Efficient and Secure Statistical DDoS Detection Scheme

Hussein Majed[1], Hassan N. Noura[1,2], Ola Salman[2], Mohammad Malli[1] and Ali Chehab[2]
[1]*Arab Open University, Department of Computer Sciences, Beirut, Lebanon*
[2]*American University of Beirut, Department of Electrical and Computer Engineering, Lebanon*

Keywords: DDoS, Intrusion Detection, Traffic Aggregation, Network Security.

Abstract: One of the hardest challenges in cybersecurity is the detection and prevention of Distributed Denial of Service (DDoS) attacks. In this paper, a lightweight statistical approach for DDoS detection is presented, in addition to preventive and corrective countermeasures. The proposed solution is designed to be applied at the Internet Service Provider (ISP) level. Based on aggregated NetFlow statistics, the proposed solution relies on the Z-score and co-variance measures to detect DDoS traffic as a deviation from normal traffic. The implementation results show a high detection rate (up to 100%) for 30 seconds time slot.

1 INTRODUCTION

Cyber threats are becoming more sophisticated and more severe than ever before, taking advantage of existing security vulnerabilities such as the enormous network scale, the heterogeneity of the adopted communication protocols and connected devices, and the wide deployment of applications. Organizations are relying on different security techniques to protect their systems and networks against cyber attacks to minimize the associated financial losses.

Distributed Denial of Service (DDoS) attacks are among the most disruptive threats in the cyber world. DDoS is a variant of DoS attacks, where the attacker attempts to make a device or network resources unavailable to its legitimate users, by temporarily or permanently disrupting its services. DoS attacks can be divided into two types (Zargar et al., 2013; Gulihar and Gupta, 2020): the first type is when an attacker mystifies the protocol or application running on the victim's machine by sending malicious packets (vulnerability attack). The second one is when the attacker sends huge traffic to take up all the resources of the victim's machine, making it unable to interact with legitimate users (bandwidth/flooding attack). The difference between DoS and DDoS attacks is that the former uses only one single entity as a source, while the latter uses a distributed set of source devices. Also, DDoS attacks can be launched against one or several victims (Douligeris and Mitrokotsa, 2004). The distributed nature of DDoS attack makes them more devastating compared to DoS attacks.

In February 2000, a high-profile DDoS attack was presented by targeting popular websites such as ebay, Yahoo, Buy.com, Amazon.com, Excite.com, and Cable News Network (CNN) (Radware, 2017). Unfortunately, the number of DDoS incidents is continuously growing. Recently, DDoS attackers have been using techniques that are very hard to detect, and the attack size and frequency are rising rapidly (Zargar et al., 2013). A recent major attack was recorded in 2016, reaching approximately 1 Tbps and targeting a hosting cloud computing company by using more than 152,000 Internet of Things (IoT) devices (Bertino and Islam, 2017; Kolias et al., 2017; De Donno et al., 2018)). Because of the drastic impacts of DDoS attacks, researchers are continuously trying to build efficient, robust and comprehensive DDoS detection and mitigation solutions (Bhatia et al., 2018). Defending against this type of attacks is crucial due to its negative impact on online services availability, especially that almost every sector is becoming connected to the Internet, including governmental, health, banking, etc.

Current DDoS detection solutions are based on statistical or machine learning approaches. However, these solutions suffer from performance and/or accuracy issues, as explained in Section 2. Hence, there is a pressing need for a reliable and efficient solution to respond better to these security challenges. This work focuses on the early detection of DDoS attacks at the ISP level. The main goal is to achieve a high level of efficiency and speed with minimum computational complexity. Thus, we present a fast and reliable solution based on the NetFlow statistics at edge routers of an ISP network.

The rest of the paper is organized as follows. Section 2 reviews the different approaches used in DDoS

Majed, H., Noura, H., Salman, O., Malli, M. and Chehab, A.
Efficient and Secure Statistical DDoS Detection Scheme.
DOI: 10.5220/0009873801530161
In *Proceedings of the 17th International Joint Conference on e-Business and Telecommunications (ICETE 2020) - DCNET, OPTICS, SIGMAP and WINSYS*, pages 153-161
ISBN: 978-989-758-445-9

detection. Section 3 presents the proposed solution for DDoS detection including data collection, data aggregation and features extraction, attack detection, and attack mitigation. Section 4 presents the experimental setup and results. Finally, we conclude the paper in Section 5.

2 RELATED WORK

The various sub-classes of DDoS detection solutions are described in this section, mainly the ones based on the statistical and machine learning approaches.

2.1 Statistical Approach

In the statistical approach for DDoS detection, traffic statistics are extracted, and these include the number of destination/source Internet Protocol (IP) addresses, Transmission Control Protocol (TCP) flags, packet sizes, flow rate, and others, to identify any abnormal traffic behavior.

In (Hofstede et al., 2013), a lightweight intrusion detection solution was presented using three metrics, the number of flows per second, the number of bytes per flow, and the number of packets per flow. Two algorithms were applied on a dataset captured on the backbone link of the Czech national research and education network CESNET. The first algorithm is the Exponentially Weighted Moving Average (EWMA) for mean calculation, extended by thresholds and cumulative sum (CUSUM), and the second algorithm is a combination of the first algorithm with seasonality modeling. Both algorithms rely on EWMA for calculating the mean over the past values. The results showed that a response time of 5 seconds can be achieved with a reduction in detection delays to about 10% compared to the Netflow Sensor (NfSen) tool, which uses time slots of 5 minutes, resulting into an average delay of 150 seconds. In fact, the delay is calculated based on the difference between the time a packet is metered and the time flow data is made available to analysis. The distance-based statistical approach was introduced in (You et al., 2007) for DDoS detection using two techniques: the average distance estimation and the distance-based traffic separation. The DDoS attack detection is based on analyzing distance values and traffic rates by computing the Minimum Mean Square Error (MMSE) of the traffic rates from different distances. However, one limitation of the proposed method is that detection is unfeasible when final TTL values of all packets are equal because the distance values, based on TTL, will be the same for all packets. In (Girma et al., 2015), multiple statistical approaches such as co-variance matrix, Kendall's Tau, and entropy were evaluated and a hybrid method was proposed based on entropy and co-variance matrices. Finding the best threshold that separates normal and abnormal traffic is the hardest part for statistical-based DDoS detection. In (David and Thomas, 2015), David et al. proposed an adaptive threshold algorithm to update the threshold based on traffic conditions. The authors used fast entropy to calculate the mean and standard deviation of the flow count during a particular time interval, and the difference between the mean value and the fast entropy, to decide if a DDoS attack is taking place.

While most of the proposed DDoS detection solutions rely on the flow source IP address, attackers can spoof these addresses, yet they cannot control the hop counts. This key point motivated the authors in (Shamsolmoali and Zareapoor, 2014) to propose a model based on hop counts and the frequency count of each packet. The model has the ability of quickly detecting DDoS attacks with minimum storage overhead; it showed a 97% accuracy with minimum false alarms.

Passively monitoring abrupt changes, in network traffic fractal parameters, was also used to detect abnormal changes in network traffic. In (Xia et al., 2012), an auto-regressive system was used to estimate the Fractal dimension D and Hurst parameter H of normal traffic. The proposed method relies on the maximum likelihood estimate-based detection approach in order to determine the change point of parameters D and H. The changes in these two points indicate the occurrence of a DDoS flood attack.

Other methods rely on the Domain Name Servers Block List (DNSBL) to check if the traffic originated from sources with bad reputations. This approach consists of studying the flow source in a network to detect the Weird Host List (WHL) (Sawaya et al., 2011). Analysing the destination port for each entity in the WHL aims to check if the port is receiving many connections from unknown hosts. The researchers used DNSBLs to check if the entities in the WHL are blacklisted. The method showed 90% detection rate, but it needs synchronization of DNSBLs.

2.2 Machine Learning Approach

Recently, machine learning techniques have been adopted for the detection of network attacks. The machine learning models are used to extract patterns from the network traffic, and they can be used for clustering and detecting abnormalities. In (Zekri et al., 2017), Zekri et al. proposed a DDoS detection system based on the C4.5 algorithm for classification,

and Naive Bayes was applied for anomaly detection, whereas Snort was used for signature-based attack detection. Four classes were considered in the proposed system: normal, TCP SYN attack, User Datagram Protocol (UDP) attack, and Internet Control Message Protocol (ICMP) attack. High detection and efficiency rates (up to 98%) were achieved. In (Alkasassbeh et al., 2016), Alkasassbeh at al. compared different machine learning methods, Multi Layer Perceptron (MLP), Random Forest (RF), and Naïve Bayes (NB), on a dataset including HTTP flood, SQL Injection DDoS (SIDDOS), UDP flood, and smurf attacks. The obtained accuracy results were 98.63%, 98.02% and 96.91% for MLP, RF and NB, respectively. However, RF and NB failed to show good rates for the Smurf class, while MLP achieved a high precision rate. In (Hou et al., 2018), Hou et al. applied RF on NetFlow for DDoS detection. The results had 99% true positives and 0.5% false positives. Deep learning was also applied for DDoS detection. An approach called DeepDefense was proposed to detect DDoS attacks in (Yuan et al., 2017). The method is based on Recurrent Neural Network (RNN) by feeding historical information to the RNN in the aim of recognizing repeated patterns and locating them. The method was evaluated based on a dataset recorded for 7 days and containing DDoS attacks in 2 days out of the 7. The experimental results demonstrated that DeepDefense reduces the error rate by 39.69% for the first dataset and from 7.517% to 2.103% for the second, compared with traditional machine learning methods. For abnormality detection, the Modified Global K-means algorithm (MGKM) was considered as an incremental clustering algorithm to identify clusters in (Zi et al., 2010). The linear correlation coefficient for feature ranking was used to recalculate the clusters. The proposed method proved to be effective and adaptive in detecting the different phases of DDoS. Entropy-based approaches were also considered for DDoS detection. In (Singh et al., 2018), Singh et al. considered normalized router entropy, which is the overall probability distribution of the captured flow for some time window, the router entropy for a particular flow (calculated by combining the entropy of distinct flows during the time window), packet rate, and entropy rate to measure the growth rate of the entropy of any random process, to define the threshold of the attack detection. For a series of n random variables, they calculated the entropy rate (ER) of a stochastic process (xi). The threshold values used in the algorithm were evaluated offline by considering the malicious and legitimate traffic flows. Another hybrid approach using entropy, with Support Vector Machine (SVM), was proposed in (Hu et al., 2017) to measure network features. The SVM classifier was applied to identify network anomalies, and the experimental results showed minimal overhead and high accuracy. Similarly, in (Yun Liu et al., 2010), SVM and Traffic Feature Conditional Entropy (TFCE) were considered for DDoS detection. The proposed solution is based on analyzing the multiple-to-one relation between the attacked IP and the source IP addresses. SVM was applied to identify DDoS attacks based on the output of the entropy extracted by TFCE. The experimental results showed an accuracy of 93% for DDoS detection, within a time window of 5 seconds.

Detecting DDoS attacks is a complicated task due to the variety of attack types (TCP flooding attack, UDP flooding attack, ICMP flooding attack, etc.). Addressing this problem requires real traffic and up-to-date datasets before applying any approach for detection. A common limitation of the existing research works is the quality of datasets that have been used during the experiments. Most of these datasets are old or based on simulating attacks in virtual labs. Another issue that should be addressed is the speed of detection in the real world scenario because applying machine learning or statistical methods on current datasets will indicate the time taken for detection after collection, when the time taken for collecting traffic should be added as well. In this paper, a lightweight and high-speed statistical approach is proposed for DDoS detection. A real and new dataset was collected in an ISP network. NetFlow statistics are collected and aggregated to lessen the time overhead and complexity. Thus, the proposed method is easy to deploy, scalable, robust, and flexible. It is also efficient to detect well known and unknown ("Zero-day") attacks with low false alarm rate. It exhibits an adaptive technique for the definition of the corresponding rules and thresholds values. It incurs a low computational cost and resources compared to existing solutions.

3 PROPOSED DDoS DETECTION SCHEME

The proposed DDoS attack detection method consists of 4 main steps: data collection, data aggregation, DDoS detection, and mitigation (see Figure 1).

3.1 Data Collection

In this step, the incoming network traffic at the ISP is collected at regular intervals, such as one minute. This can be achieved by using a software installed at the edge router or by using an external monitoring de-

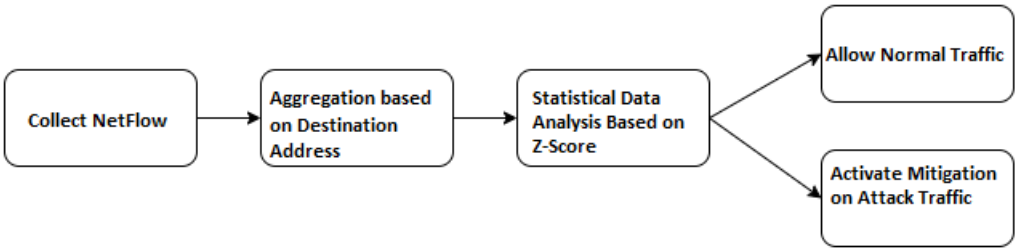


Figure 1: Proposed DDoS detection scheme.

Table 1: Table of notations.

Symbol	Definition
μ	Mean
σ	Standard deviation
CV_{SA}	The coefficient variation of the number of source addresses per destination IP
CV_{TCP}	The coefficient variation of a sequence of TCP flows
CV_{UDP}	The coefficient variation of a sequence of UDP flows
CV_{ICMP}	The coefficient variation of a sequence of ICMP flows
ZSC_{SA}	The Z-score of the number of flows per destination IP
ZSC_{TCP}	The Z-score of the number of TCP flows per destination IP
ZSC_{UDP}	The Z-score of the number of UDP flows per destination IP
$ZSC_{InputByte}$	The Z-score of the sum of incoming bytes per destination IP
$ZSC_{TimeDuration}$	The Z-score of the time duration per destination IP
ZSC_{ICMP}	The Z-score of the number of ICMP flows per destination IP
NIP	The number of unique destination IP

vice. The collected NetFlow statistics will be aggregated in the next step.

3.2 Data Aggregation and Features Extraction

At this stage, the collected NetFlow statistics are aggregated per destination IP. NetFlow gives information about the flows including: source IP address, destination IP address, source port number, destination port number, duration, number of packets, number of bytes, and IP protocol. The collected statistics at the first step are aggregated in a custom manner. As such, the proposed aggregation method calculates the following features for each destination address:

- The sum of received bytes
- The sum of received packets
- The count of source addresses who communicated with each destination address
- The sum of time duration
- The count of UDP Flows
- The count of TCP Flows
- The count of ICMP Flows
- The count of ”S” Flag
- The count of ”F” Flag
- The count of ”R” Flag
- The count of ”U” Flag
- The count of ”A” Flag
- The count of ”P” Flag

Then, the aggregated statistics are analyzed using the Z-score metric (see Algorithm 1. The basic Z-score formula for a sample x is:

$$z = \frac{x - \mu}{\sigma} \quad (1)$$

where μ and σ represent the mean and the standard deviation of the tested samples, respectively.

The Z-score of several features (number of TCP/UDP/ICMP flows per destination IP, sum of incoming packets per destination IP, sum of incoming bytes per destination IP, time duration per destination IP, count of source IP per destination IP) are computed for each address. These values are used as input to the next step, which can be seen as a form of non-reversible compression. The advantage of this step is to reduce the required processing complexity and storage overhead.

3.3 DDoS Detection

For each aggregated flow, the obtained Z-scores of three features are compared to the corresponding threshold values that are based on the coefficient of variation (CV). CV represents a statistical indicator for the dispersion of data points in a data set (series)

around the mean. It is the ratio of the standard deviation to the mean, and it is a useful statistical metric for comparing the degree of variation from one data series to another, even if the means are different. CV is computed as illustrated in the following equation:

$$CV = \frac{\sigma}{\mu} \quad (2)$$

Accordingly, a DDoS attack is detected if the obtained Z-score values of these features are greater or equal to the corresponding thresholds (see lines 14, 17 or 20 of Algorithm 1). These thresholds are set based on a training step (initial step), which depends on the ISP profile (traffic and network characteristics). Using statistical thresholds is not practical because they need to be updated manually, each time a bandwidth upgrade takes place. In addition, they might lead to false positives and false negatives. To make the proposed solution based on a dynamic threshold, the Z-score value of each feature is compared to its Coefficient of Variation (CV). In case of an attack, the attacked IP address will definitely receive a large number of packets from multiple sources within a time window. The attack will increase the mean value and standard deviation for each affected feature of the whole set, resulting into a high Coefficient of Variation for each affected feature. By calculating the Z-score of each feature for each data point, the data point of the attacked IP will have a high z-score value for each affected feature, indicating that it is relatively different. Since we have the CV value of each feature, we can compare it to the z-score of each data point; if the z-score is higher than the CV, it's considered an outlier compared to the rest of the flows.

Receiving a high number of packets or bandwidth from a high number of source IPs on a specific IP will definitely make the standard deviation and mean values high for each addressed feature. The Z-score is a well known method for the detection of outliers. Applying the Z-score on each value in the aggregated flow will clearly reveal the outlier, but each type of an attack has its own characteristics. TCP flood attacks are fast, from multiple sources, and share the same flag. UDP flood attacks rely on high incoming bytes, from multiple sources, and all the packets are destined to same IP. ICMP flood attacks try to overwhelm the targeted device ability to respond to the high number of requests and/or overload the network connection with bogus traffic, it is fast, and the traffic will be destined to the same IP from probably multiple sources. Thus, the following detection features are selected and used for each DDoS attack type:

- TCP: time duration, number of TCP packets, and number of Source IP.
- UDP: sum of incoming bits, number of UDP packets, and number of source IP.
- ICMP: sum of incoming bits, number of ICMP packets, and number of source IP.

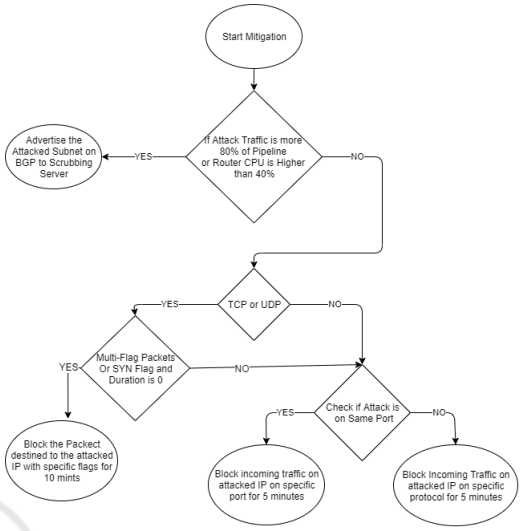


Figure 2: Mitigation process.

3.4 Mitigation Process

When a DDoS attack is detected, the prevention countermeasure is activated to mitigate it. The proposed mitigation process is illustrated in Figure 2, and it can be realized locally by blocking the incoming traffic for a certain period of time (depending on the DDoS attack scenario such as TCP or UDP), or by using the scrubbing server (high volume traffic case). Indeed, if the attack traffic ($\geq$ 80%) of the total traffic and the router CPU $\geq$ 40% of its total capacity, the attack cannot be mitigated by blocking the incoming traffic, but by advertising the attacked sub-net on the Border Gateway Protocol (BGP) to the scrubbing server.

4 EXPERIMENTAL SETUP AND RESULTS

In this section, we explain the implementation of the proposed DDoS detection and prevention scheme in detail.

4.1 NetFlow Collection

For collecting data, the setup was realized in an ISP network with full traffic visibility. NetFlow is a feature introduced by Cisco to enable in/out IP network traffic statistics collection at the router interface(s).

Algorithm 1: Proposed detection algorithm.

Input: Aggregated Flow AF
Output: R
procedure $R = \text{Detection}(AF)$
  $NIP \leftarrow NumberofRows(AF)$
  $CV_{SA} \leftarrow \frac{std(C_{SA})}{mean(C_{SA})}$
  $CV_{TCP} \leftarrow \frac{std(C_{TCP})}{mean(C_{TCP})}$
  $CV_{UDP} \leftarrow \frac{std(C_{UDP})}{mean(C_{UDP})}$
  $CV_{ICMP} \leftarrow \frac{std(C_{ICMP})}{mean(C_{ICMP})}$
  for $i \leftarrow 0$ to NIP **do**
    $ZSC_{SA} \leftarrow Zscore(C_{SA})$
    $ZSC_{TCP} \leftarrow Zscore(C_{TCP})$
    $ZSC_{UDP} \leftarrow Zscore(C_{UDP})$
    $ZSC_{ICMP} \leftarrow Zscore(C_{ICMP})$
    $ZSC_{InputByte} \leftarrow Zscore(C_{InputByte})$
    $ZSC_{TimeDuration} \leftarrow Zscore(C_{TimeDuration})$
    if $|ZSC_{SA}| \geq CV_{SA}$ && $|ZSC_{TCP}| \geq CV_{TCP}$ && $ZSC_{TimeDuration} \leq 0$ **then**
      TCP DDoS Attack Detected on DA
    end if
    if $|ZSC_{SA}| \geq CV_{SA}$ && $|ZSC_{UDP}| \geq CV_{UDP}$ && $|ZSC_{InputByte}| \geq CV_{InputByte}$ **then**
      UDP DDoS Attack Detected on DA
    end if
    if $|ZSC_{SA}| \geq CV_{SA}$ && $|ZSC_{ICMP}| \geq CV_{ICMP}$ && $|ZSC_{InputByte}| \geq CV_{InputByte}$ **then**
      ICMP DDoS Attack Detected on DA
    end if
  end for
end procedure

A virtual CentOS machine was created with vmware ESXi 6.0. This machine has 2 network interfaces, the first one is for NetFlow collection, and the second one is for traffic management. To isolate the communication between this machine and the edge routers, a separate VLAN was created. Two Cisco edge routers (ASR 9K) running IOS XR Release 7.0.1 were configured to send all incoming NetFlow traffic to the CentOS machine.

"nfcapd" was installed on the CentOS machine to capture the NetFlow information. Multiple instances of "nfcapd" were configured to listen on different ports and to write the captured flows, each minute, in a separate file, and for each router, in a separate folder.

nfcapd -w -D -l /flow_base_dir/router1 -p 23456
nfcapd -w -D -l /flow_base_dir/router1 -p 23456

NetFlow statistics of both routers were collected for one month. During this period, multiple DDoS attacks hit the ISP network. A time table was updated each time an attack occurred to be used later on to differentiate the benign traffic from the attack one. The attack usually saturates the international links of the ISP and increases the CPU usage on edge routers.

The "nfcapd" tool saves the files in raw format, and the "nfdump" tool is applied to read the input file, and to provide statistics, aggregation, searching, sorting, and conversion to another format such as CSV. Thus, the conversion option was used to convert the raw files to CSV files, which represent the dataset used for evaluating the proposed detection method.

4.2 Data Aggregation

The captured flows can be visualized using "nfdump" statistics feature by aggregating the traffic by destination IP and sorting the results by number of packets received for each destination IP. The results showed that the attacked IP always had the highest number of packets during the specified time duration (here one minute) (See Figure 3).

Examining the traffic destined to the IP with the highest number of received packets, the results showed that, for UDP traffic, this IP receives traffic from more than 40,000 different IPs on a specific port, during one minute. Repeating the same test on multiple files, containing different types of attacks, the out-

```
$ nfdump -r nfcapd.201910261112 -n 5 -A dstip -O packets
 Duration       Dst IP Addr   Packets    Bytes      bps    Bpp Flows
   64.903     213.[redacted].10     26978   35.9 M    4.4 M   1331  2483
  345.371      18.[redacted].146     16177    1.0 M    23763     63  1447
  334.086      18.[redacted].145      6943   455672    10911     65  1708
  339.030      18.[redacted].160      5249   327894     7737     62  1028
  337.181      15.[redacted].19       5032   479597    11378     95  1961
```

Figure 3: NetFlow aggregation using nfdump.

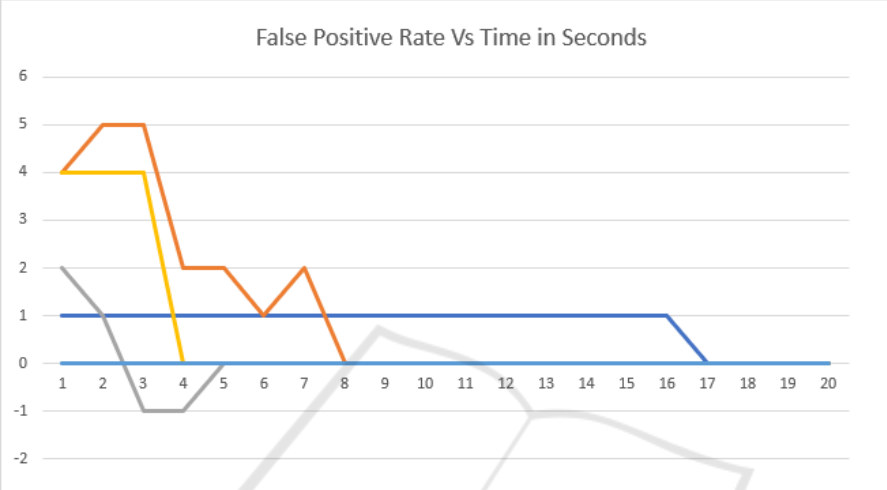
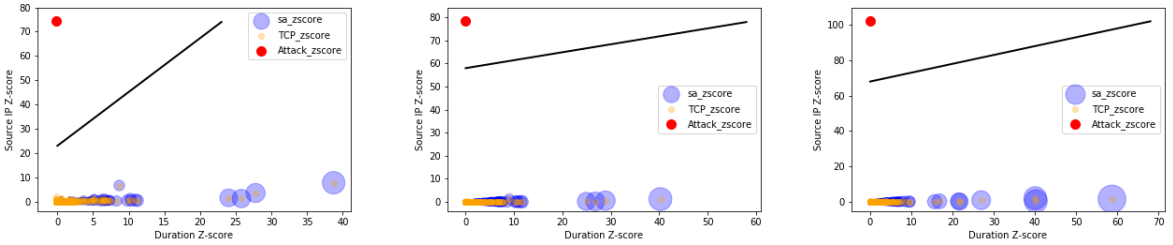


Figure 4: False positive VS time.

come was similar, and the only difference was in TCP attacks where the flows shared the same TCP flag (SYN flooding, Ack flooding, etc.). Figure 3 shows that the statistics provided by ”nfdump” are limited and not enough in order to detect the abnormality in the NetFlow traffic, because the destination port and many fields that might be useful for DDoS detection are missing. Although NetFlow aggregation using ”nfdump” presents promising results in detecting outliers, yet some features such as the count of source IP addresses, the sum of UDP/ICMP/TCP packets are key for DDoS detection. Consequently, a customized aggregation method of NetFlow is needed for detecting various types of DDoS attacks. Another challenge is the use of the appropriate statistical approach on the customized aggregated flow to detect the outliers. This was considered by our proposed NetFlow aggregation method, which extracts main features for DDoS detection. The Z-score and co-variance are calculated for the extracted features to detect DDoS as shown in the next section.

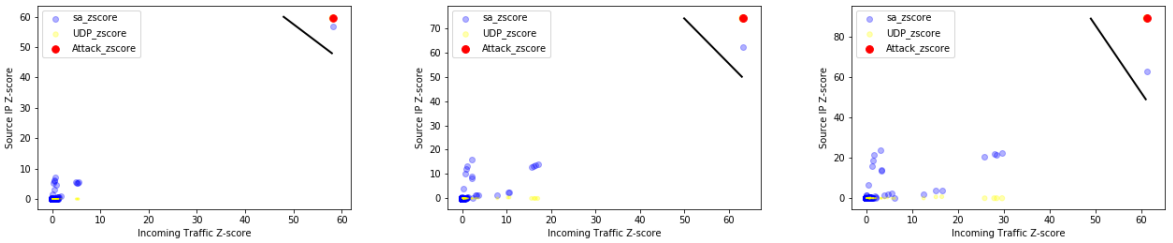
4.3 DDoS Detection

Identifying the best time window to detect attacks is very difficult. For this purpose, the proposed solution was tested for different time slots. The collected datasets were aggregated based on 10, 30, and 60 seconds time windows. The results, shown in Figure 5 for TCP DDoS attacks and in Figure 6 for UDP DDoS attacks, clearly indicate that using the coefficient of variation as a dynamic threshold is efficient for correctly detecting DDoS attacks. For TCP attacks, the method worked perfectly for 10, 30, and 60 seconds time windows. For UDP attacks, as shown in Figure 6, some destination IPs had the number of source addresses greater than the threshold (blue dots), but other features (number of bytes and number of packets) are not higher than the thresholds. Thus, these flows are not detected as attacks. To lower the detection time, we studied the minimum required delay (seconds) to detect DDoS attacks. Figure 4 presents the variation of false-positives versus the time for several DDoS cases (each color represents a type of DDoS attack). According to the results in Figure 4, a time window greater or equal to 30 seconds showed zero false positives and 100% true positives for TCP DDoS attacks. Note that for a time window of 10 seconds, several of these attacks were detected, but other ones were not detected. Therefore, the results indicated that the proposed method is efficient if applied on a time window of 30 seconds, but some crucial real-time applications and high sensitive servers might get harmed if the attack lasts for 30 seconds



(a) DDoS detection within 10 seconds (b) DDoS detection within 30 seconds (c) DDoS detection within 60 seconds

Figure 5: TCP DDoS detection within 10, 30, and 60 seconds.



(a) DDoS detection within 10 seconds (b) DDoS detection within 30 seconds (c) DDoS detection within 60 seconds

Figure 6: UDP DDoS detection within 10, 30, and 60 seconds.

before being detected. For this reason, the maximum Z-score of each feature, in the datasets showing false positive alerts, was studied. We concluded that the threshold of Z-score, which is the coefficient of variation, should not be less than 17 for sensitive services that need detection within 10 seconds. The script should ignore the alert if the CV value is less than 17 since these are most likely due to false positives.

4.4 DDoS Mitigation

The proper mitigation technique should be based on the type and the volume of the attack. For this purpose, a workflow for mitigation was created in Figure 2. If the attack volume and severity are acceptable, the mitigation can be done locally by blocking the attack traffic on the edge router of the ISP before reaching the customers network. However, if the attack severity and volume are huge, the proposed solution forces the edge router to advertise the attacked sub-net on BGP to the scrubbing server, which blocks the attack traffic and sends the benign traffic back to the ISP using Generic Routing Encapsulation (GRE) tunnels. Cleaning the traffic locally, if possible, is better than sending it to scrubbing servers since the GRE tunnel increases the delay and consumes 5% of the traffic.

5 CONCLUSION

DDoS attacks are very serious security threats that need to be detected and prevented in current and future networks. In this paper, we proposed detective, preventative and corrective measures against inbound DDoS attacks. The proposed solution was designed at the ISP level, and it is based on a lightweight statistical approach. Technically, the proposed method applies first aggregation on NetFlow, then statistical analysis to detect DDoS traffic as outliers by using the Z-score and co-variance variation metrics. The proposed technique requires 30 seconds to be confident that a DDoS attack is taking place. The main target of this work is to solve the issue of DDoS attacks by reducing the required resources and latency, which is considered as one of the hardest obstacles to overcome in cybersecurity.

REFERENCES

Alkasassbeh, M., Al-Naymat, G., Hassanat, A., and Almseidin, M. (2016). Detecting distributed denial of service attacks using data mining techniques. *International Journal of Advanced Computer Science and Applications*, 7(1):436–445.

Bertino, E. and Islam, N. (2017). Botnets and internet of things security. *Computer*, 50(2):76–79.

Bhatia, S., Behal, S., and Ahmed, I. (2018). Distributed denial of service attacks and defense mechanisms: Cur-

rent landscape and future directions. In *Versatile Cybersecurity*, pages 55–97. Springer.

David, J. and Thomas, C. (2015). Ddos attack detection using fast entropy approach on flow-based network traffic. *Procedia Computer Science*, 50:30–36.

De Donno, M., Dragoni, N., Giaretta, A., and Spognardi, A. (2018). Ddos-capable iot malwares: Comparative analysis and mirai investigation. *Security and Communication Networks*, 2018.

Douligeris, C. and Mitrokotsa, A. (2004). Ddos attacks and defense mechanisms: classification and state-of-the-art. *Computer Networks*, 44(5):643–666.

Girma, A., Moses, G., Li, J., Lui, C., and Abayomi, K. (2015). Analysis of ddos attacks and an introduction of a hybrid statistical model to detect ddos attacks on cloud computing environment.

Gulihar, P. and Gupta, B. (2020). Cooperative mechanisms for defending distributed denial of service (ddos) attacks. In *Handbook of Computer Networks and Cyber Security*, pages 421–443. Springer.

Hofstede, R., Bartoš, V., Sperotto, A., and Pras, A. (2013). Towards real-time intrusion detection for netflow and ipfix. In *Proceedings of the 9th International Conference on Network and Service Management (CNSM 2013)*, pages 227–234. IEEE.

Hou, J., Fu, P., Cao, Z., and Xu, A. (2018). Machine learning based ddos detection through netflow analysis. In *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*, pages 1–6. IEEE.

Hu, D., Hong, P., and Chen, Y. (2017). Fadm: Ddos flooding attack detection and mitigation system in software-defined networking. In *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pages 1–7. IEEE.

Kolias, C., Kambourakis, G., Stavrou, A., and Voas, J. (2017). Ddos in the iot: Mirai and other botnets. *Computer*, 50(7):80–84.

Radware (2017). Ddos attack history — radware security. https://security.radware.com/ddos-knowledge-center/ddos-chronicles/ddos-attacks-history/. (Accessed on 05/17/2020).

Sawaya, Y., Kubota, A., and Miyake, Y. (2011). Detection of attackers in services using anomalous host behavior based on traffic flow statistics. In *2011 IEEE/IPSJ International Symposium on Applications and the Internet*, pages 353–359.

Shamsolmoali, P. and Zareapoor, M. (2014). Statistical-based filtering system against ddos attacks in cloud computing. In *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1234–1239.

Singh, K., Dhindsa, K. S., and Bhushan, B. (2018). Threshold-based distributed ddos attack detection in isp networks. *Turkish Journal of Electrical Engineering & Computer Sciences*, 26(4):1796–1811.

Xia, Z., Lu, S., and Li, J. (2012). Ddos flood attack detection based on fractal parameters. In *2012 8th International Conference on Wireless Communications, Networking and Mobile Computing*, pages 1–5.

You, Y., Zulkernine, M., and Haque, A. (2007). Detecting flooding-based ddos attacks. In *2007 IEEE International Conference on Communications*, pages 1229–1234. IEEE.

Yuan, X., Li, C., and Li, X. (2017). Deepdefense: identifying ddos attack via deep learning. In *2017 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 1–8. IEEE.

Yun Liu, Jianping Yin, Jieren Cheng, and Boyun Zhang (2010). Detecting ddos attacks using conditional entropy. In *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*, volume 13, pages V13–278–V13–282.

Zargar, S. T., Joshi, J., and Tipper, D. (2013). A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks. *IEEE communications surveys & tutorials*, 15(4):2046–2069.

Zekri, M., El Kafhali, S., Aboutabit, N., and Saadi, Y. (2017). Ddos attack detection using machine learning techniques in cloud computing environments. In *2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech)*, pages 1–7. IEEE.

Zi, L., Yearwood, J., and Wu, X. (2010). Adaptive clustering with feature ranking for ddos attacks detection. In *2010 Fourth International Conference on Network and System Security*, pages 281–286.