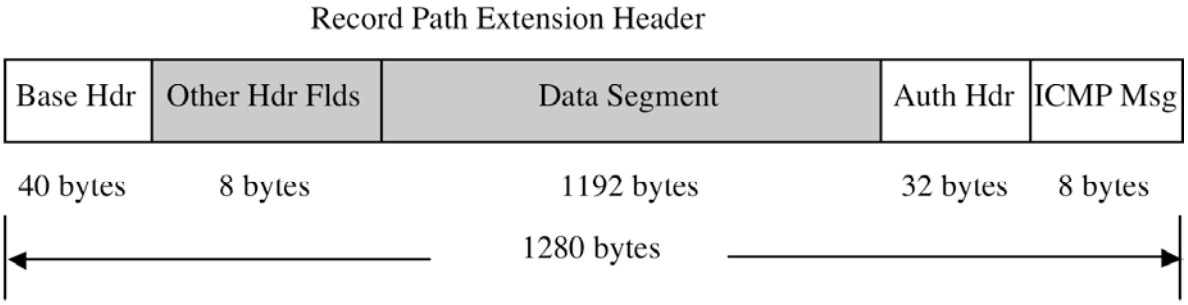


Figure 7. Record Path Header in IPv6 Packet Authentication Using SHA-1

address storage is still maintained.

In case, authentication service is provided for the detection message. The Authentication extension header will be inserted into the packet. An Authentication extension header contains some fields of 12 bytes altogether, indicating a fixed sequence of protocol elements, and a variable size of authentication data [11]. The total size of this authentication header is determined by the cryptographic algorithm used in authentication. Two algorithms must be supported by every implementation by default: MD5 and SHA-1. The MD5 will produce a 128-bit message digest whereas the SHA-1 produces a 160-bit message digest. Fig.7 shows that 74 IP addresses (approximately takes 1184 bytes) are allowed in Record Path header even though authentication is provided in the packet using SHA-1 algorithm.

The Record Path header is not defined to be strictly used with ICMPv6 messages. Any messages that require collecting IP address of each processing node may employ this header. Therefore, if the packet to be sent is larger than the supported MTU, packet fragmentation is needed. Unlike IPv4, the packet fragmentation in IPv6 only occurs on the sending hosts. Intermediate routers do not do fragmentation. Each fragmented packets contains a Fragment header to maintain their sequence and identification for reassembly in the destination. The Record Path header is an unfragmentable part in a packet as it is processed by nodes along the path to the destination. In case of fragmentation, the Record Path header will appear in every fragment, followed by the Fragment header and then the fragmentable data.

4.2 Advantages over IPv4

The Record Path extension header for the triangle routing detection in IPv6 has advantages over the solution in IPv4.

4.2.1 Security Issue

The source-route option in IPv4 exposes the weakness of a network and it may cause IP spoofing attack as mentioned in previous section. Our proposed IPv4 solution employs the record-route option in the Options field in header. Once, the source-route option is disabled, the Options field in header is then unrecognized in the node. Our proposal is led to be unfeasible. Our proposed record-route option in IPv6 overcomes this limitation by storing the path in Record Path

extension header. The security issue addressed is no longer applicable as the source-route option in IPv4 is handled by the Routing extension header in IPv6. Disabling the Routing header in IPv6 network will not interrupt the operation of the triangle routing detection as different extension headers are used.

On the other hand, the Record Path header does not require any security service in IPv6. Likely to the Hop-by-hop extension header or other extension headers that requires header updates in each processing node, the security services provided in IPv6 is not necessary either in transport mode or tunnel mode. However, for some reasons, it is possible. Intensive computation for security services is required whenever the header has changes in each processing node. Thus, it degrades the network performance. Nevertheless, solutions for efficient router designs are available. [13][14][15]

4.2.2 Sufficient Length of Path

In IPv4, there is inherent limitation to our detection solution. The length of the path is limited to nine nodes only. A triangle routing loop of size larger than nine nodes would make our solution infeasible. The Record Path header changes the situation completely. The length of the path can be up to 128 nodes, which is 14 times longer than the path in IPv4. In case, the packet size is limited by the MTU of a link, a size reduction of the Record Path header may occur. A possible size of path can still be maintained at 74. To some extent, the path length limitation is removed.

4.2.3 Dependency Issue

Recalled that our path analysis implementation in IPv4 employs the Options field in header. Disabling the Options field will turn down our path analysis solution. Therefore, the processing of the Options field in IPv4 header is an important issue to our detection method. Our newly defined Record Path header in IPv6 removes such dependency. The implementation of the path analysis on this new header works on its own and does not rely on any other extension header.

5 Conclusion

The triangle routing detection method we proposed analyzes the traveling path of packets. The path tells that the triangle routing attacks occur in a network if some particular patterns can be observed. The implementation in IPv4 employs the record routing feature in IPv4. It is easy to use and available in most of the operating systems. However, this IPv4 solution has some limitations: security, loading, length of path and packet discard.

This paper has also presented that triangle routing attacks exist in IPv6 network and the path analysis is used. To enable IPv6 with the path analysis feature, we proposed a new extension header, called Record Path header. This new header does the record routing job in a more flexible way and does not have the limitations occurred in IPv4. It resolves the security issues of record routing addressed in IPv4 as IP spoofing is no longer an obstacle to our solution. It provides sufficient space for path tracking. The path analysis implemented by the Record Path header is no longer disabled by network administrators due to the router loading issue on the processing of record-route enabled packets.

Our IPv6 solution vitalizes the triangle routing detection method by removing the limitation occurred in IPv4. However, the packet discard problem is still an issue to our path analysis detection method. In the future, our solution will be extended with a correctness verification by mathematical analysis or simulation experiments in software routers.

6 Acknowledgement

The authors would like to acknowledge the research grant (No.: RP/ESAP-3/2008) offered by Macao Polytechnic Institute to support for this project.

References:

[1] K. Bradley et al., Detecting Disruptive Routers: A Distributed Network Monitoring Approach, *IEEE Network*, Vol. 12, No. 5, 1998, pp.50-60.

[2] A. Chakrabarti and G. Manimaran, Internet Infrastructure Security: A Taxonomy, *IEEE Network*, Vol. 16, No. 6, 2002, pp. 13-21.

[3] K. H. Yeung, F. Yan and C. Leung, Improving Network Infrastructure Security by Partitioning Networks Running Spanning Tree Protocol, *International Conference on Internet Surveillance and Protection*, Cote d' Azur, France, Aug., 2006, pp. 19.

[4] D. Montgornery and S. Murphy, Toward Secure Routing Infrastructures, *IEEE Security & Privacy*, Vol. 4, No. 5, 2006, pp. 84-87.

[5] C. Hedrick, Routing Information Protocol, *RFC1058*, 1988.

[6] J. Moy, OSPF Version 2, *RFC2328*, 1998.
[7] G. Malkin, RIP Version 2, *RFC2453*, 1998.
[8] P. Papadimitratos and Z. Haas, Securing the Internet Routing Infrastructure, *IEEE Communications Magazine*, Vol. 40, No. 10, 2002, pp. 60-68.
[9] S. Bellovin, Security problems in the TCP/IP Protocol Suite, *ACM SIGCOMM Computer Communication Review*, Vol. 19, No. 2, 1989, pp. 32-48.
[10] P. Fransson and A. Jonsson. End-to-End Measurements on Performance Penalties of IPv4 Options, *Global Telecommunications Conference*, Dallas, USA, Nov. 2004, pp. 1441-1447.
[11] S. Kent, IP Authentication Header, *RFC2402*, 1998.
[12] S. Deering, Internet Protocol, Version 6 (IPv6) Specification, *RFC2460*, 1998.
[13] A. Kuznetsov and S. Berkovich, Router Architectures Using Combinatorial Designs, *WSEAS TRANSACTIONS on COMPUTERS*, Vol. 5, No. 12, Dec. 2006, pp. 2956-2961.
[14] O.-I. Lepe-Aldama and J. Garcia-Vidal, Characterizing and Modeling a PC-Based Software Router, *WSEAS TRANSACTIONS on INFORMATION SCIENCE and APPLICATIONS*, Vol. 1, No. 1, Jul. 2004, pp. 35-44.
[15] D. Vasiliadis, G. Rizos, L. Tsiantis, S. V. Margariti, Comparative Study of blocking mechanisms for Packet Switched Omega Networks, *Proceedings of the 6th WSEAS International Conference on ELECTRONICS, HARDWARE, WIRELESS and OPTICAL COMMUNICATIONS (EHAC '07)*, Corfu Island, Greece, Feb. 2007, pp. 18-22.