Intrusion Detection System with Data Stream Modeling using Conditional Privileges

Ravindra Bhat
Student, M. Tech (CSE),
MVJ College of Engineering, Bangalore
ravindrabhat.cse@gmail.com

Abstract- **IDS for computer network is capable of detecting and alerting the systems administrator on potential intrusion, providing guidance against any potential loss of integrity and confidentiality to the enterprise's valuable intellectual assets. In this paper, the layered model for IDS and alert aggregation technique is used. In this layered IDS architecture, each layer assesses, filters, and/or aggregates information produced by a lower layer. Thus, relevant information gets more and more condensed and certain, and, therefore, also more valuable. Alert may originate from low-level IDS such as those mentioned above, from firewalls (FW), etc. Alerts that belong to one attack instance must be clustered together and meta-alerts must be generated. The main goal is to improve performance by reducing the amount of alerts substantially without losing any important information which is necessary to identify on-going attack instances.**

Keywords: Intrusion detection, Network Security, Intrusion Detection systems (IDS), Alerts.

I. INTRODUCTION

Intrusion Detection systems (IDS) are becoming more and more widely deployed to supplement the security provided by firewalls. Firewalls and authentication are effective in protecting and preventing unauthorized access up to certain level but lacks capabilities to monitor the network where majority of attacks are taking place, if an attacker able to breach the firewall, he can roam freely through the whole network. IDSs function in the digital world much the same way as a burglar alarm does in the physical world. Like all alarms, IDSs have weak points and flaws that can be exploited by an attacker to get around the system. These attacks could be initiated by disgruntled employees and others who have legitimate network access and use that privilege to do harm. Firewall and authentication systems are vital, but they work at the point of entry to the network.

Most IDS are quite reliable in detecting suspicious actions by evaluating TCP/IP connections or log files, for instance. Once an IDS finds a suspicious action, it immediately creates an alert which contains information about the source, target, and estimated type of the attack (e.g., SQL injection, buffer overflow, or denial of service). As the intrusive actions caused by a single attack instance which is the occurrence of an attack of a particular type that has been launched by a specific attacker at a certain point in time are often spread over many network connections or log file entries, a single attack instance often results in hundreds or even thousands of alerts. In addition, even low rates of false alerts could easily result in a high total number of false alerts if thousands of network packets or log file entries are inspected. As a consequence, the IDS create many alerts at a low level of abstraction. It is extremely difficult for a human security expert to inspect this flood of alerts, and decisions that follow from single alerts might be wrong with a relatively high probability. To keep a constant eye on network traffic and to know anything out of ordinary is happening, network security should be supplemented with Intrusion Detection Systems (IDS).

II. TERMINOLOGIES

Alert: It is an indication that s system has attacked or is under attack.
False Positive: Occurrence of an alert in in the absence of actual attack.

III. THE ARCHITECTURE

Most existing IDS are optimized to detect attacks with high accuracy. However, they still have various disadvantages that have been outlined in a number of publications and a lot of work has been done to analyze IDS in order to direct future research. Besides others, one drawback is the large amount of alerts produced. Alerts can be given only in System logs. Existing IDS does not have general framework which cannot be

customized by adding domain specific knowledge as per the specific requirements of the users or network administrators.

A. Layered Model

This method uses probabilistic modelling methods. Assuming that attack instances can be regarded as random processes producing alerts, aiming at modelling these processes using approximative maximum likelihood parameter estimation techniques. Thus, the beginning as well as the completion of attack instances can be detected.
It is a data stream approach, i.e., each observed alert is processed only a few times. Thus, it can be applied online and under harsh timing constraints.Data stream modelling does not degrade system performance as individual layers are independent and are trained with only a small number of features, thereby, resulting in an efficient system [1].
This model is easily customizable and the number of layers can be adjusted depending upon the requirements of the target network. The framework is not restrictive in using a single method to detect attacks. Different methods can be seamlessly integrated in framework to build effective intrusion detectors.
The framework has the advantage that the type of attack can be inferred directly from the layer at which it is detected. As a result, specific intrusion response mechanisms can be activated for different attacks.
Alerts can be sent to handheld mobile devices. This makes the process easier and comfortable.

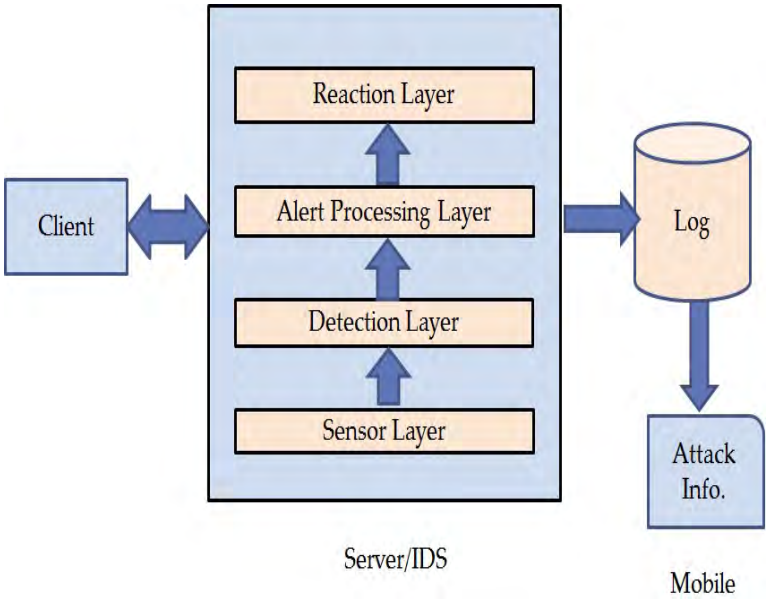


Figure 1: Intrusion Detection System

Sensor layer: Provides the interface to the network and the host on which the agent resides. Sensors acquire raw data from both the network and the host, filter incoming data, and extract interesting and potentially valuable (e.g., statistical) information which is needed to construct an appropriate event.

Detection layer: It has different detectors, e.g., classifiers trained with machine learning techniques such as support vector machines (SVM) or conventional rule-based systems such as Snort [3] , assess these events and search for known attack signatures (misuse detection) and suspicious behaviour (anomaly detection) [2]. In case of attack suspicion, they create alerts which are then forwarded to the alert processing layer.

Alert processing layer: The alert aggregation module combine alerts that are assumed to belong to a specific attack instance. Thus, so called meta-alerts are generated.

Reaction layer: An important task of the reaction layer is intrusion prevention.

IV. SYSTEM COMPONENTS

A. Server:

Server acts as the Intrusion Detection System. This module consists of four layers viz. sensor layer (which detects the user/client etc.), Detection layer, alert processing layer and reaction layer.
In addition there is also Message Log, where all the alerts and messages are stored for the references. This Message Log can also be saved as Log file for future references for any network environment.

B. Client **:**

Used for testing the Intrusion Detection System.

Conditional Privileges

Client can enter only with a valid user name and password. If an intruder enters with any guessing passwords then the alert is given to the Server and the intruder is also blocked. Even if the valid user enters the correct user name and password, the user can use only for minimum number of times. For example even if the valid user makes the login for repeated number of times, the client will be blocked and the alert is sent to the admin.

In the process level intrusion, each client would have given a specific process only. For example, a client may have given permission only for P1 process. If the client tries to make more than these processes the client will be blocked and the alert is given by the Intrusion Detection System.

The client can also be able to send data. Here, whenever data is sent Intrusion Detection System checks for the file. If the size of the file is large then it is restricted or else the data is sent.

C. DARPA Dataset:

This is an offline type of testing the intrusions. The DARPA Data Set is used to check the technique of the Intrusion Alert Aggregation with Generative Data Stream Modeling [4].

The DARPA data set is downloaded and separated according to each layers. So test the instance of DARPA Dataset using the open file dialog box. Whenever the dataset is chosen based on the conditions specified the Intrusion Detection System works.

D. Mobile:

The traditional system uses the message log for storing the alerts. In this system, the system admin or user can get the alerts in their mobile. Whenever alert message received in the message log of the server, the mobile too receives the alert message.

E. Attack Simulation:

The attack simulation is made to test the system. Attacks are classified and made to simulate here. Whenever an attack is launched the Intrusion Detection System must be capable of detecting it. For example if an IP trace attack is launched, the Intrusion Detection System must detect it and must kill or block the process.

V. EXPEREMENTAL RESULTS

This section evaluates the new intrusion detection approach. Three different methods are used to demonstrate the feasibility of the proposed architecture. The first method is conditional privileges, for the second well-known DARPA intrusion detection evaluation data set [5] and the third contains simulation tests. All the experiments were conducted on an PC with 2.20 GHz and 2 GB of RAM.

A. Real time Attacks :

User level privileges : The system is tested with various set of username and passwords. For each pair of valid username and password the system allowed user to access the system. Also for wrong username and password the intrusion detection system detected as intruder and blocked the user.

Process level privileges: All legitimate user allowed to access the system and to execute certain process say P1.If user tried to access process other than P1; say P2,P3 etc, the IDS detected it as process level intruder and blocked the process. Also proper alert is generated at different layers.

Packet Level privileges: Users are allowed to transfer data with certain limitations. System is tested with various size packets. For each transfer which is within certain size limitation is allowed. The file transfer exceeding the limitation is aborted and proper alert is generated.

B. DARPA Dataset Test:

The system is designed to detect mainly two types of attacks which are DOS and probe attacks. This system detected attacks properly for all the DARPA dataset inputs and proper reaction is generated by intrusion detection system.

C. Attack Simulation:

Simulation attacks are categorized as Information gathering(port scanning , sniffing), Flooding (DOS, Buffer overflow), Authentication bypass (resource exhaustion, password attack), Malware (viruses, worms, Trojan horses) attack. For each type of attack IDS system responded properly and alerts are generated.

D. Alert Aggregation:

Alerts are generated at various levels of intrusion detection system. These alerts are aggregated and stored in a database. These alerts can be easily accessed by the system admin.

Log files provide useful information about attacks and the reactions by the intrusion detection system. These are stored for future reference.

For all the above experiments intrusion detection system responded properly and alerts are generated.

Table 1: Benchmark Results

Operating Point	p[%]	Tavg[ms]	reduction%	d[s]
DARPA DATA				
Idealized	100	0.13	99.99	1
OP1	100	0.19	99	6
OP2	66	0.28	99	7
Real-Time attacks				
Idealized	100	0.13	99.99	1
OP1	75	0.20	99	2
OP2	60	0.30	80	2
Attacks Simulation				
Idealized	100	0.13	99.99	1
OP1	100	0.13	80	2
OP2	100	0.20	60	5

E. Description of the Benchmark Data Sets

Percentage of detected instances (p): The percentage of detected attack instances p can thus be determined by dividing the number of instances that are detected by the total number of instances in the data set. The measure is computed with respect to the instances covered by the output of the detection layer, i.e., instances missed by the detectors are not considered.

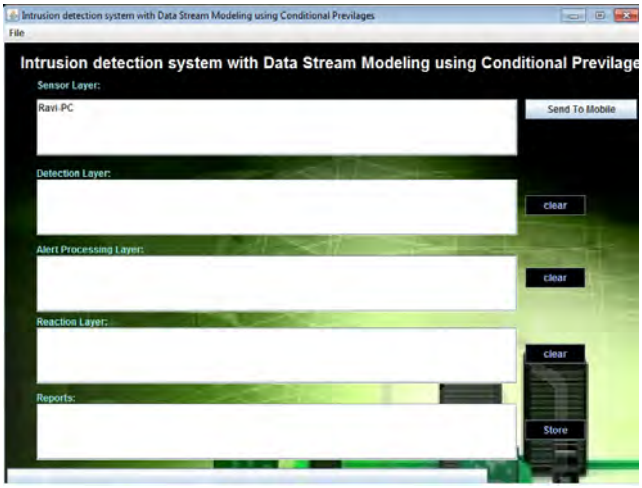
Average runtime(Tavg): The average runtime is measured in milliseconds per alert. Assuming up to several hundred thousand alerts a day, Tavg should stay clearly below 100 ms per alert.

Alert creation delay(d): It is obvious that there is a certain delay until a meta-alert is created for a new attack instance. The meta-alert creation delay d measures the delay between the actual beginning of the instance (i.e., the creation time of the first alert) and the creation of the first meta-alert for that instance.

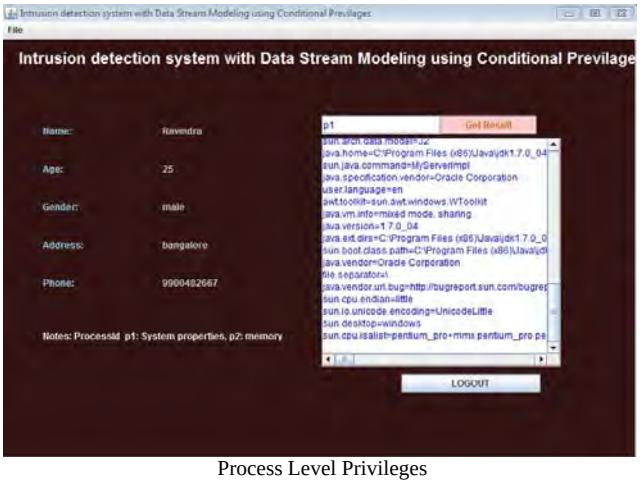
F. Performance:

The network performance can be determined by few terms such as the Admin alerts busy time, File utilization level, efficiency, fairness and imbalance. The amount of the time the User allocated for communication with the admin server to interact with the Receivers. Similarly channel is sometimes being idle during communication. The unit of time which makes delay to transmit a packet is called channel access delay time. The channel or medium utilization level can be defined as average rate of reliable packets delivered through the channel. The different layers utilization level can be determined by noticing whether the medium is busy or idle. The binary values are used for indicating the different layers utilization level.

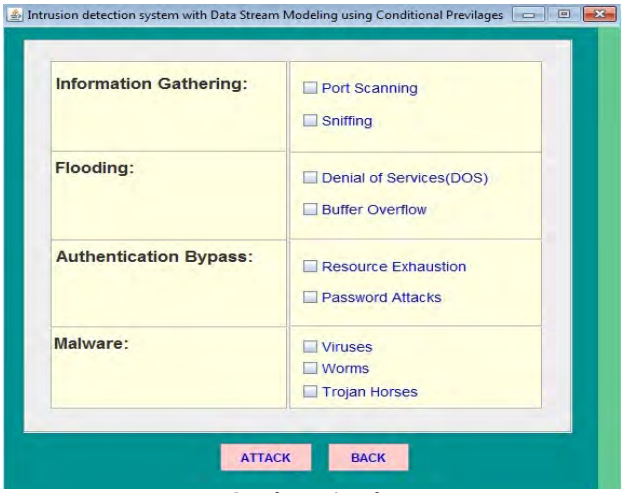
SCREENSHOTS



IDS Server

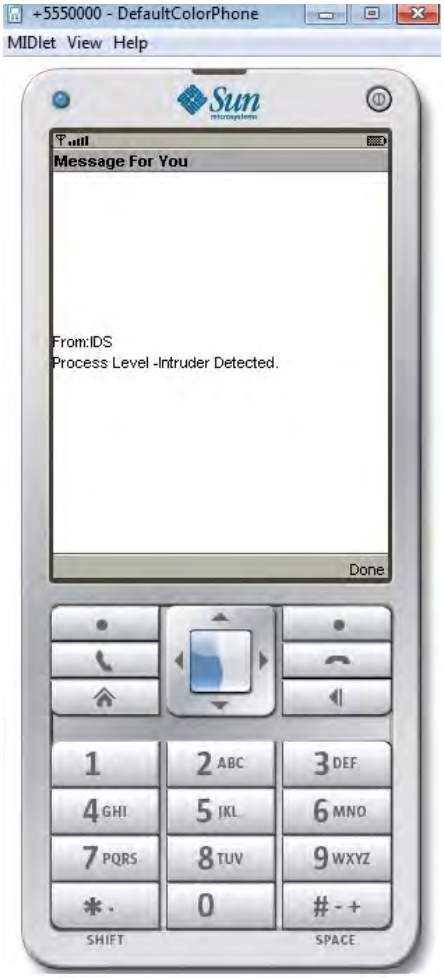


Process Level Privileges



Simulation Attacks

Alert aggregation



Mobile Alerts

VI. CONCLUSION

An intrusion detection system is a part of the defensive operations that complements the defenses such as firewalls etc. The intrusion detection system basically detects attack signs and then alerts. The presented IDS is a layered model for IDS with alert aggregation. With this model the sheer amount of data that must be reported to a human security expert or communicated within a distributed intrusion detection system, for instance, can be reduced significantly. At the same time, the number of missing attack instances is extremely low and the delay for the detection of attack instances is within the range of some seconds only. In terms of performance, customizable layered model allows efficient deployment in various heterogeneous environments. Also an intrusion detection system becomes more accurate as it detects more attacks and raises fewer false positive alarms.

REFERENCES

[1] Alexander Hofmann and Bernhard Sick, "Online Intrusion Alert Aggregation with Generative Data Stream Modeling", IEEE Transactions on Dependable and Secure Computing, Vol. 8, No. 2, March – April 2011.
[2] G. Giacinto, R. Perdisci, and F. Roli, "Alarm Clustering for Intrusion Detection Systems in Computer Networks," Machine Learning and Data Mining in Pattern Recognition, P. Perner and A. Imiya, eds. pp. 184-193, Springer, 2005.
[3] M. Roesch, "Snort—Lightweight Intrusion Detection for Networks," Administration (LISA '99), pp. 229-238, 1999.
[4] M.R. Henzinger, P. Raghavan, and S. Rajagopalan, "Computing on Data Streams". Am. Math. Soc., 1999.
[5] R.P. Lippmann, D.J. Fried, I. Graf, J.W. Haines, K.R. Kendall, D.McClung, D. Weber, S.E. Webster, D. Wyschogrod, R.K.Cunningham, and M.A. Zissman, "Evaluating Intrusion Detection Systems: The 1998 DARPA Offline Intrusion Detection Evaluation," Proc. DARPA Information Survivability Conf. and Exposition (DISCEX),vol. 2, pp. 12-26, 2000