Computers and Electrical Engineering 38 (2012) 1062–1072

Contents lists available at SciVerse ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Automatic network intrusion detection: Current techniques and open issues ☆

Carlos A. Catania [a,*], Carlos García Garino [a,b]

[a] ITIC, Universidad Nacional de Cuyo, Mendoza, Argentina
[b] Facultad de Ingeniería, Universidad Nacional de Cuyo, Mendoza, Argentina

ARTICLE INFO

Article history:
Available online 14 June 2012

ABSTRACT

Automatic network intrusion detection has been an important research topic for the last 20 years. In that time, approaches based on signatures describing intrusive behavior have become the de-facto industry standard. Alternatively, other novel techniques have been used for improving automation of the intrusion detection process. In this regard, statistical methods, machine learning and data mining techniques have been proposed arguing higher automation capabilities than signature-based approaches. However, the majority of these novel techniques have never been deployed on real-life scenarios. The fact is that signature-based still is the most widely used strategy for automatic intrusion detection. In the present article we survey the most relevant works in the field of automatic network intrusion detection. In contrast to previous surveys, our analysis considers several features required for truly deploying each one of the reviewed approaches. This wider perspective can help us to identify the possible causes behind the lack of acceptance of novel techniques by network security experts.

1. Introduction

A network intrusion detection system (NIDS) is the software tool that automates the network intrusion detection process. From an architectural point of view a NIDS can be analyzed from several angles (i.e. traffic capture process, system location, appropriate measures selection, among others). However, from a more simplified point of view, intrusion detection can be seen just as a classification problem in which a given network traffic event is assigned as *normal* or *intrusive*.

In the past 20 years, several techniques have been proposed to address the embedded classification problem inside NIDS. Perhaps the most successful approach has been the one based on pattern signatures describing known attacks behavior [1]. Under this approach, a malicious event is detected when some monitored event matches against a signature pattern. Despite signature-based NIDS are considered the *de facto* standard, they face the problem of needing a new set of signature patterns each time a new attack emerges. In addition, signatures describing such attacks have to be written by experts, which are not always available. In other words, the signature-based approach has failed in providing the level of automation required by security staff members.

Alternatively, techniques including statistical methods, machine learning and data mining methods have been proposed as a way of dealing with some of the issues regarding signature based-approaches. Such techniques aim at facilitating the work of the network security staff, providing a higher automation in the intrusion detection process along with good detection capabilities. Despite the success in obtaining high accuracy levels, most of these techniques have actually not been

☆ Reviews processed and proposed for publication to Editor-in-Chief by Guest Editor Dr. Gregorio Martinez.

* Corresponding author. Tel.: +54 02614291000.
E-mail addresses: ccatania@itu.uncu.edu.ar (C.A. Catania), cgarcia@itu.uncu.edu.ar (C.G. Garino).

0045-7906/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compeleceng.2012.05.013

deployed in real-life scenarios. This situation suggests that accuracy is not the only goal in the pursuit of automatic intrusion detection.

The present work reviews the most relevant network intrusion detection techniques for wired networks, putting special emphasis on the embedded classification problem. However, in opposition to previous surveys on this field, analysis is performed considering not only accuracy results but also other features required for implementing the discussed techniques in real-life scenarios.

The rest of this work is organized as follows: Section 2 provides background information about the intrusion detection problem, including attack definitions, a taxonomy and a simplified NIDS architecture. Then, in Section 3, the most relevant approaches applied to intrusion detection are reviewed and compared based on the taxonomy along with common measures related to NIDS. Section 4 remarks the remaining open issues, which aim to explain why all except the signature-based approach are not being deployed on current networks. Finally, concluding remarks are provided in Section 5.

2. Background

Before discussing the most relevant approaches to NIDS, we proceed to describe the fundamental elements inside the intrusion detection problem.

2.1. Attack definition and classification

A computer *attack* can be defined as the intelligence of evading or evading attempt of computer security policies, acceptable use policies, or standard security practices. In the security research community, the terms *attack* and *intrusion* are often used with the same meaning.

In the past years, there have been several attempts to build taxonomies aimed at classifying attacks. One of the most accepted taxonomy is the one proposed by Kendall [2], in which attacks can be classified into four categories:

Probing: Attacks oriented to gather information about the system, for further intrusion. These attacks include network traffic sniffing and port/address scanning.
Denial of Service (DoS): Attacks attempting to diminish or totally interrupt the use of a system or a service to their legitimate users.
User to Root (U2R): Attacks that aim to gain superuser access to the system by means of exploiting vulnerabilities in operating systems or software applications. The attacker has a valid account in the system.
Remote to Local (R2L): Attacks oriented to gain local access from outside the network.

A broad attack taxonomy is presented by Lazarevic et al. in [3], in which a new category is added for programs that replicate on host machines or propagate through the network. This new category includes programs such as *viruses*, *worms* and *trojan horses*.

2.2. A simple NIDS architecture

In general, from an architectonic point of view, a NIDS is based on the following modules:

Traffic Data Acquisition: This module is used in the data collection phase. In the case of a NIDS, the source of the data are raw network frames or information from upper protocol layers (i.e. IP or UDP protocols).
Traffic Features Generator: This module is responsible for extracting a set of selected traffic features from captured traffic. Network traffic features can be classified in *low-level* features and *high-level* features. A *low-level* feature can be directly extracted from captured traffic (e.g. IP header). Whereas a *high-level* feature consists of traffic information deduced from captured traffic by a subsequent process. Features can be also classified according to the network traffic source used for generating them. *Packet* features are those directly obtained from network raw packets headers. *Flow* refers to features containing aggregated information related to network connections. Finally, *Payload* stands for those features obtained from packet payload.
Incident Detector: This module processes the data generated by the *Traffic Features Generator* module to identify intrusive activities. Traditionally, network intrusion detection methodologies have been classified into two broad categories [4]: *misuse detection* (matches the input data against a definition of an attack) and *anomaly detection*(based on a definition of normal behavior of the target system). No matter the detection methodology implemented by the *Incident Detector*, once a malicious event has been detected, an alert will be raised and sent to the *Response Management* module.
Traffic Model Generator: This module contains the reference data used by the *Incident Detector* to compare with. The source of information of the *Traffic Model Generator* could come from human knowledge or from some automatic knowledge acquisition procedures.
Response Management: Once an alert is received, this module has the responsibility to initiate actions in response of a possible intrusion.

2.3. A taxonomy for NIDS

Researchers have proposed several taxonomies for NIDS. Here, we summarize the elements of a taxonomy commonly accepted in the intrusion detection research community [3].

Detection method: The two broad methods for detection are considered: *anomaly-based* and *misuse-based*.
Model acquisition: Traffic model is based on human knowledge or generated by some automatic generation process.
Usage frequency: Detection can be performed in real-time (continuous monitoring) or by batch (periodic analysis).
Architecture: Data collection and processing can be done only from a single monitored point of the network (centralized) or from multiple points of the network (distributed).

Finally, we summarize the four more relevant measures when considering true deployment feasibility:

Prediction accuracy: Measures how good is a NIDS in detecting intrusion.
Processing time: Considers the rate at which events are processed. A NIDS should be able to perform detection as soon as possible.
Adaptability: Indicates the NIDS capability to deal with new attacks techniques. A NIDS should be able to re-adapt itself in the presence of new threats.
Resource consumption: Measures how much memory and storage resources are required by the system.

3. Intrusion detection approaches

Because of the large number of works presented during the past years for both misuse and anomaly detection, it is convenient to group them according to the techniques used by each one of them. In this sense, we rely on the categorization proposed by Patcha and Park [5] and Lazarevic et al. [3].

3.1. Misuse based

The idea behind misuse detection consists of comparing network traffic against a *Model* describing known intrusion events. This approach has proved to be very effective at detecting known threats but largely ineffective at detecting unknown threats. Among the most commonly used techniques for misuse detection we can find pattern recognition, implication rules and data mining techniques.

3.1.1. Pattern recognition methods

Nowadays, the most successful strategy for intrusion detection is the so called *signature-based*. In this approach each malicious event is described by a pattern signature using its own descriptive language. A malicious event is detected when some monitored event matches against this signature pattern. SNORT is a lightweight NIDS originally developed by Roesch in 1999 [1], which has become the de facto standard for intrusion detection. It is a complete and fully functional NIDS that follows a signature-based approach.

3.1.2. Implication rules-based methods

Basically, in this approach the *Incident Detector* module of the NIDS is implemented by a rule-based expert system. In this regard, we can mention P-BEST [6], a rule-based forward chaining expert system located inside the EMERALD IDS [7]. The idea behind this approach consists of giving a set of rules describing network events. A series of these asserted network events may lead the system to infer that a previously defined intrusion event has occurred.

3.1.3. Data mining techniques

The idea behind data mining techniques aim to eliminate the need of manually created *Traffic Models* by automatically building them from some reference data [8]. Major benefits provided by these algorithms consist of being able to detect not only known attacks but also their variations.

The use of Artificial Neural Networks (ANNs) for misuse detection has been one of the most analyzed data mining approaches for NIDS. Since the first work of Cannady [9], many other authors have proposed different ANN based approaches. On this subject, we can cite the work of Liu et al. [10], in which they applied Principal Component Analysis (PCA) combined with ANN for automatic feature extraction. More recently, Kumar and Selvakumar [11] have applied ANN using the resilient back propagation algorithm for detecting distributed DoS attacks. A summary of most of the research done on ANN applied to the NIDS field in the last 20 years can be found in Ahmad et al. [12]

Evolutionary computation (EC) is another data mining approach that has played an important role in the NIDS research field due to its adaptive characteristics.

In particular, Genetic Algorithms (GAs) have been applied in several tasks regarding NIDS. Such tasks include the automatic design of models structure [13] and the generation of function transformations [14]. However, GAs have been mostly used for building rule-based models for classification [15–18]. Instead of manually writing the rule set, as needed by traditional rule-based system, the idea of using GAs for classification is to automatically obtain such classification rules. The main advantage of rules is that they are simple and easy to understand.

As mentioned in the work of Gomez and Dasgupta [19], the normal and the abnormal behaviors in networked computers are hard to predict because the boundaries cannot be well defined. This has lead some authors [14,20–22] to suggest that the quality of the rules obtained by EC can be improved by means of combining EC approaches with fuzzy logic.

Beyond the use of ANNs, GAs and fuzzy logic, many other data mining approaches have been applied to the intrusion detection field. Some authors [23,24] preferred using fuzzy association rules, avoiding the large building time requirements of the fuzzy EC approaches. Ye et al. [25] proposed an approach for signature recognition based on classification trees. Mukkamala et al. [26] analyzed the behavior of Support Vector Machines (SVMs) for recognizing intrusion patterns [26,27]. Swarm intelligence [28] is another technique that has been deeply analyzed regarding the network intrusion detection problem. In this regard, Ant Colony Optimization (ACO) has been analyzed in [29–31]. On the other hand, Particle Swarm Optimization (PSO) techniques have been applied in [32].

To conclude, special attention should be given to multiple classifiers approaches [33]. From a architectonic point of view, these approaches have an *Incident Detection* module with distributed classification capabilities as mentioned in Section 2.3. The claimed benefits of these multiple classifiers approaches are that by combining the opinions of an ensemble of classifiers, the new opinion would be better than the individual ones. Recently, the intrusion detection community has started to consider the viability of this compound approach. In this regard, perhaps one of the most relevant study on this topic is the work of Giacinto et al. [34]. Here the authors analyze the behavior of a multiple classifier approach using multiple ANNs *Traffic Models* generated from a different number of *Traffic Features*. In addition, five fusion strategies are tested for combining the output of the different models.

3.2. Anomaly based

The anomaly detection method is based on the analysis of the profiles that represent normal traffic behavior. First, an anomaly detector creates a baseline profile of the normal *legitimate* traffic activity. Thereafter, any new activity that deviates from the normal model is considered an anomaly. This methodology has the major benefit of potentially recognizing unforeseen attacks. However, its major drawback is a potentially high false alarm rate. Among the most commonly used techniques for anomaly detection we can find statistical methods, machine learning and data mining techniques.

3.2.1. Statistical methods

The idea behind statistical methods consists of maintaining two profiles during the anomaly detection process: the currently observed profile and the previously stored statistical profile. As a new network event is observed the current profile is updated and compared with the stored profile. These profiles are based on measures of certain variables over the time.

The EMERALD IDS [7] is one of earliest NIDS based on statistical anomaly detection. The statistical module inside EMERALD is focused on providing real-time surveillance of TCP/IP-based networks for malicious or exceptional network traffic.

SPICE (Stealthy Port scan and Intrusion Correlation Engine) [35] is another statistical-based approach focused on detecting stealthy scans in real-time. The architecture of SPICE consists of an anomaly sensor and a correlator. The sensor monitors the network and assigns an anomaly score to each event. The anomaly score for a packet is based on a frequency-based mechanism. The fewer times a packet is observed the higher its anomaly score will be.

Other interesting statistical approaches have been proposed as a way to deal with the non-stationary property of network traffic. Lakhina et al. [36] focused on backbone network traffic characterization by means of an exploratory PCA. Whereas Gu et al. [37] use some information-theoretic measures as a way to distinguish anomalies that change the traffic either abruptly or slowly.

3.2.2. Machine learning techniques

The use of Machine Learning (ML) techniques for anomaly detection are focused on building a model that improves and adapts its performance based on previous results.

One of the most remarkable efforts in the study of anomaly detection is the work of Mahoney and Chan [38]. They have proposed several anomaly detection models based on ML techniques [38–40].

One of the first approaches proposed is the PHAD system (Packet Header Anomaly Detector) [39]. PHAD performs anomaly detection using previous information from packet headers. PHAD *Traffic Features Generator* module considers 33 low-level features based on fields from the Ethernet, IP and transport layers. The anomaly score for each feature is calculated only considering the recent novel events and discarding the rest.

ALAD (Application Layer Anomaly Detection) is another approach proposed by Mahoney and Chan [38], which instead of analyzing single packets, it considers incoming server TCP connections. The *Features Generation* module considers low-level features from TCP connections, as well as information from payload. LERAD (LEarning Rules for Anomaly Detection) [41] is similar to the ALAD approach, but instead of using a fixed set of probabilistic rules, LERAD learns these rules using previously acquired network traffic data.

3.2.3. Data mining techniques

Data mining techniques have also been used for anomaly detection. Lee and Stolfo [8] propose the use of inductive rule generation algorithms. These algorithms combine the application of association rules with frequent episode patterns to classify network traffic. The *Feature Generation* module described in [8] considers low-level features based on packet information as well as high-level connection features. Then, using the RIPPER [42] algorithm for rules induction, the model is generated from attack-free network traffic data. Finally, during the *Incident Detection* stage, any new event not matching against any of the learned rules is considered an anomaly.

Alternatively, to avoid the need of attack-free data, some authors have proposed the use of unsupervised learning techniques. For instance, Portnoy et al. [43] propose an unsupervised approach that uses clustering techniques applied to the intrusion detection problem. Clustering techniques consist of grouping data into clusters according to some distance or similarity measure. The *Features Extraction* module uses features similar to the ones used by many of the previously discussed works. The *Traffic Model* generation follows a single-linkage [44] clustering approach. During the construction model stage, traffic events are labeled following the assumption that the number of normal events exceeds the number of intrusions. Then, during the *Incident Detection* stage, any new traffic event is classified according to the label of its closest cluster. Other authors [45–47] have followed the ideas of Portnoy et al. but using different strategies for computing cluster membership.

SVM is another technique applied to unsupervised anomaly detection. In [48,49] different authors have proposed the use of the SVM variant for anomaly detection. SVM techniques for anomaly detection are well known for their capability for handling data with not only normal traffic but also anomalies.

Finally, it is worth mentioning that multiple classifiers approaches have also been applied for anomaly detection. In the work of Giacinto et al. [50], the authors proposed an unsupervised Multiple Classifier System (MCS). Each unsupervised classifier is used for modeling a particular group of similar protocols or network services. The use of a modular MCS allows the security staff to choose a different traffic model and decision threshold for different groups of network services. The work of Rehak et al. [51] is another relevant approach that applies multiple classifiers. In this case five well-known anomaly detection algorithms are combined by means of a *trust modeling framework* [52] used to assign proper trustworthiness and reputation to each traffic model.

3.3. Discussion

Tables 1 and 2 summarize the main characteristics of the intrusion detection approaches described in the previous sections.

Proposed intrusion detection approaches are arranged by row. Each row describes the most representative approach of its kind. The first three columns refer to some of the NIDS taxonomy categories along with the measures discussed in Section 2.3. The first column symbolizes the techniques used for generating the model in each work. The second column denotes whether the proposed approach is capable of dealing with traffic detection in real-time. The third column refers to the potential adaptability to new intrusions observed by each approach. Notice that adaptability is measured considering the less possible human interaction into the model acquisition processes as well as the periodic model adjustment caused by the occurrence of new threats. The fourth column alludes to the kind of features used by each approach. Features are cataloged according the classification previously mentioned in the *Traffic Features Generator* module discussed in Section 2.2. The fifth column refers to the kind of attacks potentially detected by each approach. The recognized attacks are mentioned according to the attack taxonomy presented in Section 2.1. Finally, in the last column, other works following the same approach are referenced.

3.3.1. Misuse-based approaches analysis

Table 1 describes approaches following the misuse detection method. Misuse has been the most successful approach for intrusion detection so far. There have been some fully implemented NIDS available in the market for more than 10 years and their benefits have been proved beyond doubt.

Table 1
Comparative analysis of misuse-based intrusion detection approaches.

Authors	Model generation technique	Usage frequency	Model acquisition/adaptability	Features	Attack type	Related works
Lindqvist and Porras [6]	Implication rules	Real[a]	*Human*: rules written by experts	*Low and high-level*: packet, payload, flow	DoS, R2L	[7]
Roesch [1]	Pattern signature	Real[b]	*Human*: signature patterns written by experts	*Low-level*: packet, payload	R2L, U2R	[53,54]
Cannady [9]	*Data mining*: ANNs	Batch	*Automatic*: retraining with new attack patterns	*Low-level*: packet	DoS	[26,10,11]
Li [15]	*Data mining*: GA	Batch	*Automatic*: retraining with new attack patterns	*Low and high-level*: packet, flow	DoS, Probe	[16–18]
Gomez et al. [55]	*Data mining*: fuzzy/GA	Batch	*Automatic*: retraining with new attack patterns	*Low and high-level*: packet, flow, payload	DoS, Probe, U2R, R2L,	[14,20–22]

[a] Real time performance only with a reduced set of rules.
[b] Real time performance under some bandwidth scenarios.

Table 2
Comparative analysis of anomaly-based Intrusion detection approaches.

Authors	Model generation technique	Usage frequency	Model acquisition/adaptability	Features	Attack type	Related works
Porras and Valdes [56]	*Statistical*: chi-square-like	Real[a]	*Automatic*: readjust model with new attack-free patterns	*Low and high-level*: packet, flow, payload	DoS, Probe, R2L, U2R	
Staniford et al. [35]	*Statistical*: Naive Bayes networks	Real	*Automatic*: readjust model with new attack-free patterns	*Low-level*: packet, flow	Probe	
Lakhina et al. [36]	*Statistical*: PCA	Real[a]	*Automatic*: readjust model with new attack-free patterns	*Low-level*: flow	DoS, Probe	[59–61]
Mahoney and Chan [58]	*Machine learning*: rules learning and Markov models	Batch	*Automatic*: retraining with new attack-free patterns	*Low and high-level*: packet, flow, payload	DoS, Probe, R2L, Worm	[39,38,40]
Lee and Stolfo [8]	*Data mining*: rules learning and frequent pattern count	Batch	*Automatic*: retraining with new attack-free patterns	*Low and high-level*: packet, flow	Non-specified	
Portnoy et al. [43]	*Data mining*: unsupervised Clustering	Batch	*Automatic*: retraining with patterns containing a reduced amount of attacks	*Low and high-level*: packet, flow,payload	Probe, R2L, DoS	[46,47]
Eskin et al. [49]	*Data mining*: unsupervised SVM	Batch	*Automatic*: retraining with patterns containing a reduced amount of attacks	*Low and high-level*: packet, flow payload	Probe, R2L, DoS	[48]

[a] The use of flow information requires performing some sort of batch detection.

At first (probably owing to the tendency of the 1990s), most of misuse-based NIDS were basically expert-systems like P-BEST [6]. Despite their initial success, as the number of different attacks grows, it quickly became clear that these approaches were not suitable to perform real-time detection. Since the reasoning algorithms inside expert-systems are extremely computer-intensive, as the rule set gets bigger the system becomes slower and eventually turns unusable.

Alternatively, SNORT [1] proposes the use of signatures for *misuse* detection. SNORT performs packet inspection using fast pattern-matching algorithms. In some way, this signature-based approach emerges to solve the lack of detection speed for processing network events that rules-based system had shown. SNORT is capable of full real time detection when deployed in network up to 1Gbps. However, it is worth noting that current network throughput has overcome initial signature-based capabilities. SNORT [1] and even Bro [53], have exhibited high resource consumption when confronted with the overwhelming amount of data found in today's high-speed networks.

Both SNORT and P-BEST deeply rely on human activity for their traffic model acquisition process. Every time new attacks emerge, a set of rules (or signature patterns) describing such attacks have to be written by experts. This need of human experts results in poor adaptability to deal with new threats. First, when a new kind of attack emerges, the answer time of human experts could be not fast enough to adapt the system to this new threat. Second, it is practically impossible for human experts to be aware of all possible emerging attacks. This could lead to situations where those systems will simply be unable to detect new intrusions.

Misuse data mining techniques [8,9,15,19] arise with the aim of providing a way to reduce the need of human experts for building and adjusting the model. These systems are able to adapt their models to new security threats, if they have access to the network traffic containing such event. Using previously seen network traffic helps to build more accurate models with potentially more capacity for adapting themselves to new kinds of attacks. However, it should be noted that all misuse data mining approaches are unable to detect any attack type not previously seen.

There are also some other major issues to be considered when using misuse data mining techniques. First, data used for building traffic model ought to be large enough for being representative of network traffic. Certainly, since network traffic follows a non-stationary behavior, obtaining such representative sample cannot always be guaranteed. Second, for meeting building model requirements, network traffic data have to be previously labeled as *normal* or *intrusive*. This labeling process of thousands of traffic records is a very time consuming task that demands a lot of expert knowledge. Finally, since the building model processes of data mining algorithms are highly computer intensive, most of them are certainly not capable of performing real-time detection.

3.3.2. Anomaly-based approaches analysis

Table 2 describes those approaches based on anomaly detection. The interest of NIDS relying on anomaly-based *Incident Detection* modules has increased considerably in the recent years, mostly because their ability to detect forms of intrusions never seen before.

Statistical-based approaches [35,36,56] have the benefits of not requiring prior knowledge of attacks for their model generation process. Most of the statistical-based approaches have proved to be suitable for real-time detection. In this regard, approaches proposed by Porras and Valdes [56] and Staniford et al. [35] have been tested and have shown an acceptable performance in real traffic situations. The work of Lakhina et al. [36] has not been fully implemented. However, in their work, the authors provides experiment details that seem to prove the validity of the approach for real-time detection.

Like in all the anomaly detection approaches, in the case of statistical-based, adaptability does not mean to adjust the model for detecting new kind of intrusions. Instead adaptability refers to the required adjustments in the *normal* traffic model each time the network traffic behavior changes. Such adjustments are done using attack-free network traffic data.

Statistical-based approaches assume that network traffic responds to a quasi-stationary process. A situation that is not always realistic [57]. This incorrect assumption is perhaps the major drawback regarding the adaptability of statistical-based approaches and one of the causes behind the high false alarm rate reached by these methods.

On the other hand, the use of ML and data mining techniques in the intrusion detection field has been beneficial due to their potential adaptability to changes as new information is acquired. For instance, the approach proposed by Mahoney and Chan [58] has the major advantage that no distribution assumption is made during the traffic model generation, a situation that facilitates the automatic adaptation process.

Despite their potential improvements, data mining (and also ML) approaches for anomaly detection share some of the issues of misuse-based data mining techniques. First, since they have a computational intensive model generation process, these techniques have been considered mostly for batch detection. Second, they need a large amount of network traffic data labeled as normal for the model generation process. However, there is a subtle difference regarding labeled traffic requirements. Misuse-based data mining techniques assume the availability of fully labeled data, whereas anomaly-based just require attack-free data. This last approach is usually referred as *supervised anomaly detection* [44]. At first glance this could look as a improvement over misuse approaches. In practice, however, it is difficult to obtain attack-free data to implement these approaches. Verifying that no attacks are present in the training data can be an extremely demanding task, and for large samples this is simply infeasible. On the other hand, if the data containing attacks is treated as clean, intrusions similar to the ones present in the training data will be accepted as normal patterns, resulting in a increment in the number of misdetections.

Unsupervised anomaly detection approaches [43,46,45,49] arise as a way to deal with the *supervised anomaly detection* limitations regarding labeled traffic requirements. Portnoy et al. [43] propose the use of unsupervised clustering techniques that seem to be efficient in terms of frequency usage as long as the number of features used for *Model* generation remains low. On the other hand, Eskin et al. [49] suggest the use of SVM for anomaly detection. SVM are well prepared for dealing with high dimensional data at the expense of a higher computational cost that might be not suitable for real-time detection.

As mentioned by Portnoy et al. [43], unsupervised anomaly detection approaches are suitable to deal with the intrusion detection problem as long as the number of attacks remains below the 1.5%. However, in practice this assumption is not always true. There are some situations in which for specific periods of time, the presence of intrusions could exceed the number of normal traffic records. For instance, when a new vulnerability is discovered and it has been widely announced, it is possible to find attacks exploiting this vulnerability encompassing a extremely high percentage of the network traffic.

3.3.3. Feature selection analysis

If we consider the information in the fourth and fifth columns of Tables 1 and 2, it is clear that there is a relation between the type of feature used by each approach and the kind of attacks detected. For instance, performing *R2L* detection without considering *payload* information such as a failed login attempt or anomalous HTTP requests (just to mention a few) it is extremely difficult. The previous relation explains why those approaches relying on *payload* information [1,6,43,49,55,56,58] are capable of detecting *R2L* and in some cases even *U2R* attacks [1,55,56]. On the other hand, a common approach for detecting *Probes, Worms and DoS*attacks consists of using information regarding network *flows*. For example, recognizing a horizontal scan attack requires observing the number of connections from a host to different destinations in a predefined time window. In this regard, most of the discussed approaches capable of detecting *Probes* and *DoS* attacks except in [9,35] use flows information for their *Traffic Features Generation* module.

Finally, another issue that should be considered is the relation between the selected features and the computational resources required for extracting such features. For example, obtaining high-level features from *flows* networks (such as number of connection from the same host) demands higher computational resources than just extracting low-level features available in the packet header. Moreover, since flow information such as bytes transferred or transfer rate are not directly available, obtaining such information requires processing packets inside a time window. In these sense, it can be said that those systems claiming real-time detection based on flow-based features [6,56,60], are actually performing some sort of batch detection. Clearly, a precise selection process must be conducted, considering the trade-off between the consumed computational resources and the attack detection accuracy provided by each type of feature.

4. Remaining open issues

The majority of the previously discussed works focus on the classification problem behind intrusion detection. If we considered the extremely precise results obtained by some approaches, we would say that the detection problem is near to be solved. Then, we should ask why none beyond pattern signature-based approach it is currently being used by network administrators. The fact is that previously analyzed works only cope with a subset of the problems that are essential to truly achieving intrusion detection, while not addressing the others. Issues like the still *high level of human interaction* and the *lack of model adjustment information* are critical to the detection process, specially if we consider that the ultimate goal of intrusion detection is to make security staff's life easier. In addition, a *proper traffic features identification* and the *lack of resource* *consumption information* are two other issues that should certainly be considered for an appropriate deployment on real networks. Finally, the *lack of public network traffic data* for proper evaluation of the different approaches is another issue that should be addressed by any further research made on this topic.

4.1. High level of human interaction

All current approaches still need a high degree of human interaction during the model construction process. SNORT [1] requires expert knowledge for writing signature patterns. Similar is the case of P-BEST, for which the writing of a set of implication rules could demand a considerable human effort.

On the other hand, most of the current approaches aiming at automatically generating network traffic models still need a high level of human preprocessing of the input data. For instance, data mining and machine learning misuse-based approaches require network traffic data labeled as normal or intrusive, whereas anomaly-based approaches require traffic records labeled as normal, in other words attack-free traffic data. Even unsupervised approaches, yet to a lesser extent, require input data remains under some specific distribution, a situation that can only be guarantee by human experts. This need of human preprocessing is perhaps one of the major drawbacks in the deployment of those approaches aiming at automatically generating traffic models.

The intrusion detection research community have started to react to this issue providing the so-called hybrid approaches [62,63]. Hybrid approaches usually combine well-established NIDS like SNORT with automatically generated traffic models techniques. Such combinations make the deployment of automatically generated models techniques more feasible. In addition, they seem to help reduce the required human preprocessing.

4.2. Lack of model adjustment information

Most of the discussed approaches using automatic traffic models seem to be aware of the high network variability and provide methods for adapting themselves as needed. However, the appropriate time for performing such adjustments seems not to be analyzed enough. Approaches should be able to transfer their results into information that allow the network security staff to easily evaluate their possible course of actions for further system adjustment. In addition, more methods focused on determining when the traffic model is no longer representative of current network traffic might help the acceptance of these automatic approaches.

4.3. Proper traffic feature identification

The majority of the current approaches address the detection problem from a broad point of view. These approaches rely on a large set of features to recognize several types of intrusion or, in some other cases, every possible type of intrusion. Unfortunately, using such broad strategies could complicate the already difficult problem of intrusion detection.

Alternatively, it seems more appropriate to focus on a per-attack detection strategy (i.e. specific to the kind of attack) and then analyze the adequate set of features capable of detecting it. Clearly, there is a relation between the kind of intrusion and the type of feature used for detecting such intrusion type. For instance, let's consider the Botnets detection problem. A major difference between Botnets and other detection problems consists of the use of a Command and Control communication channel (CC) for coordinating bots activities. We can infer that a proper identification of CC flows could eventually lead to detect infected machines on the network. In that case, the problem will be reduced to find just a concise set of features for proper CC flow characterization. In particular, this Botnet detection approach is currently being explored by members of the intrusion detection community [64].

This per-attack detection strategy however, brings about a new and important issue regarding the proper and efficient interaction between each one of these per-attack approaches. Possibly, the application of multi classifiers approaches and information fusion could provide some insights into this subject [65].

In addition, the size of the network and their associated security policies also have consequences when selecting network traffic features. As suggested by Garcia-Teodoro et al. [66], aspects like the behavior of a feature over the time or the source of traffic features can vary according to network size. On the other hand, when considering network security policies, there could be certain activities considered normal in some environments but not in others. Recently, researchers have started to analyze the consequences of applying some of the reviewed techniques under different network sizes and network authorities [67].

In summary, for a better addressing of the problem, current approaches should select their features according to a set of attacks sharing similar behavior, considering local policies and explicitly defining the appropriate network size.

4.4. Lack of resources consumption information

Determining the proper usage frequency for a given detection approach is crucial for analyzing its potential deployment on real networks. However, despite being a subject always present in surveys on intrusion detection, it is still difficult to establish the true usage frequency for many of the proposed approaches.

The problem is that only a few of the previously discussed intrusion detection approaches have analyzed the performance in terms of the computational resources required for generating the model as well as for evaluating a set of new network traffic records. As a result, it is difficult to establish the proper usage frequency of those approaches performing batch and real-time detection.

For instance, a batch detection approach could be able to perform detection every 5 min while another could be able to do it only every 12 h. Since no measure is provided, it is difficult to establish the proper network conditions in which such approaches will be adequate. Even worse is the case of those approaches claiming ability to perform real-time detection. In many cases such claims have proved to be valid only under certain network conditions (bandwidth, throughput, etc.). Moreover, as stated in Section 3.3.3 some of those approaches claiming real-time detection are actually performing batch detection.

This lack of resource consumption information could be one of the reasons why (with the exception of signature-based approaches) none of these approaches have been successfully deployed on real networks. Consequently, a better analysis about the needed computational resources could help in establishing the adequate usage frequency and therefore facilitating the deployment on real networks.

4.5. Lack of public network traffic data

Finally, another significant issue regarding intrusion detection is the lack of appropriate public data sets for evaluating the different approaches. Nowadays, the most commonly used data sets used for evaluation [68,69] are almost 12 years old, which make them practically obsolete if we consider the fast evolution of the network security field. Current data sets should contain information on automatic attacks (e.g. Botnets), Peer-to-Peer traffic and distributed DoS attacks, among others. Moreover, since IPv6 based network have become a reality [70], security threats and attack types that can affect such kind of network should also be included.

There have been some efforts for providing a framework for dataset generation in a proper and replicable way [71,72]. However, in many cases, the research community continues evaluating its intrusion detection approaches using their own data without providing information about data set generation. A situation that seriously affects the principle of replicability of experiments required for scientific research.

5. Conclusions

Several approaches have been proposed during the last 20 years of research in the intrusion detection field. All of these approaches aimed to facilitate the work of the network security staff providing some level of automation in the intrusion detection process. Certainly, such task cannot be considered easy since the non-stationary behavior of network traffic along with the permanent growth of the network throughput.

Nowadays, NIDS most successful approaches are those based on pattern signatures describing known attacks behavior. However, since signatures describing such attacks have to be written by experts, the signature-based approach lacks of the required level of automation needed by the intrusion detection community.

In order to deal with the drawbacks regarding signature-based approaches, some novel techniques were applied to intrusion detection. Such techniques includes statistical methods, machine learning and data mining methods, among others. However, despite being successful in terms of accuracy detection, there are still several concerns to be considered for deploying these techniques on real life scenarios. Issues like the need of proper resources consumption information along with a more attack-specific feature selection have not been analyzed in depth. But above all these issues, perhaps the main reason for their failure when deployed on real network environment is that these automatic detection techniques still require a high degree of human interaction.

In summary, it can be said that most of the discussed techniques have shown being capable to obtain high accuracy levels in a more automatic way. However, to reach the objective of fully automatizing the intrusion detection process, issues like the labeling of network traffic and its associated resources should be more deeply analyzed. In addition, reducing the problem scale by considering more attack-specific oriented approaches could also aid to achieve the complete deployment of automatic detection techniques.

References

[1] Roesch M. SNORT – lightweight intrusion detection for networks. In: Proceedings of the 13th USENIX conference on system administration, LISA '990. Berkeley, CA, USA: USENIX Association; 1999. p. 229–38.
[2] Kendall K. A database of computer attacks for the evaluation of intrusion detection systems. Master's thesis, AAI3006082; 1999.
[3] Lazarevic A, Kumar V, Srivastava J. Intrusion detection: a survey. In: Kumar V, Srivastava J, Lazarevic A, editors. Managing cyber threats. Massive computing, vol. 5. US: Springer; 2005. p. 19–78.
[4] Mukherjee B, Heberlein L, Levitt K. Network intrusion detection. Netw IEEE 1994;8:26–41.
[5] Patcha A, Park J-M. Network anomaly detection with incomplete audit data. Comput Netw 2007;51:3935–55.
[6] Lindqvist U, Porras P. Detecting computer and network misuse through the production-based expert system toolset (P-BEST). In: Proceedings of the 1999 IEEE symposium on security and privacy; 1999. p. 146–61.
[7] Porras PA, Neumann PG. EMERALD: event monitoring enabling responses to anomalous live disturbances. In: Proceedings of the 20th national information systems security conference; 1997. p. 353–65.

[8] Lee W, Stolfo SJ. Data mining approaches for intrusion detection. Proceedings of the 7th conference on USENIX security symposium, vol. 7. Berkeley, CA, USA: USENIX Association; 1998. p. 6.
[9] Cannady J. Artificial neural networks for misuse detection. In: National information systems security conference, Arlington, VA, USA; 1998. p. 368–381.
[10] Liu G, Yi Z, Yang S. A hierarchical intrusion detection model based on the PCA neural networks. Neurocomputing 2007;70:1561–8.
[11] Kumar PAR, Selvakumar S. Distributed denial of service attack detection using an ensemble of neural classifier. Comput Commun 2011;34:1328–41.
[12] Ahmad I, Abdullah AB, Alghamdi AS. Artificial neural network approaches to intrusion detection: a review. In: Proceedings of the 8th Wseas international conference on telecommunications and informatics. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS); 2009. p. 200–5.
[13] Xu Q, Pei W, Yang L, Zhao Q. An intrusion detection approach based on understandable neural network trees. Int J Comput Sci Netw Secur 2006;6:229–34.
[14] Abraham A, Grosan C. Evolving intrusion detection systems. In: Nedjah N, Mourelle L, Abraham A, editors. Genetic systems programming. Studies in computational intelligence, vol. 13. Berlin/Heidelberg: Springer; 2006. p. 57–79.
[15] Li W. Using genetic algorithm for network intrusion detection. In: Proceedings of the United States department of energy cyber security group 2004 training conference, Kansas City, Kansas, Department of Computer Science and Engineering, Mississippi State University, Mississippi State; 2004. p. 24–7.
[16] Gong RH, Zulkernine M, Abolmaesumi P. A software implementation of a genetic algorithm based approach to network intrusion detection. In: International conference on software engineering, artificial intelligence, networking and parallel/distributed computing and international workshop on self-assembling wireless networks; 2005. p. 246–53.
[17] Bankovic Z, Stepanovic D, Bojanic S, Nieto-Taladriz O. Improving network security using genetic algorithm approach. Comput Electr Eng 2007;33:438–51.
[18] Vollmer T, Alves-Foss J, Manic M. Autonomous rule creation for intrusion detection. In: IEEE Symposium on computational intelligence in cyber security (CICS); 2011. p. 1–8.
[19] Gomez J, Dasgupta D. Evolving fuzzy classifiers for intrusion detection. In: Proceedings of the 2002 IEEE workshop on information assurance. New York: IEEE Computer Press; 2002.
[20] Bridges S, Vaughn R. Fuzzy data mining and genetic algorithms applied to intrusion detection. In: Proceedings of the 23rd national information systems security conference, Citeseer, held in Baltimore, MA; 2000. p. 13–31.
[21] Chen C, Mabu S, Yue C, Shimada K, Hirasawa K. Analysis of fuzzy class association rule mining based on genetic network programming. In: ICCAS-SICE; 2009. p. 3480–4.
[22] Abadeh MS, Mohamadi H, Habibi J. Design and analysis of genetic fuzzy systems for intrusion detection in computer networks. Expert Syst Appl 2011;38:7067–75.
[23] Luo J. Integrating fuzzy logic with data mining methods for intrusion detection. Master's thesis, Department of Computer Science, Mississippi State University; 1999.
[24] Florez G, Bridges S, Vaughn R. An improved algorithm for fuzzy data mining for intrusion detection. In: Fuzzy information processing society. Proceedings. NAFIPS. 2002 Annual meeting of the North American; 2002. p. 457–62.
[25] Ye N, Li X, Emran S. Decision tree for signature recognition and state classification. In: Proceedings of IEEE systems, man and cybernetics information assurance and security workshop; 2000.
[26] Mukkamala S, Janoski G, Sung A. Intrusion detection using neural networks and support vector machines. In: Proceedings of the international joint conference on neural networks, IJCNN '02, vol. 2; 2002. p. 1702–7.
[27] Sung A, Mukkamala S. Identifying important features for intrusion detection using support vector machines and neural networks. In: Proceedings of 2003 symposium on applications and the internet; 2003. p. 209–16.
[28] Bonabeau E, Dorigo M, Theraulaz G. Swarm intelligence: from natural to artificial systems. New York, NY, USA: Oxford University Press, Inc.; 1999.
[29] Ramos V, Abraham A. ANTIDS: self organized ant-based clustering model for intrusion detection system. In: Abraham A, Dote Y, Furuhashi T, Koppen M, Ohuchi A, Ohsawa Y, editors. Soft computing as transdisciplinary science and technology. Advances in intelligent and soft computing, vol. 29. Berlin/Heidelberg: Springer; 2005. p. 977–86.
[30] Banerjee S, Grosan C, Abraham A. IDEAS: Intrusion detection based on emotional ants for sensors. In: International conference on intelligent systems design and applications; 2005. p. 344–9.
[31] He J, Long D, Chen C. An improved ant-based classifier for intrusion detection. Int Conf Nat Comput 2007;4:819–23.
[32] Guolong C, Qingliang C, Wenzhong G. A PSO-based approach to rule learning in network intrusion detection. In: Cao B-Y, editor. Fuzzy information and engineering. Advances in intelligent and soft computing, vol. 40. Berlin/Heidelberg: Springer; 2007. p. 666–73.
[33] Kittler J, Hatef M, Duin RPW, Matas J. On combining classifiers. IEEE Trans Pattern Anal Mach Intell 1998;20:226–39.
[34] Giacinto G, Roli F, Didaci L. Fusion of multiple classifiers for intrusion detection in computer networks. Pattern Recognit Lett 2003;24:1795–803.
[35] Staniford S, Hoagland JA, McAlerney JM. Practical automated detection of stealthy portscans. J Comput Secur 2002;10:105–36.
[36] Lakhina A, Papagiannaki K, Crovella M, Diot C, Kolaczyk ED, Taft N. Structural analysis of network traffic flows. In: Proceedings of the joint international conference on measurement and modeling of computer systems. SIGMETRICS '04/Performance '04. New York, NY, USA: ACM; 2004. p. 61–72.
[37] Gu Y, McCallum A, Towsley D. Detecting anomalies in network traffic using maximum entropy estimation. In: Proceedings of the 5th ACM SIGCOMM conference on internet measurement, IMC '05. Berkeley, CA, USA: USENIX Association; 2005. p. 32.
[38] Mahoney MV, Chan PK. Learning nonstationary models of normal network traffic for detecting novel attacks. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. KDD '02. New York, NY, USA: ACM; 2002. p. 376–85.
[39] Mahoney MV, Chan PK. PHAD: packet header anomaly detection for identifying hostile network traffic. Technical report CS-2001-4 2004. Florida Institute of Technology; 2001.
[40] Mahoney MV. Detecting novel attacks by identifying anomalous network packet headers. Technical report CS-2001-2. Florida Institute of Technology; 2001.
[41] Mahoney M, Chan P. Learning models of network traffic for detecting novel attacks. Technical report CS-2002-08. Florida Institute of Technology; 2002.
[42] Cohen W. Fast effective rule induction. In: Proceedings of the 12th international conference on machine learning. Morgan Kaufman; 1995. p. 115–23.
[43] Portnoy L, Eskin E, Stolfo S. Intrusion detection with unlabeled data using clustering. In: Proceedings of ACM CSS workshop on data mining applied to security, Philadelphia, PA.
[44] Chandola V, Banerjee A, Kumar V. Anomaly detection: a survey. ACM Comput Surv 2009;41:15:1–15:58.
[45] Lazarevic A, Ertöz L, Kumar V, Ozgur A, Srivastava J. A comparative study of anomaly detection schemes in network intrusion detection. In: Proceedings of the third SIAM international conference on data mining.
[46] Ertöz L, Eilertson E, Lazarevic A, Tan PN, Kumar V, Srivastava J, et al. MINDS – minnesota intrusion detection system. MIT Press; 2004.
[47] Su M-Y. Using clustering to improve the knn-based classifiers for online anomaly network traffic identification. J Netw Comput Appl 2011;34:722–30.
[48] Wang F, Qian Y, Dai Y, Wang Z. A model based on hybrid support vector machine and self-organizing map for anomaly detection. Int Conf Commun Mobile Comput 2010;1:97–101.
[49] Eskin E, Arnold A, Prerau M, Portnoy L, Stolfo S. A geometric framework for unsupervised anomaly detection. In: Barbara D, Jajodia S, editors. Advances in information security. Springer; 2002.
[50] Giacinto G, Perdisci R, Del Rio M, Roli F. Intrusion detection in computer networks by a modular ensemble of one-class classifiers. Inform Fusion 2008;9:69–82.
[51] Rehak M, Pechoucek M, Grill M, Stiborek J, Bartos K, Celeda P. Adaptive multiagent system for network traffic monitoring. Intell Syst IEEE 2009;24:16–25.

[52] Sabater J, Sierra C. Review on computational trust and reputation models. Artif Intell Rev 2005;24:33–60.
[53] Paxson V. Bro: a system for detecting network intruders in real-time. Comput Netw 1999;31:2435–63.
[54] Suricata. Suricata intrusion detection system; 2011.
[55] Gomez J, Dasgupta D, Nasraoui O, Gonzalez F. Complete expression trees for evolving fuzzy classifier systems with genetic algorithms and application to network intrusion detection. In: Proceedings NAFIPS fuzzy information processing society 2002 annual meeting of the North American. p. 469–74.
[56] Porras P, Valdes A. Live traffic analysis of TCP/IP gateways. In: Internet society symposium on network and distributed system security, San Diego.
[57] Fowler H, Leland W. Local area network characteristics, with implications for broadband network congestion management. IEEE J Select Areas Commun 1991;9:1139–49.
[58] Mahoney M, Chan P. Learning rules for anomaly detection of hostile network traffic. In: Third IEEE international conference on data mining. ICDM 2003; 2003. p. 601–4.
[59] Lakhina A, Crovella M, Diot C. Mining anomalies using traffic feature distributions. SIGCOMM Comput Commun Rev 2005;35:217–28.
[60] Lakhina A, Crovella M, Diot C. Diagnosing network-wide traffic anomalies. SIGCOMM Comput Commun Rev 2004;34:219–30.
[61] Callegari C, Gazzarrini L, Giordano S, Pagano M, Pepe T. Detecting network anomalies in backbone networks. In: Jha S, Sommer R, Kreibich C, editors. Recent advances in intrusion detection. Lecture notes in computer science, vol. 6307. Berlin/Heidelberg: Springer; 2010. p. 490–1.
[62] AydIn MA, Zaim AH, Ceylan KG. A hybrid intrusion detection system design for computer network security. Comput Electr Eng 2009;35:517–26.
[63] Hwang K, Cai M, Chen Y, Qin M. Hybrid intrusion detection with weighted signature generation over anomalous internet episodes. IEEE Trans Depend Secure Comput 2007;4:41–55.
[64] Gu G, Perdisci R, Zhang J, Lee W. Botminer: clustering analysis of network traffic for protocol- and structure-independent botnet detection. In: Proceedings of the 17th conference on Security symposium, USENIX. Berkeley, CA, USA: USENIX Association; 2008. p. 139–54.
[65] Shoemaker L, Hall LO. Anomaly detection using ensembles. In: Proceedings of the 10th international conference on Multiple classifier systems. MCS'11. Berlin, Heidelberg: Springer-Verlag; 2011. p. 6–15.
[66] Garcia-Teodoro P, Diaz-Verdejo J, Macia-Fernandez G, Vazquez E. Anomaly-based network intrusion detection: techniques, systems and challenges. Comput Secur 2009;28:18–28.
[67] Mehdi S, Khalid J, Khayam S. Revisiting traffic anomaly detection using software defined networking. In: Sommer R, Balzarotti D, Maier G, editors. Recent advances in intrusion detection. Lecture notes in computer science, vol. 6961. Berlin/Heidelberg: Springer; 2011. p. 161–80.
[68] Lippmann RP, Cunningham RK. Improving intrusion detection performance using keyword selection and neural networks. Comput Netw 2000;34:597–603.
[69] Lippmann R, Haines JW, Fried DJ, Korba J, Das K. The 1999 darpa off-line intrusion detection evaluation. Comput Netw 2000;34:579–95.
[70] Zagar D, Grgic K, Rimac-Drlje S. Security aspects in ipv6 networks – implementation and testing. Comput Electr Eng 2007;33:425–37.
[71] Wright C, Connelly C, Braje T, Rabek J, Rossey L, Cunningham R. Generating client workloads and high-fidelity network traffic for controllable, repeatable experiments in computer security. In: Jha S, Sommer R, Kreibich C, editors. Recent advances in intrusion detection. Lecture notes in computer science, vol. 6307. Berlin/Heidelberg: Springer; 2010. p. 218–37.
[72] Shiravi A, Shiravi H, Tavallaee M, Ghorbani AA. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Comput Secur 2012;31:357–74.

Carlos Catania received the BS degree in information systems from Universidad Champagnat, Argentina, in 2004 and the M.Sc. degree in networking from Universidad de Mendoza, Argentina in 2007. He is presently pursuing the PhD degree in computer sciences at UNICEN, Tandil, Argentina. His research interests include Internet security and distributed computing systems.

Carlos Garcia Garino graduated in engineering at University of Buenos Aires in 1978 and received a Ph.D. degree from UPC, Barcelona, Spain in 1993. Currently he is Full Professor at the School of Engineering and Head of the ITIC Research Institute of UNCuyo, Argentina. His research interests include Computational Mechanics, Computer Networks, and Distributed Computing. He has more than 50 papers published in scientific journals and proceedings of international conferences.