Experiences on Designing an Integral Intrusion Detection System

Yoseba K. Penya
S$^3$Lab
Deusto Technology Foundation
Bilbao, Basque Country
ypenya@tecnologico.deusto.es

Pablo G. Bringas
S$^3$Lab
Deusto Technology Foundation
Bilbao, Basque Country
pgb@tecnologico.deusto.es

Abstract

Network Intrusion Detection Systems (NIDS) have the challenge to prevent network attacks and unauthorised remote use of computers. In order to achieve this goal, NIDS usually follow two different strategies. The first one aims at detecting forbidden usage of the network and the second one concentrates on finding illegitimate behaviour. The first methodology accomplishes its goal by defining all possible attacks and the second by modelling the normal usage to detect anything that does not fit on that muster; this difference has rendered both alternatives so far incompatible. In previous works we have presented ESIDE-Depian, the first inherently unified misuse and anomaly detector. This paper focuses on the problems and difficulties that arose in the integration process and the solutions designed to overcome them.

1. Introduction

Nowadays, the use of Internet is extended worldwide and its phenomenon brings multiple advantages but also several problems as well. As the number of users grows, the frequency, variety, and virulence of the attacks suffered through the network has increased drastically in the last years. Therefore, old-fashioned computer-centred protections have been forced to adapt to the new situation in order to provide an effective protection against any kind of threaten. In this way, Network Intrusion Detection Systems (NIDS) have been postulated as this effective protection since they focus exactly on distinguishing legitimate from malicious network users.

To this end, based on the sort of menaces they try to prevent, NIDS can be divided into misuse or anomaly detectors, which refers to the sort of danger they prevent. On the one hand, misuse detection systems deal with menaces already known beforehand. Basically, these systems manage a comprehensive attack base and their work consists of invigilating at all incoming traffic to detect any sequence that appears in that knowledge base.

On the other hand, anomaly detection systems are more ambitious and intend to discover new unknown menaces (the so-called zero-day attacks). These systems model legitimate system usage in order to afterwards obtain a certainty measure of potential deviations from that normal profile. Each deviation that is found significant enough will be considered anomalous and notified to a human operator.

The working way is in either cases the same: the vigilant system inspects network traffic and, if a threat is detected, it triggers the alarm. This alarm can be analysed manually or processed automatically either to filter intruder actions (in line with the intrusion prevention paradigm), to reconfigure the environment or to collect audit information.

Now, the problem of network intrusion detection is that one approach focuses only on the bad part of the network traffic and the other just on the good one. Network traffic is, in both cases, the same vast information source from which they extract information and learn. The bridge that can connect misuse and anomaly detection is the ability to induce knowledge from a well-known data corpus in order to apply it to a new one. In other words, a tool or methodology that, based on the past good and bad network traffic, is able to distinguish both good and bad parts in the incoming one.

In this way, Bayesian networks [10] represent the sort of tool that can help us to achieve this integration. After a training period, the Bayesian network learns the behaviour of the system, so it is able to foresee the result. In all applications of Bayesian networks, the respective system is modelled as a set of interconnected variables whose output is always the result of the production. In this way, we can model a NIDS as
a constellation of variables controlling the type of the traffic, information on packet headers, packet payload or their temporal relationships (i.e. to check whether they form a coordinated attack). If we connect this representation to an attack variable, we will be able, after a proper training, to predict when do incoming packets represent a menace to the system.

Against this background, we have presented [4] and evaluated [11] ESIDE-Depian (Intelligent Security Environment for Detection and Prevention of Network Intrusions), the first NDIS to combine and unify misuse and anomaly detection. In this paper, we focus on the difficulties of integrating both approaches and explain how we overcame those problems. The remainder of the paper is structured as follows. Section 2 illustrates the differences between misuse and anomaly detections systems. Section 3 introduces the concept of a Bayesian Network and details the creating and training process of the one tailored to network intrusion detection. Section 4 describes how the integration of misuse and anomaly strategies was achieved. Section 5 concentrates on the problems appeared and the solution designed to solve them. Section 6 discusses related work and, finally, section 7 draws the avenues of future work.

2. Misuse vs Anomaly Detection Systems

Currently, misuse detection is the most extended approach for intrusion prevention, mainly due to its efficiency and easy administration. It’s philosophy is quite simple: based on a rule base that models a high number of network attacks, the system compares incoming traffic with the registered patterns to identify any of these attacks. Hence, it does not produce any false positive (since it always finds exactly what is registered) but it cannot detect any new threat. Further, any slightly-modified attack will pass unnoticed. Finally, the knowledge base itself poses one of the biggest problems to misuse detection: as it grows, the time to search on it increases as well and, finally, it may require too long to be used on real-time.

Anomaly detection systems, on the contrary, start not from malicious but from legitimate behaviour in order to model what it is allowed to do. Any deviation from this conduct will be seen as a potential menace. Unfortunately, this methodology is a two-sided sword since, though it allows to discover new unknown risks, it also produces false positives (i.e. packets or situations marked as attack when they are not). Moreover, anomaly detection presents a constant throughput since its knowledge base does not grow uncontrollably but gets adapted to new situations or behaviours. Again, an advantage is also source of problems because it is theoretically possible to make use of this continuous learning to little by little modify the knowledge so it ends seeing attacks as proper traffic (in NIDS jargon, this phenomenon is known as session creeping). This is, its knowledge tends to be unstable. Finally, anomaly detection, unlike misuse, demands high maintenance efforts (and costs). In sum, both alternatives present notable disadvantages that demand a new approach for network intrusion prevention.

3. Bayesian-Network-based Intrusion Detection

The Bayes’ theorem is the basis of the so-called Bayesian inference, a statistical inference method that allows, upon a number of observations, to obtain or update (if the system is already working) the probability that a hypothesis may be true. In this way, Bayes’ theorem adjusts the probabilities as new informations on evidences appear.

Thus, Bayesian networks are probabilistic models for multivariate analysis. Formally, they are directed acyclic graphs associated to a probability distribution function [2]. Nodes in the graph represent variables (any kind, be it a premise or a conclusion), and the arcs, conditional dependencies between such variables. Further, the probability function illustrates the strength of these relationships (i.e. arcs) in the graph.

To our needs, the most important ability of Bayesian networks is their capability of inferring the probability that a certain hypothesis becomes true out of the values that the rest of variables forming the Bayesian network take. In this way, we have divided the network traffic according to its type (TCP-IP, UDP-IP and ICMP-IP) and created three Bayesian networks (experts) to analyse their respective packet headers (which is an strategy already proven successful in this area [1]). Moreover, in order to cover all possible kind of menaces, we also have to take into account the payload (i.e. body) of the packet and the potential temporal dependencies between packets. Therefore, we have added 2 further experts, the protocol payload and the connection tracking one, respectively.

In each case, the Bayesian network is composed of several variables depending on the protocol and the expert; the value to induce is always the probability that the analysed packet is part of an attack. See [4] for a more accurate description of the Bayesian experts.

---

1 An early version of this paper appeared at the 18th International Workshop on Database and Expert Systems Applications (DEXA 2007), 3-7 September 2007, Regensburg, Germany. IEEE Computer Society, pp.365-371.
3.1 Bayesian network obtaining process

The creation and setting-up of each Bayesian network (say expert) comprises the following phases:

- **Traffic sample obtaining.** First we need to establish the information source in order to gather the sample. This set usually includes normal traffic (typically gathered from the network by sniffing, arp poisoning or so), as well as malicious traffic generated by the well-known arsenal of hacking tools (e.g. MetaSploit).

- **Structural Learning.** The next step is devoted to define the operational model ESIDE-Depian should work within. With this goal in mind, we have to provide logical support for knowledge extracted from network traffic information. Packet parameters need to be related into a Bayesian structure of nodes and edges, in order to ease the later conclusion inference over this mentioned structure.

- **Parametric Learning.** The knowledge model fixed so far is a qualitative one. Therefore, the following step is to apply parametric learning in order to obtain the quantitative model representing the strength of the collection of previously learned relationships, before the exploitation phase began.

- **Bayesian Inference.** Next, every packet capture from the target communication infrastructure needs one value for the posterior probability of a badness variable, given the set of observable packet detection parameters. So, we need an inference engine based on Bayesian evidence propagation. Thereby, already working in real time, incoming packets are analysed by this method (with the basis of observable detection parameters obtained from each network packet) to define the later probability of the attack variable. The continuous probability value produced here represents the certainty that an evidence is good or bad.

3.2 Naive Bayesian Network of the Expert Modules

Having different Bayesian modules is a twofold strategy. On the one hand, the more specific expertise of each module allows them to deliver more accurate verdicts but, on the other hand, there must be a way to solve possible conflicting decisions. In other words, an unique measure must emerge from the diverse judgements.

To this end, ESIDE-Depian presents a two-tiered schema where the first layer comprises the expert modules (TCP-IP, UDP-IP, ICMP-IP, Connection Tracking and Protocol Payload) and the second layer includes only one class parameter: the most conservative response of the experts (in order to prioritize the absence of false negatives in front of false positives). Both layers form, in fact, a naive Bayesian network (also called Naive classifier [2]). This approach provides a good balance between representative power and performance, and also affords interesting flexibility capabilities which allow, for instance, ESIDE-Depian’s dynamical enabling and disabling of expert modules.

4 Integration of Misuse and Anomaly detection

The internal design of ESIDE-Depian is principally determined by its dual nature. Being both a misuse and anomaly detection system requires answering to sometimes clashing needs and demands. This is, it must be able to simultaneously offer efficient response against both well-known and zero-day attacks. The Bayesian network, according to the ability to extrapolate its knowledge and apply it to not-previously seen cases, is the ideal tool for these zero-day attacks. Still, we have to integrate detection of already registered threads and provide an efficient methodology to update and to continuously adapt to changes. ESIDE-Depian achieves this objectives in two ways. First, it incorporates Snort (a rule-based state of the art misuse detection system [13]) to the training of the Bayesian network. There, Snort provides information regarding the legitimacy or malice of the network packets. Specifically, Snort’s main decision about a packet is added to the set of detection parameters, receiving the name of attack variable. In this way, it is possible to obtain a complete sample of evidences, including, in the formal aspect of the sample, both protocol fields as well as Snort labelling information. Therefore, it combines knowledge about normal behaviour and also knowledge about well-known attacks, or, in other words, information necessary for misuse detection and for anomaly detection.

Second, already in working-time, every packet gets Snort’s opinion added as a badness variable. In this way, experts know again the decision of Snort in beforehand and can act in consequence according to their knowledge model.

5 Problems and solutions

This section gives account of the main problems that emerged during the design and test phase. More accurately, they were:
• **Integration of Snort:** The first difficulty we faced was to find an effective way of integrating Snort in the system. Our first attempt placed the verdict of Snort at the same level as those of the Bayesian experts in the Naive classifier. This strategy failed to capture the real possibilities of Bayesian networks since it simply added the information generated by Snort at the end of the process, more as a graft than a real integrated part of the model. The key aspect in this situation was letting the Bayesian network absorb Snort’s knowledge to be able to actually replace it. Therefore, in the next prototype we recast the role of Snort as a kind of advisor, both in training and in working time. In this way, the Bayesian experts use Snort’s opinion on the badness of incoming packets in the learning procedure and afterwards (as described in section 4) and manage to exceed Snort’s knowledge [11].

• **Different parameter nature:** The next challenge consisted on the different nature of the parameters that ESIDE-Depian has to control. Whereas TCP, UDP and ICMP are static and refer exclusively to one packet (more accurately to its header), the connection tracking and payload analysis experts are dynamic and require the introduction of the time notion. In this way, the connection tracking expert checks if packets belong to an organised sequence of an attack [3], so time is needed to represent predecessor and successor events. Similarly, the payload analysis expert must model state transitions between symbols and tokens that appear on it. Thus, in the same way that different tests had to be performed (see [11]), we had to prepare a special traffic sample tailored to the kind of traffic those expert should focus to inspect.

• **Disparity between good and bad traffic amount:** Another problem to tackle was the composition of the traffic sample used to train the first group of experts (TCP, UDP, ICMP). In order to help the acquisition of the initial reference knowledge in the training phase, the BN is fed with a traffic sample basically based on the attack-detection rules battery provided by Snort. Therefore, the training acquaints the BN with either kind of traffic simultaneously, good and bad. Still, due to the disparity in the amount of packets belonging to one or another, traces containing attacks have to be fed several times (in the so-called presentation cycles) in order to let the BN learn to evaluate them properly.

• **Task parallelization:** Bayesian networks require many computational resources. Thus, some of the tasks to be performed were designed in a parallel way to accelerate it. For instance, the structural learning was devoted concurrently in 60 computers. In this way, the traffic sample (about 900.000 packets) was divided in blocks of 10.000 of packets that were processed with the PC-Algorithm (see section 3). Moreover, already on real-time, each expert was placed in a different machine not only to divide the amount of resources consumed but also to prevent from having a single point of failure.

• **False positives and false negatives:** Finally, we coped with a usual problem related to anomaly detection systems: false positives (i.e. packets marked as potentially dangerous when they are harmless). In fact, minimising false positives is one of the pending challenges of this approach [8]. Nevertheless, the double nature of ESIDE-Depian as anomaly and misuse detector reduces the presence of false positives to a minimum [11]. False negatives, on the contrary, did threaten the system and, in this way, in the experiments accomplished in ESIDE-Depian, security was prioritized above comfort, so quantitative alarm-thresholds were set upon the production of the minimum false negatives, in spite of the false positive rates. It is possible to find application domains, e.g. anti-virus software, in which false positive numbers are the target to be optimized, in order not to saturate the final user or the system administrator. Also in these cases ESIDE-Depian is able to manage the detection problem, simply by the specific setting-up of the mentioned thresholds.

### 6. Related Work

Different approaches to develop network misuse detectors include expert systems [1] or intelligent agent systems [5] (see [6] for a detailed analysis of related work in this area). Research in network anomaly detection has applied several well-known Artificial Intelligence paradigms such as support-vector machines [9] or diverse data-mining-based approaches [7]. Still, there is only one attempt to bring these two strands of work together. More specifically, in [14], they achieve to combine anomaly and misuse but its analysis of network packets is too superficial to yield any good results in real life. In particular, despite the brilliant main contribution about integrating misuse-based and anomaly-based detection in one inherently unified and compact knowledge representation model, this work presents several shortcomings that prevent it from being applied in real scenarios: on the one hand, this approach only
considers 7 detection parameters (against 33 considered in the work we present here and in [4] [11]) which are, besides, extracted only from TCP and IP protocol headers. Popular protocols as UDP connection-less protocol or the very-very problematic ICMP protocol are not taken into consideration. On the other hand, Bayesian Networks’ full capabilities are not really used. Thus, one of the most important topics provided by the Bayesian approach, the structural learning concept, is not definitively applied. Instead, they propose the Naïve approach, which assumes the (unrealistic) hypothesis that there is no statistical dependence among the collection of detection parameters. Finally, time notion does not play any role in the analysis model, even under the focus achieved over the TCP target protocol, which is, of course, connection-oriented and, so, chronological dependence among events is sure to appear.

7 Conclusion and Future Lines

Network Intrusion Detection Systems (NIDS) aim at preventing attacks and unauthorised access through the network. Traditionally, there have been two different philosophies to achieve this goal. On the one hand, misuse detection systems (which deal with menaces already known in beforehand). On the other hand, anomaly detection systems (which intend to discover new unknown menaces). Both alternatives pursue different strategies with the result that each one is specialised only on a kind of attack but fails to detect the other kind.

In this way, in [4] and [11], we presented and evaluated ESIDE-Depian, the first inherently unified misuse and anomaly detector. ESIDE-Depian successfully integrates both approaches by using a Bayesian network that is trained for either kind of attack.

Nevertheless, the integration of misuse and anomaly was very challenging and we had to cope with the following problems. First of all, the most effective placement for Snort within the model. Then, the composition of these training samples posed also a problem, since the ratio between good and bad traffic was too low. Furthermore, is very resource demanding and, finally, integrating misuse and anomaly simultaneously prevented it from presenting a high rate of false negatives, which is a typical inconvenience of anomaly detectors, but, still, we had to cope with the problem of false negatives.

Future work will focus on the so-called data-aging problem. The constant feeding of upcoming data issues poses a new challenge to the Bayesian network: it extends and enhances its knowledge base but, in parallel, information about these new traffic has too less importance compared to older ones, and therefore, predictions about new packets are not as exact as they should be. We have tackled this phenomenon in a research in which we used a Bayesian network to predict the apparition of a defect in a foundry process [12]. Therefore, further work will be concentrated on how to extrapolate the techniques developed to the very special case we deal here with.

References