Improving the Generation of Labeled Network Traffic Datasets Through Machine Learning Techniques.

Jorge Guerra[1] and Carlos Catania[2]

[1] ITIC-CONICET, Universidad Nacional de Cuyo (UNCuyo), Mendoza, Argentina
[2] Ingeniera, Universidad Nacional de Cuyo (UNCuyo), Mendoza, Argentina
jguerra@uncu.edu.ar
ccatania@itu.uncu.edu.ar

Abstract. The problem of detecting malicious behavior in network traffic has become an extremely difficult challenge for the security community. Consequently, several intelligence-based tools have been proposed to generate models capable of understanding the information traveling through the network and to help in the identification of suspicious connections as soon as possible. However, the lack of high-quality datasets has been one of the main obstacles in the developing of reliable intelligence-based tools. A well-labeled dataset is fundamental not only for the process of automatically learning models but also for testing its performance. Recently, RiskID emerged with the goal of providing to the network security community a collaborative tool for helping the labeling process. Through the use of visual and statistical techniques, RiskID facilitates to the user the generation of labeled datasets from real connections. In this article, we present a machine learning extension for RiskID, to help the user in the malware identification process. A preliminary study shows that as the size of labeled data increases, the use of machine learning models can be a valuable tool during the labeling process of future traffic connections.

Keywords: Machine Learning, dataset generation, network security

1 Introduction

In the field of network security research, intelligence-based detection approaches emerged as a tool for dealing with the fast evolution of the different network scenarios. Probably the most significant challenge during the developing of such systems is the lack of appropriate public datasets [9]. Just before deploying in any real world environment, an intelligence-based Network Intrusion Detection System (NIDS) must be trained and evaluated using real labeled network traffic traces with an intensive set of intrusions or attacks [5].

One of the reason behind behind the lack of public datasets arises from the data's sensitive nature. It is no secret to anyone that bringing to light network

traffic can reveal sensitive information. Such information is generally related to confidential and personal communications coming from organization business data or in other cases user private access behaviors. It is understandable that in the face of such high risks, researchers frequently encounter insurmountable organizational and legal barriers when they attempt to provide datasets to the community [9].

For the previous reason, the main strategy points to create synthetic or benchmark dataset to deal with the data confidentiality problem. Synthetic datasets are created to represent certain problem domains. Specifically to cover specific needs or certain condition [5]. On the other hand, benchmark datasets are often very useful but suffer excessive preprocessing that separates them from real network environments. Examples of known benchmark datasets are: KDD-cup99 [10] who was built upon the data captured in the DARPA98 IDS evaluation program, DEFCON [2] that contains network traffic captured during a hacker competition called "Capture The Flag", CAIDA dataset [1] that contain particular kind of attack, among others.

Other solutions point to real life datasets. These datasets are created generally capturing traffic from institutional networks. Recently, the Stratosphere Intrusion Prevention System (IPS) project [3] has emerged as a project focused on providing an state of the art IPS to the Non Governmental Organizations (NGO). One of the goals of the project consists of generating high quality datasets for testing and developing new malware detection techniques. The particular encoding of the network behavior used by stratosphere IPS project facilitates the release of network data to the community.

However, the problem of labeling all the published data remains a very difficult task. The labeling process not only requires a considerable human effort but also the responsible of labeling must be a security specialist, who could not be always available. The fact is that the difficulty behind the labeling process could be the real reason behind the lack of high quality real life datasets.

With these issues in mind, we developed RiskID [7]. Still in an early stage, RiskID aims at being a collaborative labeling tool based on visual analytics and statistical techniques. In particular, RiskID is based on the combinations of several visualizations strategies with clustering algorithms working together for facilitating the recognition of malicious traffic. In this paper, we propose an extension for RiskID based on machine learning techniques. The general idea behind using machine learning techniques inside RiskID consists of generating a classifier trained on the subset of already labeled connections and use classifier output for helping the user in the decision process. Here, the goal is just to provide to the user another tool for supporting his decision process.

The rest of the article is organized as follows: Section 2 describes the problem statement, details the strategy for generating the behavioral models proposed by the Stratosphere project and shows the general aspects of the RiskID tool. Then section 3 exposes the proposed machine learning extension for RiskID. Section 4 presents the experiments design and the performance evaluation of the proposed

extension for RiskID. Finally, concluding remarks and future works are described in section 5.

2 Problem Statement

The RiskID tool takes a dataset containing network traffic as input and uses visual representation techniques to help users to identify malicious behaviors. The main goal behind RiskID is to provide to the network security community labeled datasets. We believe that by including machine learning techniques inside RiskID, the effort of identifying malicious behavior can be reduced. As an extension of current RiskID labeling tool, we propose the inclusion of a Machine Learning classification algorithm for helping the user during the labeling process. The idea is simple: Based on the subset of already labeled SC connections, the classification algorithm can provide the probability that a given connection. Notice that the final decision will still be user responsibility. We proceed to discuss the main aspects of the current version of RiskID.

2.1 The RiskID Tool

As you can see in Figure 1, RiskID is a visual analytics tool that combines visualization with clustering techniques to assist the user in the process of labeling connections [7]. The goal is to obtain a real life data set with, as much as possible, labeled connections. Specifically, the application aim at labeling traffic generated by botnet attacks. The process for labeling in RiskID is the following:

1. RiskID receives a JSON file that contains for every SC (from here referred simply as connection) some basic network information, such as IP addresses and Ports, together with its corresponding behavioral encoding.
2. The feature extraction module analyzes all connection behavioral encodings and creates for every connection a new vector summarizing the information in terms of periodicity, duration and size.
3. The cluster composition module analyzes the feature vectors and groups them according to a standard similarity measure.
4. The UI represents the list of connections with a Heatmap of the feature vectors, using different colors for each type of feature.
5. The user explores the connection list to discover common patterns through color similarity. During this process she can select potentially similar connections.
6. Upon selection, details about the connection composition are shown in order to facilitate comparison.
7. Eventually, when the user finds a high coincidence between selected connections she can proceed to label them as "Botnet" or "Normal".

It is worth noting that a correct labeling process mainly depends on the user selection strategy.

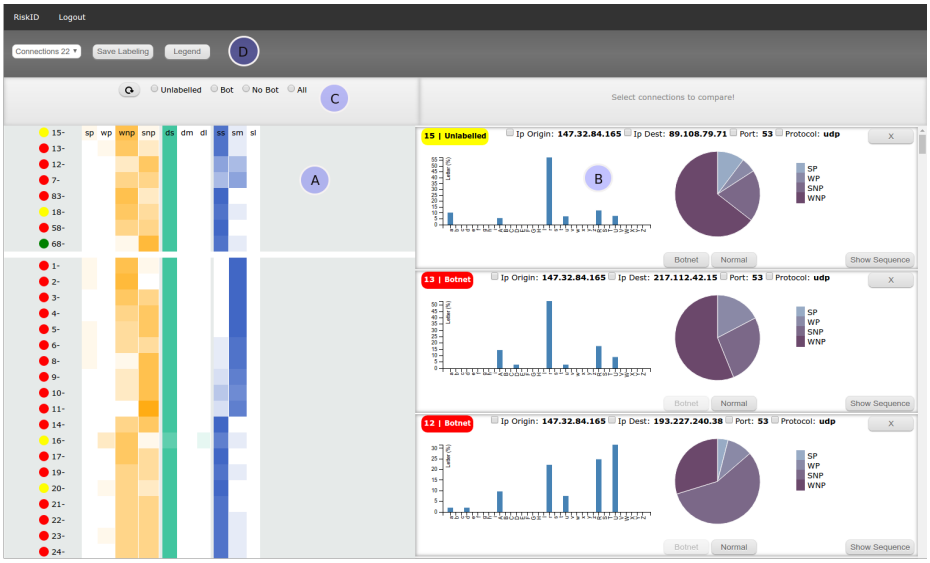


Fig. 1: RiskID User Interface. A: left panel collects descriptors for all connections with heatmaps of different colors for periodicity, size and duration features. B: right panel shows details for up to three connections. C: control panel to get a set of connections with similar filters selections. D: control panel to select Stratosphere datasets, get visual representation legends and save the labeled dataset resultant after connections labeling task by the user.

2.2 Behavioral Models

To deal with the confidentiality problem of network data, RiskID uses the encoding proposed by the Stratosphere IPS project. Such encoding only consider the size, duration and periodicity of network flows. The encoding start aggregating the flows according to a 4-tuple composed of: the source IP address, the destination IP address, the destination port and the protocol. They create *Stratosphere connection* (SC) putting together all the flows that match a tuple. From a traffic capture several of these SC are created. Each one of the these SC contains a group of flows. A sample **behavioral encoding** is shown in Fig. 2. The figure shows the symbols representing all the flows for a SC based on UDP protocol from IP address 10.0.2.103 to port 53 of IP address 8.8.8.8.

2.4.2*4.R.R*a*b*a*a*b*b*a*R.R*R.R*a*a*b*a*a*a*a*

Fig. 2: An example behavioral encoding of connection from IP address 10.0.2.103 to destination port 53 at IP address 8.8.8.8 using UDP.

RiskID as part of its data conversion generate for each connection a 10-dimensional numerical vector (denoted as feature vector) where the first four dimensions represent the periodicity (strong periodicity, weak periodicity, weak non periodicity and strong non periodicity respectively), the other three refer to duration (duration short, duration medium and duration large respectively) and the last three represent the size (size short, size medium, size large). The feature vector for a given connection is generated considering, for the complete symbol sequence, the cumulative frequency of the corresponding values associated with the behavioral encoding. At the end of the sequence, a percent of each feature is calculated and normalized between the values [0,1]. These final vector are used to create the heatmap visual representation where the intensity in the color scale indicates a given feature is predominant over the rest. To improve the heatmap the connections are organized making clusters using k-means algorithm [7]. For each connection we use the 10-dimensional feature vector and cluster ID to shape the training set.

2.3 Visualization Techniques

Main visualization tools used in RiskID consists of a heatmap representation supported by cluster strategies. Both components joined to filter options, histograms and pie graphs Figure 1(B) make up the bulk of the user interface. The heatmap representation is used for the list of Stratosphere connections Figure 1(A), that shows the connections grouped by the similarity in their encoding behavior. With the heatmap, it is intuitive to recognize the predominant features of each connection and, more importantly, relate connections with similar features. First, the connections are grouped by clusters. The clustering process helps the user get a first approximation of similar connections [7]. Once any connection is selected by a user the connection details section add this to the connection details list and show relevant information about it. A histogram graph displays their character distributions while a pie graph shows specifically their periodicity feature percent. One important advantage of RiskId is the possibility to compare two or more connections [7]. Each newly selected connection is placed under the previously selected one, and the details are stacked in the Detailed Connection View. Thus, the user can start a detailed comparison.

3 A Probability Estimator Based on Machine Learning Techniques

Machine Learning techniques are extensively used by malicious detection systems [9]. Their ability to learn with little data and then predict or detect similar behaviors make it a fundamental strategy in network detection field. In particular, we use a bagging strategies know like Random Forest (RF). RF is a general class of ensemble building method using a decision tree as the base classifier [8]. We use an RF to through the labeled connections to get a probability of botnet in unlabeled connections.

Once we get a botnet probability of whole unlabeled connections we help the user's decision for the next labeled. This information can be useful to represent in the connection list the output classification given by RF allowing the users get another evaluation criteria. We don't pretend to be determinants of the labeling process but yes be an influence in users decision. This new feature could reduce labeling time improving whole labeling task. We get first labeled connections by the RiskID users and create a training set to a machine learning algorithm and predict botnet behavior in remainder connections. Formally, suppose we have C like whole dataset and let C_r and C_t be the training set and test set respectively, we require $C_r \cap C_t = \emptyset$ and $C_r \cup C_t = C$.

4 Evaluation

For measuring the actual impact of the proposed machine learning extension, we need to consider not only a statistical evaluation but also the user interaction and confidence of the proposed extension. However, in this paper we will focus solely on the computational evaluation, leaving the evaluation with users for a later experiment. Specifically, we evaluate what is known as learning rate: the speed at which our extension learns new information or trends and updates the probability by connections accordingly [4]. We address the following questions:

1. Can random forest face at the beginning the disproportion between labeled and unlabeled and get a correct botnet probability for unlabeled connections?
2. How random forest behaves by types of connections?

The first question aims at the study of the classification performance of the proposed algorithm when it is trained with an small portion of labeled data and to predict the rest. Just like recommender system faces the cold start problem at first the model faces the challenge of giving a botnet probability from the few data labeled by users so far. Presumably, these probabilities will not be the best, but how well or poorly does the algorithm predict against this difficulty is the scenario we want to evaluate. For the second question we want to analyze the classification performance of the proposed algorithm considering the different type of traffic connection. It is clear that the traffic behavior of SMTP connections could be very different compared with HTTP. Such difference can certainly impact the in the performance of the learning rate of the classifier.

4.1 Dataset Description

For evaluating the performance of the proposed algorithm, we use the CTU-13 Dataset. The CTU-13 dataset consist of a group of thirteen different malware captures done in a real network environment taken from CVUT university campus networks. Datasets are publicly available as part of the Malware Capture Facility Project (**MCFP**) [6].

ID	IRC	SPAM	CF	PS	DDoS	FF	P2P	US	HTTP	Botnet Conn.	Normal Conn.	MCFP IDs
A	X	X	X	-	-	-	-	-	-	911	415	CTU13-42
B	X	X	X	-	-	-	-	-	-	344	82	CTU13-43
C	X	-	-	X	-	-	-	X	-	13	292	CTU13-44
D	X	-	-	-	X	-	-	X	-	70	441	CTU13-45
E	-	X	-	X	-	-	-	-	X	31	147	CTU13-46
F	-	-	-	X	-	-	-	-	-	211	159	CTU13-47
H	-	-	-	X	-	-	-	-	-	29	135	CTU13-49
I	X	X	X	X	-	-	-	-	-	4000	513	CTU13-50
J	X	-	-	-	X	-	-	X	-	22	174	CTU13-51
K	X	-	-	-	X	-	-	X	-	2	21	CTU13-52
L	-	-	-	-	-	-	X	-	-	57	209	CTU13-53
M	-	X	-	X	-	-	-	-	X	479	231	CTU13-54

Table 1: Characteristic of botnet scenarios and general information about datasets.(CF: Click Fraud, PS: Port Scan, FF: Fast Flux, US: Compiled and controlled by us)

Table 1 provides brief information about each of the thirteen datasets. The first column shows the ID used for referencing the dataset. The next nine columns show the characteristics of the botnet scenarios. Then, in column ten and eleven, the number of connections labeled as botnet and normal. Finally, the last column shows the ID of the dataset in MCFP. For the purpose of the study, the three datasets were merged.

4.2 Experiment Design

During the training phase, we apply k-fold cross validation with 5 folds. To answer the first question, we trained the algorithm on different sizes portion of the dataset and tested on the remaining portion. In next iterations, we increase the number of data in the training set. This way, we simulate the use of the algorithm within the application over time. We started by taking only a random sample of 200 connections for the training set and tested with the remaining 8788 connections. Following this, we randomly took another 200 connections from the test set to add them to the training set. We perform this operation until there are approximately 200 connections in the test set. We use F1 Score to evaluate the RF performance at each iteration. This metric can be interpreted as a weighted average of the fraction of relevant instances among the retrieved instances (precision) and the fraction of relevant instances that have been retrieved over total relevant instances (recall). Each experimental scenario was simulated 30 times (i.e. 1320 simulations in total) to ensure the statistical robustness of results. As the final result, we use the mean value for F1 Score. To answer the second question we use the probabilities resultant of RF when was training with 2 percent, 50 percent, and 90 percent of the training set. These percentage values were selected to represent the initial, middle and final phase, respectively, of the labeling process. We perform a study of the algorithm behavior by types of connections. The connections were divided by port type, keeping only the most representative of all of them.

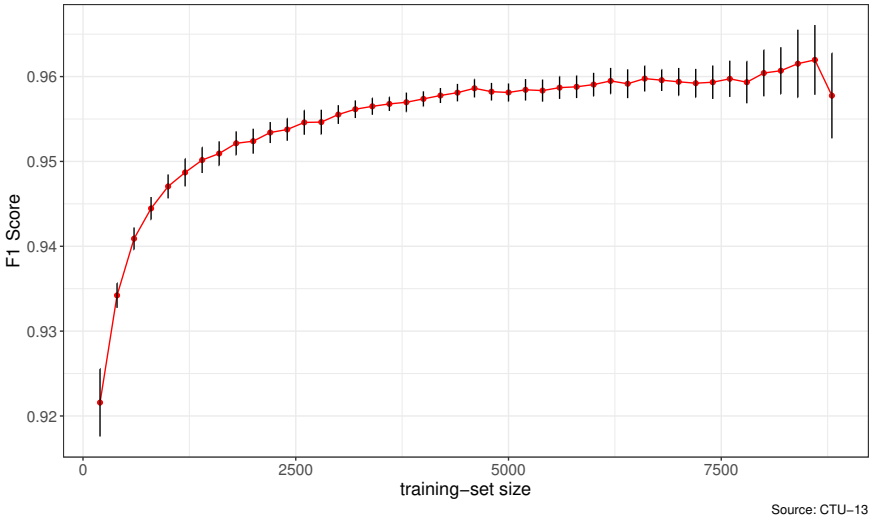


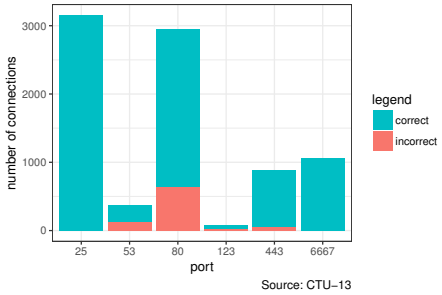
Fig. 3: Random Forest performance with incremental training data

4.3 Random Forest Performance with Incremental Training Data

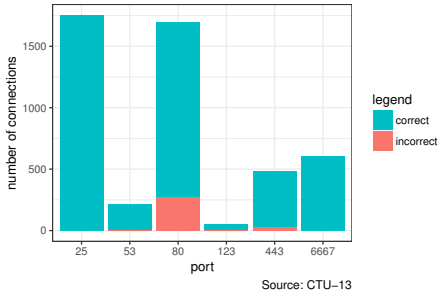
The idea behind this experiment is to analyze the impact of incrementing the number of data in training set. Specifically, we evaluate the F1 score metric by each iteration of RF. Figure 3 shows that as the number of data in the training set increases, the mean of F1 score improves. On the X-axis we have the different sizes in training set. Each point in the graph represents the number of times that RF is training and testing. On the Y-axis we have the mean of F1 score returned by RF in each iteration. When we training RF with the first 200 labeled connections (approximately 2 percent of the whole CTU-13 dataset) and then test with the remaining portion of the dataset we get a mean F1 Score about 0.92 value. This is a good first result but the F1 score still increases until 25 iterations when it stays oscillating close to 0.96 value. This result shows that approximately from the first 5000 (55 percent of the whole CTU-13 dataset) labeled connections, RF predicts with good results the rest of the connections.

4.4 Detection Performance Analysis of Random Forest by Type of Connections

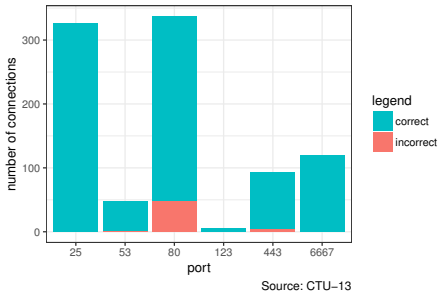
In this section, we analyze the detection performance of RF by type of connections. Specifically we analyze the connections by ports. In the previous experiment, we evaluate RF with different training set size. The Figure 4 displays the proportion of connections correctly classified (show in color green) and incorrectly classified (show in color red) by the most representative ports when RF was training with 2 percent, 50 percent, and 90 percent of the training set. The information provided by Figure 4 exposes the fact that most of the traffic in CTU-13 come from 25 port. The port 25 refers to Simple Mail Transfer Protocol



(a) Prediction performance by port with 2 percent of the training set



(b) Prediction performance by port with 50 percent of the training set.



(c) Prediction performance by port with 90 percent of the training set.

Fig. 4: Prediction performance by port.

(SMTP), used for email routing between mail servers. As can be seen, the RF detection model is able to detect the 100 percent of all the cases where port 25 is present. Even when the proposed algorithm was generated using only the 2 percent of the labeled data. This good performance happens because SMTP traffic may be very similar to each other and with just a few examples we can learn a lot. By the other hand the port 80 (refers to Hypertext Transfer Protocol "HTTP") present some detection error. This error is present due to the variability in the connections that share port 80. The remaining ports, although they have bad qualifications, most of the times connections are well identified.

5 Concluding Remarks and Future Work

In the present article we presented a machine based extension to RiskID [7], a tool for generating labeled network traffic datasets for research community. The preliminary study shows the viability of using a probability estimator generated from the subset of labeled data. To start using this extension inside RiskID tool, some connections need to be labeled previously. The Figure 3 has shown that

is not necessary much data for train our strategy and get good results. Once the new extension starts to suggest probability of botnet for each connection the users will have a new evaluation criterion. This favors the increase of connections that will be used to train and as we saw in Figure 3 this improve the detection performance. In this way, the new extension proposed in this article decreases the labeling time of the analyzed dataset. A similar analysis considering the type of connections displayed same results. Figure 4 showed decrease in error (bar with color red) when RF was trained with more data. In some cases, such as SMTP connections, the amount of needed labeled connections is considerable small (about 2 percent of the dataset). On the other hand, HTTP connections required a higher number of labels to reduce the classification errors. A user experience analyze is required to get a final evaluation for this new extension. We are working on some test to collect users experience and will be exposed in future works. Last observation is about the quality of training dataset, which determines that the proposed strategy be able to create a good prediction model. RiskID users could label connections with some errors and this could impact in RF results. Noise robustness is a very important problem in machine learning algorithms and our strategy don't escape this. The quality of first labels created by RiskID users is crucial for a good future prediction of our RF algorithm. Feature analysis about noise robustness of our new extension proposed for RiskID will be accomplished in next works.

References

1. Center for applied internet data analysis. `http://www.caida.org/`, October 2011. [Online; accessed May-2017].
2. The shmoo group. `http://cctf.shmoo.com/`, October 2011. [Online; accessed October-2016].
3. Stratosphere ips project. `https://stratosphereips.org/`, October 2015. [Online; accessed Jun-2017].
4. Iman Avazpour, Teerat Pitakrat, Lars Grunske, and John Grundy. Recommendation Systems in Software Engineering. 2014.
5. Monowar H. Bhuyan, Dhruba K. Bhattacharyya, and Jugal K. Kalita. Towards generating real-life datasets for network intrusion detection. *International Journal of Network Security*, 17(6):683–701, 2015.
6. Sebastian Garcia. *Identifying, Modeling and Detecting Botnet Behaviors in the Network*. PhD thesis, UNICEN University, 2014.
7. Jorge Guerra, Carlos Adrián Catania, and Eduardo Veas. Visual Exploration of Network Hostile Behavior. *Proceedings of the 2017 ACM Workshop on Exploratory Search and Interactive Data Analytics - ESIDA '17*, pages 51–54, 2017.
8. Ludmila I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms: Second Edition*. 2014.
9. Robin Sommer and Vern Paxson. Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. *2010 IEEE Symposium on Security and Privacy*, 0(May):305–316, 2010.
10. Irvine University of California. Knowledge discovery in databases darpa archive. `http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html/`, October 1999. [Online; accessed September-2016].