

Chapter 1

A Statistical Pattern Mining Approach for Identifying Wireless Network Intruders

Nur Al Hasan Haldar, Muhammad Abulaish and Syed Asim Pasha

Abstract In this paper, we present a statistical pattern mining approach to model the usage patterns of authenticated users to identify wireless network intruders. Considering users activities in terms of ICMP packets sent, DNS query requests and ARP requests, in this paper a statistical approach is presented to consolidate authenticated users activities over a period of time and to derive a separate feature vector for each activity. The proposed approach also derives a local threshold for each category of network data analyzed. The learned features and local threshold for each category of data is used during detection phase of the system to identify intruders in the network. The novelty of the proposed method lies in the elimination of redundant and irrelevant features using PCA that often reduce detection performance both in terms of efficiency and accuracy. This also leads our proposed system to be light-weight and deployable in real-time environment.

1.1 Introduction

Due to easy installation, portability and mobility features, Wireless Local Area Networks (WLANs) are frequently being used by users from different walk of life. Although, WLANs solve some problems that exist in traditional wired LANs, there still exists certain vulnerabilities due to flaws in some IEEE 802.11 standard protocol which makes wireless network a highly desir-

Nur Al Hasan Haldar
Comviva Technologies Pvt. Ltd., Gurgaon, Haryana, India, e-mail: nurjamia@gmail.com

Muhammad Abulaish (*corresponding author*)
Center of Excellence in Information Assurance, King Saud University, Saudi Arabia
(On leave from Jamia Millia Islamia, New Delhi, India), e-mail: abulaish@ieee.org

Syed Asim Pasha
Ericsson India, Gurgaon, Haryana e-mail: asim.pasha2@gmail.com

able target in terms of security breach. Vulnerability exploits based remote attacks are one of the most destructive security issues faced by the security community as most of the high profile security threat of automatic dissemination attacks or worms are based on remote exploitation of vulnerabilities in compromised systems [10]. The open nature of wireless medium makes it easy for attacker to listen and analyze network traffic. Various useful security techniques like, AES, WEP, WPA or WPA2 can protect data frames, but an attacker can still spoof control or management frames to damage security. These attributes makes wireless network potentially vulnerable to several different types of attacks. Although, a number of security measures have been already proposed by security researchers and they are operational to protect wireless networks, one cannot make the assumption that wireless users are trusted. Malicious individuals can easily sit outside or inside an organization's premises, and freely connect to a wireless network to do malicious activities.

Intrusion detection is generally used to secure any system in a network by comparing the set of baselines of the systems with their present behavior [8]. The main two known models of intrusion detection system (IDS) are (i) signature-based intrusion detection, and (ii) anomaly-based intrusion detection. The first one uses signatures of the well-known attacks to detect intrusion. A signature-based detection technique facilitates intrusion detection by investigating various routing protocols when attack signatures are completely known. This type of detection monitors the wireless networks for finding a match between the network traffic and a well-known attack pattern. But, this approach suffers with a number of limitations including the inability to cope up with exponential increase in new malicious exploits and consequently the huge size of signature database, inability to detect new and unfamiliar intrusions even though they are very similar to the known attacks. Further, the signature-based techniques can not be used to detect zero-day exploits. In contrast to signature-based IDS, anomaly-based IDS creates profiles that are based on the normal behavior of the underlying network [2]. Anomaly-based IDS first establishes a model of normal system behavior and anomaly events are then distinguished based on this model. So in this approach, the detection is performed by learning the normal behavior of a network and comparing it with the behavior of monitored network. This type of detection has the ability to detect new unknown attacks (zero-day attacks) without any prior knowledge about them. However, the false alarm rate in such systems is usually higher than that in the signature-based systems [11].

Considering this scenario which necessitates to monitor users activities of a network to identify possible intruders, in this paper we have proposed a statistical pattern mining approach to design a wireless intrusion detection system. A poster version of this work appears in [14]. The proposed approach gathers authenticated users activity data in the network and applies statistical pattern mining to derive feature vectors to characterize them in the system. The characterization is later used during detection phase of the system to identify malicious users or intruders whose characteristics substan-

tially deviates from the activity patterns of the normal users. Some of the features monitored by the proposed IDS are ICMP packets sent, DNS query requests, and ARP requests. In order to enhance the performance of the proposed system both in terms of efficiency and detection accuracy, Principal Component Analysis (PCA) is applied on captured traffic data to filter out irrelevant components and map them into a lower-dimensional space. The method described in [1] is found to be very sensitive to small changes in a noisy environment at a reasonable label, which causes many non-negligible numbers of false alarms. In order to reduce such false alarms and to increase the effectiveness, we enhance the PCA-based detection to identify anomalous nodes. Also the threshold estimation approach presented in this paper helps in reducing the false alarm rate.

Starting with a brief review of the existing intrusion detection techniques in Section 1.2, section 1.3 introduces the process of dimension reduction and pattern extraction using PCA. Section 1.4 presents the design of the proposed system, followed by the experimental results in section 1.5. Finally, section 1.6 concludes the paper with future directions of work.

1.2 Related Work

Due to increasing security issues related to wireless networks, a number of research efforts have been directed towards identification and prevention of network attacks. As a result, there exists many intrusion detection systems that can detect an attack at the IP layer or above [3]. Hu and Perrig [3] proposed a signature-based intrusion detection method to detect wormhole attacks. In [13], Mukkamala et al. proposed a method for audit trail and intrusion detection using SVM and neural network approach. Some other approaches including statistical methods, wavelet analysis, data mining techniques, and Kolmogorov-Smirnov test have been also proposed by various researchers. Besides its usefulness to detect the anomalous traffic, the Kolmogorov-Smirnov test [4] fails to identify the cause of the anomaly. Similarly, the anomaly detection methods using wavelet analysis suffers with heavy computational overheads and consequently they can not be deployed in real-time environment. In [12], Portnoy et al. proposed an algorithm to detect both known and new intrusion types using a clustering approach which does not need labeled training data, but it requires a predefined parameter of clustering width which is not always easy to determine.

It can be seen that the above-described methods are mostly heavy-weighted and most of them cannot be deployed in real-time as those require complete dataset for processing. On the other hand, PCA-based methods are relatively light-weight and processing speed is also high. The proposed work by Feather et al. in [6] is very similar to PCA-based method. They used some signature matching algorithm to detect anomalies. Another work is re-

ported by Dickinson in [5], where anomalous traffic is compared against certain threshold. In the combined work reported by Wang et al. [7], the intrusion detection techniques are developed using profile-based neighbor monitoring approach. They used Markov Blanket algorithm for the purpose of feature selection to decrease the number of features dramatically with very similar detection rate. But the profile-based neighbor monitoring intrusion detection approach requires a lot of network features to monitor. For improved detection rate, our method identify effective features by ignoring uncorrelated variables using PCA.

1.3 Dimensionality Reduction and Pattern Extraction using PCA

Generally, consideration of a large set of features results in too many degrees of freedom, which leads to poor statistical coverage and thus poor generalization. In addition, each feature introduces a computational overhead both in terms of efficiency and storage. Principal Component Analysis (PCA) is a statistical method for dimension reduction that maps high-dimensional data points onto a lower-dimensional set of axes that best explain the variance observed in the dataset [9]. The purpose behind using PCA is to replace the original (numerical) variables with new numerical variables called “principal components” that have the following properties. (i) The principal components can be ranked by decreasing order of importance, and the first few most important principal components account for most of the information in the data. (ii) There is almost as much information in the principal component variables as there are in the original variables. In this way, PCA can be perceived as a technique for dimensionality reduction to eliminate irrelevant data from the original dataset. The working principle of PCA can be summarized as follows.

- Organize data as an $m \times n$ matrix $A_{m \times n}$, where m is the number of measurements and n is the number of trials (dimensions).
- Subtract mean from each of the data dimensions, where mean is the average across each dimension. This produces a data set whose mean is zero.
- Generate covariance matrix $Cov_{n \times n}$ for $A_{m \times n}$ using equation 1.1 to calculate the covariance between a pair of dimensions dim_i and dim_j .

$$cov(dim_i, dim_j) = \frac{\sum_{k=1}^m (dim_{i_k} - \overline{dim_i})(dim_{j_k} - \overline{dim_j})}{m - 1} \quad (1.1)$$

- Calculate the eigenvectors and eigenvalues of the covariance matrix $Cov_{n \times n}$. Since $Cov_{n \times n}$ is a square matrix, we can calculate the eigenvectors and eigenvalues, describing important information about the patterns in data.
- Select eigenvalues and form feature vectors.

Here the role of PCA in data compression and dimensionality reduction comes in picture. Generally, different eigenvalues are quite different values and the eigenvector with the highest eigenvalues is the principal component of the dataset. For feature selection, we order the eigenvectors by their eigenvalues (highest to lowest), which gives us the components in order of their significance and ignore the components of lesser significance. Finally, we form the feature vector by arranging the selected eigenvectors in the form of a matrix.

1.4 Proposed Statistical Pattern Mining Based IDS

In this section, we present the procedural detail of the proposed statistical pattern mining based intrusion detection system, which is shown in figure 1.1. Our system processes the following three types of data in the network to learn the normal usage pattern of authorized users using PCA and generates a profile as a feature vector for each of them. The generated profiles are then used as a baseline to identify intruders in the network.

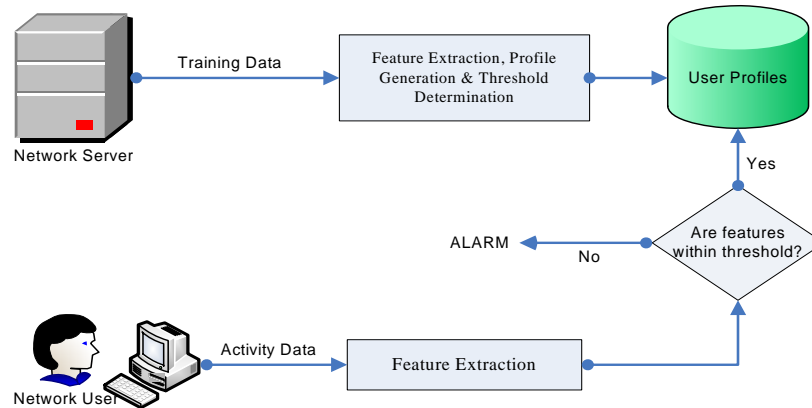


Fig. 1.1 Architecture of the proposed intrusion detection system

ICMP packets sent – Internet Control Message Protocol (ICMP) is a network layer protocol, which is generally responsible to relay query message and also to send some error message if either a requested service is not available or a host or a router could not be reached. Attacks like *Smurf* and *Papasmurf* happen when intruder sends a large amount of ICMP packets (ping) to a subnet broadcast address. ICMP can be used as a first step in an attack because it can determine the alive hosts before attacking. Therefore, ICMP packet is considered as a parameter and the number of ICMP requests sent and received is monitored by our system.

DNS query requests – Domain Name System (DNS) query is used to translate domain names into the actual IP address of destination machine. DNS spoofing attack changes the entry of IP address of a domain name to some other IP address. Therefore, the DNS requests of a user are considered to model his/her domain of interests.

ARP requests – Address Resolution Protocol (ARP) is responsible to map an IP address to its associated data link layer address (MAC address). The ARP requests can be spoofed by the intruders to divert the packets to a wrong destination. Also, attacker can block traffic resulting in Denial of Service (DoS) attack. So, ARP request is also an important feature to characterize intruders in the system.

Like standard classification methods such as Naive Bayes, Decision Tree, Support Vector Machine (SVM) and Neural Network, our intrusion detection system is implemented as a two-phase process given below.

Phase-1 (Training phase): This is basically profile generation phase. In this phase, the activity data of authorized users are collected and their profiles are generated using PCA algorithm. Threshold value to determine the maximum profile deviation of a normal user from the profiles of the authenticated users is also determined during this phase.

Phase-2 (Detection phase): This is also called profile detection phase. In this phase, the learned profiles are used as a baseline to identify possible intruders in the system. If the Euclidean distance of a user profile with the learned profiles of the authentic users is greater than the pre-determined threshold then an alarm is generated and the user is classified as a possible intruder and consequently the packets are dropped. Otherwise, the user is classified as a normal user and its profile is added in the profile set of the authenticated users as shown in figure 1.1. In this way, the profile set and thereby the intrusion detection accuracy of the system increases with time. Further detail about the profile generation and threshold determination phase is presented in the following sub-section.

1.4.1 Profile Generation and Threshold Determination

In this phase, we train the application to learn the usage patterns of the known users after analyzing their activity data. For this, we capture the activity data related to different type of features for each authenticated user u_i in the system and organize them into an $n \times m$ matrix A , where n is the number of slots and m represents the number of days for which data are captured. Thereafter, we apply PCA on matrix A to select $p < m$ eigenvectors corresponding to high eigenvalues, which forms an $n \times p$ matrix. We say this matrix a weight matrix and represent it using W . In order to get an aggregated pattern for the user under consideration over the considered time-period, we apply scalar product between each column vectors of W and

matrix A using equation 1.2. This scalar product gives a matrix P of order $m \times p$ in which the columns correspond to the selected eigenvectors and the rows corresponds to the days considered for profile generation. Finally, each column of matrix P is averaged and used to generate a profile of the user u_i as a p -dimensional vector $\psi(u_i) = \langle w_1, w_2, \dots, w_p \rangle$, where the value of w_k ($k = 1, 2, \dots, p$) is calculated using equation 1.3.

$$P = A^T.W \quad (1.2)$$

$$w_k = \frac{\sum_{j=1}^m \text{col}_{k_j}(P)}{m} \quad (1.3)$$

For each class of user activity data, this process is repeated to generate the profile of all authenticated users. All generated profiles are considered to form a single cluster and its centroid is calculated as an average of the corresponding elements in the profile vectors. Dissimilarity between a pair of two profile vectors $\psi(u_i) = \langle w_1, w_2, \dots, w_p \rangle$ and $\psi(u_j) = \langle w'_1, w'_2, \dots, w'_p \rangle$ is calculated as an Euclidean distance between them using equation 1.4. And, the maximum of these distances over all user profiles is decided as threshold value as given in equation 1.5. This threshold value is used in profile detection phase to identify possible intrusions in the network. For a given user, his/ her profile is generated using the above-discussed method and its distance from the generated centroid is calculated. If the distance value is within the threshold then the profile and thereby the data packet is considered as benign, otherwise it is considered as malicious and an alarm is raised.

$$\delta(\psi(u_i), \psi(u_j)) = \sqrt{(w_1 - w'_1)^2 + (w_2 - w'_2)^2 + \dots + (w_p - w'_p)^2} \quad (1.4)$$

$$\theta = \left[\max_{1 \leq i < n, i < j \leq N} \{ \delta(\psi(u_i), \psi(u_j)) \} \right] \quad (1.5)$$

1.5 Experimental Setup and Results

In this section, we discuss our experimental setup and results. All experiments are performed on a PC with Intel Core Duo 1.66 GHz processor, and 2 GB RAM. For simulation, we have used SIGCOMM¹ 2008 dataset, which contains three types of anonymized traces – 802.11a, Ethernet and Syslog. After downloading dataset, we discarded other irrelevant data like wired traces, Syslog traces, etc. and took only wireless data for our experiment.

¹ <http://uk.crowdad.org/meta.php?name=umd/sigcomm2008>

Table 1.1 Sample data files and the no. of filtered ICMP, DNS and ARP packets

Data File Name	ICMP packets	DNS packets	ARP packets
sigcomm08_wl_10_2008-08-19_11-22_23_2008-08-19_18-07_56_b88796010035_36.pcap	565	8250	709
sigcomm08_wl_13_2008-08-19_13-10_42_2008-08-19_19-24_12_b88796010035_36.pcap	379	4530	409
sigcomm08_wl_10_2008-08-20_10-50_35_2008-08-20_17-46_56_b88796f1e2c7_52.pcap	4616	18655	1044
sigcomm08_wl_16_2008-08-20_11-29_26_2008-08-20_17-52_37_b887965a5527_36.pcap	35188	16724	1092
sigcomm08_wl_9_2008-08-20_10-46_31_2008-08-20_17-44_57_b887965a5527_36.pcap	22610	10367	733
sigcomm08_wl_8_2008-08-20_10-58_15_2008-08-20_17-50_06_b88796a13ec7_149.pcap	1536	1982	495

Since the wireless dataset contains “pcap” files, we have used Wireshark² for analysis and filtering purposes. For all days, the overall start packet time and last packet time are noted at 10:32:32 hrs and 19:24:12 hrs respectively, resulting in total 31900 seconds data per day. After filtering the dataset, we merged all individual file data day-wise with respect to available packets’ time. The all days data were sorted after merging in ascending order with respect to start packet time. Then we calculated the number of packets in a fixed interval (100 seconds) for each day. Some sample instances along with data file name and no. of different types of packets are given in table 1.1.

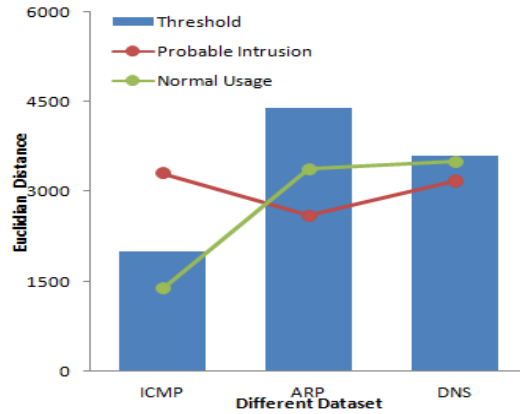
$$W = 1.0e + 003 * \begin{pmatrix} -0.2458 & 0.1224 & 0.0033 \\ -0.2458 & 0.1227 & 0.0042 \\ -0.2458 & 0.1227 & 0.0042 \\ \dots & \dots & \dots \\ -0.2449 & 0.1167 & -0.0064 \\ -0.2457 & 0.1198 & 0.0051 \\ -0.2458 & 0.1227 & 0.0042 \end{pmatrix} \quad (1.6)$$

We have selected a particular machine (MAC address) from the wireless dataset and gathered the total number of packets (ICMP, DNS and ARP) transmitted by that machine. We took five days data (31900 seconds data each day) for profile generation from the above-mentioned dataset. After observing the captured data pattern, we manipulated them to fit the total number of packets in time interval of 100 seconds. The time interval may vary according to the packet rates of the captured dataset for better result. Thus, we got total 319 slots and calculated the number of packets in each slot of 100 second interval. In this way, the length of the column vector, representing each day data, is 319 resulting in 319×5 matrix for the five days data. After applying PCA on its covariance matrix, the first three eigenvectors corresponding to top-three eigenvalues forming 319×3 weight matrix is shown in equation 1.6.

² <http://www.wireshark.org/download.html>

Table 1.2 Weight components for day-wise data

	w_1	w_2	w_3
Day 1	1562891	-916210	-202630
Day 2	2173000	-6086500	545010
Day 3	39677000	485680	104580
Day 4	-49121	-236750	-17780
Day 5	2735800	-1673300	-1838700

**Fig. 1.2** Visualization of threshold values for ICMP, ARP and DNS datasets, and two user profiles - one (green curve) benign and other (red curve) malicious

Thereafter, the weight components corresponding to the highest three eigenvalues for each day is calculated using equation 1.2. The resultant matrix data after this operation is shown in table 1.2. Finally, each column of table 1.2 is averaged to generate the user profile as a 3-dimensional vector. This process is repeated for all the authenticated users and for all three different types of data - ICMP, DNS, and ARP. The centroid vector for each data type is calculated by taking the average of the corresponding components of the user profile vectors. Similarly, the threshold value for each type of data packet is calculated using equations 1.4 and 1.5.

Once the centroids and threshold values for all datasets are fixed they are used to identify possible intrusion attempts in the system. For a user under consideration, if any of the corresponding thresholds is crossed then this indicates that a probable intrusion attempt is made by the user, whereas on the other hand if the user profile regards all threshold values, it is considered as benign and added in the profile set to derive finer tuned user profile. Figure 1.2 shows the threshold values for ICMP, DNS and ARP datasets, and two user profiles – one (green color) identified as benign user, whereas the other one (red color) as a malicious user.

1.6 Conclusion and Future Work

In this paper, a light-weight statistical pattern mining based wireless intrusion detection system is proposed to model authenticated users activities to identify intrusions in wireless networks. Various network layer data are collected and statistical pattern mining approach is applied to identify usage patterns of authenticated users for their characterization. For efficiency purpose, the proposed system applies PCA algorithm on traffic data to map them into lower-dimensional space. Presently, we are working towards the evaluation of the proposed system on different real datasets. The proposed method can be extended to protect wireless routers from malicious attacks.

References

1. Lakhina, A., Crovella, M., Diot, C.: Diagnosing network-wide traffic anomalies. In: Proc. of the ACM SIGCOMM'04, NY, USA, pp 219–230 (2004)
2. Sekar, R., Gupta, A., Frullo, J., Shanbhag, T., Tiwari, A., Yang, H., Zhou, S.: Specification based anomaly detection: a new approach for detecting network intrusions. In: Proc. of the 9th ACM CCS, NY, USA, pp 265–274 (2004)
3. Hu, Y.C., Perrig, A., Johnson, D.B.: Wormhole attacks in wireless networks. *Journal on Selected Areas in Communications*, 24(2), pp 370–380 (2006)
4. Caberera, J.D., Ravichandran, B., Mehra, R.K.: Statistical traffic modeling for network intrusion detection. In: Proc. of the 8th Int'l Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, pp 466–473 (2000)
5. Dickinson, P., Bunke, H., Dadej, A., Kraetzl, M.: Median graphs and anomalous change detection in communication networks. In: Proc. of the Information, Decision and Control, Australia, pp 59–64 (2002)
6. Feather, F., Siewiorek, D., Maxion, R.: Fault detection in an ethernet network using anomaly signature matching. In: Proc. of the ACM SIGCOMM'93, NY, USA, pp 279–288 (1993)
7. Wang, X., Lin, T.L., Wong, J.: Feature Selection in intrusion detection system over mobile ad-hoc network. Technical Report, Iowa State University, USA (2005)
8. Mishra, A., Nadkarni, K., Patcha, A.: Intrusion detection in wireless ad-hoc networks. *IEEE Wireless Communications*, 11(1), pp 48–60 (2004)
9. Smith, L.I.: A tutorial on Principal Components Analysis (2002)
10. Wang, H.J., Guo, C., Simon, D., Zugenmaier, A.: Shield: vulnerability-driven network filters for preventing known vulnerability exploits. *SIGCOMM Comput. Commun. Rev.* (2004)
11. Garcia-Teodoro, P., Diaz-Verdejo, J., Macia-Fernandez, G., Vezquez, E.: Anomaly-based network intrusion detection: techniques, systems and challenges. *Computers and Security*, 18–28 (2009)
12. Portnoy, L., Eskin, E., Stolfo, S.: Intrusion detection with unlabeled data using clustering. In: Proc. of ACM CSS Workshop on Data Mining Applied to Security, pp 5–8 (2001)
13. Mukkamala, S., Janoski, G., Sung, A.: Intrusion detection using neural networks and support vector machines. In: Proc. of the Int'l. Joint Conf. on Neural Networks, pp 1702–1707 (2002)
14. Haldar, N. Al-H., Abulaish, M., Pasha, S. A.: An activity pattern based wireless intrusion detection system. In: Proc. of the 9th Int'l. Conf. on Information Technology – New Generations, Las Vegas, USA (2012)