A Dynamic Cryptographic Approach To Defend Against Distributed DoS Attacks In Multiparty Applications

Anju. K. S [*], Tintu Alphonsa Thomas [†], G. S. Santhoshkumar [†]

PG Scholar[*], Assistant Professor[†], Department of Computer Science &Engineering
Amal Jyothi College of Engineering, Kanjirappally

Abstract

The existing port-hopping to support multi-party applications must support changing port numbers dynamically during the data transfer. Since, there occurs a serious situation that, if anyone of the port within the interval is attacked, then the entire communication can be affected. We can propose a new secure adaptive multi client algorithm called S-BIGWHEEL, which the data which can be encrypted using the RSA algorithm .The encrypted data is chosen to send from the client to the server. During data transfer client port number automatically change with respect to the time interval. The data is transferred through various ports and reaches the server. This encrypted data cannot be viewed by the attackers even when the dynamic port numbers are tracked. The same encryption algorithm will be used for both single and multi client communication.

1. 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. Any attack that can deny system resources or get the system into fault status sometimes even crashes should be identified as a DoS attack. DoS problems are not new; they have been there for more than 20 years and keep evolving over time. The first well-known DoS is the Morris Worm which is an Internet worm developed by a graduate student. This worm can exploit and infect vulnerable systems automatically and then replicate itself. The network can be easily congested with the massive messages created by this worm, so Morris Worm is definitely a DoS, even though it was developed without malicious intention. From then on, more and more DoS incidents were observed and reported. Figure 1 shows application-layer attacks are on the rise, according to Arbors sixth annual Report [6].

Following the rapid growth of the Internet, by taking its amazing ability of information searching, retrieving and exchanging, more and more social services rely on the network applications and communications. These services include scheduling travel itineraries, receiving/reporting information about severe weather or potential disasters, e-commerce, on-line medical diagnostics and scheduling emergency management events, etc. Any denial of such services can cause huge damage, not only loss of money but may also loss of human lives.

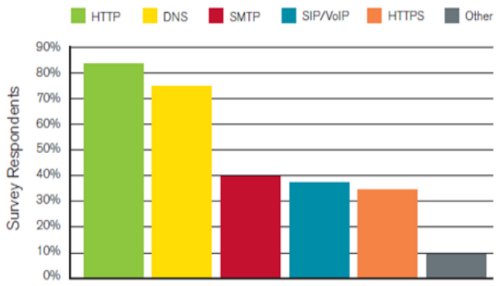

Figure1.Layer7 DDoS Attacks-Arbors 6th annual Report.

Many network-based applications commonly provide some open port(s) for communication, making themselves targets for DoS attacks. Attackers that have the ability of eavesdropping messages exchanged by the application can identify open ports and launch directed attacks to those ports as opposed to blind attacks that can be launched to arbitrary ports, even by non-eavesdropping adversaries.

The application parties communicate via ports that change periodically over time, according to a pattern known by both the sender and receiver, such as a (pseudo)random sequence with common seed-called port hopping. One of the critical issues involved in port-hopping is synchronizing communication parties [3]. Two main kinds of

synchronization mechanisms are acknowledgment-based and the other one depends on synchronized clocks.

1. A port hopping against DDoS

The solution to overcome a very common vulnerability, present at the application layer, in which certain programs may open well-known ports in order to perform whatever action they're meant to do thus, an attacker, can eaves drop some packages, discover which port is being used and launch a directed attack over such ports[5]. Taking this scenario in consideration, the solution studied is based on the idea that the parties involved are capable of communicating with each other hopping in between different available ports over time. In order to achieve such behaviour, and overcome the need of a global synchronization mechanism in the system, two algorithms were proposed; (1)Secure-BIGWHEEL, which is used for servers to support communication with multiple clients, in a port-hopping manner and; (2)Secure-HOPERAA, used for servers to support communication with single clients, synchronous port hopping in the presence of clock-drifts[4].

The port hopping where 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. The main strength of the mechanism lies in the simplification of both the detection and filtering of malicious attacks packets and that it does not require any change to existing protocols. 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]. We performed both theoretical analysis and empirical studies through actual implementation to study the effectiveness of the scheme against DoS/DDoS flooding attacks. Our experiments show that the port hopping technique is effective in detecting and filtering malicious traffic, and hence improved the reliability of good traffic flow.

3. Implementation of new secure multi client protocol: S-BIGWHEEL

The multi client communication is based on a idea: since each client considers the servers clock as the reference clock, it can interact with the server independently of the other clients. 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.

The existing BIGWHEEL to support multi- party applications must support changing port numbers dynamically during the data transfer [1]. Since, there occurs a serious situation that, if any one of the port within the interval is attacked, then the entire communication can be affected. We can propose a new secure adaptive multi client algorithm called Secure-BIGWHEEL, which the data which can be encrypted using the RSA algorithm .The encrypted data is chosen to send from the client to the server.

During data transfer client port number automatically change with respect to the time interval. The data is transferred through various ports and reaches the server. This encrypted data cannot be viewed by the attackers even when the dynamic port numbers are tracked. The same encryption algorithm will be used for both single and multi client communication. The procedure is described in

Buffer B stores reply messages that are waiting to be sent.

- receiving contact-initiation message:

receive$< Init; timestampi, key >$

$h_c(t_1 \longleftarrow$ timestamp

$t_2 \longleftarrow Time_{now}$

$/* p_j^i$ is the i_{th} port number in hopping sequence j.

Suppose its open time is the closest to the current time. $*/$

$\lambda_j \longleftarrow$ seed for hopping sequence j

$\sigma \longleftarrow$ the corresponding

index value for p_j^i in hopping sequence j put the reply message.

$< ReMsg, \lambda_j, \sigma, timestamp, hc(t_1, t_2, key >$ into buffer B

- sending reply messages:

whenever a new worker port is opened

send all the reply messages in buffer B to the corresponding clients Clear B

detailed below.

Algorithm for server using :(Secure-BIGWHEEL)

When using BIG WHEEL, worker ports still remain open for L + μ units of time but now the Server will support m port number sequences instead of just one, as in the previous section this afford more clients and also decrease the maximum waiting time for each one of them. In the Clients side, by using λ and the pseudo-random function fψ it is possible to generate different port number sequences if different values of are given [2]. Apart from these changes, the phases previously explained and the actions performed in each one of them are the same, so when the server receives a contact-initiation message from the Clients, it will send the reply at the closest opening time of a worker port (considering all m sequences) along with the corresponding value of λfor the sequence to which that worker port belongs. In data transmission C sends encrypted data messages to the worker ports of S. After C gets the reply from the server in the contact-initiation part, C has the seed λ for the pseudorandom function f to generate the sequence of the worker ports. The open interval of the worker ports is L + μ time units, where L > μ . The new worker port will be opened μ time units earlier than the closing time of the old one. When S receives the contact-initiation messages from C, it will send the reply message at the time when the next worker port is opened, and the integer

$p_{old} \longleftarrow f\psi(\lambda,\sigma)$

$p_{new} \longleftarrow f\psi(\lambda,(\sigma+1))$

/* P_{old} is the destination port in the current period, and P_{new} is the destination port for the next period. */

- sending data messages

while has more data to send **do**

send < $encrptdata, P_{old}, key$ >

if($iL-\mu \le T_c \le iL$)**then**

send< $encrptdata, P_{new}, key$ >

end if

end while

- changing the destination port

if($\{T_c = iL\}$)**then**

$p_{old} = p_{new}$

$p_{new} \longleftarrow f\psi(\lambda,(\sigma+i+1))$

end if

Algorithm for secure data transmission- client side:

σ has the value for generating the next worker port. When C gets the integer σ from Ss reply, it will send the data messages immediately to the port computed from fψ(λ,σ) C has a timer Tc which will be assigned to 0 when C receives the reply message from S. Tc increases at the same rate as the local clock of C.

4. Experiments and Results

The simulation was done using NS-2 simulator [7] to evaluate the performance of our DDoS detection with results obtained from the experiment. We tested in Ubuntu 13.04 environment. This section introduces the experimental setup and reports performance results.

4.1 Experiment study

Our simulation includes 3 clients, 1 gateway routers and 1 server as shown in Figure.2.We are using two agents in ns2, BIGWHEEL agents and RSA agents for encryption.RSA is an asymmetric security protocol. Two different keys are used for encryption and decryption processes. In symmetric key protocol there is key- exchange problem. But in RSA there is no key-exchange problem as two different keys are used for encryption and decryption purposes.

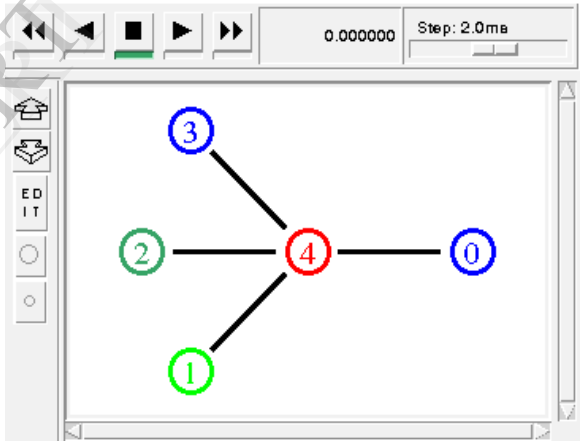

Figure.2. DDoS implementation in ns2

4.2 Performance Evaluation

We will run two different simulations; both simulations will use a Δ = 0.1 and they will set a common background on how the algorithm behaves under the assumption that the Clients Clock Drift ρ is constant and greater than the servers. Regardless of the value of ρ, by looking at Figure.3 we can observe that the S-BISWHEEL Execution Intervals increase exponentially.

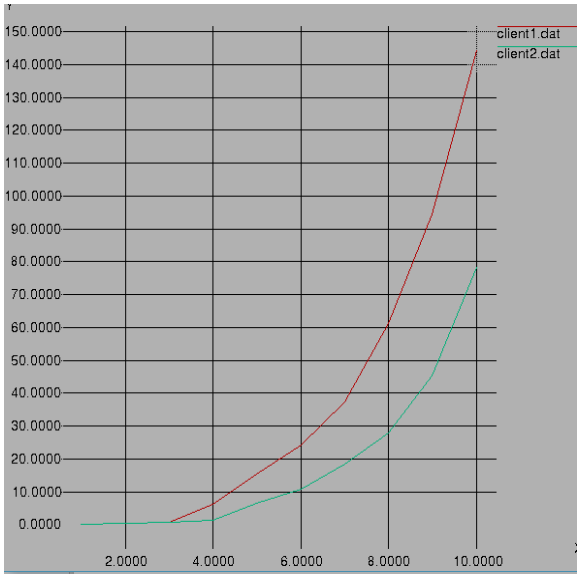

Figure.3 Execution Interval grow rate for a simulation with $\rho = 3$

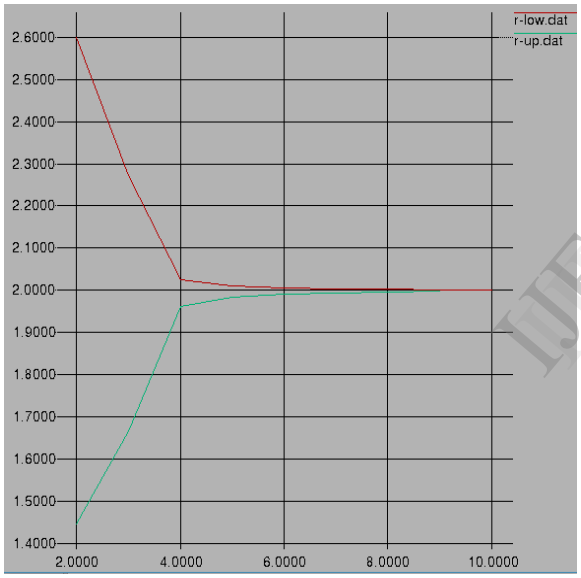

Figure.5 The r-Low and r-Up as the simulation progresses, at $\rho = 2$

We can see that rho-Low and rho-Up both remain in between the range ρ and, effectively approximating to the values defined for each simulation; in fig5 we observe that the range slowly closes around 2.

5. Conclusion

In this paper we have presented a detailed study on how to mitigate DoS and DDoS attacks in the presence of clock drift. In this work, we investigate application-level protection against DoS attacks. An algorithm is presented for a server to support port hopping with many clients. The new secure adaptive multi client algorithm called S-BIGWHEEL, provide more secure communication with little overhead. A main conclusion is that it is possible to employ the port hopping method in multiparty applications in a scalable way. Another main conclusion is that the adaptive method can work under timing uncertainty and specifically fixed clock drifts.

In the Future work, we can extend it to prevent Network attacks. There are many types of network attack, we investigate to prevent server from all such types of network attacks.

6. 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*, 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. Lempel and H. Greenberger, Families of Sequences with Optimal Hamming Correlation Properties, *IEEE Trans*. Information Theory, vol.IT-20, no. 1, pp. 90-94, Jan. 1974.

[4] K Hari and T. Dohi, Sensitivity Analysis of Random Port Hopping, *Proc.Seventh Intl Conf. Ubiquitous Intelligence Computing and Seventh Intl Conf. Autonomic and Trusted Computing (UIC/ ATC), pp.* 316-321, ct.2010.

[5] 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, 2007.

[6] Preventing ADDOS Attack by Using Secure TRNG Based Port Hopping-*AJER*-2013

[7] Introduction to Network Simulator NS2-Teerawat Issariyakul, Ekram Hossain-*Springer*