SYN Flood Attack Prevention using Particle Swarm Optimization in Cloud Computing Environment

Zonayed Ahmed
Lecturer
Department of Computer Science and Engineering,
Stamford University Bangladesh
Dhaka, Bangladesh

Maliha Mahbub
Lecturer
Department of Computer Science and Engineering,
Stamford University Bangladesh
Dhaka, Bangladesh

ABSTRACT

Security issues in Cloud Computing is growing as it continues to offer innovative business model and collaboration capabilities for organizations to boost productivity. There are numerous security issues for cloud computing as it encompasses many technologies including networks, virtualization, resource scheduling, load balancing, concurrency control and memory management. A Cloud infrastructure that comprises the vulnerability of Denial of Service (DoS) attacks denies legitimate users from accessing information or services. A DoS attack can be launched in the transport layer using the very old, but still effective, SYN Flood technique. In a SYN flood attack the attacker sends a flood of TCP SYN requests that gets the server busy without actually completing the three-way handshake procedure used in the setup of TCP sessions. This paper presents a Particle Swarm Optimization (PSO) based approach to enhance the defence mechanism of the system against such attacks. The theoretical analysis and simulations show that the proposed optimization model analyses the situation of under attack server and based on the intensity of the attack situation, it provides the best solution to the server. The server can tune itself dynamically to the optimized solution which increases its chance against SYN Flood attacks**.**

Keywords

SYN Flood Attack, DoS attacks, Cloud Computing, PSO Algorithm

1. INTRODUCTION

Cloud Computing is now emerging as a mechanism for high level computation as well as serving as a storage system for resources. Cloud services include delivery of software, platform to develop applications and providing a complete infrastructure over the Internet. A cloud is more economical compared to other distributed services, with one of the appealing aspects of cloud computing being pay-as-you-go. Sustainability can be ensured by maintaining a strong infrastructure with abundant availability of resources. Virtualization, elasticity, scalability, load balancing, instant service and pay-as-you-go are the main properties that convert a data centre into a cloud. With virtualization the users have the privilege of running their instances with a variety of application options provided by the cloud provider, or the user's chosen applications, without any initial cost. [1].

Cloud Computing could be the most promising improvement in IT infrastructure area in recent times, but a great deal of work is still warranted in the domain of security to minimize the threats. Unfortunately, the services provided by a cloud could be hindered if the system is compromised by some denial of service (DoS) or distributed denial of service (DDoS) attacks. This problem is serious when the attacker send huge packet to destination by spoofing IP address. In this condition the victim's system crash and over all cannot service any type of request whether, legal or illegal [4]. SYN flooding attacks aim to exhaust TCP buffer space and do not affect the parameters such as link bandwidth, processing resources and so on [2].

In virtualization environment, resources such as CPU, memory, disk and network are shared by VMs and the host. An attacker aims to exhaust the resources from a physical host in order to deny service to the other VMs in the machine. The main properties of a cloud can, unfortunately, enhance a DoS attacks. Due to the load balancing feature in a cloud, if a single adversary sends spoof packets to a cloud server, once the server becomes overloaded it will offload its validating tasks to the nearest server. The newly assigned server also eventually becomes overloaded and offloads its task to another server, thus propagating the flooding attack over the entire network. The cloud properties of elasticity and scalability provide further disadvantages during an attack. A server overloaded with enormous validation tasks will scale up to engage more of its resources and computation nodes to validate the spoof packets, continuously exhausting each server with its assigned resources.

As a novel approach to secure the cloud communication, this paper shows the utilization of Particle Swarm Optimization, a well known optimization technique generally used in neural network and fuzzy control systems, to find optimal solution to the problem. The derived result leads to an optimal solution which monitors the system performance continually and then dynamically tries to direct the system to the best defense position by appropriate setting of some system parameters.

2. RELATED WORKS

There are some proposed defenses for this TCP SYN flooding as well as other DoS attacks. Long et al. [8] proposed two queuing models for the DoS attacks in order to obtain the packet delay jitter and the loss probability. Wang et al. [9] studies the DoS attacks analytically by using a more general queue model, a two-dimensional embedded Markov chain, which can more accurately capture the dynamics of the actual DoS attacks. *D-WARD* [10] , [11] scheme aims to detect DDOS flooding attack traffic by monitoring both inbound and outbound traffic of a source network and comparing the network traffic information with predefined normal flow models. MULTOPS [12] is a heuristic and a data-structure that network devices (e.g., routers) at the source subnet can use to detect and filter DDoS flooding attacks. A Graphic Turing test [13] could be a suitable approach to defend DoS attacks for a high traffic web server to distinguish between human interaction and automated attack zombies. In another study, cloud security was ensured by increasing virtual machine security [14]. In their work, VMs with obsolete packages were marked with ICS (Image Creation Station) and the starting of a VM with obsolete packages was prevented by XGE (Xen Grid Engine) [15]. HTTP DoS (HDoS) and XML DoS (XDoS) attacks are comparatively easy to accomplish but twice as hard for the security experts to protect against [16]. Chonka et al. [16] in their study claimed to provide protection against HDoS and XDoS attacks using a trace back mechanism they named Cloud Trace Back (CTB). Hop-count filtering [17] uses information about a source IP address and its corresponding hops from a destination that are recorded in a table at the destination side when the destination is not under attack. Khan and Traore [18] analyzed the influence of DoS attacks on three parameters: the queue-growth-rate, the arrival rate, and the response time, which were used for the attack detection. However, providing the solution to the server by optimizing the control parameters, using swarm intelligence while improving the system performance against attack using the same parameters is a novel approach compared to the other solutions.

This paper defines the issue of SYN flood attack in cloud environment as an optimization problem and solves it with PSO algorithm utilizing the control parameters and delivering the parameters to the victim server to help improve its defense against the attack.

3. SYN FLOOD ATTACK AND PSO ALGORITHM

3.1 TCP SYN Flood Attack

TCP is a connection oriented protocol that needs “handshaking” to start communication in client-server architecture. A TCP connection is established in what is known as a 3-way handshake. When a client efforts to start a TCP connection to a server, first, the client requests a connection by sending an SYN packet to the server. Then, the server returns a SYN–ACK, to the client. Finally, the client acknowledges the SYN–ACK with an ACK, at which point the connection is established and data transfer commences [3].

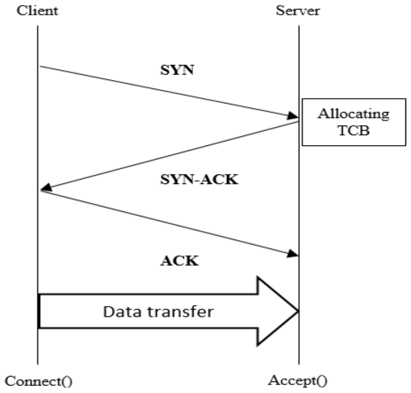

Fig 1: TCP Three Way Handshake

The above process is completed when client and server are legal and they have legitimate request. If the client is an attacker, then this process in the second step is stopped and leading to a waste of server’s resource. Actually attacker by spoofing IP addresses and attempts to fill the victim’s back log buffer cause to wasted resources and victim system is down to the received request from legitimate users, then denial of service is occurred. In other words an attacker based on SYN-Flooding takes place according to the following figure:

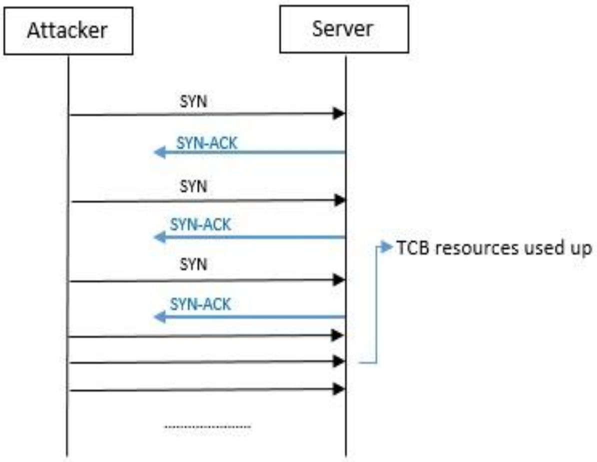

Fig 2: TCP SYN Flood Attack

Since the TCP protocol of communication protocol is very important in today’s electronic society, so completely avoidance of this kind of attack is not possible. Therefore in such circumstances where it is required to use this protocol inherently we cannot completely prevent unwanted conditions which occur because of the security weakness of such protocols. Among of the methods that have already been proposed to detect and deal with this type of attacks, the method based on optimization and collective intelligence is more applicable. Accordingly in this paper, widely acclaimed PSO algorithm has been modified to prevent SYN flood attack..

3.2 Particle Swarm Optimization (PSO) Algorithm

The method of PSO optimized was introduce in 1995 by James Kennedy and Russel Eberhart. PSO algorithm is based on information sharing between members or groups of particles in the search space or swarm which are called particles. All the particles in the search space have fitness value that is obtained by an objective function and the aim of objective function is to optimize the proposed solution. Each particle moves in the problem search space looking for the optimal position. A particle adjusts its position as time passes according to its own personal experience as well as according to the experience of neighbor particles. The particles are essentially characterized by two properties: the particle position, which defines where the particle is located with respect to other solutions in the search space, and the particle velocity, which defines the direction and how fast the particle should move to improve its fitness compared to the rest of the neighbor particles.

The basic PSO algorithm, which minimizes an objective function *f* (*x*) of a variable vector *x* defined on a n-dimensional space, uses a swarm of *m* particles. Each particle *i* of the swarm is associated with a position in a continuous n-dimensional search space. Similarly, the velocity is also an n-dimensional vector. Denoting with x^{ki} and V^{ki} respectively the position and velocity of particle *i* at iteration *k* of the PSO algorithm, the following equations are used to iteratively modify the velocities of the particles and positions:

$$V_t^{k+1} = wv_t^k + c_1r_1(Lbest_t^k - x^k) + c_2r_2(Gbest^k - x^k)$$

$x^{k+1} = x^k + V_t^{k+1}$

where V_t^{k+1} represents the distance to be traveled by the particle from its current position in the *kth* iteration, x^{k+1i} represents the particle position in the *kth* iteration, *w* is the *inertia* parameter that weights the previous particles velocity, i.e., *w* controls the impact of previous historical values of particle velocities on its current velocity, *pbest* represents its best personal position (i.e. its own experience), and *gbest* represents the best position among all particles in the population. Parameters c_1 and c_2 are positive constant parameters called acceleration coefficients which control the maximum step size of the particle and determine the relative “pull” of *pbest* and *gbest*. The parameter c_1 is a factor determining how much the particle is influenced by the nostalgia or memory of his best location, and c_2 is a factor determining how much the particle is socially influenced by the rest of the swarm. Parameters r_1 and r_2 are two random numbers uniformly distributed in [0, 1] that are used to weight the velocity toward the particle personal best and toward the global best solution [19].

According to the velocity equation, a particle decides where to move next, considering its own experience, which is the memory of its best past position, and the experience of the most successful particle in the swarm. The new particle position is determined by adding to the particle current position the new velocity computed.

4. PROPOSED MODEL

4.1 Structure of the Problem Based on PSO

The proposed approach considers a cloud infrastructure where the physical server provides sharing of different resources such as CPU, memory to multiple VMs. The VMs get access through an interface and one of the VMs is considered to be an attacker which acts as a source of TCP SYN flood packets and the server is considered to be under DoS attack.

The attacker VM uses any spoofing tool to scan the IP address of the other VMs in the network and continuously sends SYN requests to the server. The server must answer with a SYN–ACK and must allocate a memory space to this half-open connection. In other words, attacker tries to exhaust the memory space allotted to the TCP protocol. To model this attack, we consider only one resource i.e. the memory space of the victim server and consider it as a queue thus employing queuing theory.

In this mechanism, a queue has been used to maintain both attack and legal requests. When a request arrives at the system, it is allocated a buffer space in the backlog queue. If queue is full, any kind of request whether legal or attack is discarded. After creating a connection between source and destination, the request is not removed from the queue until it completes the three way handshaking of TCP protocol. Assume that in this model, each half open connection is held for at most the period of time t seconds and at most p concurrent half-open connections are allowed. The arrivals of the regular connection request packets and the attack connection request packets are both Poisson processes with arrive rates λ_1 and λ_2, respectively.

As the proposed system is under SYN flooding attack, the half open connections which are actually sent by the attacker would take much longer than the holding time t of the queue and eventually exhaust all the memory space of the buffer such that no new request can be accepted by the victim server.

It goes without saying that the time duration at which the system reaches to this deadlock point is affected by the values of t and p. For example, when t increases, attack half open connections stay longer in the queue causing to consumption of the memory spaces. As a result of this, less percentage of the buffer space is allocated to legal requests, and major part of this space will be occupied by the attacker requests.

Since the main aim here is to reduce the buffer occupancy of attack requests, this paper uses the values of t and p as the control parameters which are dynamically tuned to their best defense positions by using the PSO algorithm. Figure 3 illustrates an abstract framework for the proposed model. In this figure, a server has been shown with a single queue where the SYN requests from the VMs are held waiting for their services. The PSO algorithm continuously assesses the server performance in terms of the control parameters and then it used as an objective function. PSO generates the best possible values of the parameters t and p using the objective function and the server uses the parameters to achieve the required defense against attack.

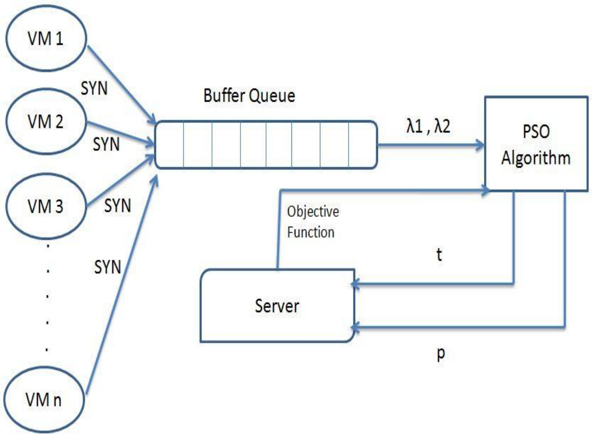

Fig 3: Basic Model of the Proposed System

4.2 Objective function Analysis

Since PSO optimizes the search space by minimizing the objective function, this is the crucial part. The goal here is to minimize the occupancy of attack requests in the queue and preventing the loss of legal requests while simultaneously increasing the space occupied by legitimate requests. So, the objective function of this problem is formulated as

Objective Function=

min [(Attack half open connection*rate of packet loss) / (Regular half open connections)]

By the objective function at time and comparing it to the objective function obtained from the previous steps of all particles in swarm, the parameters t and p of the queue at any moment can be calculated. This objective function reflects the intention of the model i.e. maximization of RRBOP and at the same time minimization of Ploss and ARBOP. According to the PSO algorithm values of t and p will be updated by the following equations:

$V_t^{k+1} = wv_t^k + c_1r_1(Lbest_t^k - t^k) + c_2r_2(Gbest_t^{\ k} - t^k)$

$t^{k+1} = t^k + V_t^{k+1}$

$V_p^{k+1} = wv_p^k + c_1r_1(Lbest_p^k - p^k) + c_2r_2(Gbest_{\ p}^{\ k} - p^k)$

$p^{k+1} = p^k + V_p^{k+1}$

Here, (*$Lbest_t^k$* , *$Lbest_p^k$*) are the local best positions and (*$Gbest_t^{k,}$ $Gbest_p^k$*),are the global best positions. As stated earlier, PSO algorithm minimizes the objective function and each particle of the swarm tries to tune its positions i.e. t and p seeking the best position in the whole swarm. The best defense positions of the parameters t and are therefore utilized by the server to defend the flooding attack.

The parameters c_1 and c_2 enable the movement of the particle for its personal best position towards the global best position and their values should satisfy the condition, $c_1 + c_2>4$. In this case, the values of acceleration parameters have been chosen to be 2.

The working procedure of the whole proposed system is demonstrated by the following flow chart in figure 4.

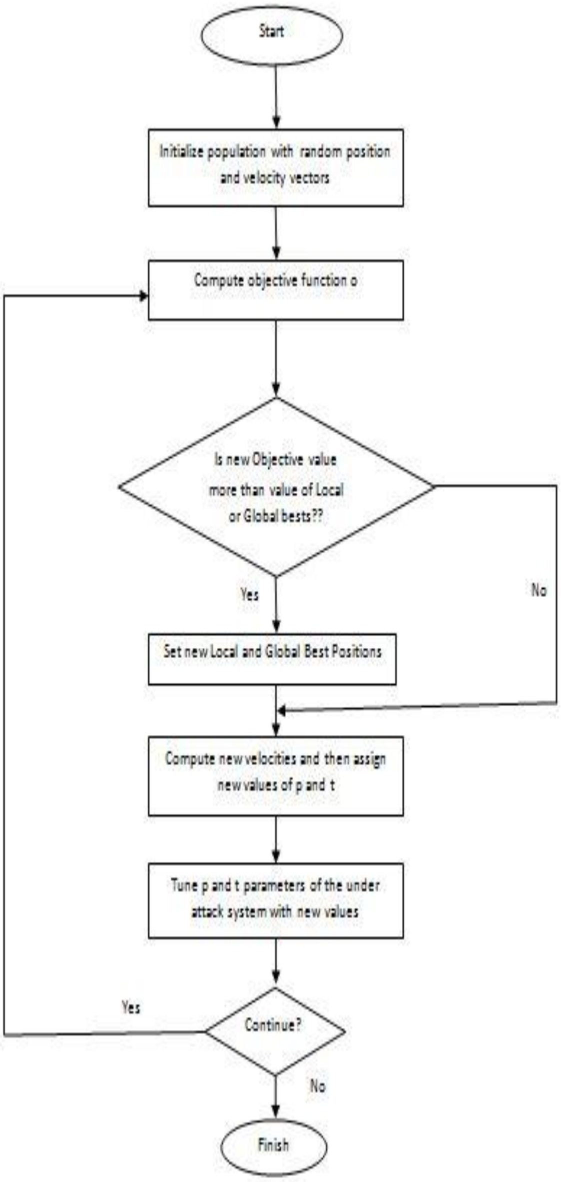
Start
Initialize population with random position and velocity vectors
Compute objective function o
Is new Objective value more than value of Local or Global bests??
Yes
No
Set new Local and Global Best Positions
Compute new velocities and then assign new values of p and t
Tune p and t parameters of the under attack system with new values
Yes
Continue?
No
Finish

Fig 4: Flow Chart of PSO Algorithm

5. SIMULATION AND RESULT

For performance analysis, MATLAB has been used for simulating the results of the PSO algorithm. The list of parameters that have been used are noted in the following table.

Table 1. List of Parameters

PARAMETERS	VALUE
INERTIA, W	1.0
CORRECTION FACTORS C_1 AND C_2	2.0
ARRIVAL RATE OF REGULAR REQUESTS λ_1	10
SWARM_SIZE	49

To evaluate the performance of the PSO algorithm in this case, different attack intensities are applied to the algorithm determining their effect on the values of the parameters and overall system performance under various attack situations. The attack intensity is defined as $k = \lambda_2 / \lambda_1$ where λ_1 is the arrival rate of the regular SYN request packet whereas λ_2 is the arrival rate of attack request packets and both are Poisson arrival rates. The objective function is formulated as follows

Objective function =
$\min [(\lambda_2 (p- \lambda_1 t) / (\lambda_1 (p- \lambda_1 t) + \lambda_1 t(\lambda_2+ \lambda_1))]$

Here, t = holding time of half open connection
p= total number of half open connection in the queue

The following scenarios provide the evaluation of the PSO algorithm in various attacking situations.

5.1 Low Attack Intensity (k=0.1)

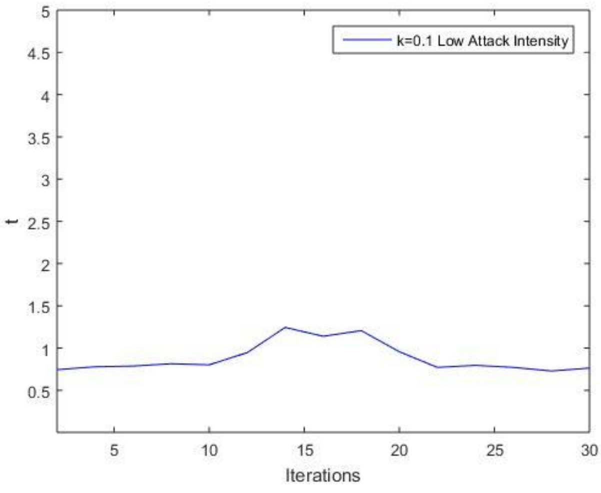

Fig 5: Value of parameter t at low attack intensity, k=0.1

In the figure 5, the behavior of the system is shown where the victim server goes under a SYN flooding attack whose intensity is k = 0.1 i.e. low attack intensity. This figure shows the varying values of the parameter t, which is the holding time of each half open connection in the parameter.

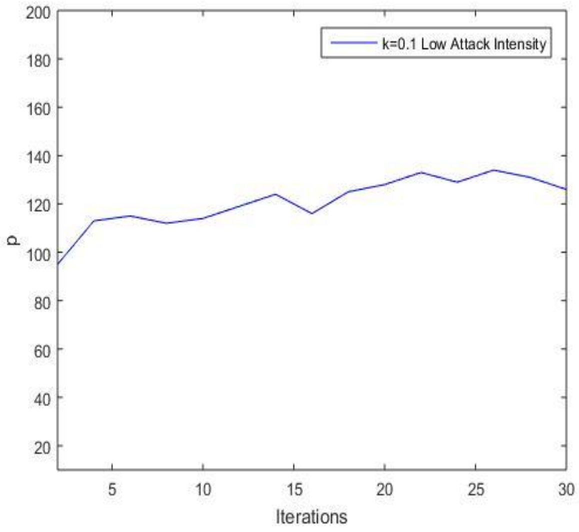

Fig 6. Value of parameter p at low attack intensity, k=0.1

The figure 6 shows the varying values of the parameter p, which is the maximum number of half open connections allowed in the queue adjusted with PSO algorithm where the attack intensity is k= 0.1, i.e. the attack intensity is low.

5.2 Medium Attack Intensity (k=1)

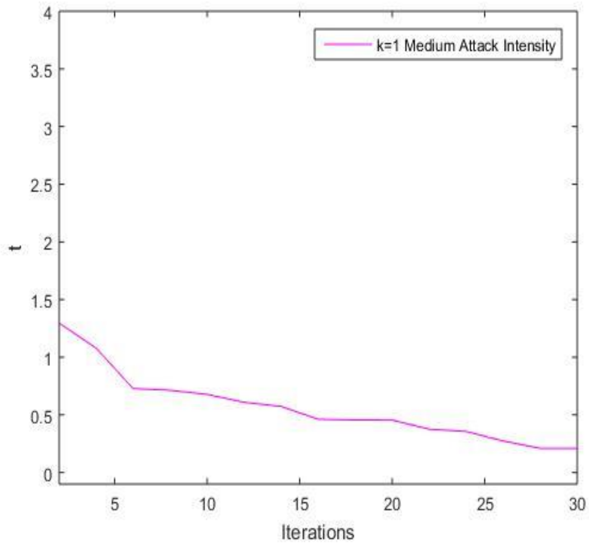

Fig 7: Value of parameter t at medium attack intensity, k=1

The figure 7 shows the behaviour of the system in medium attack intensity k= 1. The figure clearly shows that when the attack intensity increases, the value of t is reduced by the PSO algorithm in order to defend the SYN attack requests.

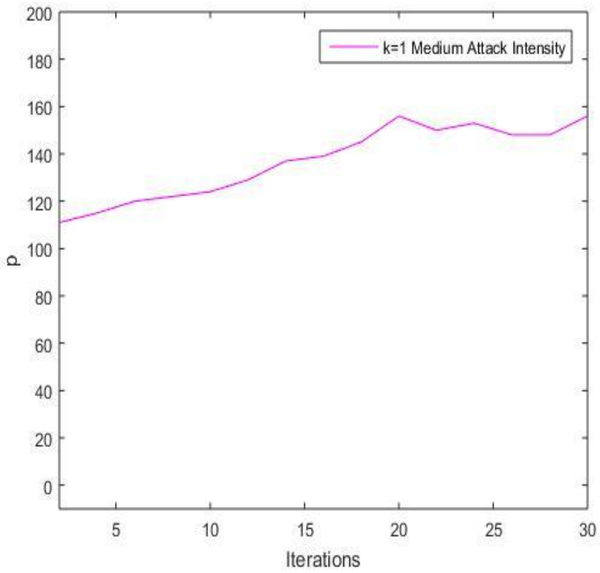

Figure 8. Value of parameter p at medium attack intensity, k=1

The above figure 8 shows that PSO algorithm increases the value of number of half open connections in queue, p is increased when the attack intensity increases so that new requests can be allocated.

5.2 High Attack Intensity (k=2)

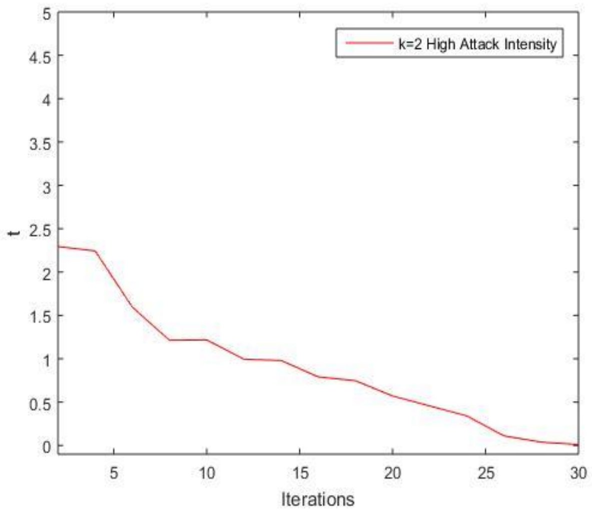

Fig 9: Value of parameter t at high attack intensity, k=2

The above figure 9 shows that when the system is under high attack intensity, then the parameter t is further reduced by the PSO algorithm to eliminate the attack requests from the queue.

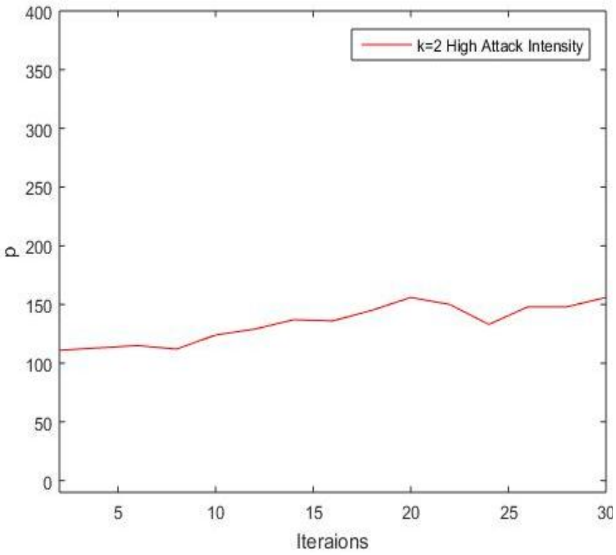

Fig 10: Value of parameter p at high attack intensity, k=2

The figure 10 shows that when the attack request increases, the PSO algorithm increases the value of p so that the legitimate requests does not get lost while the server is under higher risk of SYN attack.

From all the above simulations and result, it is clear that as the attack intensity raises, the PSO algorithm reduces the parameter t to eliminate the attack half open connections faster from the queue while simultaneously increases the value of p to accommodate new incoming requests which leads to a lesser possibility of packet loss. The overall improvement of system performances can be shown in following two figures.

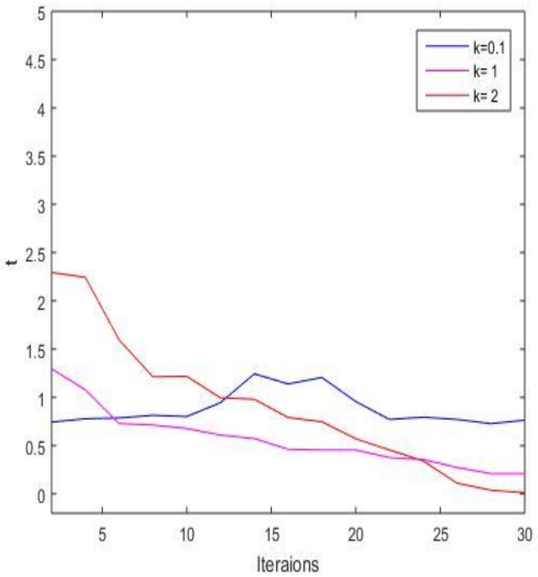

Fig 11: Value of parameter t at different attack intensities.

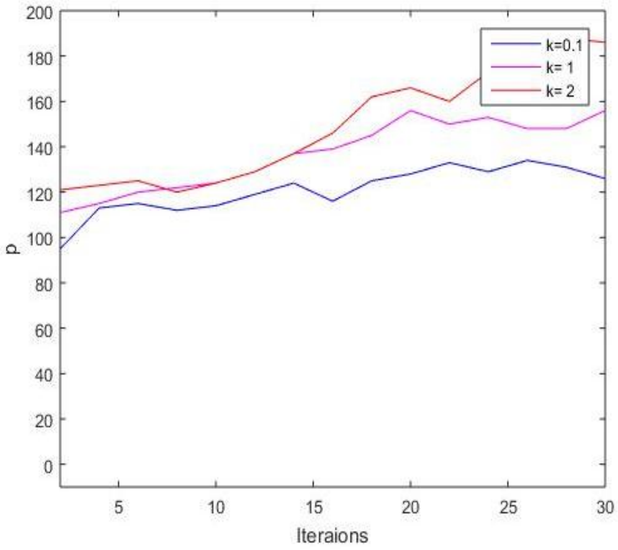

Fig 12: Value of parameter p at different attack intensities.

The above figure demonstrates the overall improved defence behaviour of the system under attack when PSO algorithm is employed.

6. CONCLUSION

This paper launches an optimized approach to reduce the effects of SYN flooding attacks in cloud computing environments where the virtualized nature of the cloud can also increase the vulnerability of the machines. For this purpose, this paper models the problem into an optimized structure using PSO algorithm which uses the under attack system's performance to dynamically set the values of control parameters t and p to their best positions. This results in reducing the possibility of packet loss due to flooding attack and increasing the memory space of the system to hold on legal requests. Since the parameters are dynamically set, the server can tune it to their appropriate values compared to other systems where these values are static. With appropriate simulation, the performance of the system using the proposed model under different intensities of attack has been tested which shows the improvement of the defence mechanism of the server. The experimental data from this work can be further adopted to include other distributed DoS attacks and eventually collaborate into effective scheduling systems too. The scope of the data collected from this work can be extensively stretched to include virtualized and parallel systems. The future scope of this work is to use more than two variables to define server issues more elaborately such as packet loss rate which would help fight the the security threats more effectively

7. REFERENCES

[1] Yang X, Wetherall D, Anderson T, "A DoS-limiting network architecture", ACM SIGCOMM Computer Communication Review, 2005.

[2] Jamali S, Shaker G, "PSO-SFDD: Defense against SYN flooding DoS attacks by employing PSO algorithm", Computers & Mathematics with Applications, 2012.

[3] Habib A, Hefeeda M, Bhargava BK, "Detecting Service Violations and DoS Attacks", NDSS, 2003.

[4] Carl G, Kesidis G, Brooks RR, Rai S, “Denial-of-service attack-detection techniques”, IEEE Internet computing, 2006.

[5] Zunnurhain K, Vrbsky SV, Hasan R, “FAPA: flooding attack protection architecture in a cloud system”, International Journal of Cloud Computing, 2014.

[6] Bedi, H.S. and Shiva, S. 2012. Securing cloud infrastructure against co-resident DoS attacks using game theoretic defense mechanisms. In Proceedings of the International Conference on Advances in Computing, Communications and Informatics. ACM.

[7] Ristenpart, T., Tromer, E., Shacham, H. and Savage, S. 2009. Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In Proceedings of the 16th ACM conference on Computer and communications security. ACM.

[8] Long M, Wu CH, Hung JY, “Denial of service attacks on network-based control systems: impact and mitigation”, IEEE Transactions on Industrial Informatics, 2005.

[9] Wang Y, Lin C, Li QL, Fang Y, “A queueing analysis for the denial of service (DoS) attacks in computer networks”, Computer Networks, 2007.

[10] Mirkovic, J., Prier, G. and Reiher, P. 2002. Attacking DDoS at the source. In Network Protocols, 2002. Proceedings. 10th IEEE International Conference on. IEEE.

[11] Mirkovic J, Prier G, Reiher P, “Source-end DDoS defense”, Network Computing and Applications, 2003.

[12] Gil TM, Poletto M, “MULTOPS: A Data-Structure for Bandwidth Attack Detection”, USENIX Security Symposium, 2001.

[13] Morein, W.G., Stavrou, A., Cook, D.L., Keromytis, A.D., Misra, V. and Rubenstein, D. 2003. Using graphic turing tests to counter automated DDoS attacks against web servers. In Proceedings of the 10th ACM conference on Computer and communications security. ACM.

[14] Schwarzkopf R, Schmidt M, Strack C, Martin S, Freisleben B, “Increasing virtual machine security in cloud environments”, Journal of Cloud Computing: Advances, Systems and Applications, 2012.

[15] Livingston F, “Implementation of Breiman’s random forest machine learning algorithm”, ECE591Q Machine Learning Journal Paper, 2005.

[16] Chonka A, Xiang Y, Zhou W, Bonti A, “Cloud security defence to protect cloud computing against HTTP-DoS and XML-DoS attacks”, Journal of Network and Computer Applications, 2011.

[17] Wang H, Jin C, Shin KG, “Defense against spoofed IP traffic using hop-count filtering”, IEEE/ACM Transactions on Networking (ToN), 2007.

[18] Khan S, Traore I, “Queue-based analysis of DoS attacks”, Information Assurance Workshop, 2005.

[19] Pacini E, Mateos C, García Garino C, “Dynamic scheduling based on particle swarm optimization for cloud-based scientific experiments”, CLEI Electronic Journal, 2014.