NEW SCHEMES FOR ANOMALY SCORE AGGREGATION AND THRESHOLDING

Salem Benferhat and Karim Tabia
CRIL - CNRS UMR8188, Université d'Artois, Rue Jean Souvraz SP 18 62307 Lens Cedex, France

Keywords: Anomaly intrusion detection, anomaly scoring and aggregating, thresholding, Bayesian networks.

Abstract: Anomaly-based approaches often require multiple profiles and models in order to characterize different aspects of normal behaviors. In particular, anomaly scores of audit events are obtained by aggregating several local anomaly scores. Remarkably, most works focus on profile/model definition while critical issues of anomaly measuring, aggregating and thresholding are dealt with "simplistically". This paper addresses the issue of anomaly scoring and aggregating which is a recurring problem in anomaly-based approaches. We propose a Bayesian-based scheme for aggregating anomaly scores in a multi-model approach and propose a two-stage thresholding scheme in order to meet real-time detection requirements. The basic idea of our scheme is the fact that anomalous behaviors induce either intra-model anomalies or inter-model anomalies. Our experimental studies, carried out on recent and real *http* traffic, show for instance that most attacks induce only intra-model anomalies and can be effectively detected in real-time.

1 INTRODUCTION

Intrusion detection aims at detecting any malicious action compromising integrity, confidentiality or availability of computer and network resources or services (Axelsson, 2000). Intrusion detection systems (IDSs) are either misuse-based such as SNORT (Snort, 2002) or anomaly-based such as EMERALD (Neumann and Porras, 1999) or a combination of both the approaches in order to exploit their mutual complementarities (Tombini et al., 2004). Anomaly approaches build profiles or models representing normal behaviors and detect intrusions by comparing current system activities with learnt profiles. In practice, anomaly-based IDSs are efficient in detecting new attacks but cause high false alarm rates which really encumbers the application of anomaly-based IDSs in real environments. In fact, configuring anomaly-based systems to acceptable false alarm rates result in failure to detect most malicious activities. However, a main advantage of anomaly detection lies in its potential capacity to detect both new and unknown (previously unseen) as well as known attacks.

Several anomaly-based systems use statistical profiles (Kruegel and Vigna, 2003) (Staniford et al., 2002) (Neumann and Porras, 1999) (Kruegel et al., 2005) to represent normal behaviors of network, host, user, program, etc. In most profile-based IDSs, anomaly score of a given audit event (network packet, system call, etc.) often depends on several local deviations measuring how much anomalous is the audit event with respect to the different normal profiles and models (Kruegel et al., 2005). Critical issues in statistical anomaly detection are normal profile/model definition and anomaly scoring and thresholding. The first issue is concerned with extracting and selecting features to analyze in order to detect anomalies. The second issue is also critical since it provides the anomaly scores determining whether audit events should be flagged normal or anomalous.

We believe that the problem of bad tradeoffs between detection rates and underlying false alarm ones characterizing most anomaly-based IDSs are in part due to problems in anomaly measuring, aggregating and thresholding methods. In this paper, we address drawbacks of existing methods for measuring and aggregating anomaly scores and anomaly thresholding. More precisely, we propose two schemes for anomaly thresholding suitable for multi-model anomaly-based approaches. The first scheme is a two-stage thresholding method aiming at effectively detecting intra-model anomalies as well as inter-model ones. The second thresholding scheme relies on ranking anomalous events according to their anomaly scores in order to cope with huge amounts of alerts characterizing most anomaly-based IDSs. As for anomaly score aggregation, we propose a Bayesian-based approach in order to exploit Bayesian network learning capa-

Benferhat S. and Tabia K. (2008).
NEW SCHEMES FOR ANOMALY SCORE AGGREGATION AND THRESHOLDING.
In *Proceedings of the International Conference on Security and Cryptography*, pages 21-28
DOI: 10.5220/0001927900210028

bilities. Moreover, Bayesian networks enable us to integrate expert knowledge. The proposed schemes overcome most existing methods' drawbacks. Experimental studies carried out on real and recent *http* traffic show the efficiency of our schemes.
The rest of this paper is organized as follows: Section 2 provides basic backgrounds about anomaly measuring and aggregating. It also points out problems with existing methods for anomaly measuring, aggregating and thresholding. A Bayesian-based approach for anomaly score aggregation and two thresholding schemes are proposed in section 3. In section 4, we present our experimental studies carried out on *http* traffic. Finally, section 5 concludes this paper.

2 RELATED WORK

SPADE (Staniford et al., 2002), NIDES(Javits and Valdes, 1993) are well-known anomaly-based IDSs where anomaly detection is ensured by computing deviations from normal activity profiles/models. Statistical profiles represent normal behaviors using statistical methods like frequencies, means, variances, etc. Then anomaly scoring functions evaluate the deviation of a given audit event with respect to learnt profiles. According to intrusion detection field, an audit event can be a packet or a connection in case of network-oriented intrusion detection, a system log record in case of host-oriented intrusion detection, a Web server log record in case of Web-oriented intrusion detection, etc.

2.1 Anomaly Measuring, Aggregating and Thresholding

Profile-based anomaly IDSs rely on the following elements:

1. **Profile/Model Definition:** Anomalous behaviors are those that do not conform to the expected normal behavior. Namely, there are aspects and characteristics of anomalous events which behave significantly different from known normal behaviors. Accordingly, normal profiles ideally consist in "all" features/aspects that can show differences between normal activities and abnormal ones. Note that most common form of audit events used in statistical-based IDSs are multivariate audit records describing network packets, connections, system calls, application log records, etc. These audit records involve different data types among which continuous and categorical data are common. In practice, several models and profiles are used in order to characterize the different aspects of normal behaviors.
2. **Anomaly Scoring Measures:** They are functions computing anomaly scores for every analyzed event. According to a fixed or learned threshold, an anomaly score associated with an event allows flagging it *normal* or *anomalous*. To compute such anomaly scores, anomaly scoring measures use the following functions:
   (a) **Set of "individual" (or "local") Anomaly Scoring Measures:** They are functions that evaluate the normality of audit event with respect to normal profiles individually. For example, in (Krugel et al., 2002) three statistical profiles represent normal *http* and *DNS* requests: *Request type profile*, *Request length profile* and *Character distribution profile*. Then three anomaly scoring functions are used in order to compute local anomaly scores. Most used anomaly measures are distance measures (which are widely used in outlier detection (Angiulli et al., 2006), clustering (Gerhard Mnz and Carle, 2007)), probability measures (Staniford et al., 2002), density measures (Ertz et al.,) and entropy measures (Lee and Xiang, 2001).
   (b) **Aggregating Functions:** Aggregating functions are used to fuse all individual anomaly scores into a single anomaly score which will be used to decide whether the analyzed event is normal or anomalous. Namely, a global anomaly score AS for an audit event E is computed using aggregating function G which aggregates all local anomaly scores $AS_{M_i}(E)$ relative to corresponding profiles/models M_i.

$$AS(E) = G(AS_{M_1}(E), AS_{M_2}(E).., AS_{M_n}(E)) \quad (1)$$

   In practice, aggregating functions range from simple summations (Javits and Valdes, 1993)(Krugel et al., 2002) to complex models such as Bayesian networks (Kruegel et al., 2003)(Staniford et al., 2002).
3. **Anomaly Thresholding:** Thresholding is needed to transform a numeric anomaly score into a symbolic value (*Normal* or *Anomalous*) in such a way an alert can be raised. Namely, thresholding is done by specifying value intervals for both normal and anomalous behaviors. Surprisingly, only few works addressed anomaly thresholding issues. In fact, some authors just use a single value(Krugel et al., 2002)(Staniford et al., 2002) to fix the limit between normal and abnormal scores while others use range of values to fix this limit and flag events as normal, abnormal or unknown. In prac-

tice, thresholds are often fixed according to the false alarm rate which must not be exceeded. Note that thresholds can be statically or dynamically set. The advantage of dynamically fixing a threshold is the ability to reassign its value in such a way to limit the amount of triggered alerts.

It is clear that the effectiveness of anomaly-based approaches strongly depend on profile/model definition and anomaly scoring measure relevance. In order to illustrate our ideas, we use a simple but widely used Web-based anomaly approach developed by Kruegel & Vigna (Kruegel and Vigna, 2003). These authors proposed a multi-model approach to detect web-based attacks relying on six detection models (Attribute length, Character distribution, Structural inference, Token finder, Attribute presence or absence and Attribute order). During detection phase, the six models output anomaly scores which are aggregated using a weighted sum. Recently, this model has been examined in depth in (Ingham and Inoue, 2007).

2.2 Drawbacks of Existing Schemes for Anomaly Measuring, Aggregating and Thresholding

Existing anomaly measuring, aggregating and thresholding methods suffer from several problems:

- **Probability Distribution Assumption Problems:** This problem is particularly encountered in mean and variance models (Denning, 1987) and anomaly measures using probability measures. For example, anomaly score relative to attribute length model in *Krugel & Vigna* model is proportional to the difference from the mean length μ. However, attributes with lesser lengths ($l \lll \mu$) are scored like attributes whose lengths are exceeding μ ($l \ggg \mu$). However, since anomalousness caused by attribute lengths are mostly due to oversized values, then anomaly measure relative to attribute length should handle differently oversized and undersized values. Basically, the problem is due to assuming that normal values follow a Gaussian distribution while this assumption is not valid in many detection models.
- **Frequency Bias:** Most frequency-based anomaly measures often associate significantly different anomaly scores to typically normal behaviors. For example, in (Krugel et al., 2002), authors use three models in order to detect anomalies in *http* requests. In this work, anomaly score relative to request type (GET, POST, HEAD, etc.) is proportional to the frequency of each request method in training data. However, consider that *GET* requests represent 95% while *POST* ones represent 3% (remaining proportion represents other request types). Then anomaly score of a *POST* request will be hundred times bigger than a *GET* score. However, all of them are typically normal request types present in training data.
- **Anomaly Score Aggregation:** As mentioned above, aggregating anomaly scores is done in most cases using "simplistic" methods (Kruegel et al., 2003). For instance, most used aggregation scheme is the weighted sum-based method which suffers from several problems such as:
  1. Firstly, weighting local anomaly scores is often done in a "questionable" way. For example, authors in (Krugel et al., 2002) neither explained how they assign weights nor why they use same weighting for *http* and *DNS* requests.
  2. The accumulation phenomena which causes several small local anomaly scores to cause, once summed, a high global anomaly score.
  3. The averaging phenomena which causes a very high local anomaly score to cause, once aggregated, a low global anomaly score.
  4. Commensurability problems are encountered when different detection model outputs do not share the same scale. Then some anomaly scores will have much more importance in the overall score than others.
  5. Ignoring inter-model dependencies existing between the different detection models.
- **Thresholding:** This problem is basically due to the fact that the border line between normal and anomalous behaviors is not well precise. Moreover, this problem is impacted by the quality of features, models and measures used to evaluate the normality of audit events.
- **Real-time Detection Capabilities:** The decision of raising an alert is taken on the basis of the global anomaly score which requires computing all local anomaly scores then aggregating them. This method causes several problems especially for effectiveness considerations. For example, when analyzing buffer-overflow attacks, the request length can be sufficient and there is not need to compute the other anomaly scores. Moreover, in buffer-overflow attacks, the request is often segmented over several packets which are reassembled at the destination host. However, such attack can be detected given the first packets of the request and there not need to wait for all packets in order to detect such an anomaly.
- **Handling Missing Inputs:** Missing data is an important issue that existing systems have not

dealt with conveniently. In fact, many intentional or accidental causes can provoke the missing of some data pieces. For example, in gigabyte networks, network packet sniffer may drop packets. Though, when applied to network traffic, how can the model proposed in (Krugel et al., 2002) deal with a request if the sniffer dropped the packet containing the request method? The problem is how to analyze audit events given that some inputs are missing.

3 NEW SCHEMES FOR ANOMALY SCORE AGGREGATING AND THRESHOLDING

In this section, we propose new schemes for aggregating anomaly scores and thresholding suitable for multi-model anomaly detection approaches.

3.1 What is "Anomalous Behavior"

The premise of anomaly-based approaches is the assumption that attacks induce abnormal behaviors. There are different possibilities about how anomalous events affect and manifest through elementary features. For instance, anomalous events can be in the form of anomalous (new or outlier) value in a feature, anomalous combination of known normal values or anomalous sequence of events. Accordingly, alerts raised by a multi-model anomaly-based approach can be caused by two anomaly categories:

- **Intra-model Anomalies:** They are anomalous behaviors affecting one single model. Namely, the anomaly evidence is obvious only through one detection model. For example, in *Krugel & Vigna* model, there are buffer-overflow attacks which heavily affect the length model without affecting the other models. Then anomaly score computed using length model should suffice in order to detect such attacks.
- **Inter-model Anomalies:** They are anomalies that affect regularities and correlations existing between different models. For instance, in *Krugel & Vigna* model, authors pointed out correlations between *Length* model and *Character distribution* model. Then audit events violating such regularities are anomalous.

It is obvious that intra-model anomalies can be detected without aggregating the different anomaly scores. Moreover, this is interesting because such anomalies can be detected in real-time. In fact, any anomaly revealed by a detection model is sufficient to raise an alert even if other detection models have not yet returned their anomaly scores. This is the idea motivating the multi-stage thresholding scheme. Namely, each detection model has its own anomaly threshold T_{M_i}. During the detection phase, once input data for detection model M_i is available, then the system can trigger an alert whenever anomaly score $As_{M_i}(E)$ exceeds corresponding threshold T_{M_i}. If no intra-model anomaly is detected, then we need to look for inter-model anomalies.

3.2 New Thresholding Schemes

In the following, we propose a two-stage thresholding scheme in order to effectively detect intra-model and inter-model anomalies and a ranking-based thresholding scheme for coping with large amounts of alerts characterizing most anomaly-based IDSs.

3.2.1 Scheme 1: Local vs Global Thresholding

Since anomalous events can either affect detection models individually or violate regularities existing between detection models, then we propose a two-stage thresholding scheme aiming at raising an alert whenever an anomalous behavior occurs be it intra-model or inter-model.

- In order to detect intra-model anomalies, we fix for each detection model M_i a local anomaly threshold in the following way:

$$Threshold_{M_i} = Max(As_{M_i}(E_{Normal})) * \theta \quad (2)$$

  Threshold $Threshold_{M_i}$ associated with detection model M_i is set to the maximum among all anomaly scores computed on normal training behaviors E_{Normal}. θ denotes a discounting/enhancing factor in order to control detection rate and underlying false alarm rate. In case when no intra-model anomaly is detected, then we need to check for inter-model anomalies.
- Similarly to intra-model thresholding, a threshold can be fixed for global anomaly score as follows:

$$Threshold = Max(As(E_{Normal})) * \theta \quad (3)$$

Note that term $As(E_{Normal})$ denotes the anomaly score aggregating function and E_{Normal} denotes a normal audit event. In order to control detection rate/false alarm rate tradeoff, one can use the discounting/enhancing parameter θ.
Local and global thresholding schemes can be combined in order to exploit their complementarities:

- Real-time detection: With local thresholding, every intra-model anomaly is detected without waiting for other detection model results.
- Handling missing inputs: Missing inputs only affect models requiring these input. Then remaining models can work normally and detect intra-model anomalies.
- Intra-model and inter-model anomaly detection: As we will see in experimental studies, combining local with global thresholding allows detecting more effectively both intra-model and inter-model anomalies.

Note that the motivation of setting the anomaly thresholds to the maximum among all anomaly scores computed on normal training behaviors is to detect any event whose anomaly score exceeds all normal behavior scores used to build the detection models. This maximum-based thresholding is intuitive and does not require any assumption about anomaly scores. In fact, the greatest anomaly score on training behaviors is the one associated with normal but unusual behavior. Then behaviors having greater anomaly score are anomalous.

3.2.2 Scheme 2: Ranking-based Thresholding

In many domains and environments, security administrators know from experience that there is always some percentage of behaviors that are not totally normal. This is for instance what happens with zero-day attacks where vulnerabilities are exploited before security patches are released. Moreover, security administrators are often incapable to manually analyze the whole amount of triggered alerts. Hence, they prefer to focus only on most anomalous behaviors. Accordingly, instead of just flagging events normal or anomalous according to a fixed threshold, we propose to rank anomalous audit events according to their anomaly scores. Then security administrator can analyze alerts according to anomaly score ranking. This simple method has several advantages:

- The administrator can firstly analyze most anomalous events and the amount of events he wants.
- Coping with zero-day attack problem since there will always be events causing alerts.
- There is not need to fix any anomaly threshold.

However, this thresholding scheme is more suitable for off-line analysis than real-time one. In off-line detection, this method returns the top n% anomalous events or a ranking of most anomalous events.

3.3 Bayesian-based Aggregation

Bayesian networks (BN) are powerful graphical models for representing and reasoning under uncertainty conditions (Jensen, 1996). They consist of a graphical component DAG (Directed Acyclic Graph) and a quantitative probabilistic one. The graphical component allows an easy representation of domain knowledge in the form of an influence network (vertices represent events while edges represent "influence" relations between these events). The probabilistic component expresses uncertainty relative to relationships between domain variables using conditional probability tables (CPTs). Learning Bayesian networks requires training data to learn structure and compute the conditional probability tables. Note that several works used BN for anomaly detection (Gowadia et al., 2005)(Staniford et al., 2002)(Valdes and Skinner, 2000). For instance, authors in (Kruegel et al., 2003) used a BN in order to assess the anomalousness of system calls. In our case, main advantages of BN are learning capabilities in order for instance to extract inter-model regularities and inference capacities which are very effective. Moreover, BN can combine user-supplied structure with empirical data.

3.3.1 Training the Bayesian Network: Extracting Intra-model and Inter-model Regularities

Given a data set of m normal audit events E_{Normal}, we build a data set of anomaly score vectors (A_1, A_2,.., A_m) where each anomaly vector is composed of all local anomaly scores (namely $A_i = (a_{i1}, .., a_{in})$ corresponds to anomaly vector relative to normal audit event E_{Normal} with respect to detection models M_1,..,M_n and anomaly measure As_{M_1},..,As_{M_n} respectively). Then learning a BN from these anomaly vectors will learn intra-model regularities as well as inter-model ones. Then network structure qualitatively represents inter-model regularities while conditional probability tables quantify inter-model influences. Note that the structure can be specified by domain expert in order to fix detection model dependencies according to expert knowledge.

3.3.2 Detection using the Bayesian Network

Once the BN built, it can be used to compute the probability of any anomaly vector. We first compute the different anomaly scores then using the BN, we compute the probability of the current anomaly vector. The normality of audit event E is proportional to the probability of the corresponding anomaly vector.

The anomaly threshold can be fixed as follows:

$$Threshold = Max(1 - p_{BN}(A_1, A_2, .., A_m)) * \theta \quad (4)$$

Term p_{BN} in Equation 4 denotes the probability degree computed using BN. This threshold flags anomalous any event having a probability degree smaller than the most improbable normal training event.

4 EXPERIMENTAL STUDIES

In order to evaluate our anomaly aggregating and thresholding schemes, we use a multi-model approach designed to detect anomalies and attacks against server-side and client-side Web applications (Benferhat and Tabia, 2008). The detection models are built on real and recent attack-free *http* traffic and evaluated on real and simulated *http* traffic involving normal data as well as several Web-based attacks.

4.1 Detection Model Definition

Our experimental studies are carried out on Web-based attack detection problem which represents major part of nowadays cyber-attacks. In (Benferhat and Tabia, 2008), authors proposed a set of detection models including basic features of *http* connections as well as derived features summarizing past *http* connections and providing useful information for revealing suspicious behaviors involving several *http* connections. Note that detection model's inputs are directly extracted from network packets instead of using Web application logs. Processing whole *http* traffic is the only way for detecting suspicious activities and attacks targeting either server-side or client-side Web applications. The detection model features are grouped into four categories:

1. **Request General Features:** They are features that provide general information on *http* requests. Examples of such features are request method, request length, etc.
2. **Request Content Features:** These features search for particularly suspicious patterns in *http* requests. The number of non printable/metacharacters, number of directory traversal patterns, etc. are examples of features describing request content.
3. **Response Features:** Response features are computed by analyzing the *http* response to a given request. Examples of these features are response code, response time, etc.
4. **Request History Features:** They are statistics about past connections given that several Web attacks such as flooding, brute-force, Web vulnerability scans perform through several repetitive connections. Examples of such features are the number/rate of connections issued by same source host and requesting same/different URIs.

Note that in our experimentations, we consider each feature as a detection model. Then numeric features are modeled by their means μ and standard deviations σ while nominal and boolean features are represented by the frequencies of possible values. During the detection phase, anomaly score associated with a given *http* connection lies in the local anomaly scores of the connection features with respect to the learnt profiles. We use different anomaly measures according to each profile type(numeric, nominal or boolean) and its distribution in training data. It is important to note that most numeric features in training data have rather exponential distributions than Gaussian ones. In order to compute anomaly score of a given feature F_i with respect to the corresponding detection model M_i, we consider two cases:

- if F_i is numerical then the anomaly score is computed as follows:

$$As_{M_i}(F_i) = e^{\frac{F_i - \mu_i}{\sigma_i}} \quad (5)$$

  Terms μ_i and σ_i denote respectively the mean and standard deviation of feature F_i in normal data. σ_i is used as a normalization parameter. Note that only exceeding values cause high anomaly scores. Intuitively, if the value of F_i is less, equal or closer to the average μ_i then the anomaly score will be negligible. Otherwise, the wider the margin, the greater will the anomaly score.

- if F_i is a boolean or symbolic feature then the anomaly score is computed according to the improbability of the value of F_i in normal training data. Namely,

$$As_{M_i}(F_i) = -log(p(F_i)) \quad (6)$$

  Term $p(F_i)$ denotes the frequency of F_i's value in normal training data. Intuitively again, the more exceptional is the value of F_i in training data, the higher will be the anomaly score. Conversely, frequent and usual values will be associated with low anomaly scores.

4.2 Training and Testing Data

Our experimental studies are carried out on a real *http* traffic collected on a University campus during 2007. Note that this traffic includes both inbound and outbound *http* connections. We extracted *http* traffic and preprocessed it into connection records using only packet payloads. As for attacks, we simulated most of the attacks involved in (Ingham and Inoue, 2007) which is to our knowledge the most extensive and uptodate open Web-attack data set.

Attacks of Table 1 are categorized according to the vulnerability category involved in each attack. Re-

Table 1: Training/testing data set distribution.

	Training data		Testing data	
Class	**Number**	**%**	**Number**	**%**
Normal connections	55342	100%	61378	58.41%
Buffer overflow	–	–	18	0.02%
Input validation	–	–	46	0.04%
Value misinterpretation	–	–	2	0.001%
Poor management	–	–	3	0.001%
Flooding	–	–	12485	11.88%
Vulnerability scan	–	–	31152	29.64%
Cross Site Scripting	–	–	6	0.01%
SQL injection	–	–	14	0.01%
Command injection	–	–	9	0.01%
Total	**55342**	**100%**	**105084**	**100%**

garding attacks effects, attacks of Table 1 include denial of service attacks, Scans, information leak, unauthorized and remote access (Ingham and Inoue, 2007).

4.3 Comparison of Thresholding and Aggregation Schemes

Table 2 compares results of different thresholding and aggregation schemes described in section 3. Note that the different schemes compared in Table 2 are:

- **Non Weighted Sum-based Aggregation:** This is a standard scheme using a non weighted sum and a maximum-based global threshold (see Equation 3). It is used as a reference scheme for evaluating our aggregation and thresholding ones.
- **Local Thresholding:** This scheme aims at detecting intra-model anomalies and it is relies on thresholding of Equation 2.
- **Global Thresholding:** Global thresholding aims at detecting anomalies violating inter-model regularities. We used a BN built on anomaly score records computed for audit event using the different detection models. Note that structure learning is performed using the hill-climbing algorithm (Heckerman et al., 1995). We fixed anomaly thresholds according to Equation 4.
- **Local+Global Thresholding:** This scheme takes advantage of both local and global thresholding schemes in order to detect both intra-model and inter-model anomalies.

Note that all the anomaly thresholds are computed on normal training data and we do not use any discounting/enhancing parameter θ (θ=1). Table 2 compares on one hand results of a sum-based aggregation using a single global threshold with a sum-based aggregation combined with local and global thresholding. On the other hand, we evaluate the Bayesian-based approach using a single global threshold and the combination of the local and global thresholding with Bayesian-based aggregation.

Table 2: Evaluation of different aggregation/threshodling schemes on *http* traffic.

Audit event class	**Sum-based aggreg**	**local thresh**	**Sum aggreg+ local thresh**	**Bayes aggreg**	**Bayes aggreg+ local thresh**
Normal connections	99.94%	97.37%	97.37%	99.79%	99.66%
Buffer overflow	16.67%	94.44%	94.44%	27.78%	94.44%
Input validation	2.17%	86.96%	86.96%	23.91%	91.30%
Value misinterpretation	100%	100%	100%	50%	100%
Poor management	100%	100%	100%	66.67%	100%
Flooding	95.46%	99.62%	99.62%	86.22%	99.93%
Vulnerability scan	0.00%	51.84%	51.84%	83.06%	90.56%
Cross Site Scripting	0.00%	100%	100%	100%	100%
SQL injection	0.00%	100%	100%	100%	100%
Command injection	0.00%	100%	100%	100%	100%
Total	69.72%	84.16%	84.16%	93.20%	97.02 %

Firstly, Table 2 shows that our schemes perform better than the reference sum-based scheme. Moreover, it is important to note that most attacks induce only intra-model anomalies and can be detected without any aggregation. In fact, the combination of sum-based scheme with local thresholding significantly enhances the detection rates without triggering higher false alarm rates. Similarly, Bayesian aggregation enhanced with global thresholding achieves better results regarding detection rates and false alarm rate. Note that best results are achieved by Bayesian aggregation combined with local and global thresholding schemes (see correct classification rates over normal connections and Web attacks). This is due to the fact that this scheme detects both intra-model and inter-model regularities learnt by the Bayesian network.

4.4 Evaluation of Ranking-based Thresholding

Table 3 provides results of ranking-based thresholding evaluation on *http* traffic involving normal traffic and several Web-based attacks (see Table 1). For different anomaly thresholds, Table 3 shows the true positive rate (attacks for which alerts are raised) and underlying false alarm rate.

Table 3: Evaluation of ranking-based thresholding on *http* traffic.

Threshold	**0.1%**	**1%**	**2%**	**3%**	**4%**	**5%**	**10%**
True positive rate	100%	99.4%	98.4%	97.2%	96.3%	94.1%	92.7%
False alarm rate	0%	0.57%	1.51%	2.73%	3.63%	5.89%	7.24%

It is important to note that this evaluation is carried out in off-line mode. Results of Table 3 clearly show that when ranked according to anomaly scores, most anomalous events are actually attacks. For instance, when anomaly threshold is set to 0.1% of analyzed events, then all the triggered alerts are actually caused by attacks. Setting the anomaly threshold to greater values causes true positive rate to decrease slightly while false alarm rate proportionally increases. Note that most false alarms correspond to new and unusual audit events. Given that security administrators can only check small amounts of alerts, then ranking-based thresholding is an interesting scheme since it focuses on most anomalous events.

5 CONCLUSIONS

The main objective of this paper is to address anomaly thresholding and aggregating issues in multi-model anomaly detection approaches. We proposed a two-stage thresholding scheme suitable for detecting in real-time intra-model and inter-model anomalies. In order to cope with large numbers of alerts characterizing most anomaly-based IDSs, we proposed a ranking-based thresholding method allowing to limit the alert quantities while focusing on most anomalous events. As for anomaly score aggregation, we proposed to use a Bayesian network whose structure can be fixed by the expert or extracted automatically from attack-free training data. Experimental studies carried out on real and recent *http* traffic showed that most Web-related attacks induce intra-model anomalies and can be detected in real-time using local thresholding scheme. Future works will explore the application of our schemes in order to detect anomalies and attacks when input data relative to audit event is uncertain or missing.

ACKNOWLEDGEMENTS

This work is supported by a French national project entitled DADDi.

REFERENCES

Angiulli, F., Basta, S., and Pizzuti, C. (2006). Distance-based detection and prediction of outliers. *IEEE Trans. on Knowl. and Data Eng.*, 18(2):145–160.

Axelsson, S. (2000). Intrusion detection systems: A survey and taxonomy. Technical Report 99-15, Chalmers Univ.

Benferhat, S. and Tabia, K. (2008). Classification features for detecting server-side and client-side web attacks. In *23rd International Security Conference*, Italy.

Denning, D. E. (1987). An intrusion-detection model. *IEEE Trans. Softw. Eng.*, 13(2):222–232.

Ertz, L., Eilertson, E., Lazarevic, A., Tan, P.-N., Kumar, V., Srivastava, J., and Dokas, P. Minds - minnesota intrusion detection system.

Gerhard Mnz, S. L. and Carle, G. (2007). Traffic anomaly detection using k-means clustering.

Gowadia, V., Farkas, C., and Valtorta, M. (2005). Paid: A probabilistic agent-based intrusion detection system. *Computers & Security*, 24(7):529–545.

Heckerman, D., Geiger, D., and Chickering, D. M. (1995). Learning bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243.

Ingham, K. L. and Inoue, H. (2007). Comparing anomaly detection techniques for http. In *RAID*, pages 42–62.

Javits and Valdes (1993). The NIDES statistical component: Description and justification.

Jensen, F. V. (1996). *An Introduction to Bayesian Networks*. UCL press.

Kruegel, C., Mutz, D., Robertson, W., and Valeur, F. (2003). Bayesian event classification for intrusion detection. In *Proceedings of the 19th Annual Computer Security Applications Conference*, page 14, USA.

Kruegel, C. and Vigna, G. (2003). Anomaly detection of web-based attacks. In *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*, pages 251–261, New York, NY, USA.

Kruegel, C., Vigna, G., and Robertson, W. (2005). A multi-model approach to the detection of web-based attacks. volume 48, pages 717–738.

Krugel, C., Toth, T., and Kirda, E. (2002). Service specific anomaly detection for network intrusion detection. In *Proceedings of the 2002 ACM symposium on Applied computing*, pages 201–208, USA.

Lee, W. and Xiang, D. (2001). Information-theoretic measures for anomaly detection. In *Proceedings of the IEEE Symposium on Security and Privacy*, USA.

Neumann, P. G. and Porras, P. A. (1999). Experience with EMERALD to date. In *First USENIX Workshop on Intrusion Detection and Network Monitoring*, pages 73–80, Santa Clara, California.

Snort (2002). Snort: The open source network intrusion detection system. http://www.snort.org.

Staniford, S., Hoagland, J. A., and McAlerney, J. M. (2002). Practical automated detection of stealthy portscans. *J. Comput. Secur.*, 10(1-2):105–136.

Tombini, E., Debar, H., Me, L., and Ducasse, M. (2004). A serial combination of anomaly and misuse idses applied to http traffic. In *Proceedings of the 20th Annual Computer Security Applications Conference*, pages 428–437.

Valdes, A. and Skinner, K. (2000). Adaptive, model-based monitoring for cyber attack detection. In *Proceedings of the Third International Workshop on Recent Advances in Intrusion Detection*, pages 80–92, UK.