# UGR'16: A New Dataset for the Evaluation of Cyclostationarity-Based Network IDSs

Gabriel Maciá Fernández*[1], José Camacho[1], Roberto Magán-Carrión[1], Pedro García-Teodoro[1], Roberto Theron[2]
[1]Dpt. of Signal Theory, Telematics and Communications - CITIC
University of Granada - Spain
Email:{gmacia,josecamacho,rmagan,pgteodor}@ugr.es
[2]University of Salamanca - Spain
Email: theron@usal.es

***Abstract*—The evaluation of algorithms and techniques to implement intrusion detection systems heavily rely on the existence of well designed datasets. In the last years, a lot of efforts have been done towards building these datasets. Yet, there is still room to improve. In this paper, a comprehensive review of existing datasets is first done, making emphasis on their main shortcomings. Then, we present a new dataset that is built with real traffic and up-to-date attacks. The main advantage of this dataset over previous ones is its usefulness for evaluating IDSs that consider long-term evolution and traffic periodicity. Models that consider differences in daytime/night or weekdays/weekends can also be trained and evaluated with it. We discuss all the requirements for a modern IDS evaluation dataset and analyze how the one presented here meets the different needs.**

***Index Terms*—Network Security, Dataset, IDS, Network Traffic, Netflow**

## I. Introduction

Intrusion Detection Systems (IDSs) appeared in the security arena as a solution for the problem of identifying malicious activities in networks and systems. In few words, an IDS consists of a monitoring module in charge of obtaining data, a preprocessing module to adapt these data to further processing steps in the system, and a decision module able to determine if an event might be considered malicious or not.

There are several types of IDSs [1]: *Network-based IDSs* (NIDS) monitor network events like flows or firewall logs, among others, while *host-based IDSs* (HIDS) consider system-related events, *e.g.* syslog, file system watchdog, CPU load, etc. IDSs are also classified according to their detection approach. This way, *signature-based IDSs* (S-IDS) rely on rules to decide if an observed behavior is malicious or not, while *anomaly-based IDSs* (A-IDS) [2] build a model from training data and consider that any behavior that deviates from this model is anomalous. Note that, although there is a semantic difference between an anomalous behavior and a malicious one, A-IDSs normally consider them equivalently.

A main problem for testing the capabilities of IDSs is the need of a representative standard dataset that allows the comparison among different proposals. In the 90s, DARPA funded a project to build such a dataset and MIT Lincoln Lab produced the well known DARPA'98 and DARPA'99 datasets [3]. After being thoroughly used and studied by several authors, some limitations were pointed out, such as the existence of duplicated records, non-balanced samples of attack and normal connections, and the inherent limitations to considering synthetic traffic. Since then, many other researchers and projects tried to provide improved versions of these datasets, *e.g.* NSL-KDD, or to build new datasets.

Despite this big number of efforts to contribute a definite dataset for IDS evaluation and, after having learned many lessons, we can see that there is still only partial solutions. In a first review, we can check that many of the recent datasets lack from either real traffic or updated attack strategies. The most important limitation from the perspective of this paper is related to the duration of the data recording. In order to enable the evaluation of detection algorithms that consider the cyclostationary evolution of traffic, *i.e.*, differences in traffic between daytime/night or weekdays/weekends, a long duration trace is needed. As we will review, there are no publicly existing datasets to evaluate these algorithms.

In this paper, our contribution is threefold. First, we make a review of past efforts in the publication of datasets for the evaluation of IDSs. From this review we extract a set of requirements that datasets for IDS evaluation should comply with. Second, we give arguments for using cyclostationarity features in modern network traffic IDSs. Third, we describe a new dataset (the UGR'16 dataset[1]) designed for the evaluation of cyclostationarity-based network IDSs, that contains real anonymized netflow traces captured in a tier-3 ISP for 4 months. In this dataset, we have included realistic attack scenarios and labeled the traffic.

The structure of the paper is as follows. In Section II we review previous efforts to capture datasets for training and testing IDSs. From this review, we derive a set of requirements for this kind of datasets and compare the different proposals. The need of using cyclostationarity in the network anomaly detection process is presented in Section III, together with a review of related works that leverage this feature. In Section IV we describe how we have built the dataset and the methodology followed to insert attack traffic. In Section V we first make an analysis of the dataset, discussing and providing a global description of the information contained on it. Then, we also discuss the labeling process. Finally, we conclude in

[1]We take the name for the dataset from the acronym of University of Granada.

Section VI.

## II. Review of Existing Datasets

In what follows, we first review the most relevant efforts done in this field during the last years. Then, from these previous works, we derive the ideal requirements for an IDS evaluation dataset. Finally, we compare all the suggested approaches and discuss the main features of our proposal.

### A. Existing Datasets

Since the apparition of IDS technologies, many different efforts to publish good datasets for their evaluation have been made. The first serious efforts to build datasets lead to those that we call *reference datasets*, *i.e.*, datasets that have been used in many research papers and that have allowed the research community to learn many lessons. After them, a large list of more *recent datasets* have appeared.

It is important to clarify at this point that we will review only those datasets that include labels indicating if attacks are present or not. There are other non-labeled traffic datasets that are extensively used for other purposes, *e.g.* CAIDA datasets [4], but their application to the IDS evaluation field is really limited.

#### *1) Reference Datasets:*

**DARPA datasets.** As previously indicated, the first effort to build a dataset for IDS evaluation was carried out by the MIT Lincoln Lab. These datasets are known as DARPA datasets [3]: *DARPA'98* and *DARPA'99*.

DARPA'98 consists of 4GB of compressed raw (binary) tcpdump data, that spans 7 weeks of synthetic network traffic with 5M of connection records. Test data comprises two weeks of traffic and around 2M connection records. DARPA'99 comprises 5 weeks of tcpdump network traffic files sniffed from two different points in a synthetic network (one inside the network and the other outside). Host based audit data is also collected from 4 victim systems.

***KDDCup'99 dataset***. This dataset [5] was prepared by Stolfo *et al.* [6] for the KDD'99 contest. It is built from the DARPA'98 dataset and it consists of 4,9M single connection vectors, each with 41 features, labeled either as normal or attack of one of the following categories: DoS, User to Root (U2R), Remote to Local (R2L) and Probing Attack. There are 24 training attack types, with an additional set of 14 different attacks that appear in the test data only. The features extracted by Stolfo *et al.* were derived as several classes: a) *Basic features* of TCP connections (*e.g.* duration, #bytes, TCP flags, port number, etc.); b) *traffic features*, derived from a two seconds sliding time window and considering connections directed to the same host or the same service (*e.g.* #connections to the same host as the current connection in the past two seconds); c) *content features*, extracted from the information in the payload, *e.g.* number of failed login attempts.

**Improvements to DARPA datasets.** McHugh *et al.* [7] made an extensive analysis on DARPA datasets, concluding that traffic data had a lack of statistical evidence of similarity to typical Air Force network traffic, low traffic rates, relative uniform distribution of the four major attack categories, skewed distribution of victim hosts, and flat network topology.

Mahoney *et al.* [8] stated that DARPA'99 dataset presents statistical problems, *i.e.*, they detected the presence of artifacts that could allow a network anomaly detection system to detect some novel intrusions based on idiosyncrasies of the underlying implementation of the simulation, with an artificially low false alarm rate.

Motivated by these studies, Tavallaee *et al.* improved KDD-Cup'99, creating the *NSL-KDD dataset* [9]. From the original dataset, they eliminated a hugh number of redundant records present in the original data, and they applied a methodology to rise the *difficulty level* of records in the dataset. Other authors also proposed additional extensions, as Perona *et al.* in [10], where *gureKDDCup dataset* was built as an extension to KDDCup'99 in which payload-related information of connections was processed from DARPA'98 and included in the dataset.

#### *2) More recent datasets:*

Although many of the current research in the IDS field still use these cited reference datasets, due to the critics and the inherent problems already pointed out, many other researchers have built new updated datasets. The main motivations in these datasets are either to provide different types of data, *e.g.* data related to more updated threats like botnets, or to collect traffic that is more realistic in current networks, under the hyphotesis that traffic features in the current Internet have evolved since the collection of DARPA datasets.

Some of the efforts to build datasets during the recent last years are chronologically summarized in what follows.

***Sperotto*** (2008). A flow based dataset (2008) was generated by Sperotto *et al.* [11]. The dataset is composed of labeled non-synthetic traffic. Its main limitation is that the amount of traffic collected is reduced and the topology is very simplistic, mainly due to the fact that, for performance reasons, they monitor a single host (honeypot) connected to a university network. The authors claim that the trace size is enough, but it is only 6 days long, what does not allow to test algorithms based on daily and weekly traffic evolution patterns. Labeling in this dataset is done based on logs from the honeypot system, what makes it dependable on the detection system configuration and the log processing. The dataset is only available upon request by mail to the authors.

***MAWI Working Lab Dataset***. This dataset [12] was contributed since 2010 as a labeled version of the MAWI (Measurement and Analysis on the WIDE Internet) archive sample for network points B and F. The archive contains tcpdump daily traces of 15 minutes captured in a link between Japan and the United States. Although the archive contains traces since 2001, only the traffic since 2007 is labeled. The labels are generated by using a combination of several anomaly detection classifiers. This dataset is really extensive and allows the analysis for a very long period. The main limitation is that the trace is obtained only during a 15 minutes period per day. In addition, the labeling is dependent on the classifiers that have been used in the process and their generation of false positives.

***UNB ISCX 2012***: This dataset was created by Shiravi *et al.* [13]. The most relevant contribution of this work is the use of profiles for the generation of traffic. The authors define certain $\alpha$ profiles for attack traffic and $\beta$ profiles for background traffic. They implement their proposal in a network with 17 Windows XP stations and a single Windows 7 computer, collecting data during 7 days. The main drawbacks of this dataset nowadays are its reduced duration, the use of somehow old operating systems (Windows XP), and the use of synthetic traffic.

***CTU-13***. This dataset is described in [14]. It is composed of 13 different malware captures in a real network environment, including labeled botnet, normal and background traffic, where background means that the authors are not really able to decide if it is malicious or not. Regarding the kind of information provided, flow traces are available in several formats. For privacy reasons, pcap files are only offered when no information about the network is present, *e.g.* a single host with synthetic traffic. There are many different short-duration captures that span from 2013 to 2016, although the only dataset that is well documented [14] corresponds to 2013. The main limitation is with the short duration of the traces, that impedes building models that consider cyclostationary evolution. Other limitation is with background traffic. The authors say that it is obtained from a university router, but no more details about topology or services are provided. Regarding attack traffic, this dataset is focused on the behavior of 13 botnet scenarios, where malware are located in Windows XP machines. Despite these issues, this dataset is a really good effort to capture real traffic, although its application is narrowed to that of botnet behavior detection.

***ADFA-LD12 (2013)***. This dataset was presented in [15]. Its focus is on host-based intrusion detection system (HIDS) evaluation. Several attacks are striked against a Linux OS machine (Ubuntu 11.04), *e.g.* password bruteforce against FTP and SSH, client side attacks, java based payloads, php RFI attacks, etc. In addition, modern hacking methodologies are employed, including the use of attack frameworks like Metasploit [16]. Traces are collected by using the `auditd` Unix tool. The main problem with this dataset is related with the trace associated to the normal behavior. Here, only one computer (with Linux OS) is considered and there is no description at all about the services and software installed. This way, it is not possible to take it as a base to generalize a common behavior of other machines in the Internet.

***UNSW-NB15***. This dataset was recently proposed by Moustafa *et al.* [17]. The authors used an attack automatic generation tool called IXIA PerfectStorm to implement nine families of real and updated attacks against several servers. They collected tcpdump traces of the network traffic, for a total duration of 31 hours at the beginning of 2015 (2M of flows). From these traces, a dataset of 49 features for every flow was built. The main problem with this dataset is the synthetic generation of traffic, which is associated to theoretical instead of realistic behaviors in the Internet..

***Application specific datasets***. Finally, it is worth mentioning the existence of different datasets that are specific for certain areas. As an example, new datasets are being built for industrial control systems (SCADA) [18] [19].

## B. *Requirements for IDS evaluation datasets*

Some authors have analyzed previous datasets and identified a set of desirable features [9] [11] [13] [20] for an ideal dataset to evaluate IDSs. From these works and our own review of existing datasets, we extract some requirements:

1) *Features*: Datasets can include network and/or host features. Network features are normally summarized in a per-flow basis, and include characteristics like flow duration, timestamps, number of bytes and packets, flags, IP addresses and port numbers, inter-packet mean time, etc. Regarding host features, some examples used in previous datasets are: number of failed login attempts or flags to indicate certain conditions, like if a root shell is obtained or if the attacker successfully logged in. To train anomaly NIDS, network features are essential, while host characteristics are normally used to train anomaly HIDS. Some datasets include both set of features, which allows the training of both types of IDS. The selected dataset should contain appropriate features for the type of IDS to be trained and tested. Anyway, it is desirable that data are shared in raw format as they are obtained from the network. This way, researchers can use their own features at will.
2) *Real background traffic*: A dataset should contain real traffic. This is especially critical with the background traffic, due to the fact that a synthetic generation of network traffic might lead to normality models and behaviors that are not correct.
3) *Updated attack traffic*: Datasets for evaluating IDS should contain realistic attack instances updated with state-of-the-art attack techniques and tools.
4) *Labeling*: The different records in the dataset should ideally be correctly labeled as malicious or not. In case of attack records, the type of attack is also needed. As we will discuss next, the labeling process presents many difficulties [12].
5) *Duration*: Datasets should span for several cycles of study (days or weeks), to make it possible to consider the cyclostationary evolution of traffic in the daytime/night period, and also during weekdays/weekends and even for different months.
6) *Documentation*: A detailed description of the dataset is needed to understand its limitations and potentialities. Some of the published datasets fail to provide comprehensive enough information.
7) *Format*: Datasets usually include information in either *pcap* (tcpdump), *csv* or *flow* (netflow) format. *pcap* format permits a more comprehensive evaluation of IDSs, mainly due to the fact that payloads are included in the information. In the *csv* format, on the other hand, there is a preprocess of *pcap* information and an extraction of relevant records for different features. Note that including payload information usually implies a problem from a privacy point of view, as it is complicated to reliably anonymize the trace. For this reason, *flow* information,

TABLE I
COMPARISON OF DATASETS FOR IDS EVALUATION

| Dataset | Ref. | Features N:Net H:Host | Real back traffic | Updated attack traffic | Labeling | Duration | Docum. | Format P:pcap F:flow |
|---|---|---|---|---|---|---|---|---|
| DARPA'98/'99 | [3] | N | ✗ | ✗ | ✓ | ✓ | ✓ | P |
| KDDCup99 | [5] | N | ✗ | ✗ | ✓ | ✓ | ✓ | csv |
| NSL-KDD'09 | [9] | N | ✗ | ✗ | ✓ | ✓ | ✓ | csv |
| gureKDDCup'08 | [10] | N | ✗ | ✗ | ✓ | ✓ | ✓ | csv |
| Sperotto'2008 | [11] | N | (✓) | (✓) | ✓ | ✗ | ✓ | F |
| MAWILab'2010 | [12] | N | ✓ | ✓ | (✓) | (✓) | ✓ | P |
| UNB ISCX'2012 | [13] | N | ✗ | ✓ | ✓ | ✗ | ✓ | csv |
| CTU-13 | [14] | N | (✓) | ✓ | (✓) | (✓) | (✓) | P&F |
| ADFA-2013 | [15] | H | (✓) | ✓ | ✓ | ✗ | ✓ | csv |
| UNSW-NB15 | [17] | N | ✗ | ✓ | ✓ | ✗ | ✓ | csv |
| UGR'16 | | N | ✓ | ✓ | (✓) | ✓ | ✓ | F |

*e.g.* netflow format, is sometimes preferred when dealing with real traffic, while *pcap/csv* formats are provided in simulated/synthetic traffic datasets. Finally, note that the volume of data is directly associated with the format. Hence, *flow* format seems more appropriate for datasets with a long duration, as *pcap* format would generate a huge data volume in this case.

### C. Comparison of Existing Datasets

Table I contains a comparison of the different datasets reviewed in Section II-A. Here, we analyze the different requirements previously described and assess if the different datasets comply with them. A sign ✓ in the table means that the requirement is succesfully met, while ✗ stands for the opposite. We use (✓) to indicate that a requirement is not completely met or that there are certain limitations. For example, the real background traffic requirement in CTU-13 dataset is not completely met, as there are no details about the services and topologies behind the router under monitoring.

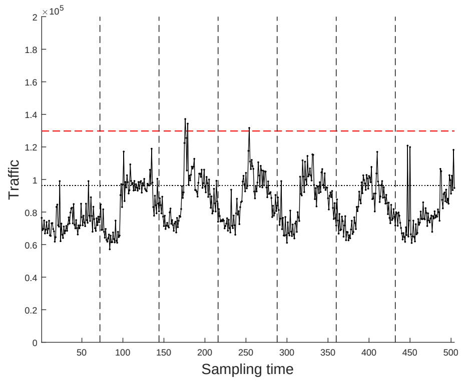
From this comparison we can derive some conclusions. First, there is a need to constantly update both background and attack traffic and, thus, new datasets are periodically demanded. Note that this problem was pointed out in [13]. Yet, the solution provided by their authors is based on building traffic models, and there is no guarantee that real traffic in a near future will follow these models. This means that reference datasets, which have been really useful for many years, are not valid anymore. Second, we can see that there has been a big effort in deploying realistic attack data in recent datasets, mainly by using powerful hacking frameworks and standard tools. In contrast, insufficient efforts have been made in the task of providing realistic background traffic. The main limitation here is the problem of privacy. For this reason, some datasets that monitor real traffic, *e.g.* CTU-13, only provide flow information and hide payloads. Finally, a main problem of recent datasets from the perspective of the claims that we make in this paper is related to the duration of the traces. If one pursues the evaluation of an IDS that builds normality models considering the traffic evolution in non-stationary periods (daytime/night, weekdays/weekends or even for different months), the existing datasets only provide a very limited information.

For the dataset we describe in this paper (UGR'16), we especially care to collect both realistic background and attack traffics. As it will explained, a modified approach of that used in [14] is followed for labeling, just to keep the balance between providing a complete labeling with synthetic traffic and dealing with realistic traffic. The duration of the dataset is enough to train models that consider non-stationary periods. In order to keep privacy requirements, we provide data in anonymized flow format.

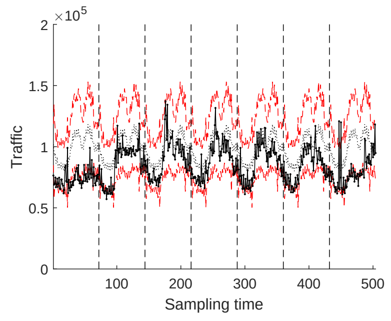
## III. CYCLOSTATIONARITY IN NETWORK IDSS

Network traffic presents short-term as well as long-term temporal behaviors. This fact has been traditionally leveraged for network design and performance evaluation aims [21], [22]. We can see in Fig. 1 an example to clarify the usefulness of considering cyclostationarity in the context of network anomaly detection. Let us consider two different simple approaches based on the observation of the network volume traffic as a random variable. In both, we define thresholds (red dashed lines) so that if traffic volume is above or below them an anomaly is signaled. The evolution of the variable during a complete week for the stationary and cyclostationary approaches is shown in Figs. 1a and 1b, respectively. On the one hand, in the stationary approach, fixed threshold values are computed from the statistical variation of the variable. On the other hand, the cyclostationary approach considers threshold values that vary over time. In the case of our example, we have considered the mean value of the observed variable computed for the same hour at all the days in the data set (pointed black series in Fig. 1b), showing two different thresholds (upper and lower thresholds) computed as a deviation from the mean values at different hours.

Fig. 2 shows the ROC curves obtained for a detector with either the stationary thresholds or the cyclostationary ones. We can see that the performance is better when cyclostationarity is considered (an improvement in Area Under Curve, AUC, values is obtained). Although this is just an example and perhaps in certain scenarios the results are not so good, we



(a) Stationary approach



(b) Cyclostationary approach

Fig. 1. Network traffic volume evolution for a single week. Thresholds (red dashed lines) and mean value (dotted black lines) are represented for (a) stationary approach and (b) cyclostationary approach.

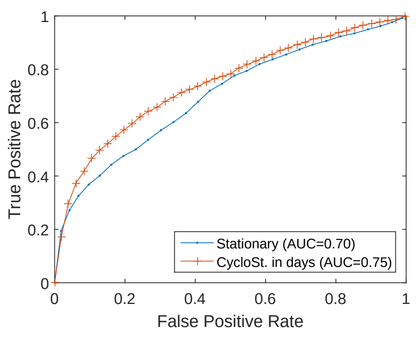


Fig. 2. ROC curves for the stationary and cyclostationary threshold-based anomaly detectors.

can still say that considering cyclostationarity would be worth in many cases.

Despite all of this, we can only mention few works that have used this feature for anomaly detection purposes in the security context [23], [24], [25]. Yet, to the best of our knowledge, no publicly available and well described traffic data sets exist at present where temporal cyclostationarity is captured. To cite two examples, authors in [22] use a trace of mobile users traffic for a single week and apply markov models to capture the volume dynamics. In a more recent work [24], the data set used is a 6 weeks duration traffic trace obtained from a campus university. As in the previous case, it is not publicly available.

## IV. Dataset Generation Methodology

Here, we describe the methodology followed to produce the dataset. In addition, we provide an extensive description of all the relevant details to use it for IDS evaluation purposes.

The main lines of the design rationale to build the dataset are these. The trace is split in two captures: calibration and test. The calibration capture is designed to allow the training of normality models. In this capture we consider only background network traffic, which should be as real as possible, as we will see in what follows. In addition, an essential requirement is that this capture has a long duration, so that it allows the training of algorithms that consider cyclostationarity.

The test capture, on the other hand, combines both background and controlled attack traffic. Attacks are generated by using state of the art hacking tools and they are representative of actual attacks. The main purpose of this capture is to evaluate if an IDS is able or not to detect the attacks. In order to allow the testing of anomaly detection algorithms that consider cyclostationarity, we define a standard batch of attacks, and it is executed at different hours and different days, so that researchers can test if the accuracy of their detectors depends on the time or the day.

### A. Network infrastructure

The data is obtained from a real network of a Tier 3 ISP. The ISP is a cloud service provider, so that many of the services implemented in the network are virtualized. We find typical hosting services, like web with proprietary or standard configurations, *e.g.* Joomla or Wordpress, email, FTP and DNS servers, etc. This network is used by many client companies that have disparate sizes and are focused on a wide variety of markets. This implies that the traffic traversing the network is expected to be very heterogeneous, as it includes both clients' accesses to the Internet, and the reception of traffic by typical servers. Thus, one potential advantage of this trace over other datasets is its representativeness of a very wide subset of Internet users. As shown before, many other traces only collect traffic from university or research center setups, where only specific traffic patterns are present.

The schematic infrastructure of the ISP network and the setup for the collection of data are shown in Fig. 3. The main aspects are:

- Two redundant border routers, *BR1* and *BR2*, provide access to the Internet. On each of their outgoing network interfaces, a *netflow* probe is configured, allowing the collection of all incoming and outgoing connections. Note that, for privacy and volume reasons, we will not provide payload information, so *netflow* information is recorded instead of *pcap* (tcpdump) files.

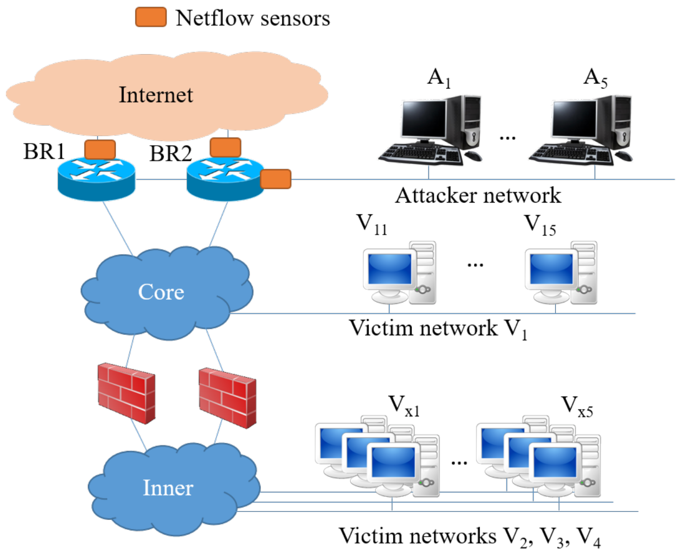


Fig. 3. Topology of the network.

- The ISP has two different subnetworks. One is termed *core network*, where the services that are not protected by firewall are located. The second is the *inner network*, where firewall services are being provided to the clients.
- An *attackers' network* with 5 attacker machines is deployed at the top level. They are referred to as $A_1$-$A_5$.
- In the core network, we configure 5 victim machines that are used only for the dataset collection. They are colocated with other real clients in an existent network that we call victim network $V_1$. These machines are named as $V_{11}$-$V_{15}$.
- Regarding the inner network, a total of 15 additional victim machines are placed in three different existing networks, each with 5 machines. We name these networks as: victim network $V_2$ (machines $V_{21}$-$V_{25}$), victim network $V_3$ (machines $V_{31}$-$V_{35}$), and victim network $V_4$ (machines $V_{41}$-$V_{45}$).

## B. Attack traffic generation

Some previous datasets, like MAWILab [12] or CAIDA [4], consider only real traffic captured in certain network probes. While this is an obvious advantage for modeling background traffic, it implies certain limitations for the identification of attacks. In fact, labeling real traffic implies the need to confirm that connections signaled as attacks are really malicious traffic. For this reason, we decide to combine real background traffic (probably containing attack instances) with attacks that are intentionally generated for the experiment.

For this purpose, the mentioned 25 virtual machines are set up with a similar configuration to those provisioned for the ISP clients, *i.e.*, they implement web, DNS, FTP and email servers. The virtual machines $A_1$ to $A_5$ are used to launch a number of specific attacks over time against the rest ($V_{x1}$-$V_{x5}$ with $x = \{1-4\}$), which play the role of victims of the attacks. As it can be seen in Fig. 3, both attackers and victims are inside the ISP infrastructure to avoid the potential detection and blocking of the attacks by other intermediate ISPs. Yet, the attackers' network is located in the border router just to generate attack traffic as if it was coming from the Internet.

**Implementation of attacks.** Due to the fact that only netflow traffic is being collected and, thus, payloads are not considered in the trace, we do not include attack types that are detectable through payload analysis. For this reason, we only consider network-related attacks. The types of attacks implemented are:

- *Low-rate DoS*: TCP SYN packets are sent to the victims using the `hping3` tool. The destination port is 80, so that the traffic is mixed with real background web traffic. The size of every packet is 1280 bits and the rate is 100 packets/s. As it can be seen, the rate of the attack is low, so that we do not affect the normal operation of the network. Three different attack scenarios are considered and repeated over time:
  - *DoS11*: One-to-one DoS attack, where the attacker $A_1$ attacks the victim $V_{21}$. The total duration of *DoS11* is 3 minutes.
  - *DoS53s*: The five attackers $A_1$-$A_5$ attack three of the victims, each one at a different network, for 3 minutes. In particular, these attacks follow this structure: $(A_1, A_2) \rightarrow V_{21}$, $(A_3, A_4) \rightarrow V_{31}$ and $A_5 \rightarrow V_{41}$. The letter '*s*' at the end of the name of the attack stands for 'synchronous', which means that the attacks are initiated by the attackers at the same time. Because of that synchronization, the duration of DoS53s is 3 minutes too.
  - *DoS53a*: The attacks are executed as in DoS53s, but now every victim is sequentially selected, being attacked for 3 minutes, with an inactivity period of 30 seconds between the three attacks. This way, the total duration of *DoS53a* is 10 minutes. In this case the letter '*a*' at the end of the name of the attack stands for 'asynchronous'.
- *Port scanning*: A continuous SYN scanning to common ports of victims is executed for 3 minutes, using the `nmap` tool. Two variants are implemented for this attack:
  - *Scan11*: One-to-one scan attack, where the attacker $A_1$ scans the victim $V_{41}$.
  - *Scan44*: Four-to-four scan attack, where the attackers $A_1$, $A_2$, $A_3$ and $A_4$ initiate a scan at the same time to the victims $V_{21}$, $V_{11}$, $V_{31}$ and $V_{41}$, respectively. As the attacks are carried out in parallel (beginning at the same instant), the total duration is 3 minutes.
- *Botnet traffic*: Due to the prominence of botnets nowadays, we have designed a methodology to insert botnet traffic in the trace. A main problem appears from the ethical point of view, as it is not feasible to infect machines in such an open network while ensuring a complete control over the potential damage that botnet malware could cause. For this reason, we suggest mixing botnet captures recorded elsewhere in a controlled environment with our background traffic. We are aware that this mixture has a main limitation: the resulting trace might not be completely realistic in certain cases, as it

does not consider the effect of botnet traffic over others. As an example, think of a period of time where a DDoS is executed against a victim. During this time, and due to the existence of overwhelmed network devices, the normal background traffic would also be affected and would not remain the same. In this scenario, mixing a capture of DDoS with normal background traffic would be unrealistic. Still, we think that the proposed solution (mixture of traffic captures) will be a good solution in many situations where the influence of botnet traffic on the background traffic is negligible.

As a proof of concept, we have included in our trace botnet traffic obtained from the execution of the malware known as *Neris*. This capture is available as the *CTU-Malware-Capture-Botnet-42* [26]. In this malware version, infected bots send SPAM, connect to an HTTP C&C server and use HTTP to perform some ClickFraud (see details in [26]).

In order to insert the botnet trace into our dataset, we follow this methodology. First, we convert the `pcap` file containing the trace to `nfcapd` format using the `softflowd` tool to generate the netflow traffic and `nfcapd` tool to collect information into `nfcapd` files. Then, we modify the `nfcapd` file to adapt the timestamps and IP addresses to be coherent with our dataset. Specifically, we inject the traffic in the dataset as if all the twenty victim machines in our setup are infected, so that algorithms can check if the correlation of traffic coming from all these bots is detected. The complete *Neris* botnet trace is then cut off so that the duration is one hour during the infection phase. This will allow us to generate attack batches to test the behavior of IDSs with background traffic at different times of the day/week, as explained in what follows.

**Attacks execution scheduling.** The attack traffic is generated in batches of 2 hours. In every attack batch, all the attack variants are executed following two possible scheduling patterns:

1) *Planned scheduling*: Each attack in the batch is executed at a fixed and known instant given by an offset from an initial batch time $t_0$. The offsets for the different attacks are shown in Table II. Note that there is no overlapping in time among the different types of attacks.
2) *Random scheduling*: The initial time for the execution of each of the attacks is randomly selected between $t_0 + 00h00m$ and $t_0 + 01h50m$, thus restricting the total duration of the batch to a maximum of 2 hours. In this case, there could exist time overlap among attacks and this will let us check the accuracy of anomaly detectors when this situation appears. For the random batches, botnet traffic is not combined in the dataset.

It is worth noting that the attacks are launched while real background traffic is traversing the network. This way, the traffic captured by the sensors for the corresponding monitoring window will include both normal and attack related traffic instances. Thus, in order to assess the potential detection of attacks by IDSs when they are mixed with background traffic at different hours of day, we launch attack batches during 12 consecutive days from the beginning of the attack experiment, thus covering all possible day hours. For a day in the attack experiment, a planned scheduling batch is launched at time $t_0$, followed by a random scheduling batch that starts at $t_0 + 12h$. In every next day during 12 days, $t_0$ is incremented by an offset of $2h$. In Table III we can see the different times and dates selected for the execution of planned and random attack batches.

TABLE II
PLANNED SCHEDULING FOR EACH OF THE ATTACKS WITHIN A TIME INTERVAL OF 2H STARTING AT $t_0$.

| Starting time | Attack | Duration |
|---|---|---|
| $t_0 + 0h00m$ | DoS11 | 3m |
| $t_0 + 0h10m$ | DoS53s | 3m |
| $t_0 + 0h20m$ | DoS53a | 10m |
| $t_0 + 0h40m$ | Scan11 | 3m |
| $t_0 + 0h50m$ | Scan44 | 3m |
| $t_0 + 1h00m$ | Botnet | 1h |

TABLE III
DATE AND TIME FOR THE EXECUTION OF THE DIFFERENT ATTACK BATCHES IN THE DATASET.

| Date | Planned | Random |
|---|---|---|
| Thu, 07/28/2016 | 00:00 | 12:00 |
| Fri, 07/29/2016 | 02:00 | 14:00 |
| Sat, 07/30/2016 | 04:00 | 16:00 |
| Sun, 07/31/2016 | 06:00 | 18:00 |
| Mon, 08/01/2016 | 08:00 | 20:00 |
| Tue, 08/02/2016 | 10:00 | 22:00 |
| Wed, 08/03/2016 | 12:00 | N/A |
| Thu, 08/04/2016 | 14:00 | 00:00 |
| Fri, 08/05/2016 | 16:00 | 02:00 |
| Sat, 08/06/2016 | 18:00 | 04:00 |
| Sun, 08/07/2016 | 20:00 | 06:00 |
| Mon, 08/08/2016 | 22:00 | 08:00 |
| Tue, 08/09/2016 | N/A | 10:00 |

## C. Dataset Captures

The flows are captured and transferred using Netflow v9 format from the indicated sensors in the network (see Fig. 3). Default parameters are maintained for the Cisco netflow configuration of the routers, *i.e.*, inactive timer is 15 seconds, while active timer for flows is 30 minutes.

The complete dataset contains two different captures: a *calibration set* and a *test set*. The *calibration set* lasts 100 days, and ranges from 10:47-03/18/2016 until 18:27-06/26/2016. Its primary purpose is to assist in building and calibrating normality models, mainly because no attacks were synthetically generated. Note that this does not imply that no attacks are present, as we will discuss in what follows.

Although the recording of the *calibration set* was automatically programmed to be continuous over time, it was interrupted twice by the ISP to carry out specific network maintenance procedures. The two gaps are:

1) [02:00 03/27/2016 — 03:00 03/27/2016]

TABLE IV
FEATURES OF THE CALIBRATION AND THE TEST SETS.

| Feature | Calibration | Test |
|---|---|---|
| Capture start | 10:47h 03/18/2016 | 13:38h 07/27/2016 |
| Capture end | 18:27h 06/26/2016 | 09:27h 08/29/2016 |
| Attacks start | N/A | 00:00h 07/28/2016 |
| Attacks end | N/A | 12:00h 08/09/2016 |
| Number of files | 17 | 6 |
| Size (compressed) | 181GB | 55GB |
| # Connections | $\approx$ 13,000M | $\approx$ 3,900M |

2) [00:00 04/01/2016 — 17:20 04/06/2016]

The *test set* approximately lasts one month, from 13:38-07/27/2016 to 09:27-08/29/2016. During this capture, the attacks batches are executed starting at 00:00-07/28/2016, finishing at 12:00-08/09/2016, as it is shown in Table III. This capture is intended to be used for the validation of detection algorithms.

Table IV summarizes different features for both the *calibration* and the *test* sets. We can see the dates for the two captures and the number and size of the files included in each one. In addition, the number of connections included in the two sets is shown. This number is much higher than in datasets from other authors.

## D. Dataset Preprocessing and Availability

The collected `nfcapd` binary format files for both the calibration and the test sets are grouped in a single file per week for the two capture periods. The average size of the different files is around 14GB in compressed tar format. In the calibration set we have 17 files, while 6 files are available for the test set. All these files are available for downloading from our website: http://nesg.ugr.es/nesg-ugr16.

We share the netflow information in both `nfcapd` format and a preprocessed *csv* format obtained by using the `nfdump` tool. The `nfcapd` format contains all the netflow features, so that all of them could be consider by any IDS. For the *csv* format we have selected the most common features used in previous datasets[2]: timestamp of the end of a flow (*te*), duration of flow (*td*), source IP address (*sa*), destination IP (*da*), source port (*sp*), destination port (*dp*), protocol (*pr*), flags (*flg*), forwarding status (*fwd*), type of service (*stos*), packets exchanged in the flow (*pkt*), and their corresponding number of bytes (*byt*).

The IP addresses of the different machines in the dataset have been anonymized by using the CryptoPan prefix-preserving anonymization [27], implemented in the `nfanon` tool [28]. This tool has been traditionally used for the anonymization of the traces in the CAIDA datasets. The correspondence of the anonymized IP addresses in the different victim and attacker networks are shown in Table V.

[2]We indicate with parentheses the name of the variables as termed by the `nfdump` tool to ease the identification in the dataset.

TABLE V
CORRESPONDENCE OF ANONYMIZED IP ADDRESS WITH MACHINES IN THE EXPERIMENTAL SETUP.

| Machine/s | IP Address/es |
|---|---|
| $A_1$ - $A_5$ | 42.219.150.{*246,247,243,242,241*} |
| $V_{11}$ - $V_{15}$ | 42.219.156.{*30,31,29,28,27*} |
| $V_{21}$ - $V_{25}$ | 42.219.158.{*16,17,18,19,21*} |
| $V_{31}$ - $V_{35}$ | 42.219.152.{*20,21,22,23,18*} |
| $V_{41}$ - $V_{45}$ | 42.219.154.{*69,68,70,71,66*} |

## V. DATASET ANALYSIS AND LABELING

Now we first analyze the data collected in our two captures of the dataset, calibration and test, in order to give some insights and numbers that would help to assess its usefulness for IDS evaluation. Then, we also describe the labeling process, indicating the procedures followed and their limitations.

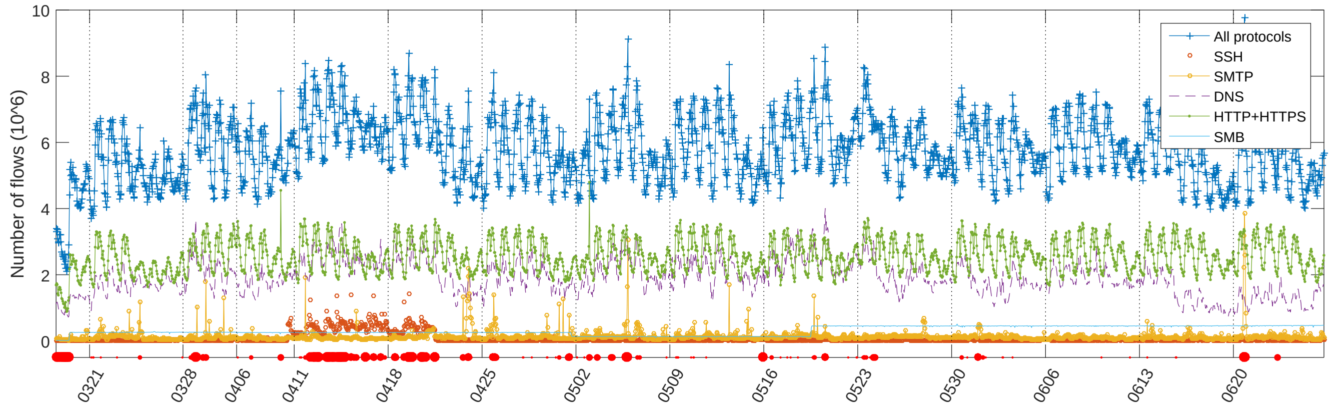
### A. Dataset figures

As shown in Table IV, the total number of flows in the dataset is above 16,900M, and the duration of the trace is longer than 4 months. This allows the evaluation of detection algorithms that leverage the cyclostationary evolution of traffic in different daytime/night patterns, as well as weekday/weekend phases. In addition, the number of external IPs observed in the trace is higher than 600M, corresponding to around 10M different (sub)networks.

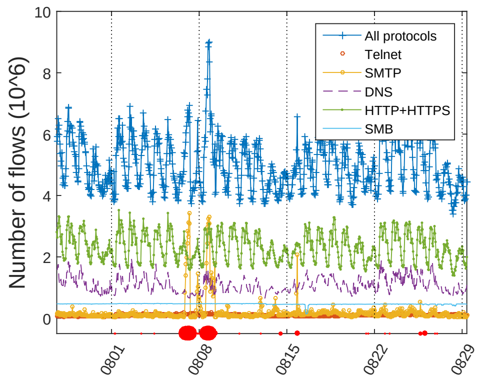
In Fig. 4 we can see the evolution of the number of flows (in million of flows) for both the calibration and the test sets. Vertical dashed lines separate the different weeks (and thus files) in the dataset. The protocols with the greatest amount of flows are differentiated. The red dots in the X-axis represent the anomalies found by three state-of-the-art anomaly detectors, where the size of the circle is related to the number of anomalies in the interval (see the discussion of this point in Section V-B).

There are some interesting insights from these results:

- We confirm that, as expected, HTTP/S protocol is predominant and the associated DNS traffic follows a coherent pattern.
- Due to the fact that most of the clients are companies that allocate their servers in the network, the traffic from BitTorrent or other P2P services is almost residual.
- Note that there is an increase in SSH traffic in the interval 04/10/16 – 04/21/16. As we will show in Section V-B, this corresponds to SSH scan attacks.
- Surprisingly, considering its insecure nature, we observe that many companies still use the Telnet service to manage their equipments (see Fig. 4(b)).
- There are peaks in the SMTP traffic spread out over time in the two sets. These peaks are sometimes email campaigns coming from legitimate companies (banks, online services, etc.), but we have also found spam campaigns (see Section V-B for more details).
- Note the constant pattern for SMB traffic. This traffic corresponds to a single retailing company with many



(a) Calibration Set



(b) Test Set

Fig. 4. Evolution of the number of flows for (a) calibration set and (b) test set for intervals of one hour. The red circles in X-axis represent the anomalies found in background traffic by three state-of-the-art anomaly detectors, where the size of the circle is related to the number of anomalies in the interval.

shops. Every establishment communicate with a central server periodically to check the status of some service or data. The peaks in the SMB traffic are due to the unsubscription of certain shops to that service.

## B. Labeling

Labeling is a critical issue in the production of datasets for IDSs evaluation. When dealing with real traffic, it is a real challenge to decide whether a given flow corresponds to an attack or not. Even when experts are manually assessing the nature of a flow there are certain cases in which this is not clear. For this reason, as discussed in Section II, many authors decide to generate only synthetic traffic, as only in these environments we are able to certainly determine which flows are due to an attack and which are legitimate. The problem with this approach is that background traffic is not representative of real networks behavior, thus opening the possibility to IDS algorithms being biased towards detecting fake scenarios.

In our case, as previously discussed, we have opted for using synthetic attack traffic interlaced with real background traffic. This fact poses a problem when trying to label the dataset. Background traffic is not clean from attacks that could be executed during the recording of the dataset and, as said, it is not clear how to identify them in many cases. This is not the case of our synthetic traffic. Obviously, we are able to label these flows as we know the rules to identify them (IP addresses and ports, timestamps, etc.).

Some authors [14] that use the same approach as ours decided to label flows or packets by using three different tags: a) an *attack* label for the flows that they positively know that correspond to an attack, b) a *normal* label for those that are generated synthetically with normal patterns, and c) a *background* label for those which no one knows exactly if they are attacks or not. In our case, we are not synthetically generating normal traffic, so that we label as *background* those traffic flows that are not proven to be attacks (according to the criteria given in what follows).

Labels are applied in a per-flow basis. It is worth to indicate that connections are two-way, so that a single connection appears as two different one-way connections. In order to identify the attacks within the background traffic our strategy is to perform both signature and anomaly detection on the dataset.

**Signature-based labeling.** First, when possible, we identify attacks by their known signatures, providing labels for the different attacks. This is the case of the attacks that we generate synthetically, since their signatures are known, as described

TABLE VI
SUMMARY OF THE PUBLIC BLACKLISTS USED FOR SIGNATURE IDENTIFICATION OF ATTACK FLOWS IN THE TRACE.

| Blacklist | Target | Source |
|---|---|---|
| hpHosts | Malware, Fraud, Ad tracking | [29] |
| Malware DL | Malware | [30] |
| Spamhaus | SPAM | [31] |
| Abuse.ch | Ransomware, Malware, C&C | [32] |

in Section IV-B. Note that not all the attacks implemented (see Section III.B) have been individually labeled. Instead, some of them have been grouped, due to the fact that their signatures are actually the same, *e.g.*, *DoS11*, *Dos53s* and *DoS53a* are all labeled as "dos".

In addition, we have selected the IP addresses in the dataset that were published in the public blacklists indicated in Table VI during the period of time that ranges from two months before the capture until two months after. The total amount of IP addresses in the dataset identified in the blacklist is 10.175, including only one of the IP addresses in the ISP range. Note that although the flows involving IPs in the blacklists are labeled, when working with the dataset we should be aware that not all of them are necessarily attacks.

**Anomaly-based labeling.** Second, we have performed anomaly detection to identify other possible attacks in the background traffic.

We have evaluated the dataset by using three different state-of-the-art anomaly detectors. The first pair of detectors, MSNM$_C$ and MSNM$_S$ are proposed in the PCA-based multivariate statistical network monitoring methodology [33]. MSNM$_C$ uses a mean centering approach for the features, while *MSNM$_S$* autoscales them before applying results. They exhibit slightly different performance depending on the considered type of attacks [33]. These detectors have proven to get an improved behavior in several different scenarios [34].

The third anomaly detector is the one-class support vector machine (OCSVM) [35] [36]. The OCSVM is a classification-based network anomaly detection method [37] reported to provide excellent results.

We have focused on the most relevant anomalies where the three detectors reach a consensus, and a manual analysis has been performed to extract a signature from them. Following this procedure, we have identified three relevant attacks:

- *UDP Scan attack.* The anomaly takes place in the short interval from 04:10-08/01/16 to 04:14-08/01/16. Here, we find an increase of ACK packets and very short connections using UDP. Inspecting the Netflow records for the time period we find a single IP from Germany creating 867,405 connections from only four origin ports (5061, 5062, 5066 and 5068). The destinations are 4,097 different hosts distributed in 16 different subnets (/24 mask). Depending on the source port of the connection, each victim host is scanned through a specific range of 60 ports (*e.g.* from source port 5068, ports 5000-5059 were scanned). Due to this pattern of connections, we conclude that it seems to be a malware driven scanning for a specific vulnerability. Thus, we label as "anomaly-udpscan" all these flows.
  Let us illustrate our manual analysis for this anomaly as an example. Our intention is only to motivate other researchers for the identification of these anomalies by using their own methods and detection algorithms. We have used an ad hoc visual analytics tools aimed at describing, exploring and analyzing data in order to discover underlying knowledge from data [38]. In addition, we have counted with the help of the ISP staff.
  In Figs. 5 to 7 we can see the evolution of the traffic for this anomaly using a multiple linked view approach, consisting of a timeline bar chart and an enhanced hive plot [39]. Fig. 5 shows the temporal evolution of the number of UDP connections during the anomaly period. We see that there are 5 different bursts of traffic from the German IP address to the ISP victims of the scan. We can visualize this evolution using animated hive plots, thus discovering patterns that, in turn, can be inspected in detail. This tool allows us to visualize a simulation of the recorded traffic in such a way that the visual identification of anomalies can easily be discovered in a preattentive way [40].
  In Fig. 6 we see a snapshot of this tool for a set of connections. In the figure, the different IPs observed in the connections are represented as points in the axes, such that the size of the point depends on the number of connections in which that IP is involved. In our case, we have split the representation in 3 different axes, where we have selected the most active IPs in the ISP range (x-axis), other IPs in Spain[3] (y-axis) and the rest of the world IPs (z-axis). The lines connecting IPs from different or same axes represent the connections. The color of lines indicates the source port of the connections.
  Fig. 6 illustrates the visual pattern conveyed by the tool for the normal behavior period during the anomaly (see this period in Fig. 5). We check that the connections observed here have a pattern that is very widespread among the different IPs. The variety of colors in the lines also show that there are no common patterns for the source port. As a contrast, we represent in Fig. 7 the snapshots of the tool for the second and fifth burst in the anomaly, where 30,662 and 35,303 connections in just a second were made respectively (see Fig. 5). In the figure, we can clearly identify a pattern related to the fact that abnormal traffic is being generated. First, a single IP is generating most of the connections. Second, in every burst we see that the same source port is used. This is identified by the colors (pink and grey, respectively) of the different connections, indicating that almost all connections in that burst have the same port as origin.
- *SSH Scan attack.* As we can see in Fig. 4(a), there is an increase in SSH traffic in the interval 04/10/16 – 04/21/16. We have manually checked that this increment in the traffic is generated by an SSH scan attack coming from a single machine hosted in the ISP (anonymized

[3]Where the ISP belongs to.

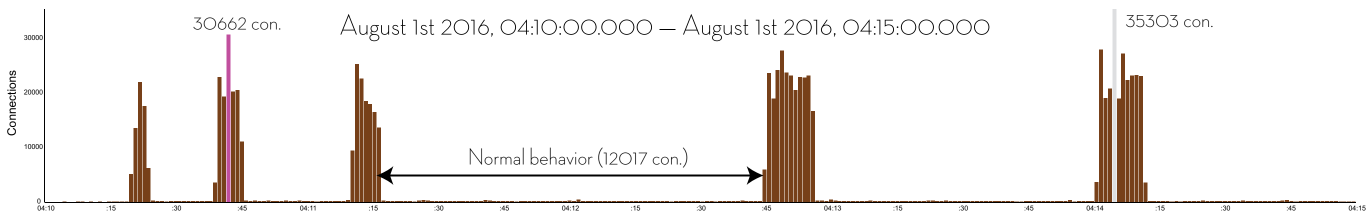


Fig. 5. Time distribution of traffic bursts for the anomaly.

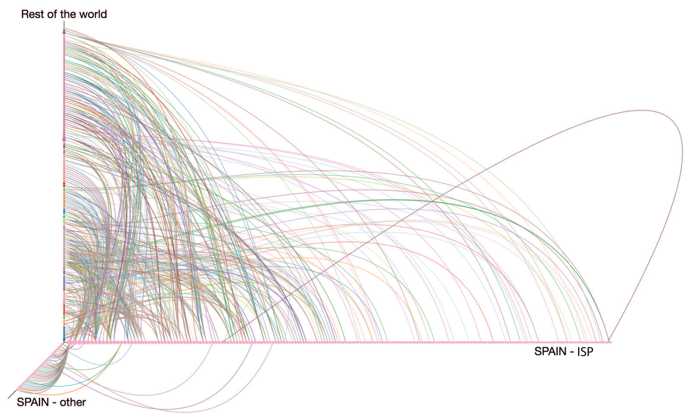
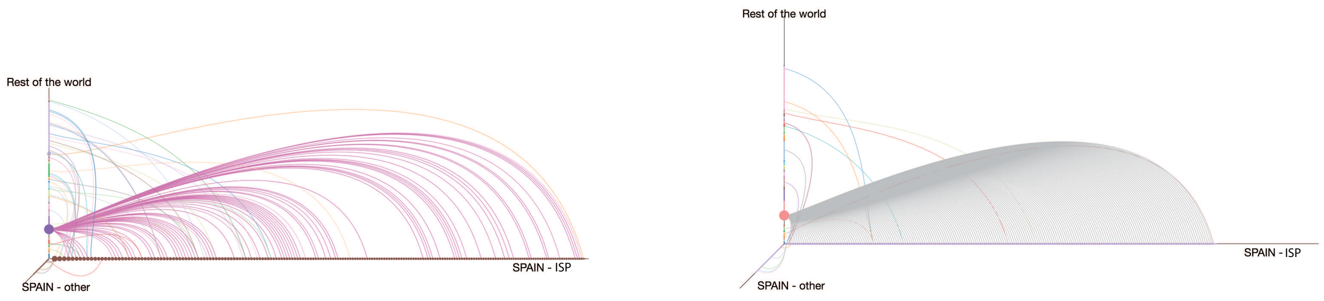


Fig. 6. Visualization of the traffic during the normal behaviour period in the anomaly.



(a) Second burst snapshot

(b) Fifth burst snapshot

Fig. 7. Visualization of two different bursts of traffic in the anomaly.

42.219.156.231). The victims of the attack have a wide range of IPs located in a South American country. In addition, this machine is also generating an unusually high load of scanning connections to other services like DNS, HTTP and others. For this reason, we label the connections involving this IP as "anomaly-sshscan"

- *SPAM attack.* We identify peaks of SMTP traffic in the background traffic in both normal and attack sets (see red dots in Fig. 4). This traffic is generated from 5 public IPs (anonymized 42.219.156.212-215 and 42.219.156.223) and sent to Yahoo email servers. A close observation of this traffic leads to conclude that it follows the pattern of a spam campaign. As an example, in the period 20:39-08/06/16 – 05:59-08/07/16 we find around 12.5M of SMTP connections. Each machine generated around 2.5M e-mails. The ISP IT team confirmed that this was a client that hired virtual machines during 2 months. The hired IPs finished in the Yahoo blacklists, and that is probably the reason why the client stopped hiring the virtual machines. Thus, we label as "anomaly-spam" all the connections to the SMTP port generated by these 5 IPs in the whole dataset.

We split attack labels in different sublabels, depending on the type of attack being executed. Table VII shows the correspondence between the labels we have used in the dataset with the type of attack.

## VI. Conclusion

In this paper, the main contribution is a new dataset for the evaluation of IDS algorithms and systems called UGR'16. The dataset has been designed to allow the training of anomaly detection algorithms that consider the cyclostationary nature of traffic data. UGR'16 is a collection of netflow traces for more than 4 months of traffic in a real network from a tier-3 ISP, together with a set of realistic networking attacks that we have specifically designed to train and test IDS algorithms.

TABLE VII
CORRESPONDENCE AND CLASSES OF THE LABELS TO THE ATTACKS FOUND IN THE TRACE.

| Type of attack | class | Label |
|---|---|---|
| DoS11 | signature (synthetic) | dos |
| DoS53s | signature (synthetic) | dos |
| DoS53a | signature (synthetic) | dos |
| Scan11 | signature (synthetic) | scan11 |
| Scan44 | signature (synthetic) | scan44 |
| Botnet | signature (synthetic) | nerisbotnet |
| IP in blacklist | signature | blacklist |
| UDP Scan | anomaly | anomaly-udpscan |
| SSH Scan | anomaly | anomaly-sshscan |
| SPAM | anomaly | anomaly-spam |

In addition to the comprehensive description and analysis of the dataset, we have made an extensive review and comparison of existing datasets, pointing out the main strengths and weaknesses of the different approaches. From this study, a set of requirements to be met by modern IDS evaluation datasets have been extracted.

In addition to the fact that the duration of the dataset makes it adequate to test algorithms that consider the cyclostationary evolution of traffic in daytime/night and weekdays/weekends, the main advantage of this dataset over others is that the background traffic is very representative of the Internet traffic, as it is captured from sensors in a ISP network where many different profiles of clients are located. This is a main difference with other datasets in which very specific traffic is collected (*e.g.* traffic from a university or a research center).

Finally, the dataset is publicly available at the authors' research group website.

## ACKNOWLEDGEMENTS

This work has been partially supported by Spanish Government-MINECO (Ministerio de Economía y Competitividad) and FEDER funds, through project TIN2014-60346-R.

## REFERENCES

[1] R. Di Pietro, L. V. Mancini, Intrusion detection systems, Vol. 38, Springer Science & Business Media, 2008.

[2] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, E. Vázquez, Anomaly-based network intrusion detection: Techniques, systems and challenges, computers & security 28 (1) (2009) 18–28.

[3] DARPA'98 and DARPA'99 datasets.
URL https://www.ll.mit.edu/ideval/docs/index.html

[4] CAIDA. The Cooperative Association for Internet Data Analysis.
URL http://www.caida.org/

[5] KDD Cup 99 Dataset.
URL https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[6] J. Stolfo, W. Fan, W. Lee, A. Prodromidis, P. K. Chan, Cost-based modeling and evaluation for data mining with application to fraud and intrusion detection, Results from the JAM Project by Salvatore (2000) 1–15.

[7] J. McHugh, Testing Intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory, ACM Transactions on Information and System Security 3 (4) (2000) 262–294.

[8] M. V. Mahoney, P. K. Chan, An analysis of the 1999 DARPA/Lincoln Laboratory evaluation data for network anomaly detection, in: International Workshop on Recent Advances in Intrusion Detection, Springer, 2003, pp. 220–237.

[9] M. Tavallaee, E. Bagheri, W. Lu, A. A. Ghorbani, A detailed analysis of the KDD CUP 99 data set, IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009 (Cisda) (2009) 1–6.

[10] I. Perona, I. Gurrutxaga, O. Arbelaitz, J. I. Martín, J. Muguerza, J. M. Pérez, Service-independent payload analysis to improve intrusion detection in network traffic, in: Proceedings of the 7th Australasian Data Mining Conference-Volume 87, Australian Computer Society, Inc., 2008, pp. 171–178.

[11] A. Sperotto, R. Sadre, F. Van Vliet, A. Pras, A labeled data set for flow-based intrusion detection, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 5843 LNCS (2009) 39–50.

[12] R. Fontugne, P. Borgnat, P. Abry, K. Fukuda, MAWILab: Combining Diverse Anomaly Detectors for Automated Anomaly Labeling and Performance Benchmarking, in: ACM CoNEXT '10, Philadelphia, PA, 2010.

[13] A. Shiravi, H. Shiravi, M. Tavallaee, A. A. Ghorbani, Toward developing a systematic approach to generate benchmark datasets for intrusion detection, Computers and Security 31 (3) (2012) 357–374.

[14] S. Garcia, M. Grill, J. Stiborek, A. Zunino, An empirical comparison of botnet detection methods, Computers & Security 45 (2014) 100 – 123.

[15] G. Creech, J. Hu, Generation of a new IDS test dataset: Time to retire the KDD collection, IEEE Wireless Communications and Networking Conference, WCNC (2013) 4487–4492.

[16] Metasploit penetration testing software.
URL https://www.metasploit.com/

[17] 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) 1–6.

[18] A. Lemay, J. M. Fernandez, Providing scada network data sets for intrusion detection research, in: 9th Workshop on Cyber Security Experimentation and Test (CSET 16), USENIX Association, 2016.

[19] T. Morris, W. Gao, Industrial control system traffic data sets for intrusion detection research, in: International Conference on Critical Infrastructure Protection, Springer, 2014, pp. 65–78.

[20] R. Koch, M. Golling, G. D. Rodosek, Towards Comparability of Intrusion Detection Systems: New Data Sets, TERENA Networking Conference (2014) 7.

[21] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, X. Xiao, Rfc3272. overview and principles of internet traffic engineering, Tech. rep. (2002).

[22] M. Z. Shafiq, L. Ji, A. X. Liu, J. Wang, Characterizing and modeling internet traffic dynamics of cellular devices, in: Proceedings of the ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems, ACM, 2011, pp. 305–316.

[23] A. Cuadra-Sánchez, J. Aracil, Traffic anomaly detection, Elsevier, 2015.

[24] A. Cuadra, J. Aracil, Context-aware security framework based on traffic anomaly detection indicator, Telecommunication Systems 65 (2) (2017) 319–330.

[25] X. Xu, Sequential anomaly detection based on temporal-difference learning: Principles, models and case studies, Applied Soft Computing 10 (3) (2010) 859–867.

[26] CTU-13 dataset.
URL https://stratosphereips.org/category/dataset.html

[27] J. Fan, J. Xu, M. H. Ammar, S. B. Moon, Prefix-preserving IP address anonymization: measurement-based security evaluation and a new cryptography-based scheme, Computer Networks 46 (2) (2004) 253–272.

[28] P. Haag, NFDUMP-NetFlow processing tools, URL: http://nfdump.sourceforge.net.

[29] Domain and IP blacklist: hpHosts.
URL https://www.hosts-file.net/

[30] Domain and IP blacklist: Malware Domain List.
URL https://www.malwaredomainlist.com/

[31] Domain and IP blacklist: SpamHaus.
URL https://www.spamhaus.org/

[32] Domain and IP blacklist: abuse.
URL https://abuse.ch/

[33] J. Camacho, A. Pérez-Villegas, P. García-Teodoro, G. Maciá-Fernández, PCA-based multivariate statistical network monitoring for anomaly detection, Computers & Security 59 (2016) 118–137.

[34] G. Maciá-Fernández, J. Camacho, P. García-Teodoro, R. A. Rodríguez-Gómez, Hierarchical PCA-based multivariate statistical network monitoring for anomaly detection, in: Information Forensics and Security (WIFS), 2016 IEEE International Workshop on, IEEE, 2016, pp. 1–6.

[35] B. Scholkopf, A. J. Smola, R. C. Williamson, P. L. Bartlett, New Support Vector Algorithms, Neural computation 12 (5) (2000) 1207–1245.

[36] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, R. C. Williamson, Estimating the Support of a High-Dimensional Distribution, Neural Computation 13 (7) (2001) 1443–1471.

[37] M. H. Bhuyan, D. K. Bhattacharyya, J. K. Kalita, Network Anomaly Detection: Methods, Systems and Tools, IEEE Communications Surveys & Tutorials 16 (1) (2014) 303–336.

[38] Juan Alvarado-Pérez and Diego H. Peluffo-Ordóez and Roberto Therón, Bridging the gap between human knowledge and machine learning, Advances in Distributed Computing and Artificial Intelligence Journal 4 (1) (2015) 54–64.

[39] M. Krzywinski, I. Birol, S. J. Jones, M. A. Marra, Hive plots-rational approach to visualizing networks, Briefings in bioinformatics 13 (5) (2012) 627–644.

[40] A. Treisman, Preattentive processing in vision, Computer vision, graphics, and image processing 31 (2) (1985) 156–177.