INTRUSION DETECTION FOR WEB APPLICATIONS (SHORT VERSION)

Nathalie Dagorn
Laboratory of Algorithmic, Cryptology and Security (LACS), University of Luxembourg,
162a Avenue de la Faïencerie, Luxembourg, Luxembourg

Keywords: Intrusion detection, anomaly detection, Web attack, false positive, Bayesian network, alarm clustering.

Abstract: Intrusion detection systems (IDS) are usually classified into two categories: misuse- and anomaly detection systems. *Misuse detection* is based on signatures; it is precise but can only accommodate already known attacks. Unlike this, *anomaly detection* models a system's usual behavior and is able to detect new attacks, but some major challenges remain to be solved in this field, in particular the improvement of the detection process and the reduction of false alarms. On the application/service level, several misuse detection systems exist and work, but only one anomaly detection system is known to be efficient for now. In this short paper, we propose a Web learning-based anomaly detection system based on this system, and resulting from the junction of academic research in several fields, which we improved. The system analyzes HTTP requests as logged by most of the Web servers; it exclusively relates to the queries containing attributes. The analysis process implements a multi-model statistical approach. A Bayesian network is used as decision process, specifying six states (one normal state and five attack states) at the classification node. The system is improved after each log analysis thanks to a technique of alarm clustering, which allows filtering false positive. Compared to traditional anomaly detection systems, the system we present globally gains in *sensitivity* (each step of the process reduces the number of false positive to be dealt with) and in *specificity* (if an attack is detected, its type is immediately specified). Moreover, a co-operation feature (alarm correlation) with other systems is proposed for distributed intrusion detection. To date, the system has only been partially implemented but the preliminary experiments in real environment show encouraging results.

1 INTRODUCTION

Intrusion detection systems (IDS) usually implement two techniques: misuse- and anomaly detection. *Misuse detection* is based on signatures: the system analyzes information collected in the traffic for comparison to a database of signatures of known attacks, and each matching activity is considered as an attack. Unfortunately, misuse detection can only accommodate already documented attacks. Unlike this, *anomaly detection* models a system's usual behavior and any significant deviation from the defined baseline is considered as the result of an attack. Anomaly-based systems have the advantage of being able to detect previously unknown attacks; however, they are not as effective as misuse detection systems for detecting known attacks. The major challenges to be solved in this field are the improvement of the detection process and the reduction of false alarms. On the application/service level, several misuse detection systems exist and work, but only one anomaly detection system is known to be efficient for now.

In this short paper, we propose a Web learning-based anomaly detection system based on this system, and resulting from the junction of academic research in several fields, which we improved. The system analyzes HTTP requests as logged by most of the Web servers; it exclusively relates to the queries containing attributes. The analysis process implements a multi-model statistical approach. A Bayesian network is used as decision process, specifying six states (one normal state and five attack states) at the classification node. The system is improved after each log analysis thanks to a technique of alarm clustering, which allows filtering false positive. Compared to traditional anomaly detection systems, the system we present globally gains in *sensitivity* (each step of the process reduces the number of false positive to be dealt with) and in

Dagorn N. (2006).
INTRUSION DETECTION FOR WEB APPLICATIONS (SHORT VERSION).
In *Proceedings of the International Conference on Security and Cryptography*, pages 32-39
DOI: 10.5220/0002097900320039
Copyright © SciTePress

specificity (if an attack is detected, its type is immediately specified). Moreover, a co-operation feature (alarm correlation) with other systems is proposed for distributed intrusion detection. To date, the system has only been partially implemented but the preliminary experiments in real environment show encouraging results.

The paper is organized as follows. Section 2 presents the context and related work on improving detection and reducing the false alarm rate by focusing on findings of Kruegel, Valdes and Julisch. Bayesian networks are also introduced. Section 3 describes our proposal, combining and improving parts of the previously presented approaches. Section 4 sums ups the system's implementation/ evaluation to date, and outlines ongoing efforts. Section 5 discusses the results and concludes.

2 RELATED WORK

In computer security, attacks on the application/service level have particularly been increasing for the last years (attacks on authentication or authorization mechanisms, client-side attacks, command execution, information disclosure, and logical attacks[i]). At the present time, they receive much attention in the research field of intrusion detection, not only because they are extremely dangerous, but also because of the increasing importance of the Web services. This paragraph presents four of these approaches, selected because of their particularly interesting contribution to the above mentioned issues.

2.1 Analysis Models for the Detection of Web Attacks

Kruegel, Toth and Kirda (Kruegel et al, 2002) propose a service-specific anomaly detection approach, which extends traditional network traffic models considering only packet header information to include as well the packet payload. Three statistical tests are implemented to detect a potential anomaly: type of request, length of request and payload distribution. At the end of the detection process, a global anomaly score is computed.

In (Kruegel and Vigna, 2003), Kruegel and Vigna increase the detection to six models: attribute length, attribute character distribution, structural inference, token finder, attribute presence or absence, and attribute order.

In (Kruegel et al, 2005), Kruegel, Vigna and Robertson still improve the system by adding three additional models, so that the nine *analysis models* finally proposed are: attribute length, attribute character distribution, structural inference, token finder, attribute presence or absence, attribute order, access frequency, inter-request (time) delay, and invocation order[ii]. The application-specific characterization of the invocation attributes enables the system to perform focused analysis and therefore to produce a reduced number of false positive. To the best of our knowledge, this approach is the first *Web anomaly detection system* (which works!).

However, the authors notice in (Kruegel et al, 2003) that the large number of false alarms may be caused by an incorrect classification of events in current systems, mainly for two reasons relating to the decision process: on the one hand, the model's outputs form a global sum often simply compared to a threshold, and on the other hand additional information on the models could certainly be helpful during the decision process. These limits can be corrected by resorting to a Bayesian network.

2.2 Contributions of Bayesian Networks

Bayesian networks are generally used to model a field containing uncertainty. A Bayesian network is a directed acyclic graph (DAG) where each node corresponds to a discrete random variable of interest, and the bonds symbolize the influences (causal relationships) between variables.

In intrusion detection, several researchers have adapted ideas from Bayesian statistics; they usually use naïve Bayesian networks[iii] to optimize or create models for anomaly detection. Unlike these proposals, Kruegel, Mutz, Robertson and Valeur (Kruegel et al, 2003) propose an *event classification based on Bayesian networks* to replace the classical threshold-based decision process, in order to improve the aggregation of the different model outputs and allow to seamlessly incorporating additional information from the environment.

Nevertheless, an improvement suggested by the authors would consist in keeping track of the recent anomalies in the Bayesian network. We take this remark into account in our proposal.

2.3 Classification Specification and System Adaptation

The approach of Valdes and Skinner (Valdes and Skinner, 2000) shows two interesting properties.

The first one is the *specification of the Bayesian network classification*, namely the detail at the root node of thirteen final state hypotheses (five normal states and eight states of network attack); so, the system is not restricted in indicating only if the detection process result is normal or anomalous, but allows to specify which hypothesis the Bayesian network tends to, with which probability, and thus to classify finely the detected state. We use this property in our proposal.

The second property is the system's *capacity to adapt*, either by reinforcing its integrated models for a current observation (by adjusting the rows of the corresponding conditional probability tables of the Bayesian network to the state observed at the parent node) or by adding a new state (hypothesis) at the parent node if the current observation is not included in the existing hypotheses. These two adaptation properties will be implemented in a future version of our system (the Bayesian network will start from only one hypothesis of normal state, then will progressively generate the attack states as discovered).

2.4 Reduction of False Alarms with the Clustering Technique

Tools aiming at automating the treatment of alarms are under development (Dain and Cunningham, 2002; Debar and Wespi, 2001; Valdes and Skinner, 2001) but to date no effective solution exists.

Julisch (Julisch, 2003a, 2003b) presents a partially automated approach, based on the observation that each alarm occurs for a reason, called *root cause*. Julisch shows that many root causes manifest themselves in alarm groups, which have certain structural properties. He formalizes these structural properties and proposes a data mining technique, called *alarm clustering*, for extracting alarm groups, which have similar properties. These alarm groups are then presented to a human expert responsible for identifying the underlying root causes. Once identified, the root causes can be removed (respectively, false positive can be filtered out) to reduce the future alarm load (up to 70% on the average).

However, only false positive issued from misuse detection systems are considered in this approach. Our proposal extends it to anomaly detection.

3 OUR PROPOSAL

The anomaly detection system we propose is an attempt to improve the detection of Web-based attacks and to decrease the large number of false positive to be dealt with, by combining Bayesian networks with some improvements we brought to the previously described approaches.

3.1 Data and Operation Modes

Just like Kruegel's approach (Kruegel et al, 2005), the system analyzes HTTP requests as logged by most of the Web servers (e.g., Apache). The analysis process focuses on the association between programs, attributes and their values.

The system can operate in training or detection mode. During the *training phase*, the system creates *profiles* for each server-side program or attribute. Once the analysis models have learnt the characteristics of normal events, the system can switch to *detection mode.* The task of the models is then to return a value for a certain request, which reflects its state compared to the profile established for the model. The assumption is that attribute values with a sufficient low probability (i.e., abnormal values) potentially indicate an attack. Based on the output probabilities of all the detection models, a query is either reported as normal or as a potential attack by the Bayesian network.

3.2 System Presentation

The overall detection process consists of three steps, which we call *analysis, decision* and *refinement* of the model.

Analysis. The first step *analyzes* each request coming from a monitored Web server. Information specific to the request is captured and serves as input to the analysis process (detection models). Ten analysis models are used; nine of them are taken up from *Kruegel's multi-model approach* (Kruegel et al, 2005).

We add an *anomaly history model*, which keeps track of the recent anomalies and checks whether an event is one of them. This model also allows measuring the events in time in a weighted manner, in the sense that an event, which has just occurred, has more weight in the system than events, which occurred a long time ago (and which are weighted so as to decrease their importance). Lastly, the anomaly history model will achieve in a future version a correlation of alarms coming from other sources, in order to enable the co-operation of the system with other intrusion detection systems in case of a distributed intrusion detection policy.

Each model outputs a real value in the interval [0,1] which reflects the deviation of the event's attributes from its profile.

Decision process. The second step *decides* whether the analyzed request is normal or an attack. An *extended Bayesian network* is substituted as a

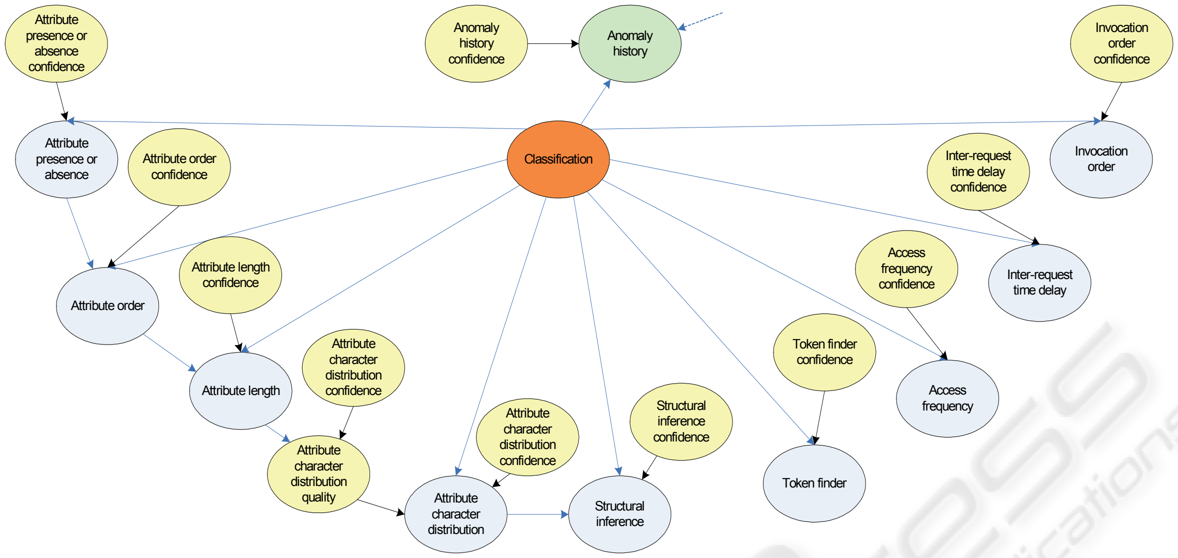
Attribute presence or absence confidence
Anomaly history confidence
Anomaly history
Invocation order confidence
Attribute presence or absence
Attribute order confidence
Classification
Inter-request time delay confidence
Invocation order
Attribute order
Attribute length confidence
Access frequency confidence
Inter-request time delay
Attribute length
Attribute character distribution confidence
Token finder confidence
Access frequency
Attribute character distribution quality
Attribute character distribution confidence
Structural inference confidence
Token finder
Attribute character distribution
Structural inference

Figure 1: Bayesian network for the characterization of Web-based attacks.

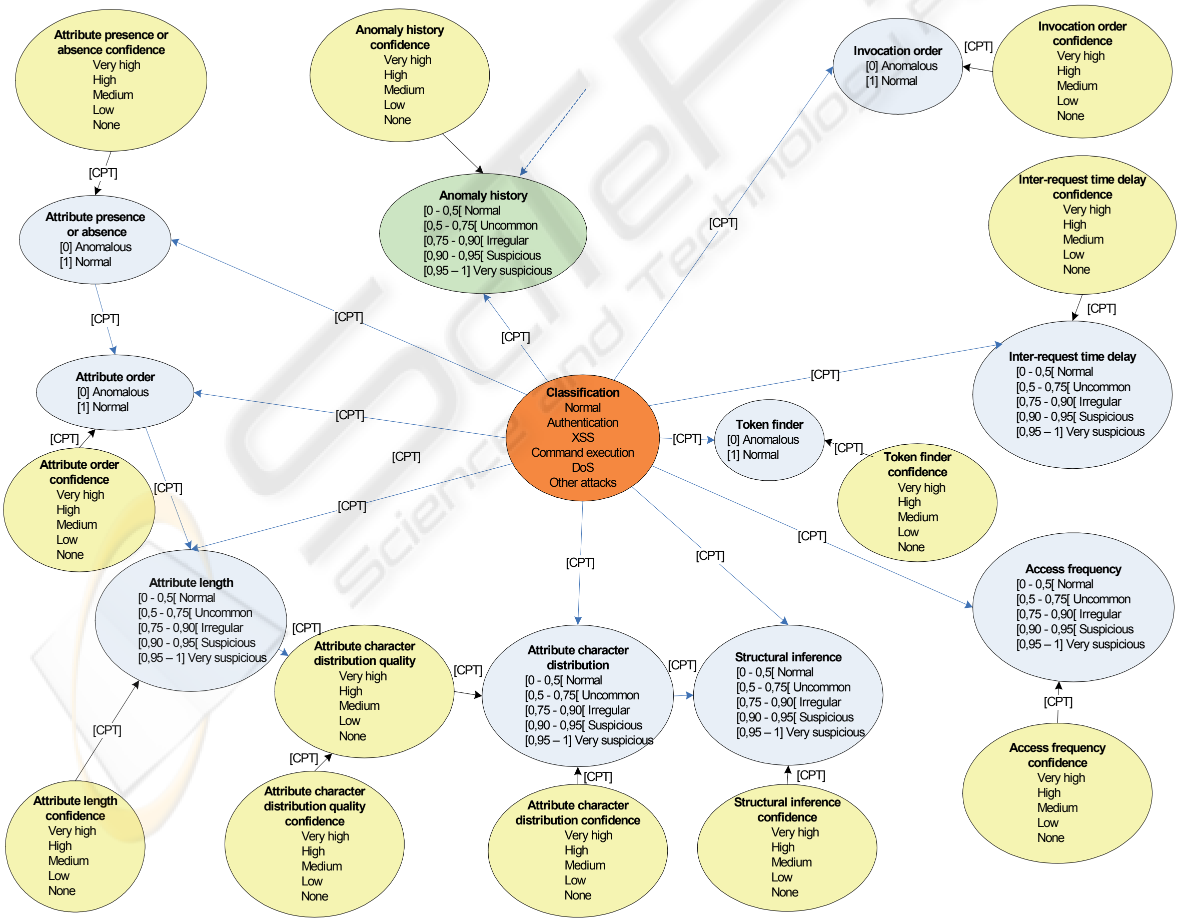

Figure 2: Developed Bayesian network.

SCITEPRESS Science and Technology Publications

decision process to the threshold technique generally used for anomaly detection, and joins thereby the *Kruegel's event classification* (Kruegel et al, 2003). The value returned by each model is incorporated as evidence in the Bayesian network. According to the model's output, each node includes two (normal, anomalous) or five possible states as defined by Kruegel (Table 1).

Table 1: Anomaly Score Intervals (Kruegel et al, 2003).

Anomaly Score Range	Level
[0.00 , 0.50[	Normal
[0.50 , 0.75[	Uncommon
[0.75 , 0.90[	Irregular
[0.90 , 0.95[	Suspicious
[0.95 , 1.00]	Very suspicious

Each node is also associated a *conditional probability table*, whose values are initially established after the training period according to our specific knowledge in the field.

To improve the detection process, just as (Kruegel et al, 2003), every node in the extended Bayesian network is associated a *confidence node*. The conditional probability tables are adjusted so that each model output has a weighted influence on the decision according to its confidence level. The model confidence is represented as one of five discrete levels: very high, high, medium, low or none (Kruegel et al, 2003).

The task of the event *classification* process is to determine whether the treated request is normal or not, given the outputs of the different models for all its attributes. If the request is anomalous, the suspected attack is indicated with its relative probability. The system is able to specify attacks on authentication mechanisms, client-side attacks (XSS), command execution, and logical attacks (denial of service). A class "other attacks" gathers attacks not distinctly specified, but which are evaluated as anomalous events by the detection models. The root node of the Bayesian network thus includes six possible states: normal, authentication, XSS, command execution, denial of service, and other attacks.

Fig. 1 shows the structure of the Bayesian network we propose for the characterization of Web-based attacks. Unlike to (Kruegel et al, 2003), the conditional probability tables initially specified for each node are adjusted in our case after each log analysis according to Julisch's technique; in other words, the probabilities chosen before the beginning of the evaluation are continuously modified thereafter.

Fig. 1 also shows *model dependencies*. In particular, based on (Kruegel et al, 2003), we identify a dependency between the nodes attribute length and attribute character distribution. Therefore, an intermediate node attribute character distribution quality is inserted between them to point out that the attribute length influences the character distribution.

In a more developed shape, the extended Bayesian network can be represented by Fig. 2. By preoccupation with readability, the conditional probability tables are not developed on the figure (the largest would count 5 rows x 150 columns at the attribute character distribution node).

At the end of the decision process, the probabilities of the six states associated with the classification node are calculated. When an event has a "high enough" probability to be anomalous at the root node, an alarm is raised. The raised alarm is also transmitted to one (or several) other system(s) in case of a distributed intrusion detection policy. All the anomalous events are stored in log files.

Refinement of the model. In the third step of the process, *Julisch's alarm clustering* technique (Julisch, 2003a, 2003b) is actively used to group the alarms in clusters and identify their root causes. False positive are filtered out. After each log analysis, the conditional probability tables in the extended Bayesian network are adjusted, so that the false positive identified will no more appear in the next sessions. This technique thus contributes to *refine* the model.

For a better comprehension, we provide two motivating examples in Appendices A and B. A theoretical example first describes the overall detection process, by focusing on a particular model facing a hypothetical attack (attack on authentication mechanisms). Then a practical example describes the fulfillment of the same attack on the Web server of the testing company (see § 4), automated with the THC-Hydra tool (THC-Hydra); this last case shows in particular how suitable values of the conditional probability tables can be fixed for a specific node.

4 IMPLEMENTATION, EVALUATION AND FUTURE WORK

For the moment, only the Bayesian network and four analysis models have been developed and tested.

4.1 Implementation

We have partially implemented our analysis models (four models of a total of ten: attribute presence or absence, attribute order, attribute length and anomaly history), following Kruegel's relevant choice (Kruegel et al, 2003) to use for the event classification module the C++ language and the SMILE Bayesian statistics library (SMILE).

SMILE has a Windows user interface, called GeNIe, which can be used to create decision theoretic models intuitively. In a pre-implementation phase, GeNIe has already allowed us to create and test manually the proposed Bayesian network. This one seems correctly designed; collected data are consistent and meet our expectations.

4.2 Evaluation

We evaluated the four implemented models of our system in real environment. We used a set of data (request logs) issued from a Web server of a real company established in Luxembourg (Luxembourg). Consolidated subsidiary of a corporate counting 158000 employees over 35 countries with revenue of €37 billions[iv], the Luxembourg company counts 86 employees and offers a broad range of IT services and products, in particular for state administrations and the financial sector.

The application to protect is a business application collecting the working times to charge on projects; it is used by all the employees of the Luxembourg company. This application was developed using Java servlets, running in a JBOSS environment, which is based on an Apache Web server. The same type of server (same architecture, same type of data) is used for the same purpose in Brussels (Belgium) by the Belgian branch of the company, whose core business is the same and which counts approximately 250 employees.

The preliminary experiments consist in sending requests containing anomalous events (additional or reversed attributes, too long attribute values…) to the implemented analysis models. About thirty anomalous events were thus tested (sometimes combined) on the four models; all the events were detected by the models and were evaluated as anomalous by the Bayesian network. At this stage of development, the Bayesian network did not have sufficient information to specify alarms[v].

4.3 Ongoing Efforts

In the months to come, we have the ambition to implement the six remaining analysis models (i.e., the attribute character distribution, structural inference, invocation order, access frequency, inter-request time delay, and token finder models), the clustering technique, and to evaluate the complete system in the testing company.

5 DISCUSSION AND CONCLUSION

This short paper describes a very first step of our ongoing research in the field of intrusion detection (ph.D. thesis). In addition to the judicious combination of several approaches, our contributions in this paper are multiple.

First, our proposal improves original work at each phase of the intrusion detection process:

- in the analysis phase, a model (anomaly history) is added to Kruegel's models, enriching the system with temporal and co-operation features;
- in the decision phase, a Bayesian network replaces the traditional summation process, confidence nodes weight the different models' outputs, and the classification (Valdes) is able to identify five different types of Web attacks;
- in the model refinement phase, the alarm clustering technique proposed by Julisch for mi-suse detection is applied to anomaly detection.

These improvements contribute to an additional reduction of false alarms in the global anomaly detection process, resulting in a gain in *sensitivity*.

Second, the specification of the Bayesian network at the classification node (six states) recognizing five specific Web attack patterns allows classifying very precisely the status of the analyzed requests. This classification is not only limited to a simple result "normal" or "anomalous". If the request is evaluated as anomalous by the Bayesian network, the suspected attack and its precise probability are returned. This improvement allows the analyst to obtain precisions on a potential intrusion, resulting in a gain in *specificity*.

Lastly, we were eager to equip the proposed detection system with a *co-operation feature*, making its implementation possible in case of a distributed intrusion detection policy. This feature is not yet effective at the present time but will be implemented for our thesis work, just like the *dynamic hypotheses generation* and a *real-time request analysis* features.

A negative report is presented that the complete implementation of our system and its evaluation in real world could not be achieved before the submission of this paper; nevertheless, our first experiments show encouraging results.

REFERENCES

Dagorn, N., 2006. Intrusion Detection for Web Applications. *Proceedings of the 5th IADIS International Conference on www/Internet (ICWI 2006)*. Murcia, Spain.

Dain, 0. and Cunningham, R.K., 2002. Fusing heterogeneous alert streams into scenarios. In D. Barbara and S. Jajodia (Eds.), *Applications of Data Mining in Computer Security*. Kluwer Academic Publishers, Boston, MA.

Debar, H. and Wespi, A., 2001. Aggregation and correlation of intrusion-detection alerts. *Proceedings of the 4thWorkshop on Recent Advances in Intrusion Detection (RAID)*. LNCS, Springer Verlag, pp. 85-103.

Julisch, K, 2003a. Clustering Intrusion Detection Alarms to Support Root Cause Analysis. *ACM Transactions on Information and System Security* 6(4).

Julisch, K., 2003b. *Using Root Cause Analysis to Handle Intrusion Detection Alarms*. PhD Thesis, University of Dortmund, Germany.

Kruegel, C., Toth, T., Kirda, E., 2002. Service Specific Anomaly Detection for Network Intrusion Detection. *Proceedings of the 17th ACM Symposium on Applied Computing (SAC)*. ACM Press, Madrid, Spain.

Kruegel, C., Vigna, G., 2003. Anomaly detection of Web-based attacks. *Proceedings of the 10th ACM Conference on Computer and Communication Security (CCS'03)*. Washington, DC. ACM Press, New York.

Kruegel, C., Mutz, D., Robertson, W., Valeur, F., 2003. Bayesian Event Classification for Intrusion Detection. *Proceedings of the 19th Annual Computer Security Applications Conference (ACSAC)*. IEEE Computer Society Press, USA.

Kruegel, C., Vigna, G., Robertson, W., 2005. A multi-model approach to the detection of web-based attacks. *Computer Networks*, Vol. 48, Issue 5. Elsevier.

State, R., 2005. *Intrusion Detection*. Tutorial Master2. Nancy1.

Valdes, A. and Skinner, K., 2000. Adaptive, Model-based Monitoring for Cyber Attack Detection. *Recent Advances in Intrusion Detection (RAID 2000)*. Lecture Notes in Computer Science, No. 1907, pp. 80–92.

Valdes, A. and Skinner, K., 2001. Probabilistic alert correlation. *Proceedings of the 4thWorkshop on Recent Advances in Intrusion Detection (RAID)*. LNCS, Springer Verlag, Berlin, pp. 54-68.

SMILE: Structural Modeling, Inference and Learning Engine. http://genie.sis.pitt.edu/.

THC-Hydra: http://www.thc.org/thc-hydra/.

APPENDIX A: THEORETICAL MOTIVATING EXAMPLE

Suppose the following request (assumed to have been extracted from a monitored Web server log):

```
192.168.10.10 - username [2/April/2006:19:36:25 -0800] "GET /scripts/cmd.pl?id=524&name=dummystring&country=passwd" 200 2122
```

Step 1: Analysis. *The request is analyzed*. The query serving as input for the ten anomaly detection models is: `id=524&name=dummystring&country=passwd`. For certain models, the complete query is analyzed; for others the analysis only focuses on the attribute values (`id`, `name`, `country`). Let us develop the case of the token finder model. The three attributes are successively injected as input of the model. The attributes `id` and `name` being of random type (i.e., not part of an enumeration), the model returns the value 0 (normal). Unlike this, the `country` attribute is a token of an enumeration and can contain only a valid country name. The attribute value `passwd` is not an acceptable input; therefore, the model returns the value 1 (anomalous) for this attribute.

Step 2: Decision. *The request is evaluated as normal or as an attack.* During the training period, the variance of the analyzed attributes was assumed relatively low for the token finder model, so that the confidence level now associated with the model in the Bayesian network is high. We consider the output value of the token finder model, provided by the analysis of the `country` attribute. The value 1 returned is injected as evidence into the Bayesian network. The anomalous state is raised by the node in the network. A message is propagated to the classification node according to the conditional probability tables, characterizing an attack on authentication mechanisms; this message is only very slightly decayed because of the high confidence in the model. So, the classification node is updated not only according to the weighted message transmitted by the token finder model, but also according to the observations resulting from the other nodes in the network (weighted by their respective confidence level). Once the complete query treated, the probability of an anomalous state at the classification node is calculated. If an anomalous state (i.e., a specified attack) is detected with a high enough probability value, the request is considered as anomalous and an alarm is raised (the raised alarm is also transmitted to one or more other systems in case of a distributed intrusion detection policy). The anomalous request detected is stored in the alarm log.

Step 3: Model refinement. *The detection process is continuously improved.* Suppose the following alarms registered in the alarm log at the end of the detection period (very simplified examples):

Table 2: Alarm log.

RC	IP source	Service	Query
1	192.168.10.10	cmd.pl	id=524&name=dummystring&country=passwd
2	212.100.5.54	access.pl	pass=superuser&user=root
3	115.58.184.32	login.pl	userid=test&password=&role=viewer

Among the three alarms logged, alarm 1 (anomaly detected by the token finder model) is identified by a human analyst as a true positive: a real intrusion (an attack on authentication mechanisms) was detected by the intrusion detection system. Alarms 2 and 3 are identified by the analyst as false alarms (root causes known as false positive). Indeed, for alarm 2, the attributes entered are correct but were not specified in the appropriate order (anomaly detected by the attribute order model). For alarm 3, the password attribute is empty (anomaly detected by the attribute length model). For both alarms, the conditional probability tables at the evidence nodes are adjusted so that they will no more interfere in the future sessions.

APPENDIX B: PRACTICAL EXAMPLE (THC-HYDRA TOOL)

In this practical example, we simulate the same attack on authentication mechanisms using the password cracker THC-Hydra (we used the Hydra for Windows version). This time, let us observe the more representative inter-request time delay model.

Training. During the training phase, all the inter-request time delays of the requests submitted to the Web server are collected. At the end of the training process, the model calculates the average and the standard deviation of the inter-request time delays and deduces the normal distribution (without any attack). In our case, the logs collected by the testing Web server over one week duration shows an average of 57,5 seconds and a standard deviation of 65,53 (curve in dotted lines on Fig. 3).

Detection. The attack, automated with the THC-Hydra tool, generated many requests in order to discover the connection password of a given user. Following this series of requests, the model observes an average of 0,19 seconds and a standard deviation of 0,16 (curve in full lines on Fig. 3).

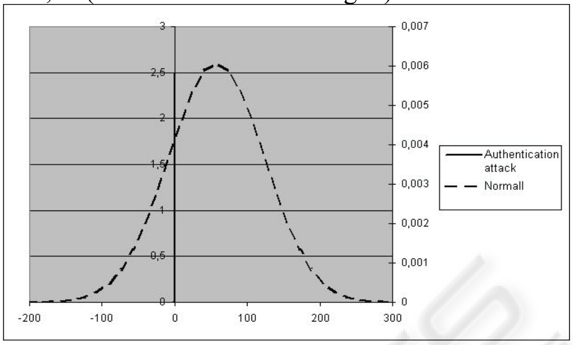

Figure 3: Inter-request time delays distribution.

The model notes that the attack distribution does not correspond to the expected distribution and must return an anomalous value (tending to 1). From the measurement of 0,19 seconds observed with THC-Hydra, a threshold of 0,25 seconds is fixed, from which a probability of 0,80 of having an anomalous state (attack) is estimated (i.e., 80% chances that the calculated inter-request time delay reveals an attack on authentication mechanisms). In the same way, the value of 57,5 seconds observed allows to fix a 50 seconds threshold, from which a probability of 0,01 is estimated (i.e., 1% chances that the calculated inter-request time delay reveals an attack on authentication mechanisms). The intermediate average values are fixed arbitrarily according to a decreasing curve (results in Table 3).

Table 3: Conditional probability tableat the inter-request time delay node.

Average (seconds)	Anomaly score range	Level	Probability of an attack on authentication mechanisms
[∞ , 50[	[0.00 , 0.50[	Normal	0,01
[50 , 15[	[0.50 , 0.75[	Uncommon	0,02
[15 , 4[	[0.75 , 0.90[	Irregular	0,07
[4, 0.25[	[0.90 , 0.95[	Suspicious	0,10
[0.25 , 0]	[0.95 , 1.00]	Very suspicious	0,80

[i] Source: Web Application Security Consortium http://www.webappsec.org/

[ii] We encourage the reader to refer to (Kruegel et al, 2005) for further explanations on these models.

[iii] A naïve Bayesian network is a two-layer network, which assumes complete independence between the information nodes.

[iv] Consolidated financial figures for fiscal year ending March 2006.

[v] Functional and technical architectures of the implementation, as well as more evaluation results, are presented in (Dagorn, 2006).