International Journal of Sciences: Basic and Applied Research (IJSBAR)

ISSN 2307-4531
(Print & Online)

http://gssrr.org/index.php?journal=JournalOfBasicAndApplied

A Trio Model for Network Insider Intrusion Detection & Prevention System

Charles B. Orhionkpaiyo[a*], Opani Aweh[b]

[a] Department of Mathematics and Computer Science, Federal University of Petroleum Resources, Effurun, Nigeria

[b] Department of Computer Science, Igbinedion University, Okada, Nigeria

[a] Email: Orhionkpaiyo@yahoo.com

[b] Email: opaniaweh@gmail.com

Abstract

The increasing reliance on computer networks and the internet by organizations have no doubt exposed their information to attacks from both outsiders and from the organization insiders. Different countermeasures are currently being adopted to secure information from attacks. These countermeasures are often deployed in isolation and they are all essentially designed to checking outsider threats or attacks. In this paper, an integrated approach to deploying these counter measures is proposed, and the possibility of deploying these counter measures to check insider attacks is presented. An objected oriented design methodology was used to design the platform upon which this integration was based. Data modification and impersonation attack scenarios were simulated and forensically analyzed to test the functionalities desired. The results showed that the integrated use of the detectors enhanced information protection and at the same time it provided for forensic evidence for establishing the culpability of the exact offender.

Keywords: Forensic Analysis; Trio model; Network Intrusion Detection; Insider Threats; Network Security; Honeypot; Horney-Token; Wireshark.

[*] Corresponding author.

E-mail address: Orhionkpaiyo@yahoo.com.

1 Introduction

As a result of the provisioning of services on networks and internet, vita information is exposed to attacks from both the outsiders and insider users of the network. These attacks can be in the form of disclosure, destruction, modification of data or denial of access to data.

Consequently, different security counter measures must be put in place by organizations to deal with these attacks intended against their information. The commonly used counter measures include firewall, antivirus software, and Intrusion Detection System (IDS).

However, these intrusion detection approaches are often deployed in isolation and have often targeted the attacks at the system level and neglected attacks on user application where most malicious activities normally occur. Also, it is a challenge for these counter measures to have knowledge of the user responsibilities in a network because they work in isolation from access control for the applications they are designed to protect. This lack of coordination and interoperation among these components constitutes a major setback in detecting and responding appropriately to the rising incidence of network attacks and network insider abuses.

To effectively address the problem of intentional malicious adversaries, [1] posit that systems must leverage multiple complementary and mutually supportive techniques to detect and deter them. What this statement advises is that organization needs to do everything possible to ensure that their information resources are adequately protected using more than one technique.

Designing and implementing appropriate network security measures must be based on identification of potential threats to the network and the related vulnerabilities. And the design objectives must seek to minimize the risks of any potential threats.

What obtains currently is that most attention is focused on threats from the outside while those of the insiders are overlooked. While organizations are likely to experience more outsider attacks than insider attacks the effects of the insider attacks may be more grievous and more difficult to detect [2]. Consequently, organizations need to have the appropriate tools to identify and detect early signs of insider threats with the view to implementing strategies to detect and prevent such attacks. This is the thrust of this paper, and the motivation is the need for a platform that will support the interoperability of intrusion detection approaches with access control and authentication mechanisms.

2 Related Literatures

In order to check the occurrences of intrusions, many organizations have adopted different strategies. The three most popular security technologies implemented by organizations to reduce their exposure to security threats are listed in [3]. These are firewalls, antivirus software, and intrusion detection systems. The most basic technology used is the firewall. A firewall is a mechanism for maintaining control over the traffic that flows into and out of a network [4]. This mechanism can be a machine or software that stands between a local network and the internet to filter out traffic that might be harmful. The whole essence of using a firewall is to limit access between networks and this is usually done in accordance with organization's security policies.

The biggest disadvantage of firewall, according to [5] is that it gives no protection against insider attacks. Insiders do not need to pass through firewall to access the network as they are already within the network perimeter. And it is these insiders that actually pose a higher risk to security of organization's information resources. The report of [6] showed a marked increase in the incidence of insider's threats to organizations information resource between the periods 2010 and 2011. Incidentally, insider intrusion detection remains an active area of research.

The first compelling reason listed in [7] for acquiring and using IDS was the detection of attacks and other security violations that were not prevented by other security measures. The second was to prevent problem behavior by increasing the perceived risk of discovery and punishment for those who would attack or otherwise abuse the system.

Ordinarily, IDS do not provide protection on the networks because intrusions are often detected after they have occurred. Secondly, IDS have limitations in detecting insider misuse of resources because they do not have knowledge of user responsibilities and the separation of duties that should be enforced. This is because present IDS work in isolation from access control for the application the system aim to protect [8]. The implication of this finding is that IDS need to be supported by other security measures such as access control mechanism to enhance proactive detection. The authors recognize this fact that even if IDS may use very good signature analysis mechanisms to detect intrusion or potential misuse, organizations must still ensure that they have strong user identification and authentication mechanism in place.

Some Researchers have suggested the use of honeypots for intrusion detection instead of the traditional IDS. A honeypot, according to [9], is a decoy computer system that uses deception to lure intruders so as to learn their behaviors. Any interaction with a honeypot is likely an unauthorized or anomalous activity [10].

Typically, honeypot are deployed to detect new attacks and to address the challenges of false alarms. The logs from honeypots are normally used for researching hacking techniques, early intrusion detection, and incident response [11].

Honeypot usefulness is dependent on the intruders' interaction with it and this is a huge limitation. To address this limitation, honeypot are always carefully designed to attract and to contain intruders, and this most time requires the use of honeytoken to lead intruders to the honeypot. Honeytokens are like honeypot except that they are not computers but digital or information resource such as a word document; excel or spread sheet document, passwords, or database records [10].

3. Methodology

The proposed integrated approach consists of multiple intrusion detection components made up of a trio of detectors, comprising of honeytoken, snort IDS, and a honeypot. These detectors were made to interact in a complementary and mutually supportive fashion on the one hand, and on the other hand, they were made to interoperate effectively with an access control and authentication mechanism.

3.1 *Conceptual Description of the Trio Model*

The model consists of three phases, namely prevention, detection, and deterrence phases. Based on this trio of phases (**PRE**vention, **D**etection and **DET**erence), the model is called the PREDDET model.

The prevention phase is the top layer of the model and is concerned with the prevention of insider attacks such as unauthorized access, data modification, and privilege escalation. This phase is the first line of defence of the model. Prevention is enforced through access control and authentication processes. A user must be authenticated before access can be granted.

On authentication, a role-based access control (rbac) mechanism is implemented. The rbac grants users access to the database based on roles authorized for each group of users. After the users are identified in the authentication module, the tasks permitted for the class of the users are displayed. So when a user is added to the system, the user is assigned roles based on his classification. The classification is determined by the job title of the user. The model also enforces a constraint, known as separation of duties. Separation of duties requires that no one individual should be able to process a transaction from initiation to completion [12]. This constrain makes it mandatory for a user to obtain authorization from the network administrator for any modification process. An unsuccessful authentication activates a token to display for use by the user. The use of this token and/or attempt to escalate privileges (such as a user attempting to authorize himself) triggers the detection phase of the model.

The detection phase is the middle layer of the model and handles detection using multiple intrusion detection components coordinated by the control system. When the control system of the middle layer is triggered, it activates the honeypot which then presents the intruder with a decoy (‘fake’) database from the database server. At this time, the IDS and the honeypot start monitoring the activities of the intruder, logging them to the audit log file. By this approach, the honeypot diverts the attention of the intruder from the real database and provides containment for him. Intrusion detection alert is also generated and sent to the administrator.

The logs from the IDS and honeypot are passed to the Forensic Analysis Tool (FAT) for analysis. This is the third phase and bottom layer of the model and is intended to establish the culpability of the offender as a deterrent to others in the system. The outcome of the forensic analysis is documented for necessary action by the network administrator. The following Figure 1 shows the conceptual view of the model.

3.2 *System Implementation*

The components specified in the trio model were assembled. These are an open source IDS called Snort; a virtual honeypot called v-honey and a honeytoken. The v-honey was designed with MySQL database and an interface for uploading approved students’ examination results.

An object oriented design approach was explored for the design of the application that provided the platform for the three components to function in a mutually supportive manner. This application also had a feature to support and enforce an authentication procedure and access control. This application was implemented with Java, PHP and JavaScript languages.

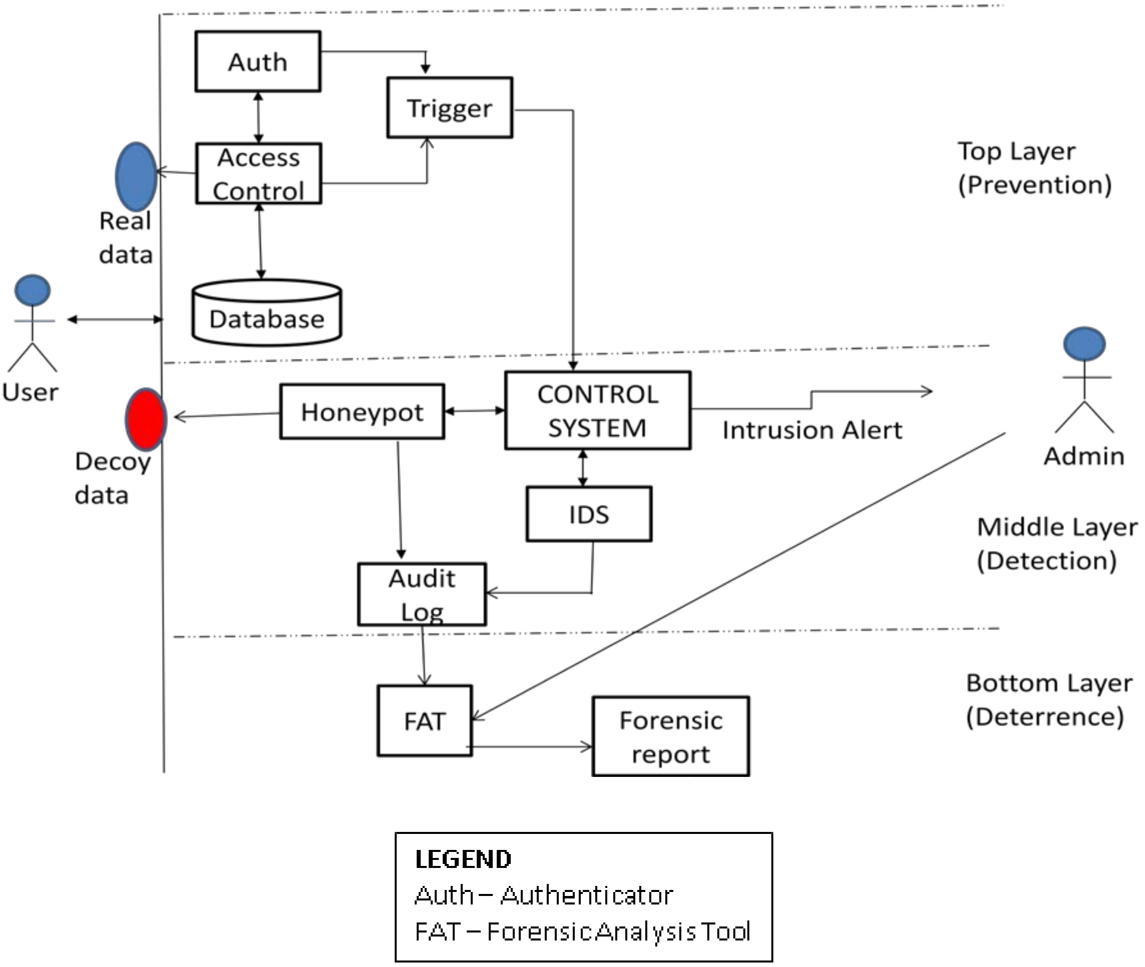


Fig.1. Conceptual view of the proposed model

These application and the components it supports were deployed on an ad-hock wireless network. The network consists of six computers connected wirelessly by means of Access Point (AP). Snort and the v-honey were installed in the same system. The database server (Tomcat server) was set up at port 8080, where both snort and v-honey were set to monitor. Attack scenarios relating to data alterations and privilege escalation were then experimented using this environment.

The network users used to test these attack scenarios were classified using a classification tree. Access rights were assigned to these users according to designated roles and a privilege table was designed and used to record access privileges of these users as shown in table 1.

Table 1: Privilege Table

Role	Permission Set	Access Control List (ACL)
ExamOfficer/ levelAdviser	[C,D,R,W]	Resultfile-object
HOD	[-,D,R,W]	Resultfile-object

Where

C (Create) – allows the creation or renaming of network object (such as database server)

D (Delete) – enables the deletion of a network object

R (Read) – allows user to read the content of an object value

W (Write) – allows user to write or modify the content of an object state

-            implies that the privilege is not granted

The above privilege table shows examOfficers and levelAdvisers are assigned access to all the privileges for the resultfile-object and is represented as resultfile-object.type = {C,D,R,W} while the HOD object type is represented as resultfile-object.type = {-,D,R,W} which assigns him all privileges except create.

Similarly, information assets were classified using a decision tree and this was to enable us to determine the security level for each of the information items. Access rights were implemented based on role-based access control (rbac) model and Mysql database was used to store users' credentials needed for authentication and their access rights.

A user is authenticated by three set of credentials namely, username, password and a system MAC (Media Access Control) address. The system MAC addresses of every user were pre-registered and are automatically authenticated during a user login operation. The authentication module interoperates with the access control module. On user authentication, access is granted based on user roles. This module operates on the privilege table shown in table1 above.

The authentication module also interoperates with Snort and v-honey by means of honey-tokens. At the entry point of the authentication module, honey-tokens in the form of login tokens were implanted. The use of the honeytoken triggers Snort IDS and v-honey to start logging the user activities on the network. It also triggers the generation of an SMS alerts that notifies the network administrator of an ongoing attack. The alert is sent to the mobile phone of the network administrator.

3.3 *System Testing*

The specific attack scenarios simulated were data modification and impersonation attack scenario on students' examination results.

The attack scenario involves an exam officer attempting to bypass the Head of Department (HOD) to modify the examination grades of a student using the HOD web address to access the result database server. The officer login with his credentials to gain access to the network application, then he initiated the modification process without seeking approval from the HOD. Ordinarily, the department enforces the principle of segregation of duties. This implies that the HOD must authorize any modification to all approved results.

To authorize modification, the authentication module randomly generates a code for the HOD to authorize the completion of any modification. The exam officer tried to forge a code but the system could not match it with the HOD credentials. The system then grants him access to the v-honey honeypot instead of the real database server. The v-honey and Snort started logging his activities. The logs from v-honey and Snort were later collected and forensically analyzed with wireshark.

3.4 *Results and Discussion*

The authentication and access control processes introduced in the intrusion detection system makes the system proactive to intrusions. The authentication module in the front end of the system was effective due to the honey-tokens implanted into it. Any unusual interaction with this module or attempt to bypass it triggers an SMS alert and the other detection components. The network administrator got the SMS alert in his mobile phone while the intrusion was ongoing. The alert shows the IP address of the intruder and the system successfully redirected the user to the v-honey

The access control system enabled the intrusion detection components to be aware of the responsibilities of network users. The exam officer who attempted to bypass the HOD was detected since the system was able to tag responsibilities to every class of users. A breach of the segregation of duties principle implemented in the authentication module triggered the detection components and the misfeasor was redirected to the virtual honeypot (i.e. v-honey).

The data from the IDS and the v-honey provided more data for wider spectrum comparison and forensic analysis.

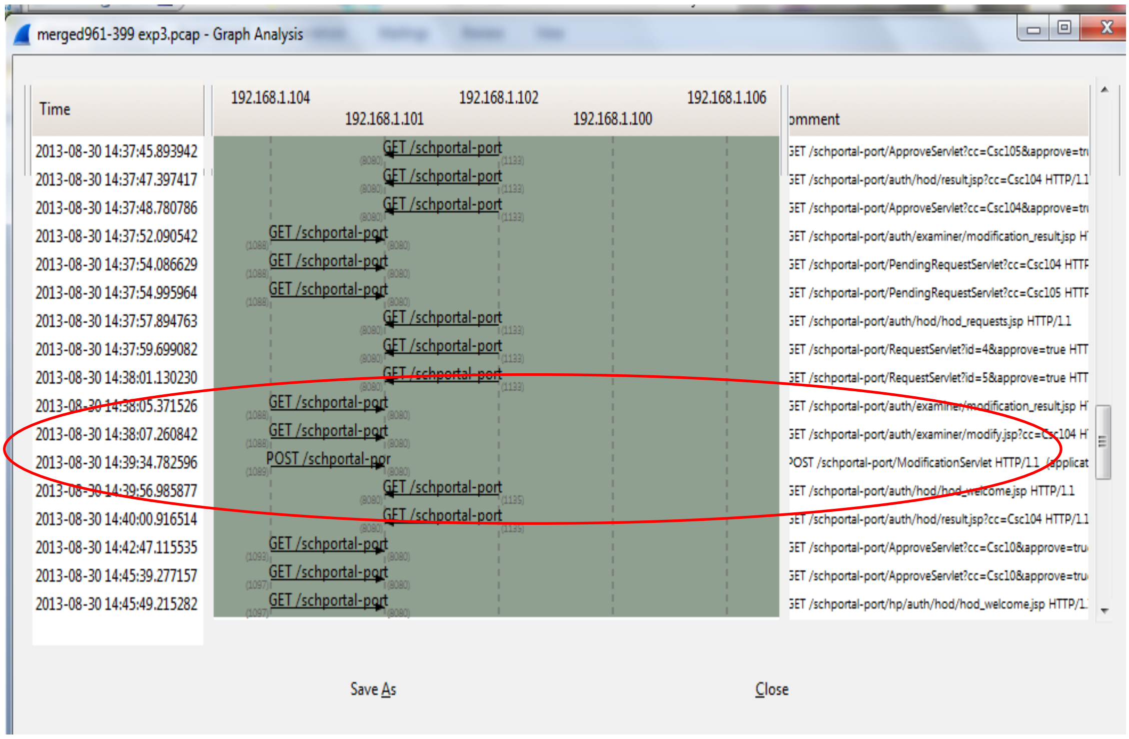
3.5 *Forensic Analysis*

A forensic (post-mortem) analysis was conducted on snort captured packets using a network packet analyzer called Wireshark. The statistical menu of the Wireshark was used to generate the graphs. For all the I/O graphs, a tick interval of 10secs and 10 pixels per tick on the x axis were used while on the y axis, the packets/tick and an auto scale were used.

The [13] procedures for digital forensic analysis were adopted. The snort log files derived from the experiments were recovered and copied into the system where Wireshark was installed. A list of search keywords in the log file was created and include the following - http, login, modify/modification, MAC address, IP address, Admin, and Password.

The recovered snort log files (1377869961 and 1377869399) were merged using the Wireshark merge menu. The merged file, merged961-399, was used as a sample for the analysis. The opened file was set to display only packets captured by http since we were interested in the user event with the web browser.

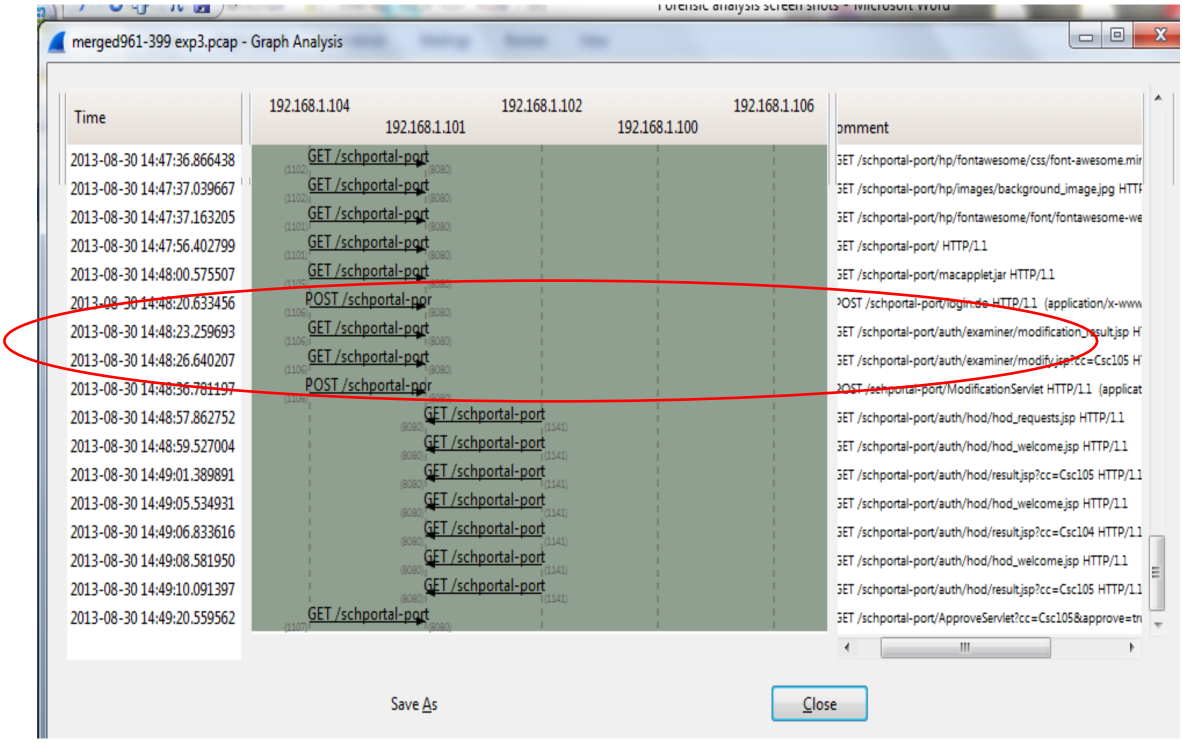
The merged961-399 log file was examined, focusing on the revelant artifacts that would support or contradict our expectation. A conversation flow graph was generated from the http filtered displayed log file to establish all the computers involved in conversation with the server. The information in the comment column of the graph were examined for login, modify, modification, admin, and password and we found in the graph modification keyword initiated by 192.168.1.104. Then a tracing of all the conversations of this IP was conducted and discovered that at time 14.38 a modification of csc104 was made by this IP (192.168.1.104). It also approved the same result at time 14.39. The following Figure 2 and Figure 3 show the conversation flow graph.



Time		omment
2013-08-30 14:37:45.893942	GET /schportal-port	GET /schportal-port/ApproveServlet?cc=Csc105&approve=tr
2013-08-30 14:37:47.397417	GET /schportal-port	GET /schportal-port/auth/hod/result.jsp?cc=Csc104 HTTP/1.1
2013-08-30 14:37:48.780786	GET /schportal-port	GET /schportal-port/ApproveServlet?cc=Csc104&approve=tr
2013-08-30 14:37:52.090542	GET /schportal-port	GET /schportal-port/auth/examiner/modification_result.jsp H
2013-08-30 14:37:54.086629	GET /schportal-port	GET /schportal-port/PendingRequestServlet?cc=Csc104 HTTF
2013-08-30 14:37:54.995964	GET /schportal-port	GET /schportal-port/PendingRequestServlet?cc=Csc105 HTTF
2013-08-30 14:37:57.894763	GET /schportal-port	GET /schportal-port/auth/hod/hod_requests.jsp HTTP/1.1
2013-08-30 14:37:59.699082	GET /schportal-port	GET /schportal-port/RequestServlet?id=4&approve=true HTT
2013-08-30 14:38:01.130230	GET /schportal-port	GET /schportal-port/RequestServlet?id=5&approve=true HTT
2013-08-30 14:38:05.371526	GET /schportal-port	GET /schportal-port/auth/examiner/modification_result.jsp H
2013-08-30 14:38:07.260842	GET /schportal-port	GET /schportal-port/auth/examiner/modify.jsp?cc=Csc104 H
2013-08-30 14:39:34.782596	POST /schportal-por	POST /schportal-port/ModificationServlet HTTP/1.1 (applicat
2013-08-30 14:39:56.985877	GET /schportal-port	GET /schportal-port/auth/hod/hod_welcome.jsp HTTP/1.1
2013-08-30 14:40:00.916514	GET /schportal-port	GET /schportal-port/auth/hod/result.jsp?cc=Csc104 HTTP/1.1
2013-08-30 14:42:47.115535	GET /schportal-port	GET /schportal-port/ApproveServlet?cc=Csc10&approve=tru
2013-08-30 14:45:39.277157	GET /schportal-port	GET /schportal-port/ApproveServlet?cc=Csc10&approve=tru
2013-08-30 14:45:49.215282	GET /schportal-port	GET /schportal-port/hp/auth/hod/hod_welcome.jsp HTTP/1.1

Fig.2. Evidence of modification

The circled part in Figure 2 above shows the data modification initiated and performed by IP 192.168.1.104 between time 14.38 and 14.39 he modified csc104



Time	192.168.1.104	192.168.1.101	192.168.1.102	192.168.1.100	192.168.1.106	omment
2013-08-30 14:47:36.866438	(1102) GET /schportal-port	(8080)				GET /schportal-port/hp/fontawesome/css/font-awesome.mir
2013-08-30 14:47:37.039667	(1102) GET /schportal-port	(8080)				GET /schportal-port/hp/images/background_image.jpg HTTI
2013-08-30 14:47:37.163205	(1101) GET /schportal-port	(8080)				GET /schportal-port/hp/fontawesome/font/fontawesome-we
2013-08-30 14:47:56.402799	(1101) GET /schportal-port	(8080)				GET /schportal-port/ HTTP/1.1
2013-08-30 14:48:00.575507	(1105) GET /schportal-port	(8080)				GET /schportal-port/macapplet.jar HTTP/1.1
2013-08-30 14:48:20.633456	(1106) POST /schportal-por	(8080)				POST /schportal-port/login.do HTTP/1.1 (application/x-www
2013-08-30 14:48:23.259693	(1106) GET /schportal-port	(8080)				GET /schportal-port/auth/examiner/modification_result.jsp H
2013-08-30 14:48:26.640207	(1106) GET /schportal-port	(8080)				GET /schportal-port/auth/examiner/modify.jsp?cc=Csc105 H
2013-08-30 14:48:36.781197	(1106) POST /schportal-por	(8080)				POST /schportal-port/ModificationServlet HTTP/1.1 (applicat
2013-08-30 14:48:57.862752		(8080) GET /schportal-port	(1141)			GET /schportal-port/auth/hod/hod_requests.jsp HTTP/1.1
2013-08-30 14:48:59.527004		(8080) GET /schportal-port	(1141)			GET /schportal-port/auth/hod/hod_welcome.jsp HTTP/1.1
2013-08-30 14:49:01.389891		(8080) GET /schportal-port	(1141)			GET /schportal-port/auth/hod/result.jsp?cc=Csc105 HTTP/1.1
2013-08-30 14:49:05.534931		(8080) GET /schportal-port	(1141)			GET /schportal-port/auth/hod/hod_welcome.jsp HTTP/1.1
2013-08-30 14:49:06.833616		(8080) GET /schportal-port	(1141)			GET /schportal-port/auth/hod/result.jsp?cc=Csc104 HTTP/1.1
2013-08-30 14:49:08.581950		(8080) GET /schportal-port	(1141)			GET /schportal-port/auth/hod/hod_welcome.jsp HTTP/1.1
2013-08-30 14:49:10.091397		(8080) GET /schportal-port	(1141)			GET /schportal-port/auth/hod/result.jsp?cc=Csc105 HTTP/1.1
2013-08-30 14:49:20.559562	(1107) GET /schportal-port	(8080)				GET /schportal-port/ApproveServlet?cc=Csc105&approve=tr

Fig.3. Modification and approval by IP 192.168.1.104

In the above figure 3, the circled part shows when (between time 14.48.20 and 14.48.36) the system with IP 192.168.1.104 login and modify csc105 without HOD’s input.

Next, an Input and Output graph was plotted to trace the http captured packets in time of day in order to locate the packet frame as shown in Figure 4.

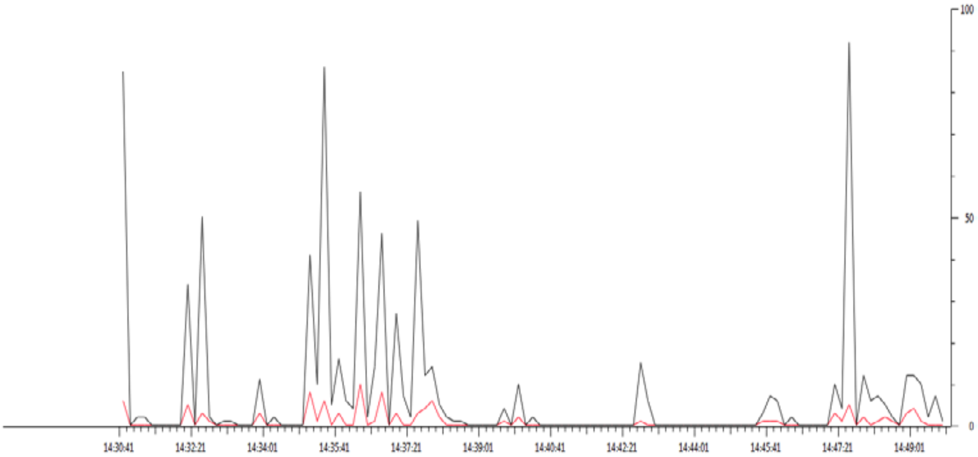
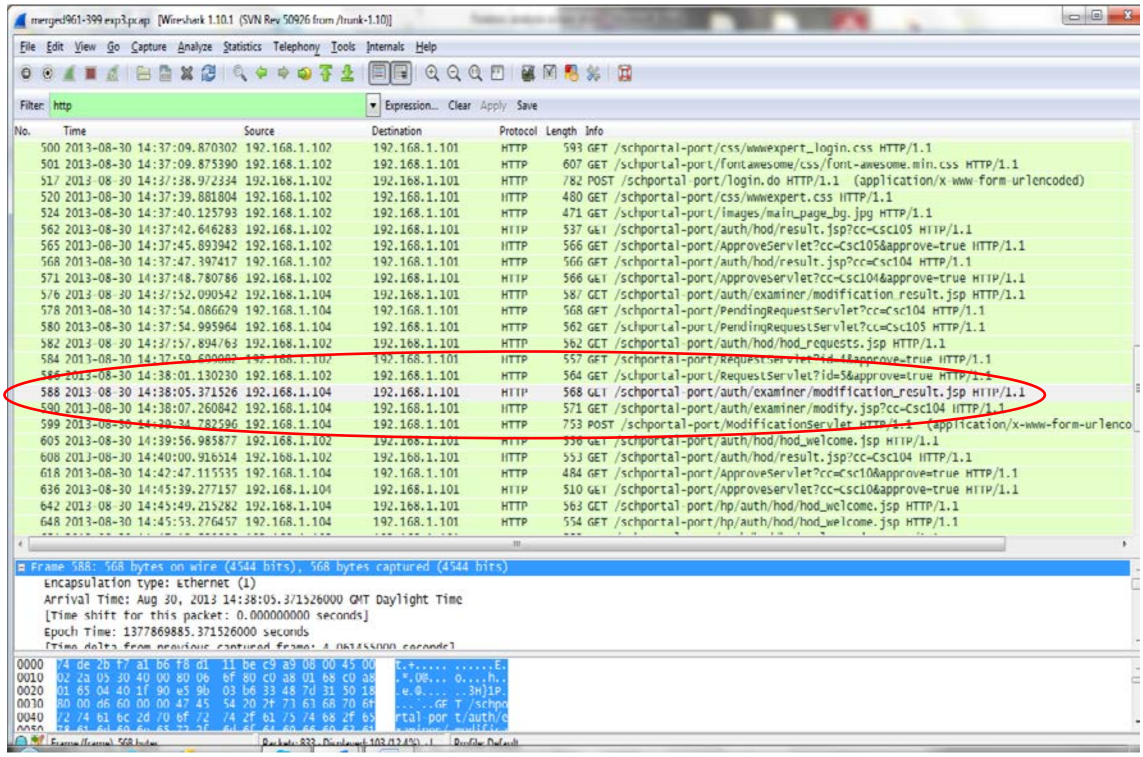


Fig.4. IO Graph showing captured packets

We located time 14.37 and 14.39 on the time axis respectively and clicked on the red graph at the specified position while looking at the packet-header detail window below the graph. A click on the spike of time 14.38 displayed frame 588 and 14.39 displayed 599 in the packet-header detail window of wireshark. We noted that in frame 599, the source IP 192.168.1.104 posted the modified csc104 result to the server as shown in the the portion circled in Figure 5.


merged961-399 exp3.pcap [Wireshark 1.10.1 (SVN Rev 50926 from /trunk-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply Save
Frame (frame), 568 bytes
Packets: 833 · Displayed: 103 (12.4%)
Profile: Default

Filter: http

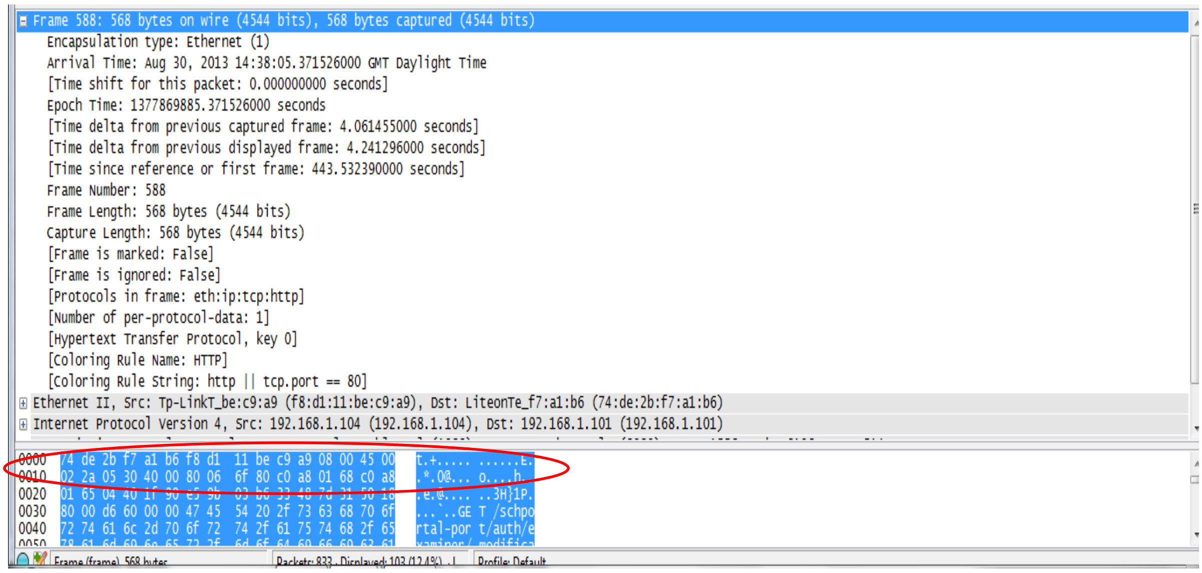
```
No.  Time                       Source         Destination    Protocol Length Info
 500 2013-08-30 14:37:09.870302 192.168.1.102  192.168.1.101  HTTP     593 GET /schportal-port/css/wwwexpert_login.css HTTP/1.1
 501 2013-08-30 14:37:09.875390 192.168.1.102  192.168.1.101  HTTP     607 GET /schportal-port/fontawesome/css/font-awesome.min.css HTTP/1.1
 517 2013-08-30 14:37:38.972334 192.168.1.102  192.168.1.101  HTTP     782 POST /schportal-port/login.do HTTP/1.1  (application/x-www-form-urlencoded)
 520 2013-08-30 14:37:39.881804 192.168.1.102  192.168.1.101  HTTP     480 GET /schportal-port/css/wwwexpert.css HTTP/1.1
 524 2013-08-30 14:37:40.125793 192.168.1.102  192.168.1.101  HTTP     471 GET /schportal-port/images/main_page_bg.jpg HTTP/1.1
 562 2013-08-30 14:37:42.646283 192.168.1.102  192.168.1.101  HTTP     537 GET /schportal-port/auth/hod/result.jsp?cc=Csc105 HTTP/1.1
 565 2013-08-30 14:37:45.893942 192.168.1.102  192.168.1.101  HTTP     566 GET /schportal-port/ApproveServlet?cc=Csc105&approve=true HTTP/1.1
 568 2013-08-30 14:37:47.397417 192.168.1.102  192.168.1.101  HTTP     566 GET /schportal-port/auth/hod/result.jsp?cc=Csc104 HTTP/1.1
 571 2013-08-30 14:37:48.780786 192.168.1.102  192.168.1.101  HTTP     566 GET /schportal-port/ApproveServlet?cc=Csc104&approve=true HTTP/1.1
 576 2013-08-30 14:37:52.090542 192.168.1.104  192.168.1.101  HTTP     587 GET /schportal-port/auth/examiner/modification_result.jsp HTTP/1.1
 578 2013-08-30 14:37:54.086629 192.168.1.104  192.168.1.101  HTTP     568 GET /schportal-port/PendingRequestServlet?cc=Csc104 HTTP/1.1
 580 2013-08-30 14:37:54.995964 192.168.1.104  192.168.1.101  HTTP     562 GET /schportal-port/PendingRequestServlet?cc=Csc105 HTTP/1.1
 582 2013-08-30 14:37:57.894763 192.168.1.102  192.168.1.101  HTTP     562 GET /schportal-port/auth/hod/hod_requests.jsp HTTP/1.1
 584 2013-08-30 14:37:59.690082 192.168.1.102  192.168.1.101  HTTP     557 GET /schportal-port/RequestServlet?id=4&approve=true HTTP/1.1
 586 2013-08-30 14:38:01.130230 192.168.1.102  192.168.1.101  HTTP     564 GET /schportal-port/RequestServlet?id=5&approve=true HTTP/1.1
 588 2013-08-30 14:38:05.371526 192.168.1.104  192.168.1.101  HTTP     568 GET /schportal-port/auth/examiner/modification_result.jsp HTTP/1.1
 590 2013-08-30 14:38:07.260842 192.168.1.104  192.168.1.101  HTTP     571 GET /schportal-port/auth/examiner/modify.jsp?cc=Csc104 HTTP/1.1
 599 2013-08-30 14:39:34.782596 192.168.1.104  192.168.1.101  HTTP     753 POST /schportal-port/ModificationServlet HTTP/1.1  (application/x-www-form-urlenco
 605 2013-08-30 14:39:56.985877 192.168.1.102  192.168.1.101  HTTP     556 GET /schportal-port/auth/hod/hod_welcome.jsp HTTP/1.1
 608 2013-08-30 14:40:00.916514 192.168.1.102  192.168.1.101  HTTP     553 GET /schportal-port/auth/hod/result.jsp?cc=Csc104 HTTP/1.1
 618 2013-08-30 14:42:47.115535 192.168.1.104  192.168.1.101  HTTP     484 GET /schportal-port/ApproveServlet?cc=Csc10&approve=true HTTP/1.1
 636 2013-08-30 14:45:39.277157 192.168.1.104  192.168.1.101  HTTP     510 GET /schportal-port/ApproveServlet?cc=Csc10&approve=true HTTP/1.1
 642 2013-08-30 14:45:49.215282 192.168.1.104  192.168.1.101  HTTP     563 GET /schportal-port/hp/auth/hod/hod_welcome.jsp HTTP/1.1
 648 2013-08-30 14:45:53.276457 192.168.1.104  192.168.1.101  HTTP     554 GET /schportal-port/hp/auth/hod/hod_welcome.jsp HTTP/1.1
```

```
⊟ Frame 588: 568 bytes on wire (4544 bits), 568 bytes captured (4544 bits)
    Encapsulation type: Ethernet (1)
    Arrival Time: Aug 30, 2013 14:38:05.371526000 GMT Daylight Time
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1377869885.371526000 seconds
```

```
0000  74 de 2b f7 a1 b6 f8 d1  11 be c9 a9 08 00 45 00   t.+..... ......E.
0010  02 2a 05 30 40 00 80 06  6f 80 c0 a8 01 68 c0 a8   .*.0@... o....h..
0020  01 65 04 40 1f 90 e5 9b  03 b6 33 48 7d 31 50 18   .e.@.... ..3H}1P.
0030  80 00 d6 60 00 00 47 45  54 20 2f 73 63 68 70 6f   ...`..GE T /schpo
0040  72 74 61 6c 2d 70 6f 72  74 2f 61 75 74 68 2f 65   rtal-por t/auth/e
```

Fig.5. Packets visualized in I/O graph

The packet-header detail window was expanded to locate the MAC address of both the source (listed as f8:d1:11:be:c9:a9) and destination (listed as 74:de:2b:f7:a1:b6) computers as shown in the circled part in Figure 6.



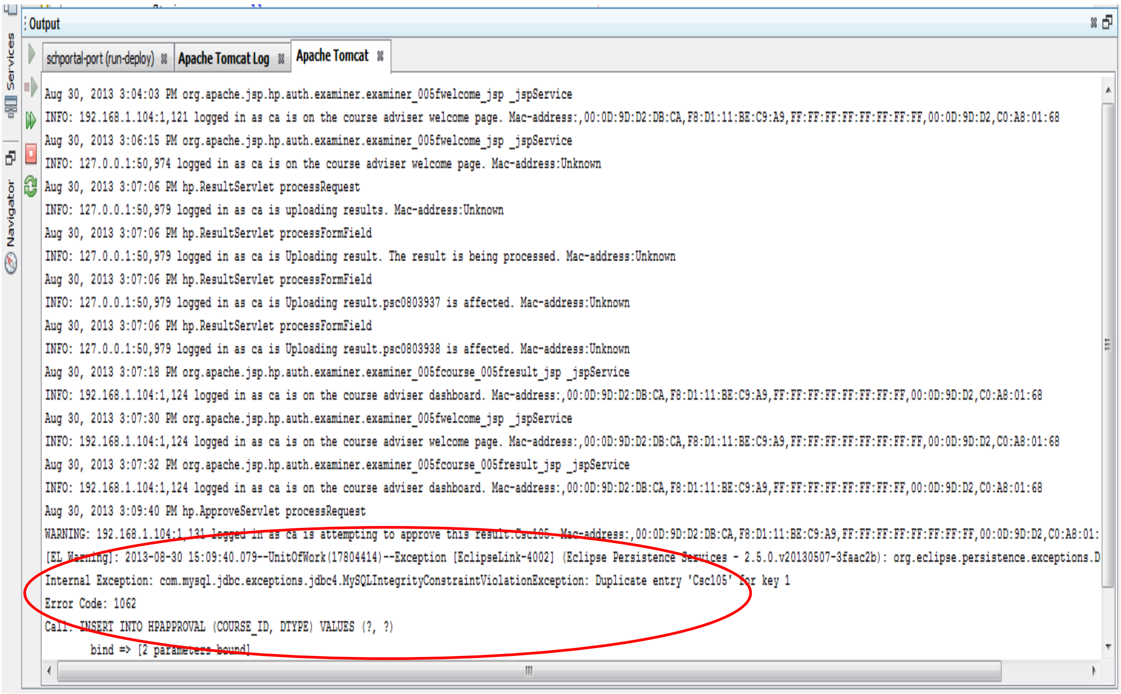
```
⊟ Frame 588: 568 bytes on wire (4544 bits), 568 bytes captured (4544 bits)
    Encapsulation type: Ethernet (1)
    Arrival Time: Aug 30, 2013 14:38:05.371526000 GMT Daylight Time
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1377869885.371526000 seconds
    [Time delta from previous captured frame: 4.061455000 seconds]
    [Time delta from previous displayed frame: 4.241296000 seconds]
    [Time since reference or first frame: 443.532390000 seconds]
    Frame Number: 588
    Frame Length: 568 bytes (4544 bits)
    Capture Length: 568 bytes (4544 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ip:tcp:http]
    [Number of per-protocol-data: 1]
    [Hypertext Transfer Protocol, key 0]
    [Coloring Rule Name: HTTP]
    [Coloring Rule String: http || tcp.port == 80]
⊞ Ethernet II, Src: Tp-LinkT_be:c9:a9 (f8:d1:11:be:c9:a9), Dst: LiteonTe_f7:a1:b6 (74:de:2b:f7:a1:b6)
⊞ Internet Protocol Version 4, Src: 192.168.1.104 (192.168.1.104), Dst: 192.168.1.101 (192.168.1.101)
```

```
0000  74 de 2b f7 a1 b6 f8 d1  11 be c9 a9 08 00 45 00   t.+..... ......E.
0010  02 2a 05 30 40 00 80 06  6f 80 c0 a8 01 68 c0 a8   .*.0@... o....h..
0020  01 65 04 40 1f 90 e5 9b  03 b6 33 48 7d 31 50 18   .e.@.... ..3H}1P.
0030  80 00 d6 60 00 00 47 45  54 20 2f 73 63 68 70 6f   ...`..GE T /schpo
0040  72 74 61 6c 2d 70 6f 72  74 2f 61 75 74 68 2f 65   rtal-por t/auth/e
```

Fig.6. Investigated Mac addresses

Ethernet II, Src: Tp-LinkT_be:c9:a9 (f8:d1:11:be:c9:a9), Dst: LiteonTe_f7:a1:b6 (74:de:2b:f7:a1:b6)

The v-honey log file was also examined for the mac addresses displayed in snort log file under examination. Figure 7 below shows the v-honey data logged simultaneously with snort on the network. The part circled shows the mac address and user event for the mac address



```
Aug 30, 2013 3:04:03 PM org.apache.jsp.hp.auth.examiner.examiner_005fwelcome_jsp _jspService
INFO: 192.168.1.104:1,121 logged in as ca is on the course adviser welcome page. Mac-address:,00:0D:9D:D2:DB:CA,F8:D1:11:BE:C9:A9,FF:FF:FF:FF:FF:FF:FF:FF,00:0D:9D:D2,C0:A8:01:68
Aug 30, 2013 3:06:15 PM org.apache.jsp.hp.auth.examiner.examiner_005fwelcome_jsp _jspService
INFO: 127.0.0.1:50,974 logged in as ca is on the course adviser welcome page. Mac-address:Unknown
Aug 30, 2013 3:07:06 PM hp.ResultServlet processRequest
INFO: 127.0.0.1:50,979 logged in as ca is uploading results. Mac-address:Unknown
Aug 30, 2013 3:07:06 PM hp.ResultServlet processFormField
INFO: 127.0.0.1:50,979 logged in as ca is Uploading result. The result is being processed. Mac-address:Unknown
Aug 30, 2013 3:07:06 PM hp.ResultServlet processFormField
INFO: 127.0.0.1:50,979 logged in as ca is Uploading result.psc0803937 is affected. Mac-address:Unknown
Aug 30, 2013 3:07:06 PM hp.ResultServlet processFormField
INFO: 127.0.0.1:50,979 logged in as ca is Uploading result.psc0803938 is affected. Mac-address:Unknown
Aug 30, 2013 3:07:18 PM org.apache.jsp.hp.auth.examiner.examiner_005fcourse_005fresult_jsp _jspService
INFO: 192.168.1.104:1,124 logged in as ca is on the course adviser dashboard. Mac-address:,00:0D:9D:D2:DB:CA,F8:D1:11:BE:C9:A9,FF:FF:FF:FF:FF:FF:FF:FF,00:0D:9D:D2,C0:A8:01:68
Aug 30, 2013 3:07:30 PM org.apache.jsp.hp.auth.examiner.examiner_005fwelcome_jsp _jspService
INFO: 192.168.1.104:1,124 logged in as ca is on the course adviser welcome page. Mac-address:,00:0D:9D:D2:DB:CA,F8:D1:11:BE:C9:A9,FF:FF:FF:FF:FF:FF:FF:FF,00:0D:9D:D2,C0:A8:01:68
Aug 30, 2013 3:07:32 PM org.apache.jsp.hp.auth.examiner.examiner_005fcourse_005fresult_jsp _jspService
INFO: 192.168.1.104:1,124 logged in as ca is on the course adviser dashboard. Mac-address:,00:0D:9D:D2:DB:CA,F8:D1:11:BE:C9:A9,FF:FF:FF:FF:FF:FF:FF:FF,00:0D:9D:D2,C0:A8:01:68
Aug 30, 2013 3:09:40 PM hp.ApproveServlet processRequest
WARNING: 192.168.1.104:1,131 logged in as ca is attempting to approve this result.Csc105. Mac-address:,00:0D:9D:D2:DB:CA,F8:D1:11:BE:C9:A9,FF:FF:FF:FF:FF:FF:FF:FF,00:0D:9D:D2,C0:A8:01:
[EL Warning]: 2013-08-30 15:09:40.079--UnitOfWork(17804414)--Exception [EclipseLink-4002] (Eclipse Persistence Services - 2.5.0.v20130507-3faac2b): org.eclipse.persistence.exceptions.D
Internal Exception: com.mysql.jdbc.exceptions.jdbc4.MySQLIntegrityConstraintViolationException: Duplicate entry 'Csc105' for key 1
Error Code: 1062
Call: INSERT INTO HPAPPROVAL (COURSE_ID, DTYPE) VALUES (?, ?)
        bind => [2 parameters bound]
```

Fig.7. Logs from v-honey

A comparison of the captured IP and MAC addresses with those in the configuration table maintained by the network administrator shows that the exam officer was the owner of the system used for the modification. It was also observed that the system with IP 192.168.1.104 and mac address f8:d1:11:be:c9:a9 did a modification of csc104 and csc105 which he by himself approved. This confirmed the expectation that the web server was compromised for unauthorized modification and privilege escalation.

3.6 *Advantages of the Proposed Model*

The model is a generic model that can be applied in different environments. It requires the identification and classification of the assets in any given environment and the model can then be applied to enforce security.

The following are specific features of the model:

- ***Uninterrupted Transaction:*** The main thrust of this model is the protection of information without interrupting transactions in the database. In most existing systems, every query execution is intercepted for hash value computation during audit logging. This frequent interruption impacts on the performance of the system. This model achieves this performance by hiding the real database and then providing a ‘fake’ database for the attacker to tamper. This approach distracts the intruder’s attention from the real database, thinking the one being modified was the real one.

- ***Prompt Alert Notification:*** Notification of intrusion alerts and authorization request and responses are routed through real time communication system. This ensures prompt reception of alerts and messages to detect on-going intrusion and events. This approach provides a framework that frees the network administrator from being constantly glued to their systems.

- ***Integrated Protection:*** The combined use of access control and multiple detectors ensure a robust protection of the database.

- ***Intruder Trace Back:*** The model provides strategies for tracing an intruder after intrusion is detected. When offenders are aware of the possibility of being caught, then the fear of the penalty as stipulated in the organization's security policy regarding unethical activities will deter potential offenders

- ***Early Detection:*** Intrusive activities are detected early in the first phase of the model. Intrusion is triggered at the top layer of the model at an early stage before any tampering is done to the database

- ***Containment:*** The 'fake' or decoy database protects the real database while presenting a fake database as containment for the intruder.

- ***Multiple Data Source for Forensic Analysis:*** It provides for data collection from multiple sources rather than a single source. The data from the IDS and the honeypot provide more data for wider spectrum analysis and comparison during forensic analysis.

- ***Knowledge of user Responsibilities:*** The role-based access control and the authentication part of the model highlight the responsibilities of the users. This feature is useful in detecting users who attempt to escalate their privileges. The incidences of false alarms are also minimized because the system is able to tag responsibilities to every class of users.

3.7 Constraints of the study: Testing the system on campus intranet was a challenge as academic institutions around us had no functional intranets. Thus setting up an ad hoc network became an option for testing. This constraint notwithstanding, we expect similar performance when deployed on a functional intranet.

4. Conclusion

Information, being one of the most valuable assets of any organization, has always been protected from unauthorized access. The approaches for providing such protection vary among organizations. Some of these measures include firewall, antivirus software, and Intrusion Detection System (IDS). In spite of these arrays of measures, there is no single approach that can guarantee protection against insider's intrusion attacks. Therefore, this proposed integrated approach provides for a fresh paradigm shift in the conception and design of robust intrusion detection systems. This study has provided an integrated detection components approach to addressing the issue of network intrusion detection and prevention strategies. Also, ability to generate evidences for post intruder activities by means of forensic analysis provides a reliable means for tracing an intruder after intrusion is detected. When offenders are aware of the possibility of being caught in an offence, then the fear of the penalty as stipulated in the organization’s security policy regarding unethical activities tend to deter them from engaging in such act. The results of all these are the emergence of an intrusion prevention, detection and deterrence system that moderates the activities of network insider’s activities.

References

[1] B. M Bowen, B. E Salem, A. D Keromytis, and S. J Stolfo. “Technologies for Mitigating Insider Threats” , Insider Threats in cyber Security, Vol. 49, Pp. 187-217, 2010

[2] A. McCormac, K. Parsons, M. Butavicius. Preventing and Profiling Malicious Insider Attacks. Defence Science and Technology Organization Document Control Data, Australia, Pp 1-17, 2012. http://www.dtic.mil/dtic/tr/fulltext/u2/a563808.pdf

[3] T. Birdi, K. Jansen. (2006) “Network Intrusion Detection: Know What You Do (Not) Need” Information Systems Audit and Control Association (ISCA) vol 1. Internet: http://www.isaca.org/Journal/Past-Issues/2006/Volume-1/Documents/jopdf0601-network-intrusion-detection.pdf (accessed on 09/08/2013)

[4] J. Andress. The Basics of Information Security: Understanding the Fundamentals of Infosec in Theory and Practice. London: Elsevier Academic Press, p117, 2011.

[5] D.W Chadwick. “Network Firewall Technologies” Security and Privacy in Advance Networking Technologies, Vol. 193, Pp 143 – 160, 2004

[6] R. Trzeciak. Risk Mitigation Strategies: Lessons Learned from Actual Insider Attacks. Internet: http://www.cert.org/insider_threat/ , 2012 (Accessed 09/08/2013)

[7] R. Bace, P. Mell. Intrusion Detection Systems. NIST Special Publications SP 800, U S Department of Defense, Pp. 40-43, 2001

[8] T. Ryuto, C. Neuman. “Integrated Access Control and Intrusion Detection for Web Servers” IEEE Transactions on Parallel and Distributed Systems, Vol. 14, No. 9, 2003

[9] P. Gaonjur, C.Bokhoree. “Risk of Insider Threats in Information Technology OutSourcing: Can Deceptive Techniques be applied?” Journal of Security and Management, Pp. 522 – 529, 2006

[10] L. Spitzner. “Honeypots: Catching the Insider Threat” Conference proceedings of Computer Security Application Pp. 170-179, 2003.

[11] B. Mcfarland. Ethical Deception and Pre-emptive Deterrence in Network Security, SANS Institute GCFW Practical Version 4.1, SANS Institute 2000-2005.

[12] B. Ruppert.” Protecting Against insider attacks” Internet: http://www.sans.org/reading-room/whitepapers/incident/protecting-insider-attacks-33168,2009 (09/08/2013)

[13] C. Eoghan, W.R Curtis. Hand Book of Digital Forensics and Investigations, Elsevier Academic Press, London, 3rd ed, pp201-219, 2010.