Chapter 5

Implementation of Advanced Intrusion Detection System

5.1 Introduction

In the implementation details, the developed system is brought into real practice for its evaluation. The implementation is done for online mode as well as offline mode of the IDS operation. The main reason behind such construction is into the existing IDSs development in recent years as mentioned in Akhlaq et al. (2012). As described in chapter 4, the integration of signature based and anomaly based detection methodologies is implemented as part of single system along with the quick response for any type of detected attacks by providing prevention policies. The whole system functionalities are provided via graphical user interface for the management console. This chapter covers all implementation strategies employed for the system implementation including system requirements identified, Run time architecture used for the construction and evaluation, module wise system details with their execution flow. Further, performance investigation of the framework is given in the chapter 6.

5.2 System Requirements Identified

In chapter 4, the theoretical and mathematical formulation for the developed IDS system is illustrated in depth. On the basis of these details the following requirements are identified during the analysis and design of the system architecture and the implementation details are then stated for the better understanding of the system.

The system requirement specifications are expressed in formal form to get each system function understanding properly and in easy way.

- The system will be able to provide the provision for the input dataset to do further analysis in the offline packet data input as well as input via online packet data capturing through any network traffic capturing tool to perform system evaluation.
- The input data both in online or offline mode will be preprocessed and will be made available in compatible form for next system operation.
• The preprocessed dataset packet feature attribute will be visualized and network administrator or analyst will have choice for selecting packet feature attributes as per the analysis requirements of the input packet data traffic.

• The system will provide the visualization portal for the attack signature generation as per the requirements of network administrators or security experts with easy graphical interface instead of remembering the difficult rule structure defined by SNORT like IDS.

• The system will provide the way to generate the signature patterns *i.e.* rules from the *A priori* algorithm execution with its modified version (as discussed in chapter 4) to get quickly the rules for the intrusion detection from the unknown traffic input dataset. Even, it should provide the flexibility for users of the system to generate the rules as per their own analysis and study intention too.

• The system will be used to do the signature based interruption location and to do the aberrance based interruption location.

• The system will be able to give the detection and analysis based on packet feature attribute with their classification type *i.e.* Basic TCP Connection Features, Packet Traffic Content Features and Packet Payload Features to get proper performance evaluation.

• The system will provide immediate response strategy once attack intrusion is detected and notified. To do this, it must have prevention policy mechanism as described in chapter 4.

• During the execution, the system will provide accurate intrusion detection with minimize false positive alarms.

• Mostly traditional IDS techniques implementation and evaluation techniques are complicated for the security analysts and for network administrators, as well as slow down with the increase in the input database, so the system will give simplified way for the IDS operation and will speed up the system computation with Data Mining techniques.
5.3 Software Used for System Development

Based on the identified system requirements, it is found that, the system should be deployed on any operating system instead of on specific operating system. Additionally, it should be executable with quicker response with huge database handling. Based on these expectations, this section illustrates about the software development platform details along with other supporting tools used during this implementation.

5.3.1 Use of Java (Programming Language)

The implemented system is developed with the Java platform with the specified details on Share.Java.net (2011) with Net beans version 6.9. The issues is of platform independency of the developed system is raised in the earlier section of the requirements, So this Java platform is used in support to resolve this concern. There are so many positive characteristics supported by Java Technology for the developers, out of these, the major one is its portability characteristic is taken into account for the implementation. Since, it means that the computer program source file written in the java language must execute on any operating system platform.

The main reason behind this, once the Java program is complied, then java program source code transforms to intermediate representation called Java Byte Code, instead of directly to running operating system machine code. These Java byte code instructions are similar with machine code, but mainly used by virtual machine (VM) developed specifically for the host machine hardware for interpretation. Generally, the end users use a Java Runtime Environment (JRE) installed and configured on their own machine for standalone Java applications. In Java, various Standardized libraries are given to provide a general way to access and use host-specific features such as graphics, threading, networking, and so on. This byte code gives the major benefit of porting and hence it is utilized in the developed system to provide platform independent feature.

5.3.2 Use of Net beans IDE version 6.9
The Net Beans IDE version 6.9 tools are integrated improvement environment accessible for Windows, Mac, Linux, and Solaris. These tool provide us an open source IDE and an application stage that offers engineers to quickly make and execute online, venture situated, desktop based, furthermore portable based applications utilizing the Java stage, and also Java development features for JavaFX, PHP, JavaScript and Ajax, Ruby and Ruby on Rails, Groovy and Vessels, and C/C++.

Following advantages are provided by the Net beans version 6.9 during the system implementation. It has given Java Debugger breakpoint gathering, debugger connect parameter history. Once more, it is backing for annotation processors in the editorial manager, configurable in the Venture Properties and even backing for New Applet and Web Start. Finally, with this tool, it is observed improved route in Stack Trace Analyzer and URLs and the feature of go to overridden/Implemented Method action.

5.3.3 Use of Oracle Database

In general, as described on Edusoft4u.com (2013) an Oracle database is a collection of data wrapped in a single unit. The reason for utilizing such database is to store and recover related data of the developed system during its execution. Since, we are dealing with huge amount of data captured for the packet information based on the supporting features either in online mode or offline mode of operation. Therefore, we focused on the database server where the database server is the key to solving the problems of information management for huge amount of data in multiuser environment with concurrent access. The main advantage of using database server, it also counteracts unapproved get to and gives productive answers for disappointment recuperation of the database.

Additionally, details provided on Pitbooks.com (2012) oracle Database is the first database which is intended for big business framework figuring on the most adaptable and financially savvy approach to oversee data furthermore applications run under such environments as the databases of them are logical structured and physical structured too. We can access the logical structure of the database without considering the physical nature of it. So, for the developed system, to maintain and access its database, the oracle database found the correct selection of the database for IDS.
5.3.4 Use of Smart Sniff Packet Capturing Tool

Smart Sniff allows us to capture TCP/IP parcels that pass through your system connector and perspective the caught information as succession of discussions in the middle of customers and servers monitoring software. We can get details of TCP/IP discussions in ASCII mode for conventions, in the same way as SMTP, HTTP, and POP3 what's more FTP or as hex dump structure for conventions, in the same way as DNS.

With the help of Smart Sniff, we can capture TCP/IP packets either any one of the following ways:

- **Raw Sockets** (Only for Windows 2000/XP or more noteworthy) This gimmick permits catching TCP/IP parcels on your system without introducing a catch driver.

- **Winpcap Capture Driver** It permits you to catch TCP/IP parcels on all Windows working frameworks. This system is by and large broadly used to catch TCP/IP bundles with Smart Sniff, and it works better than the past system.

- **Microsoft Network Monitor Driver (Only for Windows 2000/XP/2003)** Microsoft gives a free catch driver under Windows 2000/XP/2003 that can be utilized by Smart Sniff during its execution, but for this third method, driver is not introduced naturally, and we need to physically introduce it, in the recent past utilizing them and hence less preferred method for Smart Sniff operation.

5.3.5 Use of WEKA Version 3.6

As detailed given on WEKA (2011), WEKA tool is comprehensive benchmark software for machine learning (ML) and data mining algorithm evaluation as well as for data visualization. WEKA version 3.6 is used in the development system for data visualization of the input dataset. Afonso Costa (2014) observed that main strength is available in the order zone, where all present ML methodologies have been actualized inside a clean, object-oriented Java class chain of importance. Also, Regression, Association Rules and clustering algorithms have been implemented for the data mining applications.
5.3.6 Use of StarUML (Open Source UML/MDA Platform) version 5.0.2.1570

StarUML is open source software available to develop fast, flexible, extensible, feature oriented, and freely available UML/MDA platform running on Win32 platform as specified on Star UML (2012). Mainly, StarUML software is used to build software modeling tool. It supports UML 2.0 *i.e.* the latest version of UML (Unified Modeling Language) standard of modeling managed by OMG (Object Management Group). It is used in the developed system to model the running aspects of the system in visualized form with the help of behavioral, interaction, implementation and collaboration diagrams to make its details more expressive and understandable for the reader.

5.4 Module wise Implementation

The main aim behind this system implementation was to give the single solution for known as well as unknown attack detection in the input network traffic data; including the prevention policy mechanisms for the immediate response once the any kind of attack is detected. Additionally, it was focused on such implementation which will be not deployment location specific as like traditional IDSs products.

To support the main objective of the system, the input data collected from the network traffic and or collected form the MIT/LL laboratory as KDD Cup 1999 Dataset, the data preprocessing module is implemented to get compatible data for the system and in next phases, the system is implemented in such a way that, with the use of this, it is possible to cover maximum attack spectrum in intrusion detection phase which is constraint with developed and current IDSs products they are implemented with specific detection capabilities *i.e.* either signature based or anomaly based and certain class of intrusion attacks *i.e.* Probe, DoS, U2R and R2L. Also, attack detection is reported thereafter based on the information provided by the system, one can monitor anomalous traffic and can apply the suggested prevention policy mechanism. With the help of rule generation module, system analysts can generate the rules for the attack for future intrusion detection. Even, based on *A priori* algorithm with its improved version, one can easily find out the correct and potential signature rules based on given input feature attributes for the rule generation. The influence of certain intrusion attack is also possible to calculate and can flag the anomaly even within the attacks classification. Finally, the visualization of all the processed files and results is
possible with the help of provision implemented in the system. The details of each system module implementation are given in depth in the next section.

5.4.1 Data Collection and Data Preprocessing

As stated earlier, the data used for the developed system is either online captured for existing local network or offline standard dataset obtained from the internet to evaluate the system. To capture the online packet traffic data, the packet software is used. Additional details can be found on Smart Sniff (2014).

The following Figure 5.1 shows the main window of Smart Sniff with the captured packet details.

![Smart Sniff Packet Capturing Tool Main Window](image)

**Figure 5.1 Smart Sniff Packet Capturing Tool Main Window**

It is found that, this Smart Sniff is generally used to capture the packet details of TCP/IP Connections including packet features like Each Packet identifier, as specified by Cyber News and Reviews (1995) Local host Address, Protocol, Remote host Address, Local host Port, Remote host Port, Service Name, Packets, etc. So, it does not give the entire packet feature details required for the further analysis, only provides certain packet features captured by the Smart Sniff which are the subset of this entire packet feature set. The following Figure 5.2 gives the implementation flow used for this operation drawn with the help of Star UML software (Even all subsequent diagrams are modeled with same modeling software).

On the other hand if we observed the standard dataset obtained from the internet through the web portal of kdd.ics.uci.edu, KDD Cup 1999 dataset contains 41 features for the captured data for three elements of the packet data *i.e.* Basic TCP/IP Features, Traffic Features and Content Features as shown in the following Figure 5.3 where each row contains details about one captured packet with 41 features values.
Lu, Nannan (2013) concluded that; On comparison of this online captured data and KDD Cup 1999 data, it is clear that, online data feature set is identified as part of KDD Cup 1999 feature set. Even, the dataset size can be controlled with online capturing tool by running it for certain amount of time.

![Figure 5.2 Online Packet Capturing with Smart Sniff in Online Mode of System Operation](image)

Figure 5.2 Online Packet Capturing with Smart Sniff in Online Mode of System Operation

---

<table>
<thead>
<tr>
<th>Data</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcp</td>
<td>smtp</td>
</tr>
<tr>
<td>829,327,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,1,0,8,1,13,0,8,8,0,25,0,12,0,02,0,0,0,0</td>
<td></td>
</tr>
<tr>
<td>udp</td>
<td>private</td>
</tr>
<tr>
<td>105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,1,0,0,255,253,0,99,0,01,0,0,0,0,0,0</td>
<td></td>
</tr>
<tr>
<td>tcp</td>
<td>ftp_data</td>
</tr>
<tr>
<td>19,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,5,1,0,226,46,0,19,0,03,0,19,0,04,0,0,0</td>
<td></td>
</tr>
<tr>
<td>udp</td>
<td>private</td>
</tr>
<tr>
<td>105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,255,254,1,0,1,0,0,0,0,0</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.3 Standard Dataset KDD Cup 1999 Dataset details in unprocessed form

Mahbod Tavallaee (2009)
However, the KDD Cup 1999 dataset contains large sized dataset collection for IDS evaluation. The following Figure 5.4 shows the steps followed for the offline mode of system operation to get testing and training dataset files for the system evaluation.

![Figure 5.4. Offline Dataset Collection for System Evaluation](image)

Therefore, module for the data preprocessing is designed and constructed in such manner, user of the system can able to select only those features for the input file for them it supports. It means that, in the Figure 5.1, the Smart Sniff window shows the 08 valid packet features of TCP/IP Basic Connection in comparison with 41 features of KDD Cup 1999 standard Dataset (Which are illustrated in Chapter 4,Table 4.4 ), so developed system will give selection choice for the desired packet features. The following Figure 5.5(a) and (b), Sequence diagram drawn using Star UML tool gives the internal representation of the developed data preprocessing module in step wise execution.
Figure 5.5 (a) Online Captured Data Preprocessing Strategy Details

Figure 5.5 (b) Offline Standard Data Preprocessing Strategy Details

Following are the major reasons behind this data preprocessing requirement for the developed system as well as it highlights the various techniques employed for the data preprocessing.
The data obtained in real world data are generally incomplete \textit{i.e.} lacking with attribute qualities, failing to offer certain characteristics of client investment, or containing just at some point total information; boisterous means containing blunders or exceptions in the data; inconsistent data contains containing discrepancies in codes or names.

Common Techniques used in data preprocessing are Data Cleaning, Data Integration, Data Integration, Data Reduction and Data Discretization which are explained in short in below section.

- **Data Cleaning:**

  It is done with the help of fill in missing qualities, smooth loud information, distinguish or evacuate exceptions, and resolve inconsistencies in the data information.

- **Information Coordination**

  It is performed using multiple databases, information, or records to keep integrated properly.

- **Information Change**

  It does standardization and conglomereration of the information.

- **Information Diminishment**

  Dr. Bernard Chen (2009) explored as it is used for reducing the volume yet creating the same or comparable scientific results.

- **Information Discretization**

  It is a piece of information decrease and replaces numerical attributes with nominal ones.

**5.4.2 Intrusion Detection Phase**

As described in earlier in chapter 4, signature based interruption recognition took after by abnormality based interruption recognition. The execution of the system is carried out for the both types of intrusion detection \textit{i.e.} known attacks and unknown attacks.

To perform the signature based intrusion detection, system has considered the training dataset available from the portal of the KDD Cup 1999 dataset as well as based on this available training dataset, we have been generated own training dataset files for 04 attack
types and their 22 sub attack types too. The following Figure 5.6 shows the representation of training files contents for sub attacks (i.e. back, land) of DoS attack intrusion detection. Similarly, it is observed that, in original training dataset file, the attack signatures are get duplicated number of times and hence, we generated only distinct signature file for each sub attack by removing the duplicate values from the available training dataset. In original training dataset file, each tuple of the dataset shows the 41 features followed by attack label as 42nd value in each tuple and we have been done implementation in such a way that, analyst or user of the system can select their own choice training file and can separately store the attack signatures in specific attack files as given below. In the Figure 5.6, it shows the abstract view of each attack training file with their most relevant features (shown in Figure 5.6 (a) and (b) ) rather than showing all the 41 feature attribute values due to space constraint.

<table>
<thead>
<tr>
<th>Duration</th>
<th>protocol_type</th>
<th>Service</th>
<th>Flag</th>
<th>src_bytes</th>
<th>dst_bytes</th>
<th>land</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Tcp</td>
<td>http</td>
<td>SF</td>
<td>54540</td>
<td>8314</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>Tcp</td>
<td>http</td>
<td>SF</td>
<td>54540</td>
<td>8314</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>Tcp</td>
<td>http</td>
<td>SF</td>
<td>54540</td>
<td>8314</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>Tcp</td>
<td>http</td>
<td>SF</td>
<td>54540</td>
<td>8314</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>Tcp</td>
<td>http</td>
<td>SF</td>
<td>54540</td>
<td>8314</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>Tcp</td>
<td>http</td>
<td>SF</td>
<td>54540</td>
<td>8314</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>Tcp</td>
<td>http</td>
<td>SF</td>
<td>54540</td>
<td>8314</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>Tcp</td>
<td>http</td>
<td>SF</td>
<td>54540</td>
<td>8314</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>Tcp</td>
<td>http</td>
<td>SF</td>
<td>54540</td>
<td>8314</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>Tcp</td>
<td>http</td>
<td>SF</td>
<td>54540</td>
<td>8314</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>Tcp</td>
<td>http</td>
<td>SF</td>
<td>54540</td>
<td>8314</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 5.6 (a) Training File Prepared for Back Attack Mahbod Tavallaee (2009), NSL – KDD Data Set (2010)
<table>
<thead>
<tr>
<th>Duration</th>
<th>protocol_type</th>
<th>service</th>
<th>flag</th>
<th>src_bytes</th>
<th>dst_bytes</th>
<th>land</th>
<th>-------</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Tcp</td>
<td>finger</td>
<td>S0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-------</td>
</tr>
<tr>
<td>0</td>
<td>Tcp</td>
<td>finger</td>
<td>S0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-------</td>
</tr>
<tr>
<td>0</td>
<td>Tcp</td>
<td>finger</td>
<td>S0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-------</td>
</tr>
<tr>
<td>0</td>
<td>Tcp</td>
<td>finger</td>
<td>S0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-------</td>
</tr>
<tr>
<td>0</td>
<td>Tcp</td>
<td>finger</td>
<td>S0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-------</td>
</tr>
</tbody>
</table>

Figure 5.6 (b) Training File Prepared for Land Attack Mahbod Tavallaee (2009), NSL – KDD Data Set (2010)

After the preparation of the training file as per the intrusion identification need, end user can utilize these training files to do the signature based intrusion detection as well as anomaly detection with respect to the variation occurred into certain packet data feature attribute value than specified value into the training data file which is not analyzed by signature based intrusion detection and considered as undetected and normal traffic in the first phase of the detection. The following Figure 5.7 shows steps carried out for the signature based intrusion detection.
The developed system consists of alternative ways for the signature generation for the intrusion detection.

It consists of manual rule generation based on the input given by the analysts or the users for the signature generation from the given packet feature attributes selected by the analysts or user. It is really helpful for the system analysts to get quickly the signature generation without considering the format used by existing signature based systems like SNORT. This generated rule for the signature can directly used in the developed system, also can be applied to rule based IDSs with or without minor changes in the generated rule. The whole implementation procedure for the manual signature generation module is shown in the Figure 5.8.
Additionally, the use of Apriori algorithm with modification is introduced to provide the rule generation based on the threshold values of support and confidence. It is useful to get only the potential signature rule rather than getting the unwanted and redundant rules for the signatures fro other ways. Also, it is easily possible to get the potential packet feature attributes and possible associations of such feature attributes to generate the new signature rules for the future attacks. It is really new way to check possibilities of signature generation rather than using existing available rule sets with rule based IDSs. The steps for this kind of rule generation is shown in the Figure 5.9.
Lastly, once the signature based intrusion detection is performed, the anomaly detection is possible for the input data of the network traffic. The deviation in the field of any feature attributes of the packet is also considered as anomaly along with whole normal profile deviation of the input data. During this implementation, it is noticed that the total 41 features of the packet data are classified into three types of feature classifications i.e. Basic TCP/IP Connection Features, Traffic Features and Packet Content Features. So, each time checking all the values for the input dataset is time consuming and false positive task. Also, recall that, for online captured data, it does not contain all 41 features every time in their data file; they contains specific feature attributes like either Basic TCP/IP Connection Features, Traffic Features or Content Features only. So, the implementation of the system is made with the flexibility to check the anomaly at each of these specified packet feature levels as well as for all 41 features too. For identifying the anomaly at each feature attribute level, the extensive study for all the basic four attack types i.e. DoS, Probe, R2L, U2R and their sub types is carried out and then this anomaly detection module is implemented. Following Figure 5.10 depicts the same scenario of the implemented system for the packet feature attribute anomaly detection. Similarly, the anomaly detection for basic TCP/IP connection features, traffic features and packet content features is illustrated in Figure 5.11, 5.12 and 5.13 respectively.
Figure 5.10  Anomaly Detection for the Input Network Traffic
Figure 5.11 Anomaly Detection in Basic TCP/IP Connection Features of Packet

Figure 5.12 Anomaly Detection in Traffic Features of Packet
Another alternative for the potential attack anomaly detection is introduced based on the percentage of each major attack type detection. In this case, the initial attack test file is modified and new attack data with changed packet feature attributes is injected in this file and then the intrusion detection phase module is executed to get known attack detection with matching of modified test file with the training file as well as unknown attack detection based on the variation of attack signature injected in the test file as an anomaly. The cumulative percentage of each major attack is then calculated based on each attack sub types detected for that respective major attack. The following Table 5.1 gives the details of sub attack and their contribution in the major attack as part of them.

**Table 5.1 List of Major Attacks with their Sub Attack Types**

<table>
<thead>
<tr>
<th>Attack</th>
<th>Sub Attack Types</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.13 Anomaly Detection in Content Features of Packet
<table>
<thead>
<tr>
<th>Attack Type</th>
<th>Attack Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probe</td>
<td>ipsweep, portsweep, satan, nmap</td>
</tr>
<tr>
<td>DoS</td>
<td>back, land, Neptune, pod, smurf, teardrop</td>
</tr>
<tr>
<td>U2R</td>
<td>loadmodule, perl, buffer_overflow, rootkit</td>
</tr>
<tr>
<td>R2L</td>
<td>ftp_write, imap, phf, guess_passwd, multihop, spy, warezclient, warezmaster</td>
</tr>
</tbody>
</table>

It means that, to get detection percentage of Probe attack, one need to do sum of the percentage of detection of these sub attack types i.e. detection percentage of ipsweep, portsweep, satan and nmap and same way, the other attack detection percentage can be calculated as per the given equations.

\[
\text{% of Probe Attack Detection} = \sum \text{% of attack detection (ipsweep, portsweep, satan, nmap)}
\]

\[
\text{% of DoS Attack Detection} = \sum \text{% of attack detection (back, land, neptune, pod, smurf, teardrop)}
\]

\[
\text{% of U2R Attack Detection} = \sum \text{% of attack detection of (loadmodule, perl, buffer_overflow, rootkit)}
\]

\[
\text{% of R2L Attack Detection} = \sum \text{% of attack detection of (ftp_write, imap, phf, guess_passwd, multihop, spy, warezclient, warezmaster)}
\]

Once the percentage of each major attack is found, then based on certain threshold value, it is possible to highlight highly potential major attack types out of four attacks and then those kinds of alerts for these maximum devaited and penetrated attacks will be generated for further preventions. In this way, the anomaly detection is implemented in the developed system. The following Figure 5.14 shows the details of this anomaly detection of the implemented system.
Figure 5.14 Potential Attack Identification based on Attack Anomaly

Even the following Figure 5.15 and 5.16 gives more details about the New Signature generation module based on Detected Anomaly in the anomaly based IDS module implementation of the system.
Figure 5.15 New Signature Generation Based on Detected Anomaly
Figure 5.16 New Signature Generation Based on Detected Anomaly Attack Threshold

5.4.3 Prevention Phase

Since, during the implementation of the integrated IDS, after each certain intrusion detection, the prevention strategies can be applied by network administrators or users of the system based on generated alerts from the system. So, this phase implementation is actually implemented within the intrusion detection phase implementation itself. This section gives the details of each prevention policy mechanisms in short for their understanding and applications for the readers.

Before focusing on theoretical details, for each prevention mechanism, the following Figure 5.17 represents the use of prevention phase in the implemented system. Though, in the various prevention policies are shown one by one, it is not necessary for the system analysts or user of the system to use them at a time. Sometime, the use of individual prevention policy is enough for the protection. The following section gives the details of each prevention policy
which are the part of this new responsive mechanism introduced as part of IDS which is not available as integrated part of any existing and traditional IDSs Products in the commercial market. Also, because of this IDS will not be just for intrusion detection, but it provides another layer of protection within the single IDS itself. Hence, it can be concluded that, it is dual layer of protection for the information infrastructure for any organization.

Figure 5.17 Uses of Prevention Policy Mechanism in the Developed System

5.4.4 Theoretical Basis for Use of Sentiment Analysis

It is the last part of this developed system which is to be discussed as part of the research work objective. The main aim behind this sentiment analysis study was to see can we use the merits of sentiment analysis techniques to perform the better false positive and false negative attack signature classification. Before discussing this tasks in detail, the following section illustrates the purpose of sentiment analysis in the real world as significant study has been
done in the research field of the sentiment analysis and it is huge need in the decision making process for individual, society and in business organizations too.

5.4.4.1 Introduction

As described in earlier section, it is clear that user shares large the amount of information on social media and on its various elements like, wikis, forums, blogs, etc., and even it is described earlier, doing sentiment analysis or opinion mining of such huge amount of data is really challenging task for human being. Hence, there is huge need to automate sentiment analysis in today’s growing business and competitive world. In the earlier research work, this sentiment analysis problem is implemented and evaluated based on machine learning approaches and are found typically inaccurate and less trusted. Considering simple task of separating positive opinions from negative opinion sentiments on social media sites, many existing automated solutions only perform with around 80% accuracy and it is major concern for believing on opinion mining systems.

5.4.4.2 Current Status, Challenges in Sentiment Analysis

The significant study has been done on sentiment analysis in recent years as illustrated and researched extensively by Bing Liu (2010). Additionally, sentiment expression is a mixture of the structures of our utterances and the assumed knowledge of the people who we are addressing the sentiment on object. In comparison with the concept like topic identification, identifying and categorizing phrases like “good”, “bad” and “I like” is really formidable task.

When one wants to use machine learning algorithms in sentiment analysis then one has to understand deeply the context. For better understanding, consider the following examples

- It was nice hotel.
- It should have been a good hotel.
- The hotel was good in that it will make all future dinners appear to be tastier.
- In spite of an average experience I can't help the numerous surveys that says good hotel.
In this case, first sentence have been positive, however the rest are all speak to negative opinion, regardless of having the first and foremost sentence considered with them. The second sentence containing—should have been demonstrates the wanted result, leaving the real feeling expressed. The third represent the more complicated, indicates are hard for machines to distinguish. Indeed an individual may understand this as positive when perusing rapidly. The last and fourth explanation is much more muddled in light of the fact that the supposition communicated is negative, however it began with the powerless positive opinion—pleasant experience. Furthermore, closes by presuming that numerous other individuals communicated positive slant.

Hence, it seeks the learning calculations to comprehend dialect at an extremely deepest level. The above illustrated criticisms with qualifiers to be extremely gracious (even it is seen on the web as well), and all the more by and large, individuals used to be more inventive by the way they choose slants to depict the things that they dislike and want to express it via opinions in smart ways. This gives real challenge for opinion mining systems to express negative sentiments and particularly extreme for machine learning since machine adapting commonly meets expectations by gaining from example given as input to these algorithms. Additionally, ambiguity in expressed opinions is another challenge for opinion mining. Abstractions like given in statement 4 of above illustrated example are supported this ambiguity contents. So, these socially expressed negations get particularly complicated and challenging when we work with different languages, as every language expresses it differently. Consider example, as in English language, we may consider something ‘little’ when it is large in practice. Within English language itself, we found cultural differences, like British English and US English. Following Table 5.2 shows the cultural difference of English language in its statement understanding by the respective users.

<table>
<thead>
<tr>
<th>WHAT THE BRITISH SAY</th>
<th>WHAT THE BRITISH MEAN</th>
<th>WHAT FOREIGNERS UNDERSTAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>I hear what you say</td>
<td>I disagree and do not want to discuss it further</td>
<td>He accepts my point of view</td>
</tr>
<tr>
<td>With the greatest respect</td>
<td>You are an idiot</td>
<td>He is listening to me</td>
</tr>
<tr>
<td>That’s not bad</td>
<td>That’s good</td>
<td>That’s poor</td>
</tr>
</tbody>
</table>
That is a very brave proposal | You are insane | He thinks I have courage
---|---|---
Quite good | A bit disappointing | Quite good
I would suggest | Do it or be prepared to justify yourself | Think about the idea, but do what you like
Oh, incidentally/ by the way | The primary purpose of our discussion is | That is not very important
I was a bit disappointed that | I am annoyed that | It doesn’t really matter
Very interesting | That is clearly nonsense | They are impressed
I’ll bear it in mind | I’ve forgotten it already | They will probably do it
I’m sure it’s my fault | It’s your fault | Why do they think it was their fault?
You must come for dinner | It’s not an invitation, I’m just being polite | I will get an invitation soon
I almost agree | I don’t agree at all | He’s not far from agreement
I only have a few minor comments | Please rewrite completely | He has found a few typos
Could we consider some other options | I don’t like your idea | They have not yet decided

Though, US and somewhere else, it is generally simple for machine learning methodologies to comprehend that a —disappointed implies —annoyed after the calculations have seen a couple of such cases of notions. The testing point is to know how to apply British English dialect' criteria. In this way, this is the most indispensable next venture in notion examination frameworks where naturally recognizing what sort of investigation to be connected, contingent upon the class, dialect or wellspring of the articulation of the words. So, social and social connection of the articulation, one can put more intelligent presumptions about the accepted the learning of assumption holder and all the more precisely find the assessment forecasts to particular sorts of correspondence.
Similarly, in intrusion detection field, the interpretation of the terms like false positive, false negative, etc. is understand and interpreted by the users of the system, so there is need to bring out uniformity for such terms with proper identification based on collected detected attack database results.

5.4.4.3 Techniques Used In Sentiment Analysis Similar to Intrusion Detection

There are two main approaches used in sentiment classification, namely, the knowledge based and supervised machine learning approach. In the knowledge-based approach, the predefined dictionaries of opinion words are used to search the input words and find its effects to define the orientation of the sentiments towards the opinion object. Whereas, in supervised machine learning approach uses a trained statistical classifier for sentiment classification. The trained classifiers are used to predict the sentiment orientation of input documents. Both of these approaches are mainly dependent on affective vocabularies although its use is different. Following section introduces the supported machine learning algorithms in the field of sentiment analysis as like intrusion detection.

Commonly Used Machine Learning Models in Sentiment Analysis

Different supervised categorization algorithms have been utilized so far for polarity classification tasks in the sentiment analysis. Most commonly used methods and algorithms are like, Support Vector Machine (SVM), Naïve Bayesian Classifiers, etc.

On other hand, methods like Maximum Entropy, Decision Tree, Neural Network, Latent Dirichlet Allocation (LDA), and Probability Latent Semantic Analysis (PLSA) are also used for sentiment analysis or opinion mining (OM). Authors of the performed work have selected 336 related research papers, categorized the papers according to the use of Machine Learning Algorithms in the research field as shown in Table 5.3. In this survey, it is found that use of SVM is increasing, while Naïve Bayesian is also consistently used for the sentiment analysis as shown in Figure 5.18. OM field is facing several different challenges, such as how to determine which segment of text is opinionated, identify the opinion holder, and measure the positive or negative strength of opinion. OM is also deals with the concerns about the human reviews, emotions and sentimental discussion. Everyone has their own perception and concern about a particular problem, issue, or topic. Opinionated text may be fake, irrelevant and or ambiguous information.
Table 5.3: Usage of Machine Learning Techniques

<table>
<thead>
<tr>
<th>Year</th>
<th>Use of SVM</th>
<th>Use of NBA</th>
<th>Use of Other</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>2008</td>
<td>45</td>
<td>19</td>
<td>11</td>
<td>74</td>
</tr>
<tr>
<td>2007</td>
<td>32</td>
<td>23</td>
<td>17</td>
<td>72</td>
</tr>
<tr>
<td>2006</td>
<td>25</td>
<td>28</td>
<td>13</td>
<td>60</td>
</tr>
<tr>
<td>2005</td>
<td>11</td>
<td>9</td>
<td>35</td>
<td>55</td>
</tr>
<tr>
<td>2004</td>
<td>5</td>
<td>11</td>
<td>19</td>
<td>35</td>
</tr>
<tr>
<td>2003</td>
<td>6</td>
<td>9</td>
<td>8</td>
<td>23</td>
</tr>
<tr>
<td>2002</td>
<td>2</td>
<td>8</td>
<td>6</td>
<td>17</td>
</tr>
</tbody>
</table>

Figure 5.18 Machine Learning Classifiers
During the study of sentiment analysis, it is observed that, the issues which are the major part of sentiment analysis are also found in the intrusion detection field too. These issues are:

- **Authority**: It is an accepted source for the information or advice, either an expert on the subject or a persuasive force for the opinions.
- **Credibility**: It is a quality of opinion being believable, trustworthy as expressed and analyzed by opinion mining systems.
- **Spam**: Analysis of spam opinion to get exact opinion analysis.
- **Non Expert opinion**: Open forums and blogs are often suffered from non-expertise, since the easily available ways to express the opinion and not possible to control on it. Main issue of such opinion is they cannot provide review text in a proper manner as per the expectations of the opinion mining systems.
- **Domain Dependent**: Normally opinions are expressed on certain issue, problem, or topic. Therefore, the techniques used and available are normally domain specific which leads to the problem of non-generalization.
- **Language differences for sentiment expressions**: Different persons use different language context in their expressed opinions, even in English forum they may write in roman words of their own language, which makes the OM task difficult and challenging. For example, user can use the text to express opinion about the book written in roman English of Urdu language as “*ye book mujey passand hey, koink ye poori course ko cover karatha hey*”.
- **Effects of syntax used and semantics**: it affects in breaking multi-word expression, mapping of synonymous words into different relevant components, and words with multiple meaning as one single component in different context. Sentence document Complexity measurement, Contextual Sentiments identification, Heterogeneous documents separation, Reference Resolution in sentiments, use of modal operators like might, could, and should are still challenging problems in this area.
- **Effect of sense on terms**, finding subjective terms, and multi-word document analysis also major concerns in sentiment analysis.
5.4.4.4 Conclusion

The key issues in sentiment analysis are authority, credibility, spam detection, language difference, non-expert opinion, domain dependency, effect of syntax on semantic are also countable in the field of interruption discovery. Primarily, the spam location is more similar and challenging part of sentiment analysis with the intrusion detection. So, the reduction in spam enhances the trust of people on sentiment analysis field and here, the same issue is taken in account to reduce the multiple occurrences of similar intrusion attacks and reduction in spending more time to do analysis of same or similar attack detection and prevention. The techniques available to perform the spam detection in the sentiment analysis field can be applied to the outputs generated by Intrusion Detection Systems to reduced their database and as well, it will reduce the space requirement concern for growing attack data day by day in the intrusion detection. This research work gives this theoretical basis for the use of sentiment analysis in the field of intrusion detection.

5.4.5 Summary

In this, it covers the implementation description of the developed system which includes the identified major system requirements of the developed system, list of the software used and their details during the implementation, module wise implementation details description. The major reasons behind these contents are illustrated.

The prime objective behind this chapter was to make aware the readers about the contribution of the research work and how it is helpful to achieve the objective set for the proposed research work. Before this work, the individual implementations and simulations for Misuse dependent IDS and Deviation based IDS detection methodologies are studied and found their merits and demerits. Additionally, identified how the integration of these intrusion detection methodologies is possible and what are the earlier research work were carried out in the proposed research work. It is found; these methodologies rarely mixed up together. But the proposed research work integrated these detection technologies altogether by considering the merits of each of the intrusion detection methods.

The main advantage of this integration lies in the support of flexibility and extensibility attributes of any IDS development. Because of this integration, the attacks and their maximum attack coverage is possible. So, all major attacks like Probe, DoS, R2L and
U2R along with their sub attack types can be detected by this module of the developed system. Moreover, with the same integrated module implementation, third detection methodology for DoS attacks is also described in the 5.4.2 section.

Additionally, it is the system where the immediate response for the detection attack is provided with the help of prevention policies provided for the network administrators of the IDS. So, implemented IDS in the research work found advanced in this nature and no previous attempt is found in this sense of IDS and IPS fusion. Also, the given prevention policies are quite flexible for their use, so the user of the system can apply them as per the need to prevent the similar attack penetrations into the host or network of business organization.

Finally, the use of sentiment analysis techniques in intrusion detection field is introduced with the theoretical basis to reduce the detected attacks and their signature database by removing false positives and it will prevent system administrators from doing unwanted and time consuming attack analysis rather than doing the monitoring and managing the networks more securely.