Intelligent Intrusion Detection System

Rahul Goyal
(101287)

Project Guide
Dr. Pardeep Kumar
Assistant Professor (Senior Grade)

May, 2014

Submitted in partial fulfilment of the Degree of
Bachelor of Technology

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

Table of Content

List of Figures

Certificate

This is to certify that the work titled **— "Intelligent Intusion Detection System***"*, submitted by "**Rahul Goyal**" partial fulfillment for the award of degree of Bachelor of Technology in Computer Science Engineering to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Signature of Supervisor ……………………..

Name of Supervisor **Dr. Pardeep Kumar**

Designation **Assistant Professor (Senior Grade)**

Date ……………………..

Acknowledgement

As I conclude our project with the God's grace, I have many people to thank for all the help, guidance and support they lent to me, throughout the course of my endeavor.

First and foremost, I am sincerely thankful to Dr. Pardeep, my Project Guide, who has always encouraged me to put in my best efforts and deliver a quality and professional output. His methodology of making the system strong from inside has taught me that output is not the END of project. We really thank him for his time & efforts.

I am deeply indebted to all those who provided reviews and suggestions for improving the materials and topics covered in my project, and I extend our apologies to anyone I may have failed to mention.

Rahul Goyal
Enrollment no. 101287
CSE
Signature ……………………..

Summary

In today’s world it is very important to sustain a solid security to ensure that communication of information between various organizations is safe and trusted. But there is always a threat of intrusion and misuse of secured data transfer over internet and network. Thus Intrusion Detection Systems has become a must and crucial component in the area of network security. A lot off techniques have being utilized in intrusion detections, but all of the systems implemented so far are not completely perfect. So, the quest of improvement continues. In this progression, here we propose an Intrusion Detection System (IDS), with the network security for an organisation. It is to develop a System for a network, which will study the hacker’s behaviour and store it in a Data Base.

Signature of Student	Signature of Supervisor
Name: Rahul Goyal	Name: Dr Pardeep Kumar
Date	Date

CHAPTER 1

1 INTRODUCTION

1.1 Intrusion Detection System

An intrusion detection system (IDS) is an active process or device that analyzes system and network activity for unauthorized entry and/or malicious activity. The way that an IDS detects anomalies can vary widely; however, the ultimate aim of any IDS is to catch the intruder or perpetrators in the act before they do real damage to resources. In other words it is a device or software application that monitors network or system activities for malicious activities or policy violations and produces reports to a management station.

An IDS protects a system from attack, misuse, and compromise. It can also monitor network activity, audit network and system configurations for vulnerabilities, analyze data integrity, and more.

The main differences between IDS and IPS are, unlike intrusion detection systems, intrusion prevention systems are placed in-line and are able to actively prevent/block intrusions that are detected.

1.2 Types of Intrusion Detection System

1.2.1 Signature based detection System

It is also known as misuse detection, the system analyzes the information it gathers and compares it to large databases of attack signatures. Essentially, the IDS look for a specific attack that has already been documented. Like a virus detection system, misuse detection software is only as good as the database of attack signatures that it uses to compare packets against.

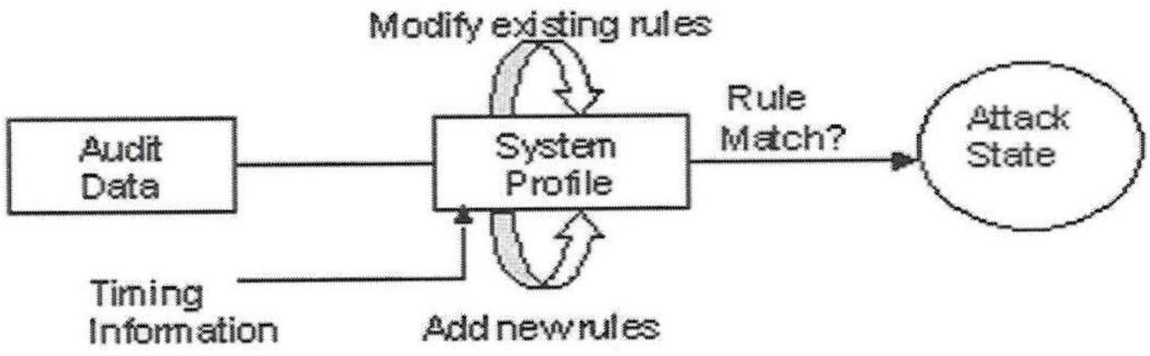


Fig 1.1 Signature Based intrusion Detection System

1.2.2 Anomaly Based detection System

In anomaly detection, the system administrator defines the baseline, or normal, state of the network traffic load, breakdown, protocol, and typical packet size. The anomaly detector monitors network segments to compare their state to the normal baseline and look for anomalies.

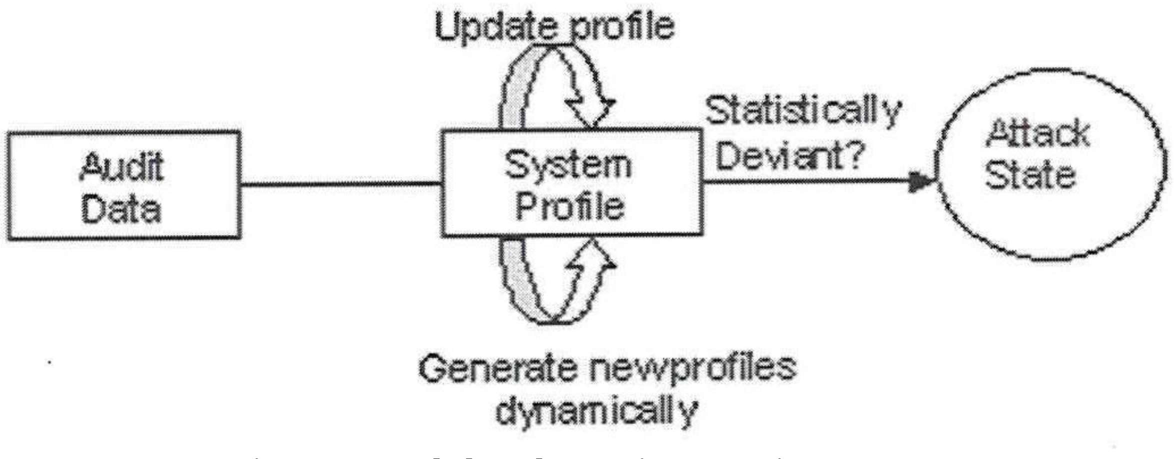


Fig 1.2 Anomaly based Intrusion Detection System

1.2.3 Host-based Detection System

A host-based intrusion detection system (HIDS) is an intrusion detection system that monitors and analyzes the internals of a computing system as well as (in some cases) the network packets on its network interfaces. A host-based IDS monitors all or parts of the dynamic behaviour and the state of a computer system. Besides such activities like dynamically inspect network packets targeted at this specific host, a HIDS might detect which program accesses what resources and discover that, for example, a word-processor has suddenly and inexplicably started modifying the system password database. Similarly a HIDS might look at the state of a system, its stored information, whether in RAM, in the file system, log files or elsewhere; and check that the contents of these appear as expected, e.g. have not been changed by intruders.

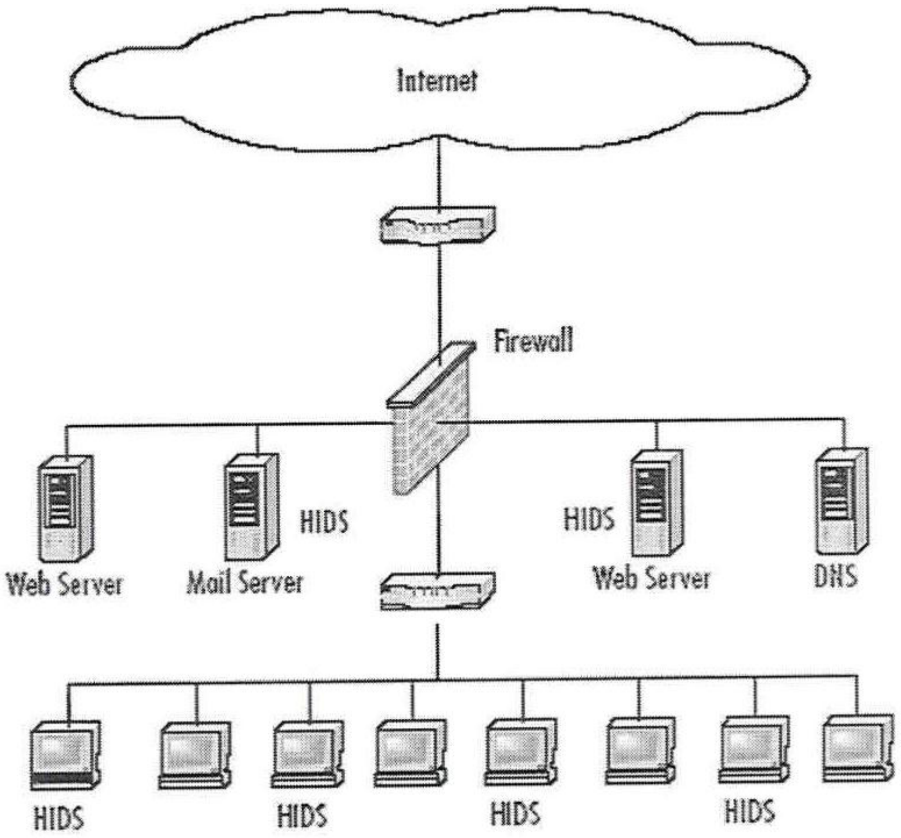


Fig 1.3 Host Based Intrusion Detection System

1.2.4 Network based Detection System

In a network-based system, or NIDS, the individual packets flowing through a network are analyzed. The NIDS can detect malicious packets that are designed to be overlooked by a firewall as simplistic filtering rules

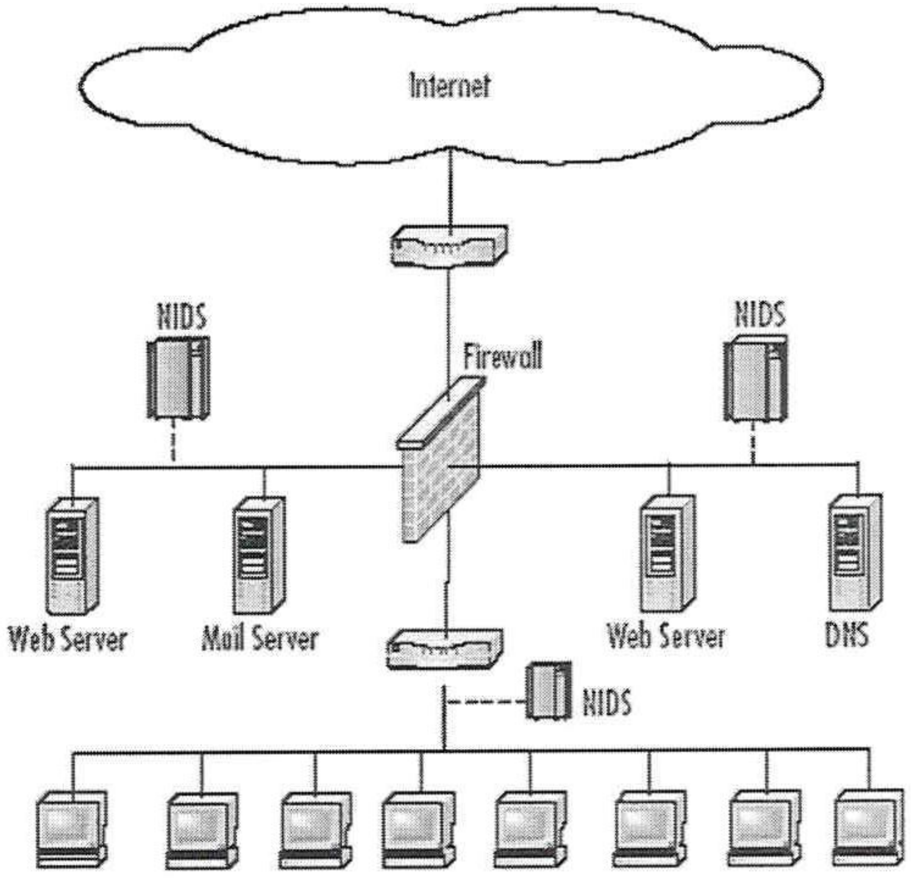


Fig 1.4 Network Based Intrusion Detection System

CHAPTER 2

2. LITRATURE SURVEY

2.1 Existing Intrusion Detection Systems

Snort: A free and open source network intrusion detection and prevention systems, was created by Martin Roesch in 1998 and now developed by Sourcefire. Snort entered InfoWorld's Open Source Hall of Fame as one of the “greatest open source software”. Snort catches port scans, thousands of worms and other suspicious behaviour by using techniques like protocol analysis, various pre-processors and content searching.

OSSEC: An open source host-based intrusion detection systems, performs various task like root kit detection, integrity checking, log analysis, active response and time-based alerting. In addition to its IDS functionality, it is commonly used for a SEM/SIM solution. Because of its powerful log analysis engine most of the universities and organisation are running OSSEC HIDS to monitor and analyse their firewalls logs.

OSSIM: The main goal of Open Source Security Information Management, it is used to provide a comprehensive compilation of tools which, when working all together, grant network/security administrators with a detailed view over each and every aspect of networks, hosts, physical access devices, and servers.

Bro: An open-source, Unix-based network intrusion detection system. Bro detects intrusions by first extracting its application-level semantics from network traffic and then executing event-oriented analysers that compare the activity with patterns deemed troublesome.

Fragroute/Fragrouter: A network intrusion detection evasion toolkit. Fragrouter helps an attacker launch IP-based attacks while avoiding detection.

BASE: The Basic Analysis and Security Engine, BASE is a PHP-based analysis engine technique to search and process a database of security events generated by various IDSs, firewalls and network monitoring tools.

Sguil: Sguil is built by network security analysts for network security analysts. Its main component is an intuitive GUI that provides real-time events from Snort/barnyard. It also includes many other components which facilitate the practice of network security and event driven analysis monitoring of IDS alerts.

2.2 Problems with Existing Systems

Most of the intrusion detection systems suffer from at least two of the following problems:

First, the information to be used by the intrusion detection system is obtained from audit trails or from packets on a network. Audit data has to traverse a longer path from its origin to the IDS and in the process can potentially be destroyed or modified by an attacker. In addition, the intrusion detection system has to infer the behaviour of the system from the data collected by the system, which can result in missed events or misinterpretations. This is known as the *fidelity* problem.

Second, the intrusion detection system continuously uses additional resources in the system even when there are no intrusions occurring while monitoring, because some of the components of the intrusion detection system have to be running all the time. This is the *resource wastage* problem.

Third, because the components of the intrusion detection system are implemented as separate programs, they are vulnerable to tampering. An intruder can easily disable or modify the programs running on a system, leaving the intrusion detection system useless or unreliable and this is the *reliability* problem.

2.3 Usual Components of Intrusion Detection System

An intrusion detection system normally consists of three functional components [8].

The first component of an intrusion detection system is a ***data source***. Data sources can be divided into four categories namely Network-based monitors, Host-based monitors, Application-based monitors and Target-based monitors.

The second component of an intrusion detection system is known as the ***analysis engine***. This type of component takes information from the data source and examines the data for symptoms of attacks or other violations. The analysis engine can use only one or both of the following analysis approaches:

Misuse/Signature-Based Detection: This type of detection engine detects intrusions that follow well-known patterns of attacks (or signatures) that exploit known software vulnerabilities [9] [10]. The main limitation of this approach is that it only looks for the known weaknesses and may not care about detecting unknown future intrusions [11].

Anomaly/Statistical Detection: An anomaly based detection engine will search for something rare or unusual [11]. They analyse the system for event streams, using statistical methods to find patterns of activity that appear to be abnormal. The main disadvantages of this system are that they are highly expensive and they can recognize an intrusive behaviour as normal behaviour because of insufficient data

The third and last component of an intrusion detection system is the ***response manager***. The response manager will only work when intrusion attacks are found on the machine, by informing user or system in the form of a response.

2.4 Types of Attacks

This section is an overview of the four major categories of network attacks. Every attack on a network can be placed into one of these groupings.

2.3.1 Denial of Service (DoS): A DoS attack is a type of attack in which the intruder or hacker makes a computer or memory resources too busy or too full to serve legitimate networking requests and hence denying real users access to a machine e.g. apache, smurf, ping of death, back, mail bomb, UDP storm etc. are example of DoS attacks.

2.3.2 Remote to User Attacks (R2L): A remote to user attack is type of an attack in which a user sends packets to a remote machine over the internet, which he or she does not have access to in order to expose the machines vulnerabilities and exploit privileges which a local user would have on the computer example are xlock, guest, xnsnoop , phf, sendmail dictionary etc.

2.3.3 User to Root Attacks (U2R): These attacks are exploitations attacks in which the hacker starts off on the system with a normal user account and attempts to abuse vulnerabilities in the system in order to gain administrator privileges example are perl, xterm.

2.3.4 Probing: Probing is an attack in which the hacker scans a machine or a networking device in order to determine weaknesses or vulnerabilities that may later be exploited so as to hack the system. This technique is commonly used in data mining example are saint, portsweep, mscan, nmap etc.

CHAPTER 3

3. Work Contribution

3.1 Introduction

The IDSN is a completely new system that replaces the old privilege based access system, where the boss had all the access. Here all the employees can be given access to company files which will be continuously monitored.

This system helps to develop a reasoned, proactive response to a threat and intrusions. In addition, they facilitate the building of contingencies thus contributing to the need to know what you don't know in good project management practices.

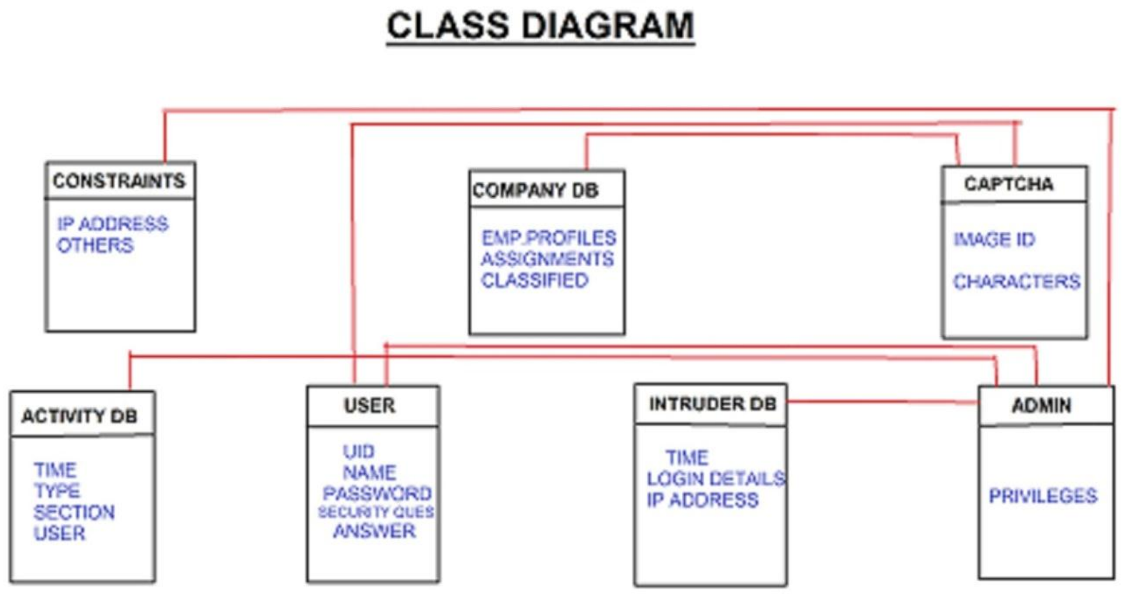


Fig 3.1

IDSN fits into the defensive plan as a way of

- Studying black hat activity.
- Developing a reasoned response to the threat
- Testing and developing new responses and tactics to a given threat
- This can also slow down another attack thus allowing time to develop a countermeasure.

3.2 Design and Implementation Constraints

The main problem that the developers will face while making this system will be memory. Continuous monitoring and parallel databases will make it very huge on memory consumption.

As the system involves monitoring and updating on a regular basis most of the maintenance work will be carried out by the developers

The system shall use the 2007 MS Access database engine**.** All Java code shall conform to the Java 7.1 standard. All scripts shall be written in Java

3.2.1 Implementation

The entire project was developed on netbeans using java swing and sql. The user interface uses various jFrames. This does not only make the implementation simpler but also makes it user friendly. There is a separate table for any activity, any intrusion, any constraint or the

Net beans platform

Net Beans is an integrated development environment (IDE) for developing primarily with Java, but also with other languages, in particular PHP, C/C++, and HTML5.] It is also an application platform framework for Java desktop applications and others.

The Net Beans IDE is written in Java and can run on Windows, OS X, Linux, Solaris and other platforms supporting a compatible JVM. The Net Beans Platform allows applications to be developed from a set of modular software components called *modules*. Applications based on the Net Beans Platform (including the Net Beans IDE itself) can be extended by third party developers.

Product Version: NetBeans IDE 6.9.1

Ms Access

Microsoft Access, also known as **Microsoft Office Access**, is a database management system from Microsoft that combines the relational Microsoft Jet Database Engine with a graphical user interface and software-development tools. It is a member of the Microsoft

Office suite of applications, included in the Professional and higher editions or sold separately.

Microsoft Access stores data in its own format based on the Access Jet Database Engine. It can also import or link directly to data stored in other applications and databases.

Product Version Ms Access 2007

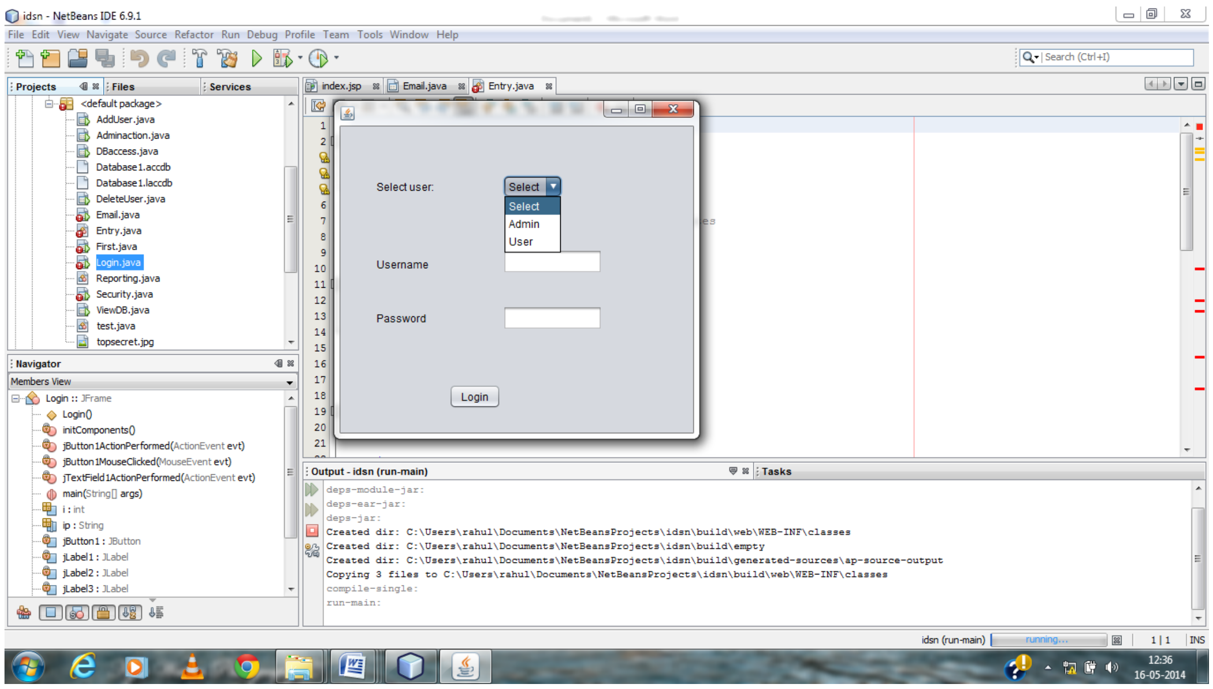
3.2.2 Issues

The admin does not have 3 chances to login. The very first incorrect login will generate an alert and give an entry in the intruder table. But since the admin himself handle the intruder alerts, he can always ignore the alert generated by his silly mistake.

Although the user have just 3 chances to login. After the 3 incorrect login users access ip address will be blocked and further ip will be saved in the database.

3.2.3 Designs

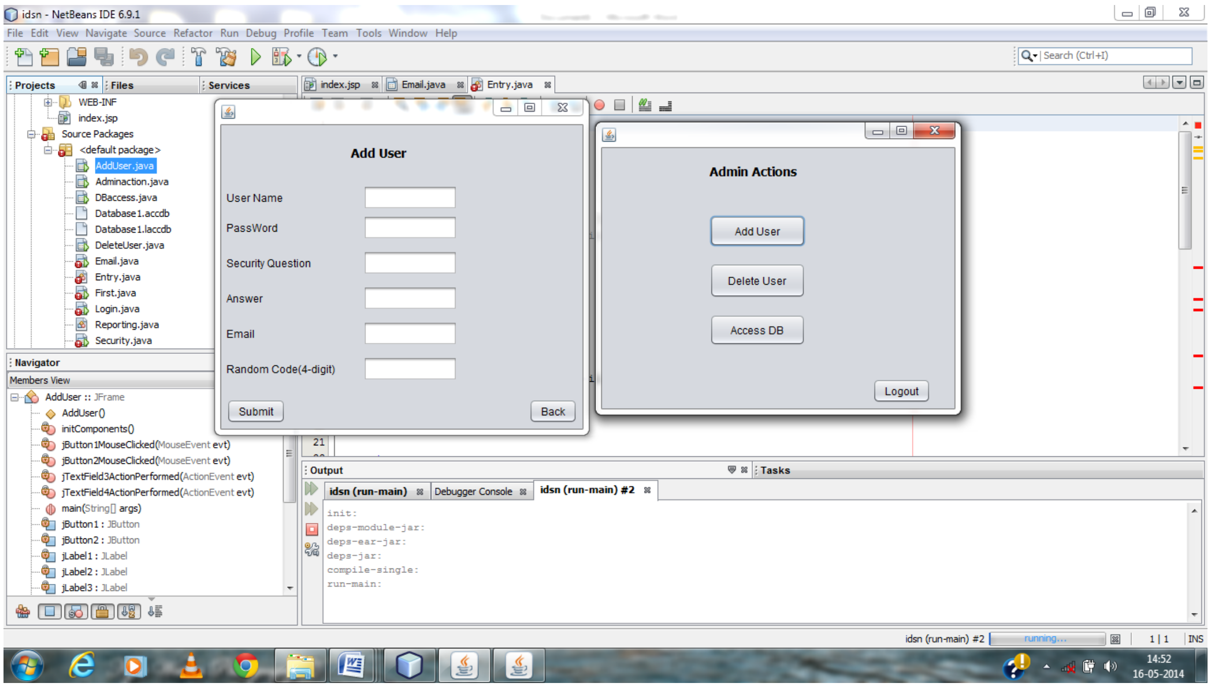
Steps of working projects.



```
deps-module-jar:
deps-ear-jar:
deps-jar:
Created dir: C:\Users\rahul\Documents\NetBeansProjects\idsn\build\web\WEB-INF\classes
Created dir: C:\Users\rahul\Documents\NetBeansProjects\idsn\build\empty
Created dir: C:\Users\rahul\Documents\NetBeansProjects\idsn\build\generated-sources\ap-source-output
Copying 3 files to C:\Users\rahul\Documents\NetBeansProjects\idsn\build\web\WEB-INF\classes
compile-single:
run-main:
```

Fig 3.2

It's the first step for this IDS system, initially is the login page. Both **user** and **admin** can login to have access to the confidential file with the system.

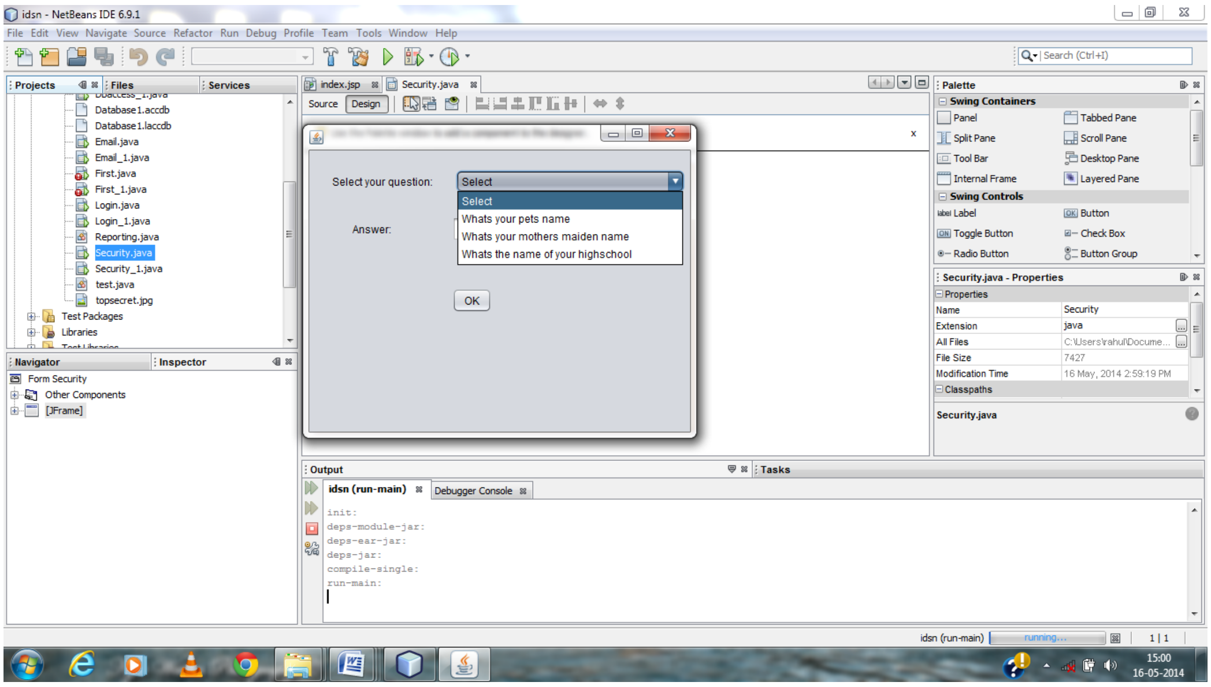


```
init:
deps-module-jar:
deps-ear-jar:
deps-jar:
compile-single:
run-main:
```

Fig 3.3

Admin can add the new user of the organization with all the details and also can delete the active member of the IDSN from accessing the top secret files.

Admin can also check the details for any intrusion by accessing the database as the system can block the intruder and save its detail.



Select your question: Select

- Select
- Whats your pets name
- Whats your mothers maiden name
- Whats the name of your highschool

Answer:

Security.java - Properties

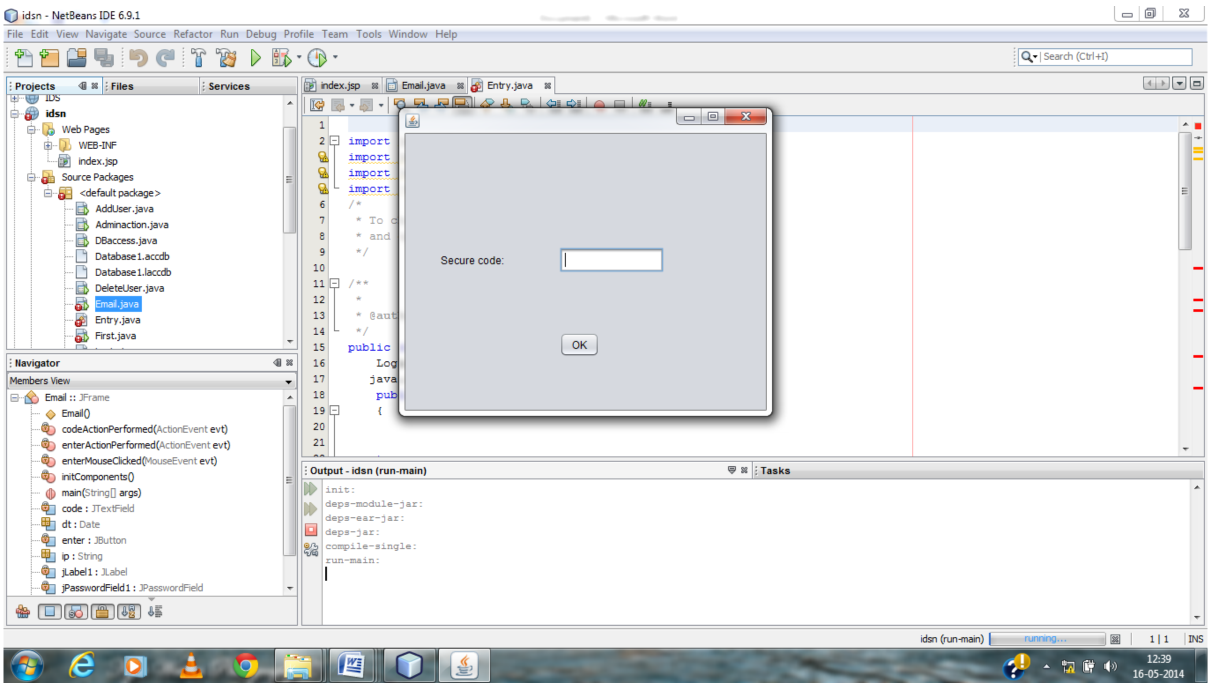
Name	Security
Extension	java
All Files	C:\Users\rahul\Docume...
File Size	7427
Modification Time	16 May, 2014 2:59:19 PM

```
init:
deps-module-jar:
deps-ear-jar:
deps-jar:
compile-single:
run-main:
```

Fig 3.4

For the user login it’s the second level of IDSN with the question confidentiality. All the answers are saved in the data base and can be accessed by the admin anytime.

Here in example the security question is “ what’s your pet name? “. Correct answer to this will give access to next level and will send a random 4 digit code to the user email id.



```
import
import
import
import
/*
 * To c
 * and
 */
/**
 *
 * @aut
 */
public
    Log
   java
    pub
   {
```

```
init:
deps-module-jar:
deps-ear-jar:
deps-jar:
compile-single:
run-main:
```

Fig 3.5

This figure shows the 3rd level security of IDSN system.

After the second level user will receive the security code on the email id which has been accessed from the user database. The valid code will allow the user to enter the secured system and have all the secret details or any of the confidential files.

All the 3 levels of IDSN can be just tried just 3 times i.e if the user is trying to access with wrong detail more than 3 times, user (intruder) will be blocked and its ip address will be saved in the data base for future use.

CHAPTER 4

FUTURE DIRECTIONS

4.1 Future Scope

In order to build an accurate (effective) base classifier, first we need to gather a accurate amount of training data and then identify a set of meaningful features from them. Both of the tasks require insight into the nature of the *audit data*, which can be very difficult without proper guidelines and tools. Here we have described some algorithms that can address these needs. We use the term "audit data" to refer to general data streams that have been properly processed for detection purposes. Thus further we can use data mining techniques on this data trails and identify the intrusion if any available for the system security.

This system can be implemented on the various routers and gateways with proper modifications and can make more secure system although the large database may be required.

Appendix

Login.java

```
import java.sql.*;
import javax.swing.*;
public class Login extends javax.swing.JFrame {
    static String user_name;
    int i=0;

    public Login() {
         i=0;
        initComponents();
    }
    @SuppressWarnings("unchecked")
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        user = new javax.swing.JComboBox();
        jTextField1 = new javax.swing.JTextField();
        jTextField2 = new javax.swing.JTextField();
        jButton1 = new javax.swing.JButton();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jLabel1.setText("Select user:");

        user.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Select",
"Admin", "User" }));

        jTextField1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jTextField1ActionPerformed(evt);
            }
        });

        jButton1.setText("Login");
        jButton1.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                jButton1MouseClicked(evt);
            }
        });
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
            jButton1ActionPerformed(evt);
          }
        });

        jLabel2.setText("Username");

        jLabel3.setText("Password");

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
          layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
              .addGroup(layout.createSequentialGroup()
                .addGap(41, 41, 41)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
                  .addComponent(jLabel1,     javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                  .addComponent(jLabel3,     javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                  .addComponent(jLabel2,     javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                .addGap(78, 78, 78)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
                  .addComponent(jTextField2,   javax.swing.GroupLayout.DEFAULT_SIZE,
113, Short.MAX_VALUE)
```

```
                    .addComponent(user,          javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jTextField1)))
                .addGroup(layout.createSequentialGroup()
                  .addGap(123, 123, 123)
                  .addComponent(jButton1)))
              .addContainerGap(111, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
          layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(layout.createSequentialGroup()
            .addGap(55, 55, 55)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel1)
                .addComponent(user,               javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(58, 58, 58)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                .addComponent(jLabel2,    javax.swing.GroupLayout.PREFERRED_SIZE,    22,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jTextField1,      javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(36, 36, 36)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
                .addComponent(jTextField2)
                .addComponent(jLabel3,          javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,   39,
Short.MAX_VALUE)
        .addComponent(jButton1)
        .addGap(27, 27, 27))
        );

        pack();
      }// </editor-fold>

      private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        try
        {
          Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        Connection conn = DriverManager.getConnection("jdbc:odbc:myDSN1");
          String sql;
          user_name=jTextField1.getText();
          if(user.getSelectedItem().toString().equals("User"))
          {
        sql="select * from Table1 where username='" +jTextField1.getText()+"' and
userpass='"+jTextField2.getText()+"'";
          }
          else
          {
        sql="select * from Table2 where adminname='" +jTextField1.getText()+"' and
adminpass='"+jTextField2.getText()+"'";
          }
          Statement st=conn.createStatement();
         ResultSet rs = st.executeQuery(sql);
           if(rs.next())
          {
```

```
                JOptionPane.showMessageDialog(null,"Logged in" );
                this.dispose();
                Security sc=new Security();
                sc.setVisible(true);

            }
            else
            {
                JOptionPane.showMessageDialog(null,"invalid Logged" );
                i++;
                if(i==3)
                {
                    System.out.println("Fuck you Bitch");

                }
                }

        }
        catch(Exception e)
        {

        }

    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

    }

    private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {
    }
    public static void main(String args[]) {

        try {
            for              (javax.swing.UIManager.LookAndFeelInfo             info             :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVE
RE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVE
RE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVE
RE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVE
RE, null, ex);
        }


        java.awt.EventQueue.invokeLater(new Runnable() {

            public void run() {
                new Login().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JTextField jTextField1;
    private javax.swing.JTextField jTextField2;
    private javax.swing.JComboBox user;
    // End of variables declaration
}
```

Security.java (second level)

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.JOptionPane;
public class Security extends javax.swing.JFrame {

    Login l =  new Login();
    public Security() {
        initComponents();
    }
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {
        jButton1 = new javax.swing.JButton();
        jTextField1 = new javax.swing.JTextField();
        jComboBox1 = new javax.swing.JComboBox();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jButton1.setText("OK");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

        jComboBox1.setModel(new javax.swing.DefaultComboBoxModel(new String[] {
"Select", "Whats your pets name", "Whats your mothers maiden name", "Whats the name of
your highschool" }));

        jLabel1.setText("Select your question:");

        jLabel2.setText("Answer:");

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(26, 26, 26)
                .addComponent(jLabel1)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 26,
Short.MAX_VALUE)
```

```
                .addComponent(jComboBox1,        javax.swing.GroupLayout.PREFERRED_SIZE,
260, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap())
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                .addGap(49, 49, 49)
                .addComponent(jLabel2)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jButton1)
                    .addComponent(jTextField1,       javax.swing.GroupLayout.PREFERRED_SIZE,
252, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                .addGap(22, 22, 22)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jComboBox1,    javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel1))
                .addGap(27, 27, 27)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jTextField1,       javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
                .addComponent(jLabel2))
            .addGap(53, 53, 53)
            .addComponent(jButton1)
            .addContainerGap(135, Short.MAX_VALUE))
    );

    pack();
}// </editor-fold>

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
      try
    {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    Connection conn = DriverManager.getConnection("jdbc:odbc:myDSN1");

        String sql1;



        sql1="select * from Table1 where username='"+Login.user_name+"' and secanswer='"+jTextField1.getText()+"' and secquestion='"+jComboBox1.getSelectedItem().toString()+"'";
        System.out.println(Login.user_name);

        Statement st=conn.createStatement();
      ResultSet rs1 = st.executeQuery(sql1);

         if(rs1.next())
        {
            JOptionPane.showMessageDialog(null,"Correct" );
            this.dispose();
            test t=new test();
            t.email();
            }
            else
            {
                JOptionPane.showMessageDialog(null,"Wrong question or answer" );
            }
        }
        catch(Exception e)
        {

        }
    }
    public static void main(String args[]) {

        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /*
         * If Nimbus (introduced in Java SE 6) is not available, stay with the
         * default look and feel. For details see
         * http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Security.class.getName()).log(java.util.logging.Level.SE
VERE, null, ex);
        } catch (InstantiationException ex) {
```

```
java.util.logging.Logger.getLogger(Security.class.getName()).log(java.util.logging.Level.SE
VERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Security.class.getName()).log(java.util.logging.Level.SE
VERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Security.class.getName()).log(java.util.logging.Level.SE
VERE, null, ex);
        }
        //</editor-fold>

        java.awt.EventQueue.invokeLater(new Runnable() {

            public void run() {
                new Security().setVisible(true);

            }

        });
    }
    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JComboBox jComboBox1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JTextField jTextField1;
    // End of variables declaration
}
```

Test.java (will send email and generate the random code for security)

```
import java.io.UnsupportedEncodingException;
import java.sql.*;
import java.util.Properties;

import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.PasswordAuthentication;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.AddressException;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;



public class test {
   static int randomNumber;
   Login l1=new Login();
   String em=new String();

   public void email() throws ClassNotFoundException, SQLException
   {


      final String username = "ojas.ag07@gmail.com";
      final String password = "arrahman";
      String email=null;

      int randomNumber = ( int )( Math.random() * 9999 );
      if( randomNumber <= 1000 ) {
      randomNumber = randomNumber + 1000;
      }
```

```
Properties props = new Properties();
props.put("mail.smtp.starttls.enable", "true");
props.put("mail.smtp.auth", "true");
props.put("mail.smtp.host", "smtp.gmail.com");
props.put("mail.smtp.port", "587");

Session session = Session.getInstance(props,
  new javax.mail.Authenticator() {
    protected PasswordAuthentication getPasswordAuthentication() {
        return new PasswordAuthentication(username, password);
    }
  });

try {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    Connection conn = DriverManager.getConnection("jdbc:odbc:myDSN1");
    String sql2;
    sql2="select * from Table1 where username='"+Login.user_name+"'";
    Statement st2=conn.createStatement();
    ResultSet rs=st2.executeQuery(sql2);
     if(rs.next())
  {
     em =String.valueOf(rs.getString(5));
System.out.println(em);
  }
   // System.out.println(sql2);
    //System.out.println(rs.next());

}
catch(Exception e)
{
        }
        try
        {
            Message message = new MimeMessage(session);
            message.setFrom(new InternetAddress("ojas.ag07@gmail.com"));
            message.setRecipients(Message.RecipientType.TO,
                InternetAddress.parse(em));
            message.setSubject("Testing Subject");
            message.setText("Dear user,"
                + randomNumber);

            Transport.send(message);

            System.out.println("Email Sent");



        } catch (MessagingException e) {
            //throw new RuntimeException(e);
            System.out.println("error");

        }
        try
        {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            Connection conn1 = DriverManager.getConnection("jdbc:odbc:myDSN1");

            String sql1;
            sql1="UPDATE Table1 set code='"+randomNumber+"' where
username='"+Login.user_name+"'";

            Statement st=conn1.createStatement();

            ResultSet rs1 = st.executeQuery(sql1);
        }
        catch(Exception e)
        {
            System.out.println("Query fired");
        }
           Email e= new Email();
            e.setVisible(true)
    }
    }
```

Email.java

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.JOptionPane;

public class Email extends javax.swing.JFrame {
    Login l2=new Login();

    public Email() {
        initComponents();
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jPasswordField1 = new javax.swing.JPasswordField();
        jLabel1 = new javax.swing.JLabel();
        code = new javax.swing.JTextField();
        enter = new javax.swing.JButton();

        jPasswordField1.setText("jPasswordField1");

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jLabel1.setText("Secure code:");

        code.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                codeActionPerformed(evt);
            }
        });

        enter.setText("OK");
        enter.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                enterMouseClicked(evt);
            }
        });
        enter.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                enterActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
```

```
                .addContainerGap(40, Short.MAX_VALUE)
                .addComponent(jLabel1)
                .addGap(63, 63, 63)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(enter)
                    .addComponent(code,     javax.swing.GroupLayout.PREFERRED_SIZE,     118,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(116, 116, 116))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(129, 129, 129)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel1)
                    .addComponent(code,              javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,   68,
Short.MAX_VALUE)
                .addComponent(enter)
                .addGap(60, 60, 60))
        );

        pack();
    }// </editor-fold>

    private void codeActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }
    private void enterMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:

        try
        {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        Connection conn = DriverManager.getConnection("jdbc:odbc:myDSN1");

            String sql2;

            sql2="select * from Table1 where username = '"+Login.user_name+"' and code =
'"+code.getText().toString()+"'";


            Statement st2=conn.createStatement();
          ResultSet rs2 = st2.executeQuery(sql2);

             if(rs2.next())
             {
                JOptionPane.showMessageDialog(null,"3 stages crossed" );
                this.dispose();
                DBaccess kun=new DBaccess();
        kun.setVisible(true);

                }
            else
            {
                JOptionPane.showMessageDialog(null,"wrong code" );
            }
        }
        catch(Exception e)
        {
            System.out.println("error");
        }
    }

    private void enterActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:

    }


    public static void main(String args[]) {

        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /*
         * If Nimbus (introduced in Java SE 6) is not available, stay with the
         * default look and feel. For details see
         * http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for          (javax.swing.UIManager.LookAndFeelInfo          info          :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Email.class.getName()).log(java.util.logging.Level.SEVE
RE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Email.class.getName()).log(java.util.logging.Level.SEVE
RE, null, ex);
```

```
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Email.class.getName()).log(java.util.logging.Level.SEVE
RE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Email.class.getName()).log(java.util.logging.Level.SEVE
RE, null, ex);
        }
        //</editor-fold>


        java.awt.EventQueue.invokeLater(new Runnable() {

            public void run() {
                new Email().setVisible(true);
            }
        });
    }
    private javax.swing.JTextField code;
    private javax.swing.JButton enter;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JPasswordField jPasswordField1;

}
```

DBaccess.java

```
import java.awt.Desktop;
import java.io.File;
import java.io.IOException;
public class DBaccess extends javax.swing.JFrame {
    public DBaccess() {
        initComponents();
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        jButton1 = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jLabel1.setFont(new java.awt.Font("Stencil Std", 1, 14)); // NOI18N
        jLabel1.setText("You've reached the vault!");

        jButton1.setText("Continue");
        jButton1.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                jButton1MouseClicked(evt);
            }
        });

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(layout.createSequentialGroup()
                    .addContainerGap(83, Short.MAX_VALUE)
                    .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 307,
javax.swing.GroupLayout.PREFERRED_SIZE))
                  .addGroup(layout.createSequentialGroup()
                    .addGap(150, 150, 150)
                    .addComponent(jButton1)
                    .addGap(0, 0, Short.MAX_VALUE)))
                .addContainerGap())
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(32, 32, 32)
                .addComponent(jLabel1,      javax.swing.GroupLayout.PREFERRED_SIZE,      93,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(76, 76, 76)
                .addComponent(jButton1)
                .addContainerGap(76, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>

    private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        try {
     if (Desktop.isDesktopSupported()) {
       Desktop.getDesktop().open(new File("D:\\a.docx"));
     }
   } catch (IOException ioe) {
  }


    }
    public static void main(String args[]) {

        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /*
         * If Nimbus (introduced in Java SE 6) is not available, stay with the
         * default look and feel. For details see
         * http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for                (javax.swing.UIManager.LookAndFeelInfo                info                :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(DBaccess.class.getName()).log(java.util.logging.Level.S
EVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(DBaccess.class.getName()).log(java.util.logging.Level.S
EVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(DBaccess.class.getName()).log(java.util.logging.Level.S
EVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
```

```
java.util.logging.Logger.getLogger(DBaccess.class.getName()).log(java.util.logging.Level.S
EVERE, null, ex);
        }
        //</editor-fold>

        java.awt.EventQueue.invokeLater(new Runnable() {

            public void run() {
                new DBaccess().setVisible(true);
            }
        });
    }
    private javax.swing.JButton jButton1;
    private javax.swing.JLabel jLabel1;
}
```

References

[1] M. Botha, R. Solms, “Utilizing Neural Networks for Effective Intrusion Detection”, ISSA,2004.

[2] R. Graham, “FAQ: Network Intrusion Detection Systems”. March 21, 2000.

[3] D. Zamboni, “Using Internal Sensors for Computer Intrusion Detection”. Center for Education and Research in Information Assurance and Security , Purdue University. August 2001.

[4] K. Scarfone, P. Mell, “Guide to Intrusion Detection and Prevention Systems (IDPS)”.Computer Security Resource Center (National Institute of Standards and Technology). February 2007.

[5]Wikipedia article on intrusion detection system and its development.
http://en.wikipedia.org/wiki/Intrusion_detection_system#Development

[6] 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.

[7] J. P. Planquart, “Application of Neural Networks to Intrusion Detection”, SANS Institute Reading Room.

[8] R. G. Bace, “Intrusion Detection”, Macmillan Technical Publishing. 2000.

[9] S. Kumar, E. Spafford, “A Software architecture to Support Misuse Intrusion Detection” in the 18th National Information Security Conference, pp. 194-204. 1995.

[10] Li, Wei. “Using Genetic Algorithm for Network Intrusion Detection ”. *Department of Computer Science and Engineering . Mississippi State University.* 2005.

[11] InfoWorld, The greatest open source software of all time, 2009;
http://www.infoworld.com/d/open-source/greatest-open-source-software-all-time
776?source=fssr

[12] Smaha, Stephen E., "Haystack: An Intrusion Detection System," The Fourth Aerospace Computer Security Applications Conference, Orlando, FL, December, 1988

[13] Intrusion Detection Techniques for Mobile Wireless Networks, ACM WINET 2003

[14] http://www.snort.org/