**American Journal of Engineering Research (AJER)**
e-ISSN : 2320-0847 p-ISSN : 2320-0936
Volume-02, Issue-05, pp-194-199
www.ajer.us

Research Paper Open Access

# Preventing ADDOS Attack by Using Secure TRNG Based Port Hopping

T. Siva[1,] E. S. Phalguna Krishna[2,] K. Pavan Kumar[3]
1,2,3 Department of CSE, Sree Vidyanikethan Engineering college, Tirupathi, A.P, INDIA.

***Abstract:*** Now a day`s each and every where we are using client-server communication for different information service systems. Normally client server communication can be differentiating by using IP Address and Protocol Port number from one machine to another machine. In network environment we are already having DOS/DDOS Attacks Another Subset of this attack scenario is DOS/DDOS attack is Application Denial of Service(ADOS)attack ,In this the adversary attacks open Ports/Ideal ports present at server side for this the adversary Know need huge machines ,zombie systems and no need sending packets of data with high bandwidth. To control this type of A-DOS attacks the existing enterprise security devices are not suitable like firewalls, anti-viruses and IDS/IPS Systems why because the adversary not using high bandwidth, spam messages, zombie`s or botnet`s for their attack scenarios.

To safeguard this type of DOS/DDOS or Application denial of service attacks we are having some port hopping mechanisms i.e Port hopping by using Pseudo Random Number Generation (PRNG)based port hopping ,Acknowledgement based port hopping and proactive Reinitialization based on this existing once and their disadvantages like in PRNG attackers can predict the random number generation by using pre-calculated list or based on mathemathical functions .we introduce new port hopping technique i.e True Random Number Generation based port hopping

***Keywords:*** *DOS/DDOS attacks, Application Denial of Service attacks (ADOS), Zombie Systems, PRNG, True Random Number Generation.*

## I. INTRODUCTION

The Internet is becoming increasingly pervasive everyday with the emergence of new wireless technologies and devices that provide anytime anywhere access to the Internet. Today, hosts connected to the Internet include mainly servers and client computers, PDA etc. In the future, other devices such as embedded appliances, cars and anything that runs on electricity will be connected, especially with the emergence of IPv6.These hosts will benefit enormously from the Internet connectivity. However, when they are connected to the Internet, they will become potential attack targets. Presently, conventional enterprise security mechanisms, such as firewall and intrusion detection/prevention systems (IDS/IPS) are used to provide managed security services in an enterprise or ISP framework. As we have more types of devices and applications connected, the current security measures will not be appropriate for the new paradigms.

Consequently, new and simple security methodologies are needed to address the new security concerns of new types of devices or appliances, which typically run very simple applications. One of the main threats of Internet security is DoS (Denial of Service) and DDoS (Distributed DoS) attacks . Such attacks have paralyzed high profile web sites. Although there has been less publicity since then, research and measurements [2] have shown that the attacks are still raging on day to day. As such devices typically have lower networking bandwidth and computing resources, they are more vulnerable to such attacks as compared to servers protected by layers of firewalls and IDS/IPS. Many techniques have been proposed and studied to address the problem of DoS(or)DDoS attacks. However, the drawback of most approaches is they are complex and that significant changes to the Internet routers are generally required.

## II. GENERAL CLIENT SERVER COMMUNICATION

The client–server characteristic describes the relationship of cooperating programs in an application. The server provides a function or service to one or many clients, Iinitiate requests for such services. The model assigns any one roles to the computers in a network: Client or server. A server is a computer system that shares its resources; a client is a computer or computer program that initiates communication with a server in order to make use of a resource. Data, CPUs, printers, and data storage devices are some examples of resources. Clients and servers exchange information in a request-response messaging pattern. The client sends a request message, and the server returns a response message. This sharing of computer system resources is called time-sharing, because it allows multiple applications and servicess to use the computer's resources at the same time. All client-server protocols operate in the application layer.

## III. IP ADDRESS AND PORT NUMBERS IN OSI REFERENCE MODEL

In computer network programming, the **application layer** is an abstraction layer reserved for communications protocols and methods designed for process-to-process communications through an Internet Protocol (IP) computer network. Application layer protocols use the underlying transport layer protocols to establish host-to-host connections.

In the OSI model, the definition of its application layer is narrower in scope. The OSI model states the application layer as being the user interface. The OSI application layer is responsible for displaying data/ Information and images to the user in a human understanding format and to interface with the presentation layer below it. The transport layer is responsible for giving services to the application layer it receives services from the network layer.

### 1.1 Process-to-Process Communication

The responsibility of a transport-layer protocol is to provide **process-to-process communication.** A process is an application-layer entity that uses services of the transport layer. The network layer is responsible for communication at the computer system level (host-to-host communication). A network layer protocol can deliver the message only to the destination computer system. However, this is an incomplete delivery. The message still needs to be handed to the correct process. This is where a transport layer protocol come into picture. A transport layer protocol is responsible for delivery of the message to the appropriate process [3]. Figure 1 shows the domains of a network layer and a transport layer.

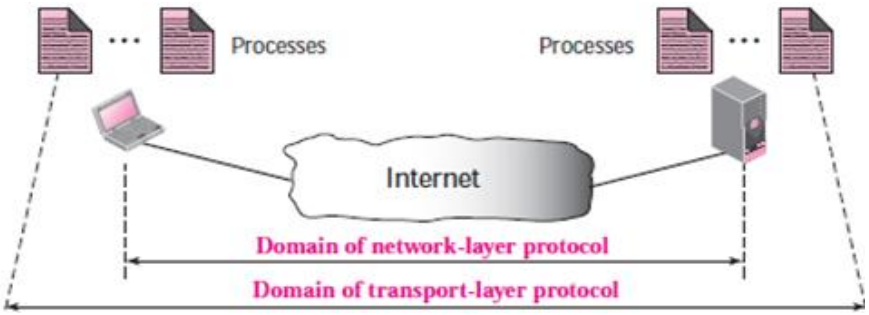


*Figure 1.Network layer vs Transport layer*

Although there are a few ways to achieve process-to-process communication, the most common thing is through the client-server paradigm. A process on the local host, called a client, needs services from a process usually on the remote host system, called a server. Both processes (client and server) have the same name. For communication, we must define the

- Local host
- Local process
- Remote host
- Remote process

The local host and the remote host are stated using IP addresses. To define the processes, we need second identifiers called port numbers. The port numbers are integers between 0 and 65,535.The client program defines itself with a port number, called the ephemeral port number[3]. The word ephemeral means short lived and is used because the life of a client is normally short time. An ephemeral port number is recommended to be greater than 1,023 for some client/server programs to work correctly. The server process must also define itself with a port number. This port number, however, cannot be chosen randomly. If the computer at the server site runs a server process and assigns a random number as the port number, the process at the client side that wants to access that server and use its services will not know the port number. TCP/IP has decided to use universal

port numbers for servers known as well-known port numbers[3]. There are some exceptions to this type of rule, for example, there are clients that are assigned well-known port numbers. Every client process knows the well-known port number of the corresponding server process. For example, while the daytime client process, can use an ephemeral (temporary) port number 52,000 to identify itself instead of original port number, the daytime server process must use the well-known (permanent) port number 13. Figure 2 shows concept of port numbers in transport layer.

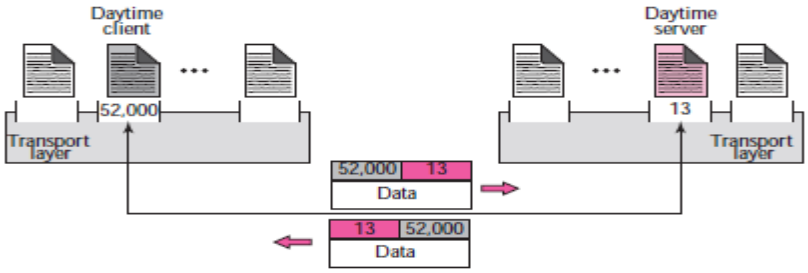


Figure 2.Port numbers used at transport layer

It should be clear by now that the IP addresses and port numbers play different roles in selecting the final destination of data. The destination IP address defines the host among the different hosts in the world. After the host has been selected, the port number defines one of the processes on this particular host

### 1.2 ICANN Ranges

ICANN has divided the port numbers into three ranges: well-known, registered, and dynamic (or private).

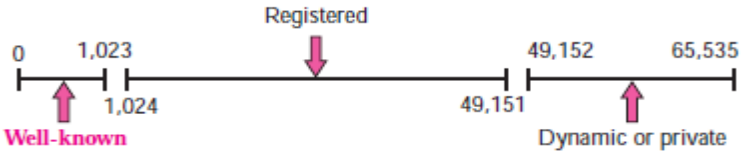


Fig 4.ICANN ranges

**Well-known ports.**

The ports ranging from 0 to 1,023 are assigned and controlled by ICANN. These are the well-known ports.

**Registered ports.**

The ports ranging from 1,024 to 49,151 are not assigned or controlled by ICANN. They can only be registered with ICANN to prevent duplication of port numbers.

**Dynamic ports.**

The ports ranging from 49,152 to 65,535 are neither controlled nor registered. They can be used as temporary or private port numbers. The original recommendation was that the ephemeral port numbers for clients be chosen from this range.

## IV. APPLICATION DENIAL OF SERVICE ATTACKS

Application DoS attacks exploit flaws in the application layer design and implementation to prevent authorized access to the victim's services[4]. They represent a set of attacks on different applications, as they are aimed specifically at disrupting operation rather than diverting the application access controls. Attacks based on exploiting these flaws can offer the attacker a number of advantages over traditional DoS attacks:

- Application DOS attacks are efficient: The attacker may not need as much resource machines at their disposal to successfully complete the attack. Application level attacks target bottlenecks and resources limitations within the application not use large amount of bandwidth and do not require many compromised "zombie" systems.
- The attacks will not be detectable or preventable by existing enterprise security monitoring solutions: Since the attacks do not consume more amounts of bandwidth and it maybe indistinguishable from normal traffic.
- Application DOS attacks are harder to trace: Application level attacks normally use HTTP or HTTPS as their transport communication. Many of these proxy servers do not keep logs of connection attempts and could therefore successfully hide the true origin of the attacking host. Proxy servers can therefore be used to obfuscate the true origin of the attacker.

Distributed denial of service (DDoS) attacks size and frequency grown dramatically as attackers take advantage of botnets and other high-speed Internet access technologies to overwhelm their victim's network infrastructure.

Arbor Networks' sixth annual report[5] on ADDOS attacks: A small serve on this Application DDOS attacks is Worldwide Infrastructure Security Report has shown that not only are DDoS attacks getting larger and more frequent, but they are also becoming more sophisticated as they pinpoint specific applications with smaller, more targeted and stealthy attacks. This means that Internet Service Providers (ISPs) with complex services must now be prepared to protect themselves from different types of DDoS attacks:

1) Volumetric DDoS Attacks: Network infrastructure and servers block with high-bandwidth-consuming flood attacks. Trace backing and detection is easy with enterprise security devices.
2) Application-Layer DDoS Attacks: This attempt to target specific well-known applications such as Hypertext Transfer Protocol (HTTP), Voice over Internet Protocol (VoIP) or domain name system (DNS).Trace backing and Detection is not possible.

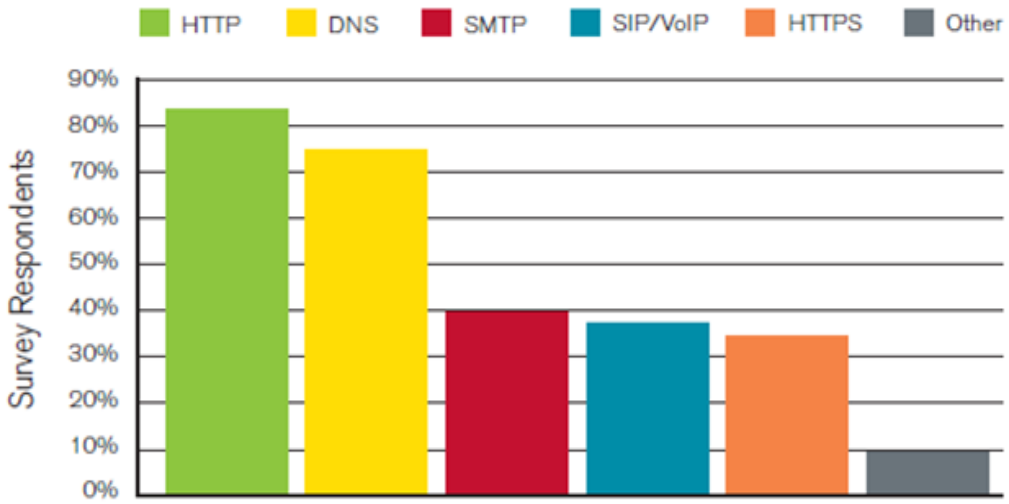


Fig 5. **Layer 7 DDoS Attacks:** Application-layer attacks are on the rise, according to Arbor's sixth annual Report. In other words, ISPs and enterprises must now be prepared to protect themselves from two very different types of DDoS attacks:

**Volumetric DDoS Attacks**

These attacks try to overwhelm the network infrastructure (e.g., switches, routers, etc.) with bandwidth-consuming such as Internet Control Message Protocol (ICMP) or User Datagram Protocol (UDP) floods. Alternatively, they can attempt to overwhelm servers, load-balancers and firewalls by using Transmission Control Protocol (TCP) state exhaustion attacks such as TCP SYN floods and idle session attacks e.t.c.

**Application-Layer DDoS Attacks**

These attacks generally consume low bandwidth and when compared to volumetric attacks. However, they can have a similar impact to service as they target specific characteristics of well-known applications such as HTTP, DNS, VoIP or Simple Mail Transfer Protocol (SMTP) e.t.c.

## V. EXISTING PORT HOPPING TECHNIQUES

This port hopping technique is compatible with the UDP and TCP protocols and can be implemented using the socket communications for the UDP protocol, and for setting up TCP communications [2]. Secure Overlay Service (SOS) that proactively mitigates DoS attacks. It is geared towards supporting services or similar types of communications. The architecture is constructed using a combinations of secure overlay tunneling[2], routing via consistent hashing and filtering mechanisms. Again, the drawback of this scheme is the necessity to make changes to the Internet infrastructure routers. Netbouncer for DoS/DDoS mitigation which proposed to test the legitimacy of arrival incoming packets. This approach is an end point based solution to DoS/DDoS protection, in that changes are made to the servers or clients, but not to the Internet routers. The tests are carried out by the end hosts, and can be conducted at the network layer (IP), transport layer (TCP) application layer. For example, at the IP layer, the end host can test if the IP address in an arrival incoming packet is associated with a live host. Such test can be carried out using ICMP echo messages. The key issue of this approach is to find a simple and effective test.

### 1.3 Random Number Generation Port Hopping Mechanism

To controling Application Denial of service attacks we are having some port hopping and reinitialization mechanisms.

PRNGs are algorithms that use mathematical formulae or simply precalculated list of tables to produce sequences of numbers that appear randomly. A good example of a PRNG is the linear congruential method. A good deal of research has gone into pseudo-random number theory, and modern algorithms for generating pseudo-random numbers are so good that the numbers look exactly like they were really random. Change the server's port numbers dynamically[2] as a function of time. In this, time is divided into discrete slots $S_i$, where i = {0, 1, 2....}, each of duration $\tau$ . In conventional TCP/UDP services, the port number used is unchanged for a specific communication interval. In this scheme, different port numbers are used in different time slots for the same communication service .Let $P_i$ represents the port number used by the server in time slot $S_i$.$P_i$ is determined by equation (1), where k is a shared cryptographic key between the server and the client communication and f is a pseudo-random number generator. $P_i = f(i, k)$ ……(1)

When a client needs to communicate with the server, it will identify the server's current port number $P_i$ using the shared secret key k and the time slot number i. When the server receives packets of data 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 identify if a packet is malicious. As a result, the computational resources needed to detect and filter off the malicious data packets is reduced. In this l as the overlapping time slot factor. This parameter is used to address two issues: the time synchronization error between the client side and the server side, and the transmission delay between them. This mechanism reserves a part of the previous and next time slot to allow for packet exchange taking place near the boundaries of significant time slots. Let $t_i$ be the start of time slot i and $P_i$ be the associated port number. Then $P_i$ is defined to be valid from time $(t_i - \frac{1}{2}\tau)$ to $(t_{i+1} + \frac{1}{2}\tau)$.

For example, When l = 0, a port number is valid only within a specific time port does not increase the chance of a security breach, as a single application is listening on all open ports. Give a generic analysis of the communication success rate over a port-based rationing channel in different attack situations. In this differentiate between directed attacks, where the adversary knows the port used and blind attacks, in which the adversary does not know the port. Not surprisingly, Directed attacks are extremely harmful: that it will have to resort to blind attacks. Our basic idea is to change the filtering criteria in a manner that cannot be predicted by the adversary. This port-hopping approach minimizes the technique of a frequency-hopping spread spectrum in radio communication channel In earlier work, Presented Drum [4], 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 present here. Describe a protocol that allows for DoS-resistant from A, and these acks allow the parties to hop through ports together. However, although the ack-based protocol works well as long as the adversary fails to attack the correct port, once the adversary identifies the port used, it can perform directed attack that renders the protocol useless. By attacking the found data port or simultaneously attacking the found data and ack ports, the attacker can effectively drop the success rate to 0,no port hopping will occur. To solve this matter, there is a time-based proactive reinitialization of the ports used for the ack-based protocol communication, independent of any messages passed in the system.

### 1.4 Ack-Based Port Hopping

We present an ack-based port-hopping[6] protocol, which uses two port-based rationing communication channels, from B to A and vice versa. For simplicity, we assume that $R_{AB} = 2R_{BA} = 2R$. B always keeps two open ports for data reception from A, and A keeps one port open for acks from B. The protocol hops ports upon a successful round-trip on the most recent port used, using a pseudorandom function $PRF^{*}$. In order to avoid hopping upon adversary messages, all protocol messages carry authentication information, using a second pseudorandom function PRF on {0,1}*. If Acknowledgement lost then attacker can attack blind attacks

.

### 1.5 Adding Proactive Reinitializations

Introduce a proactive reinitialization[6] mechanism that allows choosing new seeds for the ack-based protocol depending on time and not on the messages passed in the system. We denote by $t_A(t)|t_B(t)$the local clocks of A and B, respectively, where t is the real time. In this we get that $0 \le |t_A(t) \le \Phi$, $0 \le |t_B(t)-t| \le \Phi$ here also assume that $t_A, t_B \ge 0$. If A reinitializes the ack-based protocol then sends a message to B at time $t_A(t_0)$, this message can reach B at anywhere in the real-time interval$(t_0, t_0+\Delta]$. Therefore, the port used by A at $t_A(t_0)$ must be opened by B at least throughout this interval session. To handle the extreme case where A sends a message at the moment of reinitialization, B must use the appropriate suitable port starting at time $t_B(t_0)-\Phi$. We define $\delta$ as

the number of time units between reinitializations of the protocol and assume for simplicity and effectiveness of resource consumption that $\delta > 4\Phi + \Delta$. Every $\delta$ time units, A sends a new seed value to the ack-based protocol, and B anticipates it by creating a new instance of the protocol, which waits on the new expected ports. Once communication is established using the new protocol instance or once it is clear that the old instance is not going to be used anymore, the old instance is terminated.

### 1.6 PRNG Port hopping Disadvantages

- PRNGs generate random numbers based on mathematical formulae or precalculated list, so attacker can esily predict or know the random numbers.
- Deterministic: Given sequence of numbers may be reproduced later date if know the starting point in the sequence.
- PRNGs are periodic: Means that the sequence will eventually repeat at any point itself .

## VI. NEW TRNG PORT HOPPING

TRNGs extract randomness from physical phenomena and introduce it into a computer system. We can imagine this as a die connected to a computer system. The physical phenomenon can be very simple, like the little variations in somebody's mouse movements or in the amount of time between keystrokes. In practice, however, we have to be careful about which source you chosen. For example, it can be tricky to use keystrokes in this one, because keystrokes are often buffered by the computer operating system, meaning that several keystrokes are collected before they are sent to the programming waiting for them. To a program waiting for the keystrokes, it will seem as though the keys were pressed almost simultaneously, and there may not be a lot of randomness there after all.

However, there are many other ways to get true randomness into computer system. A really good physical phenomenon to use is a radioactive source. The points in time at which a radioactive source decays are completely unpredictable by the persons, and they can quite easily be detected and fed into a computer system, avoiding any buffering mechanisms in the operating systems. Another suitable physical phenomenon is atmospheric noise, which is quite easy to pick up with a normal radio changes. Another one is background noise from an office or laboratory, but you will have to watch out for different patterns. The fan from computer might contribute to the background noise, and since the fan is a rotating device, chances are the noise it produces won't be as random as atmospheric noise. In this way we can implement different TRNG sources and generate different random numbers for port hopping.

Table 1.Comparison of PRNGs and TRNGs

| Characteristic | PRNG | TRNG |
|---|---|---|
| Periodicity | Periodic | Aperiodic |
| Determinism | Deterministic | Nondeterministic |
| Efficiency | Excellent | Poor |

The above table represent the difference between the Pseudo Random Number Generation and True Random Number Generation based on some characteristics.

## VII. CONCLUSION

In this one we are given general client server communication by using IP Address and protocol port number. How adversaries can pose DOS/DDOS and Application Denial of Service (ADOS) attacks in client server environment. Given control measures of DOS/DDOS Attacks (like Firewalls ,Anti-viruses and IDS/IPS) and mitigating ADOS by Pseudo Random Number Generation (PRNG) port hopping based on disadvantages like prediction of next port number by using PRF and pre calculated lists. Introduce secure Random number generation i.e True Random Number Generation (TRNG) based port hopping.

## REFERENCES

[1]. Zhang Fu, Marina Papatriantafilou, and Philippas Tsigas "Mitigating Distributed Denial of Service Attacks in Multiparty Applicationsin the Presence of Clock Drifts" Ieee Transactions On Dependable And Secure Computing, Vol.9, No.3, May/June 2012.
[2]. H. Lee and V. Thing, "Port Hopping for Resilient Networks," Proc. IEEE 60th Vehicular Technology Conf. (VTC2004-Fall), vol. 5,pp. 3291-3295, 2004.
[3]. A Text book of "TCP/IP Protocol Suite", Fourth Edition by Behrouz A. Forouzan
[4]. Stephen de Vries "A Corsaire White Paper:Application Denial of Service (DoS) Attacks"1 April 2004.
[5]. Arbor Application Brief**:** **"**The Growing Threat of Application-Layer DDoS Attacks".2011.
[6]. G. Badishi, A. Herzberg, and I. Keidar, "Keeping Denial-of-Service Attackers in the Dark," IEEE Trans. Dependable and Secure Computing, vol. 4, no. 3, pp. 191-204, July-Sept. 2007.
[7]. http://www.random.org/randomness.