Honeypot Scheme for Distributed Denial-of-Service Attack

Vinu V Das
Assistant Professor, Department of Computer Science and Engineering,
SAINTGITS College of Engineering, Kottayam – 686532, Kerala, India.
Email: prof.vinuvdas@gmail.com

Abstract *– Honeypots are physical or virtual machines successfully used as intrusion detection tools to detect worm-infected hosts. Denial of service (DoS) attack consumes the resources of a remote client or network itself, there by denying or degrading the service to the legitimate users. In a DoS defense mechanism, a honeypot acts as a detective server among the pool of servers in a specific network; where any packet received by the honeypot is most likely a packet from an attacker. This paper points out a number of drawbacks such as Legitimate Attacker and Link Unreachable problem in the existing honeypot schemes. This paper proposes a new efficient honeypot model to solve all the existing problems by opening a virtual communication port for any specific communication between an authorized client and server; and by providing facility to act as an Active Server (AS) for any honeypot.*

Index Terms: Honeypot, Denial-of-Service, Network Security, Spoofing

I. Introduction

Even though the last decade has witnessed tremendous growth in the internet service and its use, a proper mechanism has not evolved to discourage or stop the internet attackers. One such internet attack is Distributed Denial of Service (DDoS) attack, which continues to pose a real threat to internet service [1, 2]. Even though many schemes have been proposed to defend against spoofing, DDoS attack [12, 7, 11, 6], none has overcome the difficulties of widespread deployment. Whereas trace back scheme [7, 11, 6, 17] can identify the real source of spoofed attack packets, to take appropriate action against the source: at least to stop them for instances. The pushback mechanism is effective to some extent by enforcing aggregate based congestion control in the containment of DDoS attack traffic. But it damages the traffic [12] in a highly dispersed DDoS attacks.

Honeypots are physical or virtual machines used to defend information from warm host [13, 14]. Past years have seen several honeypot mechanisms including a number of roaming schemes. Roaming honeypot schemes are generally used as defense mechanisms against non-spoofed service-level DoS attacks [21]. For a period of time one or more servers may act as honeypot from a pool of servers, without consuming service interruption. In other words, one or more legitimate services in the pool, in coordination with legitimate clients and remaining peer replicas, assumes the role of a honeypot for specific intervals of time called honeypot epoch. Such kind of roaming honeypot services makes its difficult for attackers to identify active servers, thereby causing them to be trapped in.

The focus of this paper is to freeze private services from unauthorized sources against address spoofing DDoS attacks. This is achieved by controlling attack traffic to its source using the pushback mechanism, for tracing back to a particular source, and by the ability to defend the attackers using roaming honeypots. The existing system has number flaws namely the honeypot schemes will not work if one of the clients in the AS is an attacker and the other one is a legitimate client; and when there is a physical breakdown in active path. They have dealt by opening a virtual or physical communication port to any client only after its authentication and for other nodes AS still acts as a virtual/physical honeypot. And by opening a temporary communication channel through the honeypot, by virtually making it to act as AS.

II. Related Works

Ingress Filtering [9] and IPsec can prevent most spoofing DDoS attacks, if they are properly deployed. However, the management hassle and per-packet performance overhead are obstacles. Overhead may occur on mobile IP support, when an ingress filtering rules are adapted. In honeypot back propagation, filtering stats,

978-0-7695-3516-6/08 $25.00 © 2008 IEEE
DOI 10.1109/ICACC.2009.146

only when a packet from an attacker is detected, based on destination address. Only when attack occurs, the honeypot back propagation scheme required light-weight per-packet filtering.

The SOS architecture [19] tackles the DoS attack in the context of a privet service with predetermined clients. It uses on overly network to hide the locations of a small number of proxy nodes (servelets) and allows only traffic from the servlets to enter the protected network. If a client requires an access to the overlay network it has to authenticate with any one of the access point (SOAPS), which routes each clients packet to one of the servelets using hash-based routing. Due to overhead of the overly routing, communication latency may increase upto 10 times. The work on this paper provides a more effective solution by avoiding overly routing and by taking action only when an attack occurs.

Packets marking [10, 7, 11] and packets logging [20] are the two different approaches to the trace back problem. Even though, hierarchical trace back [5] with inter and intra-domain trace back mechanism is similar to the proposed hieratical honeypot back propagation approach, it triggers only upon the detection of packets from attackers.

The proactive server roaming scheme has been proposed in [21], where a prototype of the scheme is evaluated, and has been studied through simulation in [22]. The location of the honeypot is roaming within a pool of server, so as to make it difficult for attackers to direct their traffic away from the honeypots and to avoid detection [3]. The scheme allows only k out of N series to be concurrently active where as the remaining N-K act as honeypot. The location at the current active server and the honeypot are changed according to a pseudo-random schedule shared among the servers and legitimate client. There fore legitimate clients always send their service request to active servers whereas attack request may reach the honeypot. The source address of any request that hits a honeypot is black listed so that all future requests from this source are subsequently dropped. The source address is not blacklisted unless a full service handshake is recorded to ensure that it is not spoofed.

III. Problem Identification

All the present honeypot systems [21, 22] have a few drawbacks such as; if one of the clients in the AS is an attacker and another one is legitimate client; and when there is a physical breakdown in active path. When one of the clients of an active server is legitimate user and trying to reach the server destination, present models have no means to block unauthorized nodes of the same ISP (which may be the present AS) to reach the same server destination. Thus the probability to take same path to reach the server by an unauthorized attacker is high. This paper refers this problem as *Legitimate Attacker problem*. Figure 1 describes a typical case, where an attacker under AS is attacking the server via the same path as that of legitimate client. The AS has opened its communication port for the legitimate clients as it is not a honeypot.

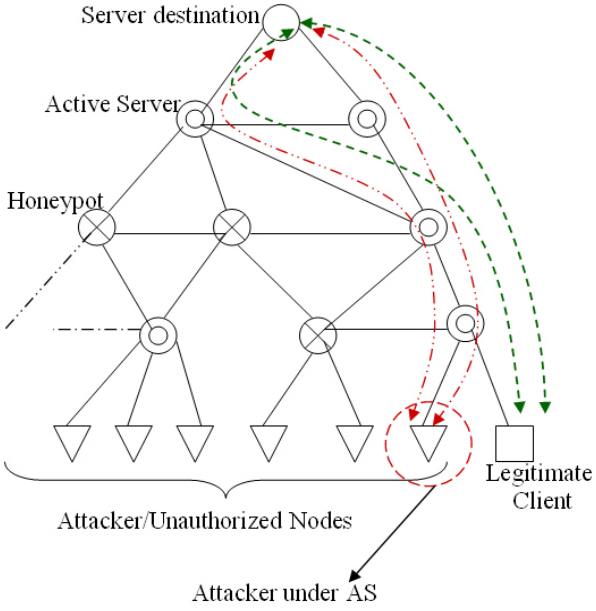

Figure 1 – Legitimate Attacker

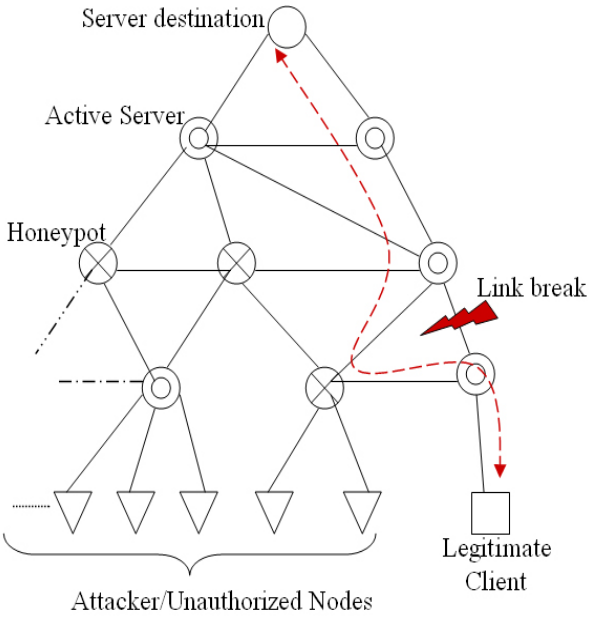

Figure 2 - Link Unreachable

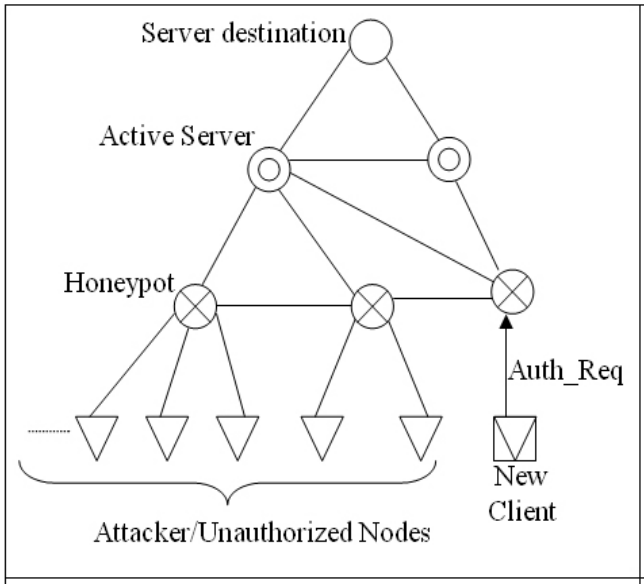

Fig 3 (a) - Authorization Request

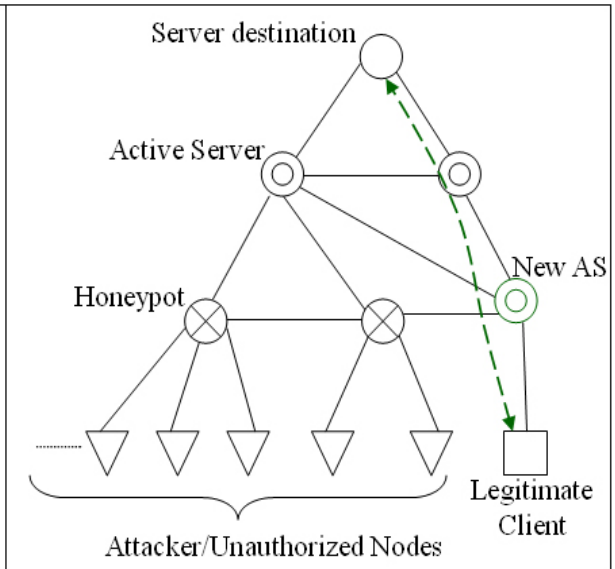

Fig 3 (b) – Rearranging and Opening the Communication Path

Present honeypot schemes will have to pay more, Denial of Service (DoS), if the one and only link between the legitimate client and server destination breaks down due to any physical damage. Even though a number of other links is still active, the whole communication of clients and server will be disturbed, because all the other reachable paths have honeypots in between. This paper refers this DoS problem as *Link Unreachable problem*. Figure 2 depicts a typical case, where a legitimate client is isolated because its one and only reachable link has broken down and nearby network routs are blocked by honeypot servers.

Apart from giving conventional security to the proposed honeypot scheme, both the above mentioned problems, *Legitimate Attacker and Link Unreachable,* are solved.

IV. Proposed Honeypot Scheme

At any time one client may contact an AS (which maybe even now acting as a honeypot) for authorization. Only authorization request packets are accepted by any AS in the network that is acting as a honeypot. This authorization request is configured along with communication protocol internally, so that its packets will reach only to a particular port of the ISP/AS. And this port does not accept/process any other messages/ requests by ignoring them. Figure 3 (a) shows the authorization request process, where a new client is contacting a honeypot to authenticate and open a new communication channel.

Auth_Req(Cli_ID, Dest_ID, Cli_IP,
Pri_Key_Enc(Req))

Once authenticated a (virtual or physical) port is opened for the particular communication and a rearrangement will be done (by authorized regional parent servers depends upon the network traffic and geographical location) within the network so that at least one shortest communication path is opened from requested Client ID to Destination Server ID. If the client contacts honeypot for authentication, then it will change to AS by shifting honeypot session temporarily into nearby idle server. Figure 3 (b) shows the rearrangement of the honeypot to AS and opening of the communication path to server destination. Moreover the communication port number will be frequently changed by all ASs with the consultation of authorized clients. So any packet reached to a port should be of by any specific client, others are from unauthorized attacker.

If one or more unauthorized clients is sending wrong Auth_Req, then it wont be processed, and if any unauthorized attacker sends any message/request to a communication port that is dedicated or not dedicated to a client it will cause an interrupt, so that, that packet will not be processed since those are not from authorized client. Moreover the probability to spoof correct client details to send it to a correct port number is negligibly small, which can be further blocked by introducing appropriate encryption algorithm. Thus the *Legitimate Attacker problem* is solved by opening a communication port to any client only after its authentication. For other nodes AS still acts as a virtual honeypot.

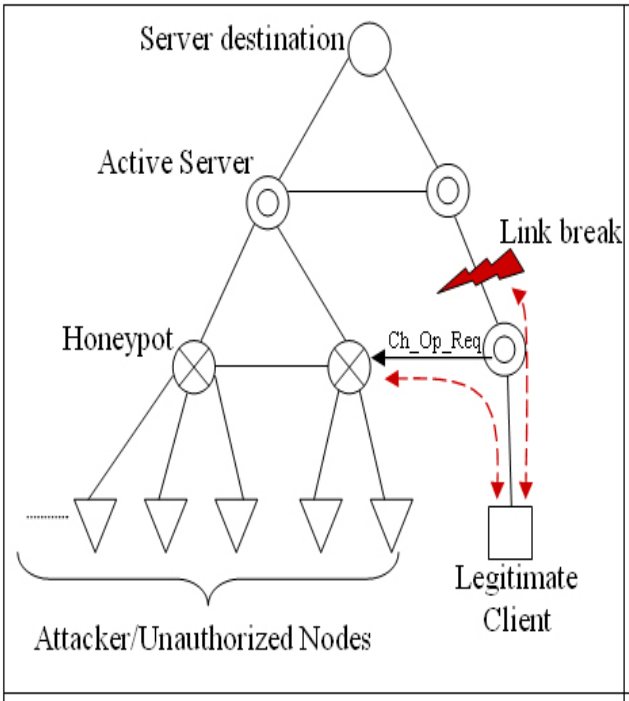

Fig 4 (a) – Channel Open Request

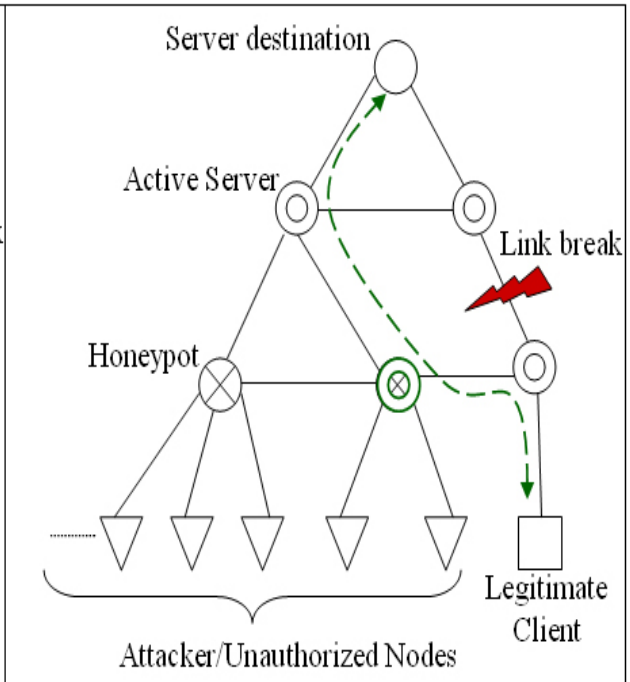

Fig 4 (b) – Temporally Opening the Communication Path through Honeypot

The honeypot generally does not accept any message/request from any nodes, but in the proposed model it accepts only channel open request (Cha_Op_Req) from nearby ASs/honeypots and no other messages/requests accepted from any client nodes. If any link breakdowns occur in the existing active communication path, then the corresponding AS will look for an alternate path via any nearby AS. If no AS could be located then it will send a Cha_Op_Req to nearby honeypot to temporarily open a channel for packets from the given client to destination server via the AS. At this point, the specific honeypot acts as virtual AS, only to the requested nearby AS, and it still remains as honeypot for other ASs and client nodes.

Cha_Op_Req(Cli_ID, Dest_ID, AS_ID, Cli_IP, Pri_Key_Enc(Req))

Figure 4 (a) shows how a channel open request is forwarded to a honeypot and Figure 4 (b) illustrates the opening of the temporary communication path through the nearby honeypot. The identity of honeypot still remains as it is for other unauthorized nodes and nearby ASs/honeypots. The *Link Unreachable problem* has been solved by opening a communication channel through the honeypot, by virtually making it to act as AS. At this point, other attackers and unauthorized nodes will not be able to intrude into the network and no DoS because it still acts as honeypot to other nodes and ASs.

V. Conclusions

This paper has analyzed existing honeypot system to identify some of problems, such as *Legitimate Attacker and Link Unreachable problem*. The *Legitimate Attacker problem* is solved by opening a virtual or physical communication port to any client only after its authentication and for other nodes AS still acts as a virtual/physical honeypot. *Link Unreachable problem* has been dealt with by opening a temporary communication channel through the honeypot, by virtually making it to act as AS; so other attackers and unauthorized nodes will not be able to intrude into the network and no DoS because it still acts as honeypot to other nodes and ASs. Apart from perfectly addressing the conventional issues, proposed system is efficient and secure in DoS.

VI. Acknowledgement

I praise, and worship by giving thanks to my Lord and Saviour Jesus Christ, without Whom nothing would have been possible for me.

VII. References

[1] CAIDA, "Nameserver DoS Attack," http://www.caida.org/projects/dns-analysis/ oct02dos.xml, 2002.

[2] CCCA CA-2003-04, "MS-SQL Server Worm", http://www.cert.org/advisories/CA-2003-04.html, 2003.

[3] S. M. Khattab, C. Sangpachatanaruk, D. Moss´e, R. Melhem, and T. Znati, "Roaming Honeypots for Mitigating Service-level Denial-of-Service Attacks," In *ICDCS, 2004*.

[4] J. Mirkovic and P. Reiher, "A Taxonomy of DDoS Attack and DDoS Defense Mechanisms," *ACM SIGCOMM Computer Communication Review*, Vol. 34, No. 2 pp. 39–53, April 2004.

[5] M. Oe, Y. Kadobayashi, and S. Yamaguchi, "An implementation of a hierarchical IP traceback architecture," In *SAINT 2003 Workshop on IPv6 and applications*, Jan 2003.

[6] K. Park and H. Lee, "On the effectiveness of probabilistic packet marking for IP traceback under denial of service attack," In *IEEE INFOCOM*, pp. 338–347, 2001.

[7] S. M. Bellovin, M. Leech, and T. Taylor, "ICMP Traceback Messages, In *draft-ietf-itrace-01.txt, internet-draft,"* October 2001.

[8] A. Perrig, D. Song, and A. Yaar, "StackPi: A New Defense Mechanism against IP Spoofing and DDoS Attacks," *Technical Report CMU-CS-02-208*, December 2002.

[9] P. Ferguson and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing," In *RFC 2827*, May 2001.

[10] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Practical network support for IP traceback," In *ACM SIGCOMM, 2000*.

[11] D. X. Song and A. Perrig, "Advanced and Authenticated Marking Schemes for IP Traceback," In *IEEE INFOCOM*, 2001.

[12] R. Mahajan, S. M. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker,. "Controlling high bandwidth aggregates in the network," In *ACM SIGCOMM Computer Communication Review*, Vol. 32, pp. 62–73. ACM Press, 2002.

[13] T. H. Project, "*Know Your Enemy*," Addison-Wisley, Indianapolis, IN, 2002.

[14] S. Staniford, V. Paxson, and N. Weaver, "How to 0wn the Internet in your spare time," In *Proc. of the 11th USENIX Security Symposium*, 2002.

[15] X. Wang and M. Reiter, "Mitigating Bandwidth-Exhaustion Attacks using Congestion Puzzles," In *ACM CCS 2004*.

[16] D. K. Y. Yau, J. C. S. Lui, and F. Liang, "Defending Against Distributed Denial-of-service Attacks with Max-min Fair Server-centric Router Throttles," In *IWQoS*, 2002.

[17] D. Dean, M. Franklin, and A. Stubblefield, "An algebraic approach to IP traceback," *ACM Trans. Inf. Syst. Secur.*,5(2):119–137, 2002.

[18] C. Perkins, "IP Mobility Support," In *RFC 2002*, October 1996.

[19] A. Keromytis, V. Misra, and D. Rubenstein, "SOS: Secure Overlay Services," In *ACM SIGCOMM*, 2002.

[20] A. C. Snoeren, "Hash-based IP traceback," In *ACM SIGCOMM*, pp. 3–14, 2001.

[21] S. M. Khattab, C. Sangpachatanaruk, R. Melhem, D. Moss´e, and T. Znati, "Proactive Server Roaming for Mitigating Denial of Service Attacks," In *ITRE, 2003*

[22] C. Sangpachatanaruk, S. M. Khattab, T. Znati, R. Melhem, and D. Moss´e, "A Simulation Study of the Proactive Server Roaming for Mitigating Denial of Service Attacks," In *ANSS, 2003*.