A comprehensive study of distributed Denial-of-Service attack with the detection techniques

H. H. Ibrahim[1], A. E. Hamzah[2], H. A. Saeed[3], H. H. Qasim[4], O. S. Hamed[5], Hussein Yahya Alkhalaf[6], M. I. Hamza[7]
[1,2,3,4]Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, Malaysia
[1,2,5,6]Department of Engineering and Built Environment, Universiti Kebangsaan Malaysia, Malaysia
[7]Department of Computer Engineering, University of Basrah, Iraq

Article Info

Article history:

Received Oct 21, 2019
Revised Feb 2, 2020
Accepted Feb 10, 2020

Keywords:

Denial of service
Distributed Denial-of-Service
Information security
Intrusion detection
Simulation

ABSTRACT

With the dramatic evolution in networks nowadays, an equivalent growth of challenges has been depicted toward implementing and deployment of such networks. One of the serious challenges is the security where wide range of attacks would threat these networks. Denial-of-Service (DoS) is one of the common attacks that targets several types of networks in which a huge amount of information is being flooded into a specific server for the purpose of turning of such server. Many research studies have examined the simulation of networks in order to observe the behavior of DoS. However, the variety of its types hinders the process of configuring the DoS attacks. In particular, the Distributed DoS (DDoS) is considered to be the most challenging threat to various networks. Hence, this paper aims to accommodate a comprehensive simulation in order to figure out and detect DDoS attacks. Using the well-known simulator technique of NS-2, the experiments showed that different types of DDoS have been characterized, examined and detected. This implies the efficacy of the comprehensive simulation proposed by this study.

Corresponding Author:

H. H. Ibrahim,
Faculty of Electrical and Electronic Engineering,
Universiti Tun Hussein Onn Malaysia,
86400 Parit Raja, Johor, Malaysia.
Email: hussampc93@gmail.com

1. INTRODUCTION

Distributed denial of service (DDoS) is one of the common attacks within wide range of networks where the recognition and prevention of such attack has always been a hot issue in network security research [1-4]. DDoS detection and defence systems have many shortcomings such as high false positive rate, low execution efficiency, and lack of linkage between detection and defence [5-7]. Therefore, eliminating false positives, improving execution efficiency, and enhancing the linkage between detection and defence processes have always been the focuses of research [8-12].

With the diversity and different characteristics of DoS, the process of detecting such attack is still facing obstacles [13-16]. Şimşek & Şentürk [17] have proposed method that utilize the pre-congestion in order to analyze the flow of data during this period. The authors had an assumption that low-rate distributed DoS is one of the hardest to be detected due to their similarity to the normal behavior. Therefore, the authors have focused on the periods have no congestions in order to diagonis the features. The features extracted from such periods have been incorporated to form a new filtering approach for detecting DDoS attacks. Results of simulation showed fair progress on characterizing DDoS attacks.

Bukharov et al. [18] have proposed a game-based method for simulating DoS attacks. The proposed method has utilized a scenario where the intruder would be attracted in order to gain information regarding his real intentions. Results of simulation showed that the proposed method has the ability to detect wide range of DoS attacks. Wang et al. [19] have proposed a DoS detection method based on honeynet technology. The proposed method was intended to observe and analyze the characteristics of every behavior in order to detect specific pattern. Finally, the proposed method aimed at detecting such patterns which might correspond to DoS attacks. Results of simulation showed progress on detecting DoS attacks. Mohd et al. [20] have examined the distributed DoS that might occur on Internet of Things (IoT) networks. The authors have utilized OMNET++ in order to create a virtual environment that simulate the IoT networks. During such simulation, the authors have characterized several DDoS attacks. As depicted from the literature, it is obvious that the examination of DoS attacks is still a challenging task where wide range of such attack would be encountered especially with the variety of networks nowadays. Therefore, this paper aims to accommodate a comprehensive simulation to examine the types of DoS, as well as, attempting to detect these attacks.

2. RESEARCH METHOD

One of the most serious problems is DDoS, and many defenses have been proposed to address this threat. In order to compare and evaluate these solutions, a common evaluation platform is required. The methodology of this paper consists of three parts: a typical attack scenario consisting of the dimensions of legitimate traffic and target network resources, testing the methodological criteria that capture performance metrics and are affected by the effectiveness of attacks and defenses it is composed. In order to do so, the following steps have been applied:

- Detect and filter one-way legitimate traffic from traffic identified as a possible attack.
- Detect attack using multiple detection criteria.
- It is legitimate from attack traffic. Finally,
- Attack samples from attack traffic, summarize attack functions in a readable format and machine-readable form, and facilitate the application of clustering methods.

This makes it easy to collect attack samples from many public traces. All of these pastes are automated by a series of tools. Figure 1 shows applying the simulation process to attack cases requiring more attackers and usage scenarios. In our simulation methodology we follow these steps:

- First Step: In first step is to create a network topology with an NS-2 tell script for each attack.
- Second Step: In second step is to attach the legitimate traffic records to perform legitimate traffic on topology nodes. After that, real-time attack tracks are linked to topologies to generate attack traffic. These attack records are analyzed.

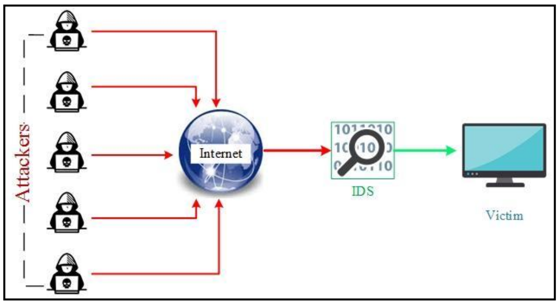

Figure 1. Framework of the proposed method

Then simulation is again performed, all traffic is monitored, and an offline analysis is performed. The output trace file is then used to measure the attack. The simulation topology used for this experiment contains a legitimate client pool containing various nodes that are used to generate legitimate traffic. To generate legitimate traffic, real-time tracks are used. With these traces, the nodes generate TCP traffic. An attacker used UDP traffic to launch an attack.

The purpose of the attack is to consume the bandwidth of the bottleneck link so that legitimate traffic could not send the packets. Each simulation time is 2 seconds. Legitimate traffic is based on TCP, so it goes through the slow boot phase. The total number of legitimate clients in the legitimate client pool is 8. The total traffic load and bottleneck bandwidth represent the scenario of a busy connection.

In our experiments, legitimate traffic is generated using real time tracks. The legitimate traffic is based on TCP. Here we have considered 13 legitimate clients that want to communicate with the TCP Sink node. Real-time data sets are again used to generate DDoS attacks. The amount and complexity of traffic in records is very high and very difficult to understand. The tracks used to create an attack are stored in tr format. Some results are simulated by gnuplot and other extracted information and then passed to excel to produce the graphical results.

2.1. Simulation

The simulator used in this paper was the NS2 simulator. Network Simulator Version 2, widely known as NS2, is an event driven simulation tool that is useful in studying the dynamic nature of communication networks [21]. The cost of building a real distributed testing environment is very high. Simulation is an important method in network research, as simulation can be used to analyze network related problems under different protocols, cross traffic and topologies with much less cost [22-24]. The most well-known network simulator is NS2. NS2 simulator covers a large number of applications, protocols, network types, network elements and traffic models. Therefore, we use NS2 simulator for this thesis. Simulation of wired as well as wireless network functions and protocols (e.g., routing algorithms, TCP, UDP) can be done using NS2. In general, NS2 provides users with a way of specifying such network protocols and simulating their corresponding behaviors. Due to its flexibility and modular nature, NS2 has gained constant popularity in the networking research community.

At the simulation level, NS-2 uses the OTcl (Object-Oriented Tool Command Language) programming language to interpret user simulation scripts [25]. The OTcl language is actually an object-oriented extension of the Tcl language. At the top level, NS is the interpreter for the user's Tcl script. Tcl language is fully compatible with the C ++ programming language.

NS creates two main analysis reports simultaneously and also explains the OTcl script. One of them is the Network Animator (NAM) object, which shows simulated visual animations. The other is a tracking object that consists of the behavior of all objects in the simulation. NS projects are usually shipped with various software packages (ns, nam, tcl, otcl, etc.) and are referred to as an "all-in-one package," but they can also be searched and downloaded separately. This study used a stable version of the ns 2.15 ns all-in-one package and installed it in the Red Hat Enterprise Linux 5 operating environment. This working ".tcl" file was written and parsed with a text editor. "tr" file. Figure 2 shows the flowchart of the simulation.

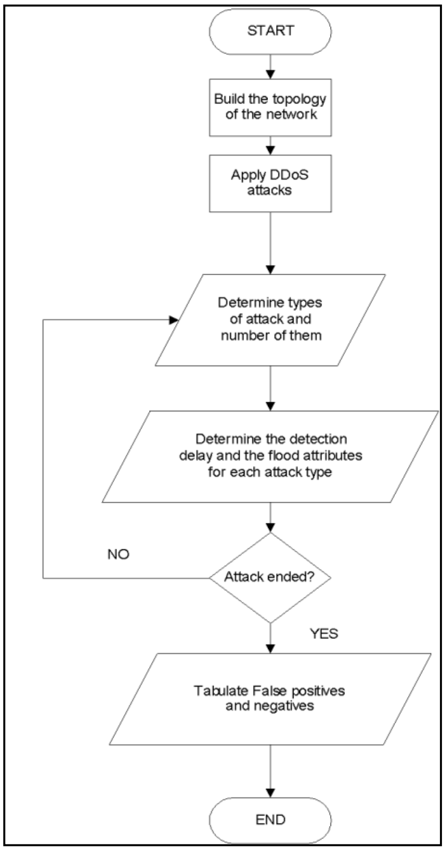
START
Build the topology of the network
Apply DDoS attacks
Determine types of attack and number of them
Determine the detection delay and the flood attributes for each attack type
NO
Attack ended?
YES
Tabulate False positives and negatives
END

Figure 2. Flowchart of the simulation

2.2. Simulation models

Models illustrate the movements of nodes and the connection between models in LAN and WLAN within the space of simulation. In WLAN the foremost manner used for simulation is Random Waypoint quality model. During this model the nodes passage from waypoint to subsequent with a haphazardly chosen speed (uniformly distributed between 0–20 m/s). A selected speed and period is chosen for each transition. When the stipulated transition period ends the node might pause for a selected period of your time before beginning its transition towards subsequent waypoint. Nodes within the simulation discovered move consistent with a model that's acknowledge because the “random waypoint” model selects an oblong field. Quality models were created for the simulations mistreatment thirty-one nodes, most speed of twenty m/s, topology boundary of a thousand × a thousand and simulation time of fifty sec.

The instructions of experimenting the simulation can be explained as follows:

- For analyzing given scenarios then write down TCL script then simulate by ns2.
- Then the traces file and name file which is created during executing TCL scripts for each single scenario.
- Create a final procedure.
- Create nodes which will be present the specific topology. Here in experiments each scenario has numbers of nodes.
- Create the way of connection between nodes waypoints to represents wireless connection or links to connect the nodes in LAN.
- Set up the LAN by specifying the nodes, and assign values for bandwidth, delay, queue type and channel to it.
- Specific the protocols to sending message or pockets such as TCP and/or UDP connection(s) and the FTP/CBR.
- Schedule the different events like simulation start and stop, data transmission starts and stop.
- Call the finish procedure and mention the time at what time your simulation will end.
- Execute the script with ns.

Tables 1 and 2 show both hyper-parameters and parameters of the simulation respectively.

Table 1. Simulation’s hyper-parameters

Parameter	Specification
Interface	Wireless
Packet Size	512 Byte
Queue Length	50
No. of Nodes	31
Simulation Area	1000x1000
Simulation Time	50 Second
Mobility Model	Random Waypoint
Transmission Range	250m
Traffic Model	CBR
Bandwidth	2 mbps

Table 2. Simulation’s parameters

Parameters	Value
Types of attacks	TCP traffic with random sequence numbers TCP flood TCP SYN flood ICMP flood Spoofing Invalid protocol number
Attackers	6 attackers node
Legitimate	1 legitimate node
Victim	1 victim node
Types of attacking	CPU-extensive attacks
Length and duration	custom packet length and duration

3. RESULTS AND ANALYSIS

To analyse the performance of the NS-2 simulator, there will be five cases of simulation that had been done. The first simulation is done in TCP traffic, second simulation is done in TCP flood, third simulation is done in TCP SYN flood, fourth simulation is done in ICMP flood and fifth simulation done in spoofing.

3.1. Simulation result of TCP traffic

A hierarchical design enforced to make wireless situation and local area network situation. This design includes a root node and 3 to four clusters sub networks it is depends on the attack as are shown within the next section. Every cluster includes mobile nodes. The hierarchical design is additionally non-public addressing theme. A pool of personal address is getting used for distribution private address to every node in each cluster. At the beginning of simulation every time associate address is being picked up from address pool and assign to the present node. When, a cluster node desires to speak with alternative node that resides in alternative cluster, all the traffic flows from root node. Within cluster node will communicate directly while not forwarding traffic to entryway.

Hierarchical design is getting used in implementation as a result of aggressor node desires to attack a node that flow most traffic of the network. During this situation, most network traffic flows from root node. As it can be seen in Figure 3 after the run simulation for DDoS attack in 222.0 MS the attack starts to send pockets from nodes to yellow node which represents the victim computer or server. After a while cause of the huge numbers of pockets sends the server will be stopped or killed and cannot receive any requests from any computer. As shown in the graph in Figure 4. As Topology concern, there are only two data connection between clusters for experiment purpose, one from cluster 15 to 16 and another with 6, 7, 8, to 9 as shown in Figure 5.

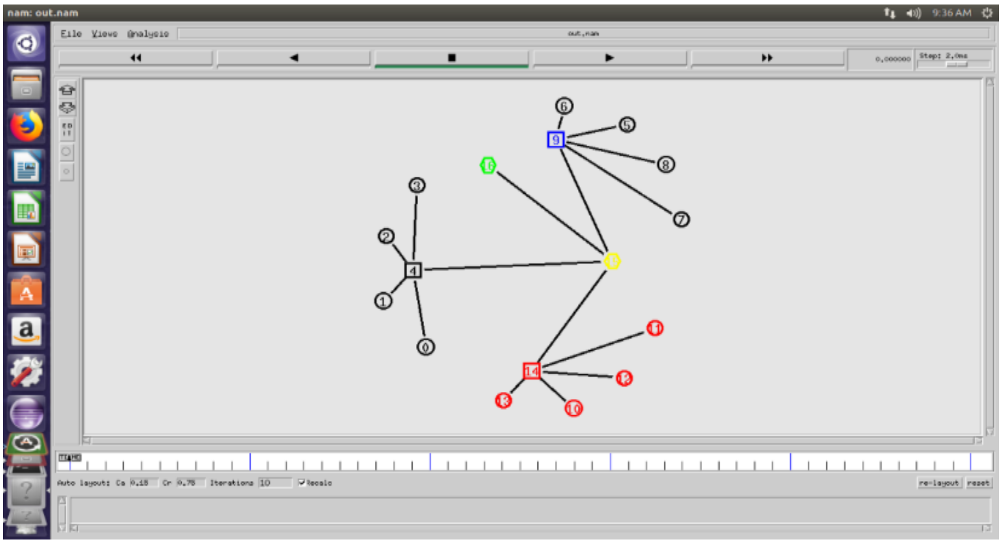

Figure 3. Network topology

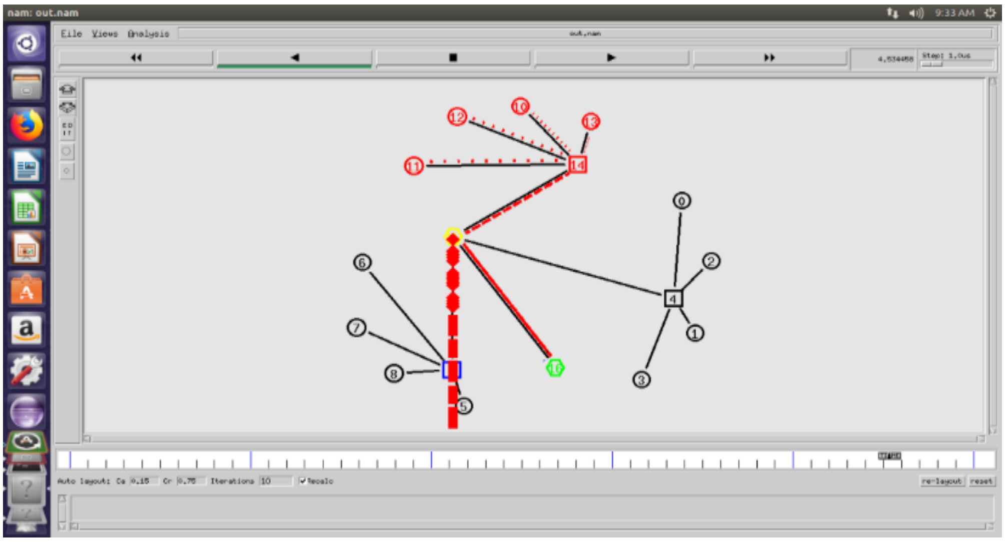

Figure 4. Network topology with TCP flow

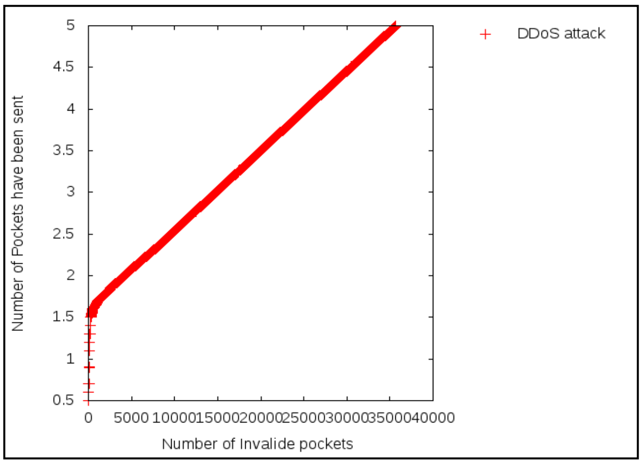

Figure 5. DDoS attacks

3.2. Simulation result of TCP flood

In this thesis, section a scenario for each DDoS attack will be simulated. Some of the topology will be wireless and others WLAN. The simulation scenario of this type of attack consists of 5 groups each group has 7 nodes.

Num_nodes = num_group * num_size

So, the Number of Nodes 35 nodes. Message port is forty-two to send message kind one cluster to alternative; every agent keeps track of what messages it's seen and solely forwards those that it hasn't seen before. This demonstration conjointly includes a server and a communications protocol backlog queue. The communications protocol backlog queue is employed to carry a packet's request, till it receives its final acknowledgement or till its period of time expires. Initial the consumer sends a SYN packet request to the server. Once the server receives the packet, it sends back to the sender node a SYN-ACK request packet. The client's request is hold on the communications protocol backlog queue. As before long because the consumer receives the SYN-ACK request, it'll reply to the server with a SYN-ACK-ACK. The server can receive the client's SYN-ACK-ACK and an association to the server is established. The client's initial request is aloof from the communications protocol backlog queue. The method can continue during this same manner whenever a brand-new request has arrived. Each message is of the shape "ID: DATA" wherever ID is a few arbitrary message symbol and knowledge is that the payload. So as to cut back memory usage, the agent stores solely the message ID as shown in Figure 6.

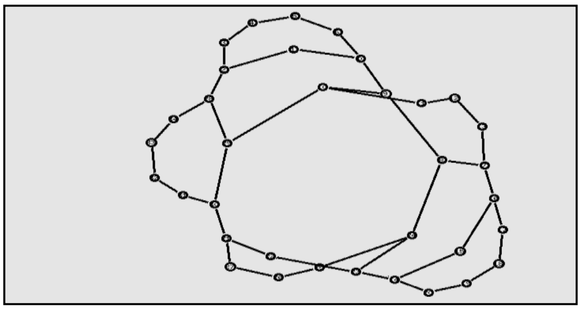

Figure 6. TCP flooding attack

3.3. Simulation result of SYN TCP flood

The SYN flood attack demonstrates a two-way acknowledgement. This demonstration exhibits of however associate degree actual SYN flood attack happens and what happens throughout that point amount. The topology includes thirty-five nodes node zero is server. Two nodes, whose color is red, represent victims.

The server is that the targeted nodes, the protocol drop tail queue stores all received SYN request with their information science addresses. The wait time is that the lifespan of every packet since it had been received by the server and waits for a final acknowledgement from the assailant. Throughout the SYN flood attack, a mix of attackers and traditional computers begin to form requests to ascertain an association to the server. Attackers can begin causation out an outsized variety of [*fr1] open SYN packets, employing a spoofed supply information science address, to form letter of invitation to attach to the server. The packet of the attacker's SYN request packet is BLACK. Once the server receives the request it'll transport a SYN-ACK request to the spoofed information science address and expect its response that it'll ne'er receive. The packet color changes to YELLOW. Every request is going to be keep within the protocol backlog queue and can expire once its wait time runs out. For this demo the wait time is found next to every packets request on the protocol backlog queue. At a similar time, the regular computers can begin creating requests to attach to the server yet. The protocol backlog queue can become full since it's attempting to method request quicker than it will handle them. At now a trash bin and a lock .The lock represents the protocol backlog queue is full so no new SYN request may be accepted. The trash bin represents a number of the packets being born. It shows access being denied as a result of the protocol backlog queue is full. Once the wait time of every packet, that is xxxii seconds for this demonstration, runs down the SYN packet are going to be aloof from the protocol backlog queue. The new incoming packets are going to be accepted as shown in Figure 7.

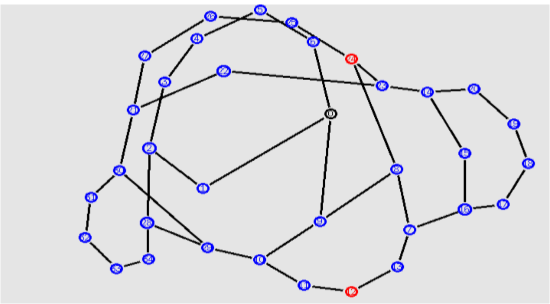

Figure 1. SYN flooding

3.4. Simulation result of ICMP flood

TCP and ICMP have been used to generating traffic for generating DDoS flood. Flooding attacks deny service in 2 ways:

- Generating an enormous volume of traffic that exhausts information measure on the backbone links.
- Generating a high packet rate that exhausts the processor at associate degree intermediate router or the target host. During this experiment, we've generated communications protocol and ICMP information measure flood with FLAT, PULSE and RAMP distributions to attain attacks victimization ping.

Ping enable node to verify that informatics exists and settle for request. Ping works by causation a web management Message Protocol (ICMP) Echo Request to such interface on the network and anticipating a reply. After execute the code the next process will be shown as follow:

- Node 1 received ping answer from 4.
- Node 2 received ping answer from 5.
- Node 3 received ping answer from 6.
- Node 4 received ping answer from 1.
- Node 5 received ping answer from 2.
- Node 6 received ping answer from 3.
- No pockets have been dropped.

After that each single node gets ping answer, here there are no packets dropped since there is a direct connection between all the pair of nodes via node 0.

- Node 2 gets ping answer from 5 with round-trip-time 222.0 ms.
- Node 3 gets ping answer from 6 with round-trip-time 201.0 ms.
- Node 5 gets ping answer from 2 with round-trip-time 222.0 ms.
- Node 6 gets ping answer from 3 with round-trip-time 201.0 ms.
- No of packets dropped: 2

Figure 8 depicts the aforementioned nodes. Therefore this is often sometimes not a decent alternative. Using simple algorithm in the victim's router to check the size of the pockets, the pocket size with greater than 1500 bytes, its IP will be blocked for 30 minutes.

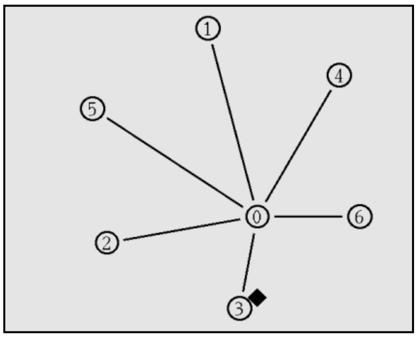

Figure 2. ICMP flood

3.5. Simulation result of spoofing

A spoofing attack could be a state of affairs during which associate assailant with success masquerades as associate other node by determination knowledge and thereby gaining an illegitimate advantage. This attack consists in targeting routing data whereas it's being exchanged: making routing loops, attracting or offensive network traffic from selected nodes, extending and shortening supply routes, generating pretend error messages, partitioning the network, etc. Which the attacker transmits bursts of duration L at rate R in a deterministic on-off pattern that has period T. When the rate R coupled with existing traffic becomes greater than the link capacity loss is incurred. Each simulation done is seen as in Figure 9 with the attacking node being seen as the red node and the genuine nodes as those in black. The nodes were initially simulated between two extremes of 7 nodes and 20 nodes and progressively increased in between that range during the stress testing phase of the project.

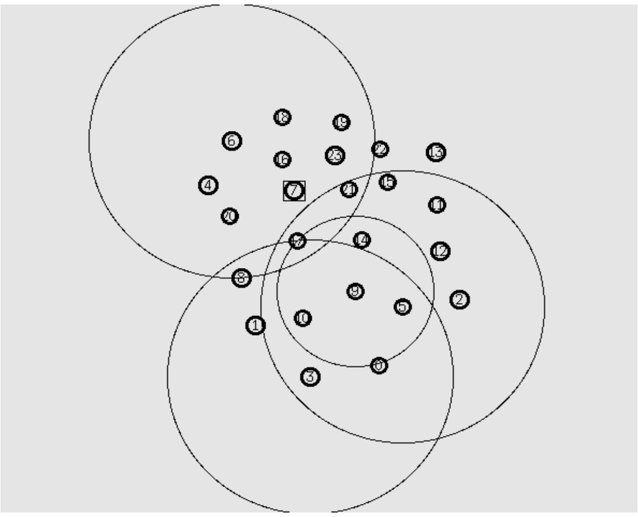

Figure 3. Spoofing attack

By setting the duration L to be more than the RTT of the flows and period T to be slightly more than minimum RTO value, TCP flows can be forced to repeatedly time out, thus obtaining virtually zero throughputs. After executed the TCL code there are three extra files will be generated the file with name ICMP with tr extension will be used to generate xgrapgh as shown in Figure 10. To sum up, this study has successfully accomplished the objectives in which a comprehensive simulation has been conducted in order to highlight new attacks.

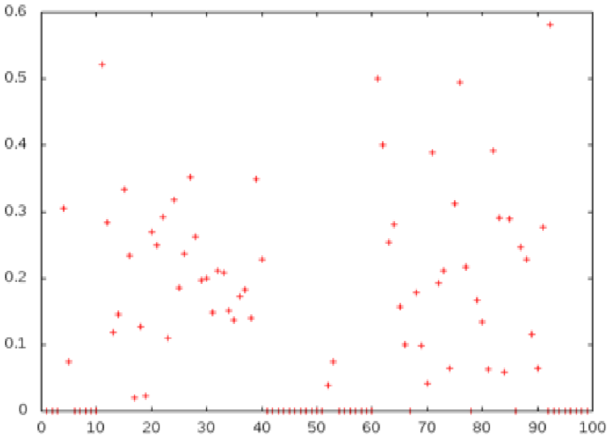

Figure 4. Xgraph for ICMP attack

4. CONCLUSION

This paper has conducted a comprehensive simulation in order to figure out and detect DDoS attacks. Using NS-2 simulator, the experiments showed that different types of DDoS have been characterized, examined and detected. This implies the efficacy of the comprehensive simulation proposed by this study. For future researches, examining specified networks such as IoT in terms of configuring DDoS would be a great opportunity.

ACKNOWLEDGEMENTS

I would like to thank Sama Abdullah for her support and encouragement during my study. Your continuous help and understanding have made my life full of love and I am grateful for everything you have done.

REFERENCES

[1] Q. Yan, and F. R. Yu, “Distributed denial of service attacks in software-defined networking with cloud computing,” *IEEE Communications Magazine,* vol. 53, no. 4, pp. 52-59, 2015.

[2] S. Swathi, and H. Yogish, “Secure data aggregation in IoT using efficient-CSDA,” *International Journal of Electrical and Computer Engineering (IJECE),* vol. 9, no. 6, pp. 4889-4897, 2019.

[3] A. Riyad, M. Ahmed, and R. Khan, “An adaptive distributed intrusion detection system architecture using multi agents,” *International Journal of Electrical and Computer Engineering (IJECE),* vol. 9, no. 6, pp. 4951-4960, 2019.

[4] B. Ambore, “Novel model for boosting security strength and energy efficiency in internet-of-things using multi-staged game,” *International Journal of Electrical and Computer Engineering (IJECE),* vol. 9, no. 5, pp. 4326-4335, 2019.

[5] S. Aljawarneh, M. Aldwairi, and M. B. Yassein, “Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model,” *Journal of Computational Science,* vol. 25, pp. 152-160, 2018.

[6] H. Hindy, D. Brosset, E. Bayne, A. Seeam, C. Tachtatzis, R. Atkinson, and X. Bellekens, “A taxonomy and survey of intrusion detection system design techniques, network threats and datasets,” *arXiv preprint arXiv:1806.03517,* 2018.

[7] P. Mishra, V. Varadharajan, U. Tupakula, and E. S. Pilli, “A Detailed Investigation and Analysis of Using Machine Learning Techniques for Intrusion Detection,” *IEEE Communications Surveys & Tutorials,* vol. 21, no. 1, pp. 686-728, 2019.

[8] Z. Tan, A. Jamdagni, X. He, P. Nanda, and R. P. Liu, “A system for denial-of-service attack detection based on multivariate correlation analysis,” *IEEE transactions on parallel and distributed systems,* vol. 25, no. 2, pp. 447-456, 2014.

[9] N. Lyamin, A. Vinel, M. Jonsson, and J. Loo, “Real-time detection of denial-of-service attacks in IEEE 802.11 p vehicular networks,” *IEEE Communications letters,* vol. 18, no. 1, pp. 110-113, 2014.

[10] H. Zhang, P. Cheng, L. Shi, and J. Chen, “Optimal denial-of-service attack scheduling with energy constraint,” *IEEE Transactions on Automatic Control,* vol. 60, no. 11, pp. 3023-3028, 2015.

[11] B. A. Tama, and K.-H. Rhee, “An in-depth experimental study of anomaly detection using gradient boosted machine,” *Neural Computing and Applications,* vol. 31, no. 4, pp. 955-965, 2019.

[12] I. Ullah, and Q. H. Mahmoud, “A Two-Level Hybrid Model for Anomalous Activity Detection in IoT Networks,” *2018 Iranian Conference on Electrical Engineering (ICEE),* pp. 1-6, 2019.

[13] C. Khammassi, and S. Krichen, “A GA-LR wrapper approach for feature selection in network intrusion detection,” *Computers & Security,* vol. 70, pp. 255-277, 2017.

[14] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, “A Deep Learning Approach to Network Intrusion Detection,” *IEEE Transactions on Emerging Topics in Computational Intelligence,* vol. 2, no. 1, pp. 41-50, 2018.

[15] S. N. Mighan, and M. Kahani, “Deep Learning Based Latent Feature Extraction for Intrusion Detection,” pp. 1511-1516, 2018.

[16] V. Hajisalem, and S. Babaie, “A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection,” *Computer Networks,* vol. 136, pp. 37-50, 2018.

[17] M. Şimşek, and A. Şentürk, “Fast and lightweight detection and filtering method for low-rate TCP targeted distributed denial of service (LDDoS) attacks,” *International Journal of Communication Systems,* vol. 31, no. 18, pp. e3823, 2018.

[18] E. Bukharov, D. Zybin, A. Soloviev, and A. Kalach, “Mathematical simulation of countermeasures to attacks of “denial of service” type with the use of game theory approach,” *Journal of Physics: Conference Series,* pp. 012076, 2019.

[19] X. Wang, N. Guo, F. Gao, and J. Feng, “Distributed denial of service attack defence simulation based on honeynet technology,” *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-16, 2019.

[20] T. K. Mohd, S. Majumdar, A. Mathur, and A. Y. Javaid, “Simulation and Analysis of DDoS Attack on Connected Autonomous Vehicular Network using OMNET++,“ *2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON),* pp. 502-508, 2018.

[21] S. Mohapatra, and P. Kanungo, “Performance analysis of AODV, DSR, OLSR and DSDV routing protocols using NS2 Simulator,” *Procedia Engineering,* vol. 30, pp. 69-76, 2012.

[22] A. Leon, O. Parra, and G. Bermudez, “LAN-WAN-LAN end-to-end Network Simulation with NS2,” *International Journal of Applied Engineering Research,* vol. 13, no. 17, pp. 13136-13140, 2018.

[23] P. S. Rathore, A. K. Pandey, and D. N. Le, “Computer Network Simulation in NS2: Basic Concepts & Protocols Implementation,” 2018.

[24] R. Patel, N. Patel, and S. Patel, “An Approach to Analyze Behavior of Network Events in NS2 and NS3 Using AWK and Xgraph,” *Information and Communication Technology for Competitive Strategies,* pp. 137-147, 2019.

[25] P. Meeneghan, and D. Delaney, “An introduction to NS, Nam and OTcl scripting,” *National University of Ireland,* 2004.