Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy

Iman Sharafaldin, Arash Habibi Lashkari, Saqib Hakak and Ali A. Ghorbani
(isharafa; a.habibi.l; saqib.hakak; ghorbani)@unb.ca
Canadian Institute for Cybersecurity (CIC), Faculty of Computer Science,
University of New Brunswick (UNB), Fredericton, NB, Canada

***Abstract*—Distributed Denial of Service (DDoS) attack is a menace to network security that aims at exhausting the target networks with malicious traffic. Although many statistical methods have been designed for DDoS attack detection, designing a real-time detector with low computational overhead is still one of the main concerns. On the other hand, the evaluation of new detection algorithms and techniques heavily relies on the existence of well-designed datasets. In this paper, first, we review the existing datasets comprehensively and propose a new taxonomy for DDoS attacks. Secondly, we generate a new dataset, namely CICDDoS2019, which remedies all current shortcomings. Thirdly, using the generated dataset, we propose a new detection and family classificaiton approach based on a set of network flow features. Finally, we provide the most important feature sets to detect different types of DDoS attacks with their corresponding weights.**

***Index Terms*—DDoS, IDS, DDoS Dataset, DDoS taxonomy, Network Traffic**

I. INTRODUCTION

Internet security is one of the most important challenges, especially when the demand of IT services increases daily. Among the many existing threats, Distributed Denial of Service (DDoS) attack is a relatively simple but very powerful technique to attack intranet and Internet resources. Usually, in this attack, the legitimate users are deprived of using web-based services by a large number of compromised machines. DDoS attacks can be implemented in network, transport and application layers using different protocols, such as TCP, UDP, ICMP and HTTP.

Many researchers have been working with different techniques, such as Machine Learning (ML), knowledge-based, and statistical to propose detection and defense mechanisms to combat the problem. On the one hand, each proposed method has different problems and shortcomings. For example, statistical methods are not able to determine with certainty the normal network packet distribution. ML techniques are good as they do not have any prior known data distribution, but defining the best feature-set is one of the main concerns for them [1].

On the other hand, since 2007, Subbulkashmi et al. [2], Prasad and Rao [3], CAIDA UCSD [4], DARPA 2000 [5], Brown et al. [6], Singh and De [7], Yu et al. [8], and shiravi et al. [9] tried to develop DDoS dataset. But due to many shortcomings and problems, such as incomplete traffic, anaonymized data, and out-dated attack scenarios, still researchers struggle to find comprehensive and valid datasets to test and evaluate their proposed detection and defense models. So, having a suitable dataset is a significant challenge itself [2].

Based on these two main concerns, our contributions in this paper are twofold. Firstly, we analyze the existing datasets to find their main shortcoming and limitations. Then, we present our approach for generating a new DDoS dataset called CICDDoS2019, which remedies the shortcomings and limitations of previous datasets. The dataset is completely labelled with 80 network traffic features have been extracted and calculated for all benign and denial of service flows by using the CICFlowMeter software that is publicly available on the Canadian Institute for Cybersecurity website [10]. Secondly, the paper analyzes the generated dataset to propose the best feature sets to detect different types of DDoS attacks, including reflective DDoS (such as DNS, LDAP, MSSQL, and TFTP), UDP, UDP-Lag and SYN. Also, we build our models to capture the patterns by training data using four common machine learning algorithms, namely ID3, random forest, Naïve Bayes, and logistic regression. We test them using the testing component. The remaining part of this paper is organized as follows: Section II explanes the available datasets, Section III presents our proposed taxonomy, Section IV describes the experiments, Section V reports the dataset, Section VI illustrates the analysis, and Section VII is the conclusion.

II. AVAILABLE DATASETS

In this section, we evaluate several publicly available DDoS attack datasets spanning 2007 to 2018 and explain the need for a comprehensive and reliable dataset to test and validate DDoS attack detection systems.

The CAIDA "DDoS Attack 2007" Dataset [4] contains an hour of traffic traces. They are in the pcap format, and detail attack traffic to the victim, as well as responses to the attack from the victim. The traces are anonymized using CrytoPAn prefic-preserving anonymization using a single key. The payload has been removed from all packages. They note that tracebacks are difficult to gather organically due to the ease of IP spoofing, the stateless nature of IP routing, link layer or MAC address spoofing and modern attack tools, featuring easy implementation of intelligent attack techniques.

In [3] a survey of different types of attacks and techniques of

978-1-7281-1576-4/19/$31.00 ©2019 IEEE

DDoS attacks and their countermeasures is conducted. They outline the two essential types of DDoS attacks as: vulnerability attacks, where attackers send malformed packets to confuse a protocol or an application; and flooding attacks, where either network, transport-level, or application level flooding interrupts a legitimate user's connectivity or services. Many defense techniques analyze both with respective advantages and drawbacks, with varying deployment locations. Traceback mechanisms are also surveyed, including the IP traceback for flooding attacks on Internet Threat Monitors using Honeypots. This method balances comparatively low overhead and no direct server damage, with processing delays and costs. They conclude that a practical defense for a real-time network is difficult to design, and perfect detection is not possible. Various performance parameters need to be considered and balanced.

DARPA dataset [5] consists of three unique datasets each produced as per consensus from the Wisconsin Re-Think meeting and the July 2000 Hawaii PI meeting. The first dataset LLDOS 1.0, includes a DDoS attack run by a novice attacker against a naive defender. The second dataset, LLDOS 2.0.2, includes a DDoS attack run by a more stealthy attacker, although still a novice, against a naive defender. The third dataset is a Windows NT Attack Data Set, which includes NT auditing of one day's traffic and attack impinging on the NT machine. Although this is a universal dataset, it is produced by a naive attacker, and does not explore the techniques of experienced attackers.

Brown et. al. [6] analyze the 1999 synthetic database commissioned by Lincoln Laboratory and DARPA to serve as a benchmark for evaluation of intrusion detection systems, and determined that the dataset has little bearing on real world attacks. The authors developed NetADHICT, a tool for understanding the structure of network traffic.

Singh et. al. [7] focuse on application layer DoS and DDoS attacks. By studying and using application layer attack tools such RUDY and slowloris, they were able to extract parameters from packet captures that serve as flags for DDoS attacks. They use a small number of items (79) for calculating the confusion matrix. They use Weka 3.6 for the classification of the dataset and computing the threshold. They determine that it is not possible to achieve a complete defense against these attacks in a single stage. They also suggest using a blacklist of IP addresses to prevent DDoS attacks. They propose looking at larger scale attacks with more classifiers in the future.

Subbulakshmi et. al. [2] generates a DDoS dataset in a testbed of LAN connected systems, which collects 14 attributes from 10 types of the DDoS attack classes. They then use Enhanced Multi Class Support Vector Machines (EMCSVM) to compare their dataset with the Kddcup 99 dataset, which has only six types of attacks. The attacks are generated artificially using attack generation scripts, and include 10 types of flooding attacks. Again, as this is not real traffic, their results can only be as accurate as their simulation. They propose deriving more attributes from a larger number of attacks in the future.

Yu et. al. [8] define a flow correlation coefficient, which they use to detect flash crowd DDoS attacks. This type of defense is based on the principle that the flow standard deviation is lower for an attack than for legitimate traffic. Testing their method of traffic generated by the DDoS tool Mstream and a couple of legitimate captured flash crowds, as well as simulated traffic, they determined that they could detect flash crowd flooding attacks as well as distinguish them from legitimate flash crowds. This is based on botnets and attack tools available in 2012. They propose looking into the possibility of super botnets capable of having a one-to-one bot to legitimate user relationship when conducting mimicking attacks. Taking into account that the current conditions do not reflect those of the past, we must explore the attacker's actions since this study, and develop new methods of detection based on current attacker behavior.

Shiravi et. al. [9] advocate for a dynamic dataset to gain real-world insight into what types of attacks are being orchestrated on a specific network. They then go on to establish a set of guidelines to obtain a valid dataset in terms of realism, evaluation capabilities, total capture, completeness, and malicious activity. The goal is to establish profiles for malicious activities which can be used in future network intrusion detection systems. They created a testbed network consisting of 21 interconnected Windows workstations, three servers and devices running Tcpdump, Snort, QRadar, OSSIM, and Ntop. With this setup they were able to capture virtually all communications and in part of their dataset they covered DoS attacks.

However, none of the mentioned datasets have executed and captured modern reflective DDoS attacks such as NTP, NetBIOS, SSDP, UDP-Lag, and TFTP. Also, most of them anonymized the traffic and removed the payloads which is one of the important parts of analyzing network packets. To put it briefly, most of the above data lack the origin of the dataset (synthetic instead of the real world), complete traffic, attack diversity, data source heterogeneity, complete interaction, and complete capture.

III. DDoS Attacks Taxonomy

There are a number of survey studies that have proposed taxonomies with respect to DDoS attacks. Mirkovic and Reiher et. al. [11] presented taxonomies for classification of DDoS attacks and possible defense mechanism. The attacks were categorized as: automation, vulnerability, source address validity, attack rate dynamics, characterization, persistence of agents, victim, and impact on the victim. In automation-based methods, the attacker searches a vulnerable machine manually/automatically. The authors explored the classification for DDoS defense mechanism based on activity level, cooperation degree (performs defensive measures either alone or in cooperation with other entities in the Internet.), and deployment location).

Asosheh and Ramezani[12] proposed a taxonomy based on known potential attacks and categorized attacks based on eight features, namely architecture, degree of automation, impact, vulnerability, attack rate dynamics, scanning strategy, propagation strategy, and packet content. The authors also proposed a taxonomy for the defense mechanism and categorized defense mechanism strategies into two groups, i.e. prevention and detection. The authors claimed that the best strategy to prevent/detect DDoS attacks is by focusing on deployment from where an attack originated, i.e. target network (original source of attack) and intermediate network (secondary targets). The authors concluded the article by proposing a framework that can detect DDoS attacks automatically using a cluster-based algorithm such as a k-nearest neighbor. However, there are no experiments being conducted to validate the proposed classification.

Bhardwaj et al. [13] focused on DDoS attacks in the cloud computing paradigm. The authors surveyed the articles published from the year 2009 to 2015. The authors propose the a taxonomy for the various potential DDoS attacks. These are four categories: degree of automation, vulnerability, attack rate dynamics, and attack impact. Although similar classification has been proposed by [11], the difference in this article lies the analysis of parameters for effective DDoS detection. A few key DDoS detection parameters that have been identified include real-time response, throughput, request, response time and zero-day attack detection ability.

The work carried out by Masdari and Jalali [14] focused on an in-depth analysis of DDoS attacks in the cloud computing. The authors illustrate the major types of DDoS attacks by identifying the vulnerabilities that lead to these attacks and finally classified the DDoS attacks based on cloud components, i.e. virtual machines, cloud scheduler, hypervisor, web services, cloud customers, IaaS and SaaS-based attacks. The major DDoS attacks in cloud computing have been identified as bandwidth attacks, connectivity attacks, resource exhaustion, limitation exploitation, process disruption, data corruption, and physical disruption. The study concludes that the severity of DDoS attacks is greater on cloud computing due to more available resources compared to traditional networks.

In another study conducted by Singh et. al. [7], the authors have provided a comprehensive analysis of HTTP-GET flood DDoS attacks. The authors have carried out a systematic study that provides a brief overview of HTTP-GET flood attacks including its operation and attack strategies. The articles used for the systematic survey have been taken from six standard electronic databases which include ACM digital library, IEEE Xplore, ScienceDirect, Wiley, and Google scholar. The authors have categorized attack strategies into a high rate and low rate. High rate includes those kinds of attacks where bots utilize their full capacity to attack the victim while low rate includes attack with low request rates by bots. The high rate attacks are further classified into server load and target webpage attacks while low rate attacks are divided into symmetric and asymmetric attacks.

The primarily features of the above-mentioned works are shown in Table I. Although, all the mentioned studies have done a commendable work in proposing new taxonomies, but the scope of attacks has been yet limited. There is a need to identify new attacks and come up with new taxonomies. Hence, we have analyzed new attacks that can be carried out using TCP/UDP based protocols at the application layer and proposed a new taxonomy. The rest of this sub-section, has been explained the detailed taxonomy of DDoS attacks and illustrated in the Figure 1, in terms of reflection-based and exploitation-based attacks.

Reflection-based DDoS: Are those kind of attacks in which identity of the attacker remains hidden by utilizing legitimate third-party component. The packets are sent to reflector servers by attackers with source IP address set to target victim's IP address to overwhelm the victim with response packets.

These attacks can be carried out through application layer protocols using transport layer protocols, i.e. Transmission control protocol (TCP), User datagram protocol (UDP) or through a combination of both. As Figure 1 shows, in this category, TCP based attacks include MSSQL, SSDP while as UDP based attacks include CharGen, NTP and TFTP. There are certain attacks that can be carried out using either TCP or UDP like DNS, LDAP, NETBIOS, and SNMP [15], [16].

Exploitation-based attacks: Are those kinds of attacks in which the identity of the attacker remains hidden by utilizing legitimate third-party component. The packets are sent to reflector servers by attackers with the source IP address set to the target victim's IP address to overwhelm the victim with response packets. These attacks can also be carried out through application layer protocols using transport layer protocols e.g. TCP and UDP. TCP based exploitation attacks include SYN flood and UDP based attacks include UDP flood and UDP-Lag.

UDP flood attack is initiated on the remote host by sending a large number of UDP packets. These UDP packets are sent to random ports on the target machine at a very high rate. As a result, the available bandwidth of the network gets exhausted, system crashes and performance degrades. On the other hand, SYN flood also consumes server resources by exploiting TCP-three-way handshake. This attack is initiated by sending repeated SYN packets to the target machine until server crashes/malfunctions.

The UDP-Lag attack is that kind of attack that disrupts the connection between the client and the server. This attack is mostly used in online gaming where the players want to slow down/interrupt the movement of other players to outmaneuver them. This attack can be carried in two ways, i.e. using a hardware switch known as lag switch or by a software program that runs on the network and hogs the bandwidth of other users.

IV. EXPERIMENTS

To create a comprehensive testbed, we have designed and implemented two networks, namely Attack-Network and Victim-Network. The Victim-Network is a highly security infrastructure with firewall, router, switches, and several common operating systems along with an agent that provides the benign behaviors on each PC. The Attack-Network is a completely separated third party infrastructure that executes different

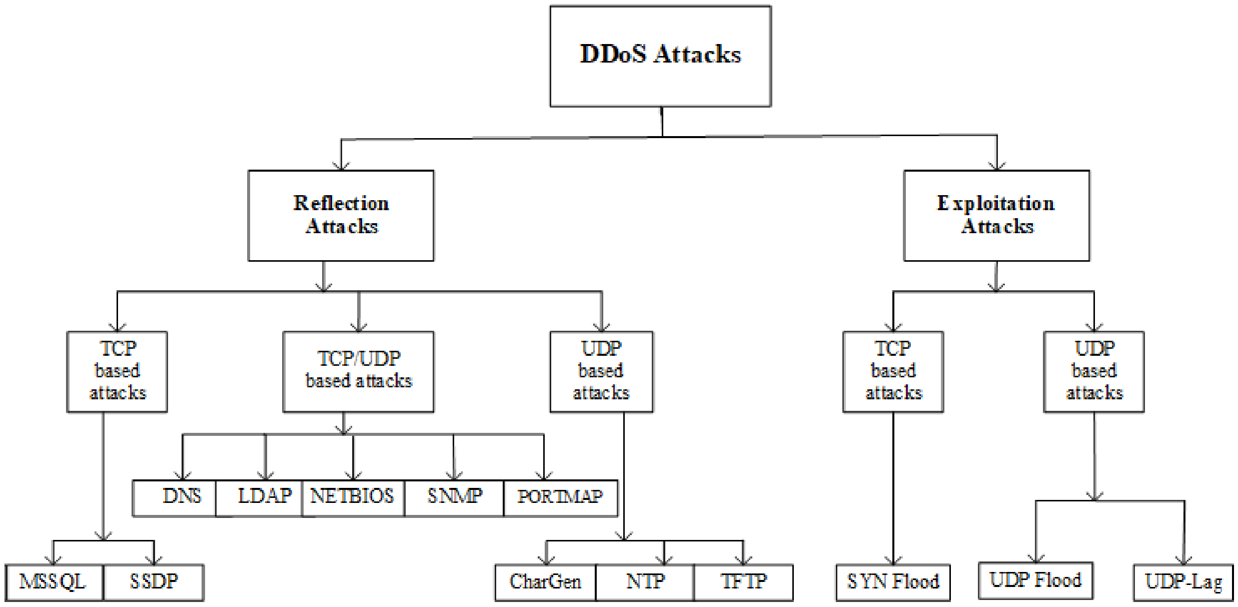

Figure 1: DDoS Attack Taxonomy

Table I: Primary features of the related works

Authors	OSI-Layer	Network-based Environment	Known attacks/ potential threats	Defense mechanism
[11]	✗	✗	✓	✓
[12]	✗	✗	✓	✓
[13]	✗	Cloud computing	✓	✓
[14]	Application, Network, Transport	Cloud computing	✓	✓
[17]	Application	✗	✓	✗

Table II: Victim-Network Operating Systems and IPs

	Machine	OS	IPs
Victim Network	Server	Ubuntu 16 (Web Server)	192.168.50.01 (Train) 192.168.50.04 (Test)
	Firewall	Fortinet	205.174.165.81
	PCs (Training day)	Win 7Pro Win Vista Win 8.1 Win 10 (Pro 32)	192.168.50.8 192.168.50.5 192.168.50.6 192.168.50.7
	PCs (Testing day)	Win 7Pro Win Vista Win 8.1 Win 10 (Pro 32)	192.168.50.9 192.168.50.6 192.168.50.7 192.168.50.8

types of DDoS attacks. The following sections discuss the infrastructure, benign profile agent and attack scenarios.

A. Testbed Architecture

As Figure 2 shows, the testbed consistes of two completely separated networks. Unlike the previous datasets, in the Victim-Network, we employ all commonly used and necessary equipment including router, firewall, switch, along with the different versions of the commonly used operating systems.

Table II shows the list of servers, firewall and workstations, with their operating systems and related public and private IPs in the training and testing days. A third party has executed the attack families in the training and testing days (Attack-Network). The Victim-Network consists of one server (Web server), one firewall, two switches and four PCs. Also, one port in the main switch of the Victim-Network has been configured as the mirror port and completely captures all send and receive traffic to the network.

B. Benign Profile Agent

Generating the realistic background traffic is one of the highest priorities of this work. For this dataset, we used our proposed B-Profile approach [18], which is responsible for profiling the abstract behavior of human interactions and generate a naturalistic benign background traffic. Our B-Profile for this dataset extracts the abstract behavior of 25 users based on the HTTP, HTTPS, FTP, SSH, and email protocols.
At first, it encapsulates network events produced by users with machine learning and statistical analysis techniques. The encapsulated features are distributions of packet sizes of a protocol, the number of packets per flow, certain patterns in the payload, the size of the payload, and request time distribution of protocols. Then, after deriving the B-Profiles from users, an agent which has been developed in Java is used to generate realistic benign events and simultaneously simulate B-Profile behavior on the Victim-Network for the predefined five protocols.

C. Attack Profiles

Since our proposed dataset is intended for testing DDoS attack detection techniques, it should cover a diverse set of DDoS attack techniques and scenarios. In this dataset, we created 11 different DDoS attack profiles listed in Table III. These

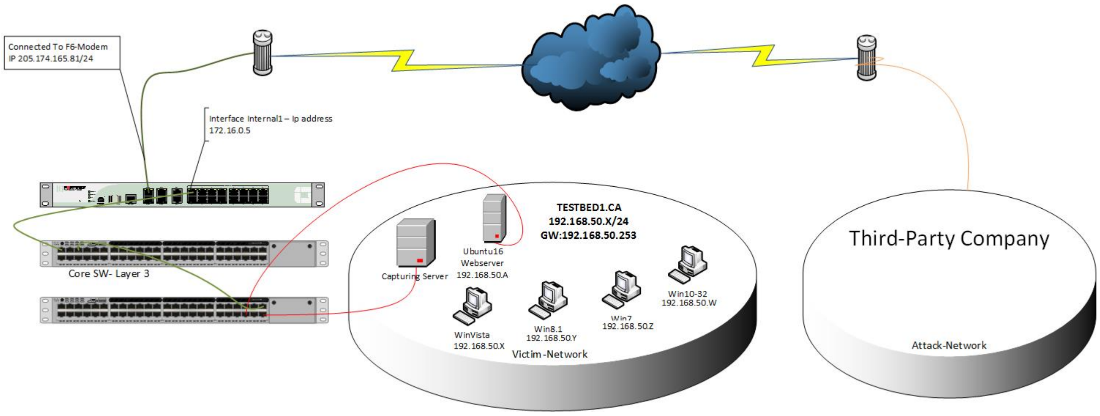

Figure 2: Testbed Architecture

Table III: Daily Label of Dataset

Days	Attacks	Attack times
Testing Set	PortMap	09:43 - 09:51
	NetBIOS	10:00 - 10:09
	LDAP	10:21 - 10:30
	MSSQL	10:33 - 10:42
	UDP	10:53 - 11:03
	UDP-Lag	11:14 - 11:24
	SYN	11:28 - 17:35
Training Set	NTP	10:35 - 10:45
	DNS	10:52 - 11:05
	LDAP	11:22 - 11:32
	MSSQL	11:36 - 11:45
	NetBIOS	11:50 - 12:00
	SNMP	12:12 - 12:23
	SSDP	12:27 - 12:37
	UDP	12:45 - 13:09
	UDP-Lag	13:11 - 13:15
	WebDDoS (ARME)	13:18 - 13:29
	SYN	13:29 - 13:34
	TFTP	13:35 - 17:15

attacks are based on proposed taxonomy (Section III) and executed them by using related tools and packages available by third party. As Figure 2 shows, we categorize these attacks into reflection-based and exploitation-based attacks from the transport and application layer.

V. DATASET

The capturing period for the training day on January 12th started at 10:30 and ended at 17:15, and for testing day on March 11th started at 09:40 and ended at 17:35. Attacks were subsequently executed during this period. As Table III shows, we executed 12 DDoS attacks includes NTP, DNS, LDAP, MSSQL, NetBIOS, SNMP, SSDP, UDP, UDP-Lag, WebDDoS, SYN and TFTP on training day and 7 attacks including PortScan, NetBIOS, LDAP, MSSQL, UDP, UDP-Lag and SYN in testing day. PortScan just has been executed in testing day and will be unknown for evaluating the proposed model. (Dataset is publicly available at http://www.unb.ca/cic/datasets/CICDDoS2019)

VI. ANALYSIS

At first we extract the 80 traffic features from the dataset using CICFlowMeter [19],[10]. Afterwards, to select the best detection feature set for each DDoS attack, we test these extracted features using RandomForestRegressor. Then, we examine the performance and accuracy of the selected features with four common machine learning algoeithms based on training and testing data.

For extracting the network traffic features, we used the CICFlowMeter [19], [10], which is a flow based feature extractor and can extract 80 features from a pcap file. The flow label in this application includes source IP, source Port, destination IP, destination port, protocol and time stamp. Then we labeled the generated flows based on the attack schedule (timestamp) that is explained in Section V. All 80 extracted features have been defined and explained in the CICFlowMeter webpage [19]. We used RandomForestRegressor class of scikit-learn [20]. First, we calculate the importance of each feature in the whole dataset, then we achieve the final result by multiplying the average standardized mean value of each feature split on each class, with the corresponding feature importance's value.

Table IV shows the list of the best selected features and corresponding weight of each section. Also, we depict Radviz diagrams for different kind of network traffic. Through Radviz, an N-dimensional dataset is projected into a 2D space wherein each dimension is represented in relation to the influence of all dimensions. We can discover interesting characteristics of different DDoS attacks from these diagrams. As we can see in Figure 3, 'packet Length Std' is one of the most influential features for benign traffic. One of the reasons is that we have more variation in the size of packets in benign traffic in comparison to different DDoS attacks, because DDoS attacks are conducted by automated tools and botnets and usually they produce fixed-size or similar packets.

Also, as shown in Figure 12 it is obvious that the two most influential features are 'ACK Flag Count' and 'Flow Duration', because this attack works by not responding to the server with the expected ACK code (it exploits a

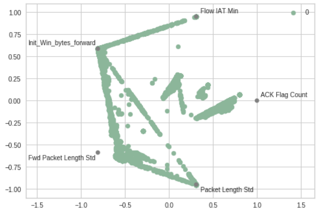

Figure 3: Bening

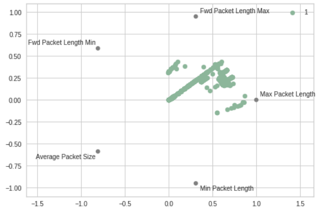

Figure 4: DrDoS-DNS

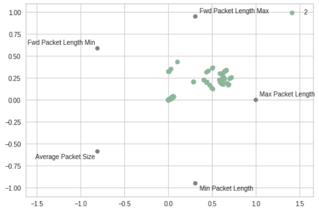

Figure 5: DrDoS-LDAP

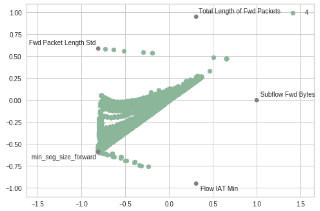

Figure 6: DrDoS-MSSQL

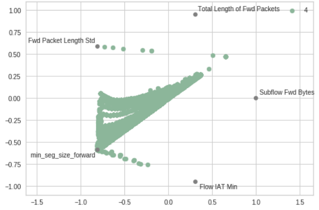

Figure 7: DrDoS-NTP

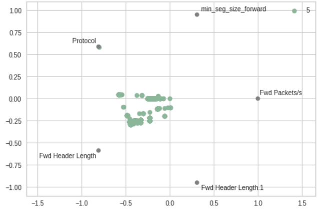

Figure 8: DrDoS-NetBIOS

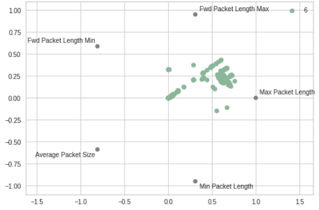

Figure 9: DrDoS-SNMP

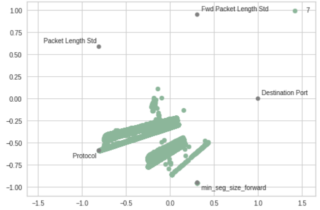

Figure 10: DrDoS-SSDP

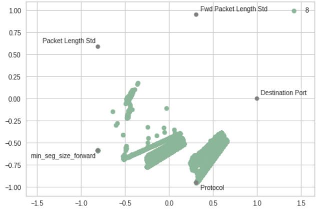

Figure 11: DrDoS-UDP

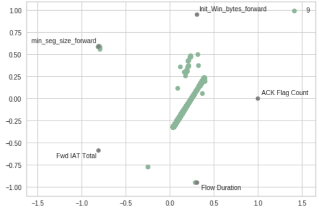

Figure 12: SYN

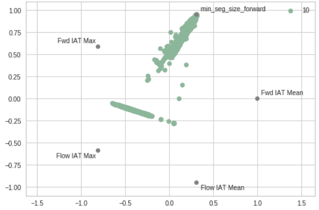

Figure 13: DrDoSTFTP

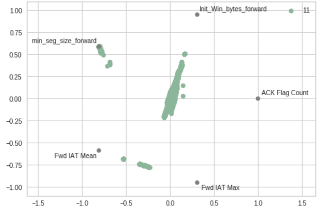

Figure 14: UDP-lag

TCP protocol weakness). Moreover, as shown in Figure 6, two main influential features are the 'Protocol' and 'Fwd Packets/s' which makes sense, because the attacker abuses the Microsoft SQL Server Resolution Protocol (MC-SQLR) and sends millions of packets to the victim. Furthermore, IAT related features are considered as influential features in many attacks such as DrDoS-TFTP, DrDoS-UDP-lag and DrDoS-NTP. One of the reasons is that many DDoS attacks show bursty behavior in sending packets to the victims, unlike benign traffic that usually does not show any bursty behavior. The bursty behavior affects the arrival rate, and so it affects IAT related features and this can be a reason why they are influential features for detecting DDoS attacks.

On the other hand, when TCP accepts data from a data stream, it first divides it into chunks and then adds a TCP header that finally will create a TCP segment. As we know, the resource battle between the victims' machine and the attacker's machine is one of the key features of DDoS attacks. It means, to make a successful DDoS attack, an attacker needs to send more packet than the victim can handle. Also, attackers use different types of packets, such as SYN or ICMP packets to send many malicious packets all of which are similar in size and small because of the low cost of computing resources but are not like the packets in a benign flow. So, the minimum segment size of the packets in a malicious flow would be less than the packets in a benign flow.

For the next step of our analysis, we have used four common machine learning algorithms namely ID3, Random Forest (RF), Naïve Bayes, and logistic regression along with three common machine learning evaluation metrics:

- Precision (Pr) or **Positive Predictive value:** It is the ratio of correctly classified attacks flows (TP), in front of all the classified flows (TP+FP).
- Recall (Rc) or **Sensitivity:** It is the ratio of correctly classified attack flows (TP), in front of all generated flows (TP+FN).
- F-Measure (F1): It is a harmonic combination of the precision and recall into a single measure.

$$Pr = \frac{TP}{TP+FP},\ Rc = \frac{TP}{TP+FN},\ F1 = \frac{2}{\frac{1}{Pr}+\frac{1}{Rc}}$$

ID3 is an algorithm designed by Ross Quinlan [21] in order to generate decision tree from a training dataset. The ID3 uses entropy (or information gain) concept in order to find the best attributes in order to split the dataset recursively and make the decision tree. Entropy is a measure of the amount of uncertainty in the set S:

$$H(X) = -\sum p(X) \log p(X) \qquad (1)$$

Table IV: Testing Dataset based on Training Part

Name	Feature	Weight	Mean
UDP-lag	ACK Flag Count	0.125438	0.86545908
	Init_Win_bytes_forward	0.002093	5061.324632
	min_seg_size_forward	0.000795	-3967113.958
	Fwd IAT Mean	0.000612	1109423.738
	Fwd IAT Max	0.000471	3310515.822
TFTP	Fwd IAT Mean	0.000207	541101.4798
	min_seg_size_forward	0.000198	-34648586.17
	Fwd IAT Max	0.000151	1562350.615
	Flow IAT Max	0.000129	1562493.187
	Flow IAT Mean	0.000124	540958.2437
WebDDoS	ACK Flag Count	0.043991	0.330296128
	Init_Win_bytes_forward	0.009357	21747.29613
	Fwd Packet Length Std	0.002881	62.69322463
	Packet Length Std	0.002068	108.2461488
	min_seg_size_forward	0.000872	32.00
DNS	Max Packet Length	1.139858	1378.802657
	Fwd Packet Length Max	0.127708	1378.773093
	Fwd Packet Length Min	0.007794	1378.522451
	Average Packet Size	0.005849	2067.363444
	Min Packet Length	0.003487	1378.521706
Benign	ACK Flag Count	0.020021	0.172801294
	Flow IAT Min	0.016769	11817.55752
	Init_Win_bytes_forward	0.003182	7560.598157
	Fwd Packet Length Std	0.001786	39.79290701
	Packet Length Std	0.001678	88.27355725
MSSQL	Fwd Packets/s	0.000204	1676967.356
	Protocol	4.60E-05	N/A
LDAP	Max Packet Length	1.278323	1463.73827
	Fwd Packet Length Max	0.143219	1463.728043
	Fwd Packet Length Min	0.008736	1463.717027
	Average Packet Size	0.006532	2194.906258
	Min Packet Length	0.003909	1463.716847
NetBIOS	Fwd Packets/s	0.000172	1580350.263
	min_seg_size_forward	7.20E-05	-41140208.12
	Protocol	4.60E-05	N/A
	Fwd Header Length	3.50E-05	-82335716.29
	Fwd Header Length.1	3.20E-05	-82335716.29
NTP	Subflow Fwd Bytes	0.106481	27903.20181
	Length of Fwd Packets	0.058022	27903.20181
	Fwd Packet Length Std	0.001081	25.03898396
	min_seg_size_forward	0.000707	-8471708.317
	Flow IAT Min	0.000573	438.699766
SSDP	Destination Port	0.000671	33266.62516
	Fwd Packet Length Std	0.000597	14.90294147
	Packet Length Std	0.000232	14.2504337
	Protocol	4.60E-05	N/A
	min_seg_size_forward	1.20E-05	-44212168.85
SNMP	Max Packet Length	1.152048	1386.280102
	Fwd Packet Length Max	0.129074	1386.258583
	Fwd Packet Length Min	0.007879	1386.226774
	Average Packet Size	0.005912	2079.140846
	Min Packet Length	0.003526	1386.226619
Syn	ACK Flag Count	0.145834	0.999478603
	Init_Win_bytes_forward	0.002432	5837.969261
	min_seg_size_forward	0.000872	20.00076092
	Fwd IAT Total	0.000571	8086250.716
	Flow Duration	0.000409	8086316.455
UDP	Destination Port	0.000699	33284.8418
	Fwd Packet Length Std	0.000615	15.28026139
	Packet Length Std	0.000239	14.61817097
	min_seg_size_forward	9.80E-05	-39820342.76
	Protocol	4.50E-05	N/A

Where $P(x)$ represents the proportion of the number of elements in class x to number of elements in the set S. Also, $H(S) = 0$ means that all elements in S have the same label. Moreover, in order to have a measure to find the difference in entropy from before to after the set S is split on an attribute A, we can use information gain $I(S, A)$ that can be calculated by the following formula:

$$I(S, A) = H(S) - H(S|A) \tag{2}$$

Where $H(t)$ is the entropy of subset t.

Random Forest (RF) [22] is a machine learning algorithm that combine two ideas of decision tree and ensemble learning. The forest contains many decision trees that use randomly picked data attributes as their input. The forest has a collection of trees with controlled variance. Finally, the result of a classification can be decided by majority voting or weighted voting. One of the advantages of random forest is that the variance of the model decreases as the number of trees in the forest increases, while the bias remains the same. Also, random forests has many other advantages such as low number of parameters and resistance to over-fitting.

Naïve Bayes is a probabilistic classifier based on Bayes Theorem with strong independence assumptions between features. We can decompose the conditional property by using Bayes' theorem as following:

$$P(C_k \mid X) = \frac{P(X \mid C_k)\, P(C_k)}{P(X)} \tag{3}$$

Where $X = (x_1, ..., x_n)$ represents a vector of n independent features and C_k represents each classes. Assuming that features are not correlated with each other is not a true assumption in many problems and it can conversely affects the accuracy of the classifier. The main advantage of Naïve Bayes is that it is an online algorithm and its training can be completed in linear time.

Multinomial Logistic Regression is classification method that uses the main idea of logistic regression to classify multiclass problems. Logistic regression is a predictive analysis like other regression analyses. Logistic regression can describe data and explain the relationship between features and classes.

Table V shows the performance examination results in terms of the weighted average of our evaluation metrics for the four selected common machine learning algorithms derived from the generated dataset. We used five-fold cross validation for our experiments. Based on our experiments ID3 took few minutes to be trained and classify the the testing set. Random forest with 100 trees took more than 15 hours for the same process. Also, multinomial logistic regression took more than 2 days to be trained and classify the testing test.

In addition, according to the weighted average of the three evaluation metrics (Pr, Rc, F1), the highest accuracy belongs to random forest and ID3 algorithms. Also, in terms of recall ID3 won the first place by far. Logistic regression achieves the worst result overall. Considering the execution time and the evaluation metrics ID3 is the best algorithm with the shortest execution time and highest accuracy.

Table V: The Performance Examination Results

Algorithm	Pr	Rc	F1
ID3	0.78	0.65	0.69
RF	0.77	0.56	0.62
Naïve Bayes	0.41	0.11	0.05
Logistic regression	0.25	0.02	0.04

VII. CONCLUSION

The main contribution of this paper is a new dataset for evaluation of IDS algorithms and systems on DDoS attacks namely CICDDoS2019. In this paper, we studied on the several DDoS attack categories and families to propose a new DDoS taxonomy for the application layer. Also, we have reviewed the most popular available DDoS datasets and listed the common shortcomings and weaknesses. In response to these shortcomings and weaknesses, we generated a new dataset including 11 DDoS attacks, namely CICDDoS2019 for evaluation of IDS/IPS algorithms and systems. Also, we provided the most important features for detecting different DDoS attacks. Furthermore, based on the 12 RadViz diagrams of the most influential features for each type of network traffic we provide a detailed analysis for each of them.

REFERENCES

[1] S. Jin and D. S. Yeung, “A covariance analysis model for ddos attack detection,” in *2004 IEEE International Conference on Communications*, vol. 4, pp. 1882–1886 Vol.4, 2004.

[2] T. Subbulakshmi, K. BalaKrishnan, S. M. Shalinie, D. AnandKumar, V. GanapathiSubramanian, and K. Kannathal, “Detection of ddos attacks using enhanced support vector machines with real time generated dataset,” in *Third International Conference on Advanced Computing*, pp. 17–22, 2011.

[3] A. M. R. K.Munivara Prasad and K. Rao, “Dos and ddos attacks: Defense, detection and traceback mechanisms - a survey,” *Global Journal of Computer Science and Technology*, vol. 14, 2014.

[4] *The CAIDA UCSD ”DDoS Attack 2007” Dataset*, accessed by Jan 2018. http://www.caida.org/data/passive/ddos-20070804_dataset.xml.

[5] *DARPA 2000 Intrustion Detection Scenario Specific Data Sets*, accessed by Jan 2018. https://www.ll.mit.edu/r-d/datasets/2000-darpa-intrusion-detection-scenario-specific-data-sets.

[6] A. H. Carson Brown, Alex Cowperthwaite and A. Somayaji, “Analysis of the 1999 darpa/lincoln laboratory ids evaluation data with netadhict,” in *Proceedings of the Second IEEE International Conference on Computational Intelligence for Security and Defense Applications*, CISDA’09, 2009.

[7] K. J. Singh and T. De, “An approach of ddos attack detection using classifiers,” *Emerging Research in Computing, Information, Communication and Applications*, 2015.

[8] S. Yu, W. Zhou, W. Jia, S. Guo, Y. Xiang, and F. Tang, “Discriminating ddos attacks from flash crowds using flow correlation coefficient,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 6, pp. 1073–1080, 2012.

[9] M. T. Ali Shiravi, Hadi Shiravi and A. A.Ghorbani, “Toward developing a systematic approach to generate benchmark datasets for intrusion detection,” *Computers and Security*, vol. 31, pp. 357–374, 2012.

[10] M. M. Arash Habibi Lashkari, Gerard Draper Gil and A. Ghorbani, “Characterization of tor traffic using time based features,” in *In Proceedings of the 3rd International Conference on Information Systems Security and Privacy (ICISSP)*, pp. 253–262, 2017.

[11] J. Mirkovic and P. Reiher, “A taxonomy of ddos attack and ddos defense mechanisms,” *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, pp. 39–53, 2004.

[12] A. Asosheh and N. Ramezani, “A comprehensive taxonomy of ddos attacks and defense mechanism applying in a smart classification,” *WSEAS Transactions on Computers*, vol. 7, no. 4, pp. 281–290, 2008.

[13] A. Bhardwaj, G. Subrahmanyam, V. Avasthi, H. Sastry, and S. Goundar, “Ddos attacks, new ddos taxonomy and mitigation solutions—a survey,” in *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, pp. 793–798, IEEE, 2016.

[14] M. Masdari and M. Jalali, “A survey and taxonomy of dos attacks in cloud computing,” *Security and Communication Networks*, vol. 9, no. 16, pp. 3724–3751, 2016.

[15] “Request for comments: 1001 (rfc1001),” in *PROTOCOL STANDARD FOR A NetBIOS SERVICE ON A TCP/UDP TRANSPORT: CONCEPTS AND METHODS*, 1987.

[16] “Request for comments: 7766 (rfc7766),” in *DNS Transport over TCP - Implementation Requirements*, 2016.

[17] K. Singh, P. Singh, and K. Kumar, “Application layer http-get flood ddos attacks: Research landscape and challenges,” *Computers & security*, vol. 65, pp. 344–372, 2017.

[18] A. H. L. Iman Sharafaldin, Amirhossein Gharib and A. Ghorbani, “Towards a reliable intrusion detection benchmark dataset,” *Software Networking*, pp. 177–200, 2017.

[19] *CICFlowMeter*, 2017. https://github.com/ISCX/CICFlowMeter.

[20] A. G. F. Pedregosa, G. Varoquaux and E. Duchesnay, *Scikit-learn: Machine learning in Python*, 2011.

[21] J. R. Quinlan, “Induction of decision trees,” *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[22] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.