www.ijsrnsc.org

Available online at www.ijsrnsc.org

IJSRNSC

Volume-1, Issue-2, June- 2013
Research Paper

Int. J. Sci. Res. in Network Security and Communication
ISSN: 2321-3256

Modelling and simulation of DDOS Attack using SimEvents

Abubakar Bala[*1] and Yahya Osais[2]

[*1]*Computer Engineering Department, King Fahd University of Petroleum and minerals, Dhahran, Saudi Arabia*

[2]*Computer Engineering Department, King Fahd University of Petroleum and minerals, Dhahran, Saudi Arabia*

Received: 10 May 2013 Revised: 15 May 2013 Accepted: 10 June 2013 Published: 30 June 2013

Abstract—At the beginning of the 21st century, a new form of denial of service (DoS) attack emerged which is the Distributed DoS (DDoS). This new form of attack was launched on huge number of prominent websites such as yahoo, e-bay, Amazon, and buy.com, resulting in tremendous financial loses. DDoS attack has continued to increase over the years due to the rapid increase in internet users. Moreover, even more alarming is the fact that developers of DDoS tools have assumed unprecedented sophistication in their design methods, thus making their attacks highly destructive and undetectable. As a result of these reason and many others, researchers have focused their attention on the study of this new method of attack, they are particularly interested in studying its evolution, and with this knowledge they are being able to design anti-DDoS tools in order to prevent networks from falling into the clutches of DDoS attack. In this research work, a DDoS attack is simulated using MATLAB's *SimEvents*, with the aim of finding the quantitative measure of its effect on the victim, experiments conducted in this study show that the server is scarcely utilized in its normal working condi-tions thus having high availability and low average utilization since it accepts requests only from legitimate clients. However, as the attacker launches an attack on the server, its utilization increases sharply and thus resulting in decrease in availability, this is because the server is flooded with illegal requests from the attacker as well as zombies from within the network domain. Additional study reveals that when a warm-up phase is added to the simulation of the server failure, the utilization suddenly increases due to the fact that the attacker seizes the opportunity of the slow recovery of the server to further overwhelm it and eventually push it into saturation.

Index Terms—Modelling, simulation, DDoS, DoS, SimEvents, MATLAB

I. INTRODUCTION

There have been widespread DoS/DDoS attacks recently, causing a lot of economical losses to organizations in addition to causing security issues, until now there is no panacea to protection against DDoS Attacks, most solutions provided for defense against DDoS has been to reduce its effect on web servers. DDoS has been defined as that which unlawfully re-duces or eliminates the availability of a service to a legitimate user [1]. DDoS attack is a method of attack by which the target system (victim) is overwhelmed with network traffic to the extent that it cannot respond to legitimate requests from users.

A. Components of DDoS attack

DDoS Attack is a very complicated process typically in-volving three system components, which are handlers, agents and the victim of the attack.

1) Handlers: As shown in figure 1 these are systems com-promised or hacked by the attacker on the network, he uses dubious methods to install DDoS attacking tools on these systems.
2) Agents: As in figure 1,the handlers then further forces clients (Zombie agents) to issue illegitimate requests to the target of the attack.
3) Victim: This is the real victim of the attack it is mostly a web-server critical to the network.

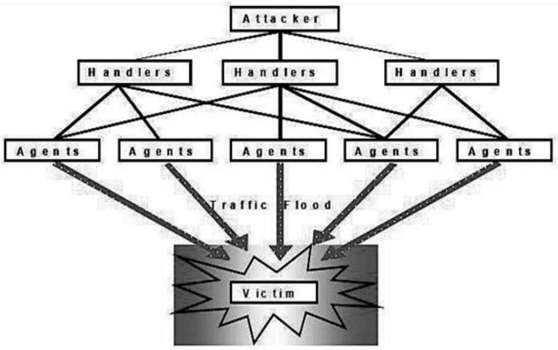


Fig. 1: Components of a DDoS Attack

B. Motivation

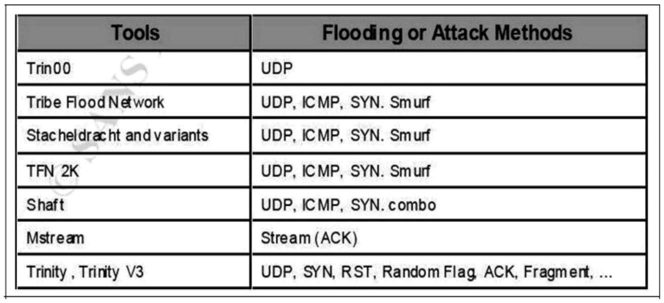
As a motivation, we provide typical kinds of DoS/DDoS attack methods: a summary is shown in figure 2.

1) Smurf Attack: In this scenario, the attacker sends a fake

Corresponding Author: *Abubakar Bala*

ICMP echo packet to the broadcast address of vulnerable networks, as a result, all the systems on the network reply to the victim with their ICMP echo replies. The aftermath of this is that it exhausts the bandwidth available to the target effectively preventing service to legitimate users.

2) TCP SYN Attack: This attack type makes use of the advantage of the weak spot of the TCP three-way handshake, the attacker issues a request aimed at the victim sever with packets with unreachable source address, because of this, the server is not being able to complete the connection request and as a consequence the victim server wastes its network resources causing an eventual shut down of the sever.
3) UDP Attack: In this scenario, the attacker sends a UDP packet to a random port on the victim system, as soon as the victim receives the UDP packet, it will attempt to determine which application is waiting on the destination port, once the victim realizes that there is no application waiting on the port, it will generate an ICMP packet of destination unreachable to the forged source address, the system eventually goes down provided that enough UDP packets are delivered to port on the victim.



Tools	Flooding or Attack Methods
Trin00	UDP
Tribe Flood Network	UDP, ICMP, SYN. Smurf
Stacheldracht and variants	UDP, ICMP, SYN. Smurf
TFN 2K	UDP, ICMP, SYN. Smurf
Shaft	UDP, ICMP, SYN. combo
Mstream	Stream (ACK)
Trinity , Trinity V3	UDP, SYN, RST, Random Flag, ACK, Fragment, ...

Fig. 2: Common DDoS attack methods

It should be noted however that any DDoS defense tool should be able to either prevent or at least detect the attack in its early stage. Thus knowledge of the behavior of the victim in its normal condition is essential, this is important because as soon as we detect that the system is deviating from its normal characteristics we can raise an alarm for possible DDoS attack- this is the motivation for this research work.

C. DDos tools

Some common DDoS tools include: Trinity which is a DDoS tool used to launch different types of flooding attacks on a victims site, communication from handler of the Attacker to the zombie agent is mostly achieved through internet relay chat (IRC). Other attack tools include Shaft, Tribe flood, Network 2k, Trinoo and Stacheldracht. The attacker usually exploits certain vulnerabilities or lapses in certain implementations of some protocol installed on victim server. In some cases the attacker applies brute force approach and issues a large amount of seemingly legitimate transactions to the victim to overwhelm it [1].

The remaining sections of this paper are outlined as fol-lows; section II ventures into works that are related to the one conducted in this research, section III outlines the methodology followed in the study. On the other hand, section IV elaborates on experiments conducted in order to validate the methodology presented. In section V experimental results are presented and discussed , a brief summary of the results is further presented in section VI, subsequently, section VII con-cludes the paper and also reveals certain areas the researchers will focus upon in their future work.

II. RELATED WORK

The authors of [2] employed the use of a software simulation tool the DDoSSim which has been developed for compre-hensive study of internet DDOS attacks, they reiterated that the DDoSSim enables one to deeply investigate different forms of attacks and protection schemes; the tool has the ability to provide useful recommendations on selecting best protection methods. They make use of the agent-based approach; fur-thermore, they conducted experiments for protection against DDoS attacks in order to demonstrate some potentials of the DDoSSim. Moreover, they considered the different phases of defense operations, which include the learning, decision making and protection. They further investigated into the adaptation of these protection methods to the actions of the attacker(s).

They suggest a common approach and simulation environment for finding adequate defense methods against DDoS attacks, Attack and defense methods they used include: the attacker which could be a Daemon or a Master, on the other hand, the defense agents are categorized into: initial information processing (sensor), secondary information processing (sampler), attack detection (detector), filtering (filter), and finally the investigation (investigator).

The simulation environment they employ is DDoSSim architecture which consists of the OMNeT++ framework, INET Framework, Multi-agent and DDoS framework.

Simulation experiments include a Learning mode: in which the authors pointed out is being used to create a model of generic traffic for the given network. The second mode is the Decision making and acting in which attack team is employed.

This study in [2] is different from one proposed in this work in that this research employs the use of MATLABs *SimEvents* rather than the OMNeT++. OMNeT++ is a more down-to-earth tool in the sense that it imitates a networking environment, in contrast, MATLAB's *SimEvents* is more of a general simulation tool, and is pretty easier to model than the OMNeT++. Another difference between the research conducted here and that of [2] Is that in this study attempt is made to only simulate a DDoS attack, this is contrast to authors of [2] Whom attempt to simulate a defense system as well.

Meanwhile the authors of [3] re-emphasize the fact that in order to fight DDoS attacks there is the need for fully understanding the theoretical basis upon which we can protect systems against such attacks, they propose an agent based framework for simulating and modeling DDoS attacks. Fur-thermore, they presented of a formal specification of a repre-sentative spectrum of DDoS attacks; finally they implement an agent based software tool that has the potential of simulating DDoS attacks and responses of victim systems.

The main aim of the experiments conducted in [3] is to check the ability of the Attack simulator to simulate different forms of attacks. Moreover, they reiterated that the purpose of the simulation based exploration of the Attacker tools is for firstly to check the network security policy at conceptual design stage and then secondly to check security policy of real life attacks.

The authors conducted experiments for various parameters of attack type specification and different victim configurations, also put into consideration is attackers intention as well as influence on input parameters such as; degree of protection given by the network and personal firewalls, victim of the attack, and the degree of the attackers knowledge of the network. Simulation results they obtained include parameters such as; number of terminal level attack options, percentage of attackers intention that are successful, percentage of effective network responses on attack actions, percentage of attack actions that were blocked by firewall, and percentage of ineffective results of attack actions.

The paper [3] is related to the work suggested here in that it attempts to presents a proposed paradigm for modelling and simulation of a broad range of DDoS attacks ,they built their simulator in Visual C++ 6.0, Java 2 version 1.3.1, KQML, and XML languages. In contrast to the work in this study, attempt is not being made to build a new simulator but use an existing promising simulation tool from MATLAB i.e. the *SimEvents*, this is the best practice however, because simulation of a new framework on a new simulator is not always a good practice in the industry, however the best practice is to run your paradigm or proposed framework on an existing standardized simulation tool, this is necessary, because then you can compare your results competitively with that of others.

On the other hand, authors of [4] attempt to propose a systematic method for DDoS attack detection. They base their detection on unusual behavior identification. Furthermore, they utilize energy distribution based on wavelets analysis to detect DDoS attacks, in addition, they mention that in attack free situations, the energy distribution will have limited variations while in attack situations, the traffic in the network will cause a significant energy distribution deviations in a short period of time. They performed experiments on typical internet traffic and results they obtained shows significant changes in energy distributions in DDoS attack situations. Moreover they suggest that this spike in energy distribution should be captured in the early stages of attack to prevent eventual congestion. They employed the use of Ns simulator, and results they obtain shows large differences in energy distribution in the traces with attack, as compared to traces without attacks, with a threshold of 0.01 their scheme is able to identify varieties of attack types.

The work in [4] is related to one in this study in that it also looks into detecting DDoS attacks at a very early stage of its occurrence; however it uses the Ns-tool, not *SimEvents* as proposed in this research work. Moreover, the paper uses energy distribution variations as a criterion for DDoS attack detections, in contrast, this work focuses collection of avail-abilities an utilization of end user devices in normal situations and if the availability of a user/server deviates from its normal characteristics alarm is raised for a possible DDoS attack.

Another method is that presented by authors of [5], they propose a new method for detection of intrusions in a network by employing the use of neural networks. They use the neural network to learn the behavior of each user, and then if this behavior deviates from its usual form, the system administrator is alerted for a possible security breach. Moreover, they employ the use of a back-propagation neural network called NIND (Neural network intrusion Detector), the network is trained in the identification task, and it is then tested on a network with 10 user systems, results they obtain shows a 96 percent accuracy in detection of abnormal activities with 7 percent false alarm rates.

The authors built the NIND system on a server that serves a total of 10 users. Data was collected on the system for 12 days. The features selected for the identification include 100 most common commands on the logs. The neural network chosen is the famous three-layer back propagation architecture, with input layer consisting of 100 units-representing the user vector, the hidden layer with 30 units, and the output layer 10 units, one for each of the 10 users. The network is implemented in the PlaNet Neural network simulator.

The work in [5] is related to one suggested here in that it provides a method for intrusion detection in a network involving 10-users, however, it does not focus on DDoS attacks as considered in this research, moreover, it employs the use of Neural network for the identification exercise which is not the case in this research work.

Whereas authors of [6] present a discrete event system (DES) based simulation of network systems for Quantitative security evaluation (QSE); they employ the use of MATLAB’s *SimEvents*. Initially the system is simulated in its normal state, and then an attacker (which is modeled as a client) with the collaboration of unwilling legitimate clients (the zombies) issues a DDoS attack on the system. The aftermath of this attack is that the target system can no longer respond to legitimate requests of clients. The authors compare the availability of the system in normal state and that of an attack situation, then based on this comparison, the effect of the DDoS attack on the network is visualized.

The work in this research is based on the network model presented by the authors of [6] this has become necessary because the authors of [6] did not provide a comprehensive de-tail of how the network model was implemented in *SimEvents*, thus making it extremely difficult for a novice in *SimEvents* to understand the implementation not to talk of reproducing the work in [6], In contrast this paper will attempt to reproduce the work of [6] While explaining in detail how the implementation was carried out in *SimEvents*.

Another important area the authors of [6] did not capture is the 'WARM-UP' phase of a server that just went through a repair phase, this is necessary because when there is a typical server failure (due to DDoS attack), after the repair or fixing the server, the server has to undergo a warm-up phase before it goes back to normal working condition. This work also considers an additional performance measure for DDoS attack which is the utilization of the target sever, this value has been found to sharply increase under attack situations.

III. METHODOLOGY

A. Normal working condition

As seen in figure 3, the normal state is captured by two Clients on the left end of the network titled Client 1 and Client 2 respectively; typically, these Clients issue service requests (modeled as entities) to servers 1 or 2 depending on some probability (randomness). These packets/entities are usually sent from Clients with destination address of either Server1 or 2, and based on this destination address, the router is able to route the packets to their intended destinations.

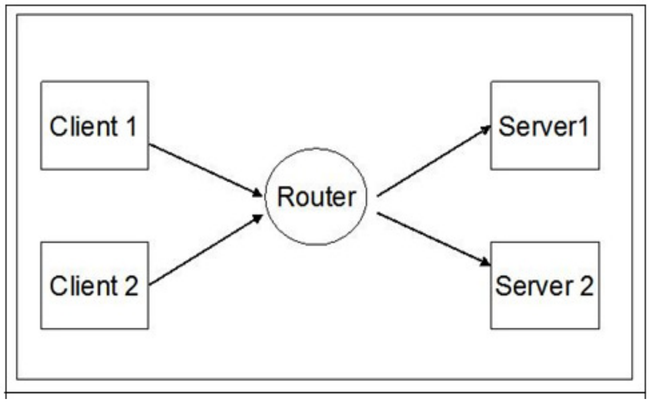


Fig. 3: Normal Working Condition

B. Under DDoS Attack

In contrast to the normal state, in the situation under DDoS attack, an additional illegitimate Client (Attacker) is seen as shown in figure 4. the Attacker, is the initiator of the DDoS attack, he makes use of cunning and deceitful means to install DDoS tools on legitimate Clients (in this case Clients 1 and 2), by so doing, he initiates them into 'zombies', and as a result, the attacker in collaboration with the newly recruited zombies issue DDoS attack on the target/victim, which is usually a server in this case Server1 as shown in figure 4, the consequence of this is that Sever1 now becomes overwhelmed and is not able to provide service to legitimate Clients.

The main aim of this research is to compare the Availability and utilization of Sever1 under normal and attack situations. Furthermore, comparison is made between the Availability and utilization when a Warm-Up phase is added after a repair phase and when it is not accounted for.

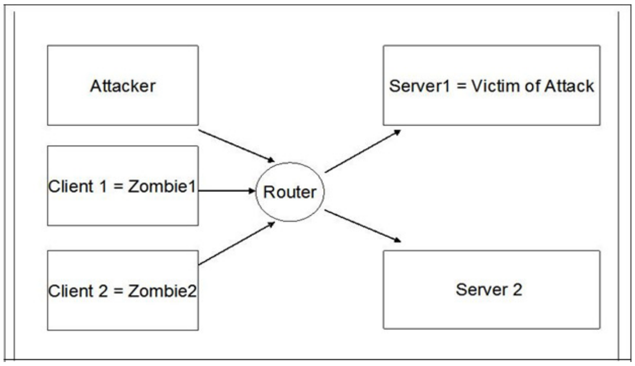


Fig. 4: Under DDoS Attack Condition

IV. EXPERIMENTS

A. Tools used

Tools used in this research work include: MATLAB R2012b Version 8.0.0.783 (released August 22 2012) which is a complete package consisting of modules like: *Simulink*, *SimEvents* and *StateFlow*. A PC with specifications: 2.40GHz processor Intel i3 and 3GB RAM is used as the work station.

Caveats: please note that, although these researchers have done their best in elaborating the experimental procedure carried out using *SimEvents*, however, basic knowledge of *SimEvents* might be required for full understanding of this section. The reader may refer to MATLAB documentation for *SimEvents* from [7]. Another important point is that the blocks with random number generators are required to have different values for seed, the recommendation is to use a unique 5 digit odd number as seed for different random blocks within the model.

B. Normal Condition

As seen in figure 5 each of the Clients is represented by 2 Event-Based Random Number Generators, a Time-Based Entity Generator, a Set Attribute Block and a FIFO (First in first out) queue.

- Block parameter explanation
- Here the authors present some functionality and configurations of main blocks in figure 5.

1) *Event-Based Random Numbers A1/A2:* As illustrated in figure 5 each of these block is responsible for attaching a random destination to each packet/entity produced. The destination can be either Servers 1 or 2; the probability distribution used is the arbitrary discrete distribution, which generates a vector value of 0 or 1 with equal probabilities.

2) *Event-Based Random Numbers B1/B2:* Each one of

these blocks is responsible for generating random service time (length) for each generated packet. This service time is then used as the service time for single Servers 1 and 2 respectively. Each of B1/B2 is produced by a uniform distribution with minimum and maximum values of 6 and 10 respectively.

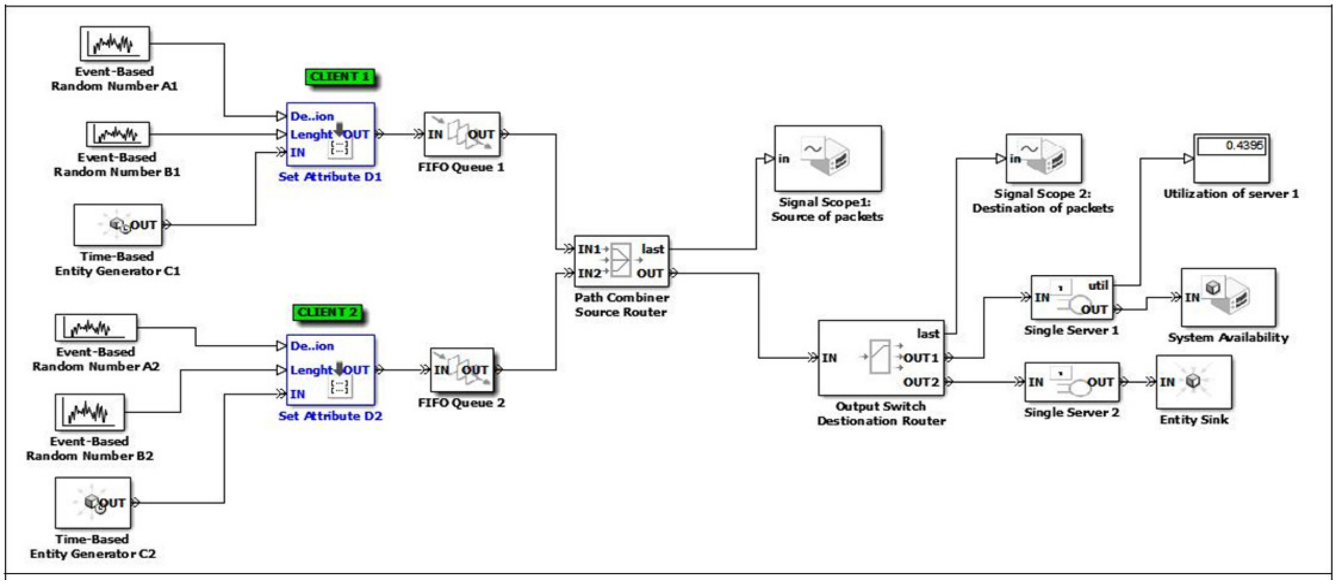


Fig. 5: over-all set-up for Normal condition

3) *Time-Based Entity Generators C1/C2:* On the other hand, each one of this block is responsible for generating ran-dom entities/packets from Clients 1 and 2 respectively, entities are produced from this block with an intergeneration time from an exponential distribution. For Client 1 the distribution has a mean of 100, while for Client 2 the mean is set as 10.

4) *The Set Attribute D1/D2:* These blocks are responsible for attaching attributes to packets emanating from each Client. The properties attached to each packet are:

1) Source: This indicates the source of the packet either from Client 1 or 2; this property is set as ’1’ for Client 1 and ’2’ for Client 2.
2) Destination: This shows the destination of the packets; the attribute is gotten from signal port of connected to the Event-Based Random Number Generators A1/A2.
3) Length: This indicates the service time of each packet; its value is gotten from signal port connected to Event Based Random Number Generators B1/B2.

5) *FIFO Queues:* The FIFO queues are set to a fix capacity of 25 each.
6) *path combiner/source Router:* This is responsible for selecting with equal probability packets from either Client 1 or Client 2 for routing to their assigned destinations.
7) *Output Switch Destination Router:* This block is responsible for routing the packets to their final destination. This block reads its switching criterion from the Destination Attribute of the set Attribute Blocks D1/D2 and based on this destination attribute it routes packets appropriately.
8) *Signal Scopes:* Signal scopes 1 and 2 are used to show graph-plots of the source and destination of the packets respectively. In addition, last scope is used to show plots of the Availability of Server1.
9) *Display:* The display at the extreme right displays the Average utilization of Sever 1 as a performance measure.

C. *Under DDoS Attack*

As seen in figure 4 an attacker intrudes into the network and issues a DDoS attack, with the aim of overwhelming its target (i.e. Server1) and stop it from replying requests of legitimate Clients, the attacker does not do this alone he forces Clients 1 and 2 to collaborate with him thus turning them into ’zombies’.

- This section explains the methodology for simulating the DDoS attack, the section is divided into two portions:
- That which does not consider a Warm-Up phase when an attack occurs named: Attack(No-Warm-Up).
- The second part is that in which a Server Warm-Up Phase is considered after failure titled Attack(Warm-Up).

1) Attack(No-Warm-Up: In this experiment, after the repair of the victim server, a Warm-Up is not considered, as such the attacked server is assumed to instantaneously spring back into action immediately after repair. The set-Up is shown in 6 an attacker is intrudes into the network; he sends packets or entities representing attacks to Server1, the TimeBased Entity Generator C1 is used to generate attacks with an exponential distribution with a mean of 50.

The service time of each attack is set from the length A1 block and generated from a uniform distribution with minimum and maximum values of 6 and 10 respectively. In addition, the destination of the packet is set as 1 (meaning Server1) from dialog in the Set Attribute Attacker block. The configurations of Client 1 (now zombie 1) and Client 2 (now zombie 2) are the same as they were in the case of normal situation above. The lower section of 6 (Entities representing security failure block) is used to model the attack situation or rather the failure of the target Server, entities produced by the Time-Based Entity Generator C4 represents failure of the server and not packets.

How the attack occurs:

As shown in figure 7 the server is initially in the 'Down' state, but because the initial value 1 Block is set to '1', Enabled Gates 1, 2 and 3 are enabled and legitimate packets from Client 1 and 2 as well as that of the attacker are routed to their various destinations, with the attacker targeting only Server1 as destination. However, Enabled Gates 4, 5 and 6 are initially disabled by the block Initial value 2.

Immediately a failure entity generated by Time-Based Entity Generator C4, enters the repair server, the Signal-Based Func-tion call Generator named FIRST causes the Chart to change state from 'Down' to 'Up', as a result the server up signal becomes 1, and the server down signal becomes 0, this again causes the Enabled Gates 1, 2 and 3 to remain activated as before, while Enable gates 4, 5 and 6 are disabled as before.

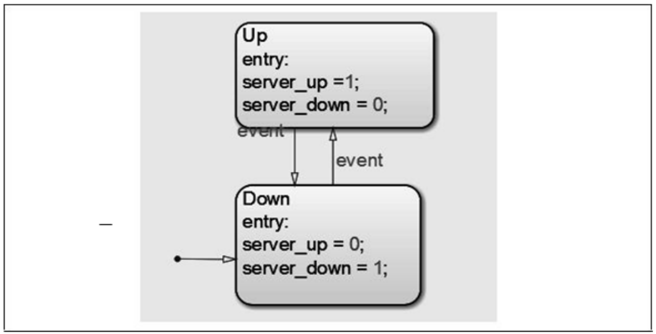


Fig. 7: State diagram for Attack (No-Warm-Up

As soon as the entity leaves the repair sever, the Chart makes transition back to the 'Down' state and in this case the sever up signal switches to 0 while server down signal is now 1, this causes the Enable Gates 1,2 and 3 to be disabled, and Enable gates 4,5, and 6 become enabled. As a consequence, packets move from the OUT 2 port of Replicates 1, 2 and 3 blocks via Enabled Gates 4, 5 and 6 to the Set Attribute Attacking DDoS 1, 2 and 3 blocks. These set attribute blocks are responsible for re-directing all packets to Server1, what they essentially do is they change the destination of all packets to Server1, thus mimicking the real situation where the Attacker forces the zombie agents to send illegal service requests to its target victim with the sole aim of flooding the server and eventually preventing it from accepting legitimate request from Clients. As soon as another entity enters the repair server again the Chart changes to Up state and the cycle is repeated.

The Path Combiner blocks 1, 2 and 3 are responsible selecting packets either from Attacker and legal requests

from Clients 1 and 2 or the redirected packets from Set Attribute DDoS attacking blocks 1,2 and 3, this decision is based on the state of the Chart (i.e. Either Down or Up state). Signal scopes are used to display source of packets, their destinations and the availability of the target of the attack (i.e. sever 1). Moreover, a display is used to display the average utilization of Server1.

2) Attack(Warm-Up): In contrast to the procedure in Attack(No-Warm-Up), this set Up considers the fact that when a server fails (due to an attack),after repair there is usually another phase called the Warm-Up phase the server goes through before it springs back to normal operating condition. The set-Up in figure 8 captures this stage. Notice the difference between figure 8 and figure 6.

In addition, the Chart now has a third state in between the 'Up' and 'Down' phases named the Warm up state as shown in figure9, thus the sever has to go through a Warm-Up phase before it goes to the back to its optimal working condition.

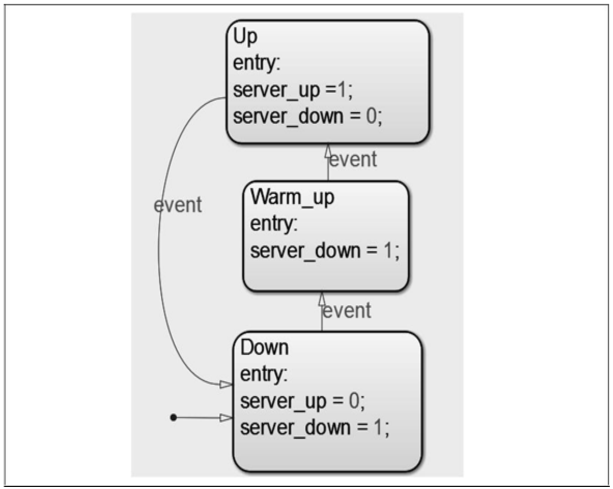


Fig. 9: State diagram for Attack(Warm-Up)

In this case as soon as the entity/failure enters the repair server, the Function call Generator named FIRST causes the Chart which is initially in Down state to move to the Warm-Up state, this Warm-up phase inherits the characteristics of Down state-where the sever-down signal is 1 and server up signal is 0, this causes Enabled Gates 4, 5 and 6 to be enabled, and Enabled Gates 1, 2 and 3 to be disabled. As a result, all packets are redirected to Server1 by the Set Attribute DDoS Attacking blocks 1, 2 and 3.

As soon as the entity leaves the repair server and enters the Warming Up server, the Chart changes state to the Up state, thus disabling Enabled Gates 4, 5 and 6, and at the same time activating the Enabled Gate blocks 1, 2 and 3, in this case Path Combiner blocks receive their inputs via IN1 port and Clients 1 and 2 are allowed to send legitimate requests to the servers. Subsequently, when the failure entity leaves the Warming Up server for the sink, the Chart changes state back to the 'Down' state and the cycle is repeated again.

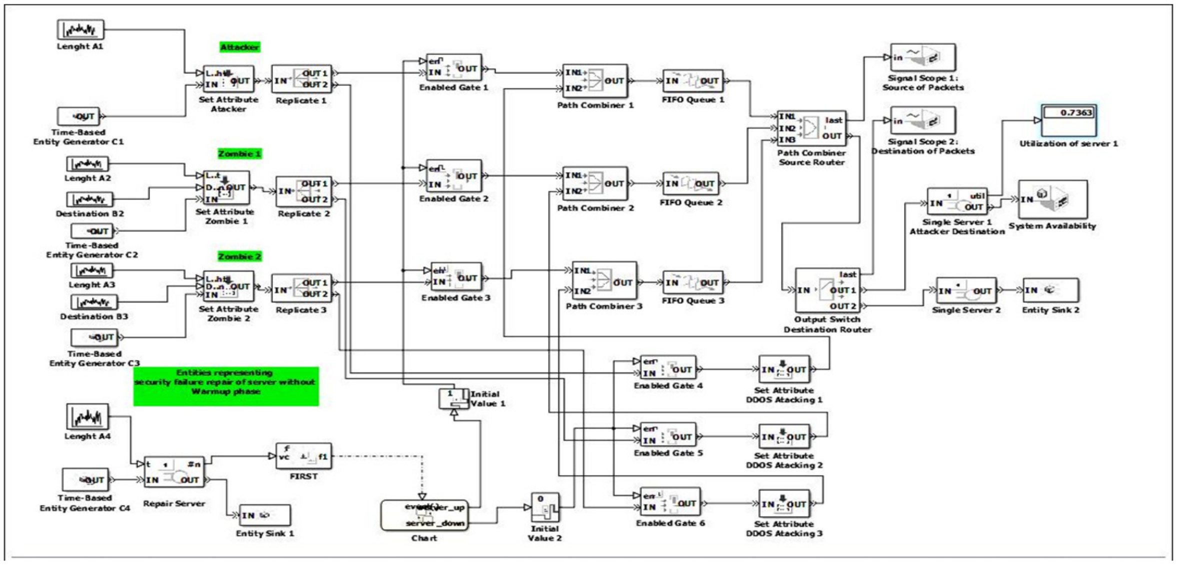


Fig. 6: over-all set-up for Attack situation with no warm-up phase

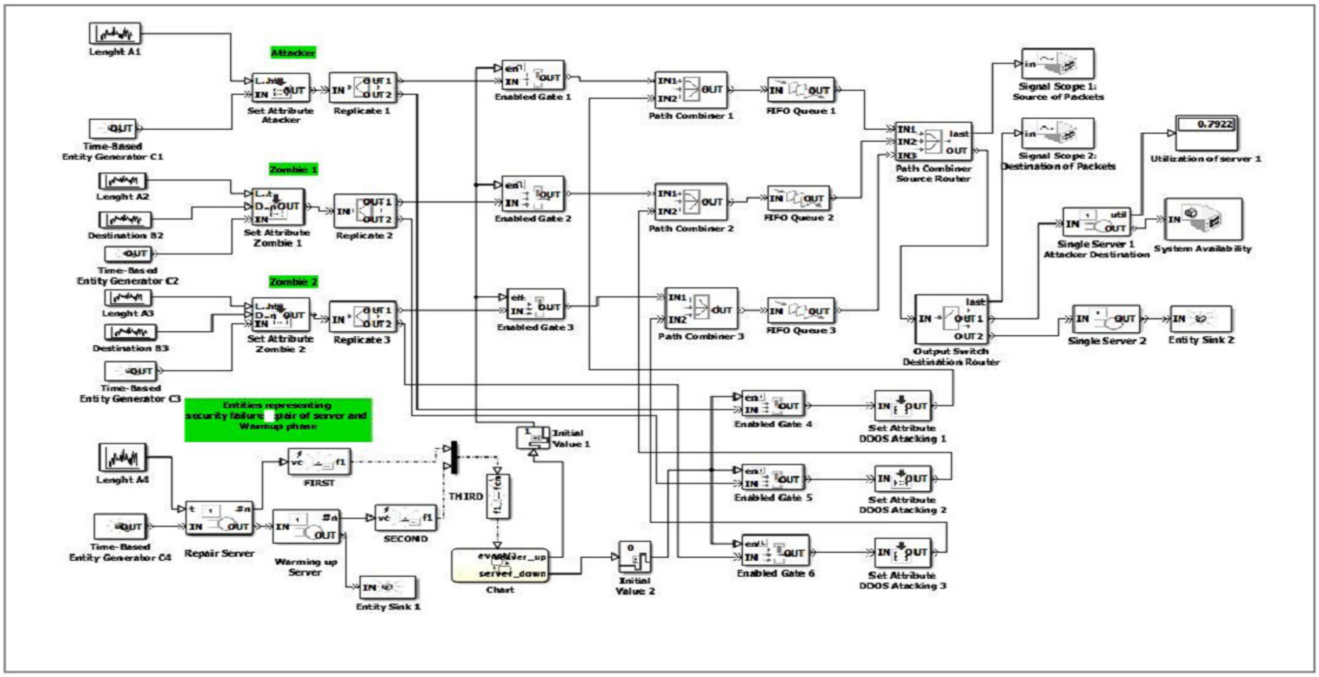


Fig. 8: over-all set-up for Attack situation with warm-up phase

V. EXPERIMENTAL RESULTS

Each of the set-ups above were run for 1000 simulation time and for each set-up, performance measures collected include: source of packets, destination of packets, Availability as well as utilization of Server1.

A. Normal condition

Figure 10 Shows the graph for the Source of packets it can be seen that, most of the packets are emanating from Client 2 this is due to the fact that the entity generator of Client 2 generates entity at a faster rate than that of Client 1. Figure 11 shows a plot for the destination of packets as seen in the figure destination is almost evenly distributed between Server1 and 2 this is due to the fact that destination attribute of Clients 1 and 2 are generated from the same distribution and with the same properties.

Figure 12 on the other hand shows the availability of Server1 as can be inferred from the figure the white space between the lines shows time zones in which the server is not being used (meaning it is Available) e.g. for time 20 to 40 the server is available. On the other hand, average utilization of the Server1 is gotten as 0.4395 from the display.

B. Attack (No-Warm-Up)

As seen in figure 13 most of the packets are generated from Client 2 (zombie 2) for similar reasons given above.

As seen in figure 14 Plots for destination, the destination of packets are mostly Server1; this is due to the fact that Server1 (victim) is overwhelmed by the attacker - thus making it busy answering illegitimate requests.

As seen in the figure 15 for the availability of Server1 have decreased from its previous value in the Normal state as the white space between the lines has decreased as compared to that in the Normal case. However, the average utilization of Server1 also sharply increases to 0.7363. This shows that the server has been busy answering illegitimate requests due to the DDoS attack.

C. Attack(Warm-Up)

As in figure 16 we can see that most of the entities are being generated from Client2 for similar reasons captured above.

From figure 17 it can be seen that most of the packets have a destination address of Server1 this is obviously due to the DDoS attack launched on the network, thus Server1 is saturated.

As seen in figure 18 the availability of sever 1 has dropped just as in the case of Attack (No-Warm-Up) above, but in contrast, the utilization of the server is now 0.7922 as seen from figure 18 this is due to the added warm-up phase that is introduced.

VI. SUMMARY OF RESULTS

As can be seen from the summary graph in figure 19, Server1's Average utilization is 0.4395 in Normal condition but it escalates to 0.7363 under the Attack (No-Warm-Up) situation, and when a warm-up state is taking into consider-ation as in Attack(Warm-Up), the utilization further increases to 0.7922. What can be inferred from this is that, under attack situation, the server is extremely burdened with illegitimate request coming from both the attacker and zombies hence the increase in utilization. Subsequently, when a Warm-Up stage is considered, the attacker takes this opportunity to further flood the server hence the increase in utilization. The overall effect of all these is that the server eventually reaches a very high utilization and is not being able to accept legitimate requests from Clients.

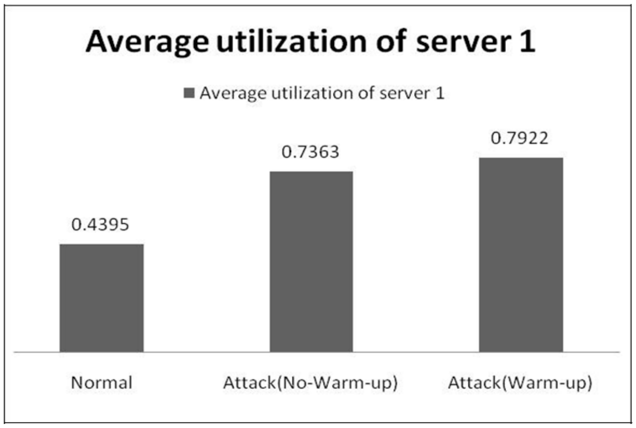


Fig. 9: Summary of Results

VII. CONCLUSION/FUTURE WORK

From experimental results obtained from section V, it can be concluded that the utilization and availability are good performance measures for a server under a DDoS attack, all victim servers of DDoS attacks usually have a very low availability when compared to their values under Normal condition, however their utilization increase sharply when they are under attack. This information can be very beneficial to the designers of DDoS defense tools, as they can make their tools act immediately they sense changes in these performance measures. It can be further inferred that the *SimEvents* of MATLAB is able to simulate a DDoS attack.

As a further work, this researchers will like to look into other complex networking scenarios involving not only zom-bies but handlers, this researchers will also like to use other simulation packages available to compare or calibrate the results obtained in this study, another area this researchers will look into is the possibility of using other performance measures that can be collected for quantitative assessment of DDoS attacks. Another interesting area of research is to sim-ulate different forms of DDoS attacks such as Smurf Attack, TCP SYN attack and UDP Attack in other to investigate their various intensities on their victims, and eventually find out the ones that causes more havoc to their victims.

ACKNOWLEDGEMENT

This author would like to thank King Fahd University of Petroleum and minerals Saudi Arabia (KFUPM) for relentless support it gives to Academic activities within and outside the Kingdom.

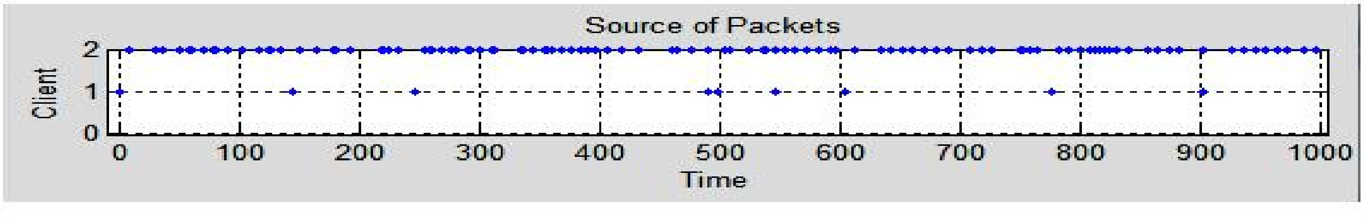


Fig. 10: Source of Packets for Normal condition

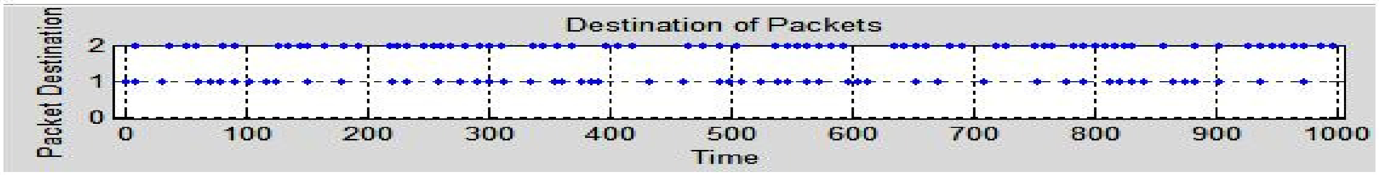


Fig. 11: Destination of Packets for Normal condition

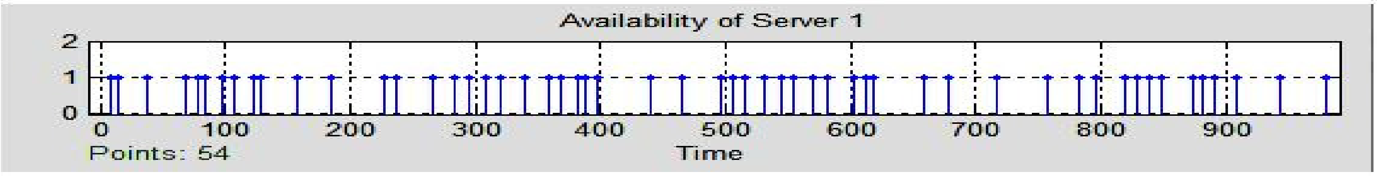


Fig. 12: Availability of Server 1 for Normal condition

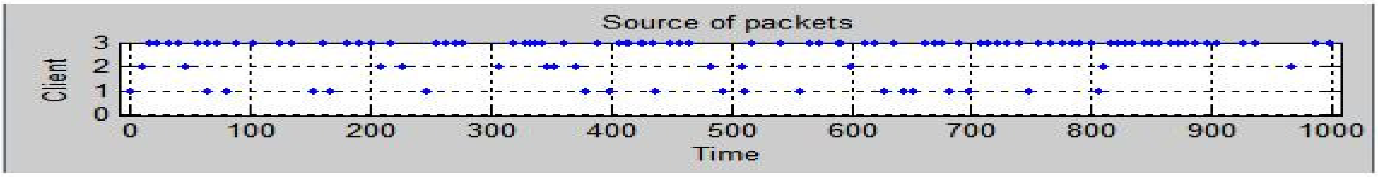


Fig. 13: Source of Packets for Attack (No-Warm-Up)

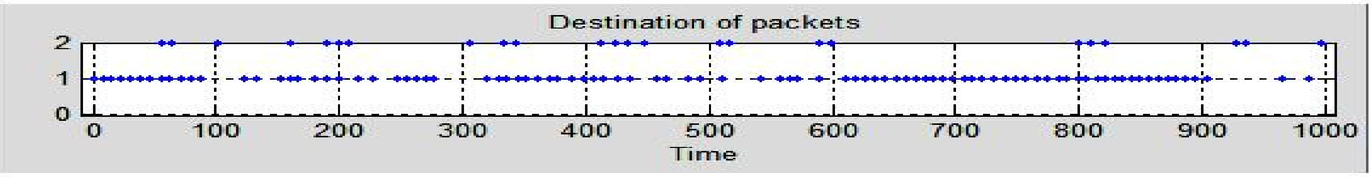


Fig. 14: Destination of Packets for Attack (No-Warm-Up)

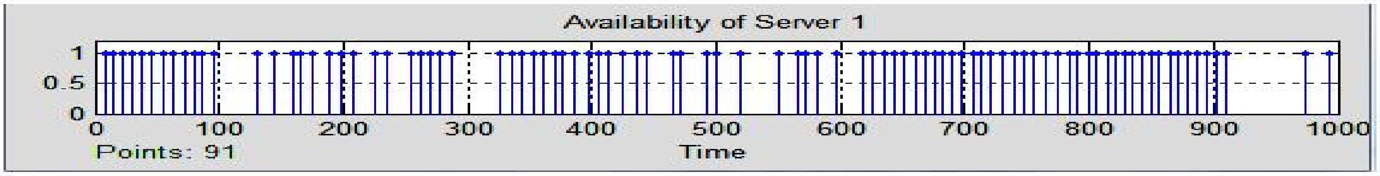


Fig. 15: Availability of Server 1 for Attack (No-Warm-Up)

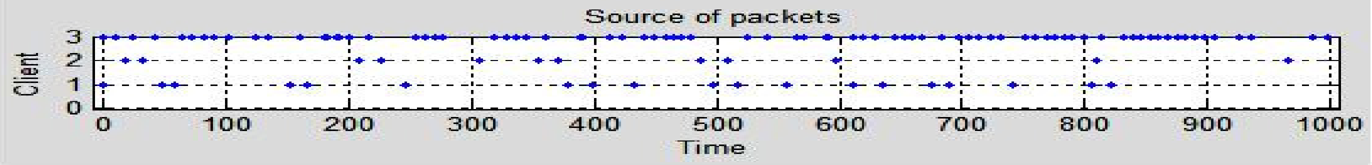


Fig. 16: Source of Packets for Attack(Warm-Up)

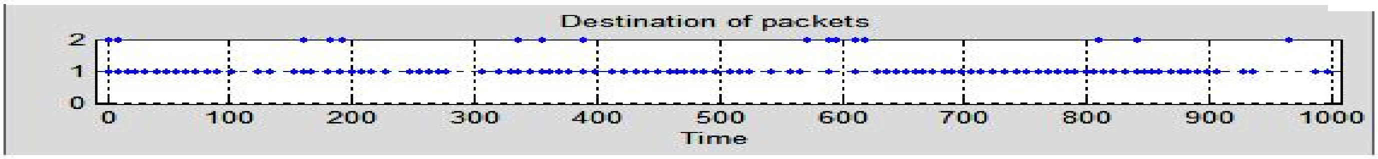


Fig. 17: Destination of Packets for Attack(Warm-Up)

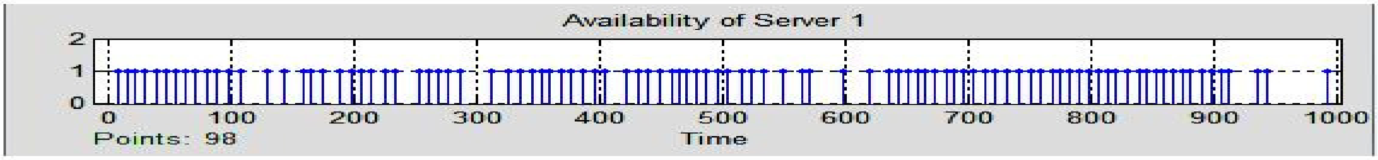


Fig. 18: Availability of Server 1 for Attack(Warm-Up)

References

[1] A. Noureldien, “Protecting web servers from dos/ddos flooding attacks. a technical overview,” in *International Conference on Web-Management for International Organisations. Proceedings. Geneva*, 2002.

[2] I. Kotenko and A. Ulanov, “Simulation of internet ddos attacks and defense,” in *Information Security*, pp. 327–342, Springer, 2006.

[3] I. Kotenko, A. Alexeev, and E. Man’kov, “Formal framework for model-ing and simulation of ddos attacks based on teamwork of hackers-agents,” in *Intelligent Agent Technology, 2003. IAT 2003. IEEE/WIC International Conference on*, pp. 507–510, IEEE, 2003.

[4] L. Li and G. Lee, “Ddos attack detection and wavelets,” *Telecommunica-tion Systems*, vol. 28, no. 3-4, pp. 435–451, 2005.

[5] J. Ryan, M.-J. Lin, and R. Miikkulainen, “Intrusion detection with neural networks,” in *Advances in neural information processing systems*, pp. 943–949, MORGAN KAUFMANN PUBLISHERS, 1998.

[6] G. Khazan and M. A. Azgomi, “A distributed attack simulation for quan-titative security evaluation using *SimEvents*,” in *Computer Systems and Applications, 2009. AICCSA 2009. IEEE/ACS International Conference on*, pp. 382–385, IEEE, 2009.

[7] Mathworks,“Simevents,http://www.mathworks.com/help /simevents/release-notes.html.”