14

Bayesian Networks for Network Intrusion Detection

Pablo G. Bringas and Igor Santos
University of Deusto
Spain

1. Introduction

The increasing use of Internet has dramatically contributed to the growing number of threats that inhabit within it. Seeking for a better protection, Computer Security and, specifically, Network Intrusion Detection Systems (NIDS) have risen to become a topic of research and concern in order to fight these threats.

More accurately, a NIDS is a type of computer software that is able to distinguish legitimate network users from malicious ones. Moreover, due to the rising complexity and volume of the attacks, NIDS are performed in an automated manner, so the NIDS software monitors system usage to identify behaviour breaking the security policy. Generally, NIDS are categorised based in their scope: misuse network detectors and anomaly detectors. On the one hand, misuse detection systems deal with menaces already known in 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 try to discover new unknown threats (the so-called zero-day attacks). To this extent, these systems model benign or legitimate system usage in order to thereafter 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. Research in network anomaly detection has applied several well-known Artificial Intelligence paradigms such as finite automata (Vigna et al., 2000), neural networks (Mukkamala et al., 2005), genetic algorithms (Kim et al., 2005), fuzzy logic (Chavan et al., 2004), support-vector machines (Mukkamala et al., 2005) or diverse data-mining-based approaches (Lazarevic et al., 2003).

Actually, these solutions, both misuse and anomaly, perform better or worse against a network attack. Misuse detection systems are overwhelmed since they cannot face menaces that have not been previously described in their rule base but they overcome very fast the ones that have. Unfortunately, anomaly detection itself may not be considered as the the perfect solution, as well. In this way, it is much less exact than misuse detectors with well-known attacks and, despite they do find zero-day threads, sometimes they also produce false positives (i.e. select as a menace what is perfectly right). Summarizing, each approach is clearly surpassed when it comes to the other's area of expertise and the goal is, thus, to find the way to integrate both system's benefits while reducing their weaknesses.

In this way, Bayesian networks (Pearl & Russell, 2000) represent the sort of tool that can help us to achieve this integration. Specifically, they are probabilistic models very helpful when facing problems that require predicting the outcome of a system consisting of a high number of interrelated variables. After a training period, the Bayesian network *learns* the behaviour of

the model and, thereafter it is able to foresee its outcome. In this way, successful applications of Bayesian networks include for instance email classification for spam detection (Yang et al., 2006), failure detection in industrial production lines (Masruroh & Poh, 2007) (Liu & Li, 2007), weather forecasting (Abramson et al., 1996) (Cofiño et al., 2002), intrusion detection over IP networks (Krügel et al., 2003) (Faour et al., 2006) or reconstruction of traffic accidents (Davis & Pei, 2003) (Davis, 2006). In all cases, the respective target problem is modelled as a constellation of interconnected variables whose output is always the result of the prediction (e.g. spam found, failure detected, intrusion noticed and so on). Therefore, 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.
Given this background, we present ESIDE-Depian (Intelligent Security Environment for Detection and Prevention of Network Intrusions), the first inherently unified misuse and anomaly detector. Besides, we focus on the integration of anomaly and misuse and show how this goal can be achieved by using a Bayesian network. In addition, we test this integration with real network attacks and show ESIDE-Depian's efficiency both as misuse and as anomaly detection.
The remainder of the chapter is organised as follows. follows. Section 2 illustrates the differences between misuse and anomaly detections systems. Section 3 details the concept of a Bayesian network and describes the used in ESIDE-Depian. Section 4 describes how ESIDE-Depian integrates misuse and anomaly prevention. Section 5 presents the experiments to evaluate this integration and discuses their results. Section 6 concentrates on the problems appeared and the solution designed to solve them. Section 7 discusses related work and, finally, section 8 concludes and outlines the avenues of future work.

2. Misuse versus Anomaly Detection

Currently, misuse detection is the most extended approach for intrusion prevention, mainly due to its efficiency and easy administration (Bringas et al., 2009). Its 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). In fact, minimising false positives is one of the pending challenges of this approach (Kruegel, 2002). Moreover, misuse 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). In other words, its knowledge tends to be unstable. Finally, anomaly detection, unlike misuse, demands high maintenance efforts (and costs).

In summary, both alternatives present notable disadvantages that demand a new approach for network intrusion prevention.

3. Bayesian-network-based intrusion detection

3.1 Background

Reverend Thomas Bayes pioneered with his work the research on cause-consequence relationships. The most important fruit of that investigation, known as the "Bayes' theorem" (Bayes, 1763) in his honour, 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.

According to its classical formulation, given two events A and B, the conditional probability $P(A|B)$ that A occurs if B occurs can be obtained if we know the probability that A occurs, P(A), the probability that B occurs, $P(B)$, and the conditional probability of B given A, $P(B|A)$ (as shown in equation 1):

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \tag{1}$$

More accurately, Bayesian Networks (Pearl & Russell, 2000) are defined as graphical probabilistic models for multivariate analysis. Specifically, they are directed acyclic graphs that have an associated probability distribution function (Castillo et al., 1996). Nodes within the directed graph represent problem variables (they can be either a premise or a conclusion) and the edges represent conditional dependencies between such variables. Moreover, the probability function illustrates the strength of these relationships in the graph (Castillo et al., 1996) (Figure 1).

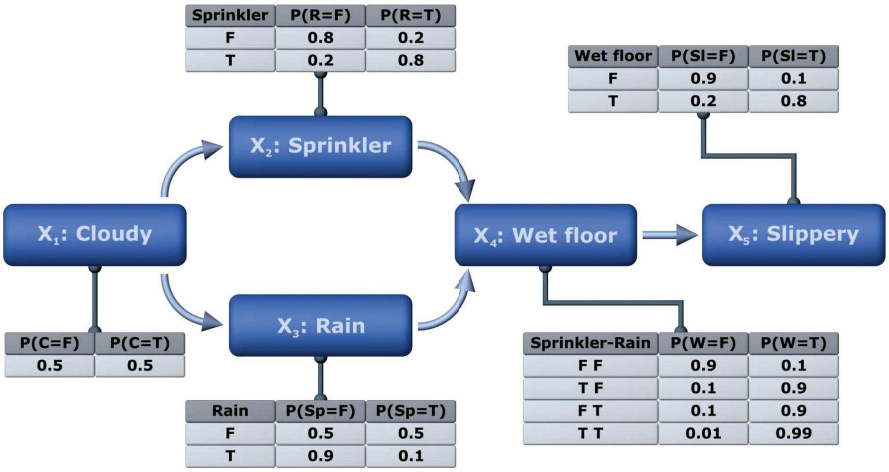

Fig. 1. Example of a Bayesian Network.

Formally, let a Bayesian Network B be defined as a pair, $B = (D, P)$, where D is a directed acyclic graph; $P = \{p(x_1|\Psi_2), ..., p(x_n|\Psi_n)\}$ is the set composed of n conditional probability

functions (one for each variable); and Ψ_i is the set of parent nodes of the node X_i in D. The set P is defined as the *joint probability density function* (Castillo et al., 1996) (equation 2)

$$P(x) = \prod_{i=1}^{n} p(x_i|\Psi_i) \tag{2}$$

The most important capability of Bayesian Networks is their ability to determine the probability that a certain hypothesis is true (e.g., the probability of an e-mail to be spam or legitimate) given a historical dataset.

3.2 Bayesian Network Obtaining Process

The obtaining of the knowledge model in an automated manner can be achieved in an unsupervised or supervised way.

Typically, unsupervised learning approaches don not take into consideration expert knowledge about well-known attacks. They achieve their own decisions based on several mathematical representations of distance between observations from the target system, revealing themselves as ideal for performing Anomaly Detection.

On the other hand, supervised learning models do use expert knowledge in their making of decisions, in the line of Misuse Detection paradigm, but usually present high-cost administrative requirements. Therefore, both approaches present important advantages and several shortcomings. Being both ESIDE-Depian, it is necessary to set a balanced solution that enables to manage in an uniform way both kinds of knowledge.

Therefore, ESIDE-Depian uses not only Snort information gathering capabilities, but also Snort's decision-based labelling of network traffic. Thereby, the learning processes inside ESIDE-Depian can be considered as automatically-supervised Bayesian learning, divided into the following phases. Please note that this sequence only applies for the standard generation process followed by the Packet Header Parameter Analysis experts (see Figure 2).

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 (Alípio et al., 2003)). 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.

Moreover, the creation and setting-up of each Bayesian network comprises the following phases:

- **Traffic sample obtaining.** First we need to stablish 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 [1]).
- **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

[1] Metasploit: Exploit research. http://www.metasploit.org

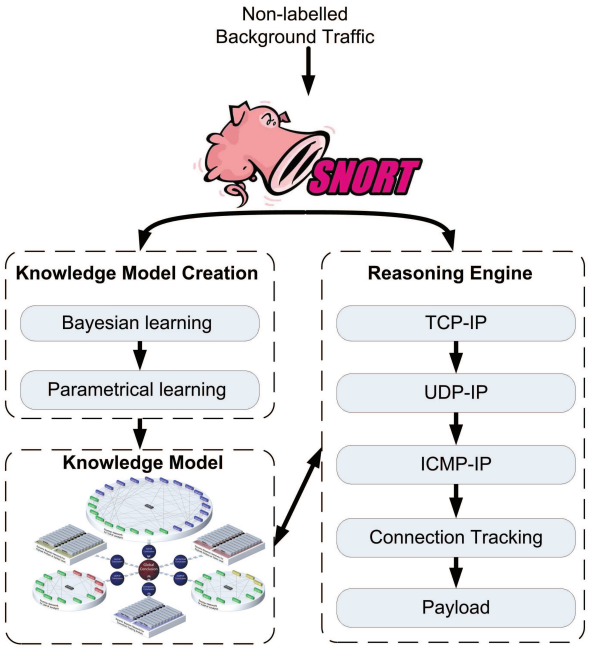

Fig. 2. ESIDE-Depian general architecture integrating misuse and anomaly detection.

Bayesian structure of nodes and edges, in order to ease the later conclusion inference over this mentioned structure.

In particular, the PC-Algorithm (Spirtes et al., 2001) is used here to achieve the structure of causal and/or correlative relationships among given variables from data. In other words, the PC-Algorithm uses the traffic sample data to define the Bayesian model, representing the whole set of dependence and independence relationships among detection parameters.

- **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.

  Specifically, ESIDE-Depian implements maximum likelihood estimate (Kjærulff & Madsen, 2008) to achieve this goal. This method completes the Bayesian model obtained in the previous step by defining the quantitative description of the set of edges between parameters. This is, structural learning finds the structure of probability distribution functions among detection parameters, and parametric learning fills this structure with proper conditional probability values.

- **Bayesian Inference.** Next, every packet capture from the target communication infrastructure needs one value for the posterior probability of a badness variable, (i.e. the Snort [2] label), given the set of observable packet detection parameters.

  Hence, we need an inference engine based on Bayesian evidence propagation. More accurately, we use the Lauritzen and Spiegelhalter method for conclusion inference over junction trees, provided it is slightly more efficient than any other in terms of response time (Castillo et al., 1996). 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. Generally, a threshold-based alarm mechanism can be added in order to get a balance between false positive and negative rates, depending on the context.

- **Adaptation.** Usually, the system operation does not keep a static on-going way, but usually presents more or less important deviations as a result of service installation or reconfiguration, deployment of new equipment, and so on.

  In order to keep the knowledge representation model updated with potential variations in the normal behaviour of the target system, ESIDE-Depian uses the general sequential/ incremental maximum likelihood estimates (Castillo et al., 1996) (in a continuous or periodical way) in order to achieve continuous adaptation of the model to potential changes in the normal behaviour of traffic.

3.3 Connection Tracking and Payload Analysis Bayesian Experts Knowledge Model Generation

The Connection Tracking expert attends to potential temporal influence among network events within TCP-based protocols (Estevez-Tapiador et al., 2003), and, therefore, it requires an structure that allows to include the concept of time (predecessor, successor) in its model. Similarly, the Payload Analysis expert, devoted to packet payload analysis, needs to model state transitions among symbols and tokens in the payload (following the strategy proposed in (Kruegel & Vigna, 2003).

Generally, Markov models are used in such contexts due to their capability to represent problems based on stochastic state transitions. Nevertheless, the Bayesian concept is even more suited since it not only includes representation of time (in an inherent manner), but also provides generalization of the classical Markov models adding features for complex characterization of states.

Specifically, the Dynamic Bayesian Network (DBN) concept is commonly recognized as a superset of Hidden Markov Models (Ghahramani, 1998), and, among other capabilities, it can represent dependence and independence relationships between parameters within one common state (i.e. in the traditional static Bayesian style), and also within different chronological states.

Therefore, ESIDE-Depian implements a fixed twonode DBN structure to emulate the Markov-Chain Model (with at least the same representational power and also the possibility to be extended in the future with further features) because full-exploded use of Bayesian concepts can remove several restrictions of Markov-based designs. For instance, it is not necessary

[2] A well-known misuse detector. Available at: http://www.snort.org

to establish the first-instance structural learning process used by the packet header analysis experts since the structure is clear in beforehand.
Moreover, according to (Estevez-Tapiador et al., 2003; Kruegel & Vigna, 2003), the introduction of an artificial parameter may ease this kind of analysis. Respectively, the Connection Tracking expert defines an artificial detection parameter, named TCP-h-flags (which is based on an arithmetical combination of TCP flags) and the Payload Analysis expert uses the symbol and token (thus, in fact, there are two Payload Analysis experts: one for token analysis and another for symbol analysis).
Finally, traffic behaviour (and so TCP flags temporal transition patterns) as well as payload protocol lexical and syntactical patterns may differ substantially depending on the sort of service provided from each specific equipment (i.e. from each different IP address and from each specific TCP destination port). To this end, ESIDE-Depian uses a multi-instance schema, with several Dynamic Bayesian Networks, one for each combination of TCP destination address and port. Afterwards, in the exploitation phase, Bayesian inference can be performed from real-time incoming network packets.
In this case, the a-priori fixed structure suggests the application of the expectation and maximization algorithm (Murphy, 2001), in order to calculate not the posterior probability of attack, but the probability which a single packet fits the learned model with.

3.4 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.
Such a Naive classifier (Castillo et al., 1996) has already been proposed in network intrusion detection, mostly for anomaly detection (Amor et al., 2004). 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. Figure 3 details the individual knowledge models and how do they fit to conform the general one.

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. In other words, 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 to the training of the Bayesian network. Second, already in working-time, Snort's opinion is passed to the experts so they can take this additional information into account.

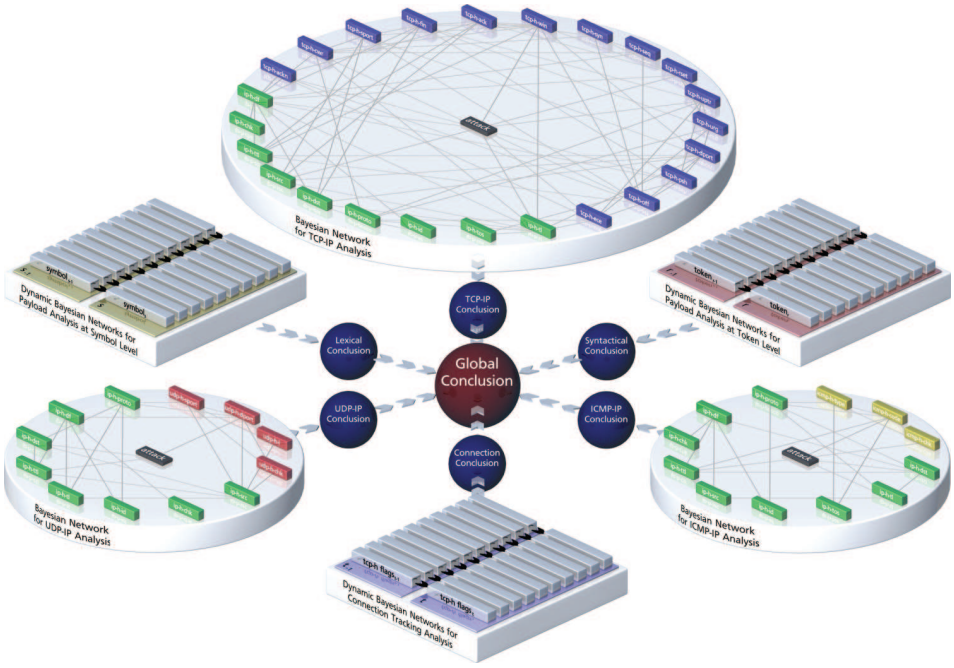

Fig. 3. ESIDE-Depian Final Knowledge Representation Model.

4.1 Snort-driven Automated Learning

The obtaining of the knowledge model in an automated manner can be achieved in an unsupervised or supervised way. In the training phase, 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.

4.2 Snort-labelled Network Traffic

Initial designs of ESIDE-Depian considered including Snort's opinion at the same level as experts' verdict in the naive Bayesian network but experiments showed that it biased the result too much. Therefore, we chose an strategy similar to the one used in the Bayesian network training (described in the previous section). Hence, already in real time, every packet gets Snort's opinion added as the badness variable mentioned before. In this way, experts know again the decision of Snort in beforehand and can act in consequence according to their knowledge model. Figure 2 illustrates how Snort is integrated within the different modules that conform ESIDE-Depian.

5. Evaluation and results

In order to asses the performance of ESIDE-Depian both as misuse and as anomaly detector, we have performed different kinds of experiments. Since Snort analyses only superficially the body of each packet, we have been forced to divide these tests into header-based and packet-body-based attacks in order to evaluate all of them more efficiently.

5.1 Header Parameter Analysis

Three are the Bayesian experts involved in this series of tests (though this does not mean that only one expert deals with the analysis; the naive Bayesian network considers all of them before obtaining the final verdict): TCP-IP, UDP-IP and ICMP-IP experts. The methodology applied intends to, first, demonstrate that the initial reference knowledge has been acquired, and second, that this reference knowledge has been superseded and exceed. In other words, we initially test the misuse detection capability and then, the anomaly detection ability.
The acquisition of the initial reference knowledge is performed already 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 (see Table 1), 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. Table 1) summarises the results of testing the initial (Snort) reference knowledge acquisition. To this end, the BN was fed with a new sample traffic merging normal one extracted from a one hour capture at the University of Deusto and also malicious packets (crafted with the tool PackIt).

Traffic type	**TCP**	**UDP**	**ICPM**
Reference knowledge good/bad traffic ratio	699,560/42	5,130/11	1,432/95
Presentation cycles required	2943	2	2
Snort's hits	38	0	450
Analysed packets	100,000	10,000	5,000
Attacks detected by Snort	5	1	600
Attacks detected by ESIDE-Depian	5 (100%)	1 (100%)	600 (100%)

Table 1. Misuse Detection Tests Analysing Packets Headers.

ESIDE-Depian shows the same performance as Snort in these tree different traffic sorts. The high number of presentation cycles required by the TCPIP expert to grasp the initial reference knowledge is due to the very high good/bad traffic ratio, much lower in the cases of UDP and ICMP. Therefore, we can conclude that gaining the reference knowledge was completed successfully. Regarding going beyond this reference knowledge (i.e. the ability of ESIDE-Depian to find zero-day attacks) we have created artificial anomalies along to the proposal of Lee et al. (2001). In this way, table 2 shows some of the TCP-IP packets that we inserted in the traffic (crafted to this end again with PackIt).
Snort was not able to detect any of them, whereas ESIDE-Depian achieved a 100% of success. Table 2 shows 15 packets labelled as potential negatives, this is, packets marked as positive (i.e., attack) by ESIDE-Depian but not by Snort. All of them correspond to the artificial anomalies we inserted and ESIDE-Depian was able to find the 100% of them. Table 3 shows some of the modified packets for the UDP-IP traffic tests

Examples of Anomalies	
`packit -nnn -s 10.12.206.2 -d 10.10.10.100 -F SFP -D 1023`	
`packit -nnn -s 10.12.206.2 -d 10.10.10.100 -F A -q 1958810375`	
`packit -nnn -s 10.12.206.2 -d 10.10.10.100 -F SAF`	
Anomaly detection results	
Potential false positives (anomalous packets)	15
Anomaly detection rate	100%

Table 2. Anomaly Detection Tests for TCP-IP Traffic.

Examples of Anomalies	
`packit -t udp -s 127.0.0.1 -d 10.10.10.2 -o 0x10 -n 1 -T ttl -S 13352 -D 21763`	
`packit -t udp -s 127.0.0.1 -d 10.10.10.2 -o 0x10 -n 0 -T ttl -S 13353 -D 21763`	
`packit -t udp -s 127.0.0.1 -d 10.10.10.2 -o 0x50 -n 0 -T ttl -S 13352 -D 21763`	
Anomaly detection results	
Potential false positives (anomalous packets)	2
Anomaly detection rate	100%

Table 3. Anomaly Detection Tests for UDP-IP Traffic.

Again, in UDP-IP traffic Snort did not discover any anomaly, as expected. The 2 false positives reflected in table 3 belong again to the artificial anomalies fed by us (and crafted with PackIt). Table tbl:table4 summarises the results obtained with ICMP-IP traffic. Similarly to the previous cases, Snort failed to detect any of the attacks, whereas the 45 false positives that appear in table 4 are exactly the anomalies introduced by us in the traffic sample.

5.2 Connection Tracking and Payload Analysis

With the goal of evaluating these analysis capabilities of ESIDE-Depian in mind, we have followed a different strategy than in the case of header parameters: Snort is mainly focused on the analysis of the latter and covers the payload inspection by applying a set of regular expressions that do not provide any useful information to the Bayesian network (basically because it presents a different morpho-syntactical structure).

Moreover, the dynamic nature of the data these experts focus on, forces this change. Therefore, we have generated a brand new traffic sample to be used in the training phase. Then, only for test purposes, we have created yet another different one with some of its packet sequences modified by means of the tool NetDude (since PackIt only allows to change packets, not sequences).

Table 5 summarises the results achieved by ESIDE-Depian for the tests focused on the connection tracking and payload analysis.

Examples of Anomalies	
`packit -i eth0 -t icmp -n 666 -s 3.3.3.3 -d 10.10.10.2`	
`packit -i eth0 -t icmp -K 0 -s 3.3.3.3 -d 10.10.10.2`	
`packit -i eth0 -t icmp -K 17 -C 0 -d 10.10.10.2`	
Anomaly detection results	
Potential false positives (anomalous packets)	45
Anomaly detection rate	100%

Table 4. Anomaly Detection Tests for ICMP-IP Traffic.

Analysis Type	Connection Tracking	Payload Analysis
Analysed network packets	226,428	2,676
Attacks contained in sample	29	158
ESIDE-Depian hits	29	158

Table 5. Connection Tracking and Payload Analysis Results.

6. 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 (Penya & Bringas, 2008).

- **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 (Estevez-Tapiador et al., 2003), so time is needed to represent predecessor and successor events. In a similar vein, the payload analysis expert must model state transitions between symbols and tokens that appear on it.

Therefore, in the same way that different tests had to be performed, we had to prepare an 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.

  Nevertheless, 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 parallelisation:** Bayesian networks require many computational resources. Hence, several 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. In addition, 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 (Lundin, 2004).

  Nevertheless, the double nature of ESIDE-Depian as anomaly and misuse detector reduces the presence of false positives to a minimum. 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.

7. Related Work

Different approaches to develop network misuse detectors include expert systems (Alípio et al., 2003), intent-specification languages (Doyle et al., 2001), intelligent agent systems (Helmer et al., 2003) or rule-induction systems (Kantzavelou & Katsikas, 1997) (in (Kabiri & Ghorbani, 2005) the reader can obtain 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 (Mukkamala et al., 2005) or diverse data-mining-based approaches Lazarevic et al. (2003). Still, there is only one attempt to bring these two strands of work together.

More specifically, in Valdes & Skinner (2000), 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
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 Naive 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.

8. Conclusions

As the use of Internet grows beyond all boundaries, the number of menaces rises to become subject of concern and increasing research. Against this, Network Intrusion Detection Systems (NIDS) monitor local networks to separate legitimate from dangerous behaviours. According to their capabilities and goals, NIDS are divided into misuse detection systems (which aim to detect well-known attacks) and anomaly detection systems (which aim to detect zero-day attacks). So far, no system to our knowledge combines advantages of both without any of their disadvantages. Moreover, the use of historical data for analysis or sequential adaptation is usually ignored, missing in this way the possibility of anticipating the behaviour of the target system.
ESIDE-Depian, a Bayesian-networks-based misuse and anomaly detection system. In another work, we detailed the composition of the Bayesian network, its training methodology and showed general performance results. Here we have focused on evaluating the integration of misuse and anomaly detection. To this end, we have adopted Snort (a well-known misuse detector) as misuse detector trainer so the Bayesian Network of five experts is able to react against both misuse and anomalies. The Bayesian experts are devoted to the analysis of different network protocol aspects and obtain the common knowledge model by means of separated Snort-driven automated learning process
Since ESIDE-Depian has passed the experiments brilliantly, it is possible to conclude that ESIDE-Depian using of Bayesian Networking concepts allows to confirm an excellent basis for paradigm unifying Network Intrusion Detection, providing not only stable Misuse Detection but also effective Anomaly Detection capabilities, with one only flexible knowledge representation model and a well-proofed inference and adaptation bunch of methods.
On the other hand, the Bayesian approach also enables to implement powerful features over it, such as Dynamic-Bayesian-Network-based full representation of time, in order to accomplish totally-characterised connection tracking and low level chronological event correlation, or explanation tracking of the inferred cause-effect reasoning processes. Furthermore, contrary to other approaches such as Neural Networks, Bayesian networks allow administrative managing of inner information structures, so specific relationships among packet detection parameters and final conclusion can be explained, in a white-box manner. Moreover, it is not only possible to recover reasoning information, but also to act on both Bayesian network

structures and conditional probability parameters, in order to adjust the whole behaviour of the Network Intrusion Detection System to special needs or configurations.
Besides, dynamic regulation of knowledge representation model can be accomplished by using the sensibility analysis proposed by Castillo et al. (1996), so as to avoid denial of service attacks, automatically enabling or disabling expert modules by means of one combined heuristic measure which considers specific throughputs and representative features. In addition, it is also possible to perform model optimization, to obtain the minimal set of representative parameters, and also the minimal set of edges among them, with the subsequent increase of the general performance.
Moreover, approximate evidence propagation methods can also be applied, in order to improve inference and adaptation time of response. Current expert models only consider exact inference, but it is possible to find methods which provide fast responses, with only a small and affordable loss of accuracy.
In addition, Bayesian knowledge representation models present one further interesting capability in current Intrusion Detection state of art, the possibility to provide an ad-hoc method for IDS evaluation. Bayesian concept provides simulation of learned knowledge corresponding samples, so it is an ideal environment for artificial anomaly generation.
At last, also unifying of Host and Network Intrusion Detection paradigms can be accomplished at low level through the Dynamic Bayesian Network concept. Specifically, both sorts of event (i.e., basically, operating system syscalls and network packets) can be characterized in one single representation model, with a dynamic approach that can obtain, for example, the posterior probability of an exploitation of one specific host service due to one specific network packet (e.g. an Unix exec syscall from a shellcode inside a packet payload). Besides, not only inference can be afforded, but even prediction of next event, due
Future work will focus on further research on exploiting the aforementioned omni-directional inference capability of Bayesian networks to the prediction of the next event, as well as on comparing ESIDE-Depian to other cutting-edge intrusion detection systems.

9. References

Abramson, B., Brown, J., Edwards, W., Murphy, A. & Winkler, R. L. (1996). Hailfinder: A Bayesian system for forecasting severe weather, *International Journal of Forecasting* **12**(1): 57–71.

Alípio, P., Carvalho, P. & Neves, J. (2003). Using CLIPS to detect network intrusions, *Lecture Notes in Computer Science* pp. 341–354.

Amor, N., Benferhat, S. & Elouedi, Z. (2004). Naive bayes vs decision trees in intrusion detection systems, *Proceedings of the 2004 ACM symposium on Applied computing*, ACM New York, NY, USA, pp. 420–424.

Bayes, T. (1763). An essay towards solving a problem in the doctrine of chances, *Philosophical Transactions of the Royal Society* **53**: 370–418.

Bringas, P., Penya, Y., Paraboschi, S. & Salvaneschi, P. (2009). Bayesian-Networks-Based Misuse and Anomaly Prevention System, *Proceedings of the Tenth International Conference on Enterprise Information Systems (ICEIS)*.

Castillo, E., Gutiérrez, J. M. & Hadi, A. S. (1996). *Expert Systems and Probabilistic Network Models*, erste edn, Springer, New York, NY, USA.

Chavan, S., Shah, K., Dave, N., Mukherjee, S., Abraham, A. & Sanyal, S. (2004). Adaptive neuro-fuzzy intrusion detection systems, *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04) Volume*, Vol. 2.

Cofiño, A. S., Cano, R., Sordo, C. & Gutiérrez, J. M. (2002). Bayesian networks for probabilistic weather prediction., *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, pp. 695–699.

Davis, G. (2006). Bayesian networks, falsification, and belief revision in accident reconstruction, *Proceedings of the Transportation Research Board* 85^{th} *Annual Meeting*.

Davis, G. & Pei, J. (2003). Bayesian networks and traffic accident reconstruction, *Proceedings of the* 9^{th} *international conference on Artificial intelligence and law (ICAIL)*, pp. 171–176.

Doyle, J., Kohane, I., Long, W., Shrobe, H. & Szolovits, P. (2001). Event recognition beyond signature and anomaly, *Proceedings of the 2001 IEEE Workshop on Information Assurance and Security*, pp. 170–174.

Estevez-Tapiador, J., Garcia-Teodoro, P. & Diaz-Verdejo, J. (2003). Stochastic protocol modeling for anomaly based network intrusion detection, *Proceedings of the* 1^{st} *IEEE International Workshop on Information Assurance (IWIAS)*, pp. 3–12.

Faour, A., Leray, P. & Eter, B. (2006). A SOM and Bayesian network architecture for alert filtering in network intrusion detection systems, *Proceedings of the International Conference on Information and Communication Technologies*, pp. 3175–3180.

Ghahramani, Z. (1998). Learning dynamic Bayesian networks, *Adaptive Processing of Sequences and Data Structures* p. 168.

Helmer, G., Wong, J., Honavar, V., Miller, L. & Wang, Y. (2003). Lightweight agents for intrusion detection, *The Journal of Systems & Software* **67**(2): 109–122.

Kabiri, P. & Ghorbani, A. (2005). Research on intrusion detection and response: A survey, *International Journal of Network Security* **1**(2): 84–102.

Kantzavelou, I. & Katsikas, S. (1997). An Attack Detection System for Secure Computer Systems-Outline of the Solution, *Computers and Security* **16**(3): 207–207.

Kim, D., Nguyen, H. & Park, J. (2005). Genetic algorithm to improve SVM based network intrusion detection system, *Advanced Information Networking and Applications, 2005. AINA 2005. 19th International Conference on*, Vol. 2.

Kjærulff, U. B. & Madsen, A. L. (2008). *Bayesian Networks and Influence Diagrams: A Guide to Construction and Analysis*, Information Science and Statistics, Springer.

Kruegel, C. (2002). *Network Alertness-Towards an adaptive, collaborating Intrusion Detection System*, PhD thesis, Technical University of Vienna.

Kruegel, C. & Vigna, G. (2003). Anomaly detection of web-based attacks, *Proceedings of the* 10^{th} *ACM conference on Computer and communications security*, ACM, pp. 251–261.

Krügel, C., Mutz, D., Robertson, W. K. & Valeur, F. (2003). Bayesian event classification for intrusion detection, *Proceedings of the* 19^{th} *Annual Computer Security Applications Conference*, pp. 14–23.

Lazarevic, A., Ertoz, L., Kumar, V., Ozgur, A. & Srivastava, J. (2003). A comparative study of anomaly detection schemes in network intrusion detection, *Proceedings of the* 3^{rd} *SIAM International Conference on Data Mining*, pp. 25–36.

Lee, W., Stolfo, S., Chan, P., Eskin, E., Fan, W., Miller, M., Hershkop, S. & Zhang, J. (2001). Real time data mining-based intrusion detection, *Proceedings of DARPA Information Survivability Conference and Exposition II*.

Liu, Y. & Li, S.-Q. (2007). Decision support for maintenance management using Bayesian networks, *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing (WiCom 2007)*, pp. 5713–5716.

Lundin, E. (2004). *Logging for intrusion and fraud detection*, PhD thesis, University of Goteborg, Department of Computer Engineering, Sweden.

Masruroh, N. A. & Poh, K. L. (2007). A Bayesian network approach to job-shop rescheduling, *Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management*, pp. 1098–1102.

Mukkamala, S., Sung, A. & Abraham, A. (2005). Intrusion detection using an ensemble of intelligent paradigms, *Journal of Network and Computer Applications* **28**(2): 167–182.

Murphy, K. (2001). An introduction to graphical models, *Technical report*.

Pearl, J. & Russell, S. (2000). Bayesian networks, *Technical Report Tech. Rep. R-216*, Computer Science Department, University of California, Los Angeles.

Penya, Y. & Bringas, P. (2008). Integrating network misuse and anomaly prevention, *Proceedings of the 6th IEEE International Conference on Industrial Informatics (INDIN)*, pp. 586–591.

Spirtes, P., Glymour, C. & Scheines, R. (2001). *Causation, prediction, and search*, The MIT Press.

Valdes, A. & Skinner, K. (2000). Adaptive, model-based monitoring for cyber attack detection, *Recent Advances in Intrusion Detection*, Springer, pp. 80–93.

Vigna, G., Eckmann, S. & Kemmerer, R. (2000). The STAT tool suite, *Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX)*.

Yang, Z., Nie, X., Xu, W. & Guo, J. (2006). An approach to spam detection by naive Bayes ensemble based on decision induction, *Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications (ISDA'06)*, IEEE Computer Society, pp. 861–866.