Layered Crf A Model To Build More Accurate Intrusion Detection System

Miss Vaishali T.Deshmukh.
Lecturer, Shreeram Polytechnic,
Airoli, Navi Mumbai.400708

Miss Shubhangi Vaikole
Asst.Professor, Datta Meghe college of Engg.
Airoli, Navi Mumbai.400708

Abstract- *With the today's changing and high-tech lifestyle the use of network becomes a part of each and everyone's routine life. Most of us keep and access sensitive data online. So it becomes most important to secure your network from various network attacks. Intruder is one of the most publicized threats to network security. Network intrusion detection system (NIDS) has become a standard component in network security infrasture.But at the same time today's intrusion detection system faces number of challenges. An intrusion detection system must reliably detect malicious activities in a network and must perform efficiently to cope with the large amount of network traffic. In this paper we address two concepts one layered architecture and second conditional random fields and integrate them to improve the overall accuracy and effiency of the system.*

Index Terms-*Intrusion Detection, NIDS, Layered Approach, Conditional Random Fields, Network Security.*

I INTRODUCTION

When we have some hackers and attackers around us while working with network, we must adopt good measures to ensure security of the system. Three main aspect s of computer security are integrity, confidentiality and non reputability. No system is perfect .There will always be some errors in any system and those will be taken as advantage of by the attacker we cannot make sure that the system doesn't have any weakness. But at least we can keep a network intrusion detection system between user and the system which would filter out all the hackers and attackers from the normal users. In this paper we have demonstrated a Network Based College Administration System and used layered approach with conditional random fields to develop more efficient and accurate intrusion detection system for the same. The attacks are categorised into four groups (Probe, DOS,U2R,R2L) . All the possible attack will be shown with this application and detected at that particular layer of attack. Each layer is modelled separately with a set of features. The sequence labelling and the classification can be done with CRF. Customization of a system can be done according to the requirements.

II. VARIOUS INTRUSION DETECTION SYSTEMS (IDS)

Intrusion detection (ID) system is a type of security management system for computers and networks. An ID system gathers and analyzes information from various areas within a computer or a network to identify possible security breaches, which include both intrusions and misuse functions, does following

- Monitors and analyzes user and system activities
- Analyzes system configurations and vulnerabilities
- Assess system and file integrity
- Recognizes patterns of typical attacks
- Analyses abnormal activity patterns
- Tracks user policy violations

Intrusion detection started in around 1980s and can also be classified as

- Network intrusion detection system (NIDS)
- Host-based intrusion detection system (HIDS)
- Protocol-based IDS(PIDS)
- Application protocol-based intrusion detection system (APIDS).

Here we are working on network intrusion detection system (NIDS). A network intrusion detection system

is a system that also tries to detect malicious activity
such as denial of service attacks [3], port scans or even attempts to crack into computers by

of TCP connection requests to a very large number of different ports are observed, one could assume that there is someone conducting a port scan of some or all of the computer(s) in the network.NIDS

III. RELATED WORK

Intrusion detection and network security has been introduced around since late 1980s. Since then, many methods and frameworks have been proposed and many systems have been built to detect intrusions.

A. Association rule mining

These are based on building classifiers by discovering relevant patterns of program and user behaviour. Association rules are used to learn the record patterns that describe user behaviour. These methods can deal with symbolic data, and the features can be defined in the form of packet and connection details. However, mining of features is limited to entry level of the packet and requires the number of records to be large and sparsely populated. Otherwise, they tend to produce a large number of rules that increase the complexity of the system.

B. Data Clustering Methods k-means and the fuzzy c-means

Clustering technique [20] is based on calculating numeric distance between the observations, and hence, the observations must be numeric. Observations with symbolic features cannot be easily used for the clustering methods. It considers the features independently and is unable to capture the relationship between different features of a single record, which further degrades attack detection accuracy.

C. Naive Baye's classifiers

These make strict independence assumption between the attributes in an observation resulting in lower attack detection accuracy when the features are correlated, which is often the case for intrusion detection. Bayesian network [9] can also be used for intrusion detection. However, they tend to be attack specific and build a decision network based on special characteristics of individual system. To overcome the weakness of a single intrusion detection system, a number

monitoring network traffic [11]. The NIDS does this by reading all the incoming packets and try to find out suspicious patterns. If, for example, a large number

is not limited to inspecting incoming network traffic only. Often valuable information about an ongoing intrusion can be learned from outgoing or local traffic as well.
attacks. Thus, the size of a Bayesian network increases rapidly as the number of features and the type of attacks modelled by a Bayesian network increases [9]. To detect anomalous traces of system calls in privileged processes, hidden Markov models (HMMs) have been applied in and however, modelling the system calls alone may not always provide accurate classification as in such cases various connection level features are ignored. Further, HMMs are generative systems and fail to modelling-range dependencies between the observations.

D. Decision trees

This method selects the finest features for each decision node during the construction of the tree based on some well defined criteria [8]. One such criterion is to use the information gain ratio. Decision trees generally have very high speed of operation and high attack detection accuracy.

E. Neural Networks

According to Debar [21] though the neural networks can work effectively with noisy data, they require large amount of data for training and it is often hard to select the best possible architecture for a neural network.

F. Support Vector Machines

Support vector machines have also been used for detecting intrusions. Support vector machines map real valued input feature vector to a higher dimensional feature space through nonlinear mapping[6].This can also provide real-time detection capability, deal with large dimensionality of data, and can be used for binary-class as well as multiclass classification.

Other approaches for detecting intrusion include the use of autonomous and probabilistic agents for intrusion detection. These methods are generally aimed at developing distributed intrusion detection

of frameworks have been proposed, which describe the collaborative use of network-

based and host based systems, systems that employ both signature based and behaviour-based techniques. The data analyzed by the intrusion detection system for classification often has a number of features that are highly correlated and complex relationships exist between them. When classifying network connections as either normal or as attack, a system may consider features such as "logged in" and "number of file creations" [2]. When these features are analyzed individually, they do not provide any information that can assist in detecting attacks. However, when these features are analyzed together, they can provide meaningful information, which can be helpful for the classification task [15].

IV. CONDITIONAL RANDOM FIELDS FOR INTRUSION DETECTION

Conditional models are probabilistic systems that are used to model the conditional distribution over a set of random variables. The CRFs [19] have proven to be very successful in such tasks, as they do not make any unwarranted assumptions about the data. Hence, we explore the suitability of CRFs for intrusion detection system may consider features such as "logged in" and "number of file creations."When these features are analyzed individually, they do not provide any information that can aid in detecting attacks.
However, when these features are analyzed together, they can provide meaningful information, which can be helpful for the classification task.

Let X be the random variable over data sequence to be labelled and Y the corresponding label sequence. In addition, let *G = (V,E)* be a graph such that $Y = (Y_v)_{v \in (V)}$, so that Y is indexed by the vertices of G. Then, *(X,Y)* is a CRF, when conditioned on X, the random variables Yv obey the Markov property with respect to the graph $p(Y_v|X, Y_w, w \neq v) = p(Y_v|X, Y_w, w \sim v)$, where $w \sim v$ means

that *w* and *v* are neighbours in *G*, i.e., a CRF is a random field globally conditioned on X. For a simple sequence (or chain) modelling, as in our case, the joint distribution over the label sequence Y given X has the following form:

$$p_\theta(y|x) \propto \exp\left(\sum_{e \in E,k} \lambda_k f_k(e, y|_e, x) + \sum_{v \in V,k} \mu_k g_k(v, y|_v, x)\right),$$

Where x is the data sequence, y is a label sequence, and *y|s* is the set of components of *y* associated with the vertices or edges in subgraph *S*. In addition, the features *fk* and *gk* are assumed to be given and fixed. For example, a Boolean edge feature *fk* might be true if the observation *Xi* is "protocol = tcp," tag *Yi-1* is "normal," and tag *Yi* is "normal." Similarly, a Boolean vertex feature *gk* might be true if the observation *Xi* is "service = ftp" and tag *Yi* is "attack."

We showed that the sequence labelling methods such as the CRFs can be very effective in detecting attacks.

In the proposed system each record represents a separate connection, and hence, we consider every record as a separate sequence. We aim to model the relationships among features of individual connections using a CRF, as shown in Fig.4. In this figure, features such as duration, protocol, service, flag, and src_bytes take some possible value for every connection.

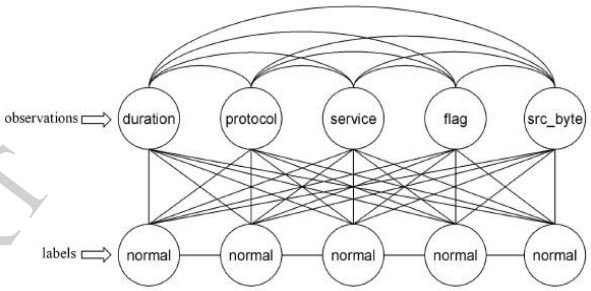

Fig. 2.Graphical representation of a CRF.

During training, feature weights are learnt, and during testing, features are evaluated for the given observation, which is then labelled accordingly.

V LAYERED ARCHITECTURE FOR INTRUSION DETECTION

The LIDS [18] draws its motivation from what we call as the Airport Security model, where a number of security checks are performed one after the other in a sequence. Similar to this model, the LIDS represents a sequential Layered Approach and is based on ensuring availability, confidentiality, and integrity of data and (or) services over a network. Fig. 3 gives a generic representation of the framework.
Most intrusions occur via network using the network protocols to attack their targets. Generally, there are four categories of attacks
They are:

(1) DoS (denial-of-service), for example, ping-of death, syn flood, etc.
(2) Probe, surveillance and probing, for example, port-scan, ping-sweep, etc.
(3) R2L, unauthorized access from a remote

machine, for example, guessing password.

(4) U2R, unauthorized access to local super user privileges by a local unprivileged user, for example, various buffer overflow

Each layer is then separately trained with a small set of relevant features. Feature selection is significant for Layered Approach and discussed in the next section. In order to make the layers independent, some features may be present in more than one layer. The layers essentially act as filters that block any anomalous connection, thereby eliminating the need of further processing at subsequent layers enabling quick response to intrusion. The effect of such a sequence of layers is that the anomalous events are identified and blocked as soon as they are detected.

The goal of using a layered model is to reduce computation and the overall time required to detect anomalous events. The time required to detect an intrusive event is significant and can be reduced by eliminating the communication overhead among different layers. This can be achieved by making the layers autonomous and self-sufficient to block an attack without the need of a central decision-maker. Every layer in the LIDS framework is trained separately and then deployed sequentially. We define four layers that correspond to the four attack groups mentioned in the data set.

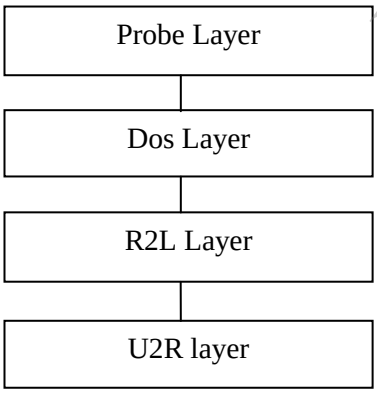

Fig.2 Layered Architecture for Intrusion Detection System

VI FEATURE SELECTION FOR EACH LAYER

In our system, every layer is separately trained to detect a single type of attack category. The attack groups are different in their impact, and hence, it becomes necessary to treat them differently.Hence,we select features for each layer based upon the type of attacks that the layer is trained to detect. Features which are selected are given in table A1.

Algorithm

Training

Step 1: Select the number of layers, n, for the complete system.

Step2: Separately perform features selection for each layer.

Step 3: Train a separate model with CRFs for each layer using the features selected from Step2.

Step 4: Plug in the trained models sequentially such that only the connections labelled as normal are passed to the next layer.

Testing

Step 5: For each (next) test instance perform Steps 6 through 9.

Step 6: Test the instance and label it either as attack or normal.

Step 7: If the instance is labelled as attack, block it and identify it as an attack represented by the layer name at which it is detected and go to Step 5. Else pass the sequence to the next layer.

Step 8: If the current layer is not the last layer in the system, test the instance and go to Step7.Else goto Step 9.

Step 9: Test the instance and label it either as normal or as an attack. If the instance is labelled as an attack, block it and identify it as an attack corresponding to the layer name .

VII IMPLEMENTATION AND RESULTS

We have developed a simple college management system to automate various operations performed in a college. The various modules in the system are given below

1. Login Module
2. Admin Module
3. Office Module
4. Admission Module
5. Exam Module
6. Library Management Module
7. Inventory Module

Once the system is ready, we can easily find its loopholes and demonstrate the various attacks which can be performed on this system.

7.1 DEMONSTRATING ATTACKS ON THE SYSTEM

On this system, we have demonstrated the following attacks

A. DENIAL OF SERVICE ATTACK

2.2250738585072012e-308
0.022250738585072012e-00306

Add this number to any field which accepts double data (Inventory -> Add dead stock -> price)
The server will run into an infinite loop and thus we can have denial of service attack.
Add any entry more than 3 times. If a single user tries to get the same service from the server or tries to enter the same data for more than 3 times, it is just to keep the server busy so that it is not able to service other users.

B. SQL INJECTION

For login, use any user id name
Eg . admin
for password use
a' or 'a'='a
This same password will be able to help any hacker log into any of the various user accounts.

C. BUFFER OVERFLOW

For each question paper, any paper whose size is more than 1 Mb can easily deplete the memory of the server used to store the question papers. There would be hundreds of question paper on the server. There would be some storage reserved for question papers. Once that limit is reached, we would have buffer overflow.

D. PROBE ATTACKS

Probe attack is any attack in which a hacker tries to access any service from the server without actually getting authenticated. There are many methods to perform this type of attack, the simplest being trying to retrieve the .class files of the application. Once the hacker gets access to the .class files, he can try to run them. There are many files which have a main method and will run even when the user is not authenticated and no login operation is performed. Here we can access any feature of the system including the service to "Download Question Paper" which is used to download question paper from server to print it before the exam.

7.2 IMPLEMENTING CONDITIONAL RANDOM FIELDS AND LAYERED APPROACH TO DETECT AND BLOCK ATTACKS

Here we use the conditional random field approach to generate rules. We generate separate rules for each of the layers. The entire network intrusion detection system is divided into 4 layers namely – Probe layer,Dos layer,U2R layer and R2L layer. Each layer is implemented one after the other. If any attack is detected in one layer, the layers below it are skipped and the service is immediately blocked. Only when the request goes through the 4 layers and is unblocked, then the request is serviced. Each layer has 2 main parts – the rules and its implementation. Once the rules are found, the various attributes of the current request is detected and then they are checked with respect to the rules. Following are given the details of what types of attacks are detected by each layer and some sample rules for each layer.

A. PROBE LAYER

This layer will have rules relating to block all the attacks in which the user tries to access any service without actually being logged in. We develop rules in such a way that the any user in no way would be able to access any service without being logged in. For this we may use many attributes of any request given below
Username – In each request, the system internally forwards the username of the current user for keeping a log of each service provided by the server. We can check for the correctness of this information. We can check whether any user requesting any information is currently logged in or not. If that user is not logged in, it must be the hacker and must be blocked.
IP Address of the user – At the NIDS level, we can identify what is the ip address of the requesting client machine. We can verify whether it is the same machine in which the current user is logged in.

B. DENIAL OF SERVICE LAYER

This layer will have rules to block all the attacks which can keep the server busy for long enough that many valid and authenticated users wont be able to get the services. There may be 2 types of attacks and both have to be handled differently
Server taking long time to reply – if there is only a single request which the server takes more than normal time to reply, it means that server is busy replying this request and therefore would give less priority to entertain other requests.
Same data and service requested many times – This is a type of "PING OF DEATH" type of attack where the server is busy performing the same service again and again and cannot service other request. Here we specify a limit (3) for same type of requests. Any similar type of requests if occurred thrice, the DOS layer will block such a request.

C.U2R LAYER (USER TO ROOT)

This layer deals with all types of attacks which can cause buffer overflows and similar attacks. We limit the size of data sent from client to server and so if the size is more we wont let the request reach the server and block it. Eg. We reserve around 100 mb to store question papers and we know that there will be around 100 to 150 papers. So we will block any question paper whose size is more than 1 mb.

D.R2L LAYER (REMOTE TO LOCAL)

This layer will deal with all the SQL injection attacks. Such types of attacks are most difficult to detect as there may be various places where SQL injection may be performed. But still we can attempt to detect it using the basic characteristics of

SQL injection. Like since SQL injection is a trial and error method, before a successful attack the hacker may have tried several unsuccessful attempts. Tracking such unsuccessful attempts can help us detect the SQL injection attacks. We can also try to detect some of the patterns like "or 1=1" or "or '?'='?'". Here? can be any character. Such patterns can help us detect SQL Injection.

LAYER NAME	FEATURES SELECTED
Probe Layer	• duration of connection • source bytes
DoS Layer	• percentage of connections having same destination host and same service source bytes • percentage of packets with errors
R2L Layer	• duration of connection • service requested • number of failed login attempts
U2R Layer	• number of file creations • number of shell prompts invoked

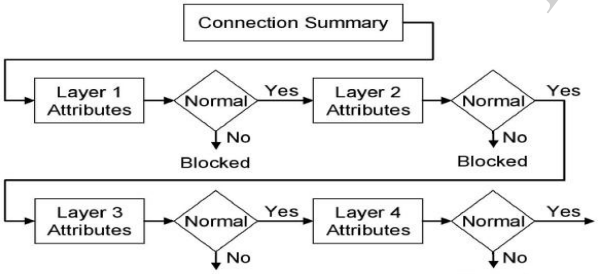

Fig.3 Real-Time Representation of the system.

VIII CONCLUSIONS

In this system we addressed the dual problem of Accuracy and Efficiency for building robust and efficient intrusion detection systems. CRFs are very effective in improving the attack detection rate and decreasing the FAR. Having a low FAR is very important for any intrusion detection system. Further; feature selection and implementing the Layered Approach significantly reduce the time required to train and test the model.This method is much suitable for detecting R2L and U2R attacks.

Our system can help in identifying an attack once it is detected at a particular layer,which expedites the intrusion response mechanism thus minimizing the impact of an attack. Our system has the advantage that the number of layers can be increased or decreased depending upon the environment in which the system is deployed, giving flexibility to the network administrator.

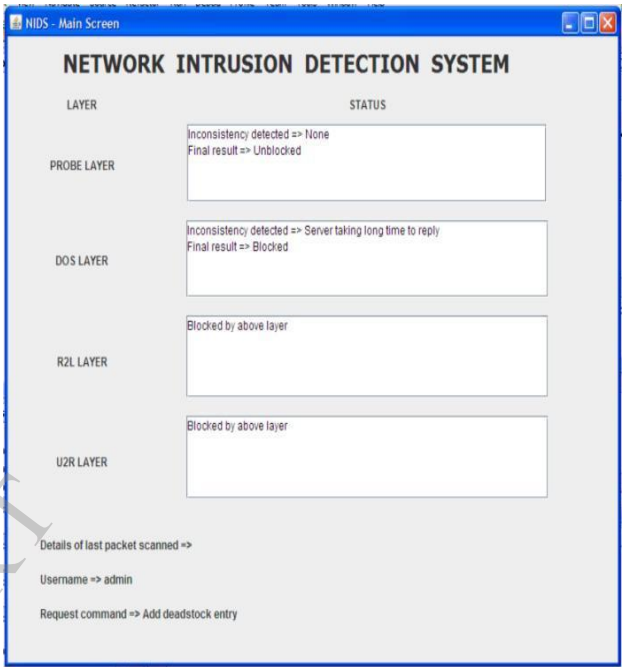

Fig.4 Sample Result of DOS Attack

REFERENCES

[1] W. Lee, S. Stolfo, and K. Mok, "Mining Audit Data to Build Intrusion Detection Models," Proc. Fourth Int'l Conf. KnowledgeDiscovery and Data Mining (KDD '98), pp. 66-72, 1998.

[2] Boswell, Wendy (n.d.). *A Short History of the Internet*.http://websearch.about.com/od/whatisthein ternet/a/historyinternet.htm.

[3] Chen, Qiang (2001). *Computer Intrusion Detection through NoiseCancellation*. Arizona: Arizona State University.

[4] Cisco (2007). *Understanding Delay in Packet VoiceNetworks*.http://www.cisco.com/warp/public/ 788/voip/delay-details.hml

[5] A. McCallum, "Efficiently Inducing Features of Conditional Random Fields," Proc. 19th Ann. Conf. Uncertainty in Artificia lIntelligence (UAI '03), pp. 403-410, 2003.

[6] D.S. Kim and J.S. Park, "Network-Based Intrusion Detection with Support Vector Machines," Proc. Information Networking, Networking Technologies for Enhanced Internet Services Int'l Conf. (ICOIN '03), pp. 747-756, 2003

[7] Inline (2005). *Snort Inline Part II*. Pete Savage. Retrieved on March 8,2007 http://linuxgazette.net/118/savage.html.

[8] A. McCallum, D. Freitag, and F. Pereira, "Maximum Entropy Markov Models for Information Extraction and Segmentation, "Proc. 17th Int'l Conf. Machine Learning (ICML '00), pp. 591-598,2000.

[9] N.B. Amor, S. Benferhat, and Z. Elouedi, "Naive Bayes vs. Decision Trees in Intrusion Detection Systems," Proc. ACM Symp. Applied Computing (SAC '04), pp. 420-424, 2004.

[10] Net Optics (n.d.). *White Paper: Deploying Network Taps with Intrusion Detection Systems*. Retrieved on April 4, 2007http://www.netoptics.com/products/pdf/Taps and-IDSs.pdf.

[11] Proctor, Paul E. (2001). *The Practical Intrusion Detection Handbook*. New Jersey: Prentice Hall.

[17]C. Ji and S. Ma, "Combinations of Weak Classifiers," IEEE Trans .Neural Networks, vol. 8, no. 1, pp. 32-42, 1997.

[18]K.K. Gupta, B. Nath, and R. Kotagiri, "Network Security Frame- work," Int'l J. Computer Science and Network Security, vol. 6, no. 7B,pp. 151-157, 2006.

[19]K.K. Gupta, B. Nath, and R. Kotagiri, "Conditional Random Fields for Intrusion Detection," Proc. 21st Int'l Conf. Advanced Information Networking and Applications Workshops (AINAW '07), pp. 203-208,

[12] Puketza, Nicholas Joseph (2000). *Approaches to Computer Security: Filtering, Testing, and Detection*. California: University of California.

[13] Savage, Pete (2005). *Snort Inline Part I*.Retrieved on March 8,2007http://linuxgazette.net/117/savage.html.

[14] Snort (2007). *Snort Users Manual*. Retrieved March 8, 2007 from http://www.snort.org./docs/snort_htmanuals/htman ual_261.

[15] Zamboni, Diego (2001). *Using Internal Sensors for Computer Intrusion Detection*. Purdue: Purdue University

[16]K.K. Gupta, B. Nath, R. Kotagiri, and A. Kazi,"AttackingConfidentiality: An Agent Based Approach," Proc. IEEE Int'l ConfIntelligence and Security Informatics (ISI '06), vol. 3975, pp. 285-296,2006.

[20]L. Portnoy, E. Eskin, and S. Stolfo, "Intrusion Detection with Unlabeled Data Using Clustering," Proc. ACM Workshop Data Mining Applied to Security (DMSA), 2001.

[21]H. Debar, M. Becke and D. Siboni, "A Neural Network Component for an Intrusion Detection System," Proc. IEEE Symp.Research in Security and Privacy (RSP '92), pp. 240-250, 1992.