Feature Engineering for Distributed Denial of Service (DDoS) attack classification

Alan Barzilay
DCC
University of São Paulo
São Paulo, Brazil
alan.barzilay@gmail.com

Caio L. Martinelli
DCC
University of São Paulo
São Paulo, Brazil
caio.martinelli@usp.br

Daniel M. Batista
DCC
University of São Paulo
São Paulo, Brazil
batista@ime.usp.br

Roberto Hirata Junior
DCC
University of São Paulo
São Paulo, Brazil
hirata@ime.usp.br

I. Introduction

DDoS (Distributed Denial of Service) attacks have always been a menace for service providers everywhere, but with the increase of IoT devices came also a number of new opportunities for attackers. The Mirai botnet [1] is only one example of such opportunities, where a botnet of vulnerable IoT devices was used to create a massive DDoS attack that took down the DNS provider Dyn. With the possibility of new and powerful attacks being ever present, the need for efficient and agile tools and models for DDoS detection is a must if we ever intend to be able to take counter-measures.

The usual Intrusion Detection System (IDS) for DDoS is composed of 3 main parts: DDoS detection, flow classification and reaction. Once an attack is detected by the IDS it will try to classify the incoming flows in order to distinguish which flows are legitimate and to be able to take counter-measures. To protect the server a plethora of actions may be taken depending on the attack, to defend against a volumetric attack the system may implement rate limiting policies and/or request the completion of CAPTCHAs. Against a low and slow attack such as Slowloris [2] it may limit the number of connections allowed by a single IP, against a botnet that spans the globe it may be useful to use Content Delivery Networks (CDN) to distribute the traffic among multiple servers. Each DDoS attack is different and has its own particularities that must be taken into account when deciding on a counter-measure.

But the massive volume of data being generated and transmitted everyday imposes a serious challenge to IDSs, most common and readily available data processing tools aren't able to keep up with the sheer volume of information being received by a usual server. If an IDS model that is too computationally expensive is deployed it may actually hinder the server since it will lead to less computational power being available to sustain the DDoS. There is a trade off between system complexity and robustness. In order to develop new models and tools we need to be capable of manipulating the data stream and easily extract features that are descriptive of the flow being analyzed.

In this work we intend to develop a new set of features that are computationally inexpensive and descriptive of the data stream. We expect to generate this set by reviewing the literature for features already in use and develop new ones. Once we have this set we will compare its descriptive power with other sets of features used in other models.

For this we will use the realistic DDoS dataset described on the article [3], which gathers a variety of attacks and modern techniques. We hope to extract relevant features to classify the bidirectional flows, using the same definition of flow as the authors: a sequence of packets with same values for (*Source IP, Destination IP, Source Port, Destination Port and Protocol*), we call this the 5-tuple flow. Additionally we will use another kind of flow, the 2-tuple flow, the sequence of packets with the same values for (*Source IP, Destination IP*). The 2-tuple flow may be seen more intuitively as the combination of multiple 5-tuple flows.

The feature extractor used by the authors in this dataset, the CICFlowMeter [4], gathers information on the bidirectional (5-tuple) flow. In this work we intend to aggregate more variables to this feature set, using information from different flows for the same IP or even on individual packets, there are more features to be obtained that do not depend exclusively on individual bidirectional 5-tuple flows. Another possible course of action is to engineer attack specific features. We believe this kind of feature will enhance the performance of classification models.

Also, we intend to create a cache policy to store information on flow history for blacklisted IPs, having always the goal of building something with low memory and computational resources consumption.

II. Literature Review

The original paper of the UNB CIC DDoS 2019 dataset [3] that we will be using describes the creation of the dataset and the use of supervised machine learning algorithms to classify bidirectional flows and distinguish the regular traffic from the attack traffic. The dataset is composed of the pcap files of the simulated attacks and the tabular data representing the set of features extracted with the CICFlowMeter tool [4]. This tool was originally developed to classify different types of Tor traffic and utilizes pcap files as an input. With these features they use a Random Forest model to calculate the feature importance for different kinds of DDoS attacks (UDP-lag, TFTP, WebDDoSDNS, regular traffic, MSSQL, LDAP, NetBIOSNTP, SSDP, SNMP, Syn and UDP).Then they evaluate the performance of the classification algorithms ID3, Random Forest, Naive Bayes and Logistic regression on the test data.

In the chapter 5 of the PhD dissertation [5] it is discussed features to detect DDoS attacks based on a different kind of flow from the one used in the original dataset. The flow is defined as a sequence of packets from the same IP source to the same IP destination, a 2-tuple flow. And the features include the number of ports and of protocols present in that flow, something that is impossible when creating features restricted to the 5-tuple bidirectional flows. We intend to recreate this features, not restricting our information to that kind of flow.

The dataset we will be using was also utilized in [6] in conjunction with the DARPA 1999/2009 [7] dataset to create a real-time attack detection alarm based on the entropy of bidirectional flows, considering sliding time windows to keep the entropy calculation computationally efficient.

III. "Proposta de trabalho"

We divide the work of this paper in three steps, first we will generate features as the ones mentioned in the literature review and attempt to generate other relevant features directly from the pcap files. With this features in hand we can compare them with the most important features for each attack as described in [3]. We will use metrics such as Information Value (IV) and Spearman's correlation with the target. In the design of the feature extraction tool it will be possible to extract features of the incomplete flow, not only the finished flow, as we intend to create a tool that can be used in real time.

Secondly we will build models with these new features and compare it's performance with the models of [3]. For this we will be calculating the features only on the finished flows in order to have a better basis of comparison between them since the authors of the dataset generated their features this way.

Finally, we will try to make a policy to filter the traffic in the network, such as ignore all the packets sent from/to some specific IP given a number of partial (or completed) flows that were classified as attacks by our model. This policy could be applied on the network by a firewall such as Linux's `iptables` rules. To analyze the efficiency of this DDoS counter-measure we will calculate metrics such as the precision, the True Positive Rate (TPR) and the False Positive Rate (FPR) of this filters, simulating how much of the attack traffic could be filtered out and how much of the benign traffic we would be impacting.

IV. Experimental Design

A. Feature generator

We aim to be able to generate features for the bidirectional flows at any given time. For this we will keep two dictionaries, one of the information on the 2-tuple bidirectional flow (*Source IP, Destination IP*) and one with information of the 5-tuple bidirectional flow (*Source IP, Destination IP, Source Port, Destination Port and Protocol*). The update procedure of the 2-tuple dictionary is represented on Fig. 1. The update procedure for the 5-tuple occurs in an analogous manner.

The input of this tool will be a pcap file that in a practical situation could be constantly updated. At any given time it will be possible to extract features of the flow from this dictionaries.

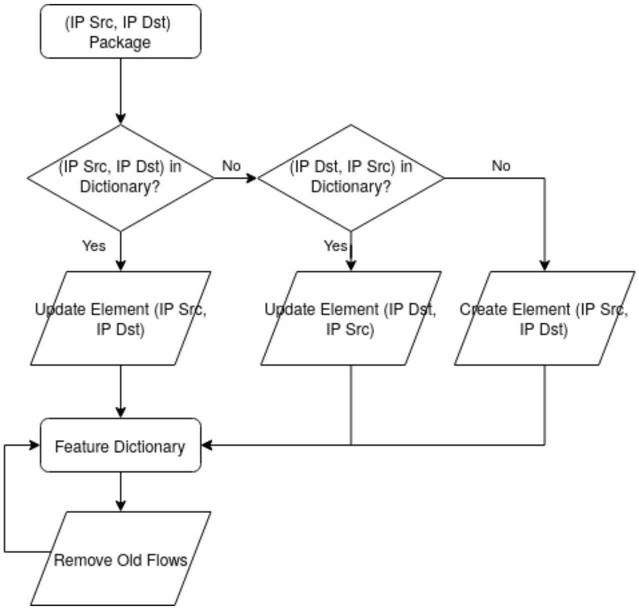

Fig. 1. Update procedure of the 2-tuple dictionary.

B. Classification Models

Once we have our feature extractor we will apply it on each flow as soon as it's completed, i.e. as the last packet of the flow is received. Apart from our own feature generator we will be also utilizing CICFlowMeter on the dataset to obtain a complementing set of features.

This step has two goals. The first one is to validate our tool by checking if it generates features consistent with the ones given by the CICFlowMeter. The second goal is to be able to compare its performance with CICFlowMeter as we intend to have a computationally efficient tool.

Lastly we will compare the performance of our models with the ones obtained in the paper, and will measure the gain in performance obtained by the addition of the new features we generated.

C. Packet Filter Policy

After testing different techniques and models in the previous step we will choose one to use, considering both their precision and their computational cost.

We will split our training data in to two sets, one to calibrate the model and one to generate the policy. Then we will evaluate the policy on the test data.

V. Conclusion and Future Work

This paper creates a tool that is cheap in computational resources and is specifically designed to detect DDoS attacks, showing improvements in the capability to detect attack flows and enabling network administrators to take counter-measures.

In the next iteration of this paper we intend to have built and validated the feature tool and to have first results of the models and the packet filter policy.

In future work, we intend to further investigate the computational resources required to keep this system we built working. Also to investigate more mechanisms of counter-measures to take against malicious packages and flows.

REFERENCES

[1] Anna-senpai, “Mirai botnet source code”, https://github.com/jgamblin/Mirai-Source-Code, Accessed: 2020-04-15

[2] Shorey, T., Subbaiah, D., Goyal, A., Sakxena, A., & Mishra, A. K. (2018, September). Performance comparison and analysis of slowloris, goldeneye and xerxes ddos attack tools. In 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 318-322). IEEE.

[3] I. Sharafaldin, A. H. Lashkari, S. Hakak and A. A. Ghorbani, “Developing realistic Distributed Denial of Service (DDoS) attack dataset and taxonomy”, 2019 International Carnahan Conference on Security Technology (ICCST), CHENNAI, India, 2019, pp. 1-8.

[4] A. H. Lashkari, G. D. Gil, M. Mamun, A. Ghorbani, “Characterization of Tor traffic using time based features” 2017, 253-262. 10.5220/0006105602530262.

[5] M. Andreoni, O. C. M. B. Duarte, G. Pujolle, “A monitoring and threat detection system using stream processing as a virtual function for big data”, 2019, 209-216. 10.5753/sbrc_estendido.2019.7789.

[6] J. Li, M. Liu, Z. Xue, X. Fan and X. He, ”RTVD: A Real-Time Volumetric Detection Scheme for DDoS in the Internet of Things,” in IEEE Access, vol. 8, pp. 36191-36201, 2020.

[7] Gharaibeh, Manaf, and Christos Papadopoulos. “Darpa-2009 intrusion detection dataset report.” Tech. Rep. (2014).