Intrusion Alert Aggregation System in Distributed Networks

V.Aruna Devi
Department of Computer Science
Pondicherry University
Karaikal

Prashant Yadav
Department of Computer Science
Pondicherry University
Karaikal

S.Bhuvaneswari
Head
Department of Computer Science
Pondicherry University
Karaikal

ABSTRACT
A novel technique is proposed to aggregate the alerts produced when an intruder comes into an existence in distributed network. This becomes an essential task to cluster different types of alerts. Meta-alerts are generated from the clusters formed with all the relevant details of the attack in detail. This Alert aggregation technique is developed as a dynamic, probabilistic model of the existing or prevailed attacks that has been created so far. To cluster the alerts, the sensitive parameters are found and generative data stream modelling version is utilized. In addition, meta-alerts are generated with a delay of typically only a few seconds after observing the first alert belonging to a new attack instance
.

Keywords
Intrusion detection system, alert aggregation, generative modelling, data stream algorithm, meta - alert, DARPA Dataset

1. INTRODUCTION
Intrusion detection systems (IDS) are besides other protective measures such as virtual private networks, authentication mechanisms, or encryption techniques very important to guarantee information security. They help to defend against the various threats to which networks and hosts are exposed to by detecting the actions of attackers or attack tools in a network or host-based manner with misuse or anomaly detection techniques .

At present, 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. IDS usually focus on detecting attack types, but not on distinguishing between different attack instances. 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 creates 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.

In our opinion, a "perfect" IDS should be situation-aware in the sense that at any point in time it should "know" what is going on in its environment regarding attack instances (of various types) and attackers. In this paper, we make an important step toward this goal by introducing and evaluating a new technique for alert aggregation. Alerts 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 for these clusters. The main goal is to reduce the amount of alerts substantially without losing any important information which is necessary to identify ongoing attack instances. Meta IDS is the solution for enterprise-wide security deployments .We want to have no missing meta alerts, but in turn we accept false or redundant meta-alerts to a certain degree.

This problem is not new, but current solutions are typically based on a quite simple sorting of alerts, e.g., according to their source, destination, and attack type. Under real conditions such as the presence of classification errors of the low-level IDS (e.g., false alerts), uncertainty with respect to the source of the attack due to spoofed IP addresses, or wrongly adjusted time windows, for instance, such an approach fails quite often.

2. SYSTEM ANALYSIS

Many authors have proposed methods on many Intrusion Alerts. As our contribution, we make the system more efficient in identify the intrusion alerts and also we extend this work by sending the Alerts as Message to the Network Administrator who governs the Network or Intrusion Detection System.

2.1. EXISTING SYSTEM

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.
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.

2.2. PROPOSED SYSTEM

Online Intrusion Alert Aggregation with Generative Data Stream Modeling is a generative modeling approach using probabilistic methods. Assuming that attack instances can be

regarded as random processes "producing" alerts, we aim at modeling these processes using approximate 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.

In the proposed scheme of Online Intrusion Alert Aggregation with Generative Data Stream Modeling, we extend our idea of sending Intrusion alerts to the mobile. This makes the process easier and comfortable.

Online Intrusion Alert Aggregation with Generative Data Stream Modeling 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.

Online Intrusion Alert Aggregation with Generative Data Stream Modeling is easily customizable and the number of layers can be adjusted depending upon the requirements of the target network. Our framework is not restrictive in using a single method to detect attacks. Different methods can be seamlessly integrated in our framework to build effective intrusion detectors.

Our 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.

3. MODULES DESCRIPTION

1. Server
2. Client
3. DARPA DataSet
4. Mobile
5. Attack Simulation

3.1. SERVER

Server module is the main module for this project. This module 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.

3.2. CLIENT

Client module is developed for testing the Intrusion Detection System. In this module the 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. 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 P1process. If the client tries to make more then these processes the client will be blocked and the alert is given by the Intrusion Detection System. In this client module the client can be able to send data. Here, when ever 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.

3.3. DARPA Dataset

This module is integrated in the Server module. This is an offline type of testing the intrusions. In this module, the DARPA Data Set is used to check the technique of the Online Intrusion Alert Aggregation with Generative Data Stream Modeling. The DARPA data set is downloaded and separated according to each layers. So we 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.

3.4. MOBILE

This module is developed using J2ME. 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.

3.5. ATTACK SIMULATION

In this module, the attack simulation is made for ourself 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. So our system will also be capable of detecting such attacks. For example if an IP trace attack is launched, the Intrusion Detection System must detect it and must kill or block the process.

.

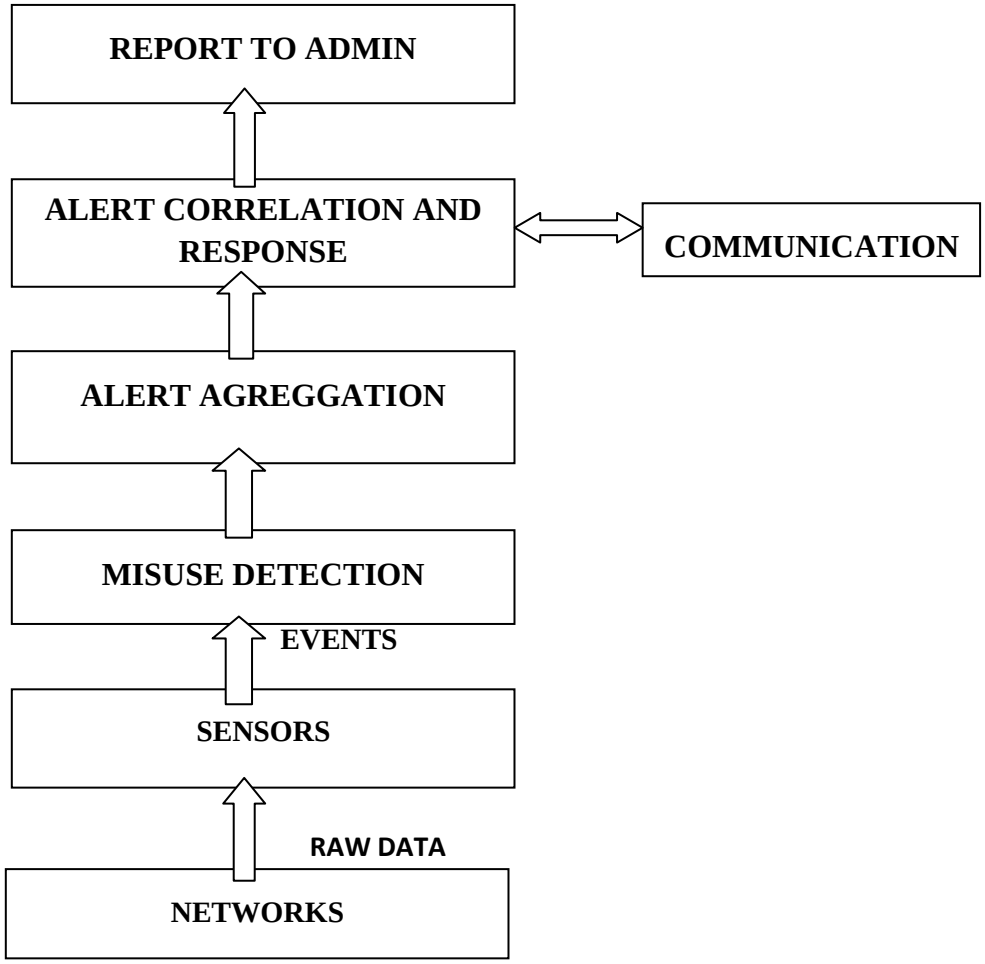


Fig 1: architecture of an intrusion detection agent

SCREEN SHOTS:

FIG 1

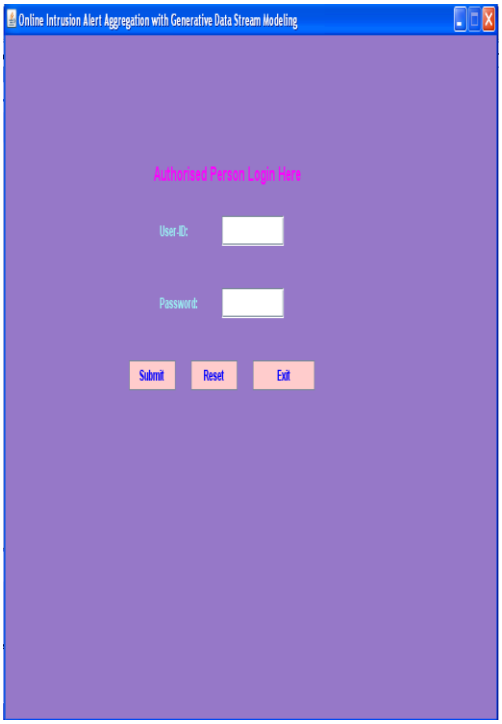


FIG 2

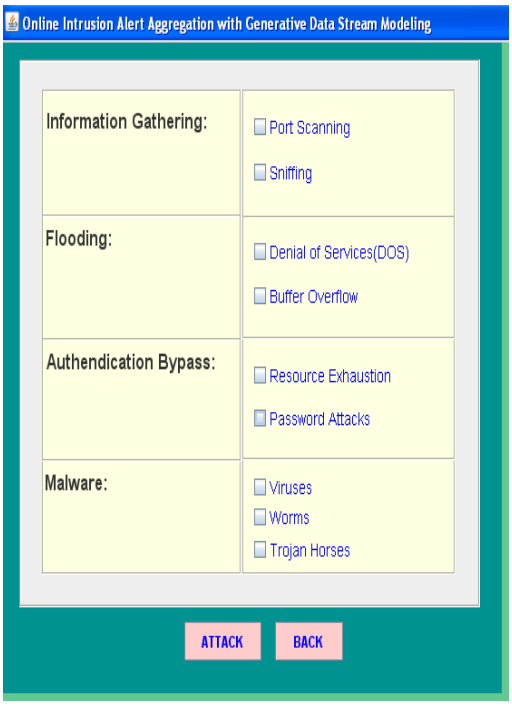
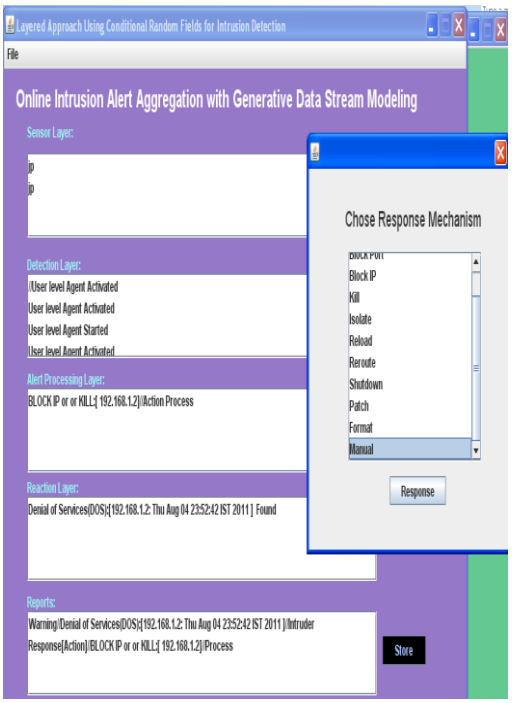


FIG 3



CONCLUSION

In this paper, we proposed a novel technique for alert aggregation produced when an intruder is detected. We also addressed the problem of accuracy and efficiency of Intrusion Detection System. We developed the presented architecture and tested the system with the misuse based anomaly detection technique. We also proposed a misuse based anomaly detection algorithm for our system. As our contribution, we make the system more efficient in identify the intrusion alerts and also we extend this work by sending the Alerts as Message to the Network Administrator who

governs the Network or Intrusion Detection System. Most of the present existing Intrusion Detection System does not have a generalized framework. Our proposed architecture is similar to layers, so according to the network environment, the network administrator can add or remove the layers. If a new updated version of detection comes in future, then it will be very easy to add the layer with our proposed system. We also tested our system by launching various attacks to the system, and we found how the system detects and reacts according to the developed IDS. As a future work, this work can be extended as not only to detect attacks and also to prevent attacks. As mentioned earlier, our proposed system allows adding new layers, the prevention layer functionality layer can also be added with our system, as a future work.

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] Kapil Kumar Gupta, Baikunth Nath and Ramamohanarao Kotagiri, "Layered Approach using Conditional Random Fields for Intrusion Detection", IEEE Transactions on Dependable and Secure Computing, Vol.7, No.1, January-March 2010.

[3] S. Axelsson, "Intrusion Detection Systems: A Survey and Taxonomy," Technical Report 99-15, Dept. of Computer Eng., Chalmers Univ. of Technology, 2000.

[4] Allen, "Intrusion Detection Systems: Perspective," Technical Report DPRO-95367, Gartner, Inc., 2003.

[5] Measurement, M.R. Endsley and D.J. Garland, eds., chapter 1, pp. 3-32, Lawrence Erlbaum Assoc., 2000.

[6] C.M. Bishop, Pattern Recognition and Machine Learning. Springer, 2006.

[7] M.R. Henzinger, P. Raghavan, and S. Rajagopalan, Computing on Data Streams. Am. Math. Soc., 1999.

[8] F. Valeur, G. Vigna, C. Krugel, and R.A. Kemmerer, "A Comprehensive Approach to Intrusion Detection Alert Correlation," IEEE Trans. Dependable and Secure Computing, vol. 1, no. 3, pp. 146-169, July-Sept. 2004.