Journal of Information Security, 2018, 9, 225-241
http://www.scirp.org/journal/jis
ISSN Online: 2153-1242
ISSN Print: 2153-1234

Generation of DDoS Attack Dataset for Effective IDS Development and Evaluation

Sabah Alzahrani, Liang Hong

Department of Electrical & Computer Engineering, Tennessee State University, Nashville, TN, USA
Email: salzahr1@my.tnstate.edu, lhong@tnstate.edu

How to cite this paper: Alzahrani, S. and Hong, L. (2018) Generation of DDoS Attack Dataset for Effective IDS Development and Evaluation. *Journal of Information Security*, **9**, 225-241.
https://doi.org/10.4236/jis.2018.94016

Received: June 11, 2018
Accepted: August 13, 2018
Published: August 16, 2018

Abstract

Distributed Denial of Service (DDoS) attacks are performed from multiple agents towards a single victim. Essentially, all attacking agents generate multiple packets towards the victim to overwhelm it with requests, thereby overloading the resources of the victim. Since it is very complex and expensive to conduct a real DDoS attack, most organizations and researchers result in using simulations to mimic an actual attack. The researchers come up with diverse algorithms and mechanisms for attack detection and prevention. Further, simulation is good practice for determining the efficacy of an intrusive detective measure against DDoS attacks. However, some mechanisms are ineffective and thus not applied in real life attacks. Nowadays, DDoS attack has become more complex and modern for most IDS to detect. Adjustable and configurable traffic generator is becoming more and more important. This paper first details the available datasets that scholars use for DDoS attack detection. The paper further depicts the a few tools that exist freely and commercially for use in the simulation programs of DDoS attacks. In addition, a traffic generator for normal and different types of DDoS attack has been developed. The aim of the paper is to simulate a cloud environment by OMNET++ simulation tool, with different DDoS attack types. Generation normal and attack traffic can be useful to evaluate developing IDS for DDoS attacks detection. Moreover, the result traffic can be useful to test an effective algorithm, techniques and procedures of DDoS attacks.

Keywords

DDoS, IDS, Signature, Anomaly, Cloud, Machine Learning, Big Data, DataSet, Simulation, Traffic Generator

1. Introduction

The success of any attack lies in the cooperation of the DDoS agents. The coop-

eration occurs in two stages, namely the compromise stage and the attack stage. An attacker will compromise available defenseless systems and install attack tools, thereby turning the machines into zombies. The second stage involves sending attack commands into the zombie machines via a secure mechanism so as to target a specific victim [1]. Cyber security experts and other researchers are faced with the challenges of unraveling DDoS attack vectors as well as ways to prevent such attacks. The scholars conduct attack simulation using either real data or simulated data based on previous attack characteristics. Simulation involves tools that have attack agents and defense agents. Attack agents are the daemon which is attack executors and master which is the attack coordinator. Defense agents are the sensors, samplers, detectors, filters and investigators [2]. Therefore, determining the various ways in which the researchers collect data for use in DDoS attack simulation is of importance in order to contribute towards enhancing the simulation methods or devising better replication mechanisms.

This study expands on the knowledge base by using the OMNET++ simulation tool to generate normal and attack traffic. The data gathered from this simulation can be used to formulate new intrusion detection systems (IDS) that are able to predict different DDoS attack types. The traffic generator also has a huge future potential as a tool for testing the accuracy of newly developed IDS.

The key contribution of this research is the identification of the need for a cloud DDoS attack dataset due to the lack of public dataset.

The paper is organized as the following: starting with a review of existing literature DDoS, including its definition, history, and its effects on computer systems. This section includes a brief review of the commonly used public DDoS datasets. Intrusion detection systems (IDS) are described by the different techniques that are employed by IDS, and their relative strengths and limitations. A detailed background of the study is then given, which provides descriptions of the most common types of DDoS attacks. Tools to prevent DDoS attacks are then described, which includes tools such as traffic simulation, DDoS datasets and traffic generators. Building up to this knowledge, different attack scenarios are then described, with sample parameters provided for each example.

2. Literature Review

Distributed denial of services abbreviated as DDoS refers to an attack consisting of a number of nodes attacking a single node at the same time interval with a specified number of messages [3]. In this type of attack, the single node is the target and it is being attacked by several systems that are already compromised. The result is that the users are denied the services that are rendered by the target system hence the phrase "denial of services". The overwhelming messages directed towards the target machine causes the machine to shut down and the legitimate user suffers the loss of service. It is still a challenge to clearly distinguish between legitimate traffic and DDoS attack traffic [4]. The attack has a number of consequences. First, the efficiency of the site is significantly affected, the rep-

utation of the organization goes down and lastly, there is a loss of revenue and productivity.

DDoS assaults are security anomalies that threaten the operation of computer networks [5]. These attacks have resulted in the loss of vital data and equally colossal monetary value. While quantifiable DDOS assaults are rarely attainable from the real network setting, it is prudent to setup simulated DDoS assaults. However, this approach is technically intensive in terms of configurations. Another practical approach that security experts have adopted is the use of gadgets from the hacker's end. In this case, when the assault applications and network traffic are appropriately configured, the mockup will have similarities with the actual DDoS attacks. Although there are various assault gadgets, the most pertinent shortfall when it comes to adopting some of these tools were developed when cyber-attacks were gaining momentum. On the other hand, current cyber-attacks are highly sophisticated and require simulations that equally advanced for successful experiments.

The reality is that computer-generated traffic underpinned by the old gadgets does not have the capacity to model the contemporary and sophisticated cyber espionage in large heterogeneous networks. At that point, it becomes paramount to deploy state-of-the-art methodologies to replicate DDoS related attacks. Essentially, a test bed simulator could be employed to depict DDoS attacks. Adopting the replication mechanism will be applied by different traffic generation standards such as the real-world traffic packets, virtual traffic flow and experiment adoption resolution [5]. Although most scholars use virtual simulators, most security-related researches utilize simulants or test bed. While this approach is rather pragmatic, because the replicated traffic reflects the actual DDoS assault traffic, the challenge comes with the hefty economic aspect and the required technological knowhow when it comes to installing and operating the controlling interface. The paper sets out to discuss various DDoS assault replication strategies to undertake broad and recurrent experiments that utilize commercial traffic-generation systems such as Spirent Test Center platform (STC) and configuration manual.

Some of researchers have used an existing DDoS Data set. It has been argued that most of the public data sets have redundant instances, thus make the detection and classification of the DDoS ineffectual [6]. The authors were also argued that no available data sets such as KDD 99 which include new DDoS types, such as HTTP flood and SIDDOS. In their research, they collected a new dataset which includes four types of attack UDP flood, Smurf, HTTP Flood and SIDDOS.

Moreover, they are a lack of public dataset. This can be affected testing and evaluating of IDSs. Many existing datasets such as KDD 99, DARPA and other public dataset, are uncontrollable, unmodifiable, and may contain old types of attack.

Mukkavilli, Shetty, and Hong [6] present an experimental platform designed for representing a practical interaction between cloud services and users. More-

over, the network traces that results from such interaction are also collected to conduct anomaly detection. In particular, this experiment is performed using Amazon web services (AWS) platform. It explores the generation of labeled datasets for quantifying the security threats impact to cloud data centers. Among the researchers, the detection of instruction is an exciting topic. Specifically, the discovery of anomaly is one of the vital factors that help in detecting several novel attacks. Due to the complexity of these systems, however, its application has not been appropriate.

In network instruction detection, the anomaly based approach usually suffers from comparison, deployment, and evaluation that results from the publicly available network trace datasets that are less adequate. As a result of the cloud computing environments ubiquity in the cloud data centers, the impacts of the network attacks in the cloud data centers need to be assessed. Apparently, no publicly available dataset can capture the anomalous and normal traces of network in the interactions between cloud data centers and users. Evidently, some of the attacks that take place in the network include Port-scan, DDos, and the man-in-the-middle or ARP spoof. Even though several services such as infrastructure and software are offered by cloud computing to their customers, they also pose significant risks of security to client data and application beyond what is expected by the use of traditional on-premises architecture. Having access to the traces of network in the cloud can help in understanding these security risks.

The attack has a number of consequences. First, the efficiency of the site is significantly affected, the reputation of the organization goes down and lastly, there is a loss of revenue and productivity.

There exist publicly available datasets that researchers use for testing their technique and algorithm performance. Some of the datasets are obtained from real attacks while others are a consolidation of simulated attacks. However, there is need to note that the statistics provided by a dataset are usually different from the real features given by the real network traffic [7]. A comparison of the different datasets used in the simulation of DDoS attacks is shown in **Table 1**. The table gives the dataset name, the provider and date of harnessing. Further, it states if the dataset is obtained from a real attack or a planned attack.

3. Background

An Intrusion Detection System (IDS) can be a software or hardware for monitoring and detection any thread against a system. There are two main approaches for detection. a signature/rule based detection and anomaly based detection. However, a signature based detection technique compares known information to already captured signatures stored in the database. This technique is only able for detection of known attacks and has low false alarm. Unlike the first approach, an anomaly Based Intrusion Detection System observes the behavior of an event and determines any forms of anomalies. Thus, it is able to detect an unknown attack but with higher false alarm.

Table 1. Comparation on different datasets used for DDoS attacks detection.

Dataset Name	Author	Date	Real or Simulated	Features	DDoS attack Types	Dataset Size	Availability	Advantage	Limitation
KDD'99 Cup dataset [8]	MIT Lincoln Labs		Simulated	-two weeks of attack-free encounters and five week attack instance -output divided into 5 categories of ;DOS, Probe, R2L, U2R, and Normal -has 38 total attack types	SYN flood	743 MB	Available	-easily obtainable -many attack type available	-heavily imbalanced dataset with 80% attack traffic.
CAIDA DDoS Attack 2007 dataset [9]	Paul Hick	Aug 4, 2007	Simulated	-consist of data anonymized within one hour - resource consumer	UDP flood	21 GB	Quasi-restricted	-available for public use -effective to handle large DDoS attack above 5 Gb -traces can be read on any software reading tcpdump	-non-attack traffic is unavailable -does not include payload packets
EPA http dataset	Laura Bottomley	Aug 29, 1995	Real	-46,014 GET requests - 1622 POST requests -107 HEAD requests -6 invalid requests -One-second accuracy on timestamp	HTTP flooding	4.4 MB	Available	-smaller da-taset size	-cannot determine legitimate and illegitimate HTTP requests -small dataset may limit the extent of attack detection
DARPA_2009_malware-DDoS_attack-20091104	University of Southern California-Information Sciences Institute	Nov 4, 2009	Real	-background traffic and malware attack on compromised hosts of 172.28.0.0/16 IP range. -Attack performed on non-local target of IP 152.162.178.254 at TCP port 499	Malware DDoS attack	346.5 MB	Quasi-Restricted	-contains vectors for attacks from real DDoS attacks	
DARPA_2009_DDoS_attack-20091105	University of Southern California-Information Sciences Institute	Nov 5, 2009	Real	-SYN floods targeted on one IP address (172.28.4.7) - The attack also has background traffic -DDoS traffic from 100 separate IPs	SYN flood	1.01 GB	Quasi-restricted	-consist attack from multiple real sources hence able to learn attack vectors	-Attack targeted to one victim only does not determine the overall network strength
NSL-KDD dataset [10]	Mahbod Tavallee, Ebrahim Bagheri, Wei Lu, Ali A. Ghorbani	2009	Simulated	-Continuous Duration -Discrete protocol -Discrete service	Back, Land, Neptune, Process table, Worm (10), Apache2.	124 MB	Available		
ISCX dataset [11]	Unknown	June 11, 2010 to June 17, 2010	Simulated	-practical network and traffic -Labeled dataset -different intrusion scenarios	HTTP, SMTP, SSH, IMAP, FTP	84.46 GB	Available		

Continued

1998 FIFA World Cup Dataset	Martin Arlitt	April 30, 1998-July 26, 1998	Real	-1.35 billion requests	HTTP attack	307 MB	available	-Timestamp resolution of 1 second	
DoS_80-20110715 [12]	University of Southern California-Information Sciences Institute	July 15, 2011	Simulation	- Consist of only one attack	TCP SYN/ ACK attack	32.31 GB	Restricted	-8 known false positives already defined	-There is a lot of many identical packets
DoS_80_timeseries-20020629	University of Southern California-Information Sciences Institute	June 29, 2002 to Nov 30, 2003	Real	-Time series of 80 DoS attacks - one millisecond granularity time series of 80 DoS attacks	Reflector attack, TCP no-flag attack, IP proto attack	783.1 MB	Quasi-restricted	-shortest time series of 1 millisecond granularity	
DoS_traces-20020629	University of Southern California-Information Sciences Institute	Jun 29, 2002 to Aug 14, 2002	Real	-Time series of 80 DoS attacks - one millisecond granularity time series of 80 DoS attacks	Reflector attack TCP-no flag attack IP-proto 255 attack	4.1 GB	Restricted		
FRGPNTP Flow Data-20131201	Colorado State University	Dec 01, 2013 to Feb 28, 2014	Anonymized	3 months daily NTP in Argus flow on 10 Gb/s link	NTP reflection attack	3.5 TB	Restricted		
FRGP_NTP_Flow_Data_anon-20131201	University of Southern California-Information Sciences Institute	Dec 01, 2013 to Feb 28, 2014	Anonymized	3 months daily NTP in Argus flow on 10 Gb/s link Attackers trigger attacks by sending monlist queries containing spoofed IP addresses to NTP running hosts. Hosts reply with list of last clients	NTP reflection attack	726.7 GB	Quasi-Restricted	Large set of data containing vectors to measure multiple attack types including spoofing	
FRGP_SSDP_Reflection_DDoS_Attack_Traffic-20140930	Colorado State University	Sept 30, 2014	Simulated	-UDP simple service discovery protocol (SSDP) attack traffic -attack flow on 10 Gb/s link -attack triggered via UPnP/SSDP discovery using spoofed source IP to vulnerable hosts running SSDP	SSDP reflection attack	26 GB	Restricted		
FRGP_SSDP_Reflection_DDoS_Attack_Traffic_anon-20140930	University of Southern California-Information Sciences Institute	Sept 30, 2014	Anonymized/ simulated	-3 hour DDoS attack traffic using Argus -UDP simple service discovery protocol (SSDP) attack traffic -attack flow on 10 Gb/s link -Uses prefix-preserving algorithm to anonymize IPs attack triggered via UPnP/SSDP discovery using spoofed source IP to vulnerable hosts running SSDP	SSDP reflection attack	4.99 GB	Quasi-Restricted		

Continued

Mirai-B-scanning-20160601	University of Southern California-Information Sciences Institute	June 01, 2016 to Mar 30, 2017	Real	-only Mirai-identified TCP SYN on ports 23 and 2323 -traffic only for IP address 130.152.184.2 and 192.228.79.0/24	TCP SYN	1.1 GB	Quasi-Restricted	Contain data of real attacks hence able to prevent future such attacks	Picking only Mirai-attacks limits researchers from other attacks in the same trace.
Mirai-FRGP-scanning-20160908	University of Southern California-Information Sciences Institute	Sept 08, 2016 to Oct 31, 2016	Real	-only traffic flow matching Mirai scanning signature identified by Argus on ports 23 and 2323	Mirai TCP attack	567 GB	Quasi-restricted	Contain data of real attacks hence able to prevent future such attacks	Picking only Mirai-attacks limits researchers from other attacks in the same trace.

Serpanos & Douligeris (2007) argue that despite the existence of different types of DDoS attacks, all have a primary role to hinder legitimate users from accessing internet traffic thereby making use of different resources. The authors note that DDoS attacks fall into three main categories: network attacks; protocol attacks; and application layer attacks. However, in our research paper, we will be focused on the main flooding attacks—HTTP flooding, ICMP attack, TCP SYN, and UDP flood.

A UDP flood describes a type of DDoS attack where a server is flooded with User Datagram Protocol packets in an attempt to overwhelm its ability to process the requests and respond appropriately. The server receives the UDP requests and keeps checking whether currently running programs are listening for requests at given ports and upon finding none, it responds with a destination unreachable message. With flooding of UDP requests, the server becomes overwhelmed and its capacity to process and respond to requests is hampered. Figure 1 shows a simple diagram of a UDP flood attack.

An ICMP attack on the other hand, describes a DDoS attack where a target resource is overwhelmed with ICMP packets, sending the packets at such a high rate without giving time to wait for reply. As the victim's server attempts to respond to requests, it becomes overwhelmed thereby shutting down. Figure 2 shows a simple diagram of an ICMP attack.

A TCP SYN flood attack exploits weaknesses in the TCP connection where instead of a SYN request being answered by a SYN-ACK response, multiple SYN requests are sent to a target forcing it to wait for responses thereby binding its resources until the response is received. Figure 3 shows a simple diagram of a TPC SYN flood attack.

HTTP flood attack exploits the HTTP GET or POST requests to attack a given resource. However, unlike the other flooding attacks that make use of spoofing techniques or compromised packets, a HTTP flood attack forces a resource to allocate. Figure 4 shows a simple diagram of an HTTP flood attack.

4. Overview of DDoS Attack Simulation Methods and Tools

Researchers and organization seeking to mitigate DDoS attacks usually simulate

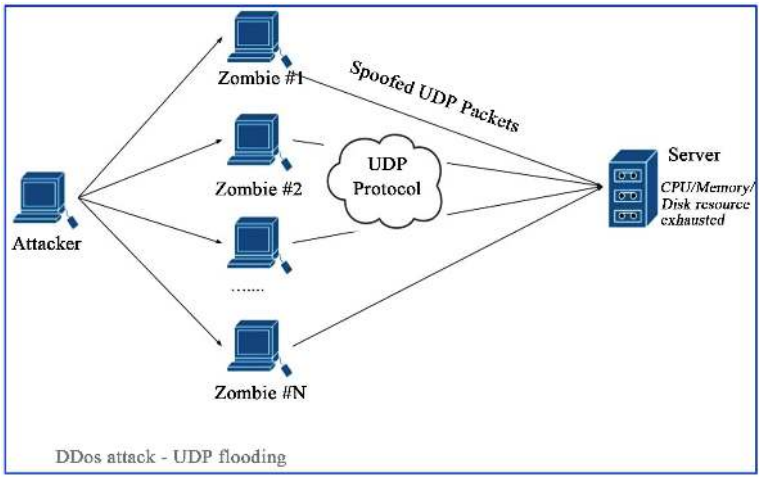

Figure 1. UDP Flooding attack.

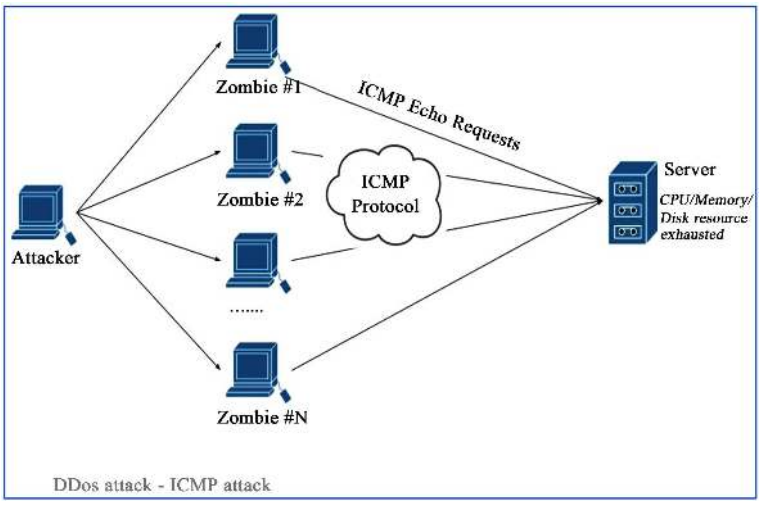

Figure 2. ICMP attack.

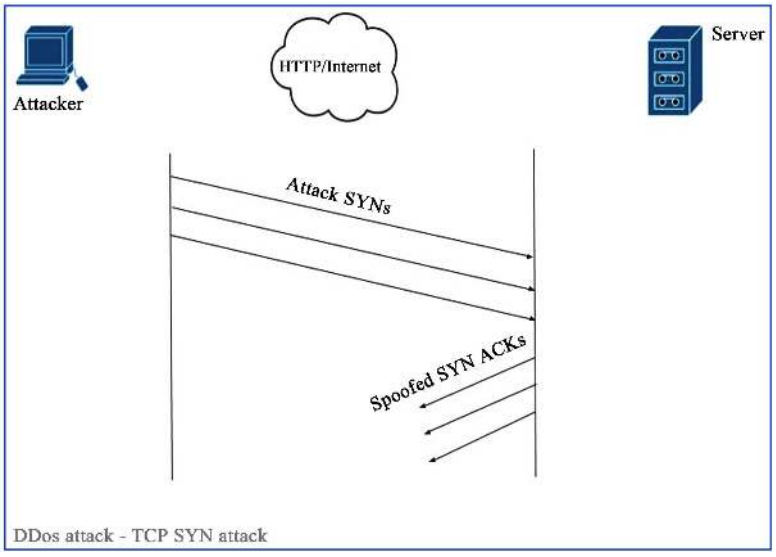

Figure 3. TCP-SYN Attack.

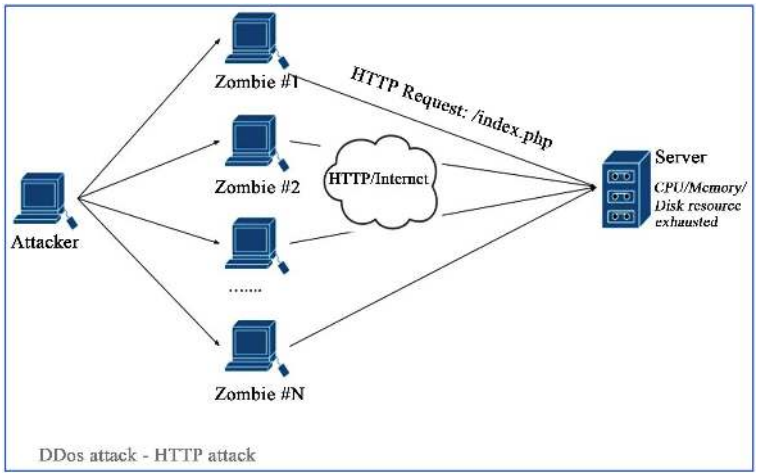

Figure 4. HTTP attack.

the attacks to determine the detective and protective measures to place on the network and machines. Umarani and Sharmila simulated a HTTP attack to depict application layer Denial of Service attacks through machine algorithms. They proposed a method that used the dataset from the 1998 FiFa World cup to categorize traffic flow as either DOS attack or legitimate access [13]. They created an access matrix using the available HTTP traces. Their simulation proved to be more effective with an increase of 0.9% in average detection rate and 4.11% increase in false positive rate.

Pushback is another defense mechanism that uses congestion-control problem to shield against DDoS attacks. The approach uses the two steps namely detection and selective drop to simulate attacks. Researchers Kumarasamy and Asokan introduced puzzle solving as a preventive measure against DDoS attacks. The puzzle method required a victim server to send a puzzle to a client sending traffic. The client gained access upon successfully solving the puzzle. When the target server determines a probable malicious client, it sends a complicated puzzle. The client is unable to solve the puzzle implying that the traffic by the client is not allowed through to the server. There is great reliance on machine learning algorithms to detect DDOS attacks [14]. Using an NS2 simulator researchers are able to analyze different forms of DDoS attacks. NS2 produces valid results that reflect scenarios in real environment.

There are different simulation tools that are in use by researchers. Researchers choose a simulation contrivance depending on the data type the tool handles as well as the report presentation. Some of the tools used in DDoS attack simulation include NS2, LOIC, XOIC, HULK, PyLoris, DAVOSET and DDoS flowgen. Table 2 shows a comparison of the different DDOS Attack tools with accompanying descriptions [15].

5. Traffic Generator Design and Implementation

This paper proposed DDoS attack traffic generator-based network simulation. A

Table 2. Some different DDOS attack available tools.

Simulation Tool	Protocol	Attack	Description
Trinoo	UDP	UDP flood	• Greatly used by research community • Bandwidth depletion tool that launches coordinated UDP floods against IP addresses • Does not spoof source address
Ddosflowgen [16]	UDP, TCP	UDP flood, TCP requests, Mirai scans	• Can handle attacks beyond 1Tbps(terabits per second) • Generates synthetic traffic datasets from N views • Ability to define number of attacking networks and adjust parameters like amplification factor, attack vectors, number of network attack sources • Human-readable topology
OMNET++ [17]	UDP, TCP, ICMP	Transport layer attack	• Capable of TCP/IP simulation • Manageable form a web server • Cannot generate traffic
Tribe Flood Network (TFN)	TCP protocol and UDP and ICMP protocols	TCP SYN and , ICMP flood, smurf	• Used to deplete bandwidth and resources • employs command line interface for attacker and control master communication • Unencrypted
TFN2K	TCP,UDP,ICMP	ICMP flood, SYN flood, UDP flood, smurf,	• Advanced version of TFN DDoS attack tool • Encrypts message among attack components • Uses CAST-256 algorithm to encrypt communication between attacker and control master program • Forges packets to appear to originate from close systems • Converts covert exercises to hide from intrusion detection systems
Stacheldraht	ICMP protocol and UDP and TCP	TCP SYN flood, UDP flood, ICMP echo request flood	• Combines features of TFN and Trinoo to eliminate weaknesses of TFN • Automatic agent updates • Encrypted telnet communication between handlers and attackers • Communicates via ICMP and TCP packets
Rnstream	TCP,UDP	TCP ACK flood	• Simple point-to-point TCP ACK flood tool that overpowers the fast routing routine table in switches • Unencrypted communication via TCP/UPD packets • Master connects to zombie via telnet • ACK packets hit target then and sends TCP RST to spoofed IP addresses • Routers responds with ICMP unreachable leading to bandwidth starvation • Creates random source IP address bits as a spoof approach
Shaft	ICMP, UDP, and TCP	TCP flood, UDP flood, ICMP flood	• It is the successor of Trinoo • Handlers and agents communicate via UDP • It randomizes source port and IP addresses in packets • Fixed packet size during attack • Switches control master servers and ports in real time thereby making it difficult for intrusion detection tools
LOIC	TCP, UDP, HTTP	UDP, TCP, HTTP flood	• IRC based anonymous attacking tool • Exists as either binary or web-based versions

Continued

Knight	TCP,UDP	PUSH and flood, TCP and SYN and UDP flooding	• Very lightweight but powerful attack tool for IRC • It provides SYN, UDP, and urgent pointer attacks • Uses Back Orifice, a Trojan horse, for target host installation • Runs on windows operating system and automatically update via http or ftp • Contains a checksum generator
Trininty v3	UDP protocol and TCP protocol	TCP fragment, established and random flag floods, RST packet floods,	• TCP floods done by randomizing all the 32-bits of the source IP address • Flood packets generated via random control flags
WinCap and JpCap [18]	TCP,UDP,ICMP	TCP dump, UDP and ICMP dump	• Windows based program for transmitting network traffic and protocol stack process

new dataset will be collected including modern types of attack. A network simulator OMNET++ will be used in this work, because it can be worked with high confidence due to its capability of producing valid results that reflect a real environment. The collected data will be recorded for different types of attack that target the most critical network layers application and network. According to OMNeT++ website, "this simulator is an extensible, modular, component-based C++ simulation library and framework, primarily for building network simulators". Figure below is shown the Block Diagram the proposed system. The goal to use a simulation tool is to be developed as a traffic generator. The simulation was developed for the following attack HTTP, TCP-SYN, UDP flood and ICMP attacks. The main DDoS network topology design on MONET++ is illustrated in Figure 5.

OMNET++ simulation tool, code was developed to simulate DDoS attack. From Figure 5, such topology was developed as a cloud environment which connected with different networks data centers. This cloud is developed to be configurable for any scenarios parameters. Six continents data centers are connected in relation to the internet cloud. These continents include North America, South America, Asia, Africa, Europe, and Australia servers. In this simulation, the internet parameters are set based on real-world internet. The internet plays the fundamental role of connecting the people across the globe. Figure 6 provides a demonstration of the outlook of network topology of each continent data center. There are Clients as bad clients (hackers), normal clients, and server/s connected through a router.

It is evident from Figure 6 that the network topology has both the bad and the general clients who are connected to the same server via the router. An example case, it is set up the number of clients to be ten irrespective of whether they are the bad or the general clients as a default.

6. Developing DDoS Attack Scenarios

1) UDP Flood Scenario

As an example scenario, the server of Africa set up with both the VICTIM server and the UDP Echo server. Therefore, all the bad clients start the UDP

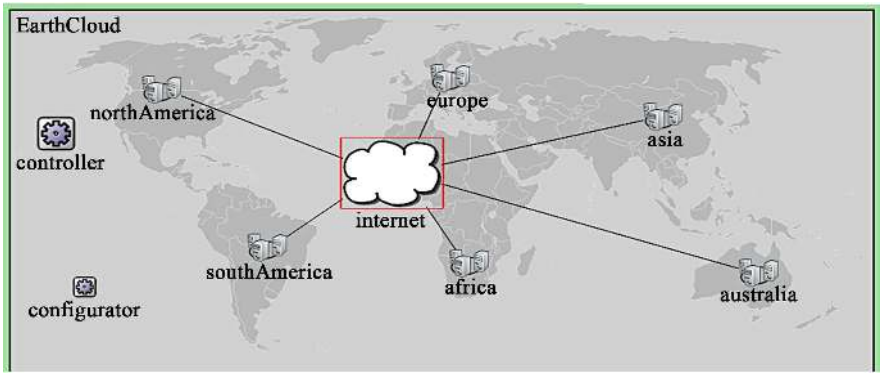

Figure 5. Main network topology in OMNET++.

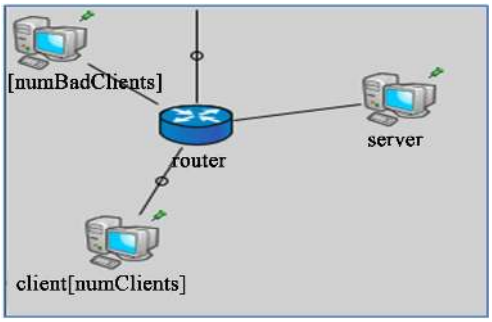

Figure 6. A network topology of each data center.

flood Attack to the server. In this scenario, the intruders use a malicious UDP network traffic to deny other users access to the server service just like in other network layer attacks. The UDP Flood attacks have more effect on the UDP Echo server for time synchronization. Figure 7 shows the parameters of the UDP flood attack which can be found in the omnet.ini file. The message size ranges between 512 to 1024 bytes and sent at an interval of 0.01 - 0.05 seconds. Therefore, all the bad clients (distributed hackers) will send 20 to 100 packets of messages per second to the victim server. The UDP attack will deny the victims' server service.

2) HTTP-GET/POST Attack Scenario

The HTTP protocol is one of the mostly used protocols that are supported by the application layer. Therefore, they are easily accessible from anywhere by using web applications. This makes the HTTP-GET attacks almost impossible to be detected by the classifying layer hence difficult to prevent. Besides, in most cases, the attackers send a legitimate request for service which acts as a usual user request for services from the server.

As an example, the server of Africa set up with both the VICTIM server and the HTTP WEB server. Thus, all the bad clients start the HTTP-GET Attack on that server. Figure 8 shows the parameters of HTTP-GET Attack.

Two different applications were installed in the browser of the general and the bad clients. The general HTTP browser application was installed on the general clients and an attack application called "HttpServerEvilA" was installed on the

```
[Config UDP]
description = "UDP Flood Attack"

**.NIDS.dataset_name = "dataset-udp-flood.txt"
**.client[*].numUdpApps = 1
**.client[*].udpApp[0].typename = "UDPBasicApp"
**.client[*].udpApp[0].destAddresses = "africa.server"
**.client[*].udpApp[0].destPort = 1000
**.client[*].udpApp[0].messageLength = uniform(512B, 1024B)
**.client[*].udpApp[0].sendInterval = uniform(0.01s, 0.05s)

**.server.numUdpApps = 1
**.server.udpApp[0].typename = "UDPEchoApp"
**.server.udpApp[0].localPort = 1000
```

Figure 7. Configuration parameters for UDP attack.

```
[Config HTTP-GET]   # HTTP-GET Attack
description = "HTTP-GET Flood Attack"

**.NIDS.dataset_name = "dataset-http-flood.txt"
**.NIDS.DDosType = "HTTP-GET"
**.httpProtocol = 11

# Server
*.africa.server.numTcpApps = 1
*.africa.server.tcpApp[0].typename = "HttpServer"
*.africa.server.tcpApp[0].hostName = "www.good.com"
*.africa.server.tcpApp[0].port = 80
*.africa.server.tcpApp[0].logFile = "http-server.log"
*.africa.server.tcpApp[0].siteDefinition = ""
*.africa.server.tcpApp[0].config = xmldoc("server_cfg.xml","//server-profile[@id='normal']")
*.africa.server.tcpApp[0].activationTime = 0.0

# This server generates documents containing a number of references to images
# stored on www.good.com. The unwitting browser will thus contribute to a DDoS attack
# against the victim site.
**.badClient[*].numTcpApps = 1
**.badClient[*].tcpApp[0].typename = "HttpServerEvilA"
**.badClient[*].tcpApp[0].hostName = "www.bad.org"
**.badClient[*].tcpApp[0].port = 80
**.badClient[*].tcpApp[0].logFile = ""
**.badClient[*].tcpApp[0].siteDefinition = ""
**.badClient[*].tcpApp[0].config = xmldoc("server_cfg.xml","//server-profile[@id='normal']")
**.badClient[*].tcpApp[0].activationTime = 0.0
**.badClient[*].tcpApp[0].minBadRequests = 3  # Very moderate attack
**.badClient[*].tcpApp[0].maxBadRequests = 8

# Clients
**.client[*].numTcpApps = 1
**.client[*].tcpApp[0].typename = "HttpBrowser"
**.client[*].tcpApp[0].logFile = "http-client.log"
**.client[*].tcpApp[0].scriptFile = "browse.script" # Lets use a script to simplify the test
**.client[*].tcpApp[0].config = xmldoc("browser_cfg.xml","//user-profile[@id='normal']")
**.client[*].tcpApp[0].activationTime = 0.0
```

Figure 8. Configuration parameters for HTTP attack.

bad client. The attack application is shown as a general server; however, it performs the attacks on the victim server.

3) TCP-SYN Attack Scenario

The below figure is shown the TCP connection (TCP 3-shake).

In the TCP-SYN attack, the bad client first sends the SYN packet to a server. For DDoS attack, after the bad client receives the SYN + ACK packet from the server, it sends the SYN packet instead of ACK packet. As results, the SYN backlog queue will be over on the server; therefore, the server will not be able to accept the general connection.

As an example, the server of Africa was set up with both the VICTIM server and the TCP Echo server. All the bad clients start the TCP-SYN Attack on the server. Figure 9 shows the parameters of TCP-SYN attack. Besides, these parameters exist in the omnet.ini file. The TCP Session App was installed on the general clients and bad clients. However, the bad client changed all the packets installed to the SYN packet using the NIDS. Figure 10 shows the TCP connection diagram (TCP 3-shake).

As a note in the TCP-SYN attack scenario on OMNET++, a message dialog will be shown because bad clients initiate TCP-SYN attacks. When many SYN packets arrive in the server, it causes an overflow queue of the SYN backlog implying that the TCP-SYN attack is successful. The overflow of the queue causes the occurrence of an OMNET++ message, but this does not mean it is an error.

```
[Config TCP-SYN]     # TCP-SYN Flood Attack
description = "TCP-SYN Flood Attack"

**.NIDS.dataset_name = "dataset-syn-flood.txt"
**.NIDS.DDosType = "SYN-FLOOD"

# Server
**.numTcpApps = 1
**.server.tcpApp[0].typename = "TCPEchoApp"
**.server.tcpApp[0].localPort = 1992
**.server.tcpApp[0].echoFactor = 2.0
**.server.tcpApp[0].echoDelay = 0

**.client[*].tcpApp[*].typename = "TCPSessionApp"
**.client[*].tcpApp[0].connectAddress = "africa.server"
**.client[*].tcpApp[0].connectPort = 1992
**.client[*].tcpApp[0].tOpen = 1s
**.client[*].tcpApp[0].tSend = 2s
**.client[*].tcpApp[0].sendBytes = 1024MiB
**.client[*].tcpApp[0].tClose = 250s

**.badClient[*].tcpApp[*].typename = "TCPSessionApp"
**.badClient[*].tcpApp[0].connectAddress = "africa.server"
**.badClient[*].tcpApp[0].connectPort = 1992
**.badClient[*].tcpApp[0].tOpen = 0.1s
**.badClient[*].tcpApp[0].tSend = 0.5s
**.badClient[*].tcpApp[0].sendBytes = 10000000B
**.badClient[*].tcpApp[0].tClose = 25s
```

Figure 9. Configuration parameters of TCP-SYN attack.

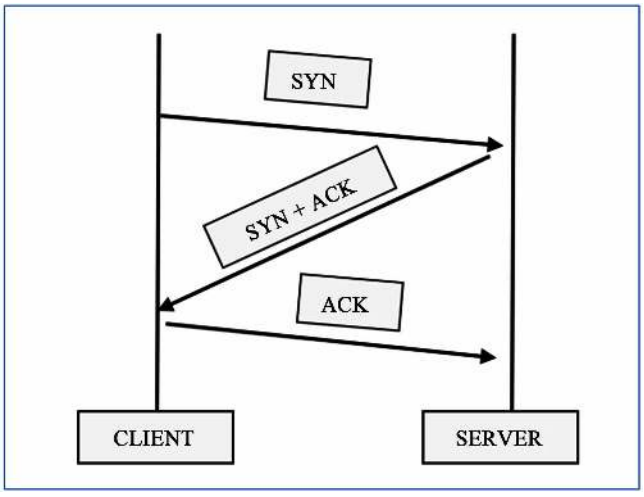

Figure 10. TCP connection (TCP 3-hands shake).

7. Result

After the simulation was developed, the traffic generator was ready for generation of normal and attack data. We were running the generator for a long time to get big data so the data will be increased as much we want based on running the generator. Table 3 summarizes the results of the simulation. The features of the data can be used for developing an IDS and evaluation.

8. Conclusion and Future Work

There are unlimited tools used to simulate real DDoS attacks in an effort to create business recovery plans in the even to a DDoS attack. However, the choice of the tool to use is determined on the type of attack that one plans to undertake. Further, there exist a plethora of datasets for previous real attacks and simulation attacks. Each attack has its attack vector. The scholar is thus supposed to check the best tool to use for the simulation process. The results of a

Table 3. Dataset features.

No	Ex. Value	NAME	Explain
1	10.0.0.98	SRC ADD	Source IP Address
2	10.0.0.26	DES ADD	Destination IP Address
3	2664	PKT ID	Identify of Packet
4	1033	FROM NODE	Identify of Low Layer (if it is −1, unknown layer)
5	1018	TO NODE	Identify of High Layer (if it is −1, unknown layer)
6	17	PKT TYPE	Type of Packet (17: UDP, 6: TCP……)
7	614	PKT SIZE	Packet size (included data)
8	NULL	FLAGS	Flags (SYN, ACK, FIN…) of Packet. This is not used in UDP. It is only used in TCP
9	NULL	FID	Identify of Transfer Layer (It is only used in TCP)
10	NULL	SEQ NUMBER	Sequence Number (It is only used in TCP)
11	4	NUMBER OF PKT	Number of Received Packet
12	3269	NUMBER OF BYTE	Number of Received Bytes
13	encap	NODE NAME FROM	Name of Low Layer
14	ip	NODE NAME TO	Name of High Layer
15	1	PKT IN	Input Packet or not (1: INPUT, 0: NOT)
16	0	PKTOUT	Output Packet or not (1: OUTPUT, 0: NOT)
17	0	PKTR	Routing Packet or not (1: ROUTING, 0: NOT)
18	0	PKT DELAY NODE	Delay is occurred at this host? (1: YES, 0: NO)
19	190.292	PKTRATE	Rate for packet receive (number of received packet per second)
20	155,516	BYTE RATE	Rate for bytes receive (number of received bytes per second)
21	817.25	PKT AVG SIZE	Average received packet size ($=\frac{\text{Total Received Bytes}}{\text{Number of Received Packet}}$)
22	1	UTILIZATION	This packet is used? (1: YES, 0: NO)

simulation can then determine the effectiveness of a process so as to redefine it or chose a separate tool for the same purpose. This paper has expounded on the existing tools and attack types as well as consolidated and robust list of dataset types and sizes. DDoS attack is one of the greatest menaces of cyber crime. In addition, traffic generator has been developed as one of the good techniques to develop the effective IDS against DDoS. In future, An IDS Intrusion Detection Systems will be developed and tested using the DDoS traffic generator for detection and evaluation.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Singh, J., Kumar, K., Sachdeva, M. and Sidhu, N. (2012) DDoS Attack's Simulation Using Legitimate and Attack Real Data Sets. *International Journal of Scientific & Engineering Research*, **3**, No. 6.

[2] Kotenko, I. and Ulanov, A. (2006) Simulation of Internet DDoS Attacks and Defense. *International Science of Journal Computing*, **4176**, 327-342. https://doi.org/10.1007/11836810_24

[3] Mukkavilli, S.K., Shetty, S. and Hong, L. (2016) Generation of Labelled Datasets to Quantify the Impact of Security Threats to Cloud Data Centers. *Journal of Information Security*, **7**, 172-184. https://doi.org/10.4236/jis.2016.73013

[4] Wagholi, P. (2014) Detection of DDoS Attacks Based on Network Traffic Prediction and Chaos Theory.

[5] Wang, J., *et al.* (2011) Advanced DDoS Attacks Traffic Simulation with a Test Center Platform. *International Journal for Information Security Research*, **1**, 168.

[6] Mukkavilli, S.K., Shetty, S. and Hong, L. (2016) Generation of Labelled Datasets to Quantify the Impact of Security Threats to Cloud Data Centers. *Journal of Information Security*, **7**, 172-184. https://doi.org/10.4236/jis.2016.73013

[7] Csubak, D., Szuks, K., Voros, P. and Kiss, A. (2016) Big Data Testbed for Network Attack Detection. *Acta Polytechnica Hungarica*, **13**, 47-57.

[8] Ozgur, A. and Erdem, H. (2016) A Review of KDD99 Dataset Usage in Intrusion Detection and Machine Learning between 2010 and 2015.

[9] The CAIDA UCSD "DDoS Attack 2007" Dataset. http://www.caida.org/data/passive/ddos-20070804_dataset.xml

[10] Dhanabal, L. and Shantharajah, S.P. (2015) A Study of NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms. *International Journal of Advanced Research in Computer and Communication Engineering*, **4**, 446-452.

[11] Shiravi, A., Shiravi, H., Tavallaee, M. and Ghorbani, A.A. (2012) Toward Developing a Systematic Approach to Generate Benchmark Datasets for Intrusion Detection. *Computers & Security*, **31**, 357-374.

[12] University of Southern California-Information Sciences Institute. (2012) DoS_80-20110715 (07/15/2011 to 07/15/2011) [Data Set]. IMPACT.

[13] Umarani, S. and Sharmila, D. (2014) Predicting Application Layer DDOS Attacks

Using Machine Language Algorithms. *International Journal of Computer and Systems Engineering*, **8**, 1912-1917.

[14] Alkasassbeh, M., Hassanat, A.B.A., Al-Naymat, G. and Almseidin, M. (2016) Detecting Distributed Denial of Service Attacks Using Data Mining Techniques. *International Journal of Advanced Science Applications*, **7**, 436-445.

[15] Bhuyan, M.H., Kashyap, H.J., Bhattacharyya, D.K. and Kalita, J.K. (2012) Detecting Distributed Denial of Service Attacks: Methods, Tools and Future Direction. *The Computer Journal*, **57**, 551.

[16] Berkes, J. (2017) Simulating DDoS Attacks with Ddosflowgen. Network Security. https://galois.com/blog/2017/04/simulating-ddos-attacks-ddosflowgen/

[17] Jonsson, V. (2009) HttpTools: A Toolkit for Simulation of Web Hosts in OMNeT++. *Proceedings of the Second International ICST Conference on Simulation Tools and Techniques*, Rome, 2 March 2009, 70. https://doi.org/10.4108/ICST.SIMUTOOLS2009.5589

[18] Shinde, P. and Parvat, T.J. (2016) DDoS Attack Analyzer: Using JPCAP and WinCap. *Procedia Computer Science*, **79**, 781-784. https://doi.org/10.1016/j.procs.2016.03.103