Computer Networks 57 (2013) 811–824

Contents lists available at SciVerse ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Review Article

RePIDS: A multi tier Real-time Payload-based Intrusion Detection System

Aruna Jamdagni [a,b], Zhiyuan Tan [a,b], Xiangjian He [a,*], Priyadarsi Nanda [a], Ren Ping Liu [b]

[a] Centre for Innovation in IT Services and Applications (iNEXT), University of Technology, Sydney, Australia
[b] ICT Centre, CSIRO, Australia

ARTICLE INFO

Article history:
Received 9 January 2012
Received in revised form 7 August 2012
Accepted 7 October 2012
Available online 26 October 2012

Keywords:
Intrusion detection
Data pre-processing
Principal component analysis
Mahalanobis Distance Map
Principal components
Iterative feature selection

ABSTRACT

Intrusion Detection System (IDS) deals with huge amount of network traffic and uses large feature set to discriminate normal pattern and intrusive pattern. However, most of existing systems lack the ability to process data for real-time anomaly detection. In this paper, we propose a 3-Tier Iterative Feature Selection Engine (IFSEng) for feature subspace selection. Principal Component Analysis (PCA) technique is used for the pre-processing of data. Mahalanobis Distance Map (MDM) is used to discover hidden correlations between the features and between the packets. We also propose a novel Real-time Payload-based Intrusion Detection System (RePIDS) that integrates a 3-Tier IFSEng and the MDM approach. Mahalanobis Distance (MD) dissimilarity criterion is used to classify each packet as either a normal or an attack packet.

The effectiveness of the proposed RePIDS is evaluated using DARPA 99 dataset and Georgia Institute of Technology attack dataset. The traffic for Web-based application is considered for validating our model. *F*-value, a criterion, is used to evaluate the detection performance of RePIDS. Experimental results show that RePIDS achieves better performance (high *F*-values, 0.9958 for DARPA 99 dataset and 0.976 for Georgia Institute of Technology attack dataset respectively, with only 0.85% false alarm rate) and lower computational complexity when compared against two state-of-the-art payload-based intrusion detection systems. Additionally, it has 1.3 time higher throughput in comparison with real scenario of medium sized enterprise network.

1. Introduction

Computer security has become a critical issue with the rapid development of business and transaction systems over the Internet and has gained worldwide attention as attacks to computer network systems become more widespread and sophisticated. Since traditional prevention measures are imperfect, monitoring for security compromises is required. Intrusion Detection System (IDS) is an important component of security mechanism. The goal of an IDS is to provide a layer of defense against malicious uses of computer systems by sensing and alerting operators the ongoing attacks. More sophisticated IDSs generally fall into two categories: misuse detection (or signature detection) and anomaly detection. Misuse-based IDSs, such as SNORT and Bro, commonly rely on rules written by domain experts and look for a match of an attack signature. Misuse detection systems have higher detection accuracies. However, the major problems of these systems are that they fail to detect new attacks or attacks whose signatures are not known and require continual signature updates to detect the latest attacks. Comparatively, anomaly-based IDSs learn the normal behavior of a user, look for anomalous patterns as significant deviations from the normal behavior of a user profile, and generate alarms. These systems can detect new attacks and variants of known attacks. Unfortunately, they are prone to false

* Corresponding author.
E-mail addresses: arunaj@it.uts.edu.au (A. Jamdagni), Zhiyuan.Tan@uts.edu.au (Z. Tan), Xiangjian.He@uts.edu.au (X. He), Priyadarsi.Nanda@uts.edu.au (P. Nanda), ren.liu@csiro.au (R.P. Liu).

1389-1286/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comnet.2012.10.002

positives which can be triggered by novel but non-malicious traffic. False positives limit the performance of anomaly-based IDSs.

Anomaly detection has been an active research area for more than two decades since it was originally proposed by Denning [1]. Reviews on Network-based Intrusion Detection System (NIDS) are given in the literature [2–7]. Reviews indicate that most previous research works in anomaly detection do not concern about the data pre-processing technique used in NIDS. Intrusion detection algorithms are used directly on the rough network data and do not mention criteria for selection of traffic features for intrusion detection. For practical applications, data pre-processing is one of the most important stages in the development of detection algorithm, and it directly impacts the accuracy and the capability of a classification algorithm.

In an ideal situation, the purpose of an IDS is to detect attacks at an early stage or in other words in real-time to minimize the impacts of attacks. Hence, for real-time intrusion detection, the system must detect an attack as soon as it is commenced. However, in real practice, it is very difficult to build such a system with low false alarm rates and high detection accuracy. In general, IDS deals with huge amount of data which contains irrelevant and redundant features causing slow training and test processes, higher resource consumption as well as poor detection rates. Thus, selecting important and suitable features, which characterize behavioral patterns of network traffic and clearly distinguish normal and abnormal activities from network traffic data, is one of the key challenges to build a real-time IDS.

Though methods for deriving discriminating features from packet headers are well established as demonstrated in [8–12], approaches for packet payload are less well defined. From the reviewed research on payload-based anomaly detection, *n*-grams [13] and libAnomaly [14] are two commonly used methods. The drawbacks of these methods are that they have to use very large size of feature sets, so they fail to provide sufficient discriminative power for correct traffic discrimination and lead to relatively high false alarms rates. Furthermore, payload-based attacks are computationally expensive to detect due to requiring deeper searches into network sessions and looking for huge number of payload features. This challenge has motivated our research work to build a real-time payload-based intrusion detection system using suitable feature subset, in order to detect attacks as soon as are commenced.

Therefore, in this paper, we address the issues related to the quality of feature set and the curse of dimensionality (data pre-processing). We also propose a real-time payload-based anomaly IDS using efficient iterative feature selection scheme. The main contributions of our work in this paper are as follows.

Firstly, we propose a 3-Tier Iterative Feature Selection Engine (IFSEng) for feature subset selection. Analysis of the raw dataset is conducted in Tier 1 using Principal Component Analysis (PCA) technique, which examines and rates the importance of various components of a multi-dimensional feature space in terms of variance that a component reserves. Mathematical solutions (cumulative power and parallel analysis criteria) and non-mathematical solution (scree test criterion) are applied independently at Tier 2 to determine the number of dominant Principal Components (PCs), which should be retained according the analysis results from Tier 1. The refinement of features (dominant PCs), and the generation and the verification of a normal training model are performed at Tier 3.

Secondly, we propose to use Mahalanobis Distance Map (MDM) approach to identify patterns of packet payloads. MDM is promising in extracting the hidden correlations between features and the correlations among network packet payloads. It also partially captures structural information of payload. These correlations and structural information help improve the detection performance and reduce false positive rates.

Thirdly, we propose a Real-time Payload-based Network Intrusion Detection System (RePIDS), which detects payload-based attacks on the network in real-time. As the key components of RePIDS, 3-Tier IFSEng and MDM facilitate effective and efficient detection to attack packets in network traffic with low false rates and high detection rates.

Fourthly, we employ *F*-value measure as a metric to evaluate the performance of RePIDS. The definition of *F*-value is presented in Section 3.2.3. The reason of using *F*-value is that, the numbers of instances in the classes (normal and anomaly) are not equally distributed. Thus, the system is biased and attains an accuracy of more than 99%, if False Positive (FP), False Negative (FN), True Positive (TP) and True Negative (TN) measures are used in evaluating the performance of the system. However, *F*-value, based on Precision and Recall rates (detailed in Section 3.2.3), is independent of the sizes of the training and test samples.

Finally, we examine the effectiveness of our proposed system by conducting several experiments on DARPA 99 dataset [15] and Georgia Institute of Technology attack dataset (GATECH 2007) [16]. We compare the detection performance (*F*-value) and computational complexity of our proposed real-time payload-based IDS with two state-of-the-art payload-based IDSs, namely PAYL [17] and McPAD [18]. Experimental results show that the processing speed of RePIDS can accommodate the speed of a real scenario of medium size enterprise network. The rest of this paper is organized as follows. In Section 2, the most relevant research works are summarized. Section 3 discusses the framework of RePIDS. Experimental results and discussion are presented in Section 4. Section 5 demonstrates the evaluation results of RePIDS in terms of computational complexity, and compares RePIDS with the state-of-the-art PAYL and McPAD intrusion detection systems. Conclusions are drawn in Section 6.

2. Payload-based Network Intrusion Detection System

In this section, we provide a brief description of recent researches on payload-based intrusion detection systems. Due to the capability of detecting attacks carried out purely using packet payloads, there has been recently an increasing interest in payload-based approaches to build

detection models for Network Intrusion Detection Systems (NIDSs). Several typical effective payload-based NIDSs, namely PAYL, McPAD and GSAD, have been proposed in the literature [17–19].

The previous research works carried out in anomaly detection are based on simple statistics on application layer payload to build normal profile of web applications. A review of research works using n-gram analysis of network traffic payloads for feature construction is presented in the remaining of this section.

Wang and Stolfo proposed PAYL [17], which used 1-gram to build a byte-frequency distribution model of network traffic payloads. The pre-processing of packet payload using 1-byte sliding window creates a feature vector containing the relative frequency count of each of the 256 possible 1-grams (bytes) in the payload. Simplified Mahalanobis distance measure was used to compare new incoming traffic against the model. The overall detection rate was close to 60% with a false positive rate less than 1%. While PAYL method is effective at displaying abnormal byte distributions, it does have several shortcomings. For examples, it does not withstand mimicry attacks [18] and considers the entire payload for anomaly detection which presents a major problem in high-speed and high bandwidth network.

Bolzoni et al. proposed POSEIDON [20], a two-tier intrusion detection model. POSEIDON combined the Self Organizing Map (SOM) with the PAYL. In their paper, SOM was used for processing of packet payload and PAYL was used as a basis for detection. The aim of the SOM was to identify similar payloads for a given destination address and port. SOM improved detection accuracy. However, both SOM and PAYL need to be trained separately, so they could present difficulties with accuracy. For SOM, the number of neurons depends on the size of the network, and hence computational load increases quadratically with the number of neurons.

To model the structure of payload, Wang et al. proposed ANAGRAM [21]. A value of $n \geqslant 1$ was used to extract byte sequence information in a 256^n dimensional feature space. Supervised learning was employed to model normal traffic and attack traffic by storing n-grams of normal packets and attack packets into two separate Bloom Filters (BFs). However, according to Perdisci et al. [18], BFs would not work in high bandwidth or high data rate networks, because ANAGRAM stores n-grams within the BFs and generates a score based on the number of unobserved and malicious n-grams during detection phase. Unfortunately, it was more difficult to construct an accurate model due to the curse of dimensionality and possible computational problem. Perdisci et al. then proposed an IDS called McPAD [18], which created $2v$-grams by using a sliding window to cover all sets of 2 bytes that are v positions away in a network traffic payload. This generated a high dimensional feature space ($256^2 = 65{,}536$) because each byte could have values in the range from 0 to 255 and $n = 2$. The dimensionality of the feature space was then reduced using a clustering algorithm. However, they did not explain the criteria for the selection of number of clusters and one class classifiers. A model [22] proposed by Rieck and Laskov also extracted high-order n-grams language features from connection payloads. The authors compared high order n-grams and words in connection payloads using vectorial similarity measures such as kernel and distance functions.

However, all the afore-discussed methods consider payload features independently and do not consider correlations between the features and among the payloads. These result in high false alarm rates. To address this problem, we proposed GSAD model [19] to detect anomalies in the packet payloads. This model used n-gram text categorization technique for data pre-processing and MDM to determine the hidden correlations between payload features. Mahalanobis distance criterion was used to evaluate the similarity between the profile generated for a new incoming network traffic and the normal profile. This method was able to detect attacks with less than 1% false positive rate. However, this model has high computational complexity, and hence limits its use for offline operations only.

Recently, there is an increasing interest towards the development of high-speed monitoring technology. Network components use Deep Packet Inspection (DPI) [23,24] as an analyser to inspect attacks on all the layers. To identify and prevent malicious attacks, researchers are exploring both the header and the payload of each incoming packet. In [25], researchers developed Network Intrusion Detection and Prevention systems (NIDPs) using DPI technology. DPI searches the entire packet headers and payloads for pattern matching and uses a large rule database to compare against the incoming packets. Unfortunately, memory consumption is prohibitively high when traditional methods such as Deterministic Finite Automata (DFA) are used for fast regular expression scanning. Due to the complexity of rules and software-based implementation of DPI, the packet processing speed of a DPI-based solution is very limited. For instance, SNORT (a network intrusion detection system) has more than 4000 rules and can handle link rates only up to 250 Mbps [25] under normal traffic conditions. These rates are not sufficient to meet the needs of even medium-speed access or edge networks.

These existing software solutions of DPIs [25] are not sufficient enough to protect networks from new attacks and do not support high processing rates. Moreover, deep packet inspection compares a packet payload with thousands of known signatures and cannot detect attacks for which signatures are unavailable. Though both DPI and our proposed GSAD model are based on analysing network packet payloads, our scheme significantly differ from DPI as we consider the normal profiles only, whereas DPI uses attack signature database to classify a network packet as a normal or an attack.

Unfortunately, payload based IDSs commonly suffer from three major issues, namely higher false alarm rates (especially the higher false positive rates), complexity of high dimensional data and dynamic structuring of protocols. This is because payload-based IDS uses large number of features to discriminate normal packets and anomalous packets in the network traffic data. The presence of redundant and irrelevant features in the feature set results in high false positive rate and this restricts the operation of payload-based IDS to off-line processing of network traffic.

We propose RePIDS (an efficient payload-based anomaly intrusion detection system) in this paper. RePIDS uses PCA [26], an efficient method which helps reduce dimensionality by providing a linear mapping of n-dimensional feature space to a reduced m-dimensional feature space (dominant PCs), to boost the detection performance and may be suitable for real-time applications. In practice, reducing complex relationship between the features and discarding any irrelevant, redundant or less significant features from the original feature space are important for the real-time application of IDS. On one hand, this will reduce not only the volume of traffic but also processing time. On the other hand, this will improve the detection rate too.

Although PCA has been applied to the field of header-based intrusion detection [27–29] to achieve sensible feature reduction, to the best of our knowledge, no work has been done on the data pre-processing using PCA for payload feature selection. Nwanze et al. [30] discussed modeling of packet payload using data mining technique based on PCA. However, they ignored the main idea of PCA and did not use the projection of original data on a new lower dimensional feature space. Furthermore, they did not consider correlations between features.

The aims of this research conducted in this paper are to reduce the dimensionality of feature space and false positive rate. We achieve our research aims through efficient pre-processing of packet payload data and present data in a suitable format so that it can be used for the real-time intrusion detection.

3. Real-time Payload-based Network Intrusion Detection System

In this section, we elaborate on our new approach. Firstly, we present the framework of our real-time payload-based intrusion detection system. Then, we discuss the modules in the framework, namely data preparation module, n-gram text categorization module, 3-Tier Iterative Feature Selection Engine (IFSEng), profile generation and traffic classification.

3.1. Framework of the Proposed Real-time Payload-based Intrusion Detection System

The complete framework of our proposed intrusion detection system has four stages as shown in Fig. 1. They are data preparation, data pre-processing, model generation and anomaly detection.

The first stage of this IDS consists of data preparation and n-gram text categorization [31]. For data preparation, the incoming network traffic is filtered according to the types of applications and payload length, and n-gram text categorization converts the network traffic packet payloads into a series of feature vectors. These feature vectors describe the patterns of the incoming traffic in original high dimensional feature space.

In the second stage, a 3-Tier IFSEng, detailed in Section 3.2.3, is used for feature subset selection. Each Tier performs a specific task. At Tier 1, PCA technique [26] is used to analyse network traffic. At Tier 2, selection of dominant PCs (subsets of features) is performed using various methods as shown in [32,33]. Tier 3 refines optimal feature subset (PCs) and evaluates the discriminative power of the feature subsets to represent packet payloads. MDM shown in [19,34,35] (to be further discussed in Section 3.2.4) is used to capture more complex non-linear correlations among the selected features, and construct a distance map which represents a network traffic profile.

In the third stage of the framework, the output of IFSEng (finally selected PCs) is used to build a normal traffic profile. An MDM is created for normal network traffic as a normal profile, which is used for the classification of the new incoming network traffic in the last stage.

In the last stage, Mahalanobis Distance criterion is used to measure the dissimilarity between the pre-developed normal profile and the profile of a new incoming network packet. The packet is classified as a normal or an attack packet depending upon the amount of deviation of its profile from the normal profile. Detailed description of each module is given in the following subsections.

3.2. Framework modules

In this section, we provide a step-wise description and technical details of all modules contained in our proposed IDS framework.

3.2.1. Data preparation module

Data preparation is the first stage of the framework, where different datasets are prepared. We group network traffic into various categories using Wireshark [36], which is a traffic analyser and separates the network traffic based on types of services, destination address, payload length and direction of network traffic flow. The source of network traffic can be real network (for real-time operation) or collected tcpdump files. The prepared dataset is used by next stage of intrusion detection system.

3.2.2. n-Gram text categorization module

n-Gram text categorization is responsible for payload feature analysis and feature construction. It extracts raw features using n-gram text categorization technique (here $n = 1$) from the packet payload and converts observations into a series of feature vectors. Each payload is represented by a feature vector in a 256-dimensional feature space using

$$f_i = \frac{O_i}{\sum_{j=1}^{256} O_j}, \tag{1}$$

where O_i is the occurrence of ith n-gram. The overall value of the relative frequencies is given by

$$\sum_{i=1}^{256} f_i = 1. \tag{2}$$

Thus, a packet payload is then denoted by a relative frequency vector $q = [f_1\, f_2 \cdots f_{256}]^T$, which represents a pattern in the network payload in a 256-dimensional feature space. Here, T stands for "transpose" of a matrix.

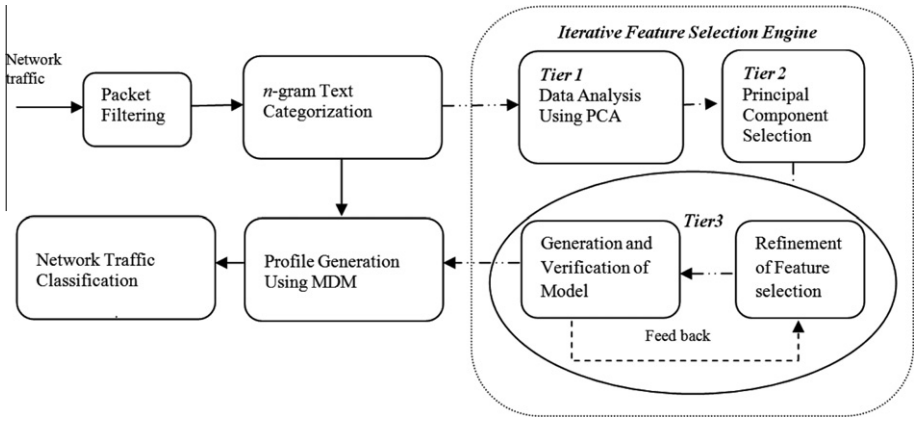


Fig. 1. Framework of real-time payload-based based intrusion detection system.

3.2.3. 3-Tier Iterative Feature Selection Engine

The 3-Tier Iterative Feature Selection Engine (IFSEng) consists of Tier 1: Data Analysis Using PCA, Tier 2: Principal Component Selection and Tier 3: Refinement of Feature Selection and Generation and Verification of Model.

At ***Tier 1***, PCA is used to analyse the original dataset. As a linear mathematical system, PCA is developed based on eigenvector-based multivariate analysis. It attempts to efficiently represent data by converting a set of observations into a new orthonormalized coordinate system, where the data are maximally decorrelated. The axes (eigenvectors or principal components) that contain greater variations (eigenvalues) make more contributions to the data representation. The first few most contributing axes are usually used to construct a new lower dimensional feature space which is believed to give efficient representations for the data.

PCA is applied on a network traffic dataset, $Q = [q_1\ q_2\ \cdots\ q_m]$, where m is the number of observations and each observation $q_i\ (1 \leqslant i \leqslant m)$ is denoted by a 256-dimensional feature vector $q_i = [f_1^i\ f_2^i \cdots f_{256}^i]^T$. First, zero-mean normalization is conducted on the dataset for all the observations to make PCA work properly. The zero-mean dataset is represented by

$$Q_{sh} = \begin{bmatrix} q_1 - \bar{q} \\ q_2 - \bar{q} \\ \vdots \\ q_m - \bar{q} \end{bmatrix}^T, \tag{3}$$

where $\bar{q} = \frac{1}{m}\sum_{i=1}^{m} q_i$. Then, the principal components are obtained by analysing the sample covariance matrix C_Q of the dataset given in (4).

$$C_Q = \frac{1}{m-1} Q_{sh} Q_{sh}^T. \tag{4}$$

Using eigen decomposition, the covariance matrix C_Q can be decomposed into a matrix W and a diagonal matrix λ. They satisfy the condition, $\lambda W = C_Q W$. The columns of the matrix W stand for the eigenvectors (called the principal components) of the covariance matrix C_Q, and the elements along the diagonal of the matrix λ are the ranked eigenvalues associated with the corresponding eigenvectors in the matrix W.

PCA only demonstrates the contribution of different components of a feature space in terms of data representation. It does not determine the number of principal components that should be retained. Thus, some other supplementary techniques are applied at Tier 2 to decide the optimal number of components to be retained based on the analysis results from PCA.

At ***Tier*** 2, several techniques, such as cumulative energy [32], scree test [33] and parallel analysis criteria, help achieve one of the main goals of good pre-processing principle that is to retain as much relevant information as possible. Cumulative energy, scree test and parallel analysis criteria are utilized independently and to select corresponding k_1, k_2 and k_3 principal components (the eigenvectors of matrix W) respectively. k_1, k_2 and k_3 are less or equal to k, where k equals to 256. These mathematical and non-mathematical criteria are used to verify the outcomes of each others. The subsets of principal components, corresponding to the selected k_1, k_2 and k_3, represent reduced feature spaces, which provide the best presentations determined by the criteria for a packet payload. By projecting the feature vector $q_i = [f_1^i\ f_2^i\ \cdots\ f_{256}^i]^T$ onto these selected reduced feature spaces, the dimension of the feature vector can be reduced significantly to smaller values, namely k_1, k_2 and k_3. In the meanwhile, the criteria guarantee that the reduced feature vector can correctly represent the packet payload. A brief explanation of individual criteria is given below.

Cumulative Energy. An energy associated with a component is represented by the corresponding eigenvalue. The greater an eigenvalue is, the larger energy the corresponding component (eigenvector) has. Suppose that (λ_1,u_1), $(\lambda_2, u_2), \cdots, (\lambda_k, u_k)$ are k eigenvalue–eigenvector pairs decomposed from the covariance matrix C_Q. The cumulative energy of the first k_1 components is defined by the sum of the energies across the components from 1 through k_1, and it is computed using (5)

$$CE = \sum_{j=1}^{k_1} \lambda_j, \tag{5}$$

where $k_1 \in \{1, \cdots, k\}$ can be determined subject to the objective function given in (6)

$$\frac{CE}{\sum_{j=1}^{k} \lambda_j} \geqslant \alpha, \tag{6}$$

in which α is the ratio of variation in the subspace to the total variation in the original space. This objective function intends to obtain a value of k_1 as small as possible while achieving a reasonably high value of *CE* on a percentage basis.

Scree Test is a graphical method, first proposed by Cattell [32] in 1966. A scree plot where all eigenvalues are plotted against all (k) principal components (eigenvectors) in the descending order. In the scree plot, we look for the k_2th point, where sharp decrease in eigenvalues levels off (the scree). This point is identified as an "elbow". After the k_2th point, the remaining $(k - k_2)$ principal components (eigenvectors) are ignored and not used in the model. This is based on the arguments that the most significant components extract a large proportion of the variances from the covariance matrix, while the remaining insignificant $(k - k_2)$ ones are associated with similar low value variances. The criticisms of scree test criterion are that there is no sharp transition where the scree begins, and the decision is not robust and reproducible. Alternatively, parallel analysis criterion is used to verify the selection of principal components (feature subset).

Parallel Analysis (PA) is a modification of Cattell's scree test. PA alleviates the component indeterminacy problem and determines which variable loadings are significant for each component. This operation is repeated twice and the obtained eigenvalues for each component are used to calculate means and Standard Deviations (SDs) in the two iterations. From the means and standard deviations, the 95th percentiles are obtained (the 95th percentile = mean + 1.65SD). If the eigenvalue of a component exceeds the 95th percentiles of the simulated values, then the component is retained.

At ***Tier 3***, feature refinement and evaluation module is used. In the refinement stage, we extend the range of the selected principal components, obtained from Tier 2, on both the upper and lower sides. Then, we observe the discriminative power of the subsets of principal components to represent packet payloads. Lastly, we select the final $k_{final} \in \{k_1, k_2, k_3\}$ principal components through iterative evaluation of normal training model using *F*-value defined in

$$F\text{-value} = (1 + \beta^2) * Recall * \frac{Precision}{\beta^2 (Recall + Precision)}, \tag{7}$$

where *Precision* defined in (8) shows how many events, predicted by an IDS as being intrusive, are the actual intrusions. A low value of precision means a higher degree of false positives and vice versa. *Recall* defined in (9) measures the missing part from the *Precision*, namely the percentage of the real intrusions covered by the classifier. A lower value of recall represents a higher degree of false negatives and vice versa.

$$Precision = \frac{TP}{TP + FP}, \tag{8}$$

$$Recall = \frac{TP}{TP + FN}. \tag{9}$$

In (8) and (9), *TP* (True Positive) indicates the number of attacks correctly detected by the intrusion detection system as attacks, *TN* (True Negative) indicates the number of normal packets correctly classified by the intrusion detection system as normal without making any mistake, *FP* (False Positive) indicates the number of normal packets incorrectly classified by intrusion detection system as attacks, and *FN* (False Negative) indicates the number of attacks incorrectly classified by intrusion detection system as normal packets.

In (7), β corresponds to the relative importance of precision versus recall and is usually set to 1. On one hand, when precision and recall have equal weights and are close to 1, the model can achieve *F*-value close to 1, which indicates good performance meaning that the classifier has 0% false alarms and 100% detection of attacks. On the other hand, *F*-value close to 0 indicates poor performance. Thus, the *F*-value of a classifier is desired to be as high as possible.

The selected k_{final} principal components are the one which facilitates the classifier to achieve the greatest *F*-value among the candidates k_1, k_2 and k_3. Then, selected k_{final} principal components are used in the profile generation, which is briefly discussed in Section 3.2.4.

3.2.4. Profile generation using Mahalanobis Distance Map

Network traffic profile is generated using Mahalanobis Distance Map (MDM) which captures complex non-linear correlations of the data. By using MDM, the hidden correlations between the features of projected feature vector $[x_1\ x_2\ \cdots\ x_{k_{final}}]$, which is obtained from the projection of original feature vector $q = [f_1\ f_2 \cdots f_{256}]^T$ onto the k_{final} dimensional feature subspace $[u_1\ u_2\ \cdots\ u_{k_{final}}]$ (outcome of IFSEng), and the correlations among packets are obtained as follows.

$$\Sigma_a = (x_a - \mu)(x_a - \mu)^T (1 \leqslant a \leqslant k_{final}), \tag{10}$$

$$d_{(a,b)} = \frac{(x_a - x_b)(x_a - x_b)^T}{\Sigma_a + \Sigma_b} (1 \leqslant a, b \leqslant k_{final}), \tag{11}$$

$$D = \begin{bmatrix} d_{(1,1)} & d_{(1,2)} & \cdots & d_{(1,k_{final})} \\ d_{(2,1)} & d_{(2,2)} & \cdots & d_{(2,k_{final})} \\ \vdots & \vdots & \ddots & \vdots \\ d_{(k_{final},1)} & d_{(k_{final},2)} & \cdots & d_{(k_{final},k_{final})} \end{bmatrix}, \tag{12}$$

where x_a represents the ath projected feature in the projected feature vector, μ denotes the average of each projected feature, $d_{(a,b)}$ defines the Mahalanobis distance between the ath projected feature and the bth projected feature, Σ_a is the covariance value of each projected feature, and finally D is the MDM (the pattern of a network packet). Distance map D is used to generate the network traffic profiles (normal and attack) of the training and test data. These profiles are used for the classification of incoming network traffic.

3.2.5. Traffic classification

Mahalanobis distance is the criterion used to measure the dissimilarity between the developed profiles and new incoming network traffic profiles. Weight score w is calculated using (13) to detect an intrusive activity.

$$w = \sum_{a,b=1}^{k_{final}} \frac{(d_{obj(a,b)} - \bar{d}_{nor(a,b)})^2}{\sigma^2_{nor(a,b)}}, \tag{13}$$

where $\bar{d}_{nor(a,b)}$ and $\sigma^2_{nor(a,b)}$ are the average and the variance of the (a, b)th element in the distance map of the normal profile given in (14), and $d_{obj(a,\ b)}$ is the (a, b)th element of the distance map of the new incoming packet shown in (15).

$$D_{nor} = [d_{nor(a,b)}]_{k_{final} \times k_{final}} \tag{14}$$

$$D_{obj} = [d_{obj(a,b)}]_{k_{final} \times k_{final}} \tag{15}$$

If the weight score exceeds the threshold, the input packet is considered as an intrusion.

4. Experimental results and analysis

In the following subsections, first, we present brief information on the dataset and types of attacks. Then, we discuss training and test of our model. Finally, we present experimental results and analysis.

A Series of experiments on DARPA 99 [15] dataset and Georgia Institute of Technology attack dataset (GATECH) [16,19] are conducted to evaluate the performance of our proposed model. Although the DARPA 99 dataset was criticized by McHugh [37] for many of its weaknesses, including the questionable collection of traffic data, attack taxonomy and distribution, and evaluation criteria, DARPA 99 dataset is the only publicly available, large and well labeled dataset, and is still the most widely used public benchmark for testing intrusion detection systems. The GATECH attack dataset is also publicly available and contains traces of real attack traffic. These two datasets are used by the state-of-the-art payload-based IDSs that we will compare in this paper.

4.1. Dataset

4.1.1. Training (normal traffic) dataset

We extract Week 1 and Week 3 inbound "HTTP request" traffic from DARPA 99 dataset for the training of our model. The extracted normal traffic corresponds to two different HTTP servers. The total numbers of packets used for training of the model after filtering are 13,933 and 10,464 for hosts marx and hume respectively.

4.1.2. Test (attack + normal traffic) dataset

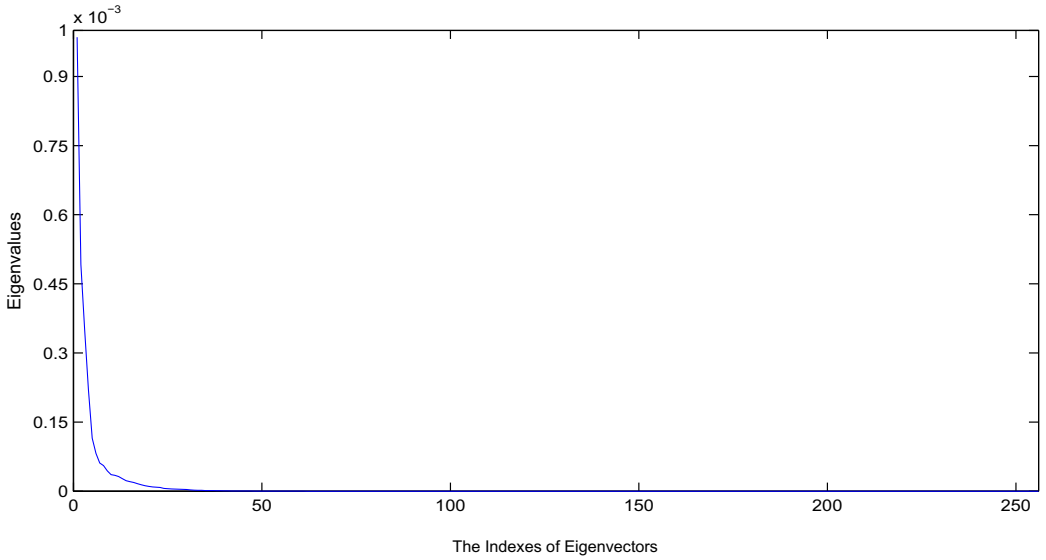
In order to test the performance of our proposed model in detecting known attacks and new attacks, we use attacks contained in DARPA 99 dataset and GATECH attack dataset. The labeled test data is further preprocessed to form two test datasets, which contains instances that do not appear in our training dataset. For our experiments, we focus on attacks coming through HTTP service only.

HTTP-based attacks are mainly from the HTTP GET/POST requests to web servers. There are several HTTP-based attack provided by DARPA 99 dataset, namely Apache2 attack, CrashIIS attack and Phf attack. The GATECH attack dataset has several non-polymorphic HTTP attacks provided by Ingham and Inoue [16] and several polymorphic HTTP attacks created using CLET engine generated by Perdisci et al. [18]. The attacks, namely Generic attack, Shell-code attack and CLET attack (polymorphic attack), are placed in different groups, and each group has attacks of the same category for the presentation of results. All HTTP request attack packets are used in our experiments, and the detailed explanation for the types of attacks can be found in Appendix A.

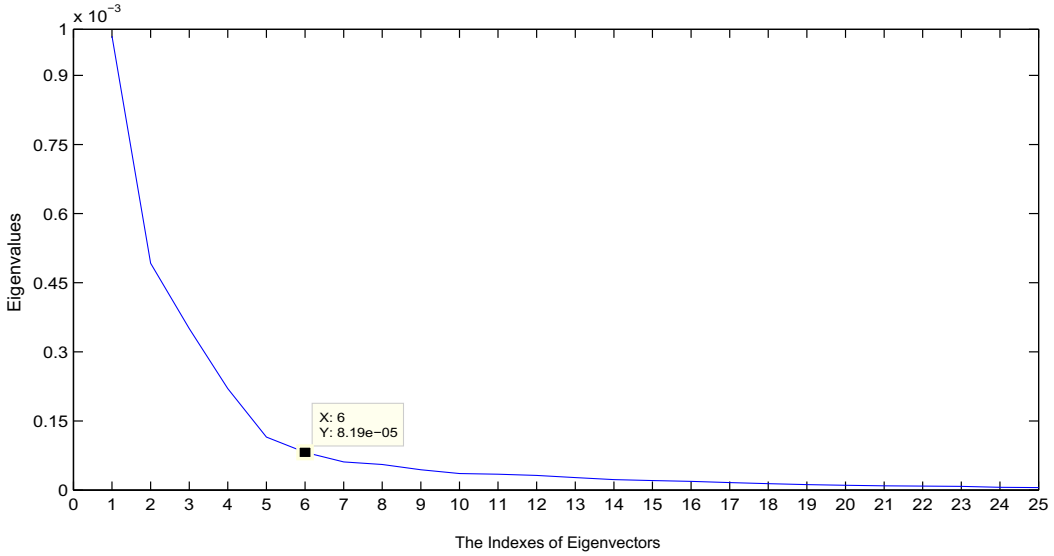
4.2. Model training and testing process

The experimental approach involves following procedure for training and testing of model:

1. As discussed in Section 3.2.2, we parse 185 bytes of packet payload of a HTTP GET request using a sliding window of length 1-byte and represent it by a feature vector q in a 256-dimensional feature space.
2. As discussed in Section 3.2.3, Tier 1 uses the PCA technique to analyse raw data, namely ASCII character occurrence frequencies, in the training dataset, by projecting raw data on a reduced feature space. At Tier 2 of IFSEng, selection of dominant Principal Components (PCs) is done by means of cumulative energy, scree test and parallel analysis criteria on the outcome of PCA.
   - First, cumulative energy criterion is applied for the selection, in which we consider 93% of cumulative energy level for (6). A k_1 equal to 7 is obtained, which means that the first seven principal components are selected as the best subspace to represent the data by cumulative energy criterion.
   - Then, we use scree test to draw scree plot as a variance captured by a given principal component and to select another set of principal components. Fig. 2a shows full scree plot, where we use k ($k = 256$ in our case) principal components (X-axis) of a particular dataset and the corresponding variances, namely eigenvalues (Y-axis), to draw a scree plot. The principal components are sorted in descending order with respect to the values of the corresponding variances. In Fig. 2a, we look for an "elbow", a flattening out of the curve. To provide better vision, we magnify the scree plot and show the first 25 principal components in Fig. 2b. It can be observed from Fig. 2b that there is a sharp decrease of variance in the front part of the plot and then it starts flattening out after the 6th principal component. In Fig. 2b, we can observe the "elbow" somewhere in the range from 6 to 9 principal components and the first $k_2 = 6$ principal components are able to capture about 92% of the variance. After the k_2th point, the remaining $(k - k_2)$ principal components capture only around 8% of the total variance and are ignored.



(a) Full scree plot



(b) Enlarged scree plot with 25 eigenvectors

Fig. 2. Scree test plot.

- We use $k_2 = 6$ as dominant principal components in our case. However, from Fig. 2b, we have observed a range of principal components from 6 to 9, and it is not very clear that what is the most appropriate value of k_2. To overcome this ambiguity, we use parallel analysis criterion as described in the following and to verify the selection of k_2.
- We verify the outcome of scree plot by using parallel analysis criterion as discussed in Section 3.2.3 on the same dataset. The result of parallel analysis also suggests a selection of first seven principal components, which are same as what have been obtained using cumulative energy criterion. The results of three feature selection criteria are given in Table 1.

3. Although these are the dominant principal components, further refinement of dominant principal components needs to be done at Tier 3 of IFSEng (as presented in Section 3.2.3) because of the ambiguity in these results. In addition, generation and evaluation of training model at Tier 3 are performed using *F*-value metric defined in (7). The MDM represents the correlations between the features obtained from the projection of the original feature vector onto the finally selected principal

Table 1
Principal Component (PC) celection.

PC selection method	Cumulative energy (0.93)	Scree test	Parallel analysis
Number of PCs	7	6	7

components and between packets. These principal components help represent normal behavior profile in the low dimensional feature space.

4. For testing, we project the extracted feature vector of an incoming packet payload on the reduced feature space (the finally selected principal components) and use Mahalanobis distance dissimilarity criterion to detect intrusive behaviors. The performance of RePIDS in detecting attacks is evaluated using *F*-value.

In the experimentation, the 10 days normal "HTTP GET request" traffic from DARPA 99 dataset is used. The normal traffic is randomly divided into three subsets. One of the subset is selected randomly and used for training the model. The remaining two subsets are reserved for the test of the model.

In the testing stage, an attack is detected as long as one of its attack packets is identified as abnormal. We conduct our experiments using the features obtained from the projection of original feature vectors onto the optimal principal components determined by the IFSEng for various types of attacks (Apache2, Phf, CrashIIS and Back attacks) present in DARPA 99 attack dataset. We further evaluate our model on GATECH attack dataset, which is comprised of generic, polymorphic (CLET) and shell-code attacks. The experiments are conducted on a computer with two 3.33 Ghz 8 MB cache Quad Core Xeon CUPs and 48 GB DDR3-1333 ECC memory. This is a shared computational environment for heavy mathematical calculation and modeling experimentation. The performance is heavily influenced by the number of processes running simultaneously. Matlab is used for the simulation.

4.3. Results and analysis

Experimental results are explained in two steps. In the first step of the experiments, we obtain the optimal subset of principal components. Then, we design a number of experiments based on Fig. 1 to determine the performance of RePIDS when using various subsets of principal components varying from five components to nine components. Experiments are also conducted for different values of threshold varying from 2σ to 3.5σ. Results are presented in Table 2 for various feature subsets and 3.5σ as the optimal value of threshold.

Table 2 shows the variation of FP, TN, TP and FN rates along the change of the number of principal components. To obtain the optimal number of principal components, *F*-value is calculated for each feature subspace (principal components) using (7). Fig. 3 shows variation of *F*-value with the number of principal components. The results show that the best *F*-value is achieved with seven principal components. In other words, the feature subspace of seven principal components has good representation, discriminative power and high accuracy. The increase and decrease of the eigenvectors both dilute the performance of RePIDS.

It can be concluded that PCA and the three selection criteria help reduce the dimensionality of dataset from 256 to 7. The amount of information extracted using IFSEng is high in the selected 7-dimensional feature space, which helps create more accurate normal traffic profiles using MDM that is used for traffic classification.

To demonstrate how MDM presents the correlations between the features, the MDMs of normal HTTP payload and some attack payloads generated using projected features, the optimal 7-dimensional space, are given in Figs. 4 and 5 respectively. It can be seen from Figs. 4 and 5 that a MDM is a symmetric matrix and the values of the elements along its diagonal are all equal to zeros. This is because the correlation of a feature to itself is always zero. MDMs also demonstrate that the correlations between normal projected features are different from the correlations between attack projected features. Besides, the 7-dimensional space is able to help differentiate normal payload and various attack payloads efficiently and accurately. Fig. 4 shows the MDM of normal HTTP payload (normal profile), and Fig. 5a–c shows the MDMs of the attack profiles for Apache2, CrashIIS and Phf attacks.

Although we can directly compare the normal profile (model) and attack profiles (MDMs) to confirm the differences between normal and various attack payloads, it is a time-consuming task. Having MDM profiles for training dataset and a new incoming packet, the weight score *w* is calculated. If the deviation in weight score *w* is greater than the pre-selected threshold, then the incoming packet is classified as an attack packet.

Moreover, to evaluate the robustness of RePIDS in recognizing unknown attacks (Generic, Shell-code and Polymorphic (CLET) attacks), we conduct experiments on GATECH attack dataset using the same setup. Table 3

Table 2
Performance scores corresponding to the number of Principal Components (PCs).

	5 PCs (%)	6 PCs (%)	7 PCs (%)	8 PCs (%)	9 PCs (%)
False Positive (FP) rate	1.37	0.67	0.85	1.31	1.99
True Negative (TN) rate	98.63	99.33	99.15	98.69	98.01
True Positive (TP) rate	98.70	99.50	100	100	99.97
False Negative (FN) rate	1.30	0.50	0	0	0.03

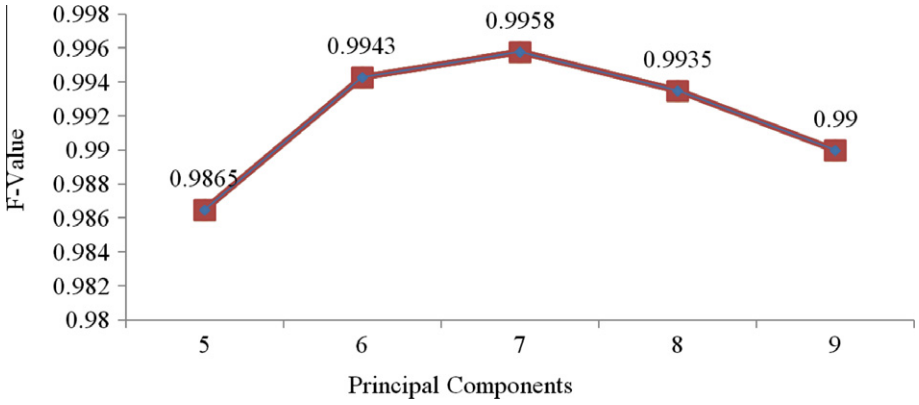


Fig. 3. Trend of *F*-value.

0	0.001406625	0.001449804	0.001332988	0.001463112	0.001270879	0.001241186
0.001406625	0	0.000289528	0.000305565	0.000268982	0.000231624	0.000208517
0.001449804	0.000289528	0	0.000239652	0.000214287	0.000194018	0.000163789
0.001332988	0.000305565	0.000239652	0	0.000287999	0.000198282	0.000158613
0.001463112	0.000268982	0.000214287	0.000287999	0	0.00016282	0.000170964
0.001270879	0.000231624	0.000194018	0.000198282	0.00016282	0	9.17989 exp-05
0.001241186	0.000208517	0.000163789	0.000158613	0.000170964	9.17989 exp-05	0

Fig. 4. MDM of normal HTTP payload.

0	7.211042exp-05	3.686978 exp-06	0.00237153	0.000102354	0.00078712	0.00072289
7.211042 exp-05	0	5.214637 exp-05	0.00163562	0.00033709	0.00132127	0.00034557
3.686978 exp-06	5.214637 exp-05	0	0.00225582	0.00012659	0.00085651	0.00065863
0.00237153	0.00163562	0.00225582	0	0.00344129	0.00586325	0.00048361
0.00010235	0.00033709	0.00012659	0.00344129	0	0.00032706	0.00135888
0.00078711	0.00132127	0.00085651	0.00586325	0.00032706	0	0.00300482
0.00072289	0.00034557	0.00065864	0.00048362	0.00135888	0.00300482	0

(a) Apache2 Attack Payload

0	0.000677245	0.00081015	0.00032632	0.00019996	0.00095956	0.00014843
0.000677245	0	0.00022798	0.00036073	0.00038006	0.00045121	0.00033855
0.00081015	0.00022798	0	0.00029764	0.0003492	0.00030296	0.00032801
0.00032632	0.00036073	0.00029764	0	8.31436139	0.00032254	0.00011205
0.00019996	0.00038006	0.0003492	8.31436139	0	0.00033113	8.634902 exp-05
0.00095956	0.000451205	0.00030296	0.00032254	0.00033113	0	0.00055453
0.000148432	0.00033855	0.00032801	0.00011205	8.634902 exp-05	0.00055453	0

(b) CrashIIS Attack Payload

0	0.05178815	0.04735877	0.04525517	0.03765384	0.03965582	0.05104155
0.051788147	0	0.03508168	0.05975747	0.05529712	0.05478485	0.03144298
0.047358766	0.03508168	0	0.03686035	0.0250256	0.0571498	0.0332321
0.045255171	0.05975747	0.03686035	0	0.05269052	0.05324839	0.05400761
0.037653843	0.05529712	0.0250256	0.05269052	0	0.03450803	0.04522816
0.039655825	0.05478485	0.0571498	0.05324839	0.03450803	0	0.04336399
0.051041546	0.03144298	0.0332321	0.05400761	0.04522816	0.04336399	0

(c) Phf Attack Payload

Fig. 5. MDMs of attack HTTP payloads.

Table 3
Performance scores.

Performance score	7 Eigenvectors (%)
False positive (FP) rate	0.85
True negative (TN) rate	99.15
True positive (TP) rate	96.29
False negative (FN) rate	3.71
F-value	0.976

reports the FP rate, TN rate, TP rate, FN rate and *F*-value on the optimal 7-dimensional space. It can be concluded from Table 3 that RePIDS has a high detection rate, a low false positive rate and a low false negative rate. The *F*-value achieved is 0.976, which confirms that the model can detect attacks with high accuracy and demonstrates its good performance.

In conclusion, the proposed RePIDS is able to detect novel attacks very well with a high *F*-value (0.976) and a low FP rate.

5. Comparison of RePIDS

In this section, comparisons between RePIDS and the state-of-the-art PAYL and McPAD anomaly based intrusion detection systems are presented. Then, we further compare throughput of our proposed model with that of a real scenario of a medium sized enterprise network.

5.1. Detection performance

In order to provide a reasonable comparison for these payload-based IDSs, the detection performance of RePIDS, PAYL and McPAD anomaly based intrusion detection systems is first compared. Thus, we use the results of false positive rate and detection rate from [18]. From Figs. 6 and 7 in [18], we estimate average detection rates for generic, shell-code and polymorphic attacks. We use false positive rate of 1% to calculate *F*-values for PAYL and McPAD on GATECH attack dataset respectively. As mentioned in [18], their results for DARPA 99 dataset are similar to those for GATECH attack dataset. Table 4 shows comparison of *F*-values for PAYL, McPAD and RePIDS on DARPA 99 dataset and GATECH attack dataset. From Table 4, we can conclude that RePIDS shows better *F*-value in comparison with PAYL and McPAD on DARPA 99 and GATECH attack datasets.

5.2. Complexity analysis

In this section, we provide an analysis of the computational complexities of the algorithms used in RePIDS, PAYL and McPAD. Only the computation involved in the test phase is taken into account in the analysis, because the training of the algorithms can be performed off-line, which does not affect efficiency of the algorithms in detection.

Given a payload P of length n and a fixed value of v, the occurrence frequencies of 1-gram and $2v$-grams can both be computed in $O(n)$. The numbers of extracted features in these algorithms are constant regardless of the actual values of n and v (2^8 features extracted by RePIDS and PAYL, and 2^{16} features extracted by McPAD).

The feature reduction process of the RePIDS can be completed by $2^8 * 2 * 7 = 3584$ simple operations including multiplications and additions. In contrast, McPAD algorithm reduces features by mapping the occurrence frequency distribution of $2v$-grams to the k feature clusters using a simple look-up table and a number of sum operations that is always less than 2^{16} (regardless of the value of k). Therefore, the feature reduction processes of RePIDS and McPAD can be computed in $O(1)$. However, no feature reduction is performed in PAYL.

Thus, the complete computational complexities of data pre-processing of the RePIDS, PAYL and McPAD algorithms can be obtained by adding up the computational complexities of the feature extraction and reduction processes. Since RePIDS uses a fixed payload length (185 bytes) to extract the occurrence frequency, the complete computational complexity of data pre-processing is $O(1)$. PAYL has a complete computational complexity of data pre-processing equal to $O(n)$, because no feature reduction is required. For McPAD, it has to be repeated m (representing the number of different one class classifiers used to make a decision about each payload P) times and to choose a different value of v every time. Hence, the complete computational complexity of data pre-processing of McPAD can be accomplished in $O(nm)$.

Once the features have been extracted and the dimensionality has been reduced to k, each payload has to be classified according to each of the m classifiers. To classify a payload P, RePIDS computes the Mahalanobis distance between the payload and the pre-determined normal profile. Given the number of features equal to 7 as determined and a single classifier used in classification, the computational complexity of the classification process of RePIDS is $O(1)$. Similarly, PAYL uses a single classifier to classify the payload P represented by 256 features. Therefore, the classification process of PAYL can be accomplished in $O(1)$ as well. Compared to RePIDS and PAYL, McPAD has m classifiers. Each classifier computes the distance between the payload P represented by k feature clusters and each of the support vector s obtained during training. Therefore, the classification of a payload using McPAD can be computed in $O(ks)$. McPAD has to repeat the classification process m times and the results are then combined. Thus, the overall classification process of McPAD can be computed in $O(mks)$. The detailed break-down of the computational complexities of the algorithms is given in Table 5.

As shown in Table 5, the overall computational complexities of RePIDS, PAYL and McPAD are $O(1)$, $O(n)$ and $O(nm + mks)$ respectively. This proves that our RePIDS has

Table 4
Performance comparison.

	RePIDS	PAYL[a]	McPAD[a]
DARPA 99	0.9958	0.969[a]	0.953[a]
GATECH	0.976	0.969	0.953

[a] *F*-values for DARPA 99 dataset and GATECH attack dataset for PAYL and McPAD have been derived from [18].

Table 5
Computational complexities of RePIDS, PAYL and McPAD.

	RePIDS	PAYL	McPAD
Complexity of data pre-processing	$O(1)$	$O(n)$	$O(nm)$
Complexity of classification	$O(1)$	$O(1)$	$O(mks)$
Overall complexity	$O(1)$	$O(n)$	$O(nm + mks)$

the lowest computational complexity in comparison with PAYL and McPAD.

We also evaluate the efficiency of our scheme by comparing the throughput of RePIDS with a similar environment used within a medium size enterprise network with a gateway speed of 1 GB. Our throughput comparison is based on the number of packets processed through such a network against the packet processing speed of our scheme considering the most ideal parameters. On one hand, the throughput calculated for a medium sized enterprise network, considering ideal parameters is 25,600 packets in one second. However, we expect in actual (real-time), throughput will be much less than what we have used for comparison. On the other hand, our proposed scheme could process 33,146 packets per second, which is 1.3 times more than the packet processing speed on the enterprise network, indicating our scheme has potential to be implemented in real-time. However such consideration involving real throughput analysis with most ideal network parameters is beyond the scope of this paper and we intend to extend it for our future work.

Summarizing the overall performance in terms of detection accuracy and computational complexities of algorithms, RePIDS performs better than the state-of-the-art PAYL and McPAD anomaly based intrusion detection systems. Furthermore, in terms of throughput, RePIDS can process more packets per second than the throughput of a medium sized enterprise network with a gateway speed of 1 GB. Hence, our model, RePIDS, is expected to be capable of processing packets in real time operation.

6. Conclusions

In this paper, we have proposed an efficient payload-based intrusion detection system (RePIDS) to detect attacks against Web applications through the analysis of HTTP payloads using 3-Tier Iterative Feature Selection Engine (IFSEng) and Mahalanobis Distance Map (MDM). Mahalanobis distance criterion is used for classification of network data. The proposed model uses selected, small-sized feature subspace to detect generic, shell-code and CLET attacks. Also, RePIDS is capable of discriminating normal patterns and attack patterns in real-time.

The proposed 3-Tier IFSEng is used to select an optimal feature subspace and to reduce the dimensionality of the data. This is because data being used in payload-based intrusion detection inherit a problem of high dimensions in nature, which significantly influence the detection efficiency.

In addition, the use of MDM offers benefits in exploring hidden correlations between features and also between packet payloads. Furthermore, MDM is able to capture structural information of the payload partially, which improves the performance of our proposed model.

Experimental results indicate that the method is effective in detecting attacks with high detection rates and low false positive rates. Using the proposed method, the number of features required to generate network profile are very few. This shows low computational complexity and low training and testing processing time of our proposed scheme. We have also shown that our approach has good potential to be used in real-time too. RePIDS has been thoroughly tested on the normal traffic of DARPA 99 dataset, and on two different datasets of attacks, namely DARPA 99 and GATECH datasets. GATECH dataset contains the real traces of attacks collected from various sites. RePIDS has achieved high *F*-value, 0.9958 on DARPA 99 dataset and 0.976 on GATECH dataset respectively. This demonstrates that RePIDs is capable of differentiating normal and attack instances accurately. In particular, we have demonstrated that RePIDS performs better in comparison with the state-of-the-art PAYL and McPAD. In addition, we have also shown that the computational complexity of RePIDS for the classification of a new incoming traffic payload is lower than PAYL and much less than McPAD. The reason of improvement in computational complexity of RePIDS algorithms is because of reduced feature space. Thus, our model decreases the average computation cost per payload while maintaining a high detection rate at low false positive rate.

Finally, in terms of throughput, RePIDS can process more packets per second than the throughput of a medium sized enterprise network with a gateway speed of 1 GB. Hence, our model, RePIDS, is expected to be capable of processing packets in real time operation.

Although RePIDS has shown good performance in protection of network against intrusion, it is used for intrusion detection of unencrypted (plain text) payload data only. It does not look into encrypted data. However, it can detect attacks coming through encrypted data when used at the host machine using an appropriate encryption key.

Acknowledgement

This work was supported by the Australian Postgraduate Awards (APA), the University of Technology Sydney (UTS) International Research Scholarship (IRS) and the Commonwealth Scientific and Industrial Research Organisation (CSIRO) Information and Communication Technologies (ICT) Centre Top-up Scholarships.

Appendix A

- *Generic Attacks*. This dataset includes all the HTTP attacks plus a shell-code attack that exploits vulnerability (MS03-022) in Windows Media Service (WMS). Generic attacks are applicable to any group. For example, attacks cause Information Leakage and Denial of Service (DoS).
- *Shell-code Attacks*. This dataset contains 11 shell-code attacks from the generic attacks dataset. Shell-code attacks are dangerous because they inject executable

code and hijack the normal execution of the target application. For example, Code-Red worm uses shellcode attacks to propagate.

- *CLET Attacks*. This dataset contains 96 polymorphic attacks generated using the polymorphic engine CLET [8]. Polymorphic attacks are polymorphic version of known attacks. Examples are: Code-Red (a famous worm that exploits vulnerability in Windows IIS (MS01-044)), DDK (an exploit to a buffer overflow vulnerability in Windows IIS (MS01-033)), etc.
- *Apache2 Attack*. Denial of service attack against an apache web server where a client sends a request with many MIME headers. These requests will cause the server to slow down, and may eventually crash it.
- *Back Attack*. Denial of service attack against apache web server where a client requests a URL containing many backslashes. As the server tries to process these requests it will slow down and be unable to process other requests.
- *Phf Attack*. Any CGI program which relies on the CGI function escape shell cmd () to prevent exploitation of shell-based library calls may be vulnerable to attack. In particular, this includes the phf program which is distributed with the example code. The phf program allows remote users to run arbitrary commands on the server.
- *Crashiis Attack*. Denial of Service attack against the NT IIS web server. The attacker sends a malformed GET request via telnet to port 80 on the NT victim. The command "GET ../.." crashes the web server (and sometimes crashes the ftp and gopher daemons as well).

References

[1] D.E. Denning, An intrusion-detection model, IEEE Transactions on Software Engineering SE-13 (1987) 222–232.

[2] L. Ertz, E. Eilertson, A. Lazarevic, P.N. Tan, V. Kumar, J. Srivastava, P. Dokas, The MINDS – Minnesota Intrusion Detection System, in: Next Generation Data Mining, MIT Press, Boston, 2004, pp. 1–21.

[3] J.M. Estevez-Tapiador, P. Garcia-Teodoro, J.E. Diaz-Verdejo, Stochastic protocol modeling for anomaly based network intrusion detection, in: Proceedings. First IEEE International Workshop on Information Assurance, 2003, pp. 3–12.

[4] A. Patcha, J.-M. Park, An overview of anomaly detection techniques: existing solutions and latest technological trends, Computer Networks 51 (2007) 3448–3470.

[5] A. Lazarevic, V. Kumar, J. Srivastava, Intrusion detection: a survey, in: V. Kumar, J. Srivastava, A. Lazarevic (Eds.), Managing Cyber Threats, Springer, US, 2005, pp. 19–78.

[6] J. Early, C. Brodley, Behavioral features for network anomaly detection, in: M. Maloof (Ed.), Machine Learning and Data Mining for Computer Security, Springer, London, 2006, pp. 107–124.

[7] S. Axelsson, Intrusion detection systems: a survey and taxonomy, in: Technical Report, 2000, pp. 1–27.

[8] S. Kotsiantis, D. Kanellopoulos, P. Pintelas, Data preprocessing for supervised leaning, International Journal of Computer Science 1 (2006) 111–117.

[9] M. Mahoney, P.K. Chan, PHAD: packet header anomaly detection for identifying hostile network traffic, in: Florida Institute of Technology Technical Report 2001, pp. 1–17.

[10] P. Garca-Teodoro, J. Daz-Verdejo, G. Macia-Fernandez, E. Vazquez, Anomaly-based network intrusion detection: techniques, systems and challenges, Computers & Security 28 (2009) 18–28.

[11] A. Lakhina, M. Crovella, C. Diot, Mining anomalies using traffic feature distributions, in: Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, ACM, Philadelphia, Pennsylvania USA, 2005, pp. 217–228.

[12] W. Lee, S.J. Stolfo, A framework for constructing features and models for intrusion detection systems, ACM Transactions on Information and System Security 3 (2000) 227–261.

[13] M. Damashek, Gauging similarity with N-grams: language-independent categorization of text, Science 267 (1995) 843–848.

[14] J.J. Davis, A.J. Clark, Data preprocessing for anomaly based network intrusion detection: a review, Computers & Security 30 (2011) 353–375.

[15] R. Lippmann, J.W. Haines, D.J. Fried, J. Korba, K. Das, The 1999 DARPA off-line intrusion detection evaluation, Computer Networks 34 (2000) 579–595.

[16] K. Ingham, H. Inoue, Comparing anomaly detection techniques for HTTP, in: C. Kruegel, R. Lippmann, A. Clark (Eds.), Recent Advances in Intrusion Detection, Springer, Berlin, Heidelberg, 2007, pp. 42–62.

[17] K. Wang, S. Stolfo, Anomalous payload-based network intrusion detection, in: E. Jonsson, A. Valdes, M. Almgren (Eds.), Recent Advances in Intrusion Detection, Springer, Berlin, Heidelberg, 2004, pp. 203–222.

[18] R. Perdisci, D. Ariu, P. Fogla, G. Giacinto, W. Lee, McPAD: a multiple classifier system for accurate payload-based anomaly detection, Computer Networks 53 (2009) 864–881.

[19] A. Jamdagni, Z. Tan, P. Nanda, X. He, R. Liu, Intrusion detection using geometrical structure, in: Fourth International Conference on Frontier of Computer Science and Technology, 2009, pp. 327–333.

[20] D. Bolzoni, S. Etalle, P. Hartel, POSEIDON: a 2-tier anomaly-based network intrusion detection system, in: Fourth IEEE International Workshop on Information Assurance 2006, pp. 156–165.

[21] K. Wang, J.J. Parekh, S.J. Stolfo, Anagram: a content anomaly detector resistant to mimicry attack, in: Proceedings of the 9th International Conference on Recent Advances in Intrusion Detection, Springer-Verlag, Hamburg, Germany, 2006, pp. 226–248.

[22] K. Rieck, P. Laskov, Language models for detection of unknown attacks in network traffic, Journal in Computer Virology 2 (2007) 243–256.

[23] Y.-M. Chu, Deep packet inspection in network intrusion detection and prevention systems, in: Institute of Communications Engineering, National Tsing Hua University, 2010.

[24] T. Porter, The Perils of Deep Packet Inspection, in: Security Focus, 2005.

[25] F. Yu, High speed deep packet inspection with hardware support, in: EECS Department, University of California, Berkeley, 2006.

[26] I. Jolliffe, Principal Component Analysis, Wiley Online Library, 2005.

[27] Y. Bouzida, F. Cuppens, N. Cuppens-Boulahia, S. Gombault, Efficient intrusion detection using principal component analysis, in: 3me Confrence sur la Scurit et Architectures Rseaux (SAR), La Londe, France, 2004.

[28] Y. Bouzida, S. Gombault, Eigenconnections to Intrusion Detection, in: Y. Deswarte, F. Cuppens, S. Jajodia, L. Wang (Eds.), Security and Protection in Information Processing Systems, Springer, US, 2004, pp. 241–258.

[29] W. Wang, X. Guan, X. Zhang, Processing of massive audit data streams for real-time anomaly intrusion detection, Computer Communications 31 (2008) 58–72.

[30] N. Nwanze, K. Sun-il, D.H. Summerville, Payload modeling for network intrusion detection systems, in: Military Communications Conference, 2009, MILCOM 2009., IEEE, 2009, pp. 1–7.

[31] Y. Liao, V.R. Vemuri, Using text categorization techniques for intrusion detection, in: Proceedings of the 11th USENIX Security Symposium, USENIX Association, 2002, pp. 51–59.

[32] L.R. Nelson, Some observations on the scree test, and on coefficient alpha, Thai Journal of Educational Research and Measurement 3 (1) (2005) 1–17.

[33] R.B. Cattell, The scree test for the number of factors, Multivariate Behavioral Research 1 (1966) 245–276.

[34] A. Jamdagni, Z. Tan, P. Nanda, X. He, R.P. Liu, Intrusion detection using GSAD model for HTTP traffic on web services, in: Proceedings of the 6th International Wireless Communications and Mobile Computing Conference, ACM, Caen, France, 2010, pp. 1193–1197.

[35] Z. Tan, A. Jamdagni, X. He, P. Nanda, Network Intrusion Detection based on LDA for Payload Feature Selection, in: GLOBECOM Workshops (GC Wkshps), 2010, pp. 1545–1549.

[36] L. Chapell, Wireshark Network Analysis: The Official Wireshark Certified Network Analyst Study Guide, Protocol Analysis Institute, 2010.

[37] J. McHugh, 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 (2000) 262–294.

Aruna Jamdagni is a PhD student at the Faculty of Engineering and Information Technology (FEIT) of the University of Technology, Sydney (UTS), also a research member of Research Centre for Innovation in IT Services and Applications (iNEXT). Her research interests include Computer and Network Security and on Pattern Recognition techniques and fuzzy set theory.

Zhiyuan Tan is a PhD student at the Faculty of Engineering and Information Technology (FEIT) of the University of Technology, Sydney (UTS), also a research member of Research Centre for Innovation in IT Services and Applications (iNEXT). His research interests are on Computer and Network Security and on Pattern Recognition techniques for efficient Network Intrusion Detection and anomalous behavior detection in P2P overlay network.

Xiangjian He is a Professor of Computer Science, School of Computing and Communications. He is also Director of Computer Vision and Pattern Recognition group, and a Deputy Director of Research Centre for Innovation in IT Services and Applications (iNEXT) at the University of Technology, Sydney (UTS). He is an IEEE Senior Member. He has been awarded 'Internationally Registered Technology Specialist' by International Technology Institute (ITI). His research interests are image processing, pattern recognition, computer vision and Network security. He is in the editorial boards of seven international journals. He has received various research grants including four national Research Grants awarded by Australian Research Council (ARC).

Priyadarsi Nanda joined UTS in 2001. He is Senior Lecturer, in the School of Computing and Communications, and Core Research Member, Centre for Innovation in IT Services Applications (iNEXT). His research interests are Network QoS, Network securities, Assisted health care using sensor networks, and Wireless networks. He has published 38 referred publications including two journals, four book chapters, one conference tutorial and 31 referred conference papers.

Ren Ping Liu is a principal scientist of networking technology in CSIRO ICT Centre, Australia and Adjunct Associate Professor, Macquarie University. His interests include scheduling, QoS modeling and performance analysis of wireless networks, including IEEE 802.11, Wireless Mesh Networks, Wireless Sensor Networks, LTE, and Cognitive Radio Networks. He delivered networking solutions to government and industrial customers, including Optus, AARNet, Nortel, Queensland Health, CityRail, Rio Tinto, and DBCDE.