ISSN:0975-9646

CBSE as Novel Approach for IDS

Mohit Angurala, Malti Rani

Computer Science Deptt,
Punjab Institute of Technology (PTU Main Campus Kapurthala/ Punjab Technical University,
Jalandhar-Kapurthala Road, India

***Abstract*— This paper gives a new approach called Component based software engineering for designing an intrusion detection system. This paper also tells that before this component based software approach, other traditional approaches were used for designing intrusion detection system. But due to various drawbacks we proposed this component based approach for intrusion detection system. A comparison table is also illustrated in this paper on the basis of class diagram designed using UML language in a simulated environment. This comparison table will illustrate main difference between the traditional and component based approach and also shows that which parameters make this approach a better approach.**

***Keywords*—Component based software engineering, intrusion detection system, Classes etc.**

I. Introduction

The Network security has become essential to users of purposes, organizations and computers. As intrusion attempts are increasing day by day therefore we are unable to find such intrusions because our Intrusion detection system is not capable enough to get updated with these new emerging attacks. Therefore we are unable to detect such emerging attacks[8]. On the other hand we have new technique known as component based software engineering which is quite fast technique in comparison with old techniques such as waterfall model, spiral model etc.

Network based intrusion detection system are most common and examine passing network traffic for signs of intrusions. Host-based systems look at user and process the activity on the local machine for the signs of intrusions. Since each type has specific strengths and weakness.

II. Components Of Intrusion Detection System

Intrusion Detection System has some of the basic components . These are explained as follows:

A. Packet decoder: Through different Network interfaces packet decoder takes packets from it. Example like point to point, Ethernet etc.

B. Preprocessors: These are components or plugins with the purpose to change data packets prior to the detection engine does several action to locate if the packet is being used by several intruders.

C. Detection Engine: Responsibility of detection engine is for detecting any intrusion activity if there in a packet. For this, Snort rules are applied to the detection engine. Internal data structure read out these rules where they are matched beside all packets. Detection engine's load depends upon the following factors like No. of rules, Power of the machine on which Snort is running, rapidity of internal bus used in the Snort machine, Network's load.

D. Logging And Alerting System: The packet may be used to log the activity or generate an alert depending upon what the detection engine looks inside a packet. Logs are set aside in tcp dump style files, text files or some other form.

E. Output Modules: These modules can carry out dissimilar operations depending on the way you want to hoard output generated by the logging and alerting system of Intrusion Detection System.

F. Database Engine: This Engine is used for keeping logs that enclose information regarding the intrusions [11].

III. Designing Issues Related To Intrusion Detection System

Several problems or issues are there related to the designing of IDS. Some Issues of IDS are discussed as follows:

A. Time: Time is the main subject or worry since at the present time technical knowledge is declining with the boost in tools that are automated [9]. Attackers or intruders are becoming further superior since they are building attacks with the assistance of ready-made tools as well as they at all times bring several new attacks but intrusion detection system designed with the a variety of models are not greatly advanced [10]. They are not able to get updated quicky as compare to new attacks. So, timely updating is necessary which is not possible in case of old models like waterfall, Iterative etc.

B. Cost: Cost is also most important worry or issue to the designing of IDS since conventional models were not that advanced that if one part of IDS fails. It is not possible to locate fault in it. So, maintenance cost is fairly elevated in this case [7]. Furthermore on the whole, development cost is also extremely elevated since the software is built from scratch as well as extra labour is desirable to create this software which leads to added cost.

C. Maintenance: In traditional models maintenance cost were quite high because if there occurs error in one part of the system, whole system has to be modified.

D. Low Quality Product: Products build with the traditional model approach were not of that good quality because those products were more prone to errors as compare to the the new introduced approach component based software engineering [3].

E. Difficult to Update: The products designed using traditional approaches were difficult to update, therefore they were more sensitive to attacks and therefore we need such system in which updation is required and chances of attacks are less [8].

Due to lot of limitations of traditional software models we have proposed a model for designing IDS with Component Based Software Engineering approach[4]. CBSE is the way to define, implement and integrate or compose loosely coupled independent components into systems [5].

IV. Proposal Approach

CBSE has turn out to be a generally used development model and an vital software development approach since software systems are becoming better and further complex and customers are demanding extra reliable software which is developed extra speedy [2]. With this approach, software systems are made by composition of reusable building blocks called components. Such type of requirement of necessities and dependency makes the components easier to reuse and consequently it allows for quick development[6]. The reward of this kind of CBD include less significant development time, lesser costs, reusability features[1].

Our proposed algorithm for Intrusion Detection system is as follows:

```
Step 1.     Start
Step 2.     Select (ALT,UPD,LF,SB,FA,AB,NN)
            ALT=Alert
            UPD=Updater
            FA=False Alarm
            LF=Log Files
            SB=Signature Based Intrusion
            AB=Anomaly Based Intrusion
            NN=Normal Network
Step 3      Select Network
Step 4      Capture Intrusion
Step 5      Check Intrusion Type
Step 6      IF ID==1
Step 7          Check Log File And Look for SB match
Step 8          If Match is True
Step 9              Then Send Alert Message
Step 10         Else Match Unsuccessful Goto step11
Step 11             Check If Match is AB
Step 12                 If match is True
Step 13                 Send Alert Message And
            Update Record In Updater
                        and stop
Step 14                 Else Generate FA and stop
Step 15     Else goto NN and stop
```

On the basis of above algorithm we made a class diagram using tool and this class diagram clearly depicts all the main classes along with the operations each class is performing. There are seven classes in our class diagram and this class diagram clearly shows working or operation or function that each class performs.

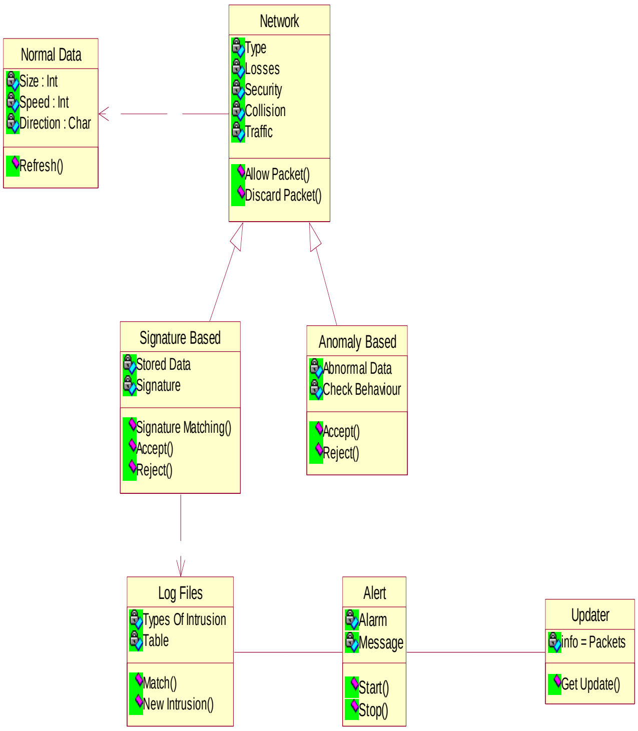

Fig. 1 Class Diagram of UML

On the basis of class diagram now we can compare our approach with the traditional approaches that were used before our proposed approach. The Following table gives us the clear comparison of the Traditional or existing Techniques with the proposed technique that is Component Based Software Engineering.

Table I

Sr No.	Comparison Table		
	Parameters	Existing Techniques	Proposed Techniques
1	Cost	High	Less than traditional
2	Maintenance	Hard	Moderate
3	Security	Moderate	Moderate
4	Testing	Moderate	Easy
5	Requirement Gathering Time	High	Low
6	Coding	Very High	Low
7	Verification	Hard	Easy
8	Validation	Hard	Easy
9	Upgradation	Hard	Easy
10	Reliability	Moderate	Moderate
11	Scalability	Hard	Easy
12	Flexibility	Hard	Easy
13	Doccumentation Time	High	Low
14	Development Time	High	Low
15	Reusability	Low	High
16	Risk Factor	Moderate	Moderate
17	Implementation	Hard	Moderate

V. CONCLUSIONS

We proposed an algorithm for intrusion detection system and on the basis of this algorithm we also created class diagram using UML language in a tool. This class diagram shows different classes that are involved in our proposed algorithm. Each class has its own operations. A comparison chart is also shown at the last which clearly shows that component based approach is better than the traditional approach. We showed this with the help of parameters that are involved in designing our Intrusion Detection system. In our approach we used component based software design to make our IDS much better than the traditional approaches like spiral model, Iterative model etc. Our table 1 lists some of the parameters which shows that how our proposed approach is better than the previous approaches.

ACKNOWLEDGMENT

I Mohit Angurala Student of Master in Technology CSE(Networking) would like to place on record my deep sense of gratitude to Assistant Professor Malti Rani Department of Computer Science Engineering (Networking), PIT, Kapurthala, India for her generous guidance, help and useful suggestions. I'm also grateful to her for believing in my potential and for support during this period. I express my sincere gratitude to Assistant Professor. Prabhdeep Singh, Department of Computer Science Engineering, KMV college jallandhar, India, for his stimulating guidance, continuous encouragement and supervision throughout the course of present work

REFERENCES

[1] Er. Mohit Angurala, Er. Malti Rani, Design and develop Intrusion Detection System Using Component Based Software Design International Journal on Recent and Innovation Trends in Computing and Communication volume: 2, issue : 4,April 2014

[2] Er. Iqbaldeep Kaur, Dr. P. K. Suri, Er. Amit Varma, Characterization and Architecture of Component Based Models International Journal of Advanced Computer Science and Applications Volume 1 –o.6, Dec 2010.p.p 66-68.

[3] Arvinder Kaur and Kulvinder Singh Mann Component Selection for Component based Software engineering, International Journal of Computer Applications (0975 – 8887) Volume 2 – No.1, May 2010. p.p 110-111.

[4] International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169 Volume: 2 Issue: 4 854 – 860

[5] Manju Kaushik and M. S. Dulawat, "A Comparision Between Traditional and Component Based Software Development Process Models" Journal of Computer and Mathematical Sciences Vol. 3, Issue 3, 30 June, 2012 Pages (248-421).

[6] Luiz Fernando Capretz, " Y: A new Component-Based Software Life Cycle Model ", Journals of Computer Science (1) : pp.76-82.

[7] Mohammad Sazzadul Hoque, Md. Abdul Mukit and Md. Abu Naser Bikas, "An Implementation Of Intrusion Detection System Using Genetic Algorithm", International Journal of Network Security & Its Applications (IJNSA), Vol.4, No.2, March 2012.

[8] K. Ilgun, R. Kemmerer, P. A. Porras, "State Transition Analysis: A Rule-Based Intrusion Detection Approach", IEEE Transaction on Software Engineering, 21(3):pp. 181-199. 1995.

[9] A. Sung, S. Mukkamala, "Identifying important features for intrusion detection using support vector machines and neural networks" in Symposium on Applications and the Internet, pp. 209–216. 2003.

[10] C. Kruegel, F. Valeur, G. Vigna, and R. Kemmerer. "Stateful intrusion detection for high-speed networks". In Proceedings of the IEEE Symposium on Security and Privacy, pp. 285-294, May 2002

[11] R. A. Kemmerer, and G. Vigna, "Sensor Families for Intrusion Detection Infrastructures", Managing Cyber Threats: Issues, Approaches and Challenges, ed. By V. Kumar, J. Srivastava and A. Lazarevic, Vol. 5, pp.1-41, Springer-Verlag, 2005.