Proceedings of the 4th WSEAS Int. Conf. on Information Security, Communications and Computers, Tenerife, Spain, December 16-18, 2005 (pp299-305)

A BAYESIAN CLASSIFICATION MODEL FOR REAL TIME COMPUTER NETWORK INTRUSION DETECTION SYSTEMS

M. Mehdi, M. Bensebti, A. Anou and M. Djebari
Electronics Dept. University of Blida, Algeria

Abstract: - Intrusion detection systems (IDSs) attempt to identify attacks by comparing collected data to predefined signatures known to be malicious (misuse-based IDSs) or to a model of legal behaviour (anomaly-based IDSs). In this paper we present a new design of an anomaly IDS. Design and development of the IDS are considered in three main steps: normal behaviour construction, anomaly-based detection intrusion and model upgrading. A parametrical mixture model is used for behaviour modelling from reference data and the associated Bayesian classification. Real-time requirements are presented as well as detection and upgrade algorithms for the special case of Gaussian parametrical model and are evaluated with respect to their real-time features.

Keywords: computer, network, security, intrusion detection, real-time.

1. Introduction

Intrusion detection can be defined as the process of identifying malicious behaviour that targets a network and its resources. Intrusion detection systems have traditionally been classified as either *misuse-based* or *anomaly-based*. Systems that use misuse-based techniques contain a number of attack descriptions, or 'signatures', that are matched against a stream of audit data looking for evidence of the modelled attacks. The audit data can be gathered from the network [1], from the operating system, or from application log files. Signature-based systems have the advantage that they usually generate few false positives (i.e., incorrectly flagging an event as malicious when it is legitimate). Unfortunately, they can only detect those attacks that have been previously specified. That is, they cannot detect intrusions for which they do not have a defined signature.

Two major variants of intrusion detection systems (IDS) have emerged [2], namely host and network based approaches. Host based systems collect local data from sources internal to a computer, usually at the OS level. Network based approach variants monitor packets on the wire by setting the network interface to promiscuous mode and analysing network traffic [3].

The information system being protected (application, computer and/or network) is usually submitted to a usage configuration or policy that describes legitimate actions allowed to each entity (user, host or service) profile. Audit data describing entity actions or system states are generated (even systematically or trigged by the IDS) and, then, analyzed by the IDS, which evaluates the probability of these states or actions being related to an intrusion.

Data processed by the IDS may be a sequence of commands executed by a user, a sequence of system calls launched by an application (for example a web client), network packets, and so on. Finally, the IDS can trigger some countermeasures to eliminate attack cause/effect, whenever an intrusion is detected.

Concerning the analysis method, IDS are usually classified in two categories [1]. A misuse or knowledge-based IDS aims at detecting the occurrence of state or action sequences that has been previously identified to be an intrusion. Thus, in this kind of IDS, attacks must be known and described a priori and IDS are usually unable to deal with new or unknown attacks. Alternatively, an anomaly or behaviour-based IDS assumes that an intrusion can be detected by observing deviations from a normal or expected behaviour of a monitored entity. The valid behaviour is extracted from previous reference information about the system. The IDS later compares the extracted model with the current activity and raises an alert each time that a certain degree of divergence from the original model is observed.

In this paper, we propose a new model for intrusion detection following the anomaly detection approach. We are especially interested in information systems that are simultaneously submitted to different behaviour profiles. Typical examples are multi-user applications and computer systems or networks carrying different communication protocols. In such cases, audit data reflecting actions or system states associated with each system behaviour profile usually cannot be separated a priori. Moreover, it is sometimes even impossible to know how many profiles are present in the system behaviour. The proposed IDS model is intended to deal with both situations.

Anomaly IDS design consists of three main steps. First, it is necessary to build a reference behaviour model for the monitored system. In our case, this reference behaviour should be modelled from observed audit data describing the use of the system by a representative set of legitimate, non-malicious entities.

The aim is to model different entities profiles that could not be separated a priori by a learning procedure. A parametrical mixture model [4] is used to construct a Bayesian classification procedure on the observations and leads to the system behaviour model. Unsupervised learning is accomplished by fitting the mixture model parameters by the expectation-maximisation (EM) algorithm [4, 5]. As the model order may also be unknown, a minimum entropy criterion is introduced to allow model order estimation [6].

The next step consists in evaluating audit data related to new system activities to detect deviations between the current and the reference behaviours. New observed data is compared to the reference model by means of both a Bayesian classification and cluster pertinence evaluations.

As behaviour can change, the behaviour model should be updated during IDS operation. This is the last step. In our design, the model update is done by re-estimation of model parameters given new data presented to the system.

The design of the learning, detection and update phases using Bayesian techniques is the first contribution of the paper, the second lies in the discussion of real-time capabilities of the proposed algorithms, especially in the detection and update phase. Adaptations of the detection algorithm for the case of Gaussian mixture models are proposed, resulting in a linear complexity for the detection algorithm. Recursive parameter estimation is also proposed, as a possible alternative to real-time model update.

2. Bayesian classification IDS model

This section presents our anomaly IDS model. The idea is to build a behaviour model that takes into account multiple use profiles and allows a Bayesian classification of data as part of the detection algorithm. A reference audit data set representing the normal system behaviour is used to create the model with a learning procedure.

Before starting to describe the model, we should note that audit data must be mapped into random variables. Mapping audit data generated by the system to random variables, both during extraction of reference data in system behaviour modelling and system usage, is out of the scope of this discussion. Hereafter, we admit that audit data can be represented by a set of realisations of a continuous random vector R, whose probability distribution function (PDF) will have to be modelled.

2.1 Parametrical Mixture and EM-Algorithm

In our behaviour model, the PDF of the (d-dimensional) random vector y, whose realisations are mapped from the audit data domain, are represented by a parametrical mixture model [4, 5]. In such models, the realisations of y are regarded as being trials of one of the K simple models designed by a kernel probability function, with each kernel function representing the model of a use profile. Realisations from y are not clustered, e.g. the profile each realisation y_i comes from is not observable. The mixture model fundamental expression, giving the probability of y_i, can be formally expressed as:

$$p(y_i) = \sum_{k=1}^{K} p(z_k) g_k(y_i, \theta_k) \tag{1}$$

where: y_i is the i-th observed data; z is the hidden vector that indicates which source (profile) data comes from (e.g. z_k = 1 if data comes from cluster k and z = 0, otherwise); g_k are kernel distribution functions with respective parameters θ_k, each of them modelling one of the use profiles; K is the model order corresponding to the number of sources being modelled.

The unknown parameters in the model (Eq. (1)) are the set of cluster probabilities $p(z_k)$ and the parameters of kernel distribution functions of each cluster k, represented by

$$\mu = \left[p(z_1), p(z_2),....., p(z_k), \theta_1, \theta_2,...., \theta_k\right] \quad (2)$$

The (finite) mixture model represented by (1) has been increasingly used to model the distribution of a wide variety of supposed random phenomena [7]. An iterative algorithm of optimising the unknown vector μ by a maximum likelihood (ML) criterion has been defined and is called the expectation-maximisation (EM) algorithm [5, 6].

Let

$$Y = \left[y_1, y_2,....., y_k\right]^T \quad (3)$$

Where the subscripts 1, 2, n denote an observed n-dimensional realisation vector of y (that has to be modelled). Y is regarded as the reference data containing representative normal behaviour information and is used to fit μ using the EM algorithm. This algorithm permits both log-likelihood and model parameter estimation to be done in an iterative manner. The recursive process should be repeated until variation in the estimated log-likelihood between two consecutive iterations becomes small, indicating that the algorithm has converged to a (local) maximum of the log-likelihood data function, given that the realisation probabilities are expressed as in Eq. (1). As the log-likelihood function evaluation at the point represented by the parameter fitted by the EM-algorithm is not guaranteed to be a global maximum, a finite number of random initialisations of the parameters are realised and the EM-algorithm is executed different times. The results (parameter estimation) corresponding to a maximum log-likelihood evaluation in all executions are kept as the optimal model parameters and are used during detection phase.

A detailed discussion of the EM-algorithm is out of the scope of this paper, as it has already been extensively discussed in the literature. The reader is asked to refer to [5, 6] for a more general description of the EM-algorithm.

In the particular case of Gaussian mixture models (GMM), e.g. mixture model with Gaussian kernel functions, that were used in our experiments presented further, the Eq. (1) should be rewritten replacing the general distributions (g_k) by the normal distribution (represented by ϕ) and the distribution parameters θ_k by the mean vector (μ_k) and covariance matrix (R_k), as stated at Eq. (4), where the probability $p(z_k)$ are also replaced by the weighting factor w_k, for notation simplicity.

$$p(y_i) = \sum_{k=1}^{K} w_k \phi(y_i, \mu_k, R_k) \quad (4)$$

For completeness, we provide the EM recursion equations (Eq. (5)-(6)) for the Gaussian mixture models:

$$p(k \mid y_i) = \frac{w_k^i \phi(y_i, \mu_k^i, R_k^i)}{\sum_{k'=1}^{K} w_{k'}^i \phi(y_i, \mu_{k'}^i, R_{k'}^i)} \quad (5)$$

$$w_k^{i+1} = \sum_{i=1}^{n} p(k \mid y_i)/n \quad (6)$$

$$\mu_k^{i+1} = \sum_{i=1}^{n} p(k \mid y_i) y_i \Big/ \sum_{i=1}^{n} p(k \mid y_i) \quad (7)$$

$$R_k^{i+1} = \sum_{i=1}^{n} p(k \mid y_i)(y_i - \mu_k^{i+1})(y_i - \mu_k^{i+1})^T \Big/ \sum_{i=1}^{n} p(k \mid y_i) \quad (8)$$

2.2 Optimal Entropy-Based Estimation of Model Order

For the purpose of the EM-algorithm, the model order K (which corresponds to the number of partitions or data sources, when using parametric mixture models for partitioning data) must be provided. Since the number of partitions is not known a priori, it is useful to be able to estimate the most probable number of partitions, as well.

Our objective is to build an “ideal partitioning” estimation for K, which should be regarded as having the posterior probability p(k | y_i) (Eq. (5) in the GMM case) close to unity for one value of k and close to zero for all the others, for each realisation.

As described in [7], this ideal partitioning should be obtained by minimizing Shannon entropy given observed data, which can be evaluated for each observation by Eq. (9):

$$H_K = -\sum_{k=1}^{K} p(k \mid y_i)\log(p(k \mid y_i)) \tag{9}$$

The expected value of this entropy is evaluated taking the mean of H_K over all observed data (Eq. (10)):

$$E^*(H_K) = -\sum_{i=1}^{n}\sum_{k=1}^{K} p(k \mid y_i)\log(p(k \mid y_i))/n \tag{10}$$

Where: E* denotes an expectation estimator and H_K is the measure in question.

We proceed by fitting K_{max} models with different order (K = 1, 2,...K_{max}) and we evaluate the expected entropy (13) for each case. The resulting model in a minimum of this measure will be considered the optimum model.

The complete algorithm of the learning phase, used to obtain a mixture model fitted with the EM-algorithm and with optimal model order (K) estimation can be summarised as follows:

2.3 EM-Algorithm with Model Order Estimation

(1) K = 0, H_{opt} = 0, K_{opt} = 1.
(2) K = K+1.
(3) Fit the K-order model to data using the EM- Algorithm (Eqs. 2-7).
(4) Calculate expected value of H_K (Eq. (7)).
(5) If $H_K < H_{opt}$ then $H_{opt} = H_K$; K_{opt} = K; and μ = μ_{opt}.
(6) If K < K_{max}, then repeat (2).
(7) Update actual model order K with optimal model order: K = K_{opt}.
(8) Update actual model parameters μ with optimal model parameters μ_{opt}.

2.4 Anomaly Detection

During detection, the behaviour model has been already fitted and is available for making inferences about a new data presented to the system. The aim is to define some penalty λ, which varies from 0 to 1 (e.g. 0≤λ≤1), indicating the degree of normality concerning this realisation from certainly abnormal (λ= 0) to a certainly normal (λ = 1) behaviour.

Many different approaches for defining such criteria from the behaviour statistical model represented by Eq. (1) are possible. We have defined a detection procedure formed by two basic steps: a (Bayesian) classification inference and a cluster pertinence inference.

The classification inference is straightforward for parametrical mixture models and consists of evaluation of the posterior cluster probabilities conditioned to new data y’,

$$p(k \mid y'), \quad \mathit{for} \quad k = (1,2,....,K) \tag{11}$$

Cluster pertinence inference is more complex. As all the kernel distributions used in our model have a continuous nature, considering data posterior probabilities conditioned to cluster probability, p(y’|k), by simple evaluation of the cluster probability density function is meaningless. A more realistic approach consists in evaluating the probability of new data being contained in some pertinence interval (Π_k), defined as a function of cluster distribution parameters (μ_k and R_k, for instance) and the observation y’, which should be formally expressed as following (Eq. (12)):

$$p(y' \in \Pi_k \mid k) = \int g_k(y,\theta_k)d\Pi_k \tag{12}$$

Such probability should, indeed, look like some kind of cumulative distribution function (CDF), if we define Π_k as stated in Eq. (13), below [10]:

$$\Pi_k = \left| y \in \Re^d \middle| \frac{\|y-\mu_k\|^2}{\|R_k\|} \geq \gamma^2 \right| \tag{13}$$

Where $\| \ \|^2$ and $\| \ \|$ denote given types of norm operators and γ is a constant that should depend on y'.

Finally, detection penalty should be defined as Eq. (14):

$$\lambda(y') = \sum_{k=1}^{K} p(k \,|\, y') p(y' \in \Pi_k \,|\, k) \tag{14}$$

2.5 Model Upgrading

As behaviour may change, the behaviour model should be updated to avoid the raising of erroneous alerts (false positives). Updating should also be regarded as actualisation of smooth changes in system behaviour, as the basic model should become invalid or incomplete in case of expressive changes.

In our approach, we simply update the estimation of model parameters. Thus, updating is done in the cluster probabilities and in the kernel parameters. Usual estimators can be used for continuous estimation of these model statistics [10]. Note that both log-likelihood and entropy should also be estimated and compared with previous values (e.g. log-likelihood and entropy obtained after learning phase), as it could give an idea about the "goodness" of the new model when compared to the reference one.

3. Real-Time Considerations

The learning procedure in the reference behaviour model construction is usually executed off-line. Computation complexity constraints are not strong at this stage. However, it is usually desirable to have detection and update phases being executed continuously. Thus, the algorithms for detection and update should be designed for real-time. In this section, we show how detection and updating algorithms presented above should be adapted for real-time execution.

Although a formal performance evaluation of the proposed real-time algorithms are still in progress, we regard them as having a linear complexity, both with respect to the number of events (new data) presented to be analysed by the system and with respect to the model order.

The IDS is designed to be implemented as software for execution in a conventional PC platform, without need of any special hardware for enhancing computational capacity. Thus, for real-time evaluation, the processing power available for the IDS execution should be comparable with those of a standard PC. As a preliminary reference, IDS in use-intensive systems should deal with thousands or even millions of new events per second (e.g. packets in a high speed network or client requests on a busy web/application server).

3.1 Real-time Detection Algorithm with Gaussian Mixture Models

Eq. (12) cannot be usually evaluated analytically. A general solution should use numerical evaluation that can be prohibitive for higher dimensions. Besides, numerical evaluation is computation-intensive even in the one-dimensional or two-dimensional cases, making real-time execution difficult and even impossible.

Although Eq. (12) would be difficult for arbitrary kernel functions g_k, a computational-efficient algorithm for evaluating this integral equation can be established for the particular case of Gaussian distribution. These algorithms, which are discussed in this section, have been successfully used in the experiments presented in the next section, where we are essentially dealing with Gaussian mixture models.

Whenever GMM is being used, evaluation of the Eq. (12) can be done by convenient choice of the undefined elements (norm operators and γ) on Eq. (13).

The idea is to define Π_k as the complementary (concave) space of the isodensity ellipsoid (in $\Re^d$), whose boundary contains y' and is centred in μ_k. This means that Π_k is bounded internally by a d-dimensional ellipsoidal surface formed by all points having the same density as y'

$$(e.g.\quad \phi(y,\mu_k,R_k)=\phi(y',\mu_k,R_k)) \tag{15}$$

Thus, rewriting of Eq.(13) gives Equation (16):

$$\Pi_k=\left| y\in\Re^d \left|\sum_{\alpha\beta}(y_\alpha-\mu_\alpha)\left[R_k^{-1}\right]_{\alpha\beta}(y_\beta-\mu_\beta)\geq\gamma^2\right.\right| \tag{16}$$

Where:
$y=(y_1,y_2,......y_d)^T$; $\mu=(\mu_1,\ \mu_2,....\mu_d)^T$; $\left[R_k^{-1}\right]_{\alpha\beta}$ is the element at α-th line and β-th column of the inverse covariance matrix, and γ is done by (Eq. (17)):

$$\gamma^2=\sum_{\alpha\beta}(y'_\alpha-\mu_\alpha)\left[R_k^{-1}\right]_{\alpha\beta}(y_\beta-\mu_\beta) \tag{17}$$

This strategy is illustrated for one and two dimensional spaces as showed in Fig 1 and 2, respectively. The latter was taken from a bivariate Gaussian distribution, with diagonal covariance matrix.

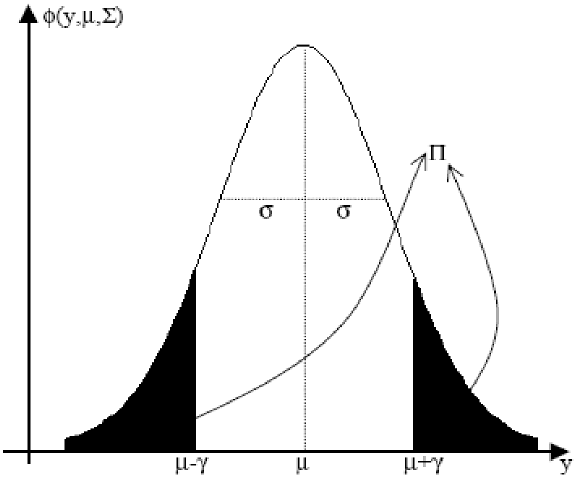

Fig1. Π for cluster with unidimensionnal Gaussian distribution

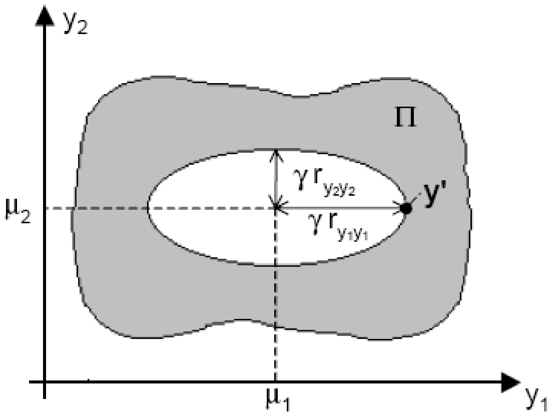

Fig2. Π for a cluster with bivariate Gaussian distribution with diagonal covariance matrix.

This procedure can be used even in the case of multivariate Gaussian distributions with unrestricted covariance matrix, as it is always possible to find a linear transformation that maps any multivariate Gaussian distribution in a equivalent new non correlated multivariate Gaussian distribution with same value for γ as in the former distribution.

As observed data can belong to a multidimensional space ($\Re^n$), a generalized distance γ', defined in Eq. (18), is introduced. This lads to a normalization of the probabilities expressed by Eq.(12) in data models belonging to different dimensional spaces, allowing computation of probabilities to be reduced to the one dimensional space, which can be executed by a simple lookup table procedure, making computational complexity feasible in real time.

4. Conclusion and Future Work

This text deals with a new anomaly IDS design using a parametric mixture model for behaviour modelling and Bayesian based detection. Continuous model update is accomplished by model parameter re-estimation. Algorithms for detection and update phases are designed for real-time operations. Preliminary experimentations show that proposed algorithms have some limitations such as that the kernel distributions are used to model numerical data with continuous and unbounded nature, the

Gaussian parametrical model may not be suitable for complex data and that the use of mixed models assumes statistical independence between trials, which can be restrictive in some cases. Despite these drawbacks the system presents real-time feasibility with no special hardware requirement. Moreover, it is being extended to detect security violations in a heterogeneous networked environment. The scalability, performance and fault tolerance can be improved when mobile agents perform distributed detection and do not need a central location where data is gathered. For instance, wireless networks such as Mobile Ad hoc NETworks are likely to be prone to security problems. The intrusion to the transmission support is relatively easy, compared to fixed networks

References:

[1] H. Debar, M. Dacier and A. Wespi, “A Revised Taxonomy for Intrusion-Detection Systems,” IBM Research Report, 1999.

[2] C. Krügel and T. Toth, “Flexible, mobile agent based intrusion detection for dynamic networks,” European Wireless 2002, Florence Italy (2002).

[3] H. Luo, P. Zerfos, J. Kong, S. Lu and L. Zhang, “Self Securing Ad Hoc Wireless Networks,” Proceeding of the 7th Inter. Symp. On computers and communications, ISCC’02(2002).

[4] P. Cheeseman and J. Stutz, “Bayesian classification (Auto Class): theory and results,” in Advances in Knowledge Discovery and Data Mining, edited by U. M. Fayyad et al., California: The AAAI Press, 1996, pp. 61-83.

[5] A. P. Dempster, N. M. Laird and D. B. Rubin, Journal of the Royal Statistical Society B 39 ,1-38 (1977).

[6] G. J. McLachlan, D. Peel, K. E. Basford and P. Adams, Journal of Statistical Software 04 (1999).

[7] S. J. Roberts, R. Everson and I. Rezek, Pattern Recognition, 33:5, pp. 833-839 (1999).

[8] Z. Marrakchi, “Détection d’Intrusion Comportementale dans les Systèmes à Objets Répartis: Modélisation des Séquences de Requêtes et de la Répartition de leurs Paramètres”, PhD Thesis, Université de Rennes I, 2002.

[9] Z. Marrakchi, L. Mé, B. Vivinis and B. Morin, “Flexible Intrusion-Detection Using Variable- Length Behaviour Modelling in Distributed Environment: Application to CORBA Objects”, Proceedings of 3rd International Workshop on the Recent Advances in Intrusion Detection (RAID 2000) 1907, 2000, pp. 130-144.

[10] R. A. Johnson, D.A. Wichern and D. W. Wichern, “Applied Multivariate Statistical Analysis – 4th Edition”, Prentice-Hall, 1998.