

Datasets are not Enough: Challenges in Labeling Network Traffic

Jorge Luis Guerra¹, Carlos Catania, Eduardo Veas

Abstract

In contrast to previous surveys, the present work is not focused on reviewing the datasets used in the network security field. The fact is that many of the available public labeled datasets represent the network behavior just for a particular time period. Given the rate of change in malicious behavior and the serious challenge to label, and maintain these datasets, they become quickly obsolete. Therefore, this work is focused on the analysis of current labeling methodologies applied to network-based data. In the field of network security, the process of labeling a representative network traffic dataset is particularly challenging and costly since very specialized knowledge is required to classify network traces. Consequently, most of the current traffic labeling methods are based on the automatic generation of synthetic network traces, which hides many of the essential aspects necessary for a correct differentiation between normal and malicious behavior. Alternatively, a few other methods incorporate non-experts users in the labeling process of real traffic with the help of visual and statistical tools. However, after conducting an in-depth analysis, it seems that all current methods for labeling suffer from fundamental drawbacks regarding the quality, volume, and speed of the resulting dataset. This lack of consistent methods for continuously generating a representative dataset with an accurate and validated methodology must be addressed by the network security research community. Moreover, a consistent label methodology is a fundamental condition for helping in the acceptance of novel detection approaches based on statistical and machine learning techniques.

Keywords: Network Security, Automatic Labeling, Assisted Labeling, Datasets, Network Traffic

¹jorge.guerra@ingenieria.uncuyo.edu.ar

1. Introduction and Motivation

A Network Intrusion Detection System (NIDS) is an active process that monitors network traffic to identify security breaches and initiate measures to counteract the type of attack (e.g., spam, information stealing, botnet attacks, among others.). Today's network environments suffer from constant modification and improvements. Therefore, a rapid adaptation by NIDS is necessary if they do not want to become obsolete [1, 2]. Consequently, NIDS based on statistical methods, machine learning, and data mining methods have increased their application in recent years mostly because of their generalization capabilities [3, 4].

However, much of the success of the so-called statistically based NIDS (SNIDS) will depend mostly on the initial model generation and the benchmarking before going into the production network infrastructure [5]. Both procedures will heavily relies on the quality of labeled datasets used.

Although dataset quality is not precisely defined, several authors [6, 7] agree that representative and accurate labels are the main two aspects for measuring the quality of a network traffic labeled dataset. A representative labeled dataset should provide all the associated behavioral patterns for malicious and normal network traces. Representativeness is particularly important when labeling network traces from normal users, where timing patterns, frequency of use and work cycle must be precisely included in the dataset. In the case of malicious network traces, the sequence of misuse actions performed on the network and their periodicity patterns are examples of representative information. On the other hand, an accurate label should be assigned only to those portions of a network trace containing the behavior of interest. A mislabeled and underrepresented dataset will have direct consequences on the performance of any model generated from the data.

Several aspects can be studied during the generation of labeled datasets for the network security field, such as the mechanism used during the traffic capture [8, 9, 10, 11, 12, 13], the subsequent cleaning process [14, 15], the method of feature extraction [16, 17, 18, 19], and the strategy for labeling the network traces, among others. In the particular case of the labeling strategy, it is possible to analyze the process as a simple detection/classification problem in which a given network traffic event is classified as normal or malicious. However, there are meaningful differences in the process of traffic labeling

compared with a conventional traffic detection process. These can be framed under the following aspects:

- **Timing:** in the labeling process, there is no need to perform the detection in a particular time frame. During the labeling process, the security analyst (automated system or expert user) can take the required time to confirm the potential anomaly or misuse. **(D1)**
- **Relevance:** a false positive is not as crucial for a real-time detection system as it is for a labeled data set creation system. A false positive is an inconvenience to the user during real-time analysis. For the labeling process, however, it merely represents part of the noise that might occur in the resulting dataset. **(D2)**
- **Qualitative:** the focus of the labeling process is to get a set of accurate labels representing the most significant characteristics of the network. The more representative is data, the better will be the resulting model for performing detection. As an example, a labeled dataset with a considerable set of confirmed malicious network traces coming from a unique source and following the same pattern could be easy to predict. However, it might not be useful for generating a proper detection model. **(D3)**
- **Scope:** the scope between the detection problem and the labeling of network traces are often different. Usually, network security datasets are created with a particular scope in mind. On the other hand, when performing real-time detection on real network environments, the detection of malicious traffic does not restrict between network traces. It has the task of classifying all traffic. **(D4)**
- **Economic:** the labeling process has no immediate economic consequences. In other words, when confronted with an undetected malicious network trace, in general, there is no consequence beyond inadequate data for the construction of a statistical prediction model. While in the case of an operational detection system, the non-recognition of malicious behavior can cause important losses to the organization. **(D5)**

Over the past 20 years, several methods have been developed to address the problem of labeling applied to network data sets. One of the most widely

used methods has been using a controlled network environment for classifying network traces within monitored time windows. The reason behind such a decision responds to the simplicity of the labeling process.

However, the method fails in capturing many of the behavioral details of realistic network traffic. Consequently, the resulting labeled dataset ends up providing a dataset representing partially the conditions observed in a real network environment. Recently, some other methods based on statistical learning, visualization, and a combination of both (assisted methods) have emerged to deal with more realistic network traffic and speed up the labeling process. Nowadays, it is not clear whether such approaches provide a significant help for the labeling process. The fact is that much of the analysis and labeling of network traffic is still performed manually: with an expert user observing the network traces [20, 21]. As mentioned by [4, 22], such a situation could be a definite obstacle for the massive adoption of SNIDS in the network security field.

The present document provides an extensive review of the works presenting methodological strategies for generating accurate and representative labels for network security datasets. The survey emphasizes the application of labeling methods based on machine learning and visualization techniques and their benefits and limitations in the generation of quality labels for building and evaluating the performance of SNIDS.

The rest of this document is organized as follows: Section 2 presents the methodology used for the selection criteria of the papers presented in this survey. Section 3 provides background information about the labeling process, including a taxonomy and a brief description of the limitations of current labeled datasets available for security research. Then, in Section 4, the current methods for labeling network traffic are reviewed and compared based on the taxonomy, while most relevant aspects of each strategy are discussed in section 5. Section 6 remarks the challenges and open issues in current labeling methods for achieving quality network traffic datasets. Finally, concluding remarks are provided in Section 7.

2. Methodology

This study conducts a review of the different labeling methods for generating network traffic datasets and investigates the published journal article in the last 20 years. We performed this review in two phases. Phase-1 identifies the information resource (search engine) and keywords to execute a query to

obtain an initial list of articles. In a second phase, the initial list is filtered under specific selection criteria, and the most related and core articles are stored into the final list reviewed in this article. The purpose of this review article is to answer the following questions: 1) What are the methodologies used by the community to obtain labeled network traffic data sets? 2) What are the recent trends for network traffic labeling methods? 3) What are the characteristics of the established traffic connection labels? 4) What are the benefits and drawbacks of each labeling methodology adopted? 5) What is the future scope of research for creating labeled network traffic datasets?

The queries include terms related to capturing and labeling network traffic datasets and were the result of the authors' experience, as well as the terminology used in prominent literature in this area [4, 22]. The terms used during the first stage of the methodology are *traffic dataset*, *dataset labeling*, *intrusion detection*, *network classification*, *dataset creation*, and *labeled dataset*. All the queries aimed at being descriptive for including the creation of network traffic datasets and labeling methodology. Whenever a combination of keywords from both categories was found in the text, the corresponding item was selected as a possible candidate.

The selected queries were applied on the all the best known scientific databases: Scopus [23], Google Scholar [24], IEEE Explorer [25], ACM Digital Library [26], Microsoft Academic Search [27], Springer [28] and Mendeley [29]. In addition, the proceedings of some of the most important conferences and journals in the field (DEFCON [30], USENIX [31], IPOM [32], CCS [33], Computers & Security [34], VizSec [35], EUROSYS [36], and others) were specifically analyzed and also a full-text keyword search was applied.

The first stage of the methodology results in a list of 100 candidate articles. The initial selection criteria for articles were intentionally made with weak constraints so as not to exclude relevant articles and to create a large candidate set. Because of these weak constraints, the initial list contained many false positives that did not meet the predefined criteria. Therefore, in the second stage, the initial list of reviewed and filtered according to i) the generation or capture of labeled network traffic datasets and ii) the methodology for network traffic labeling. To sum up, for an article to be included in this study, it must describe a methodology or framework for getting labeled data related to network traffic traces. On the other hand, those works focused on the creation of unlabeled network traffic datasets or not explicitly stating the labeling methods are omitted from this review. At the end of the second stage, a final list with less than 30 articles was obtained. Basic

information (including publication year and type) about the selected articles during phase-1 and the resulting articles after applying a more strict exclusion criteria is shown in Figure 1. Despite the short number of articles found on the generation of labeled datasets on network traffic, more articles are expected to be published in the future. Especially, given the current interest in the development of machine learning approaches in several other fields.

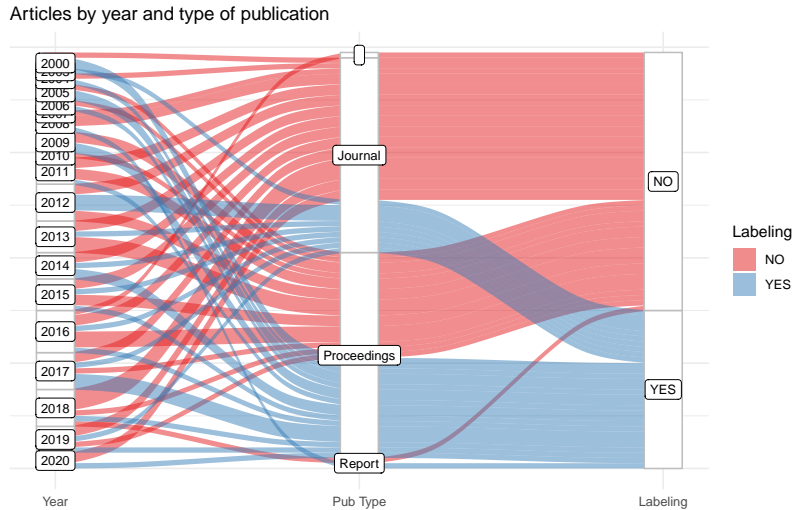


Figure 1: Publication year and type about the selected articles during phase-1. Articles focused on the methodologies for labeling network traffic datasets are highlighted in blue.

3. Background

Labeling consists of adding one or more meaningful and informative tags to provide context to data [37]. In the last years, quality dataset labeling has emerged as a fundamental aspect in the application of machine learning models in several areas. The network security field has been focused in the development of NIDS based on machine learning (referred as SNIDS) with the promising goal of achieving better performance detection [4, 22]. Consequently, the community has focused in the generation of labeled datasets for analyzing different machine learning approaches in the building of SNIDS.

This section provides a brief description of SNIDS and the need of quality labeled datasets. Followed by a brief discussion about the limitations in the current most relevant labeled datasets used for network security. The section

ends with a taxonomy for strategies used to create labeled traffic network datasets.

3.1. Statistically based NIDS

A simplified NIDS architecture is shown in Figure 2. In the first stage, the traffic data acquisition module continuously monitors the traffic, gathers all the network traces on the wire. Then such traces are evaluated by the Incident detector module based on knowledge provided by some predefined Traffic Model. When an incident is detected, an alert is raised, and the suspicious network traces together with information related to the incident are sent to the Response Management module for further expert analysis.

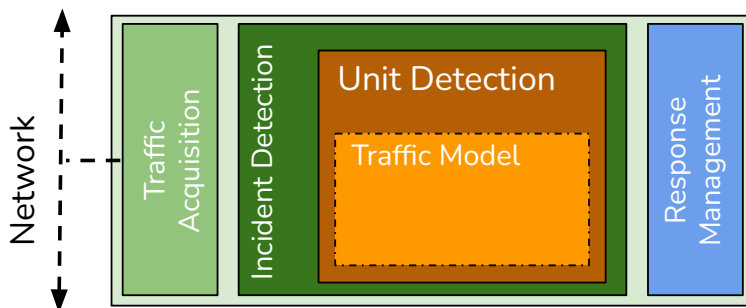


Figure 2: A simplified NIDS architecture (adapted from [4])

The traditional approach for building a traffic model consists of including a set of rules describing malicious behavior [38, 39]. One of the major inconvenience of rule-based approaches is that rules are capable to recognize only known attacks. Another issue is that rules must be regularly updated by the security experts [4]. An alternative approach consists of using a statistical learning model. Under such approach, different network traffic behavior (Email checking, regular social network interaction, Botnet command & control channel, SPAM, etc.) can be represented as a quantitative response Y while information extracted from network traces such as IP addresses, destination ports, or the number of TCP connections in the last minutes (just to mention a few) are referred as the p different predictors, $X_1, X_1 \dots X_p$. Statistical models assume there is some relationship between Y and X_p , which can be written in the very general form:

$$Y = f(X) + \epsilon \tag{1}$$

Where f is an unknown function of X_p that can potentially recognize different network traffic behaviors. In essence, statistical learning refers to a set of approaches for estimating f . One of the most successful methods for estimating f is the so-called supervised learning, where for each observation of the X measurement(s) there is an associated response measurement for Y . However, the performance of statistical learning models following a supervised method will depend on the accuracy and representativeness of the label Y .

The inclusion of statistical and machine learning techniques into a NIDS eliminates the need to manually create rules describing traffic behavior by automatically building them from some reference data [40]. Another major benefit provided by these methods consists of being able to detect not only known attacks but also their variations. On the other hand, a major inconvenience with statistically-based NIDS is they require a large amount of labeled network traces to build a traffic model. The difficulty associated with the labeling process of network traffic datasets is one considerable obstacle in the widespread adoption of SNIDS [4, 22].

3.2. Limitations of current Network traffic Datasets

When choosing a network traffic dataset to train or test a SNIDS it is necessary to consider the representativeness and accuracy of the network events included. However, obtaining representative and correctly labeled network traffic data sets could be very challenging. Moreover, maintaining such data sets could be prohibitive. The fact is that those organizations capable of producing and publishing representative and accurate data are not very comfortable about the risk of potentially exposing sensitive information. On the other hand, any effort to anonymize data is often considered prohibitively expensive.

For more than 20 years since the first DARPA dataset was published in 1998 [41] there have been numerous published datasets for network security. In a recent survey about labeled datasets for network security, Kenyon et al. [42] enumerates several common flaws in the major public labeled datasets. With more than 27 datasets surveyed, the lack of labels accuracy and network representation emerge as the most required properties of a high-quality dataset. However, some authors [43] also observe that most of the labeled datasets available for research represent the network behavior for a particular period. Given the rate of change in malicious behaviors, and the challenge to create and maintain, these labeled datasets become quickly obsolete. The

previously described situation difficult for statistically-based NIDS to generalize its performance to not previously observed attacks. Therefore, more than having only a limited number of high-quality but static labeled datasets, the focus must be on an accurate labeling methodology capable of continuously generating a representative dataset based on network traffic.

3.3. A Taxonomy for Labeling network Security Datasets

Six fundamental categories are considered in the analysis of the labeling methodologies for network security datasets. Figure 3 provides an overview of the six categories and a classification according to three aspects. i) The data resources used as input, ii) The characteristics of the resulting labeled dataset, and iii) the tools and techniques involved during the labeling process.

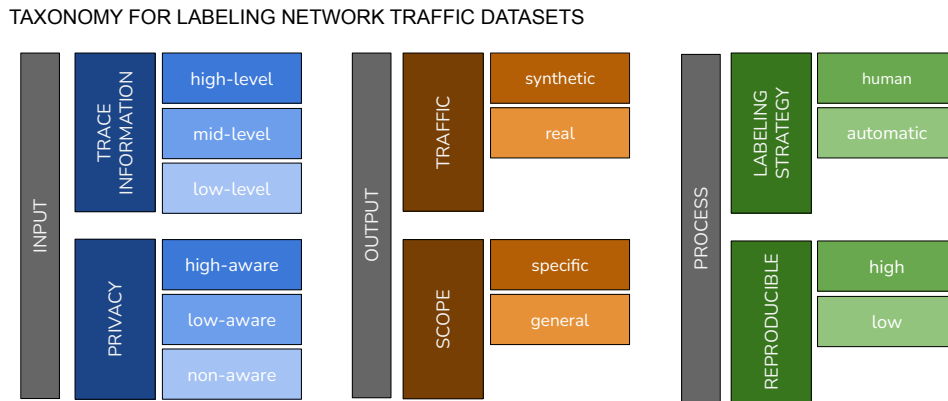


Figure 3: Proposed taxonomy for labeling Network traffic datasets

Traffic: the network traffic in the labeled dataset can be categorized into real or synthetic data. The latter refers to data captured from real networks while the former to data artificially generated with the goal of capturing different network conditions.

Scope: network traffic datasets can be categorized as *specific-scope*: when the labeling approach focus on a particular network behavior including both normal and malicious (e.g. Garcia et al. [44] aims at capturing only botnet behavior) or as *general-scope*: when no particular consideration has been made during the labeling of the traffic data.

Trace information A labeling method can be designed for working at different levels on a network trace. At *low-level*, labeling focuses on information directly extracted from a network trace, such as a list of IP addresses, ports, and network connections, among others. At a *mid level*, the labeling is carried out on network flows level (i.e. labels are per-flow). Finally, at a high-level, labeling is conducted on aggregated information considering interactions or relations between different IP addresses, network flow, user-level applications, etc. The methodology and tools involved in the labeling process will depend on the trace information level under analysis. For instance, high-level trace information such as IP address interactions can be represented as a graph structure. The labeling of such complex structures can require a more elaborated analysis than linear structures such as a list of IP addresses.

Privacy Privacy preservation is another fundamental aspect to consider during the labeling process [45]. Since dealing with anonymized or encrypted data implies losing potentially valuable information, a labeling methodology will need to be prepared for working under such circumstances. Three categories are considered: *high-privacy-aware*, the labeling methodology is capable of dealing with encrypted or anonymized data. *low-privacy-aware*, the labeling is conducted on the network data not containing the payload. However, sensitive user information such as IP addresses and the port destination remains available, and the last category is *non-privacy-aware* when all the network trace information is available during the labeling process.

Labeling strategy: Two types of labeling methods are considered: *human guided labeling* based on human interaction and *automatic labeling* that uses controlled traffic environments. Human guided labeling includes the so-called *manual labeling* which relies only on human expertise (i.e., traditional network traffic analysis with the aid of simple visual charts), and *assisted labeling* which use interactive applications (i.e., a model for recommending labels along with interactive visualizations) Among the three strategies, automatic labeling is the most widely accepted. The general idea behind automatic labeling is to set up a controlled network environment and use the knowledge about the environment to label the traffic.

Reproducibility A *low* reproducible methodology is designed for being

applied only once and producing a unique data set. A *highly* reproducible labeling methodology aims at giving the possibility to a different research team to extend the resulting dataset. For supporting reproducibility, a labeling methodology should provide detailed information about the tools and resources used during the labeling process.

4. Current Methods for Labeling Network Traffic

The present section analyzes all the articles collected based on the methodology described in Section 2. The reviewed articles are organized according to the three labeling methods described in section 3.3. A summary table with a systematic analysis is presented at the beginning of each section. Each table provides details about the all aspects mentioned in the taxonomy. Then, for each piece of research, a brief description of the labeling approach is provided with a particular focus on the tools and strategies used for conducting the labeling.

4.1. Automatic Labeling

In general, under an automatic labeling technique, the creation of a data set in a controlled and deterministic network environment facilitates recognizing anomalous activities from normal traffic, thus eliminating the process of manual labeling by experts (see Figure 4)

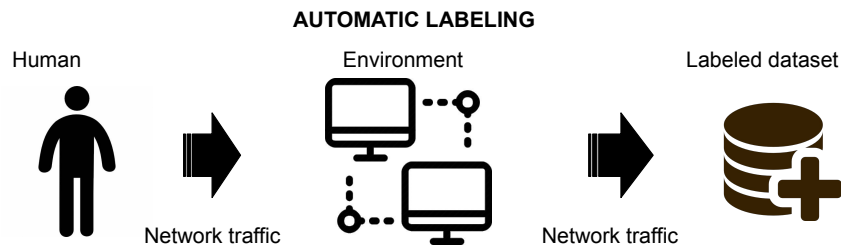


Figure 4: Under automatic labeling methods labels are the result of monitoring a controlled environment (network infrastructure) by a human (user). By having precise control of the environment, the labeling process can be systematized.

In the last years, several network security researchers have embraced automatic labeling strategies with the help of techniques based on Injection

Table 1: Summary of the methodologies using an Automatic strategy for labeling network traffic. Columns four to eight refer to Reproducibility (**Prepr.**), Scope, Traffic Type (**Traffic**), Privacy Awareness (**Privacy**) and Traces Information (**Trace**), as was discussed in the taxonomy.

AUTOMATIC LABELING							
Author	Year	Technique	Repr.	Scope	Traffic	Privacy	Trace
Garcia [44]	2014	IT	low	specific	synthetic	non	mid
Bhuyan [46]	2015	IT	low	general	real	non	low
Moustafa [47]	2015	IT + NST	low	general	synthetic	non	low
Haider [48]	2017	IT + NST	low	general	synthetic	non	low
Mukkavilli [49]	2016	IT+ BP + NST	low	general	synthetic	non	low
Lemay [50]	2016	IT	low	specific	synthetic	non	low
Sharafaldin [6]	2018	BP + IT	low	general	synthetic	low	low
Shiravi [51]	2012	BP + IT	low	general	synthetic	low	low
Lippmann [41]	2000	BP + NST	low	general	synthetic	non	low
Stolfo [52]	2000	BP + NST	low	general	synthetic	non	high
Catania [53]	2012	NST	low	general	real	non	low
Gargiulo [54]	2012	NST	high	general	real	low	low
Song [55]	2011	NST	low	general	real*	non	low
Sperotto [56]	2009	NST	low	general	real	low	low
A-Navarro [57]	2014	NST	low	general	real	non	low
Ring [58]	2017	NST + BP	low	general	synthetic	high	mid
Sangster [59]	2009	NST + BP	low	general	synthetic	non	low

Timing (IT), simulated Behavioral Profiles (BP), and commonly used Network Security tools (NST). Table 1 summarises the surveyed articles using automatic labeling techniques. In addition to the particular labeling technique, the table also provides relevant information about the remaining aspects mentioned in the taxonomy of section 3.3.

4.1.1. Injection Timing

One of the most successful methods for labeling network traffic datasets consists of generating different network traces at specific time windows. Then label all the network traces accordingly to the specific target behavior. This technique is known as Injection Timing (IT) [50].

The injection timing strategy requires a controlled network environment where the user has precise information about the different applications generating traffic on the network. Figure(Figure 5) provides a simplified overview of the injection timing strategy. At t_s the network has just become operational for the first time. At the time t_{sm} the user injects into the network

particular malware traffic (DDOS, Botnet, port scanning, etc.) Since the network has just become operational. All the background traffic from the time window from t_s to t_{sm} is labeled as normal. Beyond t_{sm} all network traffic is labeled as malicious. When the user stops injecting the malicious behavior at t_{em} , all the traffic becomes labeled as normal again until the network is permanently shut down at time t_e .

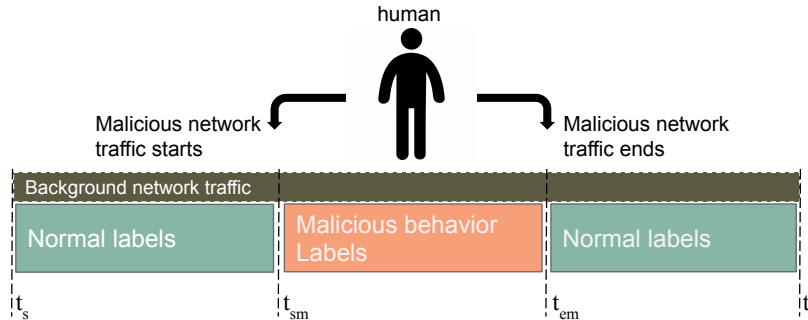


Figure 5: An example of the Injection Timing labeling strategy using specific time windows for injecting and labeling malicious network traffic.

Since the Injection Timing strategy is applied under controlled environments, it is possible to create labeled datasets with numerous network traffic behaviors. This feature provides some level of authenticity for validating the experimental results on the generated labeled dataset. However, since labels are obtained by merely contrasting the execution time window of each generated network trace, a strict time control mechanism is necessary for obtaining accurate labels.

A clear example of the application of the injection timing strategy is observed in the work of Garcia et al. [44]. In particular, the authors use Injection Timing for generating a labeled dataset focused on Botnet attacks (The CTU-13 Dataset). A topology consisting of a set of virtualized computers with the Microsoft Windows XP SP2 operating system on a Linux Debian host was used for capturing the traffic through time windows.

For the particular case of Botnet network behavior, with Injection timing, it is easier to label these network traces with higher accuracy than other types of attacks. The fact is that Botnets tend to have a temporary and very localized behavior, which means that most actions remain unchanged for several minutes. Therefore, the separation of the traffic into time windows facilitates the control of the botnet behavior of the network.

Following a similar approach, Bhuyan et al. [46] use a subset of the TUIDS (Tezpur University Intrusion Detection System) testbed network for capturing normal and malicious traffic traces. Normal traffic is collected independently from real users of the networks. At the same time, several types of malicious network traffic are injected by infecting several network stations. All the network traces from these stations are then captured considering the specific time intervals. Then after a pre-processing to extract traffic from those infected stations, all network traces are mixed using the time interval windows of each connection to classify them as either *malicious* or *non-malicious*.

Tools like IXIA PerfectStorm are also used for generating labeled data. By using this tool, it is possible to generate up to 9 families of malware. The Australian Defence Force Academy rely on the IXIA PerfectStorm tool for its labeling strategy. The captured traffic from IXIA perfectStorm and label the traces using the windows time interval and the attack information reported by the tool. This strategy was used for generating the UNSW-NB15 (Moustafa et al. [47]) and NGIDS-DS ([48]) datasets.

On the other hand, Mukkavilli et al. [49] focus on labeling traffic interaction between users and cloud services (i.e. Amazon EC2). The authors rely on the PlanetLab infrastructure. PlanetLab is a (now-extinct) group of computers available as a testbed for computer networks and distributed systems [60, 61]. The authors use specific PlanetLab nodes mimicking normal users and their interaction with cloud services based on normal and uniform traffic distributions. All traffic from normal nodes is labeled as normal. On the other hand, malicious behavior is injected at different time windows to generate malicious traffic. In particular, DDOS, port scanning, and ARP spoofing attacks take place at random intervals, while normal behavior traffic continues to run continuously. All attack nodes have the precaution to start the malicious behavior within the same time window with a minimum delay and labeled accordingly. Several other authors follow a similar approach combining tools for simulating normal behavior with injection timing techniques for capturing malicious [58, 62, 6].

The work of Lemay et al. [50] applies an Injection Timing strategy for labeling the traffic from SCADA (Supervisory Control and Data Acquisition) systems. Due to the sensitive nature of these networks, there was little publicly available data. Through the use of pre-established and simulated network architecture, the authors could generate Modbus [63] network traffic with precise knowledge about the behavior observed on each network trace

type. Then if a packet is part of a trace group that includes malicious activity, it will be labeled as 'malicious.' Otherwise, it is labeled as 'normal.'

4.1.2. Behavioral Profiles

The use of Behavioral Profiles is another strategy used for automatically labeling network traffic. Behavioral profiles provide the information to simulate a specific feature or aspect of the network. A profile encompasses an abstract representation of features and events of real-world behaviors considered from the network perspective [51]. Therefore, profiles are usually implemented as computer programs executing common tasks according to some previously defined mathematical model (usually a probability distribution). These profiles are then used by human agents or operators to simulate the specific events in the network. Their abstract property effectively makes them network-agnostic and allows them to be applied to different setup and topologies. Thus, the labeling process using this technique is straightforward; all the traffic generated by a profile simulating normal traffic will be labeled as normal. Similarly, all the traffic generated from a malicious behavioral profile is labeled as malicious.

In this way, Shiravi et al. [51] combine two classes of profiles to generate a labeled dataset with different characteristics and events. A first profile A tries to describe an attack scenario unambiguously. While a second profile - B - encapsulates distributions and mathematical behaviors extracted from certain entities and represented as procedures with pre and postconditions, thus representing normal traffic. Examples include the statistical distributions of packet sizes of a protocol, number of packets per flow, specific patterns in the payload, size of payload, the request time distribution of protocol.

Sharafaldin et al. [6], also focus on two behavioral profiles. An abstract benign profile is built upon 25 user behaviors based on the HTTP, HTTPS, FTP, SSH, and email protocols. The benign profile is responsible for modeling human interactions' abstract behavior and generating naturalistic benign background traffic. Six malicious profiles are generated based on frequent attacks. Then, by combining these profiles, several labeled datasets can be generated, each one with unique characteristics for the evaluation. By merely altering the combination of the profiles, it is possible to control the composition (e.g., protocols) and statistical characteristics (e.g., request times, packet arrival times, burst rates, volume) of the resulting data set.

Other works [41, 64, 58, 49, 59] combines behavioral profiles with other techniques for improving the representativeness of the resulting labeled datasets.

In particular, Lippmann et al. [41, 64] proposes the use of automata to simulate traffic behaviors with a traffic distribution similar to the observed between a small Air Force base and the Internet. Through custom software automata in the test bed, hundreds of programmers, secretaries, managers, and other types of users running common UNIX and Windows NT application programs are simulated. Automata interact with high-level user application programs or they implement clients for network services such as HTTP, SMTP, and POP3. Low-level TCP/IP protocol interactions are handled by kernel software and are not simulated.

4.1.3. Network Security Tools

The labeling process is carried out based on the information provided by network security tools (NST) such as sniffers, honeypots, or even using a NIDS.

The application of a NST labeling strategy was one of the strategy applied in the generation of the DARPA datasets (1998-99) [41, 64] and the KDD99 [52]. As part of the DARPA IDS evaluation program, a testbed was created with many types of live traffic using virtual hosts to simulate a small Air Force base separated by a router from the Internet. Different types of attacks were conducted outside the network and captured by a sniffer located in the network router. Any network trace coming outside the network (the internet) is considered as malicious while those from inside are normal.

A similar approach was more recently applied by Ring et al. [58] combining a virtualized network environment with traces captured from a real web server exposed to the internet. Normal traffic is generated through behaviors profiles while Malicious traffic comes from the external server as well as particular attacks injected from the virtualized infrastructure. The authors applied anonymization techniques netflow information for guaranteeing privacy.

NST tools are also applied for labeling traffic in capture-the-flag competition. Ideally, capture-the-flag competitions are a valuable source for gathering distributed normal and malicious traffic. However, by default, data sets do not contain labels. Sangster et al. [59] apply a set of pre-established rules and user roles in combination with network sniffers to capture several traffic behaviors. Then, based on that information, they provide the correct label to each network trace. The malicious traffic is captured from specific computers belonging to NSA security experts. On the other hand, Normal traffic is generated artificially using profile behaviors.

Gargiulo et al. [54] and Catania et al. [53] use rule-based NIDS for generating labels. In particular, Catania et al. analyze the performance of stand-alone NIDS for labeling traffic and provide some results when results labels are used training SNIDS. On the other hand, Gargiulo et al. use the principles of Dempster-Shafer’s theory for combining the information of several rule-based NIDS. In their approach, the authors use the basic probability assignment for calculating the final decision about a particular network trace. The resulting labels are then used for training a SNIDS.

The work of Navarro et al. [57] propose a similar strategy for automatic labeling 802.11 network traffic using a NIDS based on unsupervised anomalies. The NIDS analyzes the traffic, and for each of the network traces, the system provides three numerical values with information about the label’s belief. The belief values represent the probability of observing normal or attack behaviors. A third value is used for registering how uncertain the system is about network label and adjusting the system accordingly. The labeling process is established by using a threshold of possible values per label. The threshold is set using the mean and standard deviation of the probability values set by NIDS. Through this threshold, all network traces whose label value is not within this threshold will be discarded from the dataset due to the degree of belief they present. Connections within the value range will be labeled according to the NIDS’s highest probability between the Normal and Malicious classes. Thus, the resulting labeled dataset will contain those connections that NIDS determined with a high degree of confidence.

On the other hand, Sperotto et al. [56] with TWENTE, and Song et al. [55] with KU aim to provide the security community with more realistic data sets. Their labeling method is based on the analysis of several honeypots with different architectures inserted within a network environment. Then, all captured traffic to specific monitored services in the honeypots can be easily labeled as malicious. By using honeynets, there is no human interference during the data collection process (i.e., any form of attack injection is prevented). Therefore, the attacks present in the dataset reflect the situation of a real network.

4.2. Human-Guided Labeling

Many authors [65, 66, 67, 68] consider human experience is an essential aspect of traffic analysis and the subsequent connection labeling. Therefore, under human-guided labeling methods, the network environment is not controlled and all the work relies in the expert users. However, since experts are

an invaluable resource, labeling time has to be efficiently used (see Figure 6).

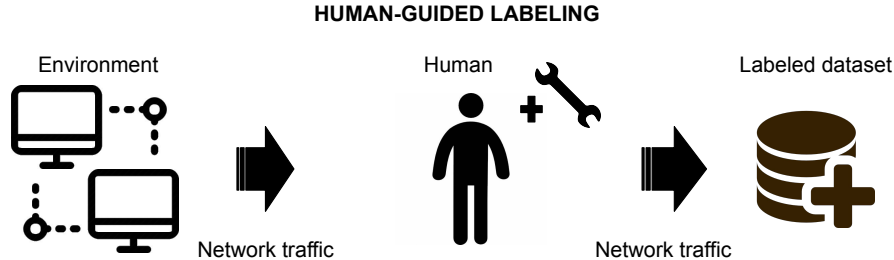


Figure 6: Under human-guided labeling methods, the environment (network infrastructure) is not controlled by a human (user). Labels are the result of human knowledge with the eventual assistance of particular tools.

4.2.1. Manual

A very significant percentage of today’s network analysis is performed manually (i.e., without assistance from any system) by security experts [62]. Manual labeling by network traffic experts requires a precise understanding of the network behavior for differentiating between malicious and normal traces. Unfortunately, many of these extensive network analysis processes are not published, and despite being widely used, the research community has limited knowledge about it.

There are several approaches to reduce human effort in the manual labeling process. Some authors propose the use of visual tools (Viz) to improve traffic behavior analysis. Others suggest collaborative environments between a label prediction model and security experts (Crowd). A summary of the most relevant approaches are presented in Table 2.

Many works try to reduce the effort involved during manual labeling through the use of visualization tools. In particular, the use of visual systems help the user during labeling by improving correlation between malicious patterns and making the user more confident about their labels [74].

NIVA [72] is one of the first examples of a data visualization tool for intrusion detection. NIVA uses information from various intrusion detectors and incorporates references and colors to give the attacks a significant value. The color of the reference represents the severity of the attacks. Yellow is moderate, while red is the most severe.

Table 2: Summary of the methodologies using a manual strategy for labeling network traffic. Columns four to eight refer to Reproducibility (**Prepr.**), Scope, Traffic Type (**Traffic**), Privacy Awareness (**Privacy**) and Traces Information (**Trace**), as was discussed in the taxonomy.

MANUAL LABELING							
AUTHOR	YEAR	TOOL	REPR.	SCOPE	TRAFFIC	PRIVACY	TRACE
Koike [69]	2006	Viz	low	general	real	low	low
Livnat [70]	2005	Viz	low	general	real	low	mid
Ren [71]	2005	Viz	low	general	real	low	high
Scott [72]	2003	Viz	low	general	real	low	high
Chen [73]	2014	Viz + Crowd	low	general	real	low	mid
Huang [20]	2020	Crowd	low	general	real	non	mid

On the other hand, IDGraphs [71], uses the visualization technique called *Histograms*: a visual technique to map the brightness of a pixel to the frequency of the data. By mapping multiple combinations of features in the input data, attacks with different characteristics can be identified. IDGraphs not only shows an overview of the underlying network data, but also allows an in-depth analysis of possible anomalies through dynamic queries [62].

Livnat et al. [70] develop a chord-based visualization for the correlation of network alerts. The approach is based on the notion that an alert must possess three attributes: what, when, and where. These attributes can be used as a basis for comparing heterogeneous events. A network topology map is located at the center with the various alert records in a surrounding ring. The ring’s width represents time and is divided into several periods of the history of each connection. A line is drawn from an alert type on the outer ring to a particular host on the topology map to represent a triggered alarm. Thicker lines show a more significant number of alerts of a single type, and the larger nodes in the topology map represent hosts that experience unique alerts.

The authors of IPMatrix [69] believe that an attacker’s IP address, even if falsified, is a significant factor in an attack, and administrators can take appropriate countermeasures based on it. Using a combination of heatmap and scatter plots, IP Matrix represents the full range of IPs. IP Matrix incorporates two 256x256 matrices. The first, the *Internet level* matrix, only maps the first two octets while the *local level* matrix maps the last two octets, allowing the local and Internet level IP addresses to be seen simultaneously. Each alert generated by an IDS is mapped using a pixel

within its appropriate cell. Pixels are color-coded to represent attacks of different nature, but because a pixel is too small to be seen, the background of a cell is colored with the most frequent attack type. A disadvantage of this system is that there are no connections between the local level and the Internet hosts, which makes the system less intuitive.

Other manual approaches make use of crowdsourcing (Crowd) for obtaining labeled datasets. Crowdsourcing has proven to be a cost-effectively way to obtain a large-scale labeled dataset. Moreover, in some fields, it has been demonstrated that nonexpert annotations were relatively useful for training a statistical model [75].

Chen et al. [73] introduce the OCEANS (Online Collaborative Explorative Analysis on Network Security) system, which integrates visual analytics methods and collaboration features as a web application. In particular, OCEANS integrates the crowd input from security experts and makes everyone contribute to labeling the network events. OCEANS visuals offer detail of individual network flows, including IP, port, time, and network attributes from both source and destination side, as well as health status and IPS logs. OCEANS provides a web interface where a user can submit events while others can view and comment on them. Users need to provide label information describing the event. Then all the crowd input is synthesized into an event graph and event timeline. The tool provides a suspicious score based on the number of events an IP address involved, adding the count of agreement on this event and subtracting the count of disagreement.

More recently, Huang et al. [20] developed and released IoT inspector. An open-source application for capturing and labeling network traffic from smart home devices. The application crowdsources the data from within home networks and provides a mechanism for simplifying the labeling. The resulting dataset is not focused on malicious behavior, but on modeling the particular behavior of the different brands of smart devices. Contrary to desktop computers, smart home devices perform very specific tasks making their networking behavior very predictable [76].

4.2.2. Assisted

To facilitate the analysis and subsequent labeling of network traffic by experts, several authors have proposed using a technique called Active Learning (AL) [77, 78].

AL refers to human-in-the-loop methods, where a prediction model is iteratively updated with input from expert users (see Figure 7). The expert

user (a) is responsible for taking decisions on those connections where the model has a high degree of uncertainty (e) (i.e. those connections near the decision boundary) than expected (a strategy known as Uncertainty Sampling [77, 79]). These final decisions are used for labeling the data (b) fed into the model (c) to improve its objective function and prediction performance on unlabeled data (d).

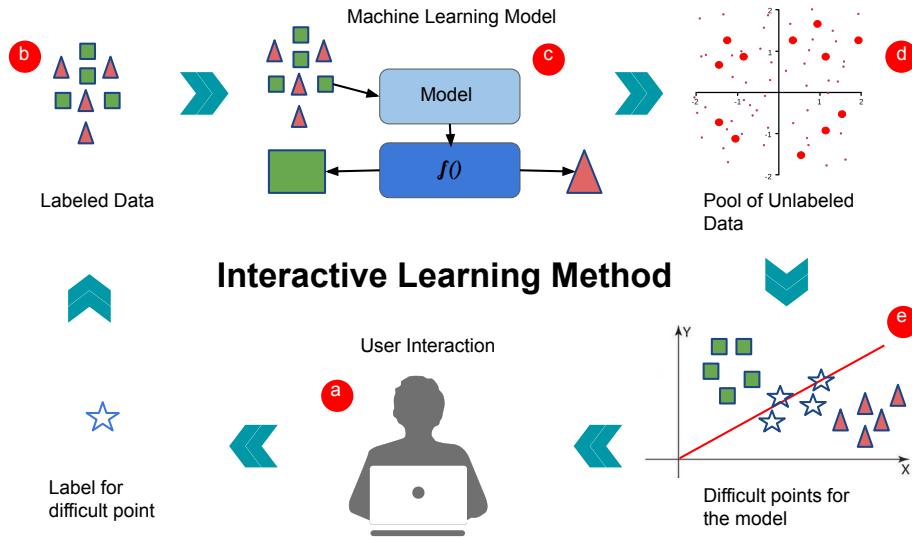


Figure 7: Work cycle of an Active Learning Strategy used for labeling

A summary of the most relevant approaches is presented in Table 3. In some cases, AL is combined with other labeling tools. In particular, the combination of Visual (Viz) and AL techniques has emerged as an effective approach for labeling network traffic.

Classical AL techniques are widely used in labeling large volumes of data in general, and it has started to be used for constructing labeled network traffic datasets.

In their work from 2004, Almgren and Jonsson [80] propose a classical AL strategy based on uncertainty sampling [77, 79] to select the most suitable network traces to be labeled by the expert users.

On the other hand, other works attempt to accelerate the AL working cycle by including several strategies for improving the quality of the network data to be labeled by expert-users. Stokes [65] includes a rare category detection algorithm [83] into to AL work cycle to encourage the discovery

of families of network traces sharing the same features. Similarly, Görnitz uses a k-nearest neighbors (KNN) approach to identify various network trace families. Both approaches guarantee that every family has representative members during the expert labeling process and reduces the sampling bias. Beaunon et al. [81] also rely on rare category detection to avoid sampling bias. Moreover, they apply a divide-and-conquer strategy during labeling to ensure good expert-model interaction focused on small traffic sections.

Similarly, McElwee [78] proposes an AL intrusion detection method based on Random Forests and k-Means clustering algorithms. The daily events are submitted to a Random Forests classifier and events receiving more than 95% of the votes are considered correct and saved into a *master* dataset. The remaining events, conforms a candidate dataset grouped into k groups using k-means clustering. Each group is analyzed and classified by an expert and then saved into the master dataset.

Other works combine a visualization component with the AL labeling strategy. The motivation behind including visual components is to improve the user experience during the AL work cycle. A better user experience translates into better quality labels for the prediction model. Xin Fan et al. [67] present one of the most recent approaches combining AL techniques with a visual tool to provide the user with a better representation of the traffic being analyzed. The authors use a graph to display a two-dimensional topological representation of the network connections. The nodes in the graph are differentiated by color to identify the connection type quickly and a color intensity matrix to show the interaction between the connections. Several

Table 3: Summary of the methodologies using a assisted strategy for labeling network traffic. Columns four to eight refer to Reproducibility (**Repr.**), Scope, Traffic Type (**Traffic**), Privacy Awareness (**Privacy**) and Traces Information (**Trace**), as was discussed in the taxonomy.

ASSISTED LABELING							
AUTHOR	YEAR	TOOL	REPR.	SCOPE	TRAFFIC	PRIVACY	TRACE
Almgren [80]	2004	AL	low	general	real	non	high
Beaunon [81]	2017	AL + Viz	high	general	real	low	high
Fan [67]	2019	AL + Viz	low	specific	real	low	mid
Guerra [82]	2019	AL + Viz	high	specific	real	high	high
Gornitz [68]	2013	AL	low	specific	real	non	low
McElwee [78]	2017	AL	low	general	real	low	high
Stokes [65]	2008	AL	low	general	real	non	high

other visual tools such as histograms and boxplots are employed during the labeling process. Histograms are used for representing the percentage of the traffic of the various protocols/ports. Boxplots are used to show the distributions of the destination ports and the number of records of the different IPs

In the work of Beaunon et al. [81, 66], the authors also implements a visual representation for the user interaction process. In this case, the visual application provides a mechanism for organizing the network traffic in different groups. A set of queries and filters facilitates the user to create families of connections for further analysis by small network traffic groups.

Otherwise, Guerra et al. present RiskID [82, 84], a modern application focus in the labeling of real traffic. Specifically, RiskID intend to create labeled datasets based in botnet and normal behaviors. The RiskID application uses visualizations to graphically encode features of network connections and promote visual comparison. A visualization display whole traffic using a heatmap representation based in features. The heatmap promotes the search of pattern inside the traffic with similar behaviors. Other visualization shows statistical report for a punctual connection using color-map, histogram and a pie-chart. In the background, two algorithms are used to actively organize connections and predict potential labels: a recommendation algorithm and a semi-supervised learning strategy (AL strategy). These algorithms together with interactive adaptations to the user interface constitute a behavior recommendation.

5. Discussion

No matter the labeling strategy used, they focused on the accuracy and representativeness of the resulting datasets. However, despite their frequent use, there are still substantial problems inherent to the labeling methodologies. Significant aspects such as privacy, reproducibility, and the level of expertise required are not discussed in depth during the implementation of each strategy. Table 4 summarizes the more significant aspects of the three labeling strategies.

5.1. Automatic Labeling

Automatic labeling strategies are the preferred approach to obtain labeled network traffic data. Such a decision responds to the low level of expertise

Table 4: Benefits and drawbacks of the strategies for labeling network traffic datasets.

Labeling Strategy	Benefits	Drawbacks
Automatic	<ul style="list-style-type: none"> • Very fast • Easy to adapt to new specific behaviors • Low expertise • Moderate accuracy 	<ul style="list-style-type: none"> • Low representativeness • Hard to reproduce • Non Privacy
Manual	<ul style="list-style-type: none"> • High representiveness • High Accuracy 	<ul style="list-style-type: none"> • Slow • High expertise • Hard to reproduce • Low Privacy
Assisted	<ul style="list-style-type: none"> • Fast • Medium expertise • High representiveness • Moderate Accuracy 	<ul style="list-style-type: none"> • Hard to reproduce • Hard to adapt to new specific behaviors • Low Privacy

required and the relative speed for generating large volumes of labeled network traffic. The fact is that automatic labeling strategies do not require a high level of expertise compared to manual labeling techniques.

Among all the automatic labeling strategies, the Injection Timing strategy is the simplest and straightforward. Unfortunately, this strategy shows several limitations regarding the critical representativeness required in the data. The main limitation is that malicious and normal traffic activities were usually captured from two different and uncorrelated environments. When both captures are merged and collectively analyzed, it could be easy to discriminate malicious from normal traffic. The background traffic, routing information, and the hosts present in the network are some aspects to be considered when capturing network traffic from several sources. Another significant issue with injection timing is the lack of a clear approach for supporting privacy awareness. Although, it is theoretically possible to apply several anonymization techniques, the fact is that most of the articles implemented the techniques has not even considered a methodology for protecting the privacy in background and normal traffic ([44, 50, 46]). The only exception is the work of Ring et al. [58] where the authors have discussed IP anonymization techniques during the labeling process.

On the other hand, the labeling process based on Network security tools is usually applied on real traffic, which provides a better representiveness. However, it could be difficult to ensure the required accuracy. As is the case of the work of Navarro et al. [57], and Gargiulo et al. [54] who use a NIDS

based on a set of rules for describing malicious behavior. Both authors use only those connections classified by NIDS with a high confidence rank. These approaches guarantee the reliability of the labels in the resulting dataset but neglect those connections that are difficult to predict and that are very useful to improve detection systems. To mitigate this bias towards easy-to-detect connections, Navarro proposes the use of an expert for manually analyzing and labeling just those connections with a high degree of hesitation.

In general, all reviewed labeling approaches based on NIDS [52, 54, 57, 48, 53] rely on some ruleset that needs to be periodically updated. i.e., Whenever a new variant of a malicious behavior emerges, an expert needs to write a new rule describing such behavior. The fact is that there is no guarantee the traffic generating an alert in NIDS do not contain an attack. Therefore, those supposedly normal traffic traces should be analyzed in depth before added to the final labeled dataset.

The honeynets alternatives [55, 56, 58] provide a straightforward procedure for labeling malicious network traces. However, similarly to the NIDS approaches, it shows serious flaws for labeling normal traffic. The simple rule of considering all traffic captured from the honeypots as malicious [56] does not guarantee the rest of the traffic is free of undetected malicious behaviors.

The fact is that ensuring the quality of the automatic labeling methods remains a challenging task. In Lemay et al. [50], they consider that if a packet is part of a connection including malicious activity, it has to be labeled as malicious. Otherwise, it is labeled as normal. However, when an attacker connects to an FTP service for sending an *exploit*, not all the traffic contains malicious behavior. Under a deeper inspection of packet capture, it can be argued that the TCP connection needed to connect to the service to send the *exploit* is not malicious. After all, the connection procedure is no different from other legitimate connections established by other clients to the server. In that case, only the packets containing the actual *exploit* should be labeled as malicious. A similar problem can be found in Bhuyan et al. [46], where the authors attempt to generate normal traffic with varied characteristics from traffic captures of users' daily activities. Malicious traffic is generated by launching attacks and infecting different users' servers. Under this scenario, it is not easy to guarantee that all the traffic captured comes from users is normal. The fact is that considering that the network is clean before the first attack occurs is a mere assumption. To sum up, Automatic Labeling methods provide a fast and simple approach for generating a considerable amount of labeled traffic. They can easily adapt to

new behavior without a high level of expertise. However, the deployment of the infrastructure for capturing and labeling traffic could be difficult to reproduce. Moreover, it is clear that despite all the precautions during the generation of synthetic traffic, these methods still have serious drawbacks regarding the level of representativeness and label accuracy. Ideally, a network traffic labeled dataset should not exhibit any inconsistent property of the network infrastructure and its relying traffic. The traffic must look as realistic as possible, including both normal and malicious traffic behaviors. In particular, traffic data should be free of noise and not include any leakage caused by the selected labeling strategy. Therefore, the Automatic Labeling method should implement a detailed specification of the capture processes to provide coherent and valuable traffic data.

5.2. Human-guided labeling

In general, the manual labeling methods generate datasets with good representativeness and accuracy. The main inconvenience relies on the difficulty of labeling the traffic volume required for current SNIDS needs. Users with high expertise are a fundamental resource during the labeling process. Recent approaches including visualization techniques and interactive labeling methods have emerged to facilitate the incorporation of users with a lower degree of expertise.

However, those manual labeling approaches relying only on visualization suffer from the same drawback [69, 70, 71, 72]. They still require a high level of expertise for performing the actual classification. Despite having attracted considerable attention for identifying malicious activities [62], their adoption in real-world applications has been hampered by their complexity.

Human-guided methods based on AL strategies aim at improving the speed of the labeling process while keeping high representativeness of the resulting data. The inclusion of a statistical learning model can be a valuable tool for helping the user during the decision process. Moreover, some of the approaches [67, 82] claim the expertise required for using such systems is reduced. Nevertheless, the role of the expert remains a fundamental aspect for guaranteeing the quality of the labels. The expert is responsible for generating the initial set of labels required for training the prediction model. Moreover, the expert is responsible for labeling during the AL working cycle when a connection is difficult to discriminate between normal or malicious. The precision of these labels could impact the overall accuracy of

the recommendations made by the relatively simple models based on Logistic Regression [66], Fuzzy c-means algorithm [67] or Random Forests [78].

Manual labeling strategies are difficult to reproduce and extend to new traffic behaviors. In many cases, these strategies will rely only on the ability of the expertise of the user to recognize new behaviors. Similar is the case of assisted approaches, although to a minor degree. In some cases, if the distribution of the new traffic behavior significantly differs from the known distribution, the prediction model has to be retrained to recognize new behaviors. Moreover, when very focused visualization techniques are combined with AL, adapting them to new traffic behavior could not be straightforward. On the other hand, privacy awareness under the surveyed manual approaches remains under minimal standards. None of them discuss the consequences of traffic encryption or anonymization during the labeling. However, in many cases, the labeling is conducted through observing mid-level trace information such as net flows [70, 67, 20, 73], which indicates that payload information is not available during labeling. Similarly, complex visuals such as [71, 72] are suitable for hiding considerable private information and still being useful for labeling. Not differently is the case of assisted labeling strategies, where most of them seem not specially prepared for dealing with privacy mechanisms. Only the work of Guerra et al. [82] have considered the inclusion of anonymized network traces during the labeling process.

To sum up, all the human-guide labeling methods seem to be more well-suited for label network traffic with high accuracy and representativeness. However, despite the considerable improvements, these strategies still show several issues regarding the capacity for rapid and continuous labeling of network traffic.

6. Open Issues

6.1. Deficiencies in the representativeness of labeling strategies

Since DARPA [41, 64], there have been several attempts to improve the quality of network traffic labeled datasets. However, there are still several problems regarding the representativeness of the network scenarios. The fact is that automatic labeling strategies have serious issues for operating on real traffic [44, 50, 46, 51, 6]. Even those strategies using NST such as honeynets which capture real attacks suffer from representativeness problems when they try to incorporate normal traces into the resulting labeled dataset. Therefore, these strategies cannot represent all the details about traffic dynamics and

potential real-world network attacks. As shown in [85], the network traffic differs between lab environments and production networks.

On the other hand, human-guided labeling strategies certainly improve the authenticity of the resulting labels. However, the labeling process is still slow and challenging for obtaining a sufficient number of representative labels for use on current SNIDS implementation.

Privacy preservation is another major issue regarding human-guide labeling strategies. In manual and assisted strategies, the expert has access to all the traffic information from real users. The previous situation is not so critical in automatic labeling strategies, since normal traffic is usually generated artificially [6, 51] or under controlled conditions [44].

A partial solution consisted of applying anonymization techniques during the capture process. Therefore, network traffic can be subjected to encryption or attribute extraction procedures for hiding different portions of the traffic during the labeling process. [86]. Many human-guided strategies rely on this approach for a minimal privacy preservation. Almgren and Fan [80, 67] for instance, perform traffic labeling at the flow level, hiding relevant information such included in the payload. Similarly, Beaugnon et al. [81] perform a complete per-flow feature extraction procedure depriving the community of using the entire network payload. However, the main problem with anonymization is that the removal of valuable information from the network has an impact on the correct representation of network behavior. When dealing with real and representative labeled datasets generation, it is essential to ensure precise and consistent network traffic information. The process requires careful monitoring and capturing of the different aspects of regular traffic, in conjunction with a fast and accurate labeling method for providing the SNIDS and the research community with an adequate dataset.

It seems that the inclusion of collaborative approaches [73] is an aspect that could improve human-guided labeling techniques. Firstly, the incorporation of multiple users in the labeling is a significant improvement of the overall speed of the process. Alleviating one of the main drawbacks of human-guided strategies. Secondly, it incorporates into the process more evaluative analysis on the different behaviors, allowing both differentiation and unification of criteria. In this way, through a kind of voting, labels could be established with greater accuracy while keeping representativeness at a high level. On the other hand, the speed of collaborative labeling techniques impacts AL-based strategies. Since traces are labeled faster, the prediction model gets earlier feedback, which accelerates the phases of the learning cy-

cle.

Another possible path for improving the speed and quality of human-guided strategies is the use of users' labeling history. It would be straightforward for current human-guided strategies [66, 81, 87, 67] to create a matrix of user preferences in relation to the set of traces that make up the traffic. The resulting matrix can be used to build a Recommendation System to focus the labeling on groups of traces with similar characteristics and according to the user's preferences. In this way, the whole labeling process is enhanced.

6.2. Support for systematic periodical updates

Recently some members of the network security community have started to mention that due to the evolution of malicious behavior and the constant innovations in attack strategies, network traffic labeled datasets need to be updated periodically [88, 42]. However, from all labeling strategies in section 4, only a few of them provide a consistent approach to continuously updating dataset information and preventing it from expiring over time.

The automatic labeling strategies require the deployment of a complex network infrastructure. The maintenance of such infrastructures complicates the extension to new behaviors. Moreover, the whole reproducibility of the process is adversely affected, since infrastructure, user profiles, the malware used are usually not precisely described,

On the other hand, some assisted labeling strategies seem to be more adaptable to new behaviors, as they depend on the generalization of their prediction model. Both Beaugnon et al. [66] and Guerra et al. [84] published the source code of their visualization tools together with the AL prediction model. However, the model performance can often decay since predictions are biased to specific network behavior. Updating these models require a continuous execution of the AL working cycle, which demands expert user assistance.

Consistently with the previous section, a collaborative approach can also be applied to guarantee a certain degree of reproducibility during the expert interaction. In the best case, if a network trace received different classifications, but most of them are from a particular behavior, it can be estimated as the correct behavior and finally set the label.

In general, given the volume, velocity and variety characteristics of network traffic [89], it is necessary to move away from strategies that result in static datasets. Having a continuous pipeline for generating accurate and representative labeled datasets is part of the so-called MLOps (Machine Learn-

ing Operations). MLOps [90], is a recent field of machine learning that aims to make building and deploying models more systematic. Current labeling strategies need to incorporate MLOps strategies capable of adapting to current traffic distributions and intrusions approaches and provide modifiable, extensible, and reproducible mechanisms for continuous labeled dataset delivery.

6.3. Lack of consistent validation methodologies

Despite the strategy employed for labeling a dataset, a consistent methodology is necessary to validate its results. The components of this methodology should be adapted depending on the applied strategy.

In the methods based on automatic labeling, the most common evaluation methodology is based on the similarity against real traffic. Several authors [6, 91] proposed similarity metrics for evaluating the resulting datasets. Metrics such as complete network configuration, labeling accuracy, available protocols, attack diversity, and metadata provide a quality standard for a dataset. However, the impact of the labeled dataset quality in creating network behavior classification models remains unknown.

In contrast, the validation of methods based on human-guided labeling is considerably more complex. It is necessary to evaluate the components included in the work cycle and the interaction between them to determine the effectiveness of the strategy. Unfortunately, AL strategies discussed in the 5 section do not analyze the benefits and the problems involved in the work cycle of labeling data. Surveyed articles [65, 66, 67, 78] do not include any process for measuring the accuracy of the prediction model as the AL cycle progresses. Other important considerations, such as the minimum number of labels needed to make accurate suggestions or how the strategy reacts when noisy data is introduced, are not explored in depth.

Similarly is the case for those strategies including visualization tools. The main goal behind these strategies is to assist the user during the labeling process. However, most of the reviewed works considering visualization tools [71, 70, 72, 69, 66, 67] have not evaluated the benefits and usefulness of the proposed visualizations. Fan et al. [67], and Guerra et al. [82] are among the few authors to analyze the performance of different visualization techniques used to improve pattern perception during the interactive process. The fact is that the availability and cost of conducting a validation with expert users and traffic analysts affect the evaluation process. As a result, analytical and

empirical evaluations of the systems often do not provide the information needed to establish the usefulness of the support tools.

It seems critical that the community starts to focus on providing user studies to measure the impact of the tools in the labeling process and get relevant information about the labeling strategy followed by users. Such studies should include information about the expertise level of the users, their interaction with the assistant tools, and the human effort associated with the complete labeling process.

Finally, current labeling strategies must provide an in-depth analysis of the correlation between labeling strategy, label quality, and the final performance of the resulting detection models

7. Conclusions

Labeled dataset generation is a fundamental resource for network security research. However, all current labeling strategies experience significant problems in terms of quality, volume, and speed. There is a trade-off between the quality of the resulting labeled dataset and the amount of network traces included. Automatic labeling method provide a large amount of labeled network traces, but the accuracy and representative could not be guaranteed. Human-guided method are an improvement for the quality of resulting labeled dataset, but since they still heavily depend on user expertise, the speed and volume of labeled data could be insufficient.

A more significant problem is that the current methodologies are oriented to create a static version of the datasets. A static labeled dataset is only suitable for research during a very short time period. The development of a validated methodology including a continuous pipeline for incorporating new representative and accurate network traces is fundamental for continue with the development of network security research. In the case of Statistically-based NIDS, the need of a standard strategy for a continuous generation of quality labeled datasets is entirely accordant with the recent MLOps roles included in the production cycle beyond the network security field.

To sum up, quality labeled datasets are not enough. The network security research community need to standardize the methodology reducing expert-user interaction with focus on reproducible and continuous validation in concordance of the data-centric models used nowadays when deploying machine learning products in real-life scenarios.

8. Acknowledgements

The authors would like to thank the financial support received by Argentinean ANPCyT- FONCYT through the project PICT 1435-2015 and the Argentinean National Scientific and Technical Research Council.

References

- [1] P. A. A. Resende, A. C. Drummond, A survey of random forest based methods for intrusion detection systems, *ACM Computing Surveys* 51 (2018). doi:10.1145/3178582.
- [2] T. R. Glass-Vanderlan, M. D. Iannacone, M. S. Vincent, Q. Chen, R. A. Bridges, A survey of intrusion detection systems leveraging host data, *arXiv* 52 (2018). arXiv:1805.06070.
- [3] A. L. Buczak, E. Guven, A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection, *IEEE Communications Surveys and Tutorials* 18 (2016) 1153–1176. doi:10.1109/COMST.2015.2494502.
- [4] C. Catania, C. Garcia Garino, Automatic network intrusion detection: Current techniques and open issues, *Computer and Electrical Engineering* 7 (2012) 1063 – 1073.
- [5] E. Vasilomanolakis, S. Karuppayah, M. Muhlhauser, M. Fischer, Taxonomy and survey of collaborative intrusion detection, *ACM Computing Surveys* 47 (2015) 1–33. doi:10.1145/2716260.
- [6] I. Sharafaldin, A. Habibi Lashkari, A. A. Ghorbani, Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization, *International Conference on Information Systems Security and Privacy* (2018) 108–116. doi:10.5220/0006639801080116.
- [7] G. Maciá-Fernández, J. Camacho, R. Magán-Carrión, P. García-Teodoro, R. Therón, Ugr'16: A new dataset for the evaluation of cyclostationarity-based network idss, *Computers & Security* 73 (2018) 411 – 424. doi:https://doi.org/10.1016/j.cose.2017.11.004.

- [8] R. Hofstede, P. Čeleda, B. Trammell, I. Drago, R. Sadre, A. Sperotto, A. Pras, Flow monitoring explained: From packet capture to data analysis with netflow and ipfix, *IEEE Communications Surveys Tutorials* 16 (2014) 2037–2064. doi:10.1109/COMST.2014.2321898.
- [9] S. Kumar, K. Dutta, Intrusion detection in mobile ad hoc networks: techniques, systems, and future challenges, *Security and Communication Networks* 9 (2016) 2484–2556. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/sec.1484>. doi:<https://doi.org/10.1002/sec.1484>.
- [10] B. Sun, L. Osborne, Y. Xiao, S. Guizani, Intrusion detection techniques in mobile ad hoc and wireless sensor networks, *IEEE Wireless Communications* 14 (2007) 56–63. doi:10.1109/MWC.2007.4396943.
- [11] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, S. C. de Alvarenga, A survey of intrusion detection in internet of things, *Journal of Network and Computer Applications* 84 (2017) 25–37. doi:<https://doi.org/10.1016/j.jnca.2017.02.009>.
- [12] T. S. Pham, T. H. Hoang, V. Van Canh, Machine learning techniques for web intrusion detection — a comparison, in: 2016 Eighth International Conference on Knowledge and Systems Engineering (KSE), 2016, pp. 291–297. doi:10.1109/KSE.2016.7758069.
- [13] V. G. T. da Costa, S. Barbon, R. S. Miani, J. J. P. C. Rodrigues, B. B. Zarpelão, Detecting mobile botnets through machine learning and system calls analysis, in: 2017 IEEE International Conference on Communications (ICC), 2017, pp. 1–6. doi:10.1109/ICC.2017.7997390.
- [14] A. Tesfahun, D. Lalitha Bhaskari, Intrusion detection using random forests classifier with SMOTE and feature reduction, *Proceedings - 2013 International Conference on Cloud and Ubiquitous Computing and Emerging Technologies, CUBE 2013* (2013) 127–132. doi:10.1109/CUBE.2013.31.
- [15] Z. Yueai, C. Junjie, Application of unbalanced data approach to network intrusion detection, in: 2009 First International Workshop on Database Technology and Applications, 2009, pp. 140–143. doi:10.1109/DBTA.2009.116.

- [16] C. Wheelus, T. M. Khoshgoftaar, R. Zuech, M. M. Najafabadi, A session based approach for aggregating network traffic data – the santa dataset, in: 2014 IEEE International Conference on Bioinformatics and Bioengineering, 2014, pp. 369–378. doi:10.1109/BIBE.2014.72.
- [17] F. Haddadi, A. N. Zincir-Heywood, Benchmarking the effect of flow exporters and protocol filters on botnet traffic classification, IEEE Systems Journal 10 (2016) 1390–1401. doi:10.1109/JSYST.2014.2364743.
- [18] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, Openflow: Enabling innovation in campus networks, SIGCOMM Comput. Commun. Rev. 38 (2008) 69–74. URL: <https://doi.org/10.1145/1355734.1355746>. doi:10.1145/1355734.1355746.
- [19] G. Cugola, A. Margara, Processing flows of information: From data stream to complex event processing, ACM Computing Surveys 44 (2012). doi:10.1145/2187671.2187677.
- [20] D. Y. Huang, N. Apthorpe, F. Li, G. Acar, N. Feamster, Iot inspector: Crowdsourcing labeled network traffic from smart home devices at scale, Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 4 (2020). doi:10.1145/3397333.
- [21] J. E. Díaz-Verdejo, A. Estepa, R. Estepa, G. Madinabeitia, F. J. Muñoz-Calle, A methodology for conducting efficient sanitization of http training datasets, Future Generation Computer Systems 109 (2020) 67–82. doi:<https://doi.org/10.1016/j.future.2020.03.033>.
- [22] R. Sommer, V. Paxson, Outside the Closed World: On Using Machine Learning for Network Intrusion Detection, IEEE Symposium on Security and Privacy 0 (2010) 305–316. doi:10.1109/SP.2010.25.
- [23] E. B.V, Scopus, <https://www.scopus.com/>, ???? [Online; accessed July-2019].
- [24] Google, Google scholar, <https://scholar.google.com/>, ???? [Online; accessed July-2019].
- [25] IEEE, Ieee explorer, advancing technology of humanity, <https://www.ieee.org/>, ???? [Online; accessed July-2019].

- [26] A. for Computing Machinery, Acm digital library, <https://dl.acm.org/>, ??? [Online; accessed July-2019].
- [27] Microsoft, Microsoft academic search, <https://academic.microsoft.com/home>, ??? [Online; accessed July-2019].
- [28] S. Nature, Springer, <https://www.springer.com/>, ??? [Online; accessed July-2019].
- [29] E. B.V, Mendeley brings your research to life, so you can make an impact on tomorrow, <https://www.mendeley.com/>, ??? [Online; accessed July-2019].
- [30] Def conference, <https://www.defcon.org/>, 1993. [Online; accessed May-2021].
- [31] USENIX, The advanced computing systems association, <https://www.usenix.org/>, 1975. [Online; accessed May-2021].
- [32] I. I. Workshop, Ipom: International workshop on ip operations and management, <https://link.springer.com/book/10.1007/978-3-642-04968-2>, 2009. [Online; accessed May-2021].
- [33] A. A. for Computing Machinery, Computer and communications security, <https://dl.acm.org/conference/ccs>, 1993. [Online; accessed May-2021].
- [34] ELSEVIER, Computers & security. the international source of innovation for the information security and it audit professional, <https://www.journals.elsevier.com/computers-and-security>, 2000. [Online; accessed May-2021].
- [35] IEEE, Ieee vis: Visualization & visual analytics, <http://ieevis.org/>, 2000. [Online; accessed May-2021].
- [36] A. A. for Computing Machinery, European conference on computer systems, <https://dl.acm.org/conference/eurosys>, 2006. [Online; accessed May-2021].
- [37] J. Bernard, M. Hutter, M. Zeppelzauer, D. Fellner, M. Sedlmair, Comparing visual-interactive labeling with active learning: An experimental

- study, *IEEE Transactions on Visualization and Computer Graphics* 24 (2018) 298–308. doi:10.1109/TVCG.2017.2744818.
- [38] M. Roesch, SNORT - lightweight intrusion detection for networks, in: *Proceedings of the 13th USENIX conference on System administration, LISA '99*, USENIX Association, Berkeley, CA, USA, 1999, pp. 229–238. ISBN 978-1-931971-59-1.
- [39] V. Paxson, BRO: a system for detecting network intruders in real-time, *Computer Networks* 31 (1999) 2435–2463.
- [40] W. Lee, S. J. Stolfo, Data mining approaches for intrusion detection, in: *Proceedings of the 7th conference on USENIX Security Symposium - Volume 7*, USENIX Association, Berkeley, CA, USA, 1998, pp. 6–6. ISBN 978-1-931971-59-1.
- [41] R. P. Lippmann, D. J. Fried, I. Graf, J. W. Haines, K. R. Kendall, D. McClung, D. Weber, S. E. Webster, D. Wyszogrod, R. K. Cunningham, M. A. Zissman, Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation, *Proceedings - DARPA Information Survivability Conference and Exposition, DISCEX 2000 2* (2000) 12–26. doi:10.1109/DISCEX.2000.821506.
- [42] A. Kenyon, L. Deka, D. Elizondo, Are public intrusion datasets fit for purpose characterising the state of the art in intrusion event datasets, *Computers & Security* 99 (2020) 102022. doi:<https://doi.org/10.1016/j.cose.2020.102022>.
- [43] X. Ugarte-Pedrero, M. Graziano, D. Balzarotti, A close look at a daily dataset of malware samples, *ACM Trans. Priv. Secur.* 22 (2019). URL: <https://doi.org/10.1145/3291061>. doi:10.1145/3291061.
- [44] S. García, M. Grill, J. Stiborek, A. Zunino, An empirical comparison of botnet detection methods, *Computers and Security* 45 (2014) 100–123. doi:10.1016/j.cose.2014.05.011.
- [45] E. Papadogiannaki, S. Ioannidis, A survey on encrypted network traffic analysis applications, techniques, and countermeasures, *ACM Comput. Surv.* 54 (2021). URL: <https://doi.org/10.1145/3457904>. doi:10.1145/3457904.

- [46] M. H. Bhuyan, D. K. Bhattacharyya, J. K. Kalita, Towards generating real-life datasets for network intrusion detection, *International Journal of Network Security* 17 (2015) 683–701.
- [47] N. Moustafa, J. Slay, UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set), 2015 Military Communications and Information Systems Conference, MilCIS 2015 - Proceedings (2015). doi:10.1109/MilCIS.2015.7348942.
- [48] W. Haider, J. Hu, J. Slay, B. P. Turnbull, Y. Xie, Generating realistic intrusion detection system dataset based on fuzzy qualitative modeling, *Journal of Network and Computer Applications* 87 (2017) 185–192. doi:10.1016/j.jnca.2017.03.018.
- [49] S. K. Mukkavilli, S. Shetty, L. Hong, Generation of Labelled Datasets to Quantify the Impact of Security Threats to Cloud Data Centers, *Journal of Information Security* (2016) 172–184.
- [50] A. Lemay, J. M. Fernandez, Providing SCADA network data sets for intrusion detection research, *Usenix Cset* (2016).
- [51] (????).
- [52] S. J. Stolfo, W. Fan, W. Lee, A. Prodromidis, P. K. Chan, Cost-based modeling for fraud and intrusion detection: Results from the JAM project, *Proceedings - DARPA Information Survivability Conference and Exposition, DISCEX 2000 2* (2000) 130–144. doi:10.1109/DISCEX.2000.821515.
- [53] C. A. Catania, F. Bromberg, C. GarciaGarino, An autonomous labeling approach to support vector machines algorithms for network traffic anomaly detection, *Expert Syst. Appl.* 39 (2012) 1822–1829. URL: <https://doi.org/10.1016/j.eswa.2011.08.068>. doi:10.1016/j.eswa.2011.08.068.
- [54] F. Gargiulo, C. Mazzariello, C. Sansone, Automatically building datasets of labeled IP traffic traces: A self-training approach, *Applied Soft Computing Journal* 12 (2012) 1640–1649. doi:10.1016/j.asoc.2012.02.012.

- [55] J. Song, H. Takakura, Y. Okabe, M. Eto, D. Inoue, K. Nakao, Statistical analysis of honeypot data and building of Kyoto 2006+ dataset for NIDS evaluation, Proceedings of the 1st Workshop on Building Analysis Datasets and Gathering Experience Returns for Security, BADGERS 2011 (2011) 29–36. doi:10.1145/1978672.1978676.
- [56] A. Sperotto, R. Sadre, F. van Vliet, A. Pras, A labeled data set for flow-based intrusion detection, in: G. Nunzi, C. Scoglio, X. Li (Eds.), IP Operations and Management, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 39–50.
- [57] F. J. Aparicio-Navarro, K. G. Kyriakopoulos, D. J. Parish, Automatic dataset labelling and feature selection for intrusion detection systems, Proceedings - IEEE Military Communications Conference MILCOM (2014) 46–51. doi:10.1109/MILCOM.2014.17.
- [58] M. Ring, S. Wunderlich, D. Grüdl, D. Landes, A. Hotho, Flow-based benchmark data sets for intrusion detection, 2017, pp. 361–369. Cited By 54.
- [59] B. Sangster, T. Cook, R. Fanelli, E. Dean, W. J. Adams, C. Morrell, G. Conti, Toward Instrumenting Network Warfare Competitions to Generate Labeled Datasets, USENIX Security’s Workshop on Cyber Security Experimentation and Test (CSET) (2009).
- [60] P. University, Planet lab, <http://www.https://www.planetlab.org/status>, 2007. [Online; accessed January-2020].
- [61] L. Peterson, A. Bavier, M. E. Fiuczynski, S. Muir, Experiences building planetlab, in: Proceedings of the 7th Symposium on Operating Systems Design and Implementation, OSDI ’06, USENIX Association, USA, 2006, p. 351–366.
- [62] H. Shiravi, A. Shiravi, A. A. Ghorbani, A survey of visualization systems for network security, IEEE Transactions on Visualization and Computer Graphics 18 (2012) 1313–1329. doi:10.1109/TVCG.2011.144.
- [63] P. Huitsing, R. Chandia, M. Papa, S. Sheno, Attack taxonomies for the modbus protocols, International Journal of Critical Infrastructure Protection 1 (2008) 37–44.

- [64] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, K. Das, The 1999 darpa off-line intrusion detection evaluation, *Computer Networks* 34 (2000) 579–595. doi:[https://doi.org/10.1016/S1389-1286\(00\)00139-0](https://doi.org/10.1016/S1389-1286(00)00139-0), recent *Advances in Intrusion Detection Systems*.
- [65] J. Stokes, J. Platt, J. Kravis, M. Shilman, ALADIN: Active Learning of Anomalies to Detect Intrusions, Technical Report MSR-TR-2008-24, 2008.
- [66] A. Beaunon, P. Chifflier, F. Bach, ILAB: An Interactive Labelling Strategy for Intrusion Detection, *International Symposium on Research in Attacks, Intrusions, and Defenses* 7462 (2012) 120–140. doi:[10.1007/978-3-642-33338-5](https://doi.org/10.1007/978-3-642-33338-5). arXiv:[9780201398298](https://arxiv.org/abs/9780201398298).
- [67] X. Fan, C. Li, X. Yuan, X. Dong, J. Liang, An interactive visual analytics approach for network anomaly detection through smart labeling, *Journal of Visualization* 22 (2019) 955–971. doi:[10.1007/s12650-019-00580-7](https://doi.org/10.1007/s12650-019-00580-7).
- [68] N. Görnitz, M. Kloft, K. Rieck, U. Brefeld, Toward Supervised Anomaly Detection, *Journal of Artificial Intelligence Research* 46 (2013) 235–262. doi:[10.1613/jair.3623](https://doi.org/10.1613/jair.3623).
- [69] H. Koike, K. Ohno, K. Koizumi, Visualizing Cyber Attacks using IP Matrix, *IEEE Workshop on Visualization for Computer Security* (2006) 11–11. doi:[10.1109/vizsec.2005.22](https://doi.org/10.1109/vizsec.2005.22).
- [70] Y. Livnat, J. Agutter, S. Moon, R. F. Erbacher, S. Foresti, A visualization paradigm for network intrusion detection, *Proceedings from the 6th Annual IEEE System, Man and Cybernetics Information Assurance Workshop, SMC 2005* 2005 (2005) 92–99. doi:[10.1109/IAW.2005.1495939](https://doi.org/10.1109/IAW.2005.1495939).
- [71] P. Ren, Y. Gao, Z. Li, Y. Chen, B. Watson, IDGraphs: Intrusion detection and analysis using histograms, *IEEE Workshop on Visualization for Computer Security 2005, VizSEC 05, Proceedings* (2005) 39–46. doi:[10.1109/VIZSEC.2005.1532064](https://doi.org/10.1109/VIZSEC.2005.1532064).
- [72] C. Scott, K. Nyarko, T. Capers, J. Ladeji-Osias, Network intrusion visualization with niva, an intrusion detection visual and haptic analyzer,

- Information Visualization 2 (2003) 82–94. doi:10.1057/palgrave.ivs.9500044.
- [73] S. Chen, C. Guo, X. Yuan, F. Merkle, H. Schaefer, T. Ertl, Oceans - online collaborative explorative analysis on network security, volume 10-November-2014, Association for Computing Machinery, 2014, pp. 1–8. doi:10.1145/2671491.2671493.
- [74] B. C. M. Cappers, P. N. Meessen, S. Etalle, J. J. van Wijk, Eventpad: Rapid malware analysis and reverse engineering using visual analytics, in: 2018 IEEE Symposium on Visualization for Cyber Security (VizSec), 2018, pp. 1–8.
- [75] H. Zhang, Y. Guo, T. Li, P. Angin, Multifeature named entity recognition in information security based on adversarial learning, Security and Communication Networks 2019 (2019). doi:10.1155/2019/6417407.
- [76] M. R. Shahid, G. Blanc, Z. Zhang, H. Debar, Iot devices recognition through network traffic analysis, in: 2018 IEEE International Conference on Big Data (Big Data), 2018, pp. 5187–5192. doi:10.1109/BigData.2018.8622243.
- [77] Y. Yang, Z. Ma, F. Nie, X. Chang, A. G. Hauptmann, Multi-Class Active Learning by Uncertainty Sampling with Diversity Maximization, International Journal of Computer Vision 113 (2015) 113–127. doi:10.1007/s11263-014-0781-x.
- [78] S. McElwee, Active learning intrusion detection using k-means clustering selection, in: SoutheastCon 2017, 2017, pp. 1–7. doi:10.1109/SECON.2017.7925383.
- [79] D. D. Lewis, J. Catlett, Heterogeneous uncertainty sampling for supervised learning, in: W. W. Cohen, H. Hirsh (Eds.), Machine Learning Proceedings 1994, Morgan Kaufmann, San Francisco (CA), 1994, pp. 148–156. doi:https://doi.org/10.1016/B978-1-55860-335-6.50026-X.
- [80] M. Almgren, E. Jonsson, Using active learning in intrusion detection, in: Proceedings of the Computer Security Foundations Workshop, volume 17, 2004, pp. 88–98. doi:10.1109/csfw.2004.1310734.

- [81] A. Beaunon, P. Chifflier, F. Bach, Ilab: An interactive labelling strategy for intrusion detection, in: M. Dacier, M. Bailey, M. Polychronakis, M. Antonakakis (Eds.), *Research in Attacks, Intrusions, and Defenses*, Springer International Publishing, Cham, 2017, pp. 120–140.
- [82] J. L. Guerra, E. Veas, C. A. Catania, A study on labeling network hostile behavior with intelligent interactive tools, in: *2019 IEEE Symposium on Visualization for Cyber Security (VizSec)*, 2019, pp. 1–10. doi:10.1109/VizSec48167.2019.9161489.
- [83] D. Pelleg, A. Moore, Active learning for anomaly and rare-category detection, *Advances in Neural Information Processing Systems* 18 (2004) 1073–1080.
- [84] J. L. Torres, C. A. Catania, E. Veas, Active learning approach to label network traffic datasets, *Journal of Information Security and Applications* 49 (2019) 102388. doi:10.1016/j.jisa.2019.102388.
- [85] R. Hofstede, A. Pras, A. Sperotto, G. D. Rodosek, Flow-based compromise detection: Lessons learned, *IEEE Security Privacy* 16 (2018) 82–89. doi:10.1109/MSP.2018.1331021.
- [86] M. Cermak, T. Jirsik, P. Velan, J. Komarkova, S. Spacek, M. Drasar, T. Plesnik, Towards provable network traffic measurement and analysis via semi-labeled trace datasets, in: *2018 Network Traffic Measurement and Analysis Conference (TMA)*, 2018, pp. 1–8. doi:10.23919/TMA.2018.8506498.
- [87] J. Guerra, C. A. Catania, E. Veas, Visual Exploration of Network Hostile Behavior, *Proceedings of the 2017 ACM Workshop on Exploratory Search and Interactive Data Analytics - ESIDA '17* (2017) 51–54. doi:10.1145/3038462.3038466.
- [88] J. O. Nehinbe, A critical evaluation of datasets for investigating IDSs and IPSs researches, *Proceedings of 2011, 10th IEEE International Conference on Cybernetic Intelligent Systems, CIS 2011* (2011) 92–97. doi:10.1109/CIS.2011.6169141.
- [89] L. Wang, R. Jones, Big data analytics in cyber security: Network traffic and attacks, *Journal of Computer Information*

Systems 61 (2021) 410–417. URL: <https://doi.org/10.1080/08874417.2019.1688731>. doi:10.1080/08874417.2019.1688731. arXiv:<https://doi.org/10.1080/08874417.2019.1688731>.

- [90] A. Banerjee, C.-C. Chen, C.-C. Hung, X. Huang, Y. Wang, R. Chevesaran, Challenges and experiences with mlops for performance diagnostics in hybrid-cloud enterprise software deployments, in: 2020 {USENIX} Conference on Operational Machine Learning (OpML 20), 2020.
- [91] A. Gharib, I. Sharafaldin†, A. H. Lashkari, A. A. Ghorbani, An Evaluation Framework for Intrusion Detection Dataset, International Conference on Information Science and Security (ICISS) 22 (2016) 1–6. doi:10.1016/0371-1951(66)80211-4.