0day anomaly detection through machine learning

Johan Mazel (jmazel@laas.fr) *,†
Philippe Owezarski (owe@laas.fr) *,†
Yann Labit (ylabit@laas.fr) *,†

Abstract: This paper proposes new cognitive algorithms and mechanisms for detecting 0day attacks targeting the Internet and its communication performances and behavior. For this purpose, this work relies on the use of machine learning techniques able to issue autonomously traffic models and new attack signatures when new attacks are detected, characterized and classified as such. The ultimate goal deals with being able to instantaneously deploy new defense strategies when a new 0day attack is encountered, thanks to an autonomous cognitive system. The algorithms and mechanisms are validated through extensive experiments taking advantage of many real traffic traces captured on the Renater network as well as on the WIDE transpacific link between Japan and the USA.

Keywords: 0day anomaly detection, machine learning

1 Introduction

Security in the Internet is a very important and strategic problem which raised and still raises significant research and engineering effort, but need continuously to be addressed. The main reason is that the threat in the Internet is moving fast: new kinds of attacks, worm, viruses appear almost every day, they use more and more advanced spreading and corruption strategies, and act so as to remain very hardly detectable. One of the problems then stands in detecting the new attacks (also called 0day - or 0d for short - attacks) the first time they are perpetrated. Current systems are unable of detecting such 0d attacks. When they are first observed, engineers first need to analyze them before searching for a detection and defense strategy, implement it, and finally deploy it. This is a reactive process which let the network vulnerable to some attacks for a too long period.

In this paper, we present our first work on designing new cognitive strategies and algorithms for detecting 0day attacks the first time they are perpetrated in the Internet. The idea is to design autonomous cognitive systems able to increase autonomously their knowledge database on attacks. As the object under concern in our research is the Internet, we will specifically focus on volume based DoS (Denial of Service) attacks which aim at decreasing network QoS (Quality of Service) and performance level by denying the access to network resources for legitimate users. In networking, such DoS attacks are part of a broader family of unwanted events called traffic anomalies. We then aim at designing a new cognitive system which is essentially able to autonomously classify anomalies in different

* CNRS; LAAS; 7 Avenue du colonel Roche, F-31077 Toulouse, France
† Université de Toulouse; UPS, INSA, INP, ISAE; LAAS; F-31077 Toulouse, France

categories. The idea is then to give the cognitive system the capability to analyze the anomaly for discovering whether it is legitimate or not, but also to autonomously extend the attack signature database of the related anomaly detection system (ADS) if the new encountered anomaly is classified as an attack which is not known by the system. For this purpose, our algorithm relies on the use of machine learning techniques for autonomously issuing models of normal traffic, as well as attack signatures when attacks are encountered for the first time. This signature is then prone to be integrated in an associated defense system (whose description is out of the scope of this paper). This approach allows a significant reduction of the time the network is not protected against a new attack as it takes a short time to issue a new detection signature for classical IDS (Intrusion Detection System) or IPS (Intrusion Protection System) which can be immediately and automatically deployed.

The paper is organized as follows. Section 2 provides an overview on related work. Section 3 gives a short overview of main machine learning techniques, and justifies our choice of using unsupervised techniques. Section 4 then presents how the new detection and classification cognitive algorithm works, the attributes we derive and how the different types of anomalies can be characterized. In Section 4, the validation data and methodology are presented, the results of evaluation being described and discussed in Section 5. Section 6 then concludes the paper.

2 Related work

There is now a large literature on the detection of network traffic anomalies. Most of the approaches analyze statistical variations of traffic volume (i.e. number of packets, bytes or new flows), traffic attributes (i.e. IP addresses and ports) distributions, or both, on a temporal or spatial manner. The anomalies can be observed from single links or network-wide data. Standard references include [Bru00, BKPR02, KSZC03, LCD04b, LCD04a], with some notable recent work as [LBC+06, DFB+07, CSKC08, SLO+07]. Dimensionality reduction of aggregated traffic data has also received recent attention, and techniques like sketches [KSZC03, LBC+06, DFB+07] and principal components analysis [LCD04b] are very promising for online anomaly detection. Sketches based algorithms can detect low intensity anomalies and can identify the anomalous IP flows (something that might not be possible with techniques that operate only on the aggregated traffic or on origin-destination flows).

In this work, we base our research on the anomaly detection approach presented in [OF09]. Its presents a two steps anomaly detection and classification algorithm that we will present later (see 4.1).

Some works have tried to apply machine learning techniques to anomaly or intrusion detection. Kuang and Zulkernine [KZ08] used a modified KNN algorithm called CSI-KNN for Combined Strangeness and Isolation measure K-Nearest Neighbors. They perform supervised learning on the KDD dataset [KDD99]. They use the feature provided by the dataset to generate two values (strangeness and isolation). These values are then processed to generate a graded confidence over the classification. Some papers push forward the use of machine learning with the goal to classify automatically the traffic [LCD05] or to discover new anomalies [EAP+02] and [Por01]. In [LCD05], Lakhina et al use clustering on the entropy of several parameters (IP addresses and ports). This approach groups anomalies with similar characteristics, but does not distinguish between different types of anomalies. Network operators still need to manually check each anomaly, but, if enough pre-labeled anomalies are part of a given cluster, they have a better way to prioritize between clusters than if no classification is done. In [EAP+02] and [Por01], Eskin et al use unsupervised learning to detect new intrusions inside the KDD dataset. However, it remains a work mainly oriented on intrusion detection and not network anomalies. The authors even consider their work as inoperant in the case of Syn flood DDoS anomalies.

3 Introduction to machine learning

3.1 Brief state of the art of machine learning

Machine learning is composed of two main groups of techniques : supervised/semi-supervised techniques and unsupervised. The two next sub-subsections will survey these two techniques.

3.1.1 Supervised and semi-supervised learning

- Principle

  Supervised learning is able to establish signatures from labeled training data. The training data are made of pairs of objects. Each pair is composed of a multi-dimensional values and the associated output. The predictive model obtained through the learning is able to give an output value from any multi-dimensional input value.

  Semi-supervised learning consists of two steps. The first phase is supervised learning which is used to create a rough scheme of what the model is going to be. The second phase is meant to improve the result of the first phase through the use of unlabeled data.

  Supervised learning can be realized through different algorithms such as decision tree, support vector machines...

- Example of supervised learning: decision tree

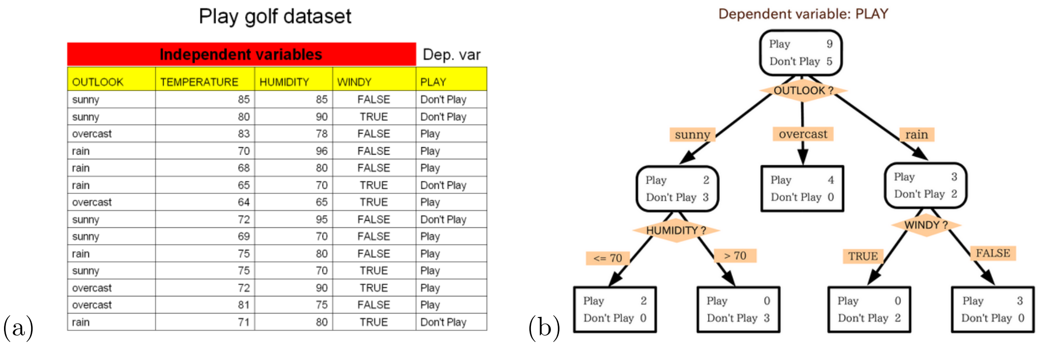
  We want to build a model to describe the behavior of the golf players according to several parameters: temperature, humidity and wind. We choose to use the decision tree algorithm. Figure 1 (a) gives the set of training data that will be used.

  The algorithm uses all the data to create the signatures that model the behavior of the player according to these parameters. The result is depicted on figure 1 (b), a tree that describes the player behavior.

3.1.2 Unsupervised learning

Unsupervised learning aims at establishing how a dataset is structured. In order to do so, unsupervised learning use unlabeled data and tries to find structures, key features or models that can describe the way the training data is organized and thus the way the system behaves.

There are different techniques of unsupervised learning among which we can quote dimensional reduction, density estimation, clustering, etc. Here is a description of these three techniques.



Play golf dataset

Independent variables				Dep. var
OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY
sunny	85	85	FALSE	Don't Play
sunny	80	90	TRUE	Don't Play
overcast	83	78	FALSE	Play
rain	70	96	FALSE	Play
rain	68	80	FALSE	Play
rain	65	70	TRUE	Don't Play
overcast	64	65	TRUE	Play
sunny	72	95	FALSE	Don't Play
sunny	69	70	FALSE	Play
rain	75	80	FALSE	Play
sunny	75	70	TRUE	Play
overcast	72	90	TRUE	Play
overcast	81	75	FALSE	Play
rain	71	80	TRUE	Don't Play

Fig. 1: Training data (a) and model of the golf player behavior (b) [1].

- Dimensional reduction tries to project the data from a set of great dimensions to a set with smaller dimensions. Let's quote few examples of methods of dimensionality reduction as for examples principal component analysis (PCA) [I.T02], multidimensional scaling (MDS) [CC94].

- Density estimation is a family of methods for "one-class" problems. We assume that there is a set of representative observations which is part of a single class. The general objective of this method is to estimate the distribution of the observations and then predict whether or not a new observation should be considered as an outlier or a "normal" member of the single class. For example, Bayesian network, mixture models, etc. are density estimation methods.

- Cluster analysis or clustering is the assignment of a set of observations into subsets (called clusters) so that observations in the same subset are similar in some senses. Let's quote a few examples of clustering algorithms like hierarchical [War63], partitional (eg.: k-means algorithm [Mac03]).

3.2 Choice between supervised/semi-supervised and unsupervised learning for 0day anomaly detection

Supervised/semi-supervised learning presents limitations because their use implies that we have labeled data at our disposal. In our case, the training can be of several types. It can be traces where all the anomalous packets have been previously marked. It may also be a trace where we know that at a certain time and for a certain duration an anomaly has occurred and has been captured. The process used to obtain these traces is long and tedious as it requires human intervention.

This whole process used to generate trace with labeled anomalies also implies that the considered anomalies are known and characterized. This excludes the goal addressed in this paper which deals with the real-time online discovery of 0day anomalies.

[1] http://commons.wikimedia.org/wiki/File:Golf_dataset.png and http://commons.wikimedia.org/wiki/File:Decision_tree_model.png

On the other hand, unsupervised learning does not present this limitation. In fact, its purpose is precisely to find structure inside unknown data. Therefore, unsupervised learning appears as the technique to use to avoid the limitation of the approach addressed in the previous paragraph.

4 Application of machine learning to 0day anomaly detection

4.1 Two steps approach

Because of the limits of both the profile and signature based approaches for detecting attacks and anomalies, a new trend deals with combining both of them in a two steps approach. In general, the flow of alarms provided by a signature based IDS is analyzed with a profile based method in order to detect anomalies in the alarm flow. Performance of such an approach is very low [Vii06]. We therefore argue that it is necessary to combine both detection techniques in a two steps approach. But we do think that the right approach deals with first using the profile based technique in order to detect traffic profile anomalies. In that case, the detection thresholds are set with very pessimistic values in order to avoid false negatives. Then, we apply a signature based detection technique which has also the capability of classifying the anomalies. It then helps to eliminate false positive, but also, by classifying the detected anomalies, to identify the kind of anomaly as well as the intension behind the anomaly (legitimate or malicious).

This paper then relies on our NADA [OF09] anomaly detection tool which has been designed following this two steps approach principle. The criterias used for the detection step are very simple and rough: it computes the number of packets, the number of bytes and the number of SYN packets. It monitors the evolution of these criterias and if a significant change is discovered, meaning it is over the specified threshold, it raises an alarm. It also relies on the use of some attributes built from the network traffic dimensions. These attributes up to now include the ones presented in table 1.

All these attributes are related to, either the detection step, either some indices built on network packets fields or directly from the network packets fields. All these attributes are then also used by the classification system. This system uses a set of signatures that have been established from expert knowledge. The signatures used during the classification step are presented in table 2.

These signatures use attributes directly linked to the packet headers and are thus easily understood by network operators. This is one of the key feature of this system.

These signatures have been built through expert knowledge in the domain of network traffic anomalies. Therefore, human intervention is required for the creation and tuning in order to work perfectly. The purpose of our method is to create these signatures automatically without human intervention in order to build a system that would be able to work autonomously. In order to achieve this goal, we plan to use machine learning. The two targets of our system are, first, update the thresholds used inside each signature and second, be able to create completely new signatures from scratch.

4.2 Representation of traffic

Let's introduce the term basis. In linear algebra, basis is the set of vector that can represent every vector in a given vector space through a linear combination. Dimension of vector space is the number of vector in the considered basis.

Attribute	Type	Description
found{p,b,s}	integer	If the corresponding metric was anomalous, value of P, zero otherwise
impactlevel{p,b,s}	integer	The impact level of the anomaly (see Section 3.1)
duration{p,b,s}	integer	For how many slots the metric stayed above the threshold
decrease{p,b,s}	float	The biggest negative deltoid during the anomaly as a fraction of the threshold
#respdest	integer	Number of responsible destinations
#rsrc/#rdst	integer	Ratio of responsible sources to responsible destinations
avg#rdstports	integer	Average number of responsible destination ports
avg#rsrcports	integer	Average number of responsible source ports
#rpkt/#rdstport	integer	Ratio of number of packets to responsible destination ports
#rpkt/#rsrc	integer	Average number of packets of responsible sources
bpprop	integer	Average packet size (only packets of the anomaly)
spprop	float	Ratio of number of syn to number of packets of the anomaly
samesrcpred	boolean	If a specific responsible source appears for the majority of destinations
samesrcportpred	boolean	If the majority of responsible sources use the same source ports
oneportpred	boolean	If only one destination port dominated
invalidpred	boolean	If the anomaly was mainly consisted of invalid packets (e.g. malformed headers)
invprotopred	boolean	If the anomaly was dominated by packets using invalid protocol numbers or types
landpred	boolean	If most packets had the same source and destination IPs
echopred	boolean	If most packets were of type ICMP Echo Request/Reply
icmppred	boolean	If most packets were ICMP of any other type
rstpred	boolean	If most packets were TCP with RST flag set

Tab. 1: Attributes derived from traffic in order to classify anomalies. p, b and s are for packets, bytes and syn respectively.

Id	Anomaly Type	Signature
1	ICMP Echo DDoS	#respdest ==1 and echopred and (#rpkt/#rdstport > 30*granularity or #rsrc/#rdest > 15)
2	TCP SYN DDoS	#respdest == 1 and founds and spprop > 0.9 and oneportpred and #rpkt/#rdstport > 10*granularity
3	Network Scan	#respdest > 200 and samesrcpred
4	SYN Port Scan	#respdest == 1 and #rsrc/#rdest == 1 and spprop > 0.8 and avg#rdstports > 5
5	Attack Response	#respdest == 1 and (rstpred or icmppred) and foundp > 20*granularity and (not (impactlevelp == 3)) and (#rsrc/#rdest == 1 or samesrcportspred)

Tab. 2: Examples of signatures used in this work.

Let's then consider each attributes of Table 1. If we segment the traffic into several parts, each part of the traffic has a defined value for this attribute.

Therefore, we can consider an attribute as a possible single vector of a basis of dimension 1. If we consider this basis, the traffic can then be represented inside this basis as several points. Corollary, if one considers several attributes forming another basis, the whole traffic (i.e. the set of all parts of the traffic) can then be projected inside a new vector space of dimension equal to the number of attributes used to form the basis.

Then, each point can either be associated to a model representing a traffic class (normal or anomalous) or, if it is considered too far from the model(s), let alone and be considered as an outlier.

This subsection addresses the way the traffic is represented inside this vector space and the way anomalous and normal traffic are represented, i.e. through class or outliers.

4.2.1 Basic principle

As we previously said, each point in the considered vector space represents a group of aggregated packets. The aggregation of the packets into groups is a crucial aspects of the way we want to represent the network traffic in the vector space. We will address this topic and how we deal with it in 4.4.

4.2.2 Representations of traffic

In all the previous work that we are aware of [LCD05, EAP$^+$02], two possible representations of network traffic through unsupervised learning have been considered.

In the first representation, the network traffic is represented by several classes and each class is associated with a part of the network traffic. By considering the figure 3 (a), it is possible to notice that there are several classes (two actually) that can represent either a class of normal traffic or a class of anomalous traffic. Considering figure (2 (a)), there is one (or more) cluster(s) for the normal network traffic (eg. here the cluster with the dots) and one (or more) cluster(s) for each type of anomaly present (eg. the clusters with squares and diamonds).

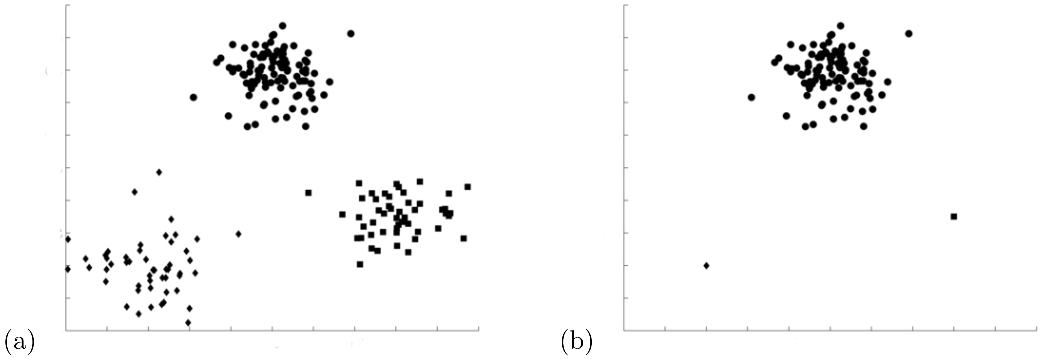


Fig. 2: Model with clusters (a) and model with outliers (b)

In the other representation, there is one or more classes for the normal traffic and any points that is not part of the class(es) is considered as an outlier and thus, part of the anomalous traffic. By considering the figure 3 (b), notice that there is one model (the only gaussian curve) that represents a single class of normal traffic. Any point located too far from the model is anomalous. Considering figure (fig 2 (b)), there is(are) several cluster(s) (in our example below, only one) for the normal traffic (here the cluster with dots) and each outlier represents an anomaly (the square dot and the diamond dot).

A part of our methodology is constituted by the analysis of the traffic to determine whether the traffic can be represented through one or several classes (see 5.3.1).

4.3 Choice between the different forms of unsupervised learning

In the next paragraphs, we are going to address the problem of the choice of the unsupervised techniques. We need a technique that can identify all the classes of traffic and that

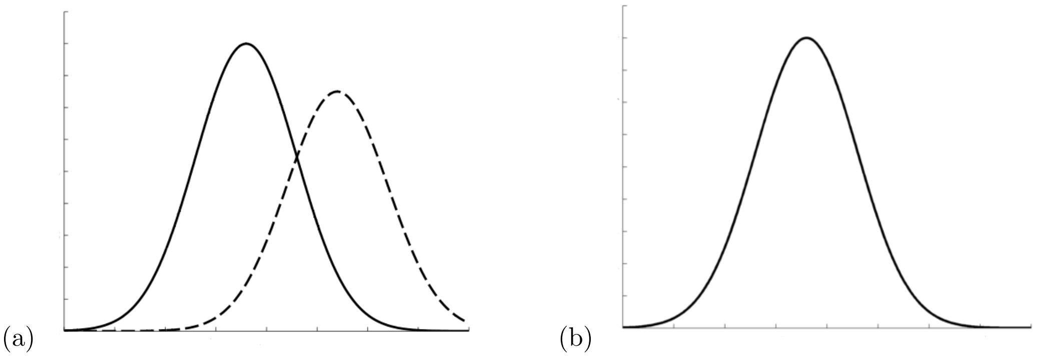


Fig. 3: One class (a) and two class (b) system

can keep some understandable attributes. We will only consider dimensional reduction, density estimation and clustering as they appear as the three most represented techniques in the literature.

- Dimensional reduction

  The principle of dimensional reduction deals with projecting the data from a vector space of high dimension to a vector space of low dimensions. In our case, it means that we would end up with a vector base with a basis built on vectors that would have no physical/concrete meaning. One of our goal being to keep some understandable attributes in order to have easy to understand and meaningful signatures (i.e. for expressing anomaly characteristics), the dimensional reduction is in clear contradiction with our requirements.

- Density estimation

  Density estimation is a family of methods for "one-class" problems. We assume that there is a set of observations who are part of a single class. The general objective is to estimate the distribution of the observations and then predict whether or not a new observation should be considered as an outlier or a "normal" member of the single class.

  In the case of figure 3 (b), the single class considered can be the one of the normal traffic. Anomalies would then be represented as outliers (points that are too far from the "model"). Density estimation is an efficient technique to detect outlier if the normal traffic is a single class (i.e. not composed of several classes) and that anomalies are only outliers (i.e. not one or more class(es)).

  In the case of figure 3 (a), we could have for example, two classes for the normal traffic (normal and dashed curves) and the anomalies would be the outliers. We could also have one class for the normal traffic (normal curve) and one class for the anomalous traffic (dashed curve). In both of these cases, density estimation would be inoperant since it is unable to consider several classes.

  Therefore, this form of unsupervised learning seems to not be adapted to our case as it is not possible to guarantee that we will be in the case of Figure 3 (b).

- Clustering

  Cluster analysis or clustering is the assignment of a set of observations into subsets (called clusters) so that observations in the same cluster are similar under some chosen criterias.

  Clustering does not have any of the limitations listed above: it keeps all the attributes in a clear and intelligible form and it can consider and analyze them without any limitation on the number of classes (in our case, the number of classes of traffic including both normal and anomalous ones). Based on our first experiences, we selected the clustering technique as it appears as the most adapted and promising form of unsupervised learning for our problem.

4.4 Discovering unknown new anomalies with machine learning

In the previous subsection, we established two facts. First, it is possible to represents the traffic inside a vector space built on attributes. Second, unsupervised learning is able to extract the structure of a dataset from its representation in a vector space.

The interest of the extracted structure is directly linked to the pertinence of the considered vector space, i.e. the considered attributes. In our case, this pertinence is also related to the choice of two parameters: first, the aggregation metric used to structure the vector space during the traffic processing which determine how the group of packets are built (see 4.4.1), and second, the attributes built from the aggregated traffic.

In fact, looking at table 2, it is noticeable that every signature is using different attributes. This implies that for trying to find these new signatures from scratch, it is needed to search for new previously unused attributes. Therefore, the discovery of new types of anomaly seems to be heavily linked to the discovery of new pertinent attributes.

The method that we intend to use to find new anomaly signatures is to look for anomalous representations (i.e. clusters or outliers) in the representation of the network traffic inside new attributes. In order to do so, we intend to generate new attributes, systematically try to find anomaly representations to assess the presence of a new anomaly, and if it is the case, build the corresponding new signature.

The problem of creating new pertinent signatures can then be split into three tasks: first, process the network traffic, second, generate new attributes, and third, search for new anomalies inside combinations of generated attributes.

4.4.1 Traffic aggregation and processing

Aggregation of traffic is the first part of traffic data processing. It is an important function because it allows us to change the point of view on the network traffic by changing the aggregation criteria.

In fact, by aggregating the traffic according to the destination address of the network traffic with a certain network mask, one aggregate traffic destined to a restrained number of destinations hosts. One can then find anomalies that are impacting a small number of destination addresses (in some case targets of attacks) no matter how many sources are involved. This enables us to target anomalous traffic such as DDoS (Distributed Denial of Service) attacks.

Corollary, for searching anomalies having a few number of sources and paying attention to the number of destinations, i.e. network scan or SYN port scan for instance, aggregation through the source IP address is the aggregation criteria to use.

In the same way, one could imagine that it is only needed to target anomalies linked to the port number. One could then use the port number as the aggregation parameter in order to characterize the behavior of an application using a fixed port.

4.4.2 Attribute generation

- Create new attribute

  Table 2 actually consists of attributes that are built on the distribution of the values of fields of packet headers. Some attributes are even built from values obtained over two different packet fields. We plan to generalize this construction by using two steps. First, process values over the distributions of values of packet fields of the layers network and transport of the OSI model (IP address, TCP/UDP ports source/destination, flags, ...). The operators used on the distributions will be simple: number of different element, proportion of the biggest element over the total, ... Second, sweep all possible combinations of one or two elements of the previously generated values. Once the combination are obtained, we generate the attribute. If the combination contain one value, then the value is turned into an attribute. If the combination contain two values, then, we process the ratio of the first value over the second. At the end, we obtain a set of attributes built over the packet headers.

  However, it is obvious that such a variety of possible combinations applied to a big number of packet fields will generate a huge amount of possible combinations. The next issue will be to eliminate the attributes that seem to be of less interest.

- Attribute interest assessment

  We want to assess whether a generated attribute contains enough information. In fact, if we want to extract clusters/outliers from the data spaces, we will need attributes that have a quantity of information as significant as possible. If all the values of the parts of the traffic for the considered attribute are close a certain value, the search for network classes or outliers inside this restricted space will be very complex and unreliable. Therefore, a first elimination of the attribute with poor interest seems relevant. Entropy is the mathematical tool that will be used for the evaluation of the quantity of information contained in the considered attributes.

4.4.3 Anomaly search

After the attribute generation step, our algorithm have several attribute built on the packet fields. The next step is now to apply unsupervised learning in order to find new anomalies.

We intend to sweep all possible combinations of the previously generated attributes and search for anomalies inside each combination.

As previously said in 4.2.2, each combination of attributes can be used to build a vector space with a basis is constituted of the attributes selected. Therefore, we search for the presence of an anomaly inside the chosen vector space.

There are two steps in the search of anomaly: first, characterization of the traffic in the chosen vector space, and then application of unsupervised learning according to the result of the first substep. These two steps will be repeated on each vector space. This process will then be repeated for each vector space built on each combinations of attributes.

- Characterization of the traffic in the chosen vector space

  As we explained in the previous paragraph, in order to find a new anomaly, we try to find a pertinent representations of a 0day anomaly in a new vector space built from new attributes. This representations may be either a cluster or an outlier depending on the representation of the traffic considered.

  In this step, we determine what representation of network traffic is accurate in term of classes inside the set of chosen attributes in order to define which unsupervised learning techniques will be used for the anomaly detection step. In order to achieve this goal, we apply clustering to the traffic. The result of this step gives us the structure of the network traffic.

  At this point, two cases arise. First, we obtain one cluster. By doing the assumption that the normal traffic is much more important in volume that the anomalous one, we deduce that the normal traffic is composed of the only found cluster. Therefore, in this case, anomaly detection will be using outlier detection.

  Second, we obtain several clusters: this means that the traffic is composed of several classes. However, nothing guarantees that one of these clusters is not actually an anomaly. This step will then require human intervention to manually identify the clusters between genuine and anomalous network traffic. As far as our statistical study have advanced (cf. 5.3.1), this case seems to be rare. Therefore, human intervention seems not to be needed. However, our study being statistical, we cannot guarantee that this case is totally irrelevant.

- Search for anomalies

  According to the results of the characterization of the network traffic, we use the appropriated unsupervised learning technique to search for a 0day anomaly and its associated signature. Several situations are possible. If all clusters belong to legitimate traffic, outlier detection will be applied in order to search for anomalies. If there are some anomalous clusters, we will still apply outlier detection because there might be other anomalies (represented as outlier) than the ones in the clusters. At this point, we are able to identify the legitimate and the anomalous network traffic and know wether there is a new anomaly inside the considered vector space or not.

  If it is the case, the matter of building the new signature is simple. In fact, once the anomalousness of each cluster and the presence of outliers is assessed, a convex or concave hull is drawn at half-distance between the normal representation(s) of traffic and the anomalous one(s). To obtain the updated signatures in terms of thresholds on a specific attribute, the hull is projected onto each axis of the vector space. In order to improve the system, we also could keep the hull and use it as a unique multi-dimensional threshold.

4.5 Updating signature thresholds

Once the anomalies have been found and their signatures built, the system knows what are the attributes linked to each anomaly. However, the thresholds built during the 0day anomaly search may become obsolete along time. Therefore, the update of these thresholds is mandatory in order to keep an online anomaly detection system accurate. The signature updates will then simply consist of searching the anomaly inside the specific vector space (the one described by the attribute associated to the considered anomaly). The techniques used to update are the same than the one used to build the new rule: we draw a convex or concave hull at half-distance between between the normal representation(s) of traffic and the anomalous one(s).

5 Validation

In this section, we present our current work on evaluating the validity of our proposed method. The section is structured as follows: first, we present the data we used to do our experimentation. Second, we explain the different steps that we plan to go through to validate our method. Third, we expose our progress among these steps and the result of our first experimentations.

5.1 Data

A proper statistical validation of anomaly detection procedures requires the use of data with known, documented anomalies. Data might be collected from a real network and labeled afterwards by expert network operators. This would generate a dataset with known real anomalies (i.e. anomalies that happened on the wild), but might be prone to human errors (i.e. network operators might manually misclassify an anomaly), and does not permit control over the anomalies' characteristics (e.g. their intensity). Generating such datasets is expensive and currently very few are publicly available. The other way to generate labeled data is to artificially produce anomalies in real or simulated networks. With this approach, anomalies can be fully documented and are not subjected to misinterpretations. Characteristics of the anomalies can also be controlled (i.e. varying its intensity, duration, etc.) to permit evaluation under different settings. The drawback is that the anomalies might not be too representative of current occurrences. We use both types of datasets to validate our algorithm: the METROSEC project [MET] traces with artificially created anomalies and the MAWI traffic repository [MAW] with anomalies seen in the wild.

The first part of the traces used during our experiments comes from the MAWI dataset. It is composed of 15 minutes packets traces collected daily at 2PM from a Japanese network called WIDE since 1999 to present. These traces are provided publicly after being anonymized and stripped of their payload data. These traces are undocumented, but the authors of [DFB$^+$07] started an effort to label anomalies found in this database (see also http://www.manaworld.org/wide/anomalies/). We analyzed all traces from January 2001 to June 2006. Traces used are from samplepoint-B which represent trans-Pacific links between Japan and the United States. Traffic on these links are mostly exchanged between Japanese universities and commercial ISPs and consistently contain anomalies [BDF$^+$09].

The second part comes from the METROSEC project. These traces consist of real traffic collected on the French National Research and Education Network (RENATER) with simulated attacks performed using real DDoS attack tools. This dataset was created in the context of the METROSEC research project to, among other goals, study the nature of anomalies and their impact on networks' QoS. The dataset has been used for validation by a number of different studies on anomaly detection (e.g. [SLO$^+$07, ABD07, FOM07]). Traces were collected using DAG systems, from late 2004 to the end of 2006, and contain anomalies that range from very low intensity (i.e. less than 4% of normal traffic volume) to very high (i.e. more than 80%). One to four attacking sites were used (French research laboratories located in Mont-de-Marsan, Lyon, Nice and Paris) to create complex and realistic DDoS attacks toward LAAS in Toulouse. The traces are fully documented with start and ending time of capture and attack, intensity, type and number of bots (i.e. attacking sources) of the attacks.

5.2 Methodology

We want to demonstrate that our system is able to find a unknown anomaly inside network traffic. The unknown aspect is to be considered from the point of view of the detection system, it means that the system must not have any priori knowledge over the anomaly characteristics.

In order to validate our approach, we plan to use an incremental implementation and validation of our algorithm. It will allow us to validate each part of our algorithm separately. The validation will then be composed of two steps:

- First, we want to find a known anomaly (from our point of view) inside a trace where we know that the anomaly is present and build its signature. The only parameters given to the algorithm will be the attributes (and thus the vector space) that will be used to find the anomaly. This implies in our case, that we skip the steps of our algorithm related to attribute generation and attribute selection. Then, we apply our algorithm and search for an anomaly using only the attributes related to the targeted anomaly as specified in table 2. Therefore, by looking at the right attributes inside a documented tracefile which contains the anomaly that will fit these chosen attributes, we are supposed to find it. We also want to extend this work to several types of anomalies with their appropriates attributes.
- Second, we want to validate the global behavior of our algorithm. In order to do so, we use a documented trace file with a known anomaly inside. We do not proceed to any restriction over the used attributes and use instead the attribute generation and attribute selection steps of our algorithm. The validation of this step will be the finding of (at least) the documented anomaly and maybe other undocumented anomalies.

This paper covers our work toward the accomplishment of the first step.

5.3 Experimentations

We chose to focus ourselves on the detection of TCP SYN DDoS. Our goal is discover the anomaly without any prior knowledge other than the attributes used for its signature, in this case: #respdest, spprop, oneportpred and #rpkt/#rdstport (cf. table 2). In order to apply our method, we use the parameters that fit this type of anomaly. The aggregation parameter used is the destination since we want to target an anomaly with several source and only one destination. The attributes used to build the vector space are the one related to this type anomaly cited above, i.e.: #respdest, spprop, oneportpred and #rpkt/#rdstport.

The next part explains the results of our investigations about the structure of the traffic in this restricted vector space, and then, the result of the anomaly search.

5.3.1 Characterization of network traffic

We proceed to the analysis of the characteristics of the traffic on the TCP SYN Flood DDoS attributes. In order to characterize the traffic inside the vector space, we study several traces from the datasets cited in 5.1. We use 64 traces from the MAWI dataset, they are dated from January 2001 to June 2006. We also use one trace from Metrosec which had not any provoked anomaly inside. This trace has been captured on the 23th october of 2006. It lasts two hours and half.

We observe manually the data in the vector space considered (i.e. the one built on the four attributes quoted above). In order to do so, we generated two images in 3D to be able to cover the attributes of the vector space. We generate these two images for all the traces. In the first image, the attributes used were spprop, oneportpred and #rpkt/#rdstport. In the second image, we used #respdest, spprop and oneportpred. Then, we analyze them by hand.

It appears that for all images of the first type, the network traffic is composed of only one cluster. A good example is provided by figures 4 (a) and (b). The images of the second type are generally composed of only one class. The data often presents more noise than for the first attributes (cf figure 5 (a)). Some clusters arise, as in figure 5 (b), but as far as we now, they are directly related to scan events (network scan or port scan or both at the same time).

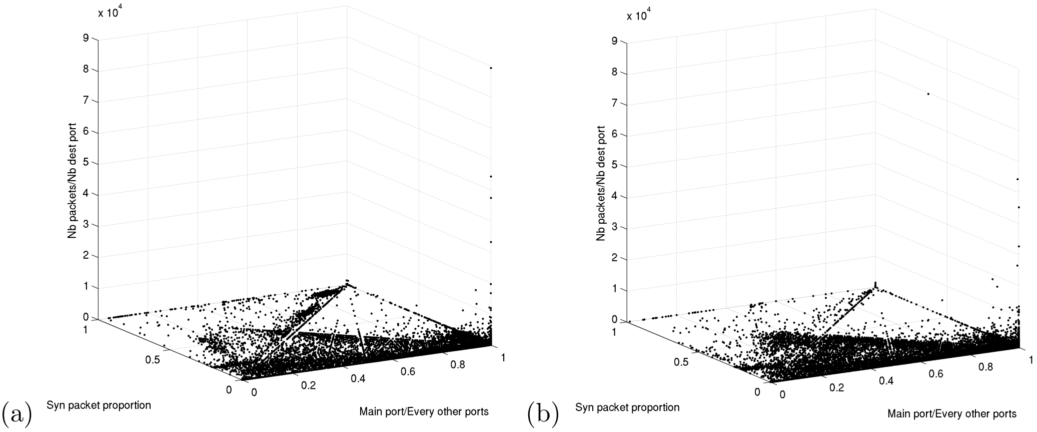


Fig. 4: Traffic representationf for *spprop*, *oneportpred* and *#rpkt/#rdstport*

However, considering the whole set of generated images, we can consider that the traffic is statistically composed of one class in the considered vector space.

5.3.2 0day anomaly search

As it was seen in the previous subsection, the traffic is generally composed of only one main class. Therefore, the technique used to find a 0day anomaly in the vector space is of the outlier detection kind. We use a documented trace from the Metrosec project where a TCP SYN Flood DDoS has been produced and captured. This trace is dated from the 9th of december 2004. The capture started at 14:15:33 and finished at 17:18:07. The coordinated attack came from three machines. All three attackers were using TFN2K

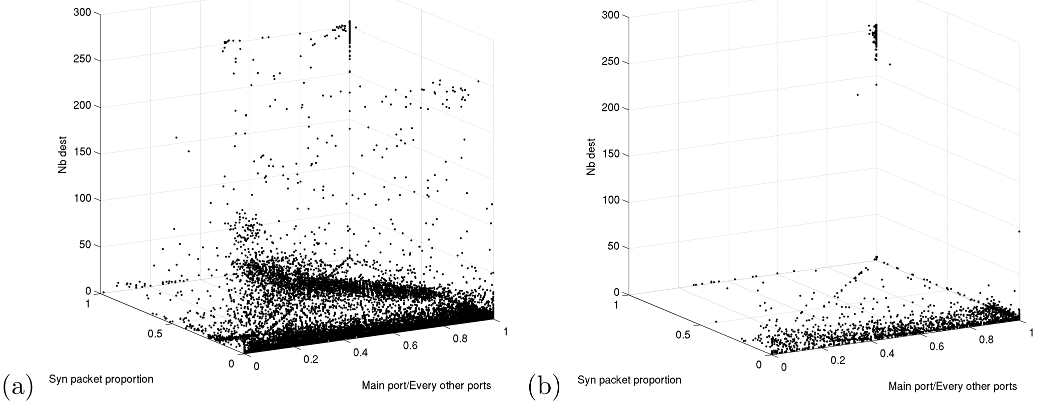


Fig. 5: Traffic representation for *spprop*, *oneportpred* and *#respdest*

[BT00]. The first attack starts 6230 seconds after the beginning of the trace and ends 6390 seconds after the beginning of the trace. We extracted a segment of the original trace situated in the middle of the attack, from seconds 6250 to 6255. We voluntary used a slice of the original trace located in the middle of the attack in order to reproduce the actual behavior of an online system that would operate on a finite windows of time. We then apply a very simple outlier detection algorithm.

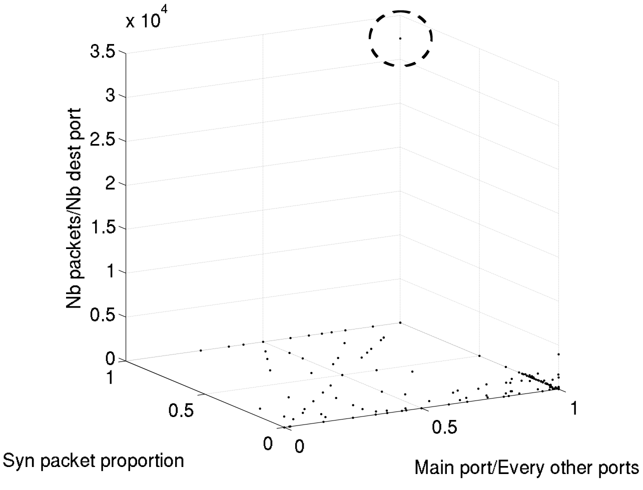


Fig. 6: Network traffic with a TCP SYN DDoS occuring

Our anomaly detection system is able to detect the outlier corresponding to the attack. This outlier corresponds to the first attack. The figure 6 show the data space representation of the traffic inside three of the four attributes used for the outlier detection. It clearly appears that the point that represents the anomaly is on the top corner of the figure, while the normal traffic appears on the horizontal bottom plan. The generated signature will then be the one on equation 1.

$$\#rpkt/\#rdstport > 15000 \tag{1}$$

6 Conclusion

We propose a complete a method to detect 0day network traffic anomalies and corresponding signatures through the use of machine learning. This method use an automatic generation of attributes in order to generate semantically interesting attributes. Machine learning is then applied to several combinations of these attributes. At the end, our algorithm is able to find a still unknown anomaly which could have been a 0day attack and generate its signature automatically eg. a TCP SYN DDoS as shown in 4.4.3.

6.1 Future work

By looking at the genrated signature of equation 1 and compare it to the one present in table 2, it is obvious that our algorithm is still not able to capture the whole semantic of the considered anomaly. This problem have to be addressed in a future work. Automatization of the process of the characterization of the traffic has also to be coped with in order to avoid the manual inspection of the traffic as in 4.4.3. Once these problem solved, our future work will then be to implement the step 2 and 3 of the methodology presented in 5.2.

References

[ABD07] P. Abry, P. Borgnat, and G. Dewaele. Invited talk: Sketch based anomaly detection, identification and performance evaluation. *International Symposium on Applications and the Internet Workshops (SAINT)*, January 2007.

[BDF+09] P. Borgnat, G. Dewaele, K. Fukuda, P. Abry, and K. Cho. Seven years and one day: Sketching the evolution of internet traffic. In *INFOCOM 2009, IEEE*, pages 711–719, April 2009.

[BKPR02] Paul Barford, Jeffery Kline, David Plonka, and Amos Ron. A signal analysis of network traffic anomalies. In *IMW '02: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurement*, pages 71–82, New York, NY, USA, 2002. ACM.

[Bru00] Jake D. Brutlag. Aberrant behavior detection in time series for network monitoring. In *LISA '00: Proceedings of the 14th USENIX conference on System administration*, pages 139–146, Berkeley, CA, USA, 2000. USENIX Association.

[BT00] Jason Barlow and Woody Thrower. Tfn2k - an analysis, February 2000. `http://packetstormsecurity.org/distributed/TFN2k%5FAnalysis-1.3.txt`.

[CC94] T. F. Cox and M. A. A. Cox. *Multidimensional scaling.* Chapman Hall, 1994.

[CSKC08] P. Chhabra, C. Scott, E.D. Kolaczyk, and M. Crovella. Distributed spatial anomaly detection. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 1705–1713, April 2008.

[DFB$^+$07] Guillaume Dewaele, Kensuke Fukuda, Pierre Borgnat, Patrice Abry, and Kenjiro Cho. Extracting hidden anomalies using sketch and non gaussian multiresolution statistical detection procedures. In *LSAD '07: Proceedings of the 2007 workshop on Large scale attack defense*, pages 145–152, New York, NY, USA, 2007. ACM.

[EAP$^+$02] Eleazar Eskin, Andrew Arnold, Michael Prerau, Leonid Portnoy, and Sal Stolfo. A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. In *Applications of Data Mining in Computer Security*. Kluwer, 2002.

[FOM07] S. Farraposo, P. Owezarski, and E. Monteiro. A multi-scale tomographic algorithm for detecting and classifying traffic anomalies. In *Communications, 2007. ICC '07. IEEE International Conference on*, pages 363–370, June 2007.

[I.T02] Jolliffe I.T. *Principal Component Analysis*. Springer, 2002.

[KDD99] KDD99. Kdd99 cup dataset, 1999. `http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`.

[KSZC03] Balachander Krishnamurthy, Subhabrata Sen, Yin Zhang, and Yan Chen. Sketch-based change detection: methods, evaluation, and applications. In *IMC '03: Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, pages 234–247, New York, NY, USA, 2003. ACM.

[KZ08] Liwei Kuang and Mohammad Zulkernine. An anomaly intrusion detection method using the csi-knn algorithm. In *SAC '08: Proceedings of the 2008 ACM symposium on Applied computing*, pages 921–926, New York, NY, USA, 2008. ACM.

[LBC$^+$06] Xin Li, Fang Bian, Mark Crovella, Christophe Diot, Ramesh Govindan, Gianluca Iannaccone, and Anukool Lakhina. Detection and identification of network anomalies using sketch subspaces. In *IMC '06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pages 147–152, New York, NY, USA, 2006. ACM.

[LCD04a] Anukool Lakhina, Mark Crovella, and Christiphe Diot. Characterization of network-wide anomalies in traffic flows. In *IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 201–206, New York, NY, USA, 2004. ACM.

[LCD04b] Anukool Lakhina, Mark Crovella, and Christophe Diot. Diagnosing network-wide traffic anomalies. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 219–230, New York, NY, USA, 2004. ACM.

[LCD05] Anukool Lakhina, Mark Crovella, and Christophe Diot. Mining anomalies using traffic feature distributions. *SIGCOMM Comput. Commun. Rev.*, 35(4):217–228, 2005.

[Mac03] David MacKay. *Information Theory, Inference and Learning Algorithms.* Cambridge University Press, 2003.

[MAW] MAWI. Mawi dataset. At `http://mawi.wide.ad.jp/`.

[MET] METROSEC. Metrosec dataset. At `http://www.laas.fr/METROSEC`.

[OF09] Philippe Owezarski and Guilherme Fernandes. Classification automatique d'anomalies du trafic. In *SARSSI'2009: Conférence sur la sécurité des architectures réseaux et des systèmes d'information*, page 15 p., Luchon France, 06 2009.

[Por01] Leonid Portnoy. Intrusion detection with unlabeled data using clustering, 2001.

[SLO$^+$07] Antoine Scherrer, Nicolas Larrieu, Philippe Owezarski, Pierre Borgnat, and Patrice Abry. Non-gaussian and long memory statistical characterizations for internet traffic with anomalies. *IEEE Trans. Dependable Secur. Comput.*, 4(1):56–70, 2007.

[Vii06] Jouni Viinikka. *Traitement de flux d'alertes en détection d'intrusion avec des méthodes d'analyse de séries temporelles.* PhD thesis, Université de Caen, November 2006.

[War63] Jr. Ward, Joe H. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.