ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 8, August 2015

Real Time Intrusion Alert Aggregation for Distributed Intrusion Detection System and Prevention of Attacks

Madhuri Chavan[1], **Manjusha Deshmukh**[2]

Dept. of Computer Engineering, Pillai Hoc College of Engineering and Technology, Rasayani, India[1]

Dept. of Computer Engineering, Pillai Institute of Information Technology, New Panvel, India[2]

Abstract: Real time intrusion alert aggregation is a concept of aggregating intrusion alerts which are generated in real time environment. The System present a real time intrusion alert aggregation strategy for distributed system. Organization use different intrusion detection System to detects unauthorized activities and different attack performed by intruder but a lot of alert is generated. Security personnel are confused with bulk of alerts. This makes them difficult to take decision immediately. They take a lot of time to analyse the alerts. The proposed system generates meta-alerts on real time alert data. IDS system cluster similar alerts and form one Meta alert and that Meta alert is send to admin by email. System consists of multiple IDS Server (IDS) and multiple Client Systems over a network. All IDS communicates with each other and central server. IDS stores alerts in central IDS Server. Central IDS server analysis all alerts which are stored in database and shows different types of attack instance occurred on system and generates Reports.

Keywords: IDS, Intrusion, Alert, Real Time Alert Aggregation, Attack, DIDS.

I. INTRODUCTION

The widespread rise of computer network has resulted in increase of attacks on information system. These attacks are used for illegally gaining access to unauthorized information system, misuse of information or to reduce the availability of the information to authorized user. Thus there is need of complete protection of organizational resources, which is driving the attention of people towards intrusion prevention and detection system. Intrusion detection system is a piece of software that monitors computer system to detect intrusion and alerts a designated authority. An intrusion is defined as an "any set of action that attempts to compromise the integrity, availability and confidentiality of computer resource"[1]. Intrusion detection system (IDS) is a network security technology for protecting computers from attacks. However it produces great amount of ineffective alerts that overwhelm the security operator. Most of these alerts are redundant [5]. When a lot of alerts are generated by IDS in order to secure network and system, it is not easy task to interpret each and every alert and come to conclusion about the risk factor and protection measure [4]. Moreover Administrator person may take wrong decisions due number of false positive alerts and their inability to correctly understand the large amount of alerts raised by the system. Alert aggregation and alert clustering process has developed to reduce alerts data. The purpose of alert aggregation is to reduce the redundancy of alerts by grouping the duplicate and all similar alerts together and making them single one [6]. Without losing important and single alerts, aggregation of the alerts is to be done carefully by IDS System. Previously alerts aggregation applies on alerts which are stored in database. This work is offline and reduces efficiency of system. New approach is developed for online intrusion alert aggregation, which aggregate the alerts in real time [17].

II. LITERATURE SURVEY

Paper [1] presents a scientific classification of intrusion detection frameworks that is then used to study and group various exploration models. The scientific classification comprises of an arrangement first of the recognition standard, and second of certain operational parts of the intrusion detection framework as being what is indicated. In paper [2] analyst observe existing IDS and established different issues of IDS. Major issue they distinguished is that IDS delivers vast number of security alerts that can make the job of security person hard to take decision rapidly because of the confounding and incompatible alerts out of the flow of them. All IDS are having the procurement of delivering security alerts as and when needed. Numerous methodologies started to be to take care of those issues. New solution for alert correlation is given in Paper [3] to reconstruct attack thread. It is also known as attack instance recognition. It has not utilized any clustering techniques, but rather straightforward sorting is utilized. The results of the sorting are displayed in a temporal window. It has copies of alerts too. This alerts issue has been counteracted in Paper [4] which is for the most part like Paper [3]. Along these lines it gives more compact method for alert presentation. This sort of methodology is likewise utilized as a part of Paper [5] where clustering techniques is utilized for the same reason. Alert Clustering methodology is utilized by paper [6] light

DOI 10.17148/IJARCCE.2015.4862

of the closeness of assault events. It considers certain time and any two cases of assault are viewed as comparative when the two happen in a particular time window other than the careful closeness of their destination and source. Because of using detection mechanism as low level IDS, these detections systems may not work effectively in real time applications as they use imperfect classifiers. In Paper[7] also alert correlation methodology is utilized with the help of an supervisor known as weighted attribute wise closeness which figures out if to merge two given alerts or not. This methodology and analogous methodologies gave in Paper [8] and Paper [9] is having a downside which is the need of giving numerous parameters to the framework. Same weakness is found with Paper [10] also where no direction is given to acquire great qualities. In Paper [11] Attribute wise comparability measures alongside parameters given by client is utilized. As it additionally includes in sorting alerts in ascending or descending order based on the source and destination, it deteriorates closeness measure. Different methodologies are proposed in Paper [12] for fusion alerts. The first approach groups related alerts based on IP addresses. The second approach and third approach use some data mining techniques known as supervised learning techniques. In the field of intrusion scenario detection as presented in Paper [13], [14] and [15] many similar tetchiness are used to making alert correlation. Out them very important procedure for scenario detection is in [13]. Base on an algorithm by name CURE, offline clustering solution is proposed in Paper [8]. The solution makes use of numeric attributes only. The problem with this approach is that the security expert who gives knowledge of domain expertise must be having knowledge about current attack instances. Another clustering solution is proposed in [8]. This proposal is closely similar to our approach. Its clustering method is known as “link based clustering”. It focuses on the reasons or Meta data about the alerts generated by IDS. Only root causes are considered here. There is a problem of ignoring alerts that form into smaller clusters. The main difference between the [8] and our approach is that the [8] supports only offline intrusion detection. It depends on the historical traces present in the log files. However, our approach supports both offline and also online intrusion detection mechanism that makes it unique from existing IDSs. The alert clustering approach in [9] is also having good feature that reduces the number of false positives. This is also based on [9] in case of alert clustering. The approach presented in [16] is different completely. It makes use of reconstruction error of AA-NN (Auto Associate Neural Network) to differentiate alerts. Its approach is that it considers all alerts are same if they have same reconstruction error and put them into the same cluster. And this works in online and offline scenarios. The training requirements for AA-NN are training phase and also an offline training Phase. In paper [17] Alert aggregation is an important subtask of intrusion detection. The goal is to identify and to cluster different Alert produced by low-level intrusion detection systems, firewalls, etc belonging to a specific attack instance which has be initiated by an attacker at a certain point in time.

III. REAL TIME INTRUSION ALERT AGGREGATION

In real Time Intrusion alert strategy aggregation depending upon their current status, the corresponding information about the user is clustered. Clustering value will change according to the data set being used. The main advantage is that it starts clustering as the individual data set until the data set ends rather than clustering whole. The data set are normalized and they are clustered and true positive, false positive errors are identified. Data stream alert aggregation of offline alert deals only with the outdated log files in which security to the data is not guaranteed. Offline alert shows only the result of archaic information. Upgraded version of offline alert is online alert. Each stream of data is check now and then to regulate the flow of information without intruder. If any abnormal flow of information persists they are termed as intruders. The user information, login time are compared. The origin of the intruders is identified and they are discarded by which false positive error is also reduced [17].

IV. DISTRIBUTED INTRUSION DETECTION SYSTEM

As intruder, and attacks routines turn out to be progressively unpredictable, the requirement for a DIDS framework in huge corporate and military systems increments definitely. With the expanded versatile nature of these intruders, experts are abandoning themselves open to the issues of correspondences breakdowns, where one examiner sees an attack on his section, and releases it as nothing. While a few different sections getting the same attacks in a composed way, their analysts may be rejecting the intensity of the attacks. Then again, when all the attack information is seen together, a drastically alternate point of view the attacks may rise. The central examination server is truly the complete set of the operation. Centre Analysis server would ideally consist of a database and Web serve. It additionally allows analyst to perform pre-modified inquiries, for example, total number of attacks, measurements collective incident, to distinguish attack patterns and to perform simple event examination, all from a Web interface. It will screen all exchange or information action that happens in the client end. DIDS monitors a heterogeneous network of computers and combines distributed monitoring and data reduction with centralized data analysis. On each host monitor Collect and analyses audit records from the host operating system. The host monitors also trace user sessions and reports anomalous behaviour to the director. In DIDS a specific IDS Agent is running on the administration and in the event of a ready it takes custom receptive activities. Furthermore broadcast the alerts in its communication groups. The DIDS framework gives the a speedier, less demanding, more productive technique to recognize attacks over numerous systems, fragments and to follow backs the activity of the attackers. By having these attacks records put away in a single place, it allows the analyst significantly more adaptability in finding different attacks, and other attack issues which may have generally gone unnoticed

DOI 10.17148/IJARCCE.2015.4862

V. IMPLEMENTED SYSTEM

In Real time Intrusion alert aggregation system each stream of data is check at this instant and then to regulate the flow of information without intruder. If any anomalous flow of information comes they are termed as intruders. The user information, login time are compared. The origin of the intruders is identified and they are discarded by which false positive error is also reduced. Each Intrusion Detection System detects attacks which occur within their network segment, and alert is generated on the server side. All IDS send their alerts to centralized IDS server. Centralized systems analyses all alert data and make analysis reports. If same number of alerts is continuously generated by the system, aggregation concepts applied on generated alerts and Meta alert is generated, and send that Meta alert via email to the admin. Alerts are stored in log files, Each IDS stored different types of alerts. System also detects real time attacks. System checked on offline data set which shows aggregation of alerts, number of different types of attacks occur on system. Figure 1 shows overview of implemented system.

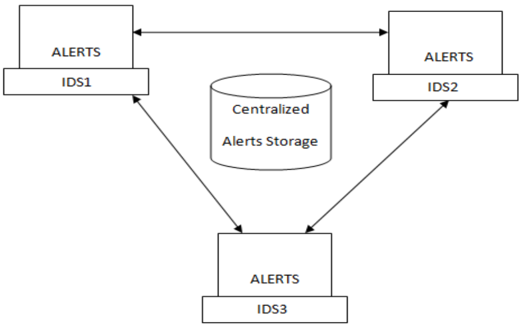
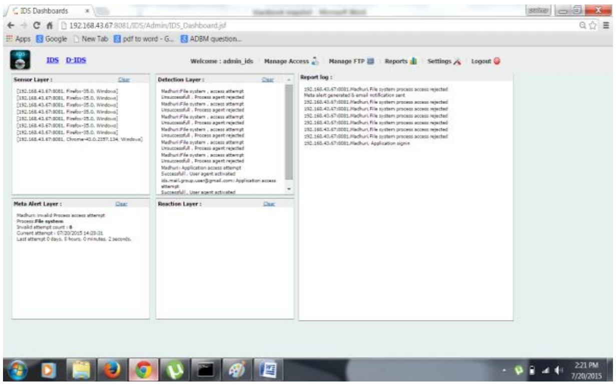


Fig.1. Overview of Implemented System

Intrusion Detection agents through self-organized association form a distributed intrusion detection system. DIDS monitors a heterogeneous network of computers and combines distributed monitoring and data reduction with centralized data analysis. System consists of server, number of clients, and database for accessing valid user data. Server has layer approach. Each layer performs its own assigned task. Client1, Client 2, client N we can be connected to the main server. Main Server gives access to the different clients for accessing process and different modules on server system. If invalid user tries to access the system then detect as an intruder. On client side user level and process level agent is activated. Client can access only those process and modules they having permission for that, if they try to for another process they are detected as an intruder. We test out proposed system on offline mode for showing the alert aggregation work. We also implement real time different attack for example SQL injection attack. Alert which are generated by the system are stored in database, for future reference. Meta alert is send to the admin on his email. Real time intrusion alert aggregation approach is used in implemented system. Whenever same types are alerts are generated continuously, after particular number of alert count Meta alert is generated, if same intruder try to access system, intrusion detection system block the client, or intruder.

VI. IMPLEMENTATION AND RESULT ANALYSIS

The main objective of Implemented System is to detect intrusion activity, alert generation and alert aggregation. When any malicious activity is performed by clients or intruder alert is generated on server side and also warning is given to client by server. The IDS system generates meta-alerts on offline and online alert data. IDS system cluster similar alerts and form one Meta alert and that Meta alert is send to admin by email. Figure 2 shows Meta alert generation by server and Figure 3 Shows how system is prevented from inside intruder.



Sensor Layer :

[192.168.43.67:8081, Firefox-35.0, Windows]
[192.168.43.67:8081, Firefox-35.0, Windows]
[192.168.43.67:8081, Firefox-35.0, Windows]
[192.168.43.67:8081, Firefox-35.0, Windows]
[192.168.43.67:8081, Firefox-35.0, Windows]
[192.168.43.67:8081, Firefox-35.0, Windows]
[192.168.43.67:8081, Firefox-35.0, Windows]
[192.168.43.67:8081, Chrome-43.0.2357.134, Windows]

Detection Layer :

Madhuri :File system , access attempt
Unsuccessfull , Process agent rejected
Madhuri :File system , access attempt
Unsuccessfull , Process agent rejected
Madhuri :File system , access attempt
Unsuccessfull , Process agent rejected
Madhuri :File system , access attempt
Unsuccessfull , Process agent rejected
Madhuri :File system , access attempt
Unsuccessfull , Process agent rejected
Madhuri :File system , access attempt
Unsuccessfull , Process agent rejected
Madhuri : Application access attempt
Successfull , User agent activated
ids.mail.group.user@gmail.com : Application access attempt
Successfull , User agent activated

Report log :

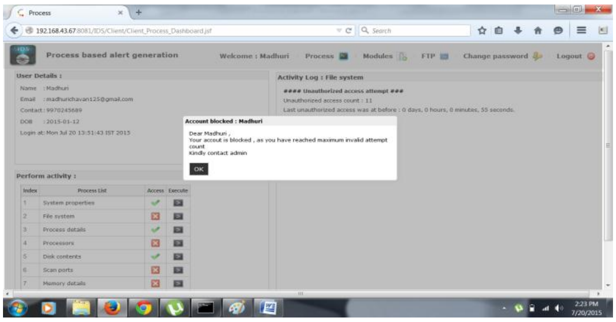
192.168.43.67:8081,Madhuri, File system process access rejected
Meta alert generated & email notification sent
192.168.43.67:8081,Madhuri, File system process access rejected
192.168.43.67:8081,Madhuri, File system process access rejected
192.168.43.67:8081,Madhuri, File system process access rejected
192.168.43.67:8081,Madhuri, File system process access rejected
192.168.43.67:8081,Madhuri, File system process access rejected
192.168.43.67:8081,Madhuri, File system process access rejected
192.168.43.67:8081,Madhuri, Application signin

Meta Alert Layer :

Madhuri: invalid Process access attempt
Process:**File system**
Invalid attempt count : 8
Current attempt : 07/20/2015 14:28:31
Last attempt 0 days, 0 hours, 0 minutes, 2 seconds.

Reaction Layer :

Fig.2. Meta Alert Generation


Process
192.168.43.67:8081/IDS/Client/Client_Process_Dashboard.jsf
Search
Process based alert generation
Welcome : Madhuri
Process
Modules
FTP
Change password
Logout
OK
2:23 PM
7/20/2015

User Details :

Name : Madhuri
Email : madhurichavan125@gmail.com
Contact : 9970245689
DOB : 2015-01-12
Login at: Mon Jul 20 13:51:43 IST 2015

Activity Log : File system

#### Unauthorized access attempt ###
Unauthorized access count : 11
Last unauthorized access was at before : 0 days, 0 hours, 0 minutes, 55 seconds.

Account blocked : Madhuri

Dear Madhuri ,
Your accout is blocked , as you have reached maximum invalid attempt count
Kindly contact admin

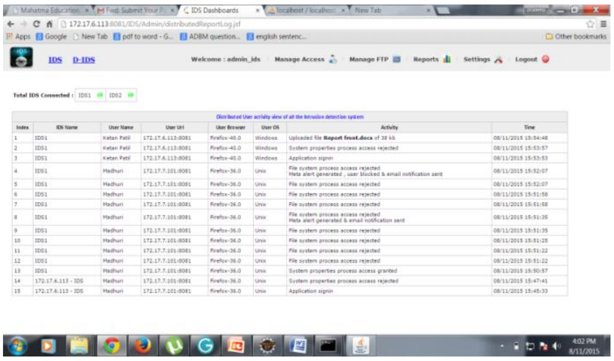
Perform activity :

Index	Process List	Access	Execute
1	System properties	✓	▶
2	File system	☒	▶
3	Process details	✓	▶
4	Processors	☒	▶
5	Disk contents	✓	▶
6	Scan ports	☒	▶
7	Memory details	☒	▶

Fig.3. Prevention From Inside Intruder

Distributed Intrusion Detection System

DIDS stores all user activity of all IDS. Following Figure 4 gives overview of DIDS.



Total IDS Connected : IDS1 IDS2

Distributed User activity view of all the Intrusion detection system

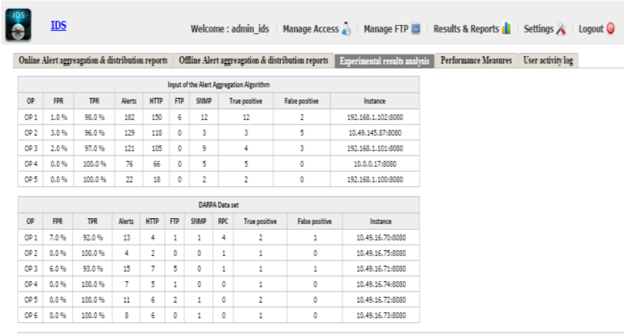
Index	IDS Name	User Name	User Url	User Browser	User OS	Activity	Time
1	IDS1	Ketan Patil	172.17.6.113:8081	Firefox-40.0	Windows	Uploaded file **Report front.docx** of 38 kb	08/11/2015 15:54:48
2	IDS1	Ketan Patil	172.17.6.113:8081	Firefox-40.0	Windows	System properties process access rejected	08/11/2015 15:53:57
3	IDS1	Ketan Patil	172.17.6.113:8081	Firefox-40.0	Windows	Application signin	08/11/2015 15:53:53
4	IDS1	Madhuri	172.17.7.101:8081	Firefox-36.0	Unix	File system process access rejected Meta alert generated , user blocked & email notification sent	08/11/2015 15:52:07
5	IDS1	Madhuri	172.17.7.101:8081	Firefox-36.0	Unix	File system process access rejected	08/11/2015 15:52:07
6	IDS1	Madhuri	172.17.7.101:8081	Firefox-36.0	Unix	File system process access rejected	08/11/2015 15:51:58
7	IDS1	Madhuri	172.17.7.101:8081	Firefox-36.0	Unix	File system process access rejected	08/11/2015 15:51:58
8	IDS1	Madhuri	172.17.7.101:8081	Firefox-36.0	Unix	File system process access rejected Meta alert generated & email notification sent	08/11/2015 15:51:35
9	IDS1	Madhuri	172.17.7.101:8081	Firefox-36.0	Unix	File system process access rejected	08/11/2015 15:51:35
10	IDS1	Madhuri	172.17.7.101:8081	Firefox-36.0	Unix	File system process access rejected	08/11/2015 15:51:28
11	IDS1	Madhuri	172.17.7.101:8081	Firefox-36.0	Unix	File system process access rejected	08/11/2015 15:51:22
12	IDS1	Madhuri	172.17.7.101:8081	Firefox-36.0	Unix	File system process access rejected	08/11/2015 15:51:22
13	IDS1	Madhuri	172.17.7.101:8081	Firefox-36.0	Unix	System properties process access granted	08/11/2015 15:50:57
14	172.17.6.113 - IDS	Madhuri	172.17.7.101:8081	Firefox-36.0	Unix	System properties process access rejected	08/11/2015 15:47:41
15	172.17.6.113 - IDS	Madhuri	172.17.7.101:8081	Firefox-36.0	Unix	Application signin	08/11/2015 15:45:33

Fig.4. DIDS

DOI 10.17148/IJARCCE.2015.4862

Result Analysis

Figure 5 shows the number of alerts produced from the different OP. In addition, the number of attack instances for which at least one alert is generated by the detection layer is also given. We are aware of the various critique on the DARPA benchmark data and the limitations that emerge thereof.


IDS
IDS
Welcome : admin_ids
Manage Access
Manage FTP
Results & Reports
Settings
Logout
Online Alert aggreagation & distribution reports
Offline Alert aggreagation & distribution reports
Experimental results analysis
Performance Measures
User activity log

Input of the Alert Aggregation Algorithm

OP	FPR	TPR	Alerts	HTTP	FTP	SNMP	True positive	False positive	Instance
OP 1	1.0 %	98.0 %	182	150	6	12	12	2	192.168.1.102:8080
OP 2	3.0 %	96.0 %	129	118	0	3	3	5	10.49.145.87:8080
OP 3	2.0 %	97.0 %	121	105	0	9	4	3	192.168.1.101:8080
OP 4	0.0 %	100.0 %	76	66	0	5	5	0	10.0.0.17:8080
OP 5	0.0 %	100.0 %	22	18	0	2	2	0	192.168.1.100:8080

DARPA Data set

OP	FPR	TPR	Alerts	HTTP	FTP	SNMP	RPC	True positive	False positive	Instance
OP 1	7.0 %	92.0 %	13	4	1	1	4	2	1	10.49.16.70:8080
OP 2	0.0 %	100.0 %	4	2	0	0	1	1	0	10.49.16.75:8080
OP 3	6.0 %	93.0 %	15	7	5	0	1	1	1	10.49.16.71:8080
OP 4	0.0 %	100.0 %	7	5	1	0	0	1	0	10.49.16.74:8080
OP 5	0.0 %	100.0 %	11	6	2	1	0	2	0	10.49.16.72:8080
OP 6	0.0 %	100.0 %	8	6	0	1	0	1	0	10.49.16.73:8080

Fig.5. Experimental Result analyses

Performance Measure

In order to assess the performance of the alert aggregation, we evaluate the following measures:

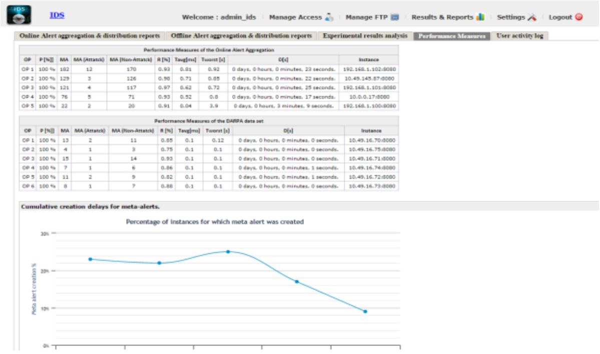
Percentage of detected instances (p): We regard an attack instance as being detected if there is at least one Meta alert that predominantly contains alerts of that particular instance. The percentage of detected attack instances p can thus be determined by dividing the number of instances that are detected by the total number of instances in the data set.

Number of meta-alerts (MA) and reduction rate (r): The number of meta-alerts (MA) is further divided into the number of attack meta-alerts MAattack which predominantly contain true alerts and the number of nonattack meta-alerts MAnonattack which predominantly contain false alerts. The reduction rate r is 1 minus the number of created meta alerts MA divided by the total number of alerts N.

Average runtime (tavg) and worst case runtime (tworst): 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. The worst case runtime tworse, which is measured in seconds, states how long it takes.

Meta-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 and the creation of the first meta-alert for that instance.
Figure 6 shows performance measure.

Online and offline alert aggregation chart
Following Figure 7 and Figure 8 gives oveall idea of offline alert aggregation and online alert aggregarion system.


IDS
IDS
Welcome : admin_ids
Manage Access
Manage FTP
Results & Reports
Settings
Logout
Online Alert aggreagation & distribution reports
Offline Alert aggreagation & distribution reports
Experimental results analysis
Performance Measures
User activity log
Cumulative creation delays for meta-alerts.
Percentage of instances for which meta alert was created
30%
20%
10%
0%
Meta alert creation %

Performance Measures of the Online Alert Aggregation

OP	P [%]]	MA	MA (Attatck)	MA (Non-Attatck)	R [%]	Tavg[ms]	Tworst [s]	D[s]	Instance
OP 1	100 %	182	12	170	0.93	0.81	0.92	0 days, 0 hours, 0 minutes, 23 seconds.	192.168.1.102:8080
OP 2	100 %	129	3	126	0.98	0.71	0.85	0 days, 0 hours, 0 minutes, 22 seconds.	10.49.145.87:8080
OP 3	100 %	121	4	117	0.97	0.62	0.72	0 days, 0 hours, 0 minutes, 25 seconds.	192.168.1.101:8080
OP 4	100 %	76	5	71	0.93	0.52	0.8	0 days, 0 hours, 0 minutes, 17 seconds.	10.0.0.17:8080
OP 5	100 %	22	2	20	0.91	0.04	3.9	0 days, 0 hours, 3 minutes, 9 seconds.	192.168.1.100:8080

Performance Measures of the DARPA data set

OP	P [%]]	MA	MA (Attatck)	MA (Non-Attatck)	R [%]	Tavg[ms]	Tworst [s]	D[s]	Instance
OP 1	100 %	13	2	11	0.85	0.1	0.12	0 days, 0 hours, 0 minutes, 0 seconds.	10.49.16.70:8080
OP 2	100 %	4	1	3	0.75	0.1	0.1	0 days, 0 hours, 0 minutes, 0 seconds.	10.49.16.75:8080
OP 3	100 %	15	1	14	0.93	0.1	0.1	0 days, 0 hours, 0 minutes, 0 seconds.	10.49.16.71:8080
OP 4	100 %	7	1	6	0.86	0.1	0.1	0 days, 0 hours, 0 minutes, 1 seconds.	10.49.16.74:8080
OP 5	100 %	11	2	9	0.82	0.1	0.1	0 days, 0 hours, 0 minutes, 1 seconds.	10.49.16.72:8080
OP 6	100 %	8	1	7	0.88	0.1	0.1	0 days, 0 hours, 0 minutes, 0 seconds.	10.49.16.73:8080

Fig.6. Performance Measure

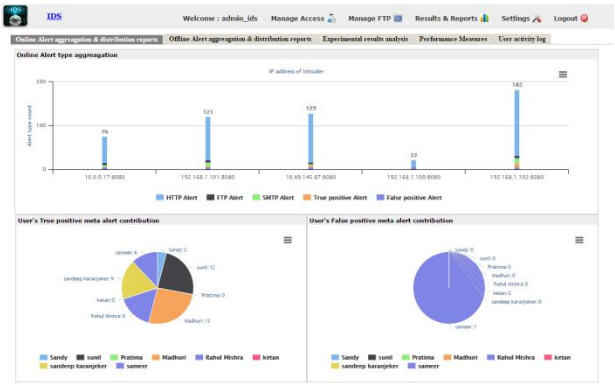


Fig 7. Online alert Aggregation chart

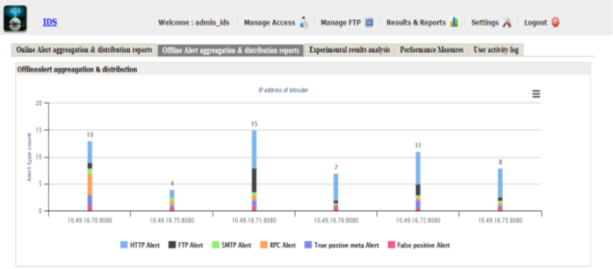


Fig. 8.Offline alert Aggregation chart

VII. CONCLUSION

Real time intrusion alert aggregation System having broad applicability of offline alert and real time generated alert data aggregation along with Meta alert generation in order to reduce the amount of alert data. System identifies and monitors unauthorized activities inside network and blocks the intruder. In process of clustering and aggregation, false positive and false negative alert rate is reduced. We have also demonstrated User level, Module level, and Process level agent with continuous streaming of HTTP and FTP data and corresponding alert aggregation and reports generation by the IDS.

VIII. FUTURE SCOPE

We will investigate how human domain knowledge can be used to improve the detection processes. The IDS product should detect, in near real-time, any kinds of attempts to exploit known weaknesses, or to probe our internal

DOI 10.17148/IJARCCE.2015.4862

network. They should also keep track of attempts to overload necessary resources. Along with this, they should perhaps sound an alarm, trigger some predefined action, and keep a good log for analysis

REFERENCES

1. S. Axelsson, "Intrusion Detection Systems: A Survey and Taxonomy," Technical Report 99-15, Dept. of Computer Eng., Chalmers Univ. Of Technology, 2000.
2. A. Allen, "Intrusion Detection Systems: Perspective," Technical Report DPRO-95367, Gartner, Inc., 2003.
3. 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.
4. H. Debar and A. Wespi, "Aggregation and Correlation of Intrusion-Detection Alerts," Recent Advances in Intrusion Detection, W. Lee, L. Me, and A. Wespi, eds., pp. 85-103, Springer, 2001.
5. D. Li, Z. Li, and J. Ma, "Processing Intrusion Detection Alerts in Large-Scale Network," Proc. Int‟l Symp. Electronic Commerce and Security, pp. 545-548, 2008.
6. F. Cuppens, "Managing Alerts in a Multi-Intrusion Detection Environment," Proc. 17th Ann. Computer Security Applications Conf. (ACSAC ‟01), pp. 22-31, 2001
7. A. Valdes and K. Skinner, "Probabilistic Alert Correlation," Recent Advances in Intrusion Detection, W. Lee, L. Me, and A. Wespi, eds. pp. 54-68, Springer, 2001.
8. K. Julisch, "Using Root Cause Analysis to Handle Intrusion¨ Detection Alarms," PhD dissertation, Universitat Dortmund, 2003. 294 IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING.
9. T. Pietraszek, "Alert Classification to Reduce False Positives in¨ Intrusion Detection," PhD dissertation, Universitat Freiburg, 2006.
10. F. Autrel and F. Cuppens, "Using an Intrusion Detection Alert Similarity Operator to Aggregate and Fuse Alerts," Proc. Fourth Conf. Security and Network Architectures, pp. 312-322, 2005.
11. 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.
12. O. Dain and R. Cunningham, "Fusing a Heterogeneous Alert Stream into Scenarios," Proc. 2001 ACM Workshop Data Mining for Security Applications, pp. 1-13, 2001.
13. P. Ning, Y. Cui, D.S. Reeves, and D. Xu, "Techniques and Tools for Analyzing Intrusion Alerts," ACM Trans. Information Systems Security, vol. 7, no. 2, pp. 274-318, 2004.
14. F. Cuppens and R. Ortalo, "LAMBDA: A Language to Model a Database for Detection of Attacks," Recent Advances in Intrusion Detection, H. Debar, L. Me, and S.F. Wu, eds. pp. 197-216, Springer, 2000.
15. S.T. Eckmann, G. Vigna, and R.A. Kemmerer, "STATL: An Attack Language for State-Based Intrusion Detection," J. Computer Security, vol. 10, nos. 1/2, pp. 71-103, 2002.
16. M.S. Shin, H. Moon, K.H. Ryu, K. Kim, and J. Kim, "Applying Data Mining Techniques to Analyze Alert Data," Web Technologies and Applications, X. Zhou, Y. Zhang, and M.E. Orlowska, eds. pp. 193-200, Springer, 2003.
17. R. Smith, N. Japkowicz, M. Dondo, and P. Mason, "Using Unsupervised Learning for Network Alert Correlation," Advances in Artificial Intelligence, R. Goebel, J. Siekmann, and W. Wahlster, eds. pp. 308-319, Springer, 2008.
18. Alexander Hofmann, Bernhard Sick "Online Intrusion alert aggregation with generative Data Modelling" IEEE transactions on dependable and secure computing, VOL.8, No.2 March- April 2011.

DOI 10.17148/IJARCCE.2015.4862