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

Design and Develop an Intrusion Detection System Using Component Based Software Design

Er. Mohit Angurala, Er. Malti Rani

[1,2](Computer Science Deptt, Punjab Institute of Technology (PTU Main Campus Kapurthala/ Punjab Technical University, Jalandhar-Kapurthala Road, India, *mohitpit@gmail.com*)
mohitpit@gmail.com, malti_87@yahoo.co.in

Abstract--Network security is very much important because attacks are increasing day by day due to automated tools that are being used by hackers. Intrusion Detection System is useful for proving security as the basic purpose of Intrusion Detection System is to capture all the Intrusions or Intrusions attempts made by the hacker. IDS is used for proving security from the threats or intrusion attempts made by hackers. There are different techniques that were/are used for designing the Intrusion Detection System like the traditional models for example waterfall model, build and fix model, spiral model etc. Problems with these traditional models are like these are not able to update IDS with the new threats or intrusions made by hackers. Moreover Time and Cost associated for designing new IDS with these traditional approach is much more. In this paper Component Based Software Engineering technique is used for designing the Intrusion Detection System which is far better approach than the traditional approaches. There are many benefits associated with component based software engineering like it takes less time and cost to develop any system and maintenance is easy in this case because if any one of the component fails, it can easily be replaced with new one. In this approach different components are selected in a way that the chances of error is very less because every component is tested before they are integrated together.

Keywords-- *Component Based software engineering, Software reuse, Types of Intrusions, Intrusion detection Process*

*****

1. Introduction

Intrusion detection system examines intrusions or network activity to find possible intrusions or attacks in that network. Intrusion detection systems are of two types, either they are network based or they are host based. 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. There are three kind of generally available engines for the analysis which are event or signature based, statistical analysis and adaptive systems. The event or signature based act similar to as like an antivirus system with which most of the persons are familiar. The Network Intrusion detection system usually has two logical components: the sensor and the management stations. The sensor sits on the network segment and monitors it for some kind of suspicious activity. The management system receives alarm from the sensors and displays them to the operator. The sensors are usually dedicated systems that exists only to monitor the network [8]. They capture network traffic for analysis and if they detect something that is unusual they pass it back to the analysis station. But the major problem associated with IDS is that they are not updated as fast as new attacks are generated. So these IDS fails to provide security in that case where new attacks are generated.

1.1 Designing Methods

There are various designing strategies of Intrusion Detection systems that were/are used. These designing strategies are the various models which are used for making IDS [4]. These designing strategies are as follows:

1.1.1 Waterfall Model: Waterfall model is a model that do not produce high quality software because in this model once we move to next stage or next step, it is very difficult to move back to the previous stage but though it also have some advantage like at end of each phase we come to know whether our software can move to next phase or not.

1.1.2 Buid and Fix Model: In build and fix model we simply build a product that is tested as many times as possible to satisfy the client. This model is best

 ISSN: 2321-8169
854 – 860

suitable for short programming exercises 100 and 200 lines in length.

1.1.3 Rapid Prototyping Model: In this model the first step is to build a rapid prototype in order to determine the client's real needs and then to use that rapid prototype as the input to the waterfall model. This approach also has a useful side effect. Some organizations are reluctant to use the rapid prototyping approach because of the risks involved in using any new technology.

1.1.4 Incremental Model: In this model operational products are delivered at each stage. Software is not written but it is built. Build by build the product is completed, and developer delivers the product in same way. A typical product consists of 10 to 50 builds. At each stage, the client gets operational quality product that do a portion of what is required from delivery.

1.1.5 Spiral Model: In spiral model emphasis is given on risk analysis techniques along with elements of waterfall lifecycle model. This model is used for large projects therefore sometimes it is impossible to implement this model.

Therefore these are some of the models that are or were used previously to build intrusion detection system. Below table will show the comparison between the traditional approaches that ere or are still used for making intrusion detection system. Some models are quite better approach which can be implemented but still there are some facts about their weaknesses. Table shown below also shows the advantage as well as disadvantage for each of them.First column is for name of the traditional model, second column is for advantages of each process model and finaly third is the disadvantages of each process model [3].

Below table illustrates comparison of the traditional models like waterfall, Build and Fix, Rapid Prototyping,etc.

Table 1: Comparison Chart

Process Model	Advantages	Disadvantages
Waterfall Model	It is simple and easy to execute, neat and concise	Cannot move to previous stage
Build and Fix	Provides immediate feedback to developers	Consumes more time and is more costly
Rapid prototyping	Prototype gives more understanding to user about its functionality in early phase, reduces risk of failure	It is very slow process and too much changes can disturb development team
Incremental Model	Easier to test, more flexible	Requires good planning and design
Spiral Model	Risk management, prototyping controls cost	Complicated, requires knowledgeable management

2. Components of IDS

Following are the components of Intrusion Detection System:

2.1 Packet Decoder: Packet decoder takes packet from various types of network interfaces and prepares it to be sent to the detection engine. The interface may be Point to Point Protocol or Ethernet etc.

2.2 Preprocessor: Preprocessors are components or plugins that can be used to change data packets before the detection engine does some operation to find out if the packet is being used by some intruder.

2.3 Detection Engine: Detection Engine's responsibility is to detect if any kind of intrusion activity exists in a packet. The detection engine applies snort rules for this purpose. The rules are read into internal data structure where they are matched against all packets. The load on the detection engine depends upon the following factors: Number of rules, Power of the machine on which Snort is running, Speed of internal bus used in the Snort machine, Load on the network.

2.4 Logging And Alerting System: Depending upon what the detection engine looks inside a packet, the packet may be used to log the activity or generate an alert. Logs are kept in tcpdump style files, text files or some other form.

 ISSN: 2321-8169
854 – 860

2.5 **Output Modules:** Output modules can do different operations depending on how you want to save output generated by the logging and alerting system of IDS.

2.6 **Database Engine:** Database Engine is used for storing logs which contain information about the intrusion [9].

3. Issues of Designing IDS

There are several issues related to the designing of Intrusion Detection System. These Issues are discussed as follows:

3.1 **Time:** Time is the major issue or concern because nowadays technical knowledge is decreasing with the increase in automated tools. Hackers are becoming more advanced because they are making attacks with the help of readymade tools and they always come up with new new attacks but IDS designed with the various models are not much advanced. They are not as fastly updated as new attack is generated. Therefore Timely updation is necessary which is not possible in case of traditional software models like waterfall, Iterative etc.

3.2 **Cost:** Cost is also major concern or issue to the designing of Intrusion Detection System because traditional models are not that advanced that if one part of IDS fails. It is impossible to find error in it. So, the cost of maintenance is quite high in this case. Moreover overall development cost is also very high because the software is built from scratch and more labour is needed to make this software which leads to more cost.

3.3 **Maintenance:** Maintenance cost is also very high in case software is build with the traditional software models.

3.4 **Low Quality Product:** The products produced with the traditional software models are of low quality because testing is rarely done in this case which ultimately leads to poor quality.

3.5 **Difficult to Update:** The software developed with the traditional software models are difficult to update because it may harm previous files or software if we try to update our previous model. In other words it is not flexible to update.

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

4. Proposal Approach

CBSE has become a commonly used development paradigm and an important software development approach because software systems are becoming larger and more complex and customers are demanding more dependable software that is developed more fast. Using this approach, software systems are built by composition of reusable building blocks called components. Such type of specification of provisions and dependencies makes the components easier to reuse and therefore it allows for fast development. The advantages of this type of component based development include lesser development time, lower costs, reusability features [1].

5. Flowchart

Flowchart for cbse goes like first of all when process starts, It with some of the parameters like set of user requirement, Set of component requirement, Component Shelf, Component repository and software requirement. Now suppose we have components from 1 to n. So we start scanning components from 1 to n and then we select a component. Now after selection of component we check if set of user requirement matches with the set of component requirement and as well as if software requirement matches with the set of component requirement. If theses condition matches and condition becomes true then we select that component and perform integeration testing on it and if condition becomes false we again scan component from 1 to n [2]. Now after performing integeration testing we set value of pass as 1. If condition comes out to be true then component is selected and SLA is generated for that component and if condition becomes false, we do not select component and the component is developed from the scratch. This is how flow of cbse goes.

6. Algorithm for CBSE

Selection of component is very much important because we select component according to the need or the type of

 ISSN: 2321-8169
854 – 860

system we want to draw. Moreover selection of component is based on whether it can provide good integeration with other components or not [7] Below is the proposed algorithm for component based software engineering.

Step 1. Start

Step 2. Select (SUR,SCR,CS, CR)
SUR = Set of User Requirements,
SCR = Set of Component Requirements,
CS = Component Shelf,
CR = Component Repository
SR =Software Requirement

Step 3 CR = {1 to n}

Step 4 Sel=Ø

Step 5 for i ← 1 to n

Step 6 If SUR = SCR && SR=SCR

Step 7 Return Selection (Sel);

Step 8 Call Integration testing ( )

Step 9 Set pass=1

Step 10 If Pass==1

Step 11 Add component && Generate SLA

Step 12 Else SUR≠ SCR

Step 13 Return "No Selection and developed the components from scratch

Step 14 End loop

Step 15 End

6.1 Work Flow For Algorithm

Work flow of the above algorithm is that In the First step when we start the process we define some parameters like set of user requirement, set of component requirement, component shelf, component repository and finally software requirement. Then in the third step we scan components in component repository from 1 to n. In fourth step we did not select anything and in next step for values from 1 to n if set of user requirement and set of component requirement matches plus if software requirement matches with set of component requirement then we select component and perform integration testing. In next step we set value of pass equals to 1 and if it becomes true we add component and SLA is generated and else if Set of user requirement is not equal to set of component requirement we develop component from scratch and end our loop here and exit.

6.2 Algorithm for Intrusion Detection System

Algorithm for intrusion detection system is proposed below:

Step 1. Start

Step 2. Select (FA,NNFDB,SBI,ABI)
FA=False Alarm
NNF=Normal Network Flow
DB=Database
SBI=Signature Based Intrusion
ABI=Anomoly Based Intrusion

Step 3 Select Network for detecting Intrusion

Step 4 Capture data using tool

Step 5 Preprocess Data

Step 6 IF ID==1

Step 7 Check Log And Match Intrusion for SBI

Step 8 If Match is successfull

Step 9 Then generate security alarm and exit

Step 10 Else Match Unsuccessfull Goto step11

Step 11 Check If Match is ABI

Step 12 If ABI match is True

Step 13 Generate security Alarm and Update Database and stop

Step 14 Else Generate FA and stop

Step 15 Else goto NNF and stop

7. Algorithm Explanation

Above proposed Algorithm is for Intrusion Detection System. In this algorithm first of all when process is started we define some values or variables FA as false alarm, NNF as Normal Network Flow, DB as DataBase, SBI as Signature Based Intrusion and ABI as Anomaly Based Intrusion and then we select network for detecting intrusion and then data is captured using some tool like tcp dump, ds sniff etc and in next step preprocess the data. Now if value of ID comes equal to 1 then check Log file and match intrusion for signature based attack or intrusion. If match is successful then generate security alarm and exit. If match is unsuccessful then match for Anomaly based attack. If Anomaly based is true then generate security alarm and update database engine and stop. Otherwise generate False Alarm and stop. And if ID is not equal to 1 then go to normal network flow.

8. Diagrams

In UML five diagrams are created. These five diagrams are class diagram, Use case diagram, Sequence diagram,

ISSN: 2321-8169
854 – 860

Component diagram, Activity diagram, State chart diagram and finally deployment diagram which are explained below.

First diagram is class diagram. Class diagram contains classes, interfaces, association and collaboration. This diagram is basically used when we want to develop any kind of system or in other words we can say that whenever we want to create any syatem we use class diagram for this purpose.

Second diagram is Use Case Diagram. It consists of use cases, actors and their relationships and a single use case diagram describes a particular functionality of a system.

Third diagram is Sequence diagram. It shows sequences of messages that flows from one object to another. The diagram is shown below

Fourth diagram is Component Diagram. It is used to describe components or physical aspects of system and their relationships.

Fifth diagram is Activity diagram. It is used to to represent flow from one activity to another in the form of flowchart. Diagram is shown below

Sixth diagram is State Chart diagram. It shows state change of class, interface etc. Diagram for state chart is shown as follows

Seventh and the last diagram represents deployment diagram. It is used to represent static deployment view of a system. It consists of nodes and their relationships. The diagram is shown below

8.1 Use Case Diagram

Fig. below is Use Case Diagram. It consists of use cases, actors and their relationships and a single use case diagram describes a particular functionality of a system.

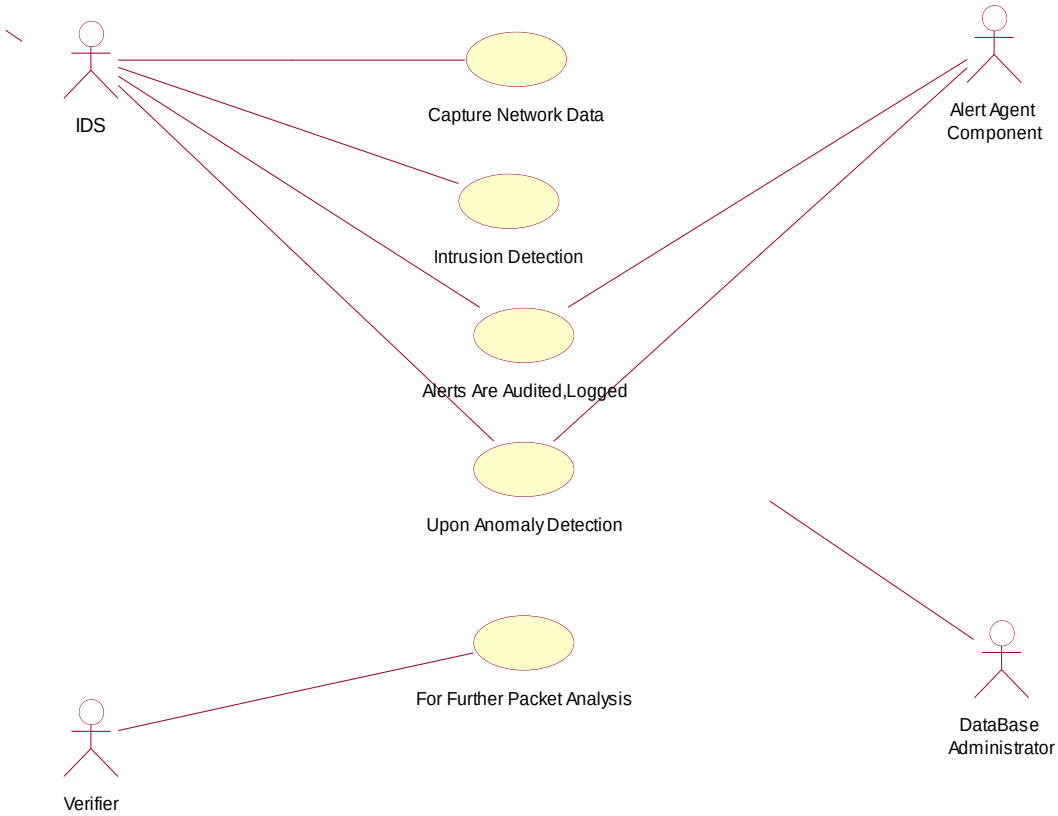

Figure 1: Use Case Diagram For IDS

The above shown diagram is use case diagram which basically shows different actors, use cases and the relationships between the use cases. This diagram is prepared in UML language using rational rose tool. The actors involved in our system are Intrusion Detection System, The alert agent, Verifier and finally the Database administrator. The use cases are to capture network data, Intrusion Detection, Audited and logged alerts, anomaly

detection and further packet analysis. Now function of Intrusion detection system is to capture network data and detect intrusion in it and corresponding database administrator will maintain information about the same in its database. If intrusions are detected then corresponding alerts will be generated for it by another actor that is alert agent and if abnormal behavour is detected it will get updated into database. Verifier will do further analysis of the packet.

8.2 Classes and Objects

Different classes and objects created in Intrusion detection system are as follows:

8.2.1 Normal Data: Normal data is the data that is flowing in the network.It do not contain intrusions or abnormal data in it.

8.2.2 Network: Network contains data that is flowing from one place to another. It has different parameters like type of network, losses in network, security in network, collision in network and finaly traffic in network.

8.2.3 Signature Based: This signature based class contains the type of intrusions that are already available in the database. These signature based Intrusions when detected are directly matched with the database or log file that is maintained in the database.It is different from anomaly based intrusion.

8.2.4 Anomaly based: Anomaly Based Class contains type of intrusion that is detected when abnormal events are detected that is when data behave in a abnormal way and this type of intrusions are more dangerous and are difficult to detect because log files are not maintained in this case. Therefore whenever such events are detected our database is updated with new record.

8.2.5 Log Files: Log File is another class which maintains list of abnormal as well as list of signature based events in such a way that if the same type of intrusion is again attempted then Log File can match with it and generate alarm.

8.2.6 Alerts: Alert class is the class which is created for the purpose of generating alerts when intrusions are generated. It can be in the form of alarm or it can be in form of message.

8.2.7 Updater: Updater class is created for the purpose of updating our database with the new intrusions that are generated or found. It will update the database engine as soon as new intrusion is found.

8. Conclusion and Future Work

In this paper new way is exploited to design Intrusion Detection System. This new technique proposed is Component Based Software Engineering. We first designed algorithm for Intrusion Detection System using this approach and then we have implemented this algorithm in UML language using rational rose. This technique is far better approach than the traditional software models that were used prior to this technique like waterfall model, iterative model, spiral model etc. Traditional approaches were slow techniques because attackers were updated with new attacks very quickly but IDS made with traditional approaches were not able to update with same speed. So our security was lacking behind. In this paper with the help of component based approach our IDS gets updated with the new attacks as soon as new attack arrives. We have explained designed Use Case diagram in UML language for our proposed algorithm and have also created classes and objects of Intrusion Detection System.

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. 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.

[2] 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.

[3] 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).

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

[5] 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.

[6] 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.

[7] 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.

[8] 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

[9] 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.

Mohit Angurala was born on 30 April 1990. He received B.tech degree in Informational Technology from Beant College of Engineering and Technology Gurdaspur (Punjab), India. He is now working towards M.Tech degree in CSE (Networking System) from Punjab Institute of Engineering and Technology which Punjab technical University Main Campus Kapurthala (Punjab), India. His research interest includes component based software engineering.