The VoIP Intrusion Detection through a LVQ-based Neural Network

Zheng Lu
Edinburgh Napier University
z.lu@napier.ac.uk

Taoxin Peng
Edinburgh Napier University
t.peng@napier.ac.uk

Abstract

Being a fast-growing Internet application, Voice over Internet Protocol shares the network resources with the regular Internet traffic. However it is susceptible to the existing security holes of the Internet. In this paper, a highly effective VoIP intrusion detection approach based on LVQ neural network is proposed. This detection approach is particularly suitable for protecting VoIP applications, in which various protocols are involved to provide IP telephony services. Experiments of the proposed approach show promising detection accuracy and a low runtime impact on the perceived quality of voice streams.

1. Introduction

The IP telephony, commonly known as *Voice over IP* (VoIP), is emerging as a viable alternative to traditional telephone systems. The classic VoIP Signaling protocols H.323 and SIP, and media transport protocols: RTP and RTCP could become a target of well-connected attacks, such as Denial of Service (DoS), eavesdropping, fraudulent usage, etc. Although a variety of research on VoIP intrusion detection systems at the application, transport and routing level has been done not only in industry but also academic community, feasible detective approaches and implementation on VoIP infrastructures are still at their early stage. This is mainly because most of VoIP intrusion detective systems (vIDS) are motivated by existing network Firewalls that ignore some critical features of VoIP networks. For instance, the Backpropagation Neural Network (BPN) vIDS which is modified from SPAM mail/Internet packet filter is widely adopted in several nation-wide softswitch infrastructures. The experiment of simulative attacks by several well-known VoIP infrastructure providers has proved that such a structure easily causes false negative and low performance due to its internal pitfalls such as local minima and convergence [1].

This paper presents a novel signaling layer's Learning Vector Quantization (LVQ) based vIDS, which is implemented in the SIP protocol in a nation-wide softswitch that was manufactured by a well-known telecom system provider in order to replace the following three existing approaches: BPN, Bayesian Inference, and Finite State Machine approach. Since the SIP protocol which is a popular signaling exchange protocol can be seen as a nucleus of a nation-wide softswitch, the vIDS technology which is successfully implemented in SIP protocol can be regarded as a feasible intrusion detective method for a large-scale softswitch.

This proposed detection approach is particularly suitable to be deployed in large-scale softswitch systems, which involve multiple VoIP protocols. Through the interaction with a well-chosen training dataset, the approach was tested with a benchmark intrusion dataset on a nation-wide VoIP infrastructure. The result of initial experiments proved that the LVQ-based intrusion detection system shows greater advantages over existing VoIP intrusion detection systems on aspects of effectiveness, response time and accuracy etc.

2. Background

2.1. SIP/VoIP security threats

As a popular commercial transmitting protocol in the Internet, the SIP protocol has been assembled with several security capabilities, such as encryption of SIP message body, and a security mechanism in Session Description Protocol (SDP). However, although message bodies are encrypted, such as the headers From, To, time stamps are unable to be encrypted step-by-step due to unbearable cost on proxy servers. In practice there are mainly 8 types of frequent attacks: Call Tracking & Messages Interception, Fraudulent Usage, User Enumerating and Password Cracking, Call Hijacking and 'Man In the Middle' attack, DoS, Attacks Against Voice Mail Server (VMS), Attacks on Media Protocols, Firewall Traversal [2]. Among these attacks, User enumerating and DoS are the most common attacks both on VoIP infrastructures and VoIP networks. Therefore it is easy to detect these two

attacks from normal activities according to their obvious intrusive patterns. On the other hand, since attacks such as Attack on Supporting Protocols infrequently occur, which proportion is less than 0.1% of attacks occurrence, they might be quite difficult to be detected due to their ambiguous patterns. Therefore manually searching attack patterns from log files seems only an effective detection method in despite of a great manpower cost [2].

2.2. An Architecture of Nation-wide Softswitch systems

The architecture of VoIP nation-wide systems is an infrastructure which consists of a virtual network based on Converged IP/MPLS Backbone networks which combine with a Multi-Service Access network that can fully support mainstream access network protocols including Frame Relay and Ethernet, TDM, ATM, 3G etc. Moreover, this VoIP architecture offers capabilities to support multiple VoIP access protocols such as SIP, TDM/SS7, MEGACO, H.323, MGCP, MEGACO as well as some emerging VoIP protocols which might be released in the future. This is achieved by employing Border Elements (BEs) on the VoIP Connective Layer. The BEs define the access points/ boundaries in the VoIP Infrastructure and translate the specifications of all other VoIP protocols into Session Initiation Protocol (SIP) – the nucleus protocol used by all VoIP foundation components.

In general, in nation-wide Softswitch Border Elements act as not only a role of the protocol translator, but also resident points of various security and administrational policies such as Bayesian and BP-based vIDS.

The Call Control Element (CCE) manages the VoIP infrastructure and provides a synchronized access point to external application servers from service providers. Involving with multiple BEs, the Call Control Element can control all call legs to create and supervise the connection between end-points. Since a state machine based vIDS works or not depending on a detection of the deviation of a normal SIP state, it is generally deployed at the CCE layer.

Application servers locating at the application layer provide both a knowledge-based repository and a log system that traces ongoing call sessions.

Since all the functionalities deployed in the hot-pluggable blade can be rapidly replaced or updated, this model offers a capability of flexibly upgrading a system without significant concerns of the VoIP infrastructure. Figure1 shows the architecture of the VoIP infrastructure.

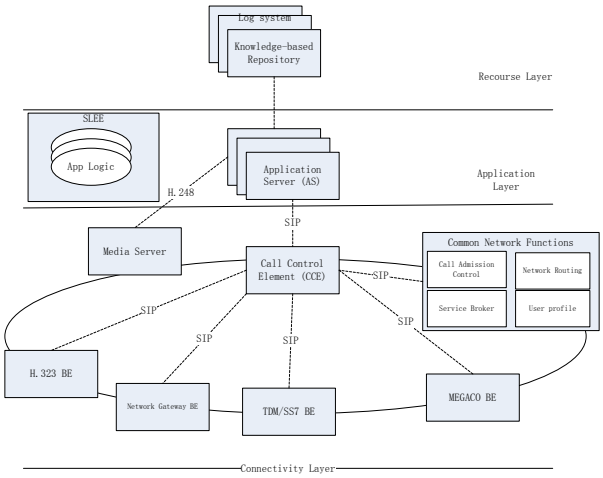

Figure1. VoIP infrastructure

2.3. Existing VoIP IDSs

Since the vIDS process usually detects data packets at application layer, it is deployed at the second line of a defense system behind Firewalls and Encryption. Firewalls are used to detect abnormal activities at the transport layer, while Encryption encodes IP packets [3]. In existing vIDSs, in order to effectively and accurately detect attacks, a detection engine is equipped with a knowledge-based repository. There are four general approaches to vIDSs, which are implemented in large-scale VoIP systems: Bayesian, BPN, Finite State Machine and Decision Tree. Among the four approaches, the BPN based achieves the highest balance capability under the condition of having the same accuracy by using a relatively low executing performance comparing with the other three [1]. Furthermore the BPN expresses a strong self-adaptive self-learning and self-repaired ability when facing new types of intrusions [1].

However, some inevitable problems restrict BPN-based approaches to be widely implemented in VoIP systems [4, 5, 6]:

a. The BP network is unpredictable to slip into bottom of local minima instead of the expected global minimum during training, or gradient descent might be on plateau during Dos or SIPT training.
b. Unavoidable Overfitting problem while system taught with million training set.
c. The momentum has to be carefully chosen during training, otherwise system might be misled.
d. Low training performance when system is taught with large number of training set.

Since a variety of research about neural networks indicates that a LVQ neural network is a promising algorithm which can overcome above inherent problems of BPN algorithm, a LVQ-based vIDS is an

apparently reasonable alternative to BP-based vIDS [5,7].

2.4. The LVQ Algorithm

A LVQ is a prototype-based supervised classification neural network algorithm, which is widely used in biological identification and firewall systems.

A detection of network attacks can be supposed as a classification process in which malicious activities are classified differently from normal activities according to their patterns [5, 7, 8].

A LVQ neural network consists of three layers: input competitive and output. A weight associates with each input neuron. During the training process, the weights of neural network are changed by training data in order to classify input data correctly. For each input data, the neuron at input layer that is closest to it is determined (called the winning neuron). The weights on the connections to this neuron are then adapted, i.e. made closer if it correctly classifies the input data or made less similar if it incorrectly classifies it. [9]

The LVQ based algorithm proposed by Zhan *et al* [5] is described below:

a. Initialize prototypes weight vectors W= { w_1, w_2, … w_n}, learning rate $\alpha \in [0, 1]$
b. Repeat following steps until stopping condition is satisfactory;
c. Select an example from training data of malicious activities collections, and compute vector cosine distance between weights and it respectively according to followed formula:

$$\mathrm{Sim}(U,V) = \frac{\sum_{k=1}^{n} W_{uK} * W_{vK}}{\sqrt{\sum_{k=1}^{n} W_{uK}^2} * \sqrt{\sum_{k=1}^{n} W_{uK}^2}}$$

Sim (U, V) is the geometric distance between vector W_u and W_v.

d. The similarities between the example and each weight vector are compared, in the result, the neuron with maximum similarity, wins a competition and output 1, other neurons output 0.

$$a' = \max(\mathrm{Sim}(x, x'))$$

e. If an example is classified to class r, the neuron c which wins competition in learning process belongs to class s, the weight will be adjusted to move neuron c toward the center of class r, otherwise move it away from class r. Change weight in accordance with formulas:

$$\begin{cases} W_c(t+1) = W_c(t) + u(t)[x(t) - W_c(t)] & \text{belongs to calss} \\ W_c(t+1) = W_c(t) - u(t)[x(t) - W_c(t)] & \text{not belong to class} \end{cases}$$

f. Increase or descend learning rate u (t), while iterating.

3. Main achievement

3.1. System design of LVQ-based vIDS

In the design, concerning the complexity of LVQ architecture and the high cost of the multiplayer LVQ, a single-hidden layer LVQ network which is taught by a purposely selected attack training pattern is used. One engine equipped with a single-layer LVQ-based vIDS can detect only one type of attacks. For example a LVQ network taught by a DoS training set can only recognize DoS intrusions in the network traffic without particular interest in other attacks. This method can effectively avoid problems such as how to prepare a complicated and accurate training dataset for teaching multilayer LVQ vIDSs and overweight LVQ architectures. However this vIDS is available only when the system hardware cost is not a significant concern in large-scale systems, and the vIDS will not cause serious performance bottlenecks, which can be achieved by adding a new hot-plug blade with a particular vIDS detective method installed into the VoIP Softswitch infrastructure. At the input layer, 24 types of input data are used (including network message intensity, system resource consumption, and message distribution,) as observed variables. These variables are selected from enterprise attack recorder log files according to their value of information gain [3]. 40 neurons are arranged at the competitive layer with a winning function that calculates the geometric distance and then takes out a competitive neuron. The selection of these numbers is based on the experiment that shows that matrix 24 inputs x 40 neurons can achieve the optimal performance and accuracy while taught with 800-1000 training sets. At the output layer, 2 output neurons is set as 2 types of classes (legitimate or malicious) which is linked with 20 neurons respectively, which determine which output class is activated. The 800 -1000 concise training sets for each intrusion engine are extracted from an enterprise attack record log by a Sample-Based training data approach which produces a typical training set. Figure 2 illustrates the internal structure of a LVQ-based vIDS.

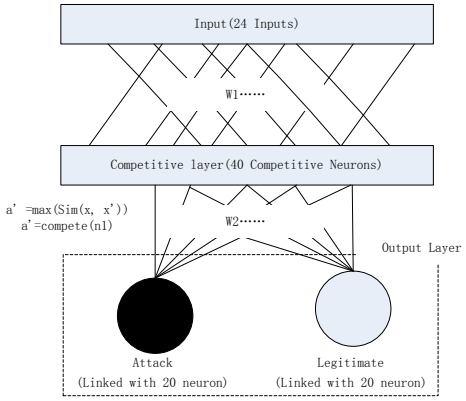

Figure 2. LVQ-vIDS Model

3.2 Design of training data

Since an effective intrusion detection usually depends on the high quality training data, a novel method that can be used to create training data more efficiently is also proposed by using BPNs with Sample-Query and Attribute-Query. The proposed method has a relatively low cost of selecting training data from mass log files.

3.2.1. Query-based training. Since a learning network is taught by accumulation of input information, a deduction algorithm can be used to extract training data from log files. For a learning protocol, the input information can be seen as examples that exemplify the concept to be learned, so-called oracle which can present whether or not the data exemplifies the concept. Consequently, after samples are obtained from log files captured from network traffic, they have to be examined by the oracle. The oracle often works in model of query-answer. When the point of query is set as y, the oracle would respond with a(y). The pair (y, a(y)) which is called queried sample can be used to extract useful training data from log files.

All training samples are stochastically separated into a training set and a query set. The arbitrary oracles are designed to follow the self-regulation rule to select samples (environment-focus) [8] which are close to a conjugate data pair (self-focus). The process of designing an oracle provides the system with an ability to interact with the real training environment to produce a precise training data by querying samples. Based on the query-based training approach, the LVQ vIDS is taught with partial specific samples from the log files to achieve the target of being completely taught with full training data. In this approach, since an oracle is designed to gain appropriate samples (boundary samples) for further training, the learning performance of an LVQ neural network is remarkably improved by well-chosen training data that is expected to be taught.

The process of the designed Query-based training consists of following steps, suggested in [11].

a. The 'rough' training samples extracted from log files are examined by oracles in order to detect whether they are classified in a wrong class, meanwhile the distance from the classification boundary to the sample is calculated to determine to what extend the training will be affected by the sample.
b. The misclassified samples can be stored in a priority queue, which can be considered as an extra training data set that is the most close to the class boundary.
c. The samples correctly classified will be ignored; because they reside deeply inside in class region which does not effects classification.
d. The training data in the priority queue is output as concise subsets of training data for the LVQ neural network training

3.2.2. Attribute-Query. According to [8], the learning accuracy and efficiency of a neural network will be degraded by having useless attributes (redundant observed variables). Generally, 48-50 network attributes are recorded in a log system from the VoIP network traffic. These attributes will be reevaluated by the Attribute-Query approach, in which the algorithm of information gain analyzes each attribute. Attributes with a relatively low information gain are not likely to be useful.

Entropy is a measure of degree of doubt. The information gain is simply the expected reduction in entropy caused by partitioning the examples according to this attribute. The entropy for each attribute is calculated based on the following formula:

$$\text{Entropy}(S) = -\sum_{i=1}^{n} P_i * \log(P_i)$$

where S denotes the training data, Entropy(S) is the entropy of S relative to n-wise classification if the target attribute can take on n-different values; P_i . is the proportion of S belonging to class i. In the proposed method, P_i is the proportion of legitimate and attack in training samples S.

Information gain G[S, A] of attribute A is calculated by formula

$$G[S, A] = \text{Entropy}(S) - \sum_{v \in \text{Value s}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

where Values(A) is the set of all possible values for attribute A (e.g. network package intensity (I, 1<I<4) has 4 possible integer values: I∈[1,2,3,4]). S_v is the subset of S for which attribute A has value v, Entropy (S_v) is the entropy of which the attribute A has value v. The second term of Equation.2 is sum of the entropies of each subset S_v.

Since the objective of Attribute-Query is to reduce

redundant attributes in training set, the information gain of each attribute is computed in order to determine whether this attribute has a strong influence on information of the whole system. The attribute which has relatively small information gain is regarded as having a relatively weak impact on the LVQ-vIDS.

Results of the experiment show that the attribute with information gain value less than 0.05 would be regarded as a weak attribute in the LVQ-vIDS. Finally 24 network attributes with relatively high information gain are chosen as parameters for computing observed variables in the LVQ-vIDS.

3.2.3. Method of producing training data. Since Attribute-Query and Sample-Query approaches can eliminate redundant information existing in training data, it is reasonable to integrate features from these two approaches into a method which can produce high-effective training data. This paper proposes such a method for extracting the concise training set for neural networks.

A step-by-step description of the proposed method is shown below. The process is terminated either the number of iterations or the root-of-mean-squared-error (RMSE) is over the given thresholds.

a. Initialize all weights in the BPN network. The iteration threshold N is a value between 1000 and 1200, which is decided by convergence of BPN training. RMSE is set as 0.003.
b. Given training dataset S with attributes which is selected by Attribute-Query. Pick out the partial training samples SS (subset) from S by stratified random sampling, in which a stratum is formed, based on their samples sharing an equal value of a specific attribute.
c. Train the neural network by SS.
d. IF (the error E < RMSE) or (the iteration number > N) then go to g.
e. Examine non-trained samples (S - SS) and heap, input the remaining training samples into BPN, extract misclassified samples and add them into a heap, use the heap to train BPN again.
f. Using predefine oracles to verify the heap to pick out samples which are close to the classification boundary in the heap and then go to d.
g. Export the heap as a concise training set to the LVQ neural network. The heap contains concise training for each attack pattern which consists of 550-600 attack training samples and 400-450 legal training samples.

Experiments show the training data chosen by above method can dramatically reduce the training time up to 30% and improve detection accuracy (4%-10% dependent on type of attack).

3.3. Results of Experiments

The evaluation compares experiment results from traditional VoIP intrusion detection methods mentioned above and the proposed LVQ-based intrusion detection approach on a set of vIDS standard measures: accuracy, sensitivity, system overhead on SIP and RTP, which are suggested as measure factors in [12], together with another popular measure factor: synthetic performance suggested in [5]. The experiment was conducted in a nation-wide softswitch infrastructure, which is constructed from a multi-service VoIP network based on the IP/MPLS Backbone Network.

The proposed LVQ-based vIDS achieves more than 99.5% average accuracy when detecting 8 mainstream types of attacks, comparing that with 94% of the BPN based approach, and 90% of the Bayesian inference based approaches. Figure 3 illustrates the attacks and the amount of detections being successfully detected by the proposed LVQ-based vIDS during the experiment:

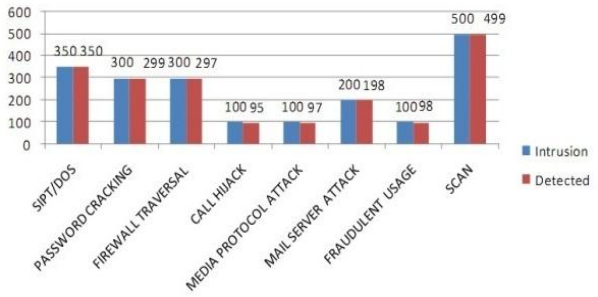

Figure 3. Accuracy of detecting 8 types of VoIP attacks

The delay caused by the LVQ-based vIDS is between 10ms and 12ms depending on the type of attacks, comparing with the average of 12ms of BPN-based and 9ms of Bayesian Inference based, which is not significant. Along with an increase of the number of incoming calls, performance of the LVQ-based vIDS doesn't obviously get worse: delay for each call will be added by 7-14 ms (vs. 10ms-23ms of BPN, 10-18ms of Bayesian) at peak of incoming calls (25 CAPS). See Figure 4 for details.

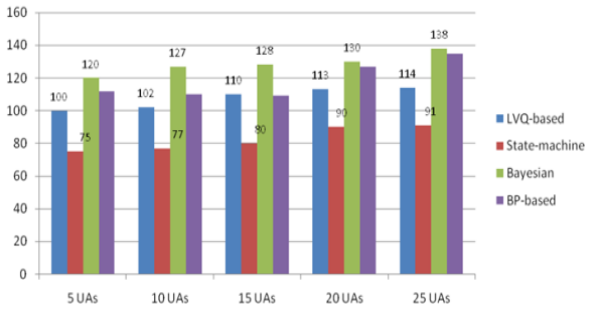

Figure 4. Call delays of 4 intrusions detect approaches

Although the system overhead (memory consumption and CPU usage) increases nonlinearly along with a growth of incoming calls, the system can still work properly at 7x24 hour peak calling under call simulation test.

The synthetic performance is a synthetic evaluation factor that represents detection accuracy and proportions of negative false while two systems are taught by the same amount of training data. Comparing with Bayesian-based and BP based approaches; The LVQ-based vIDS has the best synthetic capability. For example, with 1000 training sets, the average negative false of around 500,000 simulative attack samples is 0.004% vs. 0.05% (BPN based) and 0.11% (Bayesian based), while the average positive false is 0.06% vs. 0.15% (BPN) and 0.19% (Bayesian based).

4. Conclusion

This paper presents a formal approach to implementing the LVQ-based intrusion detection on the VoIP infrastructure of a nation-wide softswitch. Experiment confirms that the proposed approach makes the following achievements:

a. Improve the detection accuracy and response efficiency of attacks with ambiguous patterns.
b. Reduce positive and negative false in the VoIP detection.
c. Improve the vIDS capability to detect compound attacks
d. Reduce the training time and size of the training set.

The work also evaluates the proposed approach against traditional approaches, showing that the proposed approach significantly outperforms over traditional ones. The results of experiments also confirmed that the proposed LVQ-based vIDS is a feasible intrusion detective approach which can be easily to be expanded to a commercial VoIP anti-intrusion system.

The work introduces a number of future investigations, including:

a. Exploring new observed variables;
b. Using multilayer neural networks or other types of neural networks such as recurrent neural network with genetic algorithm to detect new types of compound attacks.
c. Improving the proposed method for producing training data.

References

[1] China Telecommunication Industry forum Researching Group, “China telecommunication infrastructure Market Research Investigation”, China Telecommunication Industry Forum, Beijing, 2006.

[2] VoIP Security Association, “VoIP Security and Privacy Threat Taxonomy”, retrieved on August 2008 from www.voipsa.org/Activities/VOIPSA_Threat_Taxonomy_0.1.pdf.

[3] M. Nassar, R. State, and O. Festor, “Intrusion detection mechanisms for VoIP applications”, Third annual security workshop (VSW'06), ACM Press, June 2006.

[4] M. Gori, A. Tesi, "On the Problem of Local Minima in Backpropagation," IEEE Transactions on Pattern Analysis and Machine Intelligence, USA, Jan. 1992,vol. 14, no. 1, pp. 76-86, doi:10.1109/34.107014.

[5] C. Zhan, X. Lu, M. Hou, and X. Zhou, “A LVQ-based neural network anti-spam email approach”, ACM SIGOPS Operating Systems Review Volume 39 , Issue, USA, 2005 pp. 34-39.

[6] Willamette University, “CS-449: Neural Networks”, retrieved on August 2008 from http://www.willamette.edu/~gorr/classes/cs449/intro.html

[7] Y. Liu, and C. Bao, “A New Voice Activity Detection Algorithm Based On SOM & LVQ”, International Journal of Information Technology Vol. 12 No.6, 2006.

[8] R. Chang, L. Lai, W. Su, J. Wang, and J. Kouh, “ Intrusion detection by backpropagation neural networks with sample-query and attribute-query”, International Journal of Computational Intelligence Research, vol. 3, no. 1, 2007, pp. 6-10.

[9] C. Tory, (J.C.) Fu, C. Huang (1999), “Prototype LVQ based Computerized Tool for Accent Diagnosis among Chinese Speaker of English as A Foreign Language”, Da-Yeh, Journal of Da-Yeh University, China Taiwan, Vol.8, No,2,1999, pp.53-62,

[10] Z. Zhang, H. Chen, S. Ye, and J. Zhao, “Comparison of the BP training algorithm and LVQ neural networks for e, u, π, identification”, Nuclear Instruments and Methods in Physics Research A 379, 1996, pp. 271-275 .

[11] E.W. Saad, J.J. Choi, J.L. Vian, and D.C. Wunsch, Query-Based Learning for Aerospace Application, IEEE Trans. on Neural Networks, USA, 2003, vol. 14, no. 6, pp.1437-1448.

[12] H. Sengar, D. Wijesekera, H. Wang , and S. Jajodia, “VoIP Intrusion Detection Through Interacting Protocol State Machines”, Proceedings of the International Conference on Dependable Systems and Networks, USA, 2006, pp. 392-402.