Implementation of a Secured system with Roaming Server and Roaming Ports

R. Bharathi

AP/Dept. of ECE, Anna University of Technology –Nagercoil Campus
Kanyakumari District. Tamilnadu,India

Prof. Dr. R. Sukanesh

Professor/Dept.of ECE, Thiagarajar College of Engineering, Madurai,
Tamilnadu,India

Abstract— The main goal of this paper is to design and implement a secured system against Server hijacking, which leads to Denial of Service (DoS) [5] attacks. This system uses more than one server for providing security. But only one server will be active at a time. The inactive servers act as Roaming Honeypots[9]. The source address of any request that hits a honeypot is recorded and all its future requests are dropped. Thus this system acts as an Intrusion Detection System (IDS). It is impossible to identify the active servers and the honeypots at a given moment even if attackers obtain the identities of all servers. Moreover the UDP/TCP port number used by the server varies as a function of time and a shared secret between the server and the client. This mechanism simplifies both the detection and filtering of malicious packets and it does not require any change to existing protocols. This port hopping[10] or roaming port technique is compatible with the UDP and TCP protocols. This system can be implemented in real time successfully.

Keywords-: Denial of Service, Server roaming, Honeypots

Introduction

Internet has become an integral part in many people's lives . Hence the need to keep servers protected, online, and available has become increasingly important. In recent years, denial-of-service (DoS) attacks and distributed DoS (DDoS)[4] attacks have become more sophisticated and effective at obstructing this availability. In 2000, several online companies such as eBay, Amazon.com, CNN.com, and Yahoo[5,8] were all affected by a large scale DDoS attack. During this attack, their websites were rendered virtually unreachable to many Internet users, resulting in severe financial losses, in addition to the many unsatisfied customers.

A comprehensive counter-measure for DoS attacks has four distinct elements: prevention, detection, mitigation, and traceback, shown together in Figure 1. Before an attack occurs, there should be existing prevention mechanisms that are capable of eliminating the threat of the attack. When the attack does occur, only a successful and timely detection of the attack will allow the appropriate mitigation mechanism to be deployed. During or following the attack, a method called traceback [3] can be used to determine the source of the attack. In addition, traceback can also help improve future methods for detecting and preventing a DoS attack. The dependence upon all four of these items is crucial for a successful DoS countermeasure.

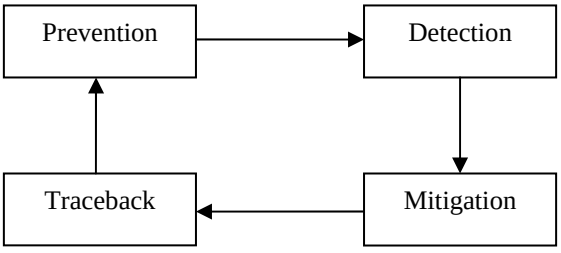

Figure 1: Components of a DoS countermeasure

I. RELATED WORK

Many techniques have been already proposed by researchers to counter Denial of Service attack. [3] employs an overlay network similar to SOS, which uses spread-spectrum like path diversity to counter DoS attacks. The system also uses secret keys to authenticate valid messages. Like SOS, it requires additional nodes to construct the overlay network, and the additional overhead has an impact on message throughput and latency. There are other several proposed methods to filter offending DoS traffic. Some proposals, for example, that by Krishnamurthy et al. [17], filter according to the source IP address. This is convenient and efficient, allowing implementation in existing packet-filtering routers. However, IP addresses are subject to spoofing; furthermore, using a white list of source addresses of legitimate clients peers is difficult, since many hosts may have dynamic IP addresses due to the use of a network address translator (NAT), Dynamic Host Configuration Protocol (DHCP), and mobile-IP. Some proposals try to detect spoofed senders using new routing mechanisms such as "path markers" supported by some or all of the routers en route, as in Pi, Stateless Internet Flow Filter (SIFF) , Active Internet Traffic Filtering (AITF) , and Pushback but global router modification is difficult to achieve. Few proposals try to detect spoofed senders using only existing mechanisms, such as the hop count (Time-to-Live (TTL)), as in Hop Count Filtering (HCF)]. However, empirical evaluation of these approaches show rather disappointing results [5].A different approach is to perform application-specific filtering for predefined protocols. Such protection schemes are cumbersome, only work for a handful of well known protocols, and are usually restricted to attackers that transmit invalid protocol packets. IPsec performs filtering at the IP layer, by authenticating messages using MACs, based on shared secret keys. IPsec ensures that higher level protocols only receive valid messages. However, the work required to authenticate each message is invested for each incoming packet that has a valid security parameter index (SPI). Once the SPI, which is sent in the clear, is known, an attacker can perform a DoS attack by overloading IPsec with many bogus packets to authenticate. In contrast, our solution ensures that the authentication phase is reached only for packets that are valid with high probability by constantly changing the cleartext filtering identifier, for example, the SPI. [20] deals with a gossip based multicast protocol resistant to DoS attacks. Drum does not use pseudorandom port hopping, and it heavily relies on well-known ports that can be easily attacked. Therefore, Drum is far less resistant to DoS attacks than the protocol we present here. Finally, Drum focuses on multicast only, and as a gossip-based protocol, it relies on a high level of redundancy, whereas the protocol presented herein sends very little redundant information. Independent of our work, Lee and Thing examined the use of port hopping to mitigate the effect of DoS attacks

I. ROAMING HONEYPOTS SCHEME

Honeypots [9] act as a proactive detection mechanism. These are machines that are not supposed to receive any legitimate traffic. Any traffic destined to a honeypot is most probably an ongoing attack and can be analyzed to reveal vulnerabilities targeted by attackers. Coupled with an Intrusion Detection System (IDS), honeypots are effective in detecting hosts exploited by Internet worms that perform random scanning. Roaming honeypots can be achieved by configuring a (continuously and unpredictably changing) subset of k out of N servers as active and provide service, while the rest acting as honeypots. It is impossible to identify the active servers and the honeypots at a given moment even if attackers obtain the identities of all servers. Roaming Honeypots [2] is a framework designed to mitigate the effects of DoS attacks. The active server changes its location among a pool of servers to defend against unpredictable and likely undetectable attacks. Only legitimate clients will be able to follow the server as it roams. To be able to know the active server location, a client needs to have at least the following information: the server address and the time that the server will be active. This information can be simply obtained by using a series of communication. To avoid the DoS attacks on the Internet, however, clients and servers need a secure communication that provides privacy and integrity to protect the information[6].A proposed secure, proactive and time-triggered roaming scheme defines an upper bound on the time interval between consecutive server roaming instances. This upper bound is adaptively changed to reflect the current threat level. For instance, in a high threat period this upper bound is set to be small, while it is set to a larger value in normal conditions.

Proactive server scheme [11] involves an initialization phase. When a legitimate client subscribes to the service, it receives some information that allows it to create a secure channel with the service. Before it starts using the service, a client waits until it receives a message from the service carrying the necessary information for calculating roaming times and roaming target addresses. This information includes a sequence of keys such that each key is used to generate the roaming time and the address of the next roaming target for the roaming instance.

II . ROAMING PORTS SCHEME

In the TCP/IP [7] protocol suite, UDP (User Datagram Protocol) and TCP (Transmission Control Protocol) protocols are used for connectionless and connection-oriented data transfer. Sockets are used to establish a communication channel for exchanging data using the UDP or TCP protocols. The end points of the communications channels are defined by the end hosts' IP addresses and their UDP and TCP port numbers, i.e**. <source and destination IP address, source and destination port number>.** A communication session using UDP or TCP, use fixed port numbers from 0 to 65535. Well known ports (0 through 1023) are used by servers to provide standard services. For example, web services running the http protocol typically listen on port 80. This paradigm has some major problems. Firstly, the well known port design is vulnerable to port scanning by attackers.

In roaming ports technique, the server's Port numbers change dynamically as a function of time and a cryptographic key shared between the server and the client. Authorized clients who have the key will be able to determine the current port number used by the server, whereas the malicious users will not know the current valid port number. The server can then easily filter off illegitimate packets by inspecting the port number contained in the UDP/TCP headers.

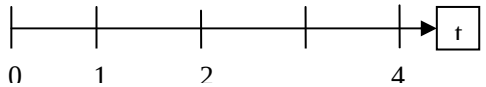

Figure 2: Time Slot Structure

Figure 2 shows the time slot structure with 4 time slots. In each time slot different port numbers are used for communication between the client and the server i.e. if during the period from t1 to t2 port number 5000 is used for communication then during the period from t2 to t3 port number 5001 may be used for communication and so on.

When the server receives packets that carry "invalid" port numbers, they can be easily detected and filtered off. There is no need for the server to examine the contents of the packets in order to determine if a packet is malicious. As a result, the computational resources needed to detect and filter off the malicious packets is reduced significantly.

III. PROPOSED SCHEME

In this scheme the server initially informs the authenticator that it has joined the server pool. After receiving this information the authenticator [7] checks whether the server is a valid server and then it sends the list of registered servers, registered clients, epoch list (roaming interval), File list, running server list to the server.

When a client logs into the network the authenticator checks whether the client is a legitimate one and then it gives the following information to the legitimate clients i)list of registered server,ii)roaming interval secret keys iii)list of permitted files for downloading iv)Port Number currently in use. Figure 3 shows the various functional units of the server. The authenticator listener accepts message only from Authenticator. Messages from others are considered as illegitimate. The control line listener helps to activate server or remove server from server pool. It issues a control signal at the end of the epoch time. Thus it helps in roaming between servers. Epoch daemon sends the current active server to all servers in the server pool. It sends this detail within the particular epoch time interval. Timer line listener helps to synchronize the clock among the servers. The data line listener receives and process the information send by the active server from the Epoch daemon. The client Request Listener listens to the requests from legitimate clients.

The client sends message to the client request listener for downloading the files. The Epoch daemon of the active server informs the end of the epoch interval to the control line listener and the data line listeners of other servers. Then another server becomes active for the next time interval. The timer line listener performs the function of time synchronization.The server has to inform the Authenticator listener of authenticator if it wants to join or leave the server pool. The Authenticator thread of Authenticator sends information to authenticator listener of the server if the server rejoins the pool. Client threads are created only when the client downloads the file from the server .The threads get vanished after the downloading process is over.

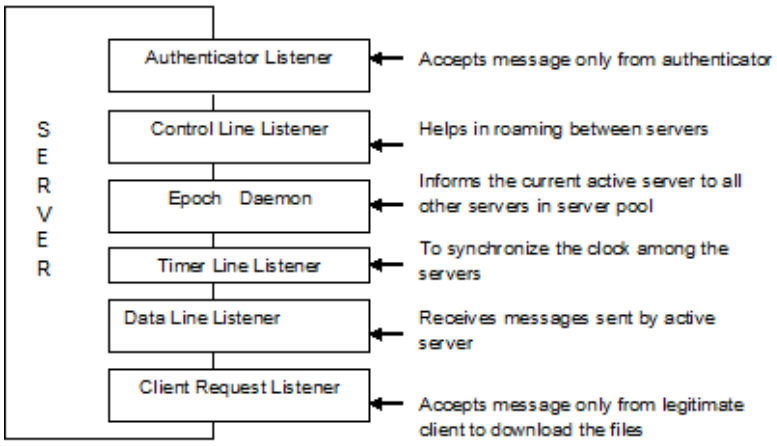

Figure 3: Functional units of server

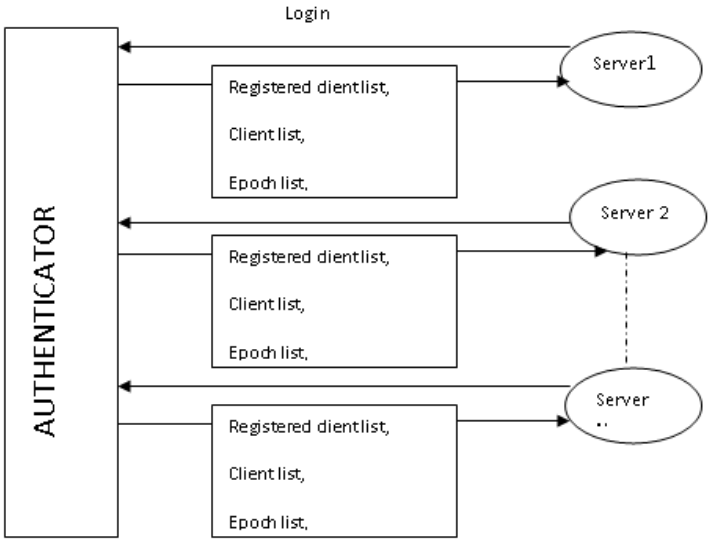

Figure 4: Information flow on Server Login

Java is used for implementing this project. A socket allows a single computer to serve many different clients at once, as well as serving many different types of information. This feat is managed by the introduction of a port, which is a numbered socket on a particular machine. A server process is said to "listen" to a port until a client connects to it. A server is allowed to accept multiple clients connected to the same port number, although each session is unique. To manage multiple client connections, a server process is multithreaded. TCP/IP reserves the lower 1024 ports for specific protocols. Port number 21 is for FTP, 23 is for Telnet, 25 is for E-mail 80 is for HTTP - and the list goes on.

There are two kinds of TCP sockets in java. One is for servers and the other is for clients. The Server Socket class is designed to be a "listener", which waits for clients to connect before doing anything. The socket class is designed to connect to server sockets and initiate protocol exchanges. The creation of a socket object implicitly establishes a connection between the client and server. The constructor used to create client sockets is **Socket (InetAddress ipAddress,int port)** - Creates a socket using a preexisting InetAddress object and a port; can throw IOException.

Once the socket object has been created, it can also be examined to gain access to the input and output streams associated with it. Each of these methods can throw an IO Exception if the sockets have been invalidated by a loss of connection on the Net. **InputStream getInputStream()** – Returns the Input stream associated with the invoking socket. **OutputStream getOutputStream()** – Returns the output stream associated with the invoking socket.

IV. ANALYSIS

The roaming interval or epoch time is the main factor which determines the efficiency of this scheme. As the roaming interval decreases, the average response time and the number of migrations[1,2] for a transfer increase. When the attack load increases, the packet drop increases in the case of no roaming. With roaming the packet drop decreases and this shows the effectiveness of this approach. When the total load increases, the competition among the connections for the limited bandwidth of the bottleneck link get more intense, leading to the longer average response time. It is seen through simulation that more number of packets are dropped in the case of a TCP attack than in an UDP attack. The following are the real time snap shots taken during the running of our system.

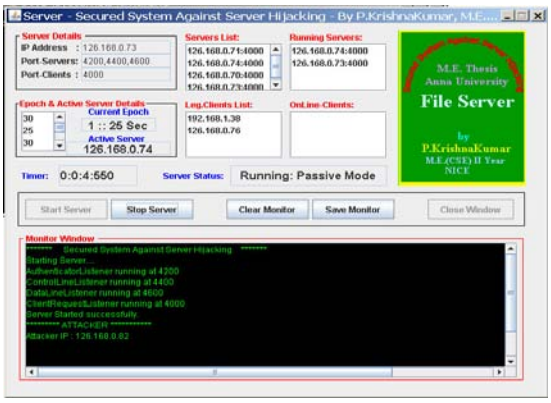

Figure 5: Active server after attack

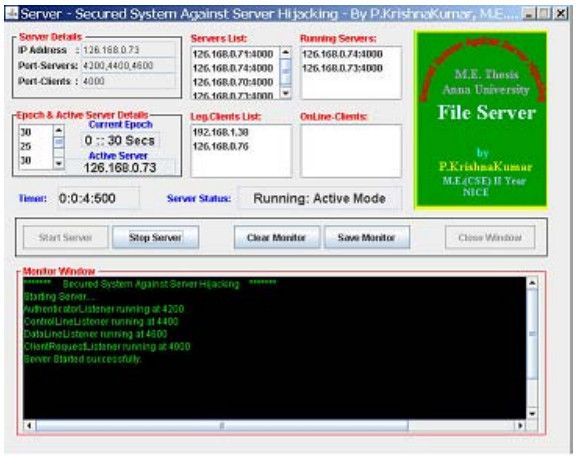

Figure 6: Active server before attack

V. CONCLUSION AND FUTURE WORK

Denial-of-service requires a distributed solution. This technique is now implemented for file server with variable roaming intervals (epoch). This provides an advantage of scaling. Moreover loose clock synchronization technique is used in this project. The results of the implementation are promising. The benefit of this technique outweighs the cost of roaming and the loss caused by the attacks. This paper has presented a crucial building block for denial of service solution by using the combined techniques of Roaming Honeypots and Roaming ports that provides a selective and dynamic response. Defeating DoS attacks involves studying various vulnerabilities in computer networks and designing a number of mechanisms that can prevent all of these problems. The combination of many mitigation methods will prove to be the most effective means to combat the serious threat of DoS attacks and to allow the Internet to become more reliable, useful, and secure. The next step is to develop better and more practical attack and roaming models. Some heuristic techniques are needed. The attack patterns in the real world can be used to draw experiment designs to improve the models. Additionally other well-known network applications, such as a web server and domain name servers, can be deployed as attacked services to

explore different behaviors of the attackers and the roaming strategies can be analyzed to fight against these attacks

ACKNOWLEDGMENT

REFERENCES

[1] Ruiliang Chen, Jung-Min Park and Randolph Marchany,"A Divide-and-Conquer Strategy for Thwarting Distributed Denial-of-Service Attacks", IEEE Transactions on parallel and distributed systems, vol. 18, no. 5, May 2007

[2] F. Sultan, K. Srinivasan, D. Iyer, and L. Iftode, "Migratory TCP: Connection migration for service continuity in the Internet," in Proceedings Of The 22nd International Conference on Distributed Computing Systems (ICDCS), 2002.

[3] M. Sung and J. Xu, "IP Traceback-Based Intelligent Packet Filtering: A Novel Technique for Defending against Internet DDoS Attacks," IEEE Trans. Parallel and Distributed Systems, vol. 14, no. 9, pp. 861-872, Sept. 2003.

[4] R. Chen and J.-M. Park, "Attack Diagnosis: Throttling Distributed Denial-of-Service Attacks Close to the Attack Sources," Proc. IEEE Int'l Conf. Computer Comm. and Networks (ICCCN), pp. 275-280, Oct.2005.

[5] J. Green, D. Marchette, S. Northcutt, and B. Ralph, "Analysis techniques for detecting coordinated attacks and probes," in Proceedings of USENIXWorkshop on Intrusion Detection and Network Monitoring, April 1999.

[6] Paxson, "An analysis of using reflectors for distributed denial of service attacks," ACM Computer Communications Review (CCR),vol. 31, no. 3, July 2001

[7] D. X. Song and A. Perrig, "Advanced and authenticated marking schemes for IP traceback," in Proceedings of IEEE Infocomm, 2001.

[8] J. Jones, "Distributed denial of service attacks: Defenses, a special publication," Global Integrity, Tech. Rep., 2000.

[9] HoneyNet Project. http://project.honeynet.org/

[10] Henry C.J. Lee (2004) 'Port Hopping for Resilient Networks' IEEE conf. on Mobile and Wireless Communications www.doc.ic.ac.uk

[11] Sangpachatanaruk C., Khattab S.M., Znati T., Melhem R. and Mosse D. (2003) 'A Simulation Study of the Proactive Server Roaming for Mitigating Denial of Service Attacks', in Proc. of ANSS.IEEE conf. on Mobile and Wireless Communications www.doc.ic.ac.uk

.Bharathi received B.E degree in the year 1998 and M.E degree in the year 2006.Currently she is pursuing Ph.D Degree in Anna University, Trichy, India. She is presently a Assistant Professor with Anna University College of Engineering ,Nagercoil, India. Her current research interests include network and security, secure communication, and secure e-commerce.

Dr. R. Sukanesh, senior professor in Biomedical Engineering received her B.E. Degree (ECE) from Government College of Technology, Coimbatore in 1982. She obtained her M.E. (Communication Systems) degree from P.S.G. Technology, Coimbatore in 1985 and Ph.D. in Bio Medical Engineering from Madurai Kamaraj University, Madurai in 1999. Since 1985 she is a faculty in the Department of ECE at Thiagarajar College of Engineering, Madurai and presently she is professor of ECE and Head of the Medical Electronics Division in the same college. Her main research areas include Biomedical Instrumentation, Neural Networks, Bio-Signal processing and Mobile Communication. She isguiding twelve Ph.D. thesis in the mentioned areas. She has published 30 papers in referred journals and around eighty papers in International and National conferences conducted both in India and aboard. She has delivered a number of invited lectures in various universities. She has a Diploma in Higher Learning and has co-authored a book on Gandhian thoughts. She is a reviewer for International Journal of Biomedical Sciences and International journal of signal processing. She is an editorial member for journal of Engineering students. She contributed a chapter titled, "Impact of Information Technology in Business" in the book, "Future Organization strategies and Business" edited by Professor Biswajeet Pattanayak. She is the recipient of the outstanding paper award at the 12th International conference on Biomedical Engineering at untec city, Singapore in the year 2005. She also received The President of India's Prize (English) 2006, The Jawaharlal Nehru Memorial prize-2006 and Woman engineer award from IE (India). She is a fellow of Institution of Engineers (India) and a life member of biomedical society of India, Indian Association of Biomedical Scientist and ISTE.

.