Hybrid Detection of Application Layer Attacks Using Markov Models for Normality and Attacks

Rolando Salazar-Hernández and Jesús E. Díaz-Verdejo

CTIC - Dpt. of Signal Theory, Telematics and Communications,
University of Granada (Spain)
rsalaza@correo.ugr.es, jedv@ugr.es

Abstract. Previous works has shown that Markov modelling can be used to model the payloads of the observed packets from a selected protocol with applications to anomaly-based intrusion detection. The detection is made based on a normality score derived from the model and a tunable threshold, which allows the choice of the operating point in terms of detection and false positive rates. In this work a hybrid system is proposed and evaluated based on this approach. The detection is made by explicit modelling of both the attack and the normal payloads and the joint use of a recognizer and a threshold based detector. First, the recognizer evaluates the probabilities of a payload being normal or attack and a probability of missclassification. The dubious results are passed through the detector, which evaluates the normality score. The system allows the choice of the operating point and improves the performance of the basic system.

Keywords: network security, intrusion detection systems, markov models.

1 Introduction

Intrusion detection systems (IDS) constitute a valuable tool for network security officers. Their primary goal is to detect attacks or intrusions to computer systems, preferably in near real-time, and trigger an alarm.

In general terms, they can be classified according to two basic criterions [1] [2]: the source of the monitored events and the type of detection. Thus, a network-based IDS (NIDS) is a system which tries to detect attacks by monitoring network traffic, while a host-based IDS (HIDS) monitors internal events of the hosts. This paper is focused in NIDS. So, in what follows, only this kind of IDS will be considered even if not explicitly stated.

On the other hand, if the IDS try to detect the attacks from a set of rules or signatures of the known attacks, that is, from the knowledge of the previously observed attacks, it is called Signature-based IDS (SIDS). On the contrary, if the IDS tries to model the normal behaviour of the system and to detect attacks from the deviations of this model, it is called Anomaly-based IDS (AIDS). For the purposes of this paper it is important to notice that none of them -AIDS vs.

M. Soriano, S. Qing, and J. López (Eds.): ICICS 2010, LNCS 6476, pp. 416–429, 2010.

SIDS- is better than the other. Both present some advantages and shortcomings related to their performance and their skills to detect previously unknown or unobserved attacks. Regarding their performance, the figures of merit are not only the attack detection rate but also the false positives rate [2]. In this respect, SIDS use to achieve better results than AIDS. On the contrary, the ability to detect new attacks, i.e. 0-day attacks, is very limited for the SIDS while the AIDS are supposed to be able to detect every attack that produces a deviation in the behaviour of the system (suspicion hypothesis) at the cost of an increase in the false positives rate.

Therefore, it would be desirable to develop a hybrid system which puts together the best of both behaviours. Some proposals are described in the bibliography [4] [5] [6]. The most common approach tries to somehow combine the scores or the outputs of a SIDS and an AIDS to obtain a classification of the events. Another approach is based on the modeling of both the attacks and the normal behaviour of the monitored system e.g. [7].

The system proposed in this work is based on the same principle, that is, explicitly considering the desired and undesired nature of the events -attack vs. normal-, although the technique used for the modelling is clearly different and a two steps procedure is finally proposed. The system uses a Markov models-based solution termed SSM (Stochastic Structural Model) [8]. This technique establishes a Markov model for the normal payloads of a given protocol both from training and the specifications of the protocol. It has been successfully used as the core of an AIDS for the detection of attacks in the payloads of protocols as HTTP and DNS [9]. In this paper we describe the use of this modelling to build a two classes recognizer providing also a confidence measure that is used to refine the results. Although this method could be applied to other network services, the present work focuses on HTTP URIs, as most network-based attacks are currently web related [10]; moreover, an important proportion of Internet traffic is HTTP-based.

The rest of the paper is structured as follows. First, the basis of the modeling technique, SSM, is briefly reviewed in Section 2. Section 3 explains the experimental setup, including a description of the traffic databases used and their origin and the figures of merit of the SSM IDS for those databases. In Section 4, the use of the SSM technique for the modelling of attack and normal payload is described and a recognizer is used to classify each payload. From the observation of the results, a confidence score of the classification is presented. Based on this score, a two steps classification method is proposed in Section 5, and a comparison of the results is presented. Finally, Section 6 presents some conclusions and insights on the possible enhancements of the system.

2 SSM for Payload Modelling

Most network protocols, especially those in the application layer, have a well defined structure for the messages exchanged by the service entities involved. This structure is given through the corresponding protocol specifications, and

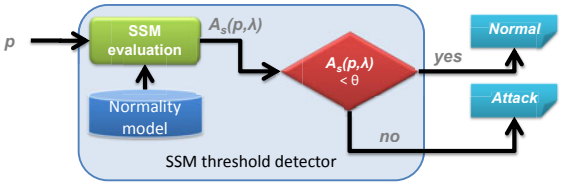


Fig. 1. Diagram of SSM-based threshold classifier

makes it possible to use formal methods to describe the way in which each message is produced.

Hence, for every protocol data unit whose contents present a known structure, its likelihood may be estimated in order to detect usage anomalies. This is the context for the Stochastic Structural Model (SSM) approach [8], in which a stochastic Markov model is proposed to represent the structure of the payloads of the packets.

SSM makes use of the Markov theory [11] to provide a production model and its associated probabilities for the observed payloads. Thus, a given payload, p, can be evaluated by a model, λ, to provide a probability, $P(p|\lambda)$, of the payload being generated by the model. From this probability, it is possible to define a *normality score*, $N_s(p)$, as

$$N_s(p) = P(p|\lambda) \tag{1}$$

As this normality score represents the probability of the observed payload according to the given model, and assuming that the model properly represents the normal behaviour of the payloads for a service, this measure can be used to classify this single payload as either normal or anomalous, (Fig. 1), according to a given threshold, Θ:

$$class(p) = \begin{cases} normal\,, if\ \ N_s(p) \geq \theta \\ anomalous\,, otherwise \end{cases} \tag{2}$$

The key points, thus, are how to obtain a model that is accurate enough to represent the normal behaviour of a system, and how to determine the elements of such a model. In SSM, the behaviour model is derived from the protocol specification and the observation of normal instances of the protocol in the monitored environment.

2.1 Elements of SSM Models

Briefly, a first order-Markov model, λ, is formally defined by a tuple of elements, $\lambda = (S, V, A, B, \Pi)$:

- A set of N states, $S = \{s_1, s_2, , s_N\}$
- A set V of M observables (or symbols), which can be produced by the system when visiting (or leaving) each of the states: $V = \{o_k, 1 \leq k \leq M\}$

- The transition probabilities among the states, $A = \{a_{ij}, 1 \leq i, j \leq N\}$ where $a :_{ij}$ is the probability of transition from the state s_i to the state s_j.
- The observation probabilities for the symbols in each state, $B = \{b_{ik}, 1 \leq i \leq N, 1 \leq k \leq M\}$ where b_{ik} corresponds to the probability of production of the symbol o_k while being in (or leaving) the state s_i.
- The probabilities of each state being the first one of a given sequence, $\mathit{\Pi} = \{\pi_i, 1 \leq i \leq N\}$

The normality score of a payload, p, as derived from the Markov model, can be evaluated according to

$$N_s(p) = P(p|\lambda) = \pi_{S_1} \prod_{t=1}^{T-1} a_{S_t S_{t+1}} b_{S_t O_t} \tag{3}$$

where T is the length of the observed sequence of symbols, S_t is the state for the system at t, and O_t is the symbol at this instant.

In the HTTP case, the structure of the finite state automaton for URIs in GET requests can be deducted from the RFCs 1945 [12], 2068 [13], 2396 [14] and 2616 [15]. The following fields in the URI carried by GET requests are considered:

- Protocol (scheme in RFC2396): Protocol. In our case, always http.
- Host: Name (recommended) or address of the machine where the resource resides. The port (:port), if present, will be considered as part of this field.
- Path segment: Each of the elements, or segments, of the path that specifies the location of the required resource within the host. If present, the fragment (#) will be considered, for our purposes, as part of the path segment.
- Query: String of information to be interpreted by the resource. For our purposes, two fields are considered here:
  - Attribute: The name of a variable or a string.
  - Value: The value assigned to the variable.

A URI consists of an optional protocol, an optional host and port, a sequence of one or more path segments, which constitutes the absolute path in the RFC2616 naming conventions, and, optionally, a query composed of a sequence of attributes, each of which has an optional value. Thus, according to RFC2616, an HTTP URI or URL presents the general form:

"http://" host [":" port] [abs_path ["?" query]]

Any URI can be easily parsed and divided into a sequence of field values (or tokens) by considering a set of delimiters, $D = \{: //, /, ?, =, \&, EOR\}$. Therefore, a field value will be the sequence of characters or segment between two consecutive delimiters. Furthermore, each delimiter can be seen as the transition trigger from one state to another, the destination state being governed by the kind of field, which, once again, is related to the delimiter (Fig. 2).

As an example, consider the URI

http://www.site.com/it/index.php?sec=100&chapter=link

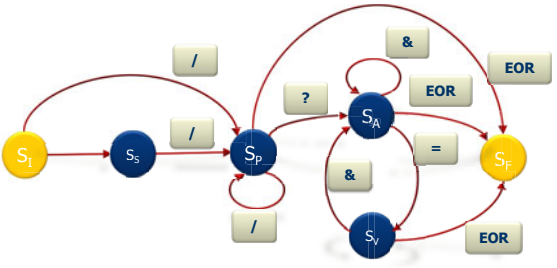


Fig. 2. FSA model for HTTP protocol

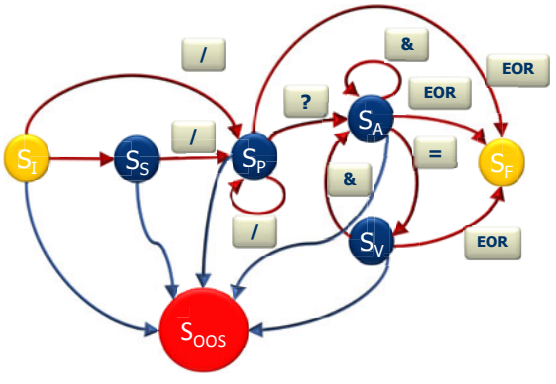


Fig. 3. SSM model for HTTP protocol

This URI can be segmented into 8 fields or tokens in accordance with the set of delimiters D, that is,

{http,www.site.com,it,index.php,sec,100,chapter,link}

In order to facilitate the application of the model in an anomaly detector, an additional state, S_{OOS}, can be added. This is reached from any other state when the observed delimiter is not allowed in the current state. Thus, S_{OOS} is an out of specification state, which means that, when it is reached, the URI analyzed is incorrect. The resulting FSA for HTTP URIs for its use in an anomaly detector is depicted in Fig. 3. Each arc is labelled with the delimiter that triggers the transition between the two states involved.

The symbols in the vocabulary, V, and their probabilities, that is, B matrix, can be obtained by training the model with a database of observed legitimate payloads. The probabilities can be estimated by a simple accounting procedure, taking into account the frequencies of appearance of each symbol. Therefore, the probability of observing a symbol o_k while in a state s_i can be estimated from

$$p(o_k|s_i) = b_{ik} = \frac{count(o_k|s_i)}{\sum_{j=1}^{M} count(o_j|s_i)} \tag{4}$$

2.2 Practical Issues

The use of Markov models presents some practical problems. The first of these concerns the evaluation of cumulative probabilities. To avoid this problem, it is usual to consider a logarithmic formulation of the processing carried out by the Markov models. The second problem is related to the possible appearance of a symbol that was not observed in the training stage, that is, a word not included in the vocabulary: the so-called out of vocabulary problem (OOV). Two approaches are possible in this situation: to dismiss the word and assign a zero probability to it, and therefore to the global payload, or otherwise assign it a low fixed value -the smoothing solution. A third problem may appear as a result of the sequences analyzed being of different lengths. Accordingly, some kind of normalization procedure is needed to provide independence with respect to the number of fields in a given payload. Nevertheless, the above-mentioned problems are well-known and their solutions are fully described in the bibliography [16].

An additional issue that should be pointed out is the need to have a clean training set [17]. In other words, the training set must be representative of this normal operation and not contain attack instances. On the other hand, the traffic should be real, not simulated, as the purpose is to model the normal operation of a real environment with real users [18]. The absence of attacks is, in fact, very difficult to accomplish as there should be no control on the traffic from users. Various approaches to this problem have been proposed in the literature [17] [19], but they all rely on the use of a S-NIDS to filter out the attacks in the captured traffic, which can be inaccurate due to false positives and to detection errors in the process.

3 Experimental Framework

The evaluation of the performance of the proposed system requires some databases with HTTP URIs to be used both for training and testing purposes. By using those databases, an SSM-based AIDS is tuned for its comparison with the proposed system.

The approach used to assess the performance of the system is to build two databases composed by GET requests. The first one is composed by traffic captured in a real environment and should be attack-free. The second database is composed by attacks gathered from different sources and will be also used to train and test the system. Both sets will be split in various subsets in order to allow the training and testing of the system in a leave-one-out procedure [20].

The performance of the system will be measured by using ROC curves [21] representing the detection rate vs. the false positives (FP) rate, that is, the percentage of attacks detected vs. the percentage of normal payloads that are classified as attacks. The results obtained will not be directly comparable with those in the literature, as the databases are different and the attacks have been collected apart from the normal traffic. This way, the FP rate will be biased [18] making unfeasible the comparison. Nevertheless, this experimental setup

Table 1. Characteristics of the databases used for assessing the performance of the system

Clean traffic	Database	Requests (bulk)	Requests (clear)	Vocab. size
	PVH	1.176.781	1.176.557	28.025
Attack traffic	Database	N. Attacks	N. Instances	Vocab. size
	RDB	338	707	5073
	OSVDB	5073	6895	11692

allows the comparison of the results provided by the systems under study. The databases and the procedure used for their acquisition are briefly described next.

The attack-free or normal traffic database has been acquired in the production real network of an academic institution. The capture was made by using tcpdump (http://www.tcpdump.org) during one week and only HTTP traffic was monitored. The resulting bulk database, called PVH, have been processed following the method described in [17] in order to obtain traffic suitable to be used to train the systems. Therefore the obtained HTTP requests were first filtered to extract GET request. To assure to a certain degree that no attacks are included in the training set, and according to the considered method, the traffic has been analyzed by Snort (http://www.snort.org) with most updated rules to filter out attack or suspicious requests. This way, 16 attacks have been detected and eliminated from the initial set of requests. Additionally, the URIs have been normalized and parsed to test its compliance with the standards. 208 non-compliant URIs have been filtered out. The most relevant features of this database are shown in Table 1.

There exist many sources for information regarding attacks and vulnerabilities in Internet, being notable Bugtraq by Securityfocus [22] and Open Source Vulnerabilities Data Base (OSVDB) [23]. Furthermore, the vulnerabilities affecting the HTTP protocol as well as their exploits are well documented on those sources, thus enabling the generation of the associated attacks. From the collection and use of these exploits, two databases have been built. The first one, called RDB, uses Bugtraq as the primary source of information. The second one, called OSVDB, is based on Open Source Vulnerabilities Data Base and presents the advantage of including a classification of the attacks according to OSVDB taxonomy [23]. In both cases, the exploits have been conveniently parameterized and executed in a controlled environment. This way, all the HTTP attacks described in those databases have been generated and, in those cases in which the attacks include variants or different possible values, many instances have been included. Finally, a manual inspection of the attacks has been made. Most relevant details of this procedure are described in [24]. The number of attacks and instances is shown in Table 1.

3.1 Reference System

A set of experiments using the SSM approach and the previously described databases have been conducted in order to tune the parameters of the reference

system. All the databases have been split in 3 sets in order to apply the leave-one-out procedure. The training of the system is made with two partitions of clean traffic (PVH database) and evaluated with the third part of clean traffic and its equivalent part of attacks (RDB or OSVDB). The vocabularies in each partition are slightly different (around 2% of different words), which implies the appearance of words during the tests that have not been observed during training. Therefore, it is necessary to tune the value of the out-of-vocabulary (OOV) probability.

A lower value of OOV penalizes the appearance of words not included in the vocabulary lowering the normality score for requests including those words, which will primarily increase the false positives rate. Furthermore, as the value of OOV decreases the risk of overtraining the system increases. On the other hand, as the attacks would present a different vocabulary than the normal requests, a higher value for OOV would decrease the detection rate. A compromise is required. The results obtained by varying the value of OOV are shown in Fig. 4. As expected, the performance is slightly different for both databases, although the behaviour is very similar. From the results, a value of 10^{-9} for OOV for subsequent experiments has been selected.

4 Recognizer-Based System

The proposed system uses SSM to represent both the normal requests and the attack requests. Therefore, a model for the normal payloads, λ_N, and a model for the attacks, λ_A, will be obtained from the training phase.

The classification of a payload, p, will be made by a recognizer that will assign p to the class of the model providing the highest probability,

$$class(p) = \begin{cases} normal, & if\ P(p|\lambda_N) \geq P(P|\lambda_A) \\ anomalous, & otherwise \end{cases} \tag{5}$$

The performance of the recognizer-based system has been evaluated using the available databases for normal and attack payloads and the leave-one-out procedure with the same 3 partitions previously established. According to the tuning of the reference system, the value for the OOV probability is also set to 10^{-9} for all the models. The results obtained from the aggregation of the experiments are shown in Table 2. It is important to notice that, as the decision is based on which model yields the highest probability, only a single point of operation for the recognizer is possible. Therefore, it is not possible to adjust a compromise among detection rate vs. false positive rate through a tunable parameter. The results obtained show a lower performance than the one obtained by the reference system, as the operating point is below the ROC curve for the reference system.

4.1 Confidence of the Classification

To assess the behaviour of the recognizer, the confusion area is analyzed. For this, a score $s(p)$ is defined as the difference between the probabilities of a payload being an attack and being normal,

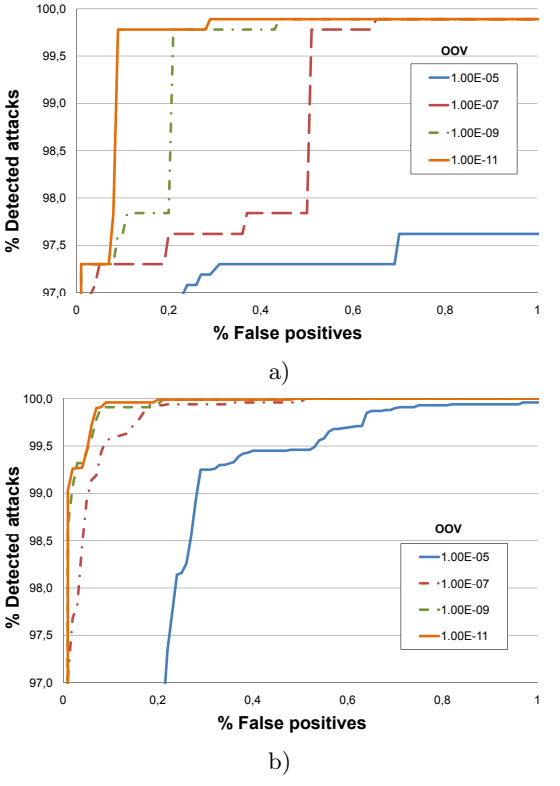


Fig. 4. Selection of OOV value for the reference system. a) RDB attacks, b) OSVDB attacks.

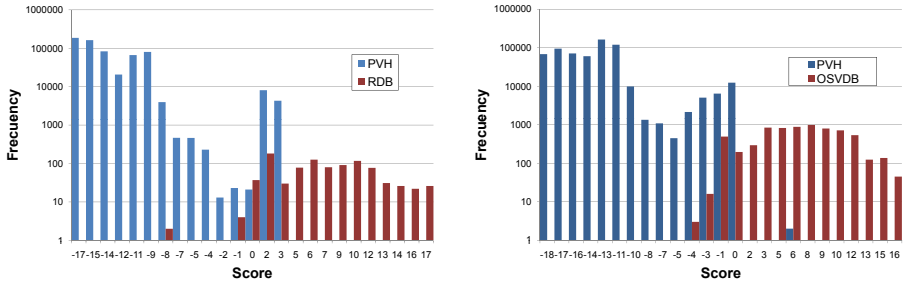
$$s(p) = P(p|\lambda_A) - P(p|\lambda_N) \tag{6}$$

This score can be interpreted as follows. A high positive value for the score for a payload means that the probability of this payload being an attack is significantly higher than that of being normal. On the contrary, a high negative value implies that the probability of being normal is higher than of being an attack. Those payloads for which the score is small (positive or negative) are near the decision boundary and, therefore, would be in the confusion area.

If we represent the histograms for the scores obtained for the normal and attack payloads for both experiments (Fig. 5) we can see that there is an small area around zero for which most of the missclassifications occur. Therefore, a confusion area can be established by setting a threshold, μ, for the score. The payloads outside this confusion area are classified with high confidence according to the recognizer. On the contrary, the confidence for the payloads with low score will be low. This way, it is possible to define a confidence measure based on the score. For example, a sigmoid function of the score and the threshold could be used. Nevertheless, we are not interested in providing this confidence but in improving the classification. For this, some experiments (Table 3) have

Table 2. Results for the recognizer-based system

Experiment	Detection rate	FP
PVH vs. RDB	95.5 %	2.0 %
PVH vs. OSVDB	90.0 %	0.05 %



a) b)

Fig. 5. Distribution of the differences of the probabilities provided by each model (attack vs. normal) for: a) PVH vs. RBD, b) PVH vs. OSVDB

been carried out in which the payloads inside the confusion area are considered unclassified, that is,

$$class(p) = \begin{cases} normal, & if\ s(p) \leq -\mu \\ anomalous, & if\ s(p) \geq \mu \\ unknown, & if\ |s(p)| < \mu \end{cases} \tag{7}$$

The results show that it is possible to correctly classify most of the payloads if a threshold around 3.0 is selected. Only around 25% of the attacks and less than 5% of the normal payloads are left unclassified. This way, the confidence on the classified payloads will be high, but at the cost of an unacceptable number of unclassified payloads. The challenge is to further process these dubious payloads in order to increase the performance of the IDS.

Table 3. Performance of the system when using a threshold on the score of the recognizer

		Normal				Attack			
		FP		Unclassified		Undetected		Unclassified	
Experiment	Threshold	N.	%	N.	%	N.	%	N.	%
PVH vs. RDB	3.0	0	0.00	12425	2.01	2	0.22	243	26.15
PVH vs. OSVDB	1.0	2	$3 \cdot 10^{-4}$	12369	2.01	650	9.42	134	1.94
	2.0	2	$3 \cdot 10^{-4}$	12649	2.05	61	0.88	1089	15.79
	3.0	2	$3 \cdot 10^{-4}$	23537	3.82	14	0.20	1689	24.50

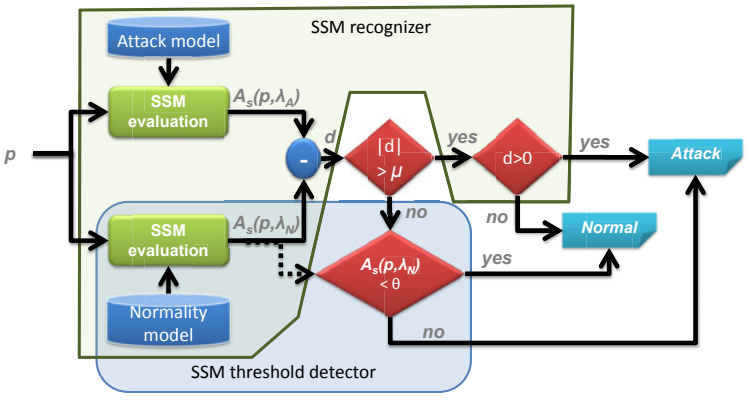


Fig. 6. Diagram of the proposed SSM-based IDS

5 Proposed Hybrid Detector

The proposed hybrid detector is a two steps detector in which the dubious payloads (Section 4.1) are passed through the basic SSM detector, that is, the normality score is considered to classify them (Fig. 6). The decision function then becomes,

$$class(p) = \begin{cases} normal, & if \begin{cases} P(p|\lambda_A) - P(p|\lambda_N) \leq -\mu \\ or \\ |P(p|\lambda_A) - P(p|\lambda_N)| < \mu \ and \ P(p|\lambda_N) \geq \Theta \end{cases} \\ anomalous, & if \begin{cases} P(p|\lambda_A) - P(p|\lambda_N) \geq \mu \\ or \\ |P(p|\lambda_A) - P(p|\lambda_N)| < \mu \ and \ P(p|\lambda_N) < \Theta \end{cases} \end{cases} \tag{8}$$

The reasoning behind this approach is to consider the original threshold-based system when the probabilities of being normal and attack are very similar. In this case, the discriminative information provided by comparing both probabilities is not enough to make and appropriate decision and is discarded. Therefore, the classification is made according to just the normality model. This implicitly considers a bigger confidence on the normality model than on the attack model. But this seems to be a good hypothesis if the number of training samples for both models is compared.

The experimental results obtained (Fig. 7) shows an improvement over the original SSM system. When compared with the recognizer (Section 4), this variant not only improves the performance but also allows the selection of the operating point of the detector through the choice of the parameters μ (confusion area) and Θ (normality score threshold). The added complexity in the detection is not relevant and a confidence on the classification can be set.

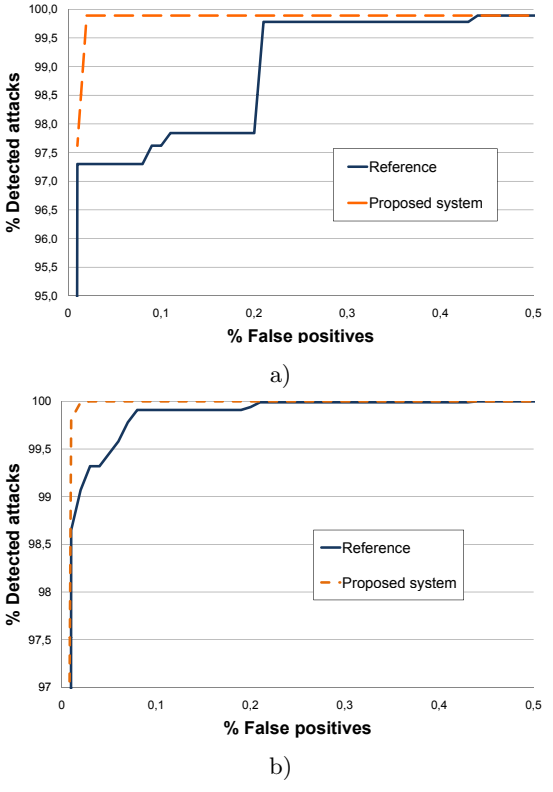


Fig. 7. Comparison of results for the basic SSM system and the proposed hybrid system: a) PVH vs. RDB, b) PVH vs. OSVDB

6 Conclusions

An enhanced version of the SSM method for intrusion detection has been proposed and evaluated. The new approach does not change the basics of this technique but uses it to model both the attack and the normal payloads. Therefore, two models (attack and normal) are considered as the basis of a recognizer, making this a hybrid IDS. The detection capabilities combine the skills of a SIDS and an AIDS. The known attacks will be explicitly modelled and detected while their new variants are expected to be included in the model. On the other hand, the system has the ability to detect novel attacks as they are expected to be different from normal payloads and their normality scores will be low.

The system is also able to provide a measure on the confidence of the classification. This measure can be used to alert the system administrator or to further analyze the payloads with low scores. An ongoing work is using this information during the training of the system to retrain the models according to the dubious payloads and/or increasing the representativeness of the dubious payloads during the estimation of the models. The idea is to improve the quality of the models

for the payloads in the confusion area in order to increase their discriminative capacities. The preliminary results are promising.

Acknowledgments

This work has been partially supported by Spanish MICINN under project TEC2008-06663-C03-02.

References

1. García-Teodoro, P., Díaz-Verdejo, J.E., Maciá-Fernández, G., Vázquez, E.: Anomaly-based Network Intrusion Detection: Techniques, Systems and Challenges. Computers & Security 28, 18–28 (2009)
2. Axelsson, S.: Intrusion Detection Systems: a Taxonomy and Survey, Technical Report 99-15, Department of Computer Engineering, Chalmers University of Technology, Goteborg (1999)
3. Sobh, T.S.: Wired and Wireless Intrusion Detection System: Classifications, Good Characteristics and State-of-the-art. Computer Standards & Interfaces 28, 670–694 (2006)
4. Depren, O., Topallar, M., Anarim, E., Kemal Ciliz, M.: An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. Expert Systems with Applications 29(4), 713–722 (2005)
5. Reis, M., Paula, F., Fernandes, D., Geus, P.: A Hybrid IDS Architecture Based on the Immune System. In: Anais do Wseg 2002: Workshop em Seguranca de Sistemas Computacionais, Buzios (2002), http://www.las.ic.unicamp.br/paulo/papers/2002-WSeg-marcelo.reis-fabricio.paula-diego.fernandes-IDS.imuno.pdf
6. Tombini, E., Debar, H., Me, L., Ducasse, M.: A serial combination of anomaly and misuse IDSes applied to HTTP traffic. In: 20th Annual Computer Security Applications Conference (2004)
7. Fontenelle, M.F., Siqueira, G., Holanda, R., Bessa Maia, J., Neuman, J.: Using Statistical Discriminators and Cluster Analysis to P2P and Attack Traffic Monitoring. In: LANOMS, pp. 68–77 (2007)
8. Estévez-Tapiador, J.M., García-Teodoro, P., Díaz-Verdejo, J.E.: Detection of Web-based Attacks Through Markovian Protocol Parsing. In: 10th Symposium on Computers and Communications, pp. 457–462 (2005)
9. Estévez-Tapiador, J.M.: Detección de intrusiones en redes basada en anomalías mediante técnicas de modelado de protocolos (Anomaly-based Network Intrusion Detection using protocol modelling techniques), Ph.D Thesis, Univ. of Granada (2003)
10. Symantec, Symantec Global Internet Security Threat Report, Trends for July-December 07, Volume XII (2008), http://eval.symantec.com/mktginfo/enterprise/white_papers/b-whitepaper_internet_security_threat_report_xiii_04-2008.en-us.pdf
11. Feller, W.: An Introduction to Probability Theory and its Applications, 3rd edn., vol. 1. John Wiley & Sons, Chichester (1968)
12. Berners-Lee, T., Fielding, R., Frystyk, H.: Hypertext Transfer Protocol – HTTP/1.0, RFC1945 (1996)

13. Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Berners-Lee, T.: Hypertext Transfer Protocol – HTTP/1.1, RFC2068 (1997)
14. Berners-Lee, T., Fielding, R., Masinter, L.: Uniform Resource Identifiers, RFC2396 (1998)
15. Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T.: Hypertext Transfer Protocol – HTTP/1.1, RFC2616 (1996)
16. Rabiner, L.R.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE 77(2), 257–285 (1989)
17. Bermúdez-Edo, M., Salazar-Hernández, R., Díaz-Verdejo, J.E., García-Teodoro, P.: Proposals on Assessment Environments for Anomaly-based Network Intrusion Detection Systems. In: López, J. (ed.) CRITIS 2006. LNCS, vol. 4347, pp. 210–221. Springer, Heidelberg (2006)
18. McHugh, J.: Testing Intrusion Detection Systems: a Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory. ACM Transactions on Information and System Security 3(4), 262–294 (2000)
19. Athanasiades, N., Abler, R., Levine, J., Owen, H., Riley, G.: Intrusion Detection Testing and Benchmarking Methodologies. In: Proc. 1st IEEE International Workshop on Information Assurance IWIA, pp. 63–72 (2003)
20. Duda, R., Hart, P.: Pattern Classification and Scene Analysis. John Wiley and Sons, Chichester (1973)
21. Provost, F., Fawcett, T., Kohavi, R.: The case against accuracy estimation for comparing induction algorithms. In: Proc. of the 15th International Conference on Machine Learning (ICML 1998). Morgan Kaufmann, San Mateo (1998)
22. Security Focus, Bugtraq (1998-2009), `http://www.securityfocus.com`
23. Kouns, J., Sullo, C., Martin, B., Shettler, D., Torino, S.: Open Source Vulnerability Data Base (2002-2009), `http://osvdb.org`
24. Salazar-Hernández, R., Díaz-Verdejo, J.: Generación de tráfico de ataque para la evaluación de sistemas de detección de intrusos. In: Actas de las VIII Jornadas de Ingeniería Telemática (JITEL 2009), pp. 439–442 (2009)