IJETAE
EXPLORING RESEARCH AND INNOVATIONS

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)

International Conference on Information Systems and Computing (ICISC-2013), INDIA.

TO DETECT AND RECOVER THE AUTHORIZED CLIENT BY USING ADAPTIVE ALGORITHM

Anburaj. S[1], Kavitha. M[2]

[1,2] *Department of Information Technology, SRM University, Kancheepuram, India.*

anburaj88@gmail.com, kavitha.mu@yahoo.com

Abstract

Computer Network is collection of computers and other hardware components interconnected by communication channels that allow sharing of information. Network security consists of the provisions and policies adopted by a network administrator to prevent and monitor unauthorized access, misuse, modification, or denial of a computer network and Network-accessible resources. Network security involves the authorization of access to data in a network, which is controlled by the network administrator. Distributed denial-of-service attack (DDoS attack) is an attempt to make a machine or network resource unavailable to its intended users. The previous work, one is acknowledgment-based and the other one depends on synchronized clocks. Acknowledgment loss can cause a situation where a port may remain open for a time interval long enough for an eavesdropping attacker to identify and launch a directed attack to it.

In proposed System an adaptive algorithm, HOPERAA, for enabling hopping in the presence of bounded asynchrony, namely, when the communicating parties have clocks with clock drifts. The solutions are simple, based on each client interacting with the server independently of the other clients, without the need of acknowledgments or time server. There are many network based solutions against DDoS attacks. These solutions usually use routers or overlay networks to filter malicious traffic.

Keywords-- **Distributed Denial of Service, Bigwheel Algorithm, Adaptive Algorithm.**

I. INTRODUCTION

A Denial of Service (DoS) attack is an attempt by the attacker to prevent the legitimate users of a service from using that service. One of the main methods that the attacker will use is depleting the computational resources, such as bandwidth, disk space, or CPU time. The situation is even worse with distributed denial of service (DDoS) attacks, where multiple compromised machines or zombie agent's flood messages or requests of a specific service to the corresponding server in order to make the service unavailable. Common methods to protect systems from DoS and DDoS attacks focus on mitigating packet flooding, as that is the most simple and common method adopted by attackers. Such methods rely on upstream routers that filter or rate-limit the malicious traffic or on secure router overlays. These solutions are suitable for filtering distinguishable network flooding but can be ineffective.

When considering network-based applications, a particularly weak point in this context is that they commonly provide some open port(s) for communication, making themselves targets for DoS attacks.

Adversaries that can eavesdrop messages exchanged by the application can identify open ports and launch directed attacks to those that remain open for long enough time as opposed to blind attacks that can be launched to arbitrary ports, even by non eavesdropping adversaries. Moreover, it is important to note that as an application may e.g. involve complex computations, it could be easier to exhaust its computational resources with small volume of messages, especially when many applications execute in one host and the resources allocated to each application become even smaller.

1.1 Distributed Denial of Service

The primary goal of these attacks is to prevent access to a particular resource like a web server. A large number of defenses against DoS attacks have been proposed in the literature, but none of them gives reliable protection. There will always be vulnerable hosts in the Internet to be used for DoS purposes. In addition, it is very difficult to reliably recognize and filter only attack traffic without causing any collateral damage to legitimate traffic. This paper describes how DoS attacks can be carried out and how a victim can mitigate them in ordinary IP networks.

Especially wireless ad hoc networks have their additional vulnerabilities, but these kind of wireless networks are not the subject of this paper. There are several types of such attacks. An attacker can possibly launch a DoS attack by studying the flaws of network protocols or applications and then sending malformed packets which might cause the corresponding protocols or applications getting into a faulty state. An example of such attacks is Teardrop attack, which is sending incorrect IP fragments to the target. The target machine may crash if it does not implement TCP/IP fragmentation reassembly code properly. This kind of attacks can be prevented by fixing the corresponding bugs in the protocols or applications. However, the attacker does not always have to do its best to study the service if it wants to make it unavailable. It can just flood packets to keep the server busy with processing packets or cause congestion in the victim's network, so that the server might not have the ability to handle the packets from legitimate hosts or even cannot receive packets from them. In order to deplete the victim's key resources (such as bandwidth and CPU time), the attacker has to aggregate a big volume of malicious traffic. Most of the time, the attacker collects many (could be millions) of zombie machines or bots to flood packets simultaneously, which forms a Distributed Denial of Service (DDoS) attack.

II. Related Works

There are many network-based solutions against DDoS attacks. These solutions usually use routers or overlay networks to filter malicious traffic. A good survey about network-based defense mechanisms against DDoS attacks is presented. In this paper,[4] we focus on application-based mitigation. We propose an ack-based port-hopping protocol focusing on the communication only between two parties, modeled as sender and receiver. The receiver sends back an acknowledgment for every message received from the sender [5], and the sender uses these acknowledgments as signals to change the destination port numbers of its messages. Since this protocol is ack-based, time synchronization is not necessary. But note that the acknowledgments can be lost in the network, and this may keep the two parties using a certain port for longer time [1].

If the attacker gets the port number during this time, then a directed attack can be launched under which the communication can hardly survive.

We present an analysis on the sensitivity of this protocol to attacks to cope with that, we also propose a solution that reinitializes the protocol [2][3]. With reinitializing periodically, the sender and receiver can use new seeds of the pseudorandom function to generate different port number sequences, so that the port number sequence used for communication is changed periodical ly. Thus, even though the attacker can launch the directed attack due to the lost of acknowledgment packets, the sender and receiver can continue the communication by reinitializing the protocol. This reinitialization is based on an assumption that the difference of clock values of the two communication parties is bounded in order to make the sender and receiver reinitialize around the same time. In this work, we assume that the differences of clock values can be arbitrary, but the clock rate of each communication party is constant.

III. Existing System

When a communication establish between server to client the port is always open until the acknowledgement is received. After sending information in mean time the DDOS will the bulk of resource continuously to block the server to overcome this in the existing system the DDOS will be found out and their access will be denied.

3.1 Overall Description

The BIGWHEEL algorithm, aiming at meeting the aforementioned goals, functions as the Big Wheel rides at amusement parks: clients queue for the next available compartment. Here each compartment represents a hopping sequence; compartments are deployed in a way that aims at balancing the load among them and also at minimizing the clients' waiting times to initiate contact with the server. Suppose bulk of the resources will come it automatically disconnects the connection in client side. The procedure is described in detail below.

Disadvantage:

- Attacking communication channels
- The messages from legitimate clients are not received to server side

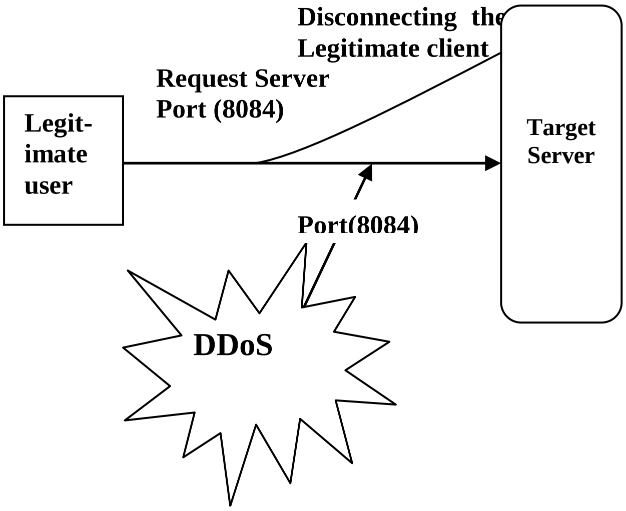

Disconnecting the Legitimate client
Request Server Port (8084)
Legit-imate user
Target Server
Port(8084)
DDoS

Fig 3.1 Bigwheel Algorithm

IV. Proposed System

In proposed System an adaptive algorithm, HOP-ERAA, for enabling hopping in the presence of bounded asynchrony, namely, when the communicating parties have clocks with clock drifts. The solutions are simple, based on each client interacting with the server independently of the other clients, without the need of acknowledgments or time server. There are many network based solutions against DDoS attacks. These solutions usually use routers or overlay networks to filter malicious traffic.

4.1 Overall Description

The most closely related results are the port-hopping protocols presented in. The ack-based protocol in that paper is focused on the communication only between two parties, modeled as sender and receiver. The receiver sends back an acknowledgment for every message received from the sender, and the sender uses these acknowledgments as the signals to change the destination port numbers of its messages. Since this protocol is ack-based, time synchronization is not necessary. But note that the acknowledgments can be lost in the network, and this may keep the two parties using a certain port for a longer time.

If the attacker gets the port number during this time, then a directed attack will be launched under which the communication can hardly survive. To cope with that, a solution that reinitializes the protocol is presented in.

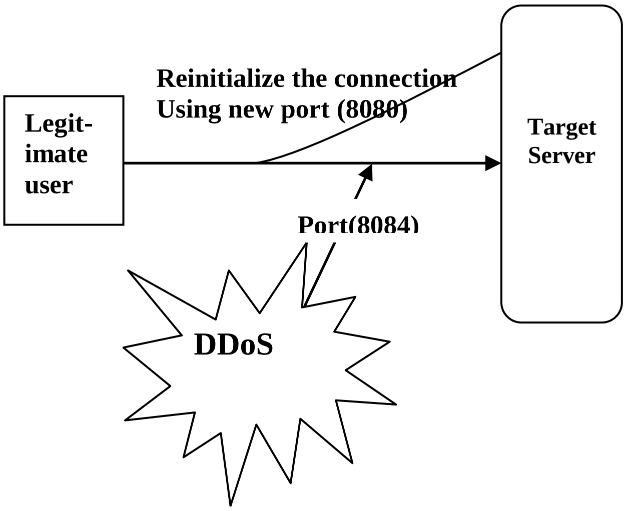


Fig 4.1 Adaptive Algorithm

V. Technologies Used

5.1 WPF (Windows Presentation Foundation)

In our project we are using asp to design the front end process.WPF (Windows Presentation Foundation). WPF combines application UIs, 2D graphics, 3D graphics, documents and Multimedia into one single framework. Its vector based rendering engine uses hardware Acceleration of modern graphic cards. This makes the UI faster, scalable and resolution Independent. The Windows Presentation Foundation is Microsoft’s next generation UI Framework to create applications with a rich user experience. It is part of the .NET framework.

5.2 MVVM (Model, View, View Model)

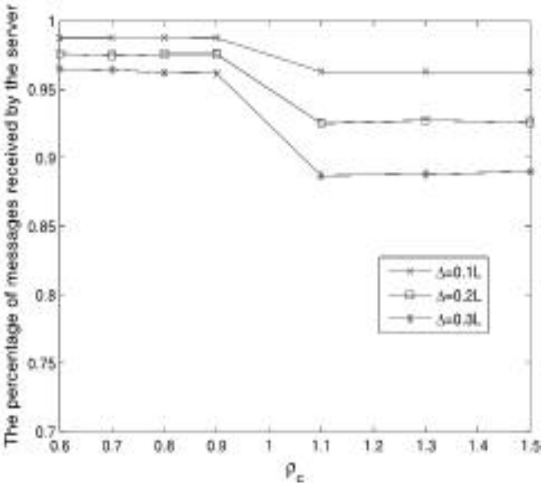
The MVVM (Model-View-View Model) Design Pattern is a Design Pattern introduced recently in the software development community. This Design Pattern is a specialized Design Pattern for WPF and Silver light applications.

5.3 Stored Procedure

A stored procedure is a precompiled group of Transact-SQL statements, and is saved to the database (under the "Stored Procedures" node). Programmers and administrators can execute stored procedures either from the Query Analyzer or from within an application as required. In our project stored procedure it's creating for our back end SQL Server. It is used for execute combination of multi SQL query or commands.

VI. EXPERIMENTAL STUDY

To further study the properties of our protocol, we basically conduct three experiments. These experiments validate some of the analytical results and give complementary measures that are not included in the analytical evaluation due to the subtle and complex relations of different parameters. In particular, we show



- The average number of contact-initiation trails that a client has to do under different parameter settings which conforms to the estimation given in the analysis section.
- The growth of HOPERAA execution interval, which Conforms to the algorithm for estimating the client's clock drift.
- The message overhead for initializing the communication can be amortized within a long time scale due to the growth of the HOPERAA execution interval.
- The messages lost due to the clock drifts can be controlled by adjusting parameters in the protocol.

In the experiments, we assume that the application that uses the proposed port hopping mechanism uses UDP as Transmission protocol. The length of HOPERAA execution interval grows with the number of HOPERAA executions.

VII. CONCLUSION

In this work, we investigate application-level protection against DoS attacks. More specifically, supporting port hopping is investigated in the presence of timing uncertainty and for enabling multiparty communications. We present an adaptive algorithm for dealing with port hopping in the presence of clock-rate drifts (such a drift implies that the peer's clock values may differ arbitrarily with time). For enabling multiparty communications with port-hopping, an algorithm is presented for a server to support port hopping with many clients, without the server needing to keep state for each client individually. A main conclusion is that it is possible to employ the port hopping method in multiparty applications in a scalable way. The method does not induce any need for group synchronization which would have raised scalability issues, but instead employs a simple interface of the server with each client. The options for the adversary to launch a directed attack to the application's ports after eavesdropping is minimal, since the port hopping period of the protocol is fixed. Another main conclusion is that the adaptive method can work under timing uncertainty and specifically fixed clock drifts. An interesting issue to investigate further is to address variable clock drifts and variable hopping frequencies as well.

Acknowledgement

We specially thank our HOD and the staff members of SRM University for rendering us valuable information and encouraging us throughout our research.

REFERENCES

[1] Z. Fu, M. Papatriantafilou, and P. Tsigas, "Mitigating Distributed Denial of Service Attacks in Multiparty Applications in the Presence of Clock Drifts," Proc. IEEE Int'l Symp. Reliable Distributed Systems (SRDS), Oct. 2008.

[2] CERT Advisory CA-1997-28 IP Denial-of-Service Attacks, http://www.cert.org/advisories/ca-1997-28.html, 2010.

[3] K. Argyraki and D.R. Cheriton, “Active Internet Traffic Filtering: Real-Time Response to Denial-of-Service Attacks,” Proc. Ann. Conf. USENIX Ann. Technical Conf. (ATEC ’05), p. 10, 2005.

[4] R. Mahajan, S.M. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S.Shenker, “Controlling High Bandwidth Aggregates in the Network,” ACM SIGCOMM Computer Comm. Rev., vol. 32, no. 3, pp. 62-73, 2002.

[5] D. Dean, M. Franklin, and A. Stubblefield, “An Algebraic Approach to IP Traceback,” ACM Trans. Information and System Security, vol. 5, no. 2, pp. 119-137, 2002.