A Novel Protocol for IP Traceback to Detect DDoS Attack

Yogesh Kumar Meena[1], Aditya Trivedi[2]

**[1] Hindustan Institute of Technology and Management,
Agra, UP, India**

**[2] ABV-Indian Institute of Information Technology and Management,
Gwalior, MP, India**

Abstract

Distributed Denial of Service (DDoS) attacks continue to pose higher threats to the internet. There are so many protocols designed to trace the attacker's address. We want to trace back attack source (i.e., "IP addresses"), we need to examine the tradeoff between different existing IP Trace back techniques. We developed a Novel protocol to trace the IP address of DDoS attack. The novel protocol is designed by using response 1, Nonce of secure- neighbor as the parameters. We developed a sample network model. We simulate the network model by applying secure-neighbor protocol in Qualnet. Through secure-neighbor, we retrieve the basic parameter value (Response 1, Nonce) and apply the decryption function on Nonce and value of neighbor-timeout to find the attackers IP address. We studied different internet topologies and aspect of DDoS attacks, used internet power low for the simulation of the internet.

Keywords: *Denial of Service, Distributed Denial of Service, Novel protocol to trace IP address, Secure-Neighbor.*

1. Introduction

This network attack have emerged as an important field in the research areas. In distributed denial-of-service (DDoS) attack, an attacker may use a computer to attack another computer. By taking advantage of security vulnerabilities or weaknesses, an attacker could take control of a computer. The attacker could then force a computer to send huge amounts of data to a website or send spam to particular email addresses. The attack is "distributed" because the attacker is using multiple computers, including yours, to launch the denial-of-service attack. Attackers use spoofed source addresses to hide their identity and location in Distributed Denial of Service (DDoS) attacks [1]. Some service providers do perform ingress filtering to check for valid source IP addresses coming into access routers, but this is not completely effective. Recent studies show source address spoofing is still a major network problem [2], [3]. Traceback mechanisms [4-8] trace the true source of the attackers to stop the attack at the point nearest to its source to reduce waste of network resources and to find the attackers' identities. The DoS attacks can be classified into two main categories: (i) Flood attacks (ii) Logic or software attacks. In Fig. 1, we have shown the simple architecture of Distributed Denial of Service (DoS) attack model.

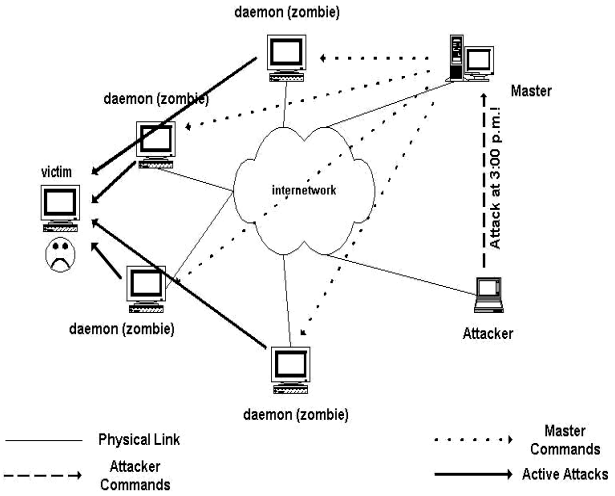


Fig. 1 A Distributed Denial of Service (DoS) attack

IP traceback is a name given to any method for reliably determining the origin of a packet on the Internet. Due to the trusting nature of the IP protocol, the source IP address of a packet is not authenticated. As a result, the source address in an IP packet can be falsified (IP address spoofing) allowing for Denial Of Service attacks (DoS) or one-way attacks (where the response from the victim host is so well known that return packets need not be received to continue the attack. There are two types of IP traceback (i) IP traceback for Direct DDoS (ii) IP traceback for reflector attacker [4]. In Fig. 2, we have shown the architecture of IP traceback for Direct DDoS and reflector attacker.

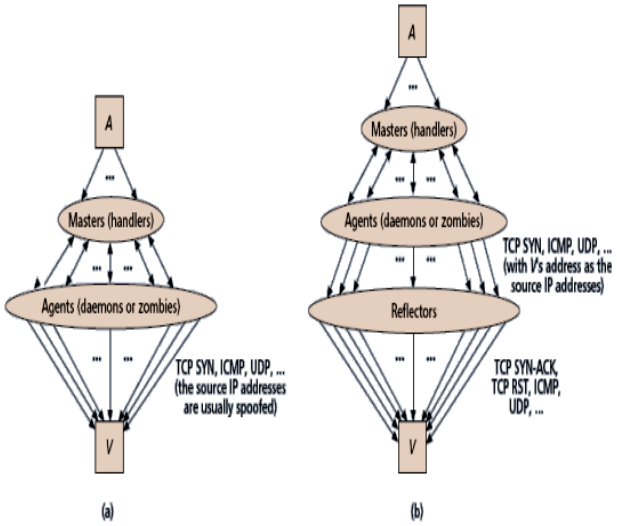


Fig. 2 IP traceback for Direct DDoS and reflector attacker

Houle and Weaver[1] is to highlight recent trends in the deployment, use and impact of DoS attack technology based on intruder activity and attack tools reported to and analyzed by the CERT/CC. This paper does not propose solutions, but rather aims to serve as a catalyst to raise awareness and stimulate further discussion of DoS-related issues within the Internet community.

Robert and Steven [2] Presents an Internet-wide active measurement spoofing project. Clients in our study attempt to send carefully crafted UDP packets designed to infer filtering policies. When filtering of valid packets is in place we determine the filtering granularity by performing adjacent net block scanning.

David Moore [3] present a detailed study of the source code of the popular DDoS attacks bots, Agobot, SDBot, RBot and Spybot to provide an in-depth understanding of the attacks in order to facilitate the design of more effective and efficient detection and mitigation techniques.

Morris and Naranker [4] describes a Non-Intrusive IP traceback scheme which uses sampled traffic under non-attack conditions to build and maintains caches of the valid source addresses transiting network routers. Under attack conditions, route anomalies are detected by determining which routers have been used for unknown source addresses, in order to construct the attack graph.

Savage[5]suggested probabilistically marking packets as they traverse routers through the Internet. They propose that the router mark the packet with either the router's IP address or the edges of the path that the packet traversed to reach the router.

Song and Perrig [6] identify that this is not robust enough against collisions and thus suggest using a set of independent hash functions, randomly selecting one, and then hashing the IP along with a FID or function id and after that encoding this. They state that this approach essentially reduces the probability of collision to (1/(211)m).

Snoeren[10] propose marking within the router. The idea proposed in their paper is to generated a fingerprint of the packet, based upon the invariant portions of the packet (source, destination, etc.) and the first 8 bytes of payload (which is unique enough to have a low probability of collision). More specifically, m independent simple hash functions each generate an output in the range of 2n-1. A bit it is then set to the index generated to create a fingerprint when combined with the output of all other hash functions. All fingerprints are stored in a 2n bit table for later retrieval.

I studied different internet topologies and aspects of DDoS attacks, used internet power law for simulation of the Internet. The objective of the analysis is searching the protocol which can be helpful in tracing back the source of distribution denial of service attacks.

The motivation for this work comes from the fact that if one becomes the victim, what can be done to make the harder target to take down and as an alert system already mentioned, it is intended to speed the process of tracking down such attacks.

Rest of the paper is organized as follows: In section II presents our research methodology; in this we describe secure neighbor protocol and our proposed novel protocol for IP trackback on DDoS attacks. In section III network modal scenario and simulation result. Conclusion and future work of our novel protocol for IP trackback on DDoS attacks are in section IV.

2. Research Methodology

Since any system is prone to be affected by DDoS attack, the objective of our analysis is to search an algorithm which can be helpful in tracing back the source of DDoS attacks. The idea is to use different simulators related to power low for simulation of Internet networks.

2.1 Secure-neighbor protocol

In secure neighbor authentication (SNAuth), every mobile node establishes an authenticated neighborhood on the move. Periodically, every mobile node X broadcasts its identity packet <SNAuth- HELLO, X> to its neighborhood. In the pair-wise shared secret variant of SNAuth, Y, a neighboring receiver of the identity

broadcast initiates a 3-way challenge-response handshake to authenticate X, the sender of the identity broadcast.

a) Suppose X and Y share a pair-wise secret k. Now Y selects a random Nonce n1, encrypts n1 with k, sends the encrypted result ENCk (n1) to X by a message <CHALLENGE, Y, ENCk (n1)>.
b) If the receiver of the challenge message is indeed X, then it can decrypt ENCk (n1) and sees n1. X selects another random Nonce n2, encrypts ENCk (n1 XOR n2), and sends back <RESPONSE1, X, n2, ENCk (n1 XOR n2)> as the response to the challenger Y.
c) When Y receives the response, Y decrypts ENCk (n1 XOR n2) and obtains n1 XOR n2. If Y can get the same result from XORing n2 in the response and its own challenge n1, then X passes the test with success. Otherwise, Y does not send any packet to X and does not receive packets from X except the response packets, until a correct <RESPONSE1> packet from X can pass the test. Upon detecting a success, Y puts X in its secure neighbor list.
d) The cryptographic term, "Nonce" is used above to mean a value that is used only once. All Nonce length is currently set to 128-bit long. Encryption block length is 128-bit. Key k can be 128-bit, 192-bit, or 256-bit. Session key means that the key n1 is used until the time when the next HELLO received by Y from X successfully passes the test again.

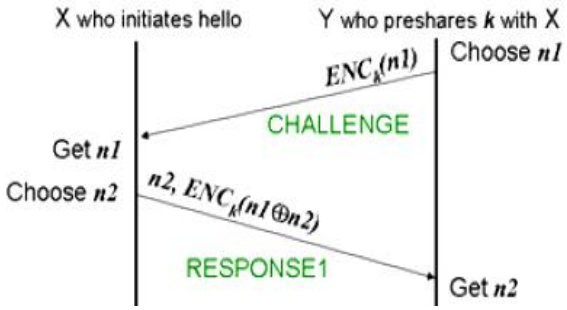


Fig. 3 Secure Neighbors challenge-response protocol

2.2 Proposed Protocol

We propose a protocol to trace the IP address of the attacker who attacked the system as shown in fig (4). To draw this protocol we taken the basic parameter values from the secure-neighbor protocol like Responce1 of particular node and the Nonce metric values

(IP-trace secure protocol)

Step1: Record the metric value (T) at which the attack is takes place for node-x.

Step2:

i. Retrieve the values of Response1 from secure-neighbor protocol, from which we can take the values of how many messages are forwarded to a particular node.
ii. Take the Nonce value of node

Step3: Decrypt the value 'n1' value with 'n2' value of nodes

i. Check the value of dec_{PK} (n1 xor n2) is equal to T. If so record the IP address of node.
Where
n1 is the Nonce value
n2 is the value of the neighbor-timeout of node.

ii If dec_{PK}(n1 xor n2) is not equal to T

Increment the value of n2 by x up to the simulation time and repeat the process.

Where x denote the default value of the neighbor-timeout of node (we taken it as 5sec)

Step4: If the dec_{PK}(n1 xor n2) is not satisfied for any value of n2 of a particular node then repeat the whole process for another node.

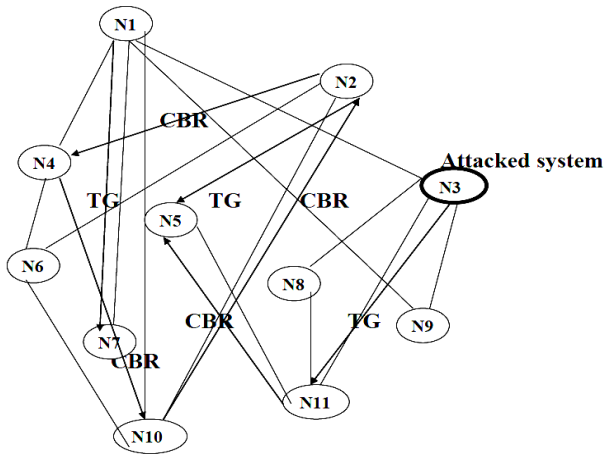


Fig. 4 Network model

3. Simulation Environment and Results

The logical operation exclusive disjunction, also called exclusive or XOR or ⊕, is a type of logical disjunction on two operands that results in a value of true if exactly one of the operands has a value of true

In our case we used XOR for define proposed modal equations like $dec_{pk}(n1 \oplus n2) = = T$

Where, n1 is the Nonce value, n2 is the value of the neighbor-timeout of node and pk is private key.

3.1 Simulation Parameter and design

A. Scenario

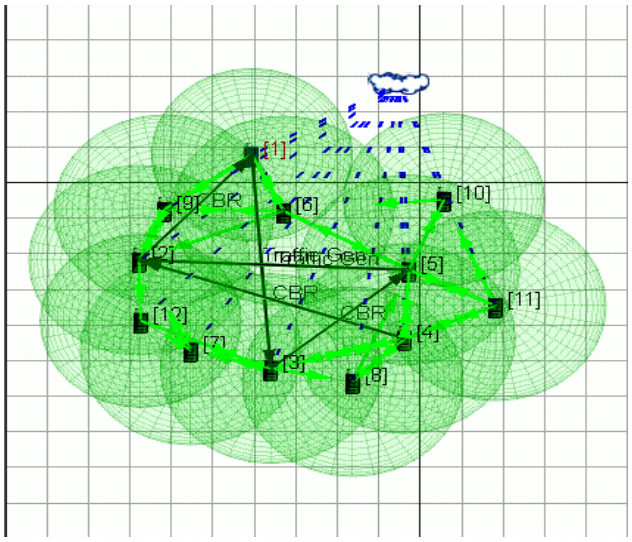


Fig. 5 Scenarios

In this scenario we consider 12 nodes and we assign the unique IP address to each under the wireless subnet.

In node properties we will tack router type Cisco 7603.

B. Routing Protocol

- *AODV*

AODV allows mobile nodes to obtain routes quickly for new destinations, and does not require nodes to maintain routes to destinations that are not in active communication. AODV allows mobile nodes to respond to link breakages and changes in network topology in a timely manner. It uses sequence numbers to prevent routing loops.

To configure the AODV parameters, perform the following steps:

1. Go to one of the following locations: To set properties for a specific wireless subnet, go to Wireless Subnet Properties Editor >Routing Protocol > General.

i. To set properties for a specific wired subnet, go to Wired Subnet Properties Editor >Routing Protocol > General.
ii. To set properties for a specific point-to-point link, go to Point-to-point Link Properties Editor> Point-to-point Link Properties > Routing Protocol.
iii. To set properties for a specific node, go to Default Device Properties Editor > Node Configuration > Routing Protocol.
iv. To set properties for a specific interface of a node, go to one of the following locations:
   - interface Properties Editor > Interfaces > Interface # > Routing Protocol
   - Default Device Properties Editor > Interfaces > Interface # > Routing Protocol.

In this section, we show how to configure AODV parameters for a specific node using the Default Device Properties Editor. Parameters can be set in the other properties editors in a similar way.

C. Multicast Routing Protocol

- *DVMRP*

DVMRP is a multicast routing protocol. It is designed for traditional wired network multicast routing, and operates similarly to a distance vector routing protocol like RIPv2 (Routing Information Protocol Version 2). DVMRP is a tree-based, multicast scheme that uses Reverse Path Multicasting (RPM).

To configure the DVMRP parameters, perform the following steps:

1. Go to one of the following locations:

i. To set wireless subnet properties, go to Wireless Subnet Properties Editor > Routing Protocol.
ii. To set properties for a specific node, go to Node Properties Editor > Node Configuration >Routing Protocol.
iii. To set properties for a specific interface of a node, go to one of the following locations:
   - Interface Properties Editor > Interfaces>Interface#> Routing Protocol.

- Default Device Properties Editor> Interfaces > Interface # > Routing Protocol.

In this section, we show how to configure DVMRP parameters for a specific node using the Default Device Properties Editor. Parameters can be set in the other properties editors in a similar way.

3.2 Static Multicast Scheduling

Static multicast routes are user-configured multicast routes. User can configure these routes in multicast static route file. Our simulator Static Multicast Routes model supports both IPv4 and IPv6.

A. Command line Configuration

To enable static multicast routes, include the following parameter in the scenario configuration (.config) file:

[<Qualifier>]MULTICAST-STATIC-ROUTE YES

The scope of this parameter declaration can be Global or Node. See General Format of Parameter Declaration for a description of <Qualifier> for each scope.

By default, static multicast routes are not enabled.

B. Static Multicast Route Parameters

Table 1: Static Multicast Routing Parameters

Parameter	Value	Description
MULTICAST-STATIC-ROUTE-FILE Required Scope Global, Node	File name	Name of the multicast static route file. The format of the static route file is described in See Format of the Static Multicast Route File..

C. Format of the Static Multicast Route File

Each line of the static multicast route file has the following format:

<nodeID><source-address><multicast address><interface-addresses>

Where:

<node ID> Node ID.
<source-address> Source address.
<multicast-address>Destination multicast group address.
<interface-addresses>List of space-separated outgoing interface addresses.

Examples

1. The following is an example of a static multicast route file for an IPv4 network. Node 1 will forward each multicast packet from source 192.168.0.1 to multicast group destination 225.0.0.1 on outgoing interface 192.168.0.1. Node 2 will forward each multicast packet from source 192.168.0.1 to multicast group destination 225.0.0.1 on outgoing interfaces 192.168.0.2 and 192.168.1.2.

1. 192.168.0.1 225.0.0.1 192.168.0.1

2. 192.168.0.1 225.0.0.1 192.168.0.2 192.168.1.2

2. The following is an example of a static multicast route file for an IPv6 network. Node 1 will forward each multicast packet from source 1000:1::1 to multicast group destination ff12::3 on outgoing interface 1000:1::1. Node 2 will forward each multicast packet from source 1000:1::1 to multicast group destination ff12::3 on outgoing interfaces 1000:2::1 and 1000:5::1.

1. 1000:1::1 ff12::3 1000:1::1

2. 1000:1::1 ff12::3 1000:2::1 1000:5::1

File we created for our network modal Multicast statics IP **.multicast-static**

1. 190.0.1.1 225.0.0.1 190.0.1.1
2. 190.0.1.2 225.0.0.1 190.0.1.2
3. 190.0.1.5 225.0.0.1 190.0.1.5
4. 190.0.1.4 225.0.0.1 190.0.1.4
5. 190.0.1.3 225.0.0.1 190.0.1.3
6. 190.0.1.6 225.0.0.1 190.0.1.6
7. 190.0.1.8 225.0.0.1 190.0.1.8
8. 190.0.1.7 225.0.0.1 190.0.1.7
9. 190.0.1.9 225.0.0.1 190.0.1.9
10. 190.0.1.10 225.0.0.1 190.0.1.10
11. 190.0.1.11 225.0.0.1 190.0.1.11
12. 190.0.1.12 225.0.0.1 190.0.1.12

At the network security level we applied secure-neighboring protocol.

3.3 Secure neighbor-specific Parameters

Table 2: Secure neighbor-specific parameters

Parameter	*Value*	*Description*
SECURE-NEIGHBOR-TIMEOUT Optional Scope Global, Node	Time Range [1 to 10000 00000 00000 0] Default: 5S	Specifies the time interval for which a node waits to do next neighbor detection handshake. Note: For fast mobile scenarios, reduce the value to get fresher snapshots. For slow mobile scenarios, enlarge the value to reduce overhead.
SECURE-NEIGHBOR-CERTIFIED-HELLO Optional Scope Global, Node	List: • YES • NO Default: NO	Specifies whether or not the network will assume that a pair-wise secret is pre-shared between two nodes. YES: If set to YES, secure neighbor uses the Certificate Variant, which is a two way challenge response scheme which bears sender's certificate in the hello message NO: If set to No, secure neighbor uses the pair-wise shared secret variant of secure neighborhood, which is a three way challenge response scheme

Examples of Parameter Usage

The following configurations enables secure neighbor in node 1:

[1] SECURE-NEIGHBOR-ENABLED YES

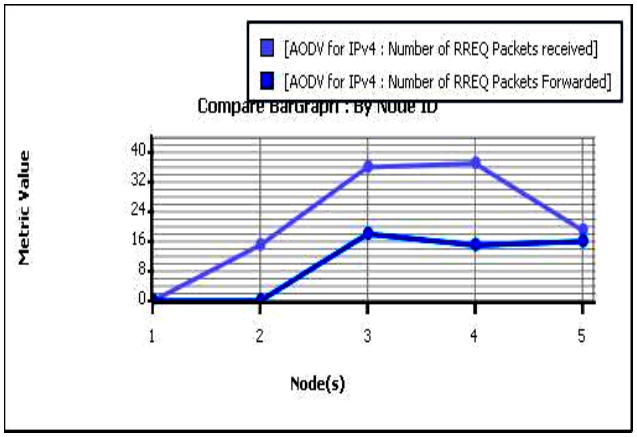
[1] SECURE-NEIGHBOR-TIMEOUT 5S

[1] SECURE-NEIGHBOR-CERTIFIED-HELLO NO

And after that attach static root multicast file send the packet source to destination using constant bit rate Constant bit rate (CBR) and traffic generator

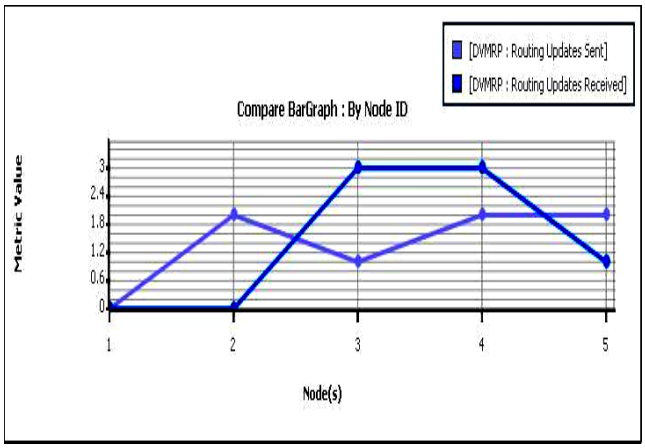
3.4 Simulation Results

The following graph shows the Number of route request packets received and forwarded of AODV.



Graph 1

The following graph shows the routing updates sent and received for all nodes by applying DVMRP.



Graph 2

A. Secure-neighbor Protocol

The following figure shows the total number of Challenge messages sent per a node.

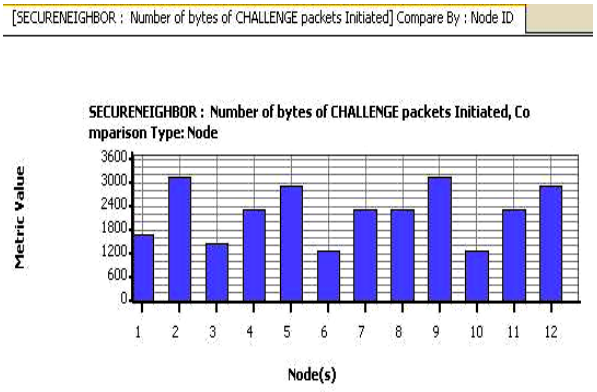


Fig. 6 No of challenge packets initiated

The following figure shows the Total number of Response1 messages sent to all nodes.

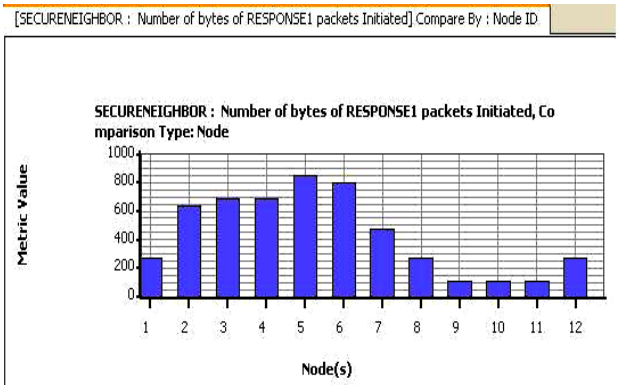


Fig. 7 Number of RESPONSE1 packets Initiated

The following figure shows the Total number of bytes of Response1 messages received.

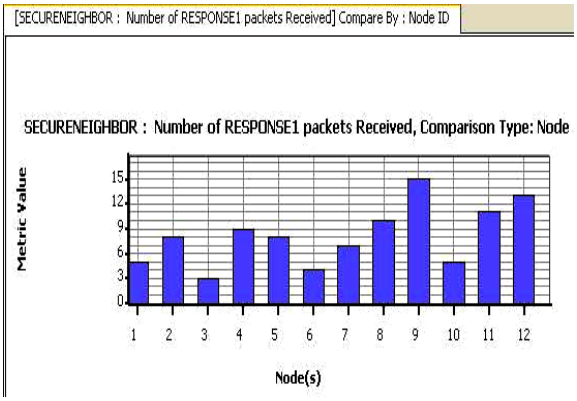


Fig. 8 Number of bytes of RESPONSE1 packets Received

The following figure shows the Total number of Hello messages received for all nodes.

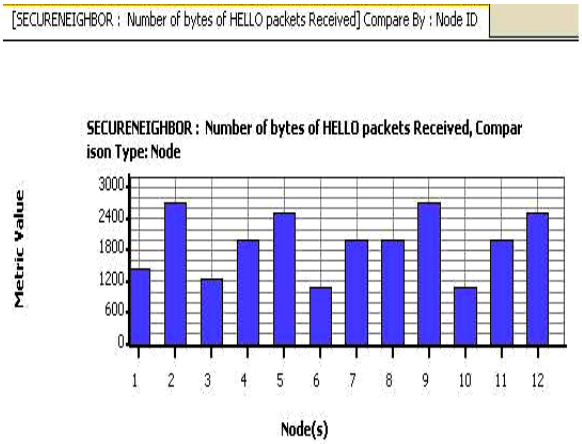


Fig. 9 Number of HELLO packets Received

4. Conclusion

We designed a novel protocol for IP trace back to detect DDoS attack based on Secure-Neighbor. We have taken the parameters of Responce1 and Nonce from Secure-Neighbor protocol and developed a novel protocol to find out the attacker's IP address at the moment the attack is taking place. The novel protocol applies the decryption function on Nonce and value of neighbor-timeout of a particular node to find the metric value at the moment the attack is taking. We formulated the approach mathematically and solved the each step of finding the IP address of an attacker for all possible entities. This scheme requires as single interface environment. The novel protocol for IP trace back will guarantees the finding of attacker's IP address. By using the Secure-Neighbor protocol every node has the information of all other nodes which are connected to that node and every node update the information of all its connecting nodes for every t-second. We developed the novel protocol for IP trace back to detect DDoS attack on a single interface model only. We used the Cisco 7306 routers to find develop our network-model. As the concepts of cryptography applications are very vast, it is possible to extend the protocol to the Response2 messages of Secure-Neighbor protocol. There is a provision to extend the novel protocol for IP trace back to detect DDoS attack on N number of interfaces too.

References

[1] K. J. Houle and G. M. Weaver, "Trends in Denial of Service Attack Technology", Oct. 2001, CERT Coordination Center, pp.1-21. http://www.cert.org/archive/pdf/DoS_trends.pdf.
[2] Robert Beverly and Steven Bauer, "The Spoofer Project: Inferring the Extent of Source Address Filtering on the

Internet", USENIX SRUTI: Steps to Reducing Unwanted Traffic on the Internet Workshop, Jul. 2005, 7(2), pp. 53-59.
[3] David Moore, et al., "Inferring Internet Denial-of-Service Activity", ACM Transactions on Computer System (TOCS), May 2006, 24(2), pp. 115-139.
[4]Vrizlynn L. L. Thing, Morris Sloman, Naranker Dulay "Non-Intrusive IP Traceback for DDoS Attacks",Nov.2007.6(2),pp. 371-373.
[5]SAVAGE, S., WETHERALL, D., KARLIN, A., AND ANDERSON, T. 2000. Practical network support for IP traceback. In Proceedings of ACM SIGCOMM. ACM, New York, 295–306
[6] SONG, D. X., AND PERRIG, A. 2001. Advanced and authenticated marking schemes for IP traceback. In Proceedings of the IEEE INFOCOM. IEEE Computer Society Press, Los Alamitos, Calif.
[7]Belenky, Andrey; Nirwan Ansari (2007). "On deterministic packet marking". Computer Networks: The International Journal of Computer and Telecommunications Networking 51 (10): 2677–2700.
[8]Rayanchu, Shravan K.; Gautam Barua (December 22-24, 2004). "Tracing Attackers with Deterministic Edge Router Marking (DERM)". Distributed Computing and Internet Technology, First International Conference. Bhubaneswar, India. pp. 400–409.
[9]Shokri, Reza; A. Varshovi, H. Mohammadi, N. Yazdani, and B. Sadeghian (September 13-15, 2006). "DDPM: Dynamic Deterministic Packet Marking for IP Traceback". IEEE International Conference on Networks. Singapore. pp. 1–6.
[10] Snoreren, Alex C.; C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, B. Schwartz, S. T. Kent, W. T. Strayer (2002). "Single-packet IP traceback". IEEE/ACM Trans. Netw. 10 (6): 721–734.
[11]Hazeyama, Hiroaki; Y. Kadobayashi, D. Miyamoto and M. Oe (June 26-29, 2006). "An Autonomous Architecture for Inter-Domain Traceback across the Borders of Network Operation". Proceedings of the 11th IEEE Symposium on Computers and Communications. Cagliari, Sardinia, Italy. pp. 378–385.
[12]Burch, Hal; Bill Cheswick (2000). "Tracing Anonymous Packets to Their Approximate Source". LISA. pp. 319–327
[13] Yang Xiang, Wanlei Zhou "An Analytical Model for DDoS Attacks and Defense" Proceedings of the International Multi Conference on Computing in the Global Information Technology, August 2006, page.66.
[14] Micah Adler" Trade-Offs in Probabilistic Packet Marking for IP Traceback" Journal of the ACM, Vol. 52, No. 2, March 2005, pp. 217–24
[15] A. Yaar, A. Perrig, and D. Song, "FIT: Fast Internet Traceback", in Proceedings IEEE INFOCOM, 2005, Vol.2, pp. 1395- 1406 .
[16]S.Karthik, R.M.Bhavadharini, Dr.Y.P.Arunachalam "Analyzing Interaction between Denial of Service (Dos) attacks and threats" Proceedings of the 2008 International Conference on Computing, Communication and Networking (ICCCN 2008),4 (2), pp. 68-75.
[17] V. Paruchuri and A. Durresi, "Study of Probabilistic Marking for IP Traceback under DDoS Attacks," CIS-LSU Technical Report, 2007,
[18] M. Walfish, M. Vutukuru, H. Balakrishnan, D. Karger, and S.Shenker, "DDoS Defense by Offense", in Proceedings of ACM SIGCOMM, September 2006.
[19] W. Feller, An Introduction to Probability Theory and Its Applications, Vol. 2, 1st ed. New York: Wiley, 1966.
[20] Skitter, CAIDA tools, www.caida.org/tools/measurement/skitter/.
[21] "University of Oregon Route Views Project,"http://www.routeviews.org/.
[22] T. Peng, C. Leckie, et. Al., "Adjusted probabilistic packet marking for IP traceback," in Networking, 2002.
[23] M. Waldvogel, "Gossib vs. IP traceback rumors," in Proceedings of 18th ACSAC, 2002.
[24] B. Duwairi, A. Chakrabarti, and G. Manimaran, "An Efficient Packet Marking Scheme for IP Traceback", in Networking 2004.
[25] M. Muthuprasanna and G. Manimaran, "Space-Time encoding scheme for DDoS attack traceback," in IEEE Globecom, Nov. 2005.
[26] D. Basheer and G. Manimaran, A novel packet marking scheme for IP traceback," in Proc. 10th IEEE ICPDS, July 2004.
[27] Q Dong, S Banerjee, M Adler, K Hirata,, "Efficient probabilistic packet marking", 13th IEEE ICNP, Nov 2005.
[28] http://www.securityfocus.com/news/9411
[29]http://www.networkworld.com/news/2005/051605-DDoSextortion. html
[30] CERT. Incident Note IN-2004-01 W32/Novarg.A Virus, 2004.

Yogesh Kumar Meena received the Integrated Masters (BTech and MTech) in ABV-Information Technology from Indian Institute of Information Technology and Management (ABV-IIITM) Gwalior, India, in 2010. In June 2010, he joined the Information Technology Department at Sharda Group of Institution, Agra, India as an Assistant Professor.

He is a member of the IEEE, IETE, AICSIT and MIR lab. Meena is a reviewer of IEEE and Springer journals. He was given the Excellent Award in Faculty Development Program, organized by Sharda Group of Institutions, Agra, India.

Prof. Aditya Trivedi is a Professor in the Information and Communication Technology (ICT) Department at ABV Indian Institute of Information Technology and Management, Gwalior, India. He has about 20 years of teaching experience. He has published around 60 papers in various national and international journals/conferences. He is a fellow of the Institution of Electronics and Telecommunication Engineers (IETE).

In 2007, he was given the IETEs K.S. Krishnan Memorial Award for best system oriented paper.