9

Integral Misuse and Anomaly Detection and Prevention System

Yoseba K. Penya, Igor Ruiz-Agúndez and Pablo G. Bringas
DeustoTech - University of Deusto
Basque Country

1. Introduction

Nowadays hardly anyone will dare to deny the serious security problems that computer networks must cope with. Old-fashioned techniques for isolating the network and providing a secure access control are just impotent to stop the attack flood since the production of code with harmful intentions grows not only in number but in quality as well. Network Intrusion Detection Systems (NIDS) were developed with this scenario in mind.

Historically, the first efficient methodology was *misuse detection*, consisting on recognising malicious behaviours based upon a knowledge base. This technique successes on discovering threads already registered in its database but fails when detecting new, unknown menaces. *Anomaly detection* was specifically designed to address this shortcoming. This kind of techniques model the legitimate usage of the system in order to afterwards notice, evaluate and, if applies, avoid deviations from that normal profile. Still, its efficiency decreases dramatically when handling well-known attacks, specially if compared to misuse detections systems. As the reader may note, both do flop when applied to each other's natural domain.

More in detail, misuse detection is currently 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. And, 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, after some time, 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.
In previous works (Bringas & Penya, 2008; Penya & Bringas, 2008b), we presented the first methodology to seamlessly integrate anomaly and misuse detection within one single system, giving a proper simultaneous response to either kind of attacks, unknown and well-known ones. This system uses a Bayesian Network that analyses incoming network packets learning to distinguish dangerous from harmless traffic. Moreover, we evaluate it against well-known and new attacks showing how it outperforms a well-established industrial NIDS.
Nevertheless, the training process of the Bayesian network is the Achilles' heel of this approach. Basically, that phase is devoted to create a knowledge representation model combining misuse and anomaly-based data that will be used later on and it may become intractable very fast in some extreme situations. Against this drawback, we propose the use of expert knowledge to enhance and optimise the design of the NIDS, shortening subsequently the training process. This expert knowledge is represented as a set of hypotheses that must be verified to justify their utility. In this way, we have tested our approach with several samples of data showing that all the hypotheses assumed were true and, therefore, that the proposed methodology to trim down the design and training processes yields an optimal Bayesian network for Intrusion Detection.
Finally, there are a number of problems and difficulties that arose in the integration process that we solved as well; this work will also describe them as well. In summary, this article presents a comprehensive survey of our work integrating misuse and anomaly prevention detection in one single system (Bringas & Penya, 2008; Penya & Bringas, 2008b; Bringas & Penya, 2009; Penya & Bringas, 2008a; Ruiz-Agundez et al., 2010).
The rest of the article is organised as follows. In section 2, we introduce the general architecture of the system to depict the big picture. In section 3, we detail the Bayesian Network, including its entire obtaining process. In section 4, we discuss the results obtained. In section 5, we present the hypotheses assumed and demonstrate them true. Section 6 is devoted to explain problems experimented and the solution adopted, and, finally, section 7 concludes and draws the avenues of future works.

2. Architecture

The internal design of ESIDE-Depian, our integrated anomaly and misuse detector system 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. In order to ease the way to this goal, ESIDE-Depian has been conceived and deployed in a modular way that allows decomposing of the problem into several smaller units. Thereby, Snort[1] (a rule-based state of the art Misuse Detection System, see (Roesch, 1999), has been integrated to improve the training procedure to increase the accuracy of ESIDE-Depian. Following a strategy proven successful in this area, (Alipio et al., 2003), the reasoning engine we present here is composed of a number of Bayesian experts working over a common knowledge representation model.
The Bayesian experts must cover all possible areas where a menace may rise. In this way, there are 5 Bayesian experts in ESIDE-Depian, as follows: 3 of them deal with packet headers of TCP, UDP, ICMP and IP network protocols, the so-called TCP-IP, UDP-IP and ICMP-IP

[1] Available at http://www.snort.org/

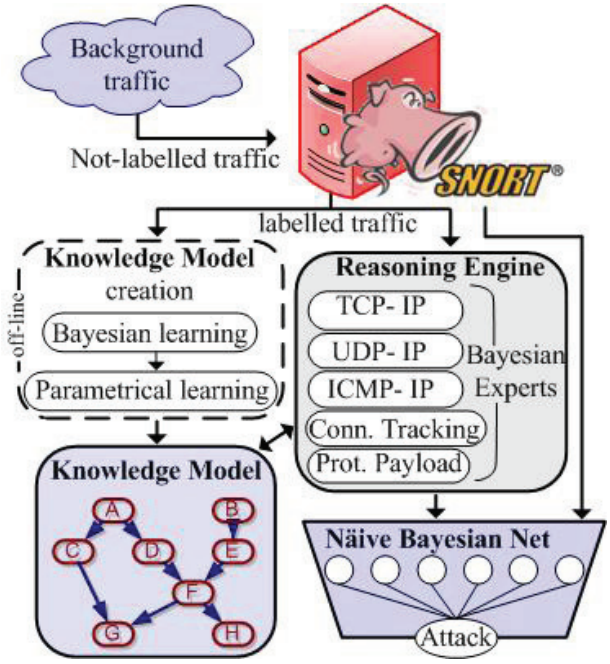

Fig. 1. ESIDE-Depian general architecture

expert modules. A further one, the Connection Tracking Expert, analyses potential temporal dependencies between TCP network events and, finally, the Protocol Payload Expert in charge of the packet payload analysis. In order to obtain the knowledge representation model, each expert carries out separately a Snort-driven supervised learning process on its expertise area. Therefore, the final knowledge representation model is the sum of the individual ones obtained by each expert. Fig. 1 shows the general ESIDE-Depian architecture.

3. Bayesian-network-based IDS

The obtaining of the knowledge representation model in an automated manner can be achieved in an unsupervised or supervised way. Typically, unsupervised learning approaches don't have 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. Thus, 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 network traffic labelling. 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, (i.e. the TCP-IP, UDP-IP and ICMP-IP expert modules):

- *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 such as (Metasploit, 2006), etc. Subsequently, the Snort Intrusion Detection System embedded in ESIDE-Depian adds labelling 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 and also 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.
- *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. 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. Due to its high requirements in terms of computational and temporal resources, this phase is usually performed off-line.
- *Parametric Learning*: The knowledge representation 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 (Murphy, 2001) 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. The high complexity of this phase suggests a deeper description, that is accomplished in section 3.
- *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 label), given the set of observable packet detection parameters. So, 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., 1997). 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*: Normally, 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 (Murphy, 2001) (in a continuous or periodical way) in order to achieve continuous adaptation of the model to potential changes in the normal behaviour of traffic.

3.1 Connection tracking and payload analysis Bayesian experts knowledge representation 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 a 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). Usually, 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 generalisation of the classical Markov models adding features for complex characterisation of states. Specifically, the Dynamic Bayesian Network (DBN) concept is commonly recognised 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.

Thus, ESIDE-Depian implements a fixed two-node 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 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) and (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 header flags) and the Payload Analysis expert uses the symbol and token (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 maximisation 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.

3.2 Naive Bayesian network of the expert modules

Having different Bayesian modules is a two-fold 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, a unique measure must emerge from the diverse judgements.

To this end, ESIDE-Depian presents a two-tiered schema where the first layer is composed of the results from the expert modules and the second layer includes only one class parameter: the most conservative response among those provided by Snort and the expert modules community (i.e. in order to prioritises the absence of false negatives in front of false positives). Thus, both layers form, in fact, a Naive Bayesian Network (as shown in Fig. 1 and Fig. 2).

Such a Naive classifier (Castillo et al., 1997) has been proposed sometimes in Network Intrusion Detection, mostly for Anomaly Detection. 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, in order to support heavy load conditions derived e.g. from denial of service attacks.

Now, Naive Bayesian Network parameters should have a discrete nature which, depending on the expert, could not be the case. To remove this problem, ESIDE-Depian allows the using of the aforementioned set of administratively-configured threshold conditioning functions.

Finally, the structure of the Naive Bayesian Network model is fixed in beforehand, assuming the existence of conditional independence hypothesis among every possible cause and the standing of dependency edges between these causes and the effect or class. Therefore, here is also not necessary to take into consideration any structural learning process for it; only sequential parametric learning must be performed, while the expert modules produce their packet classifying verdicts during their respective parametric learning stages.

Once this step is accomplished, the inference of unified conclusions and the sequential adaptation of knowledge can be provided in the same way mentioned before. Fig. 2 details the individual knowledge models and how do they fit to conform the general one.

3.3 The structural learning challenge

As it was outlined before, Structural Learning allows modelling in a completely automated way the set of dependence and independence relationships that can exist between the different detection parameters. Nevertheless, in situations that have a large volume of evidences and detection parameters with many different possible states, the aforementioned PC Algorithm (as well as similar alternative methods) presents very high computational requirements. Moreover, depending on the inner complexity of the set of relationships, those requirements can grow even more. Therefore, the complexity depends entirely on the nature of data, rendering it unpredictable so far. In this way, this task may be sometimes too resource-demanding for small and medium computing platforms. Against this background, we have developed a method that splits the traffic sample horizontally in order to reduce the complexity the structural learning process. This method is detailed next.

First of all, please note that the different structural learning methods commonly use a significance parameter in order to define in a flexible manner the strength that a dependence relationship needs in fact to be considered as a definitive one in the Bayesian Network. Thus, this significance parameter can be used to relativise the concept of equality required in the independence tests implemented inside the learning algorithms; in particular, inside the PC Algorithm. On the one hand, a high significance value produces a higher number

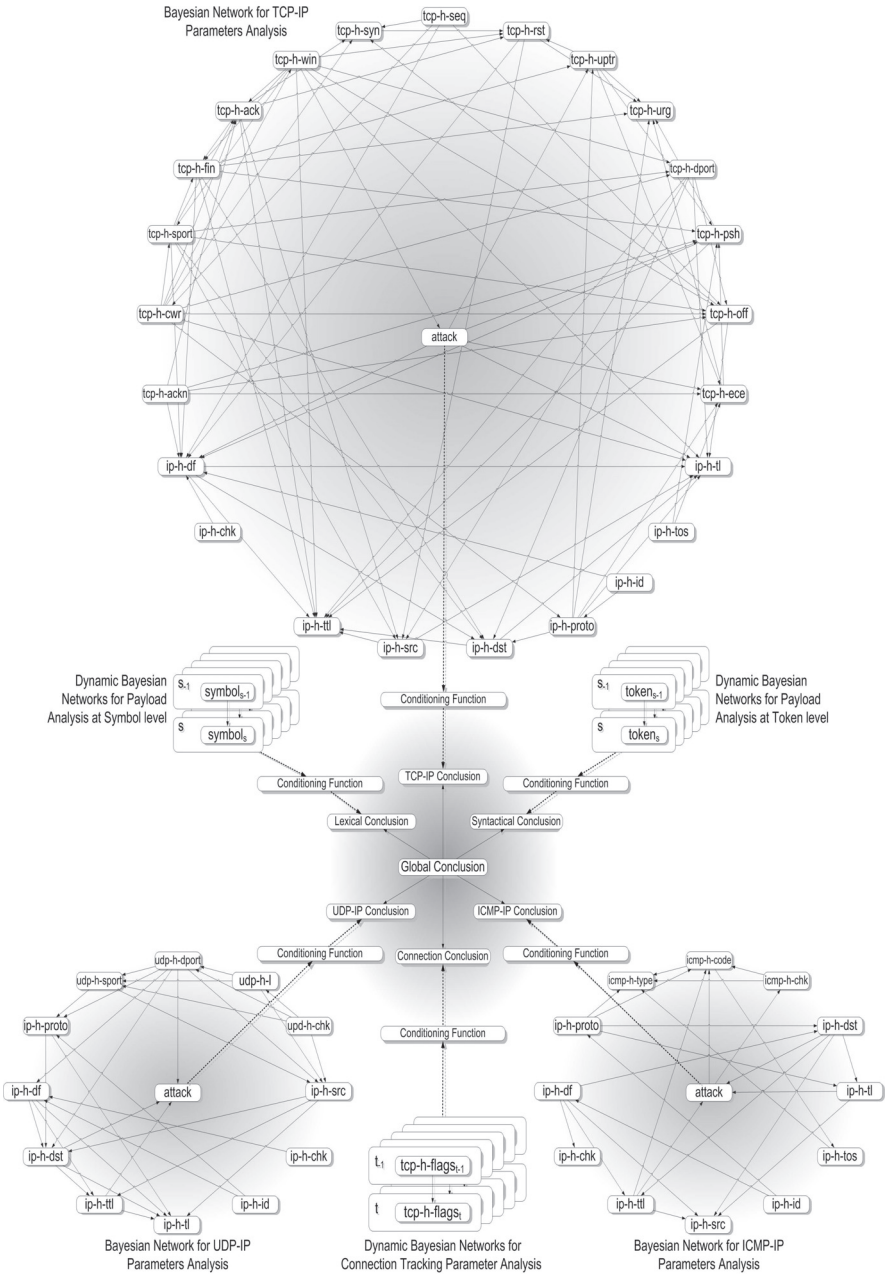

Fig. 2. ESIDE-Depian general architecture

of connections in the Bayesian model, with an increase on the degree of representativeness but also with larger requirements in terms of main memory and computational power and the possibility that overfitting occurs. On the other hand, a low significance value usually produces a sparse Bayesian Network, with lower requirements but also less semantic power. Therefore, in order to achieve a trade-off between representativeness and throughput, we have implemented the expansion of the structural learning process through multiple significance levels. More accurately, we have used the seven most representative magnitude-orders.
Once the expansion process is planned, it is possible to proceed with the structural learning process itself, applied in this case to the TCP-IP, UDP-IP and ICMP-IP protocols, which were defined a priori as a set of working dependence and independence hypotheses based on each different RFC. It is also possible to apply the PC Algorithm to the entire traffic sample in order to verify the accuracy of these hypotheses. In our case, this process was estimated to long 3,591 hours (150 days) of continuous learning. Thus, and also considering that this is a generalised problem, we propose the parallelisation of the collection of learning processes, depending on the equipment available for the researcher. In fact, this parallelisation, performed over 60 computers at the same time managed to reduce the overall time to 59.85 nominal hours. Finally, one instance of the PC Algorithm was applied to every fragment, for each of the four data-sets, and with the seven different significance values, 1197 partial Bayesian structures were obtained at this point.

3.4 Partial Bayesian structures unifying process

The collection of partial Bayesian structures obtained in the previous phase must be unified into an unique Bayesian Network that gives a full representation of the application domain. With this purpose, we defined a statistical measure based on the frequency of apparition in the set of partial structures of each dependence relationship between every two detection parameters.
In this way, it is possible to calculate one single Bayesian structure, for each of the 7 significance levels and for each of the 4 data sets, remaining only 28 partial structures from the initial 1197. The next step will achieve the definitive unifying. To this end, we use the average value for each of the edges between variables, which allows us to reach the desired balance between representativeness and throughput by means of the subsequent selective addition to the Bayesian model of those edges above a specific and configurable (depending on the application domain) significance threshold.
Still, both the horizontal splitting of the traffic sample and also the significance-based expansion present an important problem: the cycles that can appear in the Bayesian model obtained after unifying, which render the model invalid. An additional step prevents such cycles by using the average value obtained before as a criteria for the selection of the weakest edge in the cycle, which is the one to be deleted.

4. Experiments and results

In order to measure the performance of ESIDE-Depian, we have designed two different kinds of experiments. In the first group, the network suffers well-known attacks (i.e. Misuse Detection) and, in the second group, zero-day attacks (i.e. Anomaly Detection), putting each aspect of the double nature of ESIDE-Depian to the test. In both cases, the system was fed with a simulation of network traffic comprising more than 700.000 network packets that were sniffed during one-hour capture from a University network. The first experiment (corresponding to Misuse Detection) aimed to compare Snort and the Packet

Indicator	TCP	UDP	ICMP
Analysed net packets	699.568	5.130	1.432
Snort's hits	38	0	450
ESIDE-Depian's hits	38	0	450
Anomalous packets	600	2	45
False negatives	0	0	0
Potential false positives	0,08%	0,03%	3,14%

Table 1. Bayesian expert modules for TCP, UDP and ICMP header analysis results.

Header Parameters Analysis experts. To this end, Snort's rule-set-based knowledge was used as the main reference for the labelling process, instantiated through Sneeze Snort-stimulator (Roesch, 1999). The sample analysed was a mixture of normal and poisoned traffic. Table 1 details the results of this experiment.

As it can be seen, the three experts achieved a 100% rate of hitting success. Anyway, such results are not surprising, since ESIDE-Depian integrates Snort's knowledge and if Snort is able to detect an attack, ESIDE-Depian should do so. Nevertheless, not only the number of hits is important; the number of anomalous packets detected reflects the level of integration between the anomaly and the misuse detection part of ESIDE-Depian. In fact, the latter can be highlighted as the most important achievement of ESIDE-Depian: detecting unusual packets preserving the misuse detection advantages at the same time. Concerning potential false rates, it is possible to observe that very good rates are reached for TCP and UDP protocols (according to the values defined in (Crothers, 2002) to be not human-operator-exhausting), but not so good for ICMP. Table 1 shows, however, a significant bias in the number of attacks introduced in the ICMP traffic sample (above 30%), and labelled as so by Snort; thus, it is not strange the slightly excessive rate of anomalous packets detected here by ESIDE-Depian.

In the second experiment (also corresponding to misuse detection), the goal was to test the other expert modules (Connection Tracking and Payload Analysis). With this objective in mind, a set of attacks against a representation of popular services were fired through several hacking tools such as (Metasploit, 2006). The outcome of this test is summarised in Table 2.

Indicator	Conn. Tracking	Payload Analysis (Symbol)	Payload Analysis (Token)
Analysed net packets	226.428	2.676	2.676
Attacks in sample	29	139	19
ESIDE Depian hits	29	139	19
Anomalous net packets	0	0	3
False negatives	0	0	0
Pot. false positives	0%	0%	0,11%

Table 2. Bayesian expert modules for TCP, UDP and ICMP header analysis results.

Protocol	Artificial Network Anomaly	Snort	ESIDE-Depian
TCP	packit -nnn -s 10.12.206.2 -d 10.10.10.100 -F SFP -D 1023	x	$\surd$
TCP	packit nnn -s 10.12.206.2 -d 10.10.10.100 -F SAF	x	$\surd$
UDP	packit -t udp -s 127.0.0.1 -d 10.10.10.2 -o 0x10 -n 1 -T ttl -S 13352 -D 21763	x	$\surd$
UDP	packit -t udp -s 127.0.0.1 -d 10.10.10.2 -o 0x50 -n 0 -T ttl -S 13352 -D 21763	x	$\surd$
ICMP	packit -i eth0 -t icmp -K 17 -C 0 -d 10.10.10.2	x	$\surd$

Table 3. Example of Zero-day attacks detected by ESIDE-Depian and not by Snort.

As we see, ESIDE-Depian prevailed in all cases with a 0% rate of false negatives and a 100% of hitting rate success. Still, not only Snort's knowledge and normal traffic behaviour absorption was tested; the third experiment intended to assess ESIDE-Depian's performance with zero-day attacks. With this idea in mind, a sample of artificial anomalies (Lee et al., 2001) was prepared with Snort's rule set as basis and crafted (by means of the tool Packit) with slight variations aiming to avoid Snort's detection (i.e. Zero-day attacks unnoticeable for misuse detection systems). Some of these attacks are detailed next in Table 3.
Note that overcoming of Snort's expert knowledge only has sense in those expert modules using this knowledge. This is, in protocol header specialised modules, because the semantics of Snort's labelling does not fit the morphology of payload and dynamic nature parameters.

5. Optimal knowledge base design

As we have shown, ESIDE-Depian is able to simultaneously offer efficient response against both well-known and zero-day attacks. In order to ease the way to this goal, our system has been conceived and deployed in a modular way that allows decomposing of the problem into several smaller units (see section 2). Still, the design and training process of the Bayesian network (BN) demanded huge computational efforts that prevented it from being applied in the real world. Against this background, in this section we present a methodology to enhance the design of the BN (and, thus, shorten the training process) by using expert knowledge. This strategy has been previously used in data mining classifiers (Sinha & Zhao, 2008), for normalizing and binning gene expression data in BN (Helman et al., 2004) or for generalized partition testing via Bayes linear methods (Coolen et al., 2001).
As already introduced in section 3, Bayesian networks require two learning steps to be ready to infer results. The first one is the *structural learning* process that obtains the probability distribution table associated to each variable. The second one is the *parametric learning* process that refines the initial graph. Finally, the system uses a Naive Bayesian network to unify the different experts providing an excellent balance between knowledge representation capacity and performance. It assumes the existence of *conditional independence* hypotheses within every possible cause and the standing of dependency edges between these causes and the effect or class applicable to this problem domain. These hypotheses are the representation of the experts knowledge that tunes the Bayesian network design and training, creating the optimal

network.

5.1 Establishing the hypothesis of dependence and independence

Due to the fact the structural learning methods are able to infer by themselves the relations of dependence and independence of the structure, expert knowledge can refine the resulting model and, hence, the exploitation of the Bayesian network is done in the most efficient way. In this way, we use hypothesis of dependence and independence to refine our knowledge representation model. In particular, the hypotheses are based on the specific issues of four network protocols (IP, ICMP, TCP and UDP). The expert knowledge is based on the following six hypotheses of dependence and independence:

Hypothesis 1: Dependence between TCP and IP. The set of the detection parameters of the TCP protocol is dependent of the set of the detection parameters of the IP, and vice versa.

Hypothesis 2: Dependence between UDP and IP. The set of the detection parameters of the UDP protocol is dependent of the set of the detection parameters of the IP, and vice versa.

Hypothesis 3: Dependence between ICMP and IP. The set of the detection parameters of the ICMP protocol is dependent of the set of the detection parameters of the IP, and vice versa.

Hypothesis 4: Independence between TCP and UDP. The set of the detection parameters of the TCP protocol is dependent of the set of the detection parameters of the UDP, and vice versa.

Hypothesis 5: Independence between TCP and ICMP. The set of the detection parameters of the TCP protocol is independent of the set of the detection parameters of the ICMP, and vice versa.

Hypothesis 6: Independence between UDP and ICMP. The set of the detection parameters of the UDP protocol is independent of the set of the detection parameters of the ICMP, and vice versa.

These hypotheses are supported by the respective set of *Request For Comments (RFC)* of each protocol. An empirical demonstration of the hypotheses is done demonstrating that the knowledge representation model generated from them can be successfully used in the reasoning engine.

Besides, the heterogeneity of the detection parameters headers (information used by the protocols), and data (information used by the users) themselves implies a different formalization for each case. The analysis model is static (based on Bayesian networks) in the case of the head parameters and dynamic (based on Dynamic Bayesian networks) in the case of data parameters. The first group forms an information entity and it is, therefore, susceptible of being used directly in the process of analysis. On the other hand, the second group represents a variable flow of information entities in both lexical and a syntactic levels that requires a specific analysis. Considering all this, another hypothesis is pointed out:

Hypothesis 7: Independence between head and data fields. The set of detection parameters corresponding to the head fields of IP, ICMP, TCP and UDP protocols is independent from the data fields of the corresponding protocols, and vice versa.

Finally, there is an additional aspect to consider. Since for this experiment only one time step is considered, it is not possible to have the second evidence required by the dynamic model. Please note that when more temporal steps are added this restriction disappears.

Hypothesis 8: Independence between static and dynamic parameters. In the case of one temporal step Dynamic Bayesian networks, the set of detection parameters used in the static analysis methods are independent from those used in the dynamic analysis methods, and vice versa.

The specification of the previous hypotheses of dependence and independence defines separated working areas, in which different analysis methods will be applied depending on the type of the detection parameter.
On one hand, we have the head parameters of all the protocols that can be treated in a homogeneous way. These cases can be introduced straightforward into the structural learning process. On the other hand, each protocol data has its own properties and therefore has to be resolved in an independent way. In the case of dynamic parameters, multiple evidences are required, and hence, they will have an independent treatment too.

5.2 Validation of the hypotheses

In order to assess the validity of the work hypothesis described in the previous section, we have performed different kinds of experiments. We start our experiments from the knowledge representation model made up of different Bayesian networks that form the reasoning engine. From then on, and considering the hypotheses of dependence and independence, we analyse the obtained results of the structural learning process. As the results confirm the hypotheses of dependence and independence, the *RFC* specifications of each protocol are ratified. Finally, a knowledge representation model based on the different Bayesian networks can be built.
Taking that into account, and with the objective of minimising the possible appearance of noise in the results that could affect the final conclusions about the hypothesis, we have set a threshold value. Above this threshold value a link between two variables will not be representative. According to our methodology, two protocols will be independent *if and only if* there are no representative relations between the set of parameters of either of them. The hypotheses of dependence and independence are proved to be true:

- **Hypothesis 1: Dependence between TCP and IP.** Table 4 shows the relations between the parameters of TCP and IP, verifying that there are many significant links between the corresponding detection parameters of each protocol, in both ways. Therefore, there are variables of the TCP protocol Bayesian network that depend on variables of the IP protocol, and vice versa. Hence, the hypothesis of dependence between TCP and IP is confirmed.

- **Hypothesis 2: Dependence between UDP and IP.** Equally, as table 5 show the data of the experiment points out the relation between the head parameters of the UDP and IP protocols. There are enough significant links between the detection parameters of both protocols, in both ways. Therefore, there are variables of the UDP protocol in the Bayesian network that depend on variables of the IP protocol, and vice versa. Hence, the hypothesis of dependence between UDP and IP is also confirmed.

- **Hypothesis 3: Dependence between ICMP and IP.** Table 6 show the case of ICMP and IP protocols, the data of the experiment points out the relation between the head parameters of both protocols. There are enough significant links between the detection parameters, in both ways. Therefore, there are variables of the ICMP protocol in the Bayesian network that depend on variables of the IP protocol, and vice versa. Hence, the hypothesis of dependence between ICMP and IP is confirmed similarly.

child\parent	ip-h-dst	ip-h-src	ip-h-proto	ip-h-tll	ip-h-df	ip-h-tl	ip-h-tos
tcp-h-uptr	-	-	15,61	-	-	-	-
tcp-h-win	2,40	2,09	-	2,89	4,46	-	13,55
tcp-h-cwr	-	0,49	-	-	-	31,32	-
tcp-h-ece	-	-	11,08	-	-	-	2,33
tcp-h-psh	-	-	6,63	0,71	-	-	1,38
tcp-h-urg	-	-	12,43	-	-	-	-
tcp-h-ack	-	-	-	-	-	-	2,17
tcp-h-rst	-	-	1,31	1,08	1,92	-	-
tcp-h-syn	0,71	-	-	-	1,79	-	-
tcp-h-fin	-	-	-	1,02	-	-	-
tcp-h-off	-	-	-	1,98	-	-	28,74
tcp-h-ackn	-	-	-	-	-	-	-
tcp-h-seq	-	-	-	-	1,74	-	-
tcp-h-dport	3,84	1,08	-	-	-	9,04	-
tcp-h-sport	8,01	3,57	-	-	-	8,40	-

Table 4. Frequency of links appearance in the Bayesian network for relations of dependence of IP parameters over TCP parameters

– **Hypothesis 4: Independence between TCP and UDP.** Table 7 show the hypothesis of independence between TCP and UDP, the data of the experiment points out the independence between the detection parameters of both protocols, in none of both ways. There are not enough significant links between the detection parameters, in none of both ways. Therefore, there are not variables of the TCP protocol that depend on the variables of the UDP protocol, and vice versa. Hence, the hypothesis of independence between TCP and UDP is also verified.

– **Hypothesis 5: Independence between TCP and ICMP.** Similarly, table 8 shows that the data of the experiment points out the independence between the detection parameters of TCP and ICMP protocols, in any way. There are not enough significant links between the detection parameters, in any way. Therefore, there are not variables of the TCP protocol that depend on the variables of the ICMP protocol, and vice versa. Hence, the hypothesis of independence between TCP and ICMP is also proved.

child\parent	udp-h-chk	udp-h-l	udp-h-dport	udp-h-sport
ip-h-dst	-	-	1,61	-
ip-h-src	9,65	1,02	3,88	1,79
ip-h-proto	-	-	2,04	3,65
ip-h-tll	-	-	-	-
ip-h-df	-	-	1,02	-
ip-h-id	-	-	-	-
ip-h-tl	-	-	-	-
ip-h-tos	-	-	-	-
ip-h-hl	-	-	-	-

Table 5. Frequency of links appearance in the Bayesian network for relations of dependence of UDP parameters over IP parameters

child\parent	icmp-hchk	icmp-hcode	icmp-htype
ip-h-dst	-	-	-
ip-h-src	-	-	-
ip-h-proto	-	2,90	-
ip-h-tll	-	-	-
ip-h-df	-	-	-
ip-h-id	-	-	-
ip-h-tl	-	-	-
ip-h-tos	-	8,94	-
ip-h-hl	-	-	-

Table 6. Frequency of links appearance in the Bayesian network for relations of dependence of ICMP parameters over IP parameters

– **Hypothesis 6: Independence between UDP and ICMP.** Finally, in table 9 and table 10, the data of the experiment points out the independence between the detection parameters of UDP and ICMP protocols, in anyway. There are not enough significant links between the detection parameters, in none of both ways. Therefore, there are not variables of the UDP protocol that depend on the variables of the ICMP protocol, and vice versa. Hence, the hypothesis of independence between UDP and ICMP is also verified.

The validity of the hypotheses 7 and 8 is already proved by their set out. This is, the set of detection parameters corresponding to the head fields of IP, ICMP, TCP and UDP protocols is independent from the data fields of the corresponding protocols, and vice versa. In the case of one temporal step dynamic BN, the set of detection parameters used in the static analysis methods are independent from those used in the dynamic analysis methods, and vice versa. Since the results confirm the hypothesis of dependence and independence, the RFC specifications of each protocol are ratified. Therefore, a knowledge representation model

child\parent	udp-hchk	udp-h-l	udp-h-dport	udp-h-sport
tcp-h-uptr	0,49	-	2,86	-
tcp-h-win	-	-	-	-
tcp-h-cwr	-	-	-	-
tcp-h-ece	0,59	-	2,63	1,79
tcp-h-psh	-	1,79	1,79	0,59
tcp-h-urg	1,02	-	1,57	4,11
tcp-h-ack	-	-	-	-
tcp-h-rst	-	-	-	1,79
tcp-h-syn	-	-	-	-
tcp-h-fin	-	-	-	-
tcp-h-off	-	-	-	-
tcp-h-ackn	-	-	-	-
tcp-h-seq	-	-	-	-
tcp-h-dport	-	-	-	-
tcp-h-sport	-	-	-	-

Table 7. Frequency of links appearance in the Bayesian network for relations of dependence of UDP parameters over TCP parameters

child\parent	icmp-hchk	icmp-hcode	icmp-htype
tcp-h-uptr	-	2,86	-
tcp-h-win	-	-	-
tcp-h-cwr	-	-	-
tcp-h-ece	-	-	-
tcp-h-psh	-	0,59	-
tcp-h-urg	-	3,52	-
tcp-h-ack	-	-	-
tcp-h-rst	-	-	-
tcp-h-syn	-	-	-
tcp-h-fin	-	0,49	-
tcp-h-off	-	-	-
tcp-h-ackn	-	-	-
tcp-h-seq	-	-	-
tcp-h-dport	-	-	-
tcp-h-sport	-	-	-

Table 8. Frequency of links appearance in the Bayesian network for relations of dependence of ICMP parameters over TCP parameters

based on the different Bayesian networks can be built, decreasing in this way the complexity of the design of the BN and minimising its training process.

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 and manage to exceed Snort's knowledge (Penya & Bringas, 2008b).

– **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

child\parent	icmp-h-chk	icmp-h-code	icmp-h-type
udp-h-chk	-	-	-
udp-h-l	-	-	-
udp-h-dport	-	-	-
udp-h-sport	-	-	-

Table 9. Frequency of links appearance in the Bayesian network for relations of dependence of ICMP parameters over UDP parameters

child\parent	udp-h-chk	udp-h-l	udp-h-dport	udp-h-sport
icmp-h-chk	-	-	-	-
icmp-h-code	-	-	-	-
icmp-h-type	-	-	-	-

Table 10. Frequency of links appearance in the Bayesian network for relations of dependence of UDP parameters over ICMP parameters

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, 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 (Penya & Bringas, 2008b), 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*. 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. In this way, the TCP expert required 2943 presentation cycles, the UDP expert 2 and the ICMP expert also 2. The high number of presentation cycles required by the TCP-IP 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.
– **Task parallelisation**: 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. Nevertheless, the double nature of ESIDE-Depian as anomaly and misuse detector reduces the presence of false positives to a minimum (Penya & Bringas, 2008b). 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. Conclusions and future works

Network Intrusion Detection Systems 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.

Our system addresses both needs. We present here an approach integrating Snort 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. A naive Bayesian network integrates the results of the experts, all the partial verdicts achieved by them. 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 only one flexible knowledge representation model and a well-proofed inference and adaptation bunch of methods.

Moreover, we have accurately studied how to create a model of knowledge representation. First of all, we obtained a representative data sample. Second, we defined how many temporal steps we were going to use for our experiment. Third, we established the hypothesis according to the expert knowledge. Fourth, we planned the process of structural learning and performed it. After this step, we obtained statistical metrics from the partial Bayesian networks. These partial fragments were unified and adapted before verifying the hypotheses of dependence and independence. Finally, we obtained the optimal structural definition of the knowledge representation model on which we performed parametric learning. According to this experiment, we have proved the validity of the hypotheses and obtained the optimal BN for Network Intrusion Detection Systems. This knowledge model is currently being used as the expert system of our own IDS architecture.

It is worth mentioning that 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. Therefore, further work will be concentrated on how to extrapolate the techniques developed to the very special case we deal here with. Furthermore, we will research on the use of expert knowledge for Bayesian networks modelling over different domains beyond the Intrusion Detection and the creation of a formal metric. This metric will measure the impact of the use of expert knowledge in the model creation time and the final performance of a Bayesian network.

8. References

Alipio, P., Carvalho, P. & Neves, J. (2003). *Using CLIPS to Detect Network Intrusion*, Vol. 2902/2003, Springer-Verlag.

Bringas, P. G. & Penya, Y. (2008). Bayesian-networks-based misuse and anomalies detection system, *Proceedings of the 10th International Conference on Enterprise Information Systems (ICEIS)*, pp. 12–16.

Bringas, P. G. & Penya, Y. (2009). Next-generation misuse and anomaly prevention system, *in* J. Filipe & J. Cordeiro (eds), *LNBIP 19 (Lecture Notes in Business Information Processing)*, Springer-Verlag, Heidelberg Berlin, pp. 117–129.

Castillo, E., Gutierrez, J. M. & Hadi, A. S. (1997). *Expert Systems and Probabilistic Network Models*, Springer-Verlag.

Coolen, F., Goldstein, M. & Munro, M. (2001). Generalized partition testing via Bayes linear methods, *INFORMATION AND SOFTWARE TECHNOLOGY* 43(13): 783–793.

Crothers, T. (2002). *Implementing Intrusion Detection Systems: A Hands-On Guide for Securing the Network*, John Whiley & Sons Inc.

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

Ghahramani, Z. (1998). *Learning Dynamic Bayesian Networks*, Vol. 1387, Springer-Verlag.

Helman, P., Veroff, R., Atlas, S. & Willman, C. (2004). A Bayesian network classification methodology for gene expression data, *JOURNAL OF COMPUTATIONAL BIOLOGY* 11(4): 581–615.

Kruegel, C. & Vigna, G. (2003). Anomaly detection of web-based attacks, *Proceedings of the 10th ACM Conference on Computer and Communications Security*, pp. 251–261.

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 the second DARPA Information Survivability Conference and Exposition*, pp. 85–100.

Metasploit (2006). Exploit research. Available at http://www.metasploit.org/.

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

Penya, Y. & Bringas, P. G. (2008a). Experiences on designing an integral intrusion detection system, *Flexible Database and Information System Technology Workshop (FlexDBIST), at the DEXA-08 International Conference on Database and Expert Systems Applications*, pp. 675–679.

Penya, Y. & Bringas, P. G. (2008b). Integrating network misuse and anomaly prevention, *Proceedings of the 6th IEEE International Conference on Industrial Informatics (INDIN'08)*, pp. 586–591.

Roesch, M. (1999). SNORT: Lightweight intrusion detection for networks, *Proceedings of LISA99: 13th Systems Administration Conference*, pp. 229–238.

Ruiz-Agundez, I., Penya, Y. & Bringas, P. (2010). Expert knowledge for optimal bayesian network-based intrusion detection, *Proceedings of the 3rd International Conference on Human System Interaction (HSI 2010)*, pp. 444–451.

Sinha, A. R. & Zhao, H. (2008). Incorporating domain knowledge into data mining classifiers: An application in indirect lending, *DECISION SUPPORT SYSTEMS* 46(1): 287–299.

Spirtes, P., Glymour, C. & Scheines, R. (2001). *Causation, Prediction, and Search, Second Edition (Adaptive Computation and Machine Learning)*, The MIT Press.