

# **HYBRID INTRUSION DETECTION SYSTEM**

**PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE  
REQUIREMENT FOR THE DEGREE OF**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**UNDER THE SUPERVISION OF**

**MR. PUNIT GUPTA**

**BY**

**KANIKA KHANNA – 111269**

**TO**



**JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY**

**WAKNAGHAT, SOLAN- 173234, HIMACHAL PRADESH**

# CERTIFICATE

This is to certify that project report entitled “HYBRID INTRUSION DETECTION SYSTEM”, submitted by KANIKA KHANNA (111269) in partial fulfillment for the award of degree of Bachelor of Technology in Computer Science & Engineering to Jaypee University of Information Technology, Wagnaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

Mr. Punit Gupta

(Assistant Professor)

Department of CSE & IT

Jaypee University of Information Technology

Wagnaghat, Solan , H.P- 173234

Date:

# ACKNOWLEDGEMENT

*“It is not possible to prepare a project without the assistance & encouragement of other people. This one is certainly no exception.”*

On the very outset of this report, I would like to extend my sincere & heartfelt obligation towards all the personages who have helped me in this endeavor. Without their active guidance, help, cooperation & encouragement, I would not have made headway in the project.

I would like to show my greatest appreciation to Mr. Punit Gupta. I feel motivated every time I get his encouragement. For his coherent guidance throughout the tenure of the project, I feel fortunate to be taught by Mr. Punit Gupta, who gave me his unwavering support. Besides being my mentor, he taught me that there is no substitute for hard work.

I owe my heartiest thanks to Brig. (Retd.) S.P. Ghrrera (HOD, CES/IT Department) who has always inspired me to take initiatives and showed me the path for achieving my goal.

In the light of new developments and recent findings, I devote the task that was asked from me at Jaypee University of Information Technology to **“HYBRID INTRUSION DETECTION SYSTEM”**.

Date:

Kanika Khanna

111269

# CONTENTS

<b>S.No.</b>	<b>Title</b>	<b>Page No.</b>
i	List of Figures	vii
ii	List of Tables	ix
iii	Abstract	x
<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>1.1</b>	<b>Intrusion Detection Systems</b>	<b>2</b>
1.1.1	What is intrusion and intrusion detection?	2
1.1.2	What is Intrusion Detection System?	2
1.1.3	Main benefits and characteristics of IDS	3
1.1.4	Differences between an IDS and a Firewall	4
<b>1.2</b>	<b>Classification of IDS</b>	<b>4</b>
1.2.1	Network Intrusion Detection System	5
1.2.2	Host Intrusion Detection System	6
1.2.3	Behavior Intrusion Detection System	8
<b>1.3</b>	<b>Network Attacks</b>	<b>9</b>
1.3.1	Pre-intrusive activities	9
1.3.2	Intrusions	10
<b>2</b>	<b>Literature Review</b>	<b>12</b>
<b>2.1</b>	<b>A Distributed Multi-Approach Intrusion Detection System for Web Services</b>	<b>13</b>

<b>2.2</b>	Hybrid Network Intrusion Detection System using Expert Rule Based Approach	14
<b>2.3</b>	Administrative Evaluation of Intrusion Detection System	16
<b>2.4</b>	Real- Time Malware Detection Frameworks in Intrusion Detection System	17
<b>2.5</b>	Enhancement of Fault Tolerance of Intrusion Detection System using AES and DES based Heart Beat Events	19
<b>2.6</b>	Kargus: A Highly-scalable Software-based Intrusion Detection System	20
<b>2.7</b>	A distance sum-based hybrid method for intrusion detection	22
<b>2.8</b>	A Clustering based Algorithm for Network Intrusion Detection	23
<b>2.9</b>	Divided two-part adaptive intrusion detection system	24
<b>2.10</b>	Summary	25
<b>2.11</b>	Summary	27
<b>3</b>	Problem Statement	28
<b>4</b>	Proposed Model	31
<b>4.1</b>	Functionality	34
4.1.1	Jpcap	34
4.1.2	Flowcharts	35
4.1.3	Pseudo Code	37
<b>5</b>	Experiment and Result	40
<b>5.1</b>	Outputs	44

<b>6</b>	Conclusions and Future Work	58
<b>7</b>	References	60

# LIST OF FIGURES

<b>S.No.</b>	<b>Figure</b>	<b>Page No.</b>
1	Intrusion Detection System	3
2	Network Intrusion Detection System	6
3	Host Intrusion Detection System	8
4	System Architecture of Hybrid NIDS	15
5	Flow Chart of Matching Algorithm	18
6	Typical Intrusion Detection Process in Signature-based IDS	21
7	Proposed Architecture of Hybrid IDS	32
8	Flowchart showing the larger view of path that traffic follows	35
9	Flowchart showing how traffic is handled by HIDS	36
10	Pseudo Code of sending and receiving packets	37
11	Pseudo Code showing how request from host and network are handled	38
12	Log showing packets sent	43
13	TCP packets were captured from the network with the help of jpcap library	44
14	UDP packets were captured from the network with the help of jpcap library	45
15	Rate of packets captures every 10 seconds	46
16	HTML Form used for sending HTTP Request so that client information can be retrieved by the server	47
17	Page displaying that username or password may be wrong	48

<b>18</b>	Browser Information, Server side and Client side IP Addresses retrieved from an HTTP request by the browser and displayed in Chrome	49
<b>19</b>	HIDS showing NIDS Intrusion alert	50
<b>20</b>	When no request from IP is sent, value of i=0	51
<b>21</b>	After sending first request within 2 seconds, i become 1	52
<b>22</b>	After sending second request within another 2 seconds, i become 2	53
<b>23</b>	After sending third request within another 2 seconds, i become 3	54
<b>24</b>	After sending forth request within another 2 seconds, i become 4	55
<b>25</b>	As soon as fifth request is sent within another 2 seconds, i becomes $\geq 5$	56
	and alert is sent to admin	
<b>26</b>	After fifth request, i becomes 0 again	57



## LIST OF TABLES

<b>S.No.</b>	<b>Table</b>	<b>Page No.</b>
1	Tabular representation of summary of Research Papers	27
2	Comparative Study of NIDS and HIDS	30

## **ABSTRACT**

Intrusion Detection Systems (IDS) is a security system that acts as a protection layer to the infrastructure. Throughout the years, The IDS technology has grown enormously to keep up with the advancement of computer crime. Since the beginning of the technology in mid 80's, researches have been conducted to enhance the capability of detecting attacks without jeopardizing the network performance. An intrusion detection system (IDS) provides a layer of defense which monitors network traffic for predefined suspicious activity or patterns, and alert system administrators when potential hostile traffic is detected. Intrusion detection faces a number of challenges; an intrusion detection system must reliably detect malicious activities in a network and must perform efficiently to cope with the large amount of network traffic. IDS can be a piece of installed software or a physical appliance. Many IDS tools will also store a detected event in a log to be reviewed at a later date or will combine events with other data to make decisions regarding policies or damage control. In this paper, we are going to study about the different types of IDS, basically NIDS and HIDS, their strengths and weaknesses and the advantages that we can obtain by combining both of them.

# **CHAPTER 1**

## **INTRODUCTION**

With the tremendous growth of network-based services and sensitive information on networks, network security is becoming more and more importance than ever before. Intrusion detection techniques are the last line of defenses against computer attacks behind secure network architecture design, firewalls, and personal screening. Despite the plethora of intrusion prevention techniques available, attacks against computer systems are still successful. Thus, intrusion detection systems (IDSs) play a vital role in network security.

Symantec in a recent report uncovered that the number of fishing attacks targeted at stealing confidential information such as credit card numbers, passwords, and other financial information are on the rise, going from 9 million attacks in June 2004 to over 33 millions in less than a year.

## **1.1 Intrusion detection systems**

### **1.1.1 What is intrusion and intrusion detection?**

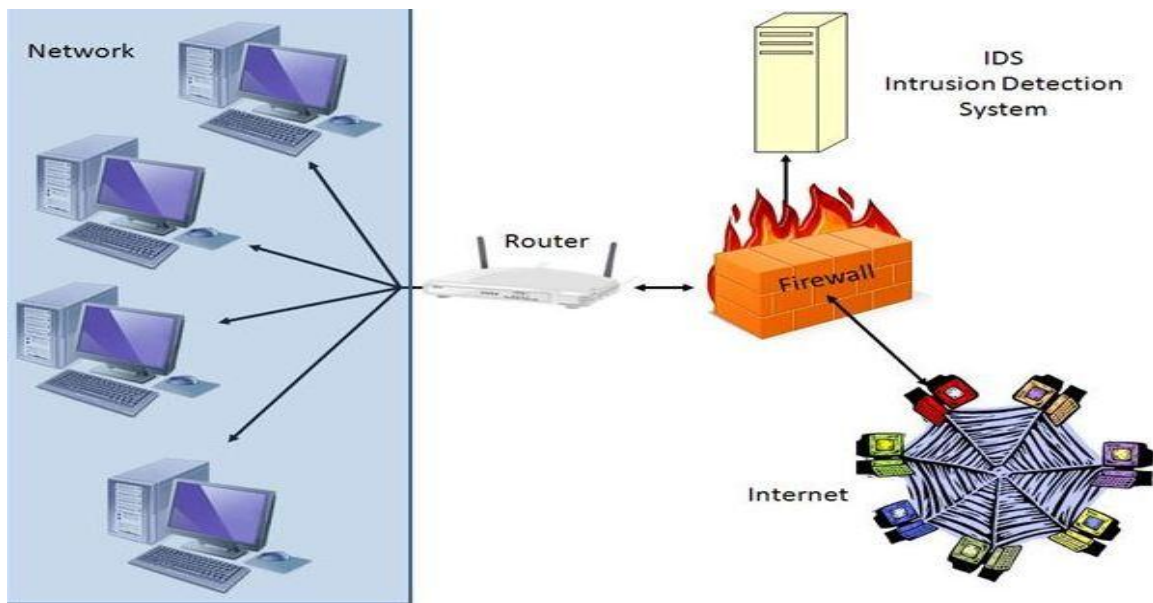
Intrusions are attempts to compromise the confidentiality, integrity and availability of a computer or network or to bypass its security mechanisms. They are caused by attackers accessing a system from the Internet, by authorized users of the systems who attempt to gain additional privileges for which they are not authorized, and by authorized users who misuse the privileges given to them.

Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of intrusions.

### **1.1.2 What is an Intrusion Detection System?**

Intrusion Detection System is any hardware, software, or a combination of both that monitors a system or network of systems against any malicious activity. This is mainly used for detecting break-ins or misuse of the network. In short, we can say that IDS is the ‘burglar alarm’ for the network because much like a burglar alarm, IDS detects the presence of an attack in the network and raises an alert.

IDS are often considered as the functionality of firewall. But there is a thin line of difference between them. A firewall must be regarded as a fence that protects the information flow and prevent intrusions where as IDS detects if the network is under attack or if the security enforced by the firewall has been breached. Together firewall and IDS enhance the security of network.



**Fig 1 – Intrusion Detection System**

### **1.1.3 Main benefits and characteristics of IDS**

The main benefits of IDS are:

- 1.* Detecting attacks and other security violations, which have not been prevented by primary protection techniques.
- 2.* Preventing problem-behaviors by increasing the perceived risk of discovery and punishment for those who would attack or otherwise abuse the system.
- 3.* Presenting traces of intrusions, allowing improved diagnosis, recovery and corrective measures after an attack.
- 4.* Documenting the existing threat from inside and outside a system, permitting security management to realistically assess risk and adapt its security strategy in response.

5. Acting as quality control for security design and implementation (highlighting some deficiencies or errors, before serious incidents occur).

### **1.1.4 Difference between a firewall and an IDS**

The risk of hacking attacks and network intrusions are facts of life for everyone who ventures online. Your system's firewall and its intrusion detection system both work to protect you against such threats. The firewall's job is to keep intruders from breaking into your network; the IDS doesn't keep them out, but it keeps track of attempts to break in.

#### **Firewalls**

A firewall restricts access to your network by screening traffic and deciding which packets should be allowed in. Boston University compares it to a security guard deciding who can get clearance. The firewall monitors the ports that connect your network to the Internet and checks data packets before allowing them to pass through. A firewall can accept a packet, drop it -- erasing it from existence -- or deny it, returning it to the sender.

#### **IDS**

If firewalls are security guards, intrusion detection systems are security cameras. An IDS monitors traffic and spots patterns of activity, alerting you if it concludes that your network is under attack. Signature detection compares network or system information to attacks already listed in the IDS database. Anomaly detection compares current network traffic to the normal levels of packet size or activity and analyzes the result statistically. If network traffic suddenly shoots up to a high level, for instance, that could indicate a hacking attack.

## **1.2 Classifications of Intrusion Detection System**

Largely, IDS are divided into 3 types:

1. Network based Intrusion Detection System (NIDS)

2. Host based Intrusion Detection System (HIDS)

3. Behavior based Intrusion Detection System (BIDS)

### **1.2.1 Network Intrusion Detection System**

A "network intrusion detection system (NIDS)" monitors traffic on a network looking for suspicious activity, which could be an attack or unauthorized activity.

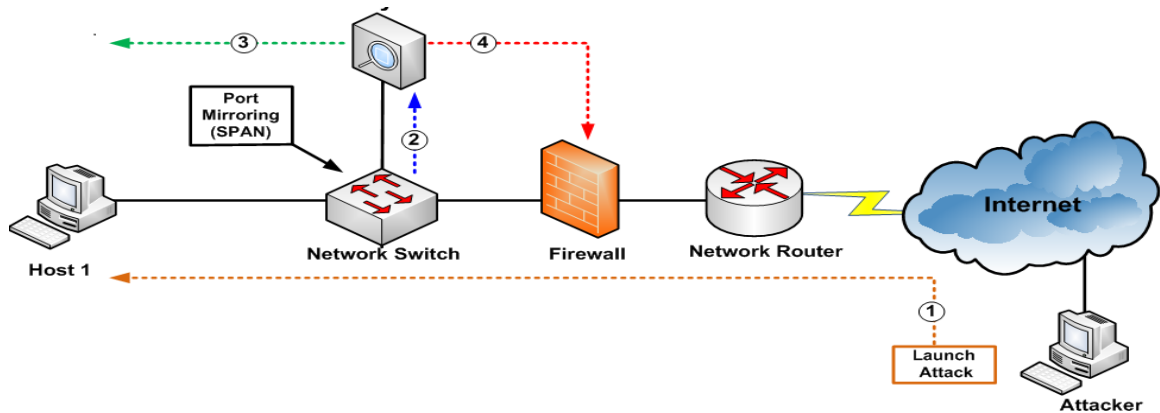
Network based intrusion detection attempts to identify unauthorized, illicit, and anomalous behavior based solely on network traffic. A network IDS, using either a network tap, span port, or hub collects packets that traverse a given network. Using the captured data, the IDS system processes and flags any suspicious traffic. Unlike an intrusion prevention system, an intrusion detection system does not actively block network traffic. The role of a network IDS is passive, only gathering, identifying, logging and alerting. Examples of Network IDS: SNORT

A network-based ID system monitors the traffic on its network segment as a data source. This is generally accomplished by placing the network interface card in promiscuous mode to capture all network traffic that crosses its network segment. Network traffic on other segments and traffic on other means of communication (like phone lines) can't be monitored.

Network-based ID involves looking at the packets on the network as they pass by some sensor. The sensor can only see the packets that happen to be carried on the network segment it's attached to. Packets are considered to be of interest if they match a signature. Three primary types of signatures are string signatures, port signatures, and header condition-signatures.

Port signatures simply watch for connection attempts to well-known, frequently attacked ports. Examples of these ports include telnet (TCP port 23), FTP (TCP port 21/20), SUNRPC (TCP/UDP port 111), and IMAP (TCP port 143). If any of these ports aren't used by the site, then incoming packets to these ports are suspicious.

Header signatures watch for dangerous or illogical combinations in packet headers. The most famous example is WinNuke, where a packet is destined for a NetBIOS port and the Urgent pointer, or Out Of Band pointer is set. This resulted in the "blue screen of death" Well-known, network-based intrusion detection systems include AXENT (acquired by Symantec), Cisco (www.cisco.com), CyberSafe (www.cybersafe.com), ISS (www.iss.net), and Shadow



**Fig 2 – Network Intrusion Detection System**

Advantages of NIDS:

- (1) Detect network based attacks.
- (2) Real time detection and quick response
- (3) Retaining evidence of attack as detection is real time

Disadvantages of NIDS:

- (1) Cannot efficiently handle high speed networks
- (2) Most of Network-based systems are based on predefined attack signatures

### **1.2.2 Host Intrusion Detection System**

Often referred to as HIDS, host based intrusion detection attempts to identify unauthorized, illicit, and anomalous behavior on a specific device. HIDS generally involves an agent installed on each system, monitoring and alerting on local OS and



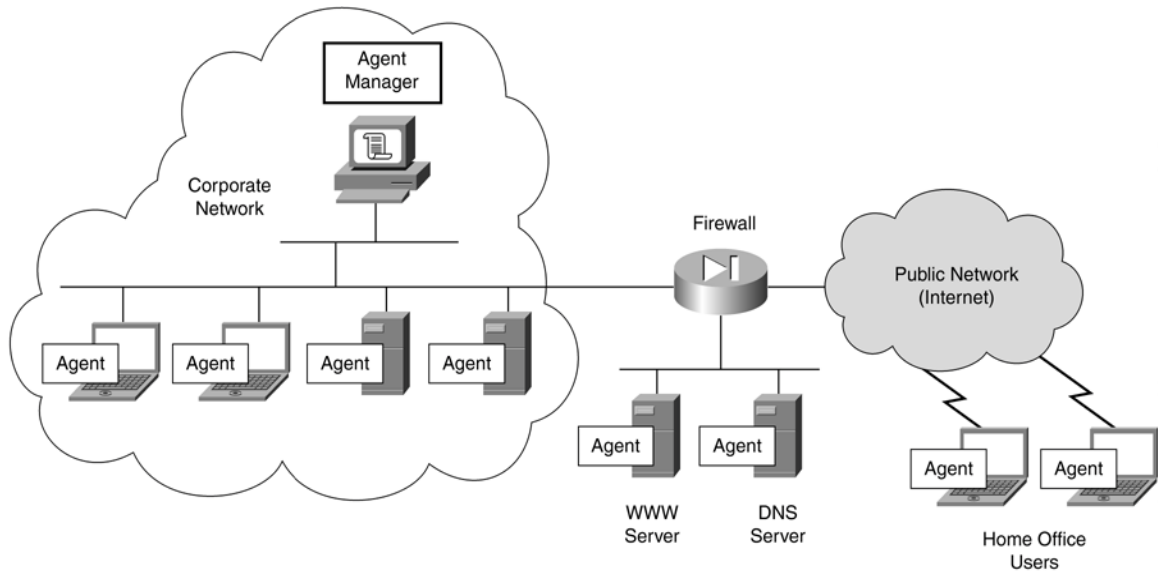
application activity. The installed agent uses a combination of signatures, rules, and heuristics to identify unauthorized activity. The role of a host IDS is passive, only gathering, identifying, logging, and alerting.

Examples of HIDS: OSSEC - Open Source Host-based Intrusion Detection System, Tripwire, AIDE - Advanced Intrusion Detection Environment, Prelude Hybrid IDS.

Host based intrusion detection (HIDS) refers to intrusion detection that takes place on a single host system. Currently, HIDS involves installing an agent on the local host that monitors and reports on the system configuration and application activity. Some common abilities of HIDS systems include log analysis, event correlation, integrity checking, policy enforcement, rootkit detection, and alerting. They often also have the ability to baseline a host system to detect variations in system configuration. In specific vendor implementations these HIDS agents also allow connectivity to other security systems. For example, Cisco CSA has the ability to send host data upstream to Cisco network IPS devices, Checkpoint Integrity can be integrated with Checkpoint Secure Client (Client VPN), and IBM Proventia Desktop is Cisco NAC certified.

Most HIDS packages now have the ability to actively prevent malicious or anomalous activity on the host system. Due to the potential impact this can have on the end user, HIDS is frequently deployed in "monitor only" mode initially. This enables the administrator to create a baseline of the system configuration and activity. Active blocking of applications, system changes, and network activity is limited to only the most egregious activities. Administrators can then tune the system policy based on what is considered "normal activity".

The HIDS agent monitors system integrity, application activity, file changes, host network traffic, and system logs. Using common hashing tools, file timestamps, system logs, and monitoring system calls and the local network interface gives the agent insight to the current state of the local host. If an unauthorized change or activity is detected, it will alert the user via a pop-up, alert the central management server, block the activity, or a combination of the three. The decision is based on the policy that is installed on the local system.



**Fig 3 – Host Intrusion Detection System**

Advantages of HIDS:

- (1) Verifies success or failure of an attack
- (2) Monitors system activities
- (3) Can operate in an environment in which network traffic is encrypted
- (4) Can help detect Trojan horse or DOS attacks
- (5) Does not require additional hardware

Disadvantages of HIDS:

- (1) May be attacked and disabled as "port attack"
- (2) Use computing resources of the hosts they monitor, so inflict a performance cost on the monitored systems.

### **1.2.3 Behavior based Intrusion Detection System**

Behavior-based intrusion detection techniques assume that an intrusion can be detected by observing a deviation from normal or expected behavior of the system or the users. The model of normal or valid behavior is extracted from reference information collected

by various means. The intrusion detection system later compares this model with the current activity. When a deviation is observed, an alarm is generated. In other words, anything that does not correspond to a previously learned behavior is considered intrusive. Therefore, the intrusion detection system might be complete (i.e. all attacks should be caught), but its accuracy is a difficult issue (i.e. you get a lot of false alarms).

Advantages of behavior-based approaches are that they can detect attempts to exploit new and unforeseen vulnerabilities. They can even contribute to the (partially) automatic discovery of these new attacks. They are less dependent on operating system-specific mechanisms. They also help detect 'abuse of privileges' types of attacks that do not actually involve exploiting any security vulnerability. In short, this is the paranoid approach: Everything which has not been seen previously is dangerous.

The high false alarm rate is generally cited as the main drawback of behavior-based techniques because the entire scope of the behavior of an information system may not be covered during the learning phase. Also, behavior can change over time, introducing the need for periodic online retraining of the behavior profile, resulting either in unavailability of the intrusion detection system or in additional false alarms. The information system can undergo attacks at the same time the intrusion detection system is learning the behavior. As a result, the behavior profile contains intrusive behavior, which is not detected as anomalous.

## **1.3 Networking Attacks**

Attacks on network computer system could be devastating and affect networks and corporate establishments. We need to curb these attacks and Intrusion Detection System helps to identify the intrusions. Without an NIDS, to monitor any network activity, possibly resulting in irreparable damage to an organization's network

Intrusion attacks are those in which an attacker enters your network to read, damage, and/or steal your data. These attacks can be divided into two subcategories: pre intrusion activities and intrusions.

### **1.3.1 Pre intrusion activities**

Pre intrusion activities are used to prepare for intruding into a network. These include port scanning to find a way to get into the network and IP spoofing to disguise the identity of the attacker or intruder.

### **1. Port scans**

A program used by hackers to probe a system remotely and determine what TCP/UDP ports are open (and vulnerable to attack) is called a scanner. A scanner can find a vulnerable computer on the Internet, discover what services are running on the machine, and then find the weaknesses in those services. There are 65,535 TCP ports and an equal number of UDP ports. Stealth scanners use what is called an IP half scan, sending only initial or final packets instead of establishing a connection, to avoid detection.

### **2. IP spoofing**

This is a means of changing the information in the headers of a packet to forge the source IP address. Spoofing is used to impersonate a different machine from the one that actually sent the data. This can be done to avoid detection and/or to target the machine to which the spoofed address belongs. By spoofing an address that is a trusted port, the attacker can get packets through a firewall.

## **1.3.2 Intrusions**

### **1. Source routing attack**

This is a protocol exploit that is used by hackers to reach private IP addresses on an internal network by routing traffic through another machine that can be reached from both the Internet and the local network [7, 8]. TCP/IP to allow those sending network data to route the packets through a specific network point for better performance supports source routing. Administrators to map their networks or to troubleshoot routing problems also use it.

### **2. Trojan attacks**

Trojans are programs that masquerade as something else and allow hackers to take control of your machine, browse your drives, upload or download data, etc. For example,

in 1999, a Trojan program file called Picture.exe was designed to collect personal data from the hard disk of an infiltrated computer and send it to a specific e-mail address. So-called Trojan ports are popular avenues of attack for these programs.

### **3. Registry attack**

In this type of attack, a remote user connects to a Windows machine's registry and changes the registry settings. To prevent such an attack, configure permissions so that the every one group does not have access.

### **4. Password hijacking attacks**

The easiest way to gain unauthorized access to a protected system is to find a legitimate password. This can be done via social engineering (getting authorized users to divulge their passwords via persuasion, intimidation, or trickery) or using brute force method.

### **5. DOS attack and Distributed Denial of Service Attack (DDoS)**

Attack designed to cause an interruption or suspension of services of a specific host/server by flooding it with large quantities of useless traffic or external communication requests. When the DoS attack succeeds the server is not able to answer even to legitimate requests any more - this can be observed in numbers of ways: slow response of the server, slow network performance, unavailability of software or web page, inability to access data, website or other resources. Distributed Denial of Service Attack (DDoS) occurs where multiple compromised or infected systems (botnet) flood a particular host with traffic simultaneously

## **CHAPTER 2**

# **LITERATURE REVIEW**

## **2.1. TITLE -**

### **A Distributed Multi-Approach Intrusion Detection System for Web Services**

## **SUMMARY -**

The proposed WS-IDS have the following important features:

- (1) To cover multiple web services deployed in an enterprise network, we use a distributed structure. It causes to mitigate the processing overhead of central system and thus make lower risk of denial service (DoS) attack to WS-IDS itself.
- (2) By using two detection approaches, misuse detection and anomaly detection, improve the detection rate of the system with lower false alarm rate.

It is possible to monitor the interactions of web services and requesters by studying request/response messages; identify and induce some types of attacks and malicious behaviours by analysing captured SOAP messages. Of course for tracking the requesters and identify it we need some information from underneath layers.

### **2.1.1 The Proposed Architecture**

The structure of the proposed WS-IDS is based on two features:

- (1) Using both anomaly and misuse detection techniques
- (2) A distributed architecture for web service hosts and the WS-IDS

## **2.2. TITLE -**

### **Hybrid Network Intrusion Detection System using Expert Rule Based Approach**

#### **SUMMARY -**

The known attack patterns are identified, with misuse detection system, using the rule base and, with anomaly detection new attacks are identified by deploying clustering techniques. The new attacks have been updated in the rule base with the knowledge from an expert database that improved the efficiency of the system.

The detection rate of the hybrid system has been found to increase as the unknown attack percentage increases whereas in misuse, detection rate is found to decrease and in anomaly detection rate remains constant. In order to alleviate the disadvantages in both of the approaches and maximizing the detection rate as well as to identify new attacks, hybrid Intrusion detection systems, combining the advantages of both the misuse and anomaly based approaches have been used.

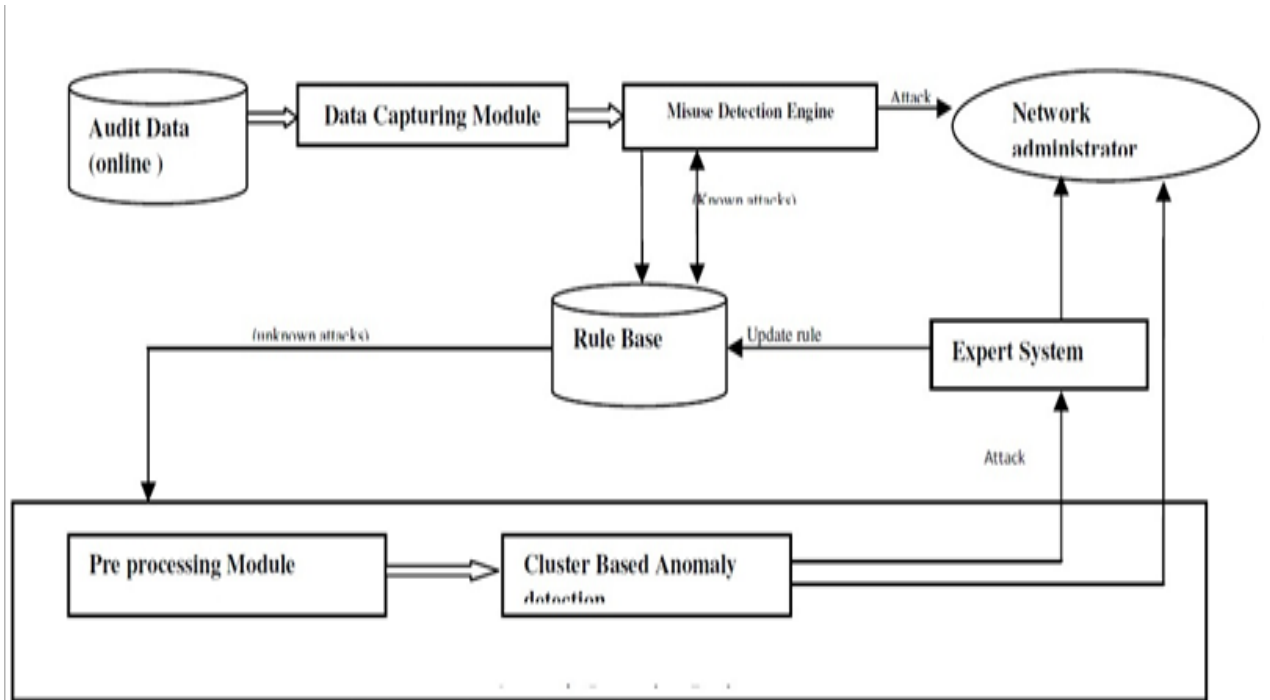
Different machine learning algorithms, data mining approaches and artificial intelligence (AI) techniques have been used to identify intrusions

The framework is divided into different modules such as data capturing, misuse detection, Rule Base, data pre-processing, clustering techniques and expert system.

The proposed frame work is done with the Global Mobile Information System Simulator (GloMoSim) which is a scalable simulation environment for the large wired and wireless communication networks.



# DIAGRAM -



**Fig 4 - System Architecture of Hybrid NIDS**

## **2.3. TITLE -**

### **Administrative Evaluation of Intrusion Detection System**

#### **SUMMARY -**

In this research, a methodology is developed to evaluate intrusion detection systems in a simulated environment. The environment is built with a combination of physical and virtual machines. Network traffic is simulated with baseline activities, which is characterized with web browsing and normal user activities, benchmark and actual intrusion attacks. Three IDSs from the open source domain were chosen for comparison (Snort Ourmon and Samhain )

A physical network was set up to simulate a real-world computing environment of a small or middle-sized business. Common network intrusions along with baseline and benchmark network traffic were injected into the simulated network

Results show that the Snort system imposes noticeably more impacts on network traffic than the other two tested IDSs. It could be a bottleneck on a high speed network. In addition, Snort occupies more system memory than the other two tested systems. However, it generates lower CPU load. The Samhain system imposes the most CPU load among the tested IDSs

## **2.4. TITLE -**

### **Real- Time Malware Detection Framework in Intrusion Detection System**

#### **SUMMARY -**

The framework generates signatures from malware families and generates corresponding detection rules. The generated signatures are not influenced by small changes of malware while they can be used to detect malware that has similar behaviors with normal programs. Our signatures are stored as an Aho-Corasick Tree form to improve signature matching performance in IDS.

The detection rule generation algorithm extracts as signatures from known malware and generates rules to identify payloads that contain parts of malware. A signature of each malware family is re-created from extracted signatures common in malware variants.

This research assumed that packets are arrived in a correct order. If not, failed offset calculation may affect the detection accuracy of the algorithm

## DIAGRAM -

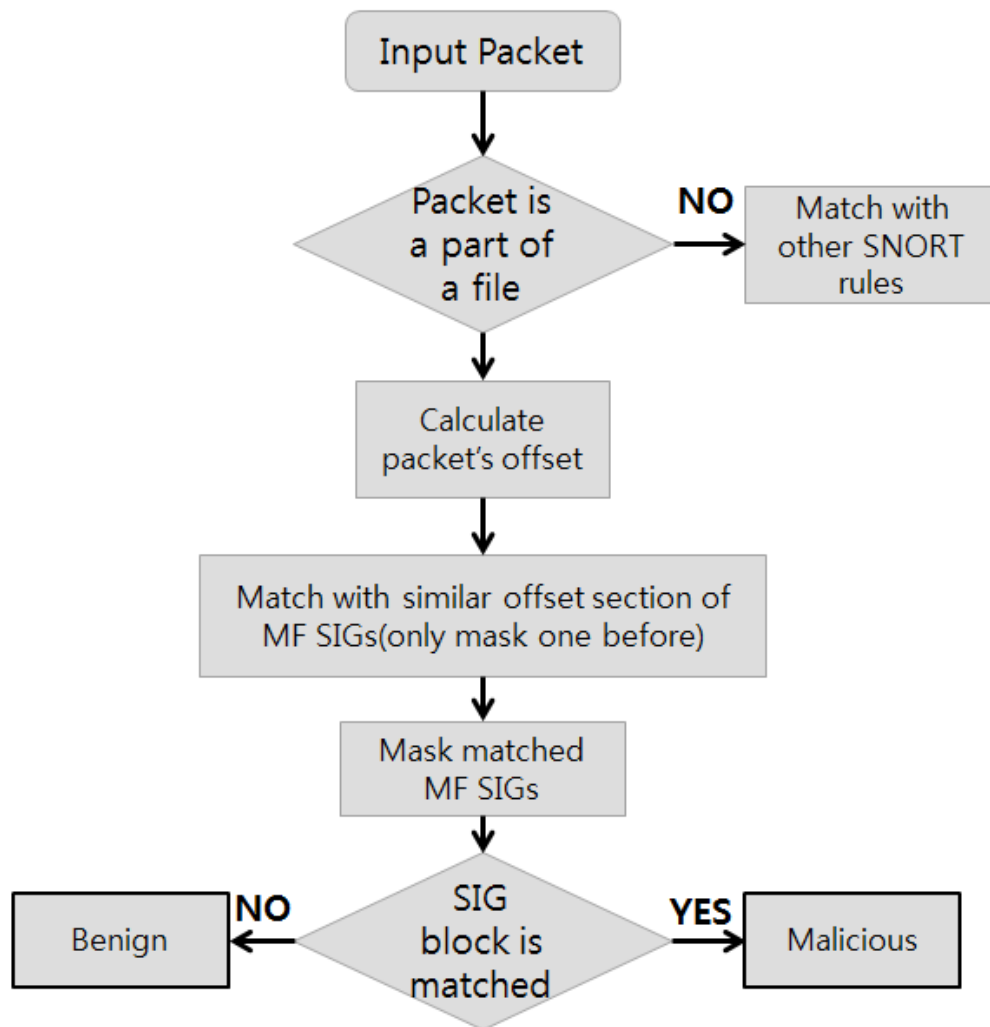


Fig 5 – Flow Chart of Matching Algorithm

## 2.5. TITLE

### **Enhancement of Fault Tolerance of Intrusion Detection System using AES and DES based Heart Beat Events**

#### **SUMMARY -**

IDS are vulnerable to various attacks like:

1. **Component Failure:** A major component fails to perform to specification.
2. **Component Corruption:** Components are corrupted.
3. **Communication Failure:** Messages are not received accurately and/or in a timely way.
4. **Sensor Failure:** Sensors fail to provide accurate observations of all relevant conditions.

Sending the “heartbeat” messages from the host to prove that it is executing and is alive is an inherently insecure endeavor, due to the fact that host available resources are vulnerable to attackers which have gained privileged access levels on the host and can easily emulate these messages and moreover,

This work proposes a mechanism to enhance the fault tolerance of IDS using Advanced Encryption Standard and Data Encryption Standard methods to encrypt these heartbeat messages. Time taken to hack the heartbeat message, which is encrypted using AES encryption algorithm is much more than the time taken to hack the message when it is encrypted using DES encryption. Thus the attacker will be unable to replicate the contents of the heartbeat message within the fixed time interval when the next heart beat message will be sent because the heartbeat message is sent periodically and can be changed after some period of time. This would lead to detection of any malicious access by the attacker.

## 2.6. TITLE -

# **Kargus: A Highly-scalable Software-based Intrusion Detection System**

## SUMMARY -

First, Kargus batch processes incoming packets at network cards and achieves up to 40 Gbps input rate even for minimum-sized packets.

Second, it exploits high processing parallelism by balancing the pattern matching workloads with multicore CPUs and heterogeneous GPUs, and benefits from extensive batch processing of multiple packets per each IDS function call.

Third, Kargus adapts its resource usage depending on the input rate, significantly saving the power in a normal situation.

Our evaluation shows that Kargus on a 12-core machine with two GPUs handles up to 33 Gbps of normal traffic and achieves 9 to 10 Gbps even when all packets contain attack signatures, a factor of 1.9 to 4.3 performance improvements over the existing state-of-the-art software IDS.

Two basic techniques that we employ for high performance are

- (i) **Batch processing** - batching in receiving packets allows a high input rate by reducing the per-packet CPU cycle and memory bandwidth cost
- (ii) **Parallel execution** with an intelligent load balancing algorithm- efficiently balancing the load of flows across multiple CPU cores and by employing a large array of GPU processing cores.

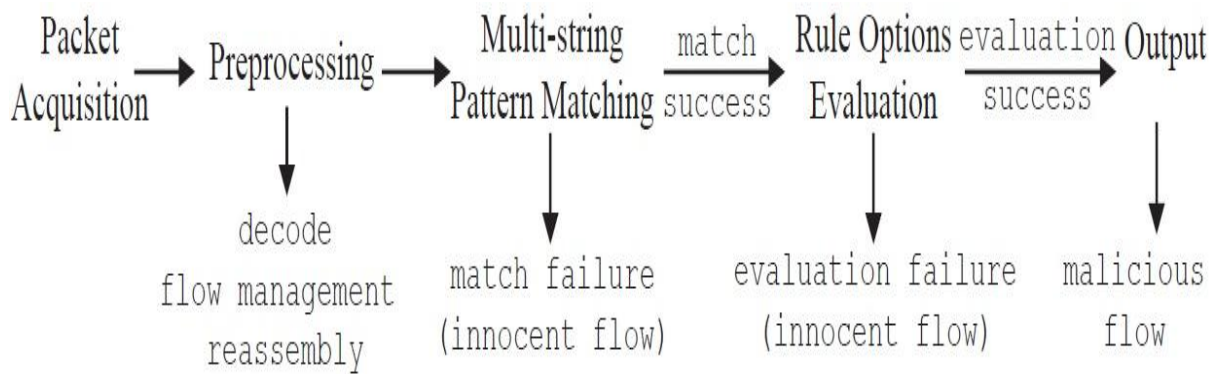
### 2.6.1 Attack Signature Matching

Phase 1: Multi-string pattern matching.

Phase 2: Rule option evaluation.

Kargus consists of multiple CPU threads where each thread reads and processes packets from NIC queues affinitized to it by RSS.

**DIAGRAM -**



**Fig 6 - Typical Intrusion Detection Process in Signature-based IDS**

## **2.7. TITLE –**

### **A distance sum-based hybrid method for intrusion detection**

#### **SUMMARY –**

Intrusion detection systems based on a hybrid approach have attracted considerable interest from researchers. Hybrid classifiers are able to provide improved detection accuracy, but usually have a complex structure and high computational costs.

In this research, a new and easy-to-implement hybrid learning method is proposed, named distance sum-based support vector machine (DSSVM), which can be used as an effective intrusion detection model. In DSSVM, the distance sum is introduced, a correlation between each data sample and cluster centers.

Consider a data set represented by  $n$ -dimensional feature vectors, each distance sum for a data sample in the data set is obtained from the distances between this data sample and  $k - 1$  of  $k$  cluster centers found by a clustering algorithm. A new data set representing the features of these distance sums is formed and used to train a support vector machine classifier.

By applying DSSVM to the KDD'99 data set, experimental results show that the proposed hybrid method performs well in both detection performance and computational cost, which suggests it, is a competitive candidate for intrusion detection.



## **2.8. TITLE –**

### **A Clustering based Algorithm for Network Intrusion Detection**

#### **SUMMARY –**

The secure information transmission is very important in the present scenario. Many intrusion detection system (IDS) have been developed in recent past which are based on either signature information or anomaly information. But all these systems do generate a lot of false detections.

In this work a hybrid IDS is being proposed which uses the signature and anomaly information together. The proposed algorithm first explore those traffic features which are changing during an intrusion activity and then based on a predefined threshold value the most prominent features related to attack are identified.

Thereafter, these features are included in snort rule set to detect the anomalous traffic. This anomaly detection process is combined with existing signature of snort to produce the better detection.

## **2.9. TITLE –**

### **Divided two-part adaptive intrusion detection system**

#### **SUMMARY –**

The paper presents an efficient approach for reducing the rate of alerts using divided two-part adaptive intrusion detection system (DTPAIDS). The proposed DTPAIDS has a high degree of autonomy in tracking suspicious activity and detecting positive intrusions. The proposed DTPAIDS is designed with the aim of reducing the rate of detected false positive intrusion through two achievements.

The first achievement is done by implementing adaptive self-learning neural network in the proposed DTPAIDS to give it the ability to be automatic adaptively system based on Radial Basis Functions (RBF) neural network. The second achievement is done through dividing the proposed intrusion detection system IDS into two parts.

The first part is IDS1, which is installed in the front of firewall and responsible for checking each entry user's packet and deciding if the packet considered is an attack or not. The second is IDS2, which is installed behind the firewall and responsible for detecting only the attacks which passed the firewall.

This proposed approach for IDS exhibits a lower false alarm rate when detects novel attacks.

## 2.10 SUMMARY

<b>Paper ID</b>	<b>Title</b>	<b>Proposal</b>
A Distributed Multi-Approach IDS for Web Server	A Distributed Multi-Approach IDS for Web Server	<ul style="list-style-type: none"> <li>- Using both anomaly and misuse detection techniques</li> <li>- Distributed architecture for web service hosts and WS-IDS-mitigates processing overhead, lower risk of DOS attack</li> </ul>
Hybrid NIDS using expert rule based approach	Hybrid NIDS using expert rule based approach	<ul style="list-style-type: none"> <li>- Combining advantages of misuse and anomaly based detection using machine learning algorithms, data mining approaches and AI techniques</li> </ul>
Administrative evaluation of IDS	Administrative evaluation of IDS	<ul style="list-style-type: none"> <li>- Evaluate IDS in simulated environment</li> </ul>
Real time Malware Detection Framework in IDS's	Real time Malware Detection Framework in IDS's	<ul style="list-style-type: none"> <li>- Detection rule generation algorithm – rules to identify payloads that contain parts of malware</li> </ul>
Enhancement of Fault Tolerance of IDS using AES and DES based heart beat events	Enhancement of Fault Tolerance of IDS using AES and DES based heart beat events	<ul style="list-style-type: none"> <li>- Intent is to hide and authenticate IDS communications to secure and ensure the reliability of IDS using encryption by AES and DES</li> </ul>

Kargus: A highly scalable software based IDS	Kargus: A highly scalable software based IDS	<ul style="list-style-type: none"> <li>- Implementing batch processing in receiving packets reducing per packet CPU cycle</li> <li>- Parallel execution with intelligent load balancing across multiple CPU cores</li> </ul>
A distance sum-based hybrid method for intrusion detection	A distance sum-based hybrid method for intrusion detection	<ul style="list-style-type: none"> <li>- Distance sum is introduced which is a correlation between each data sample and cluster centers.</li> <li>- distance sum for a data sample in the data set is obtained from the distances between data sample</li> <li>- A new data set representing the features of these distance sums is formed and used to train a support vector machine classifier</li> </ul>
A Clustering based Algorithm for Network Intrusion Detection	A Clustering based Algorithm for Network Intrusion Detection	<ul style="list-style-type: none"> <li>- Signature and anomaly information are used together.</li> <li>- First explores those traffic features that change during an intrusion activity and then based on a predefined threshold value most prominent features related to attack are identified.</li> <li>- Thereafter, these features are included in snort rule set to detect the anomalous traffic</li> </ul>

Divided two-part adaptive intrusion detection system	Divided two-part adaptive intrusion detection system	<ul style="list-style-type: none"> <li>- Adaptive self learning neural network is implemented</li> <li>- IDS1 is installed in front of firewall and IDS2 is installed behind the firewall</li> </ul>
--	--	--

**Table 1 – Tabular representation of summary of Research Papers**

## **2.11 SUMMARY**

After studying the above research papers we get to know that different architectures for a Network Intrusion Detection System (NIDS), Host Intrusion Detection System (HIDS) and Behavior Intrusion Detection System (BIDS) have been proposed.

As we can see that in Paper 1, 2 and 4 we have used signature based and anomaly based methods to detect various network attacks. But we have not come through any research paper that proposes the combination of NIDS, HIDS and BIDS so that their disadvantages can be coped up with.

So, in this paper, a model is proposed combining the advantages of NIDS and HIDS to combat their shortcomings.

## **CHAPTER 3**

### **PROBLEM STATEMENT**

## Comparative analysis of HIDS and NIDS

Function	Comments
Protection on LAN	Both systems protect you on your LAN
Protection off LAN	Only HIDS protects you when you are off the LAN
Ease of Administration	The admin of NIDS and HIDS is equal from a central admin perspective.
Versatility	HIDS are more versatile systems.
Price	HIDS are more affordable systems if the right product is chosen.
Ease of Implementation	Both NIDS and HIDS are equal from a central control perspective
Little Training required	HIDS requires less training than NIDS
Bandwidth requirements on (LAN)	NIDS uses up LAN bandwidth. HIDS does not.
Bandwidth requirements (internet)	Both IDS need internet bandwidth to keep the pattern files current
Spanning port switching requirements	NIDS requires that port spanning be enabled to ensure that your LAN traffic is scanned.
Update frequency to clients	HIDS updates all of the clients with a central pattern file.
Cross platform compatibility	NIDS are more adaptable to cross platform environments.
Local machine registry scans	Only HIDS can do these types of scans.

Logging	Both systems have logging functionality
Alarm functions	Both systems alarm the individual and the administrator.
PAN scan	Only HIDS scan you personal area networks. (unless you have the \$ to get a NIDS for your home)
Packet rejection	Only NIDS functions in this mode.
Central management	NIDS are more centrally managed.
Disable risk factor	NIDS failure rate is much higher than HIDS failure rate. NIDS has one point of failure.
Upgrade potential	It is easier to upgrade software than hardware. HIDS can be upgraded through a centralized script. NIDS is typically flashed onto the flash memory and has low overhead.
Multiple LAN detection nodes	HIDS is a more comprehensive multiple segment detection IDS than NIDS

**Table 2- Comparative Study of NIDS and HIDS**

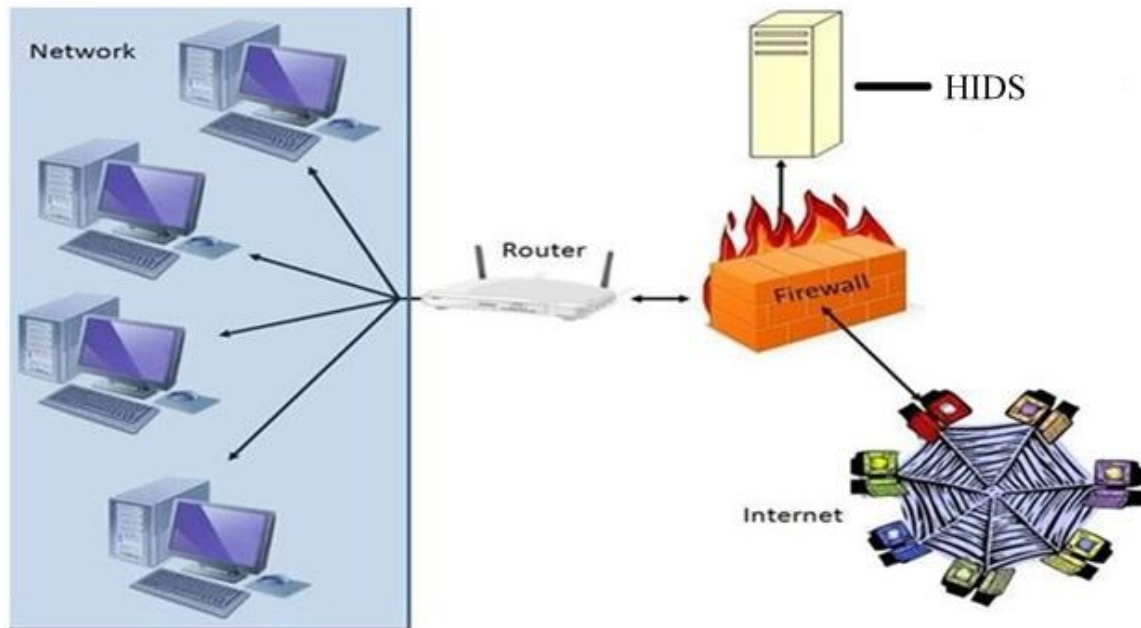
After looking at the above features of both NIDS and HIDS our problem statement is to design a Hybrid Intrusion Detection System that overcomes the disadvantages of both NIDS and HIDS



## **CHAPTER 4**

# **PROPOSED MODEL**

In order to overcome network and request based attacks over a network and develop an ability to detect both types of attacks, we propose a Hybrid Intrusion Detection System combining advantages of both HIDS and NIDS.



**Fig 7 – Proposed Architecture of Hybrid IDS**

Features of the proposed model:

1. As we know that NIDS examines all packet headers for signs of malicious and suspicious activity whereas HIDS do not see packet headers, so they cannot detect these types of attacks. For example, many IP-based denial of-service (DOS) and fragmented packet (Tear Drop) attacks can only be identified by looking at the packet headers as they travel across a network. This type of attack can be quickly identified by a network-based system looking at the packet stream in real-time. Hence by combining both NIDS and HIDS these kind of attacks could also be detected.

2. Real-time detection could be done. We know that NIDS detect malicious and suspicious attacks as they occur, and so provide faster notification and response. For example, a hacker initiating a network based denial of service (DOS) based on TCP can

be stopped by having a network-based IDS send a TCP reset to terminate the attack before it crashes or damages a targeted host. Host-based systems usually do not recognize an attack or take action until after a suspicious log entry has been written. By this time, critical systems may already be compromised, or the system running the host-based IDS may have crashed. Real-time notification allows rapid reaction according to predefined parameters. Hence, combining both NIDS and HIDS would help to cope with this problem.

3. An NIDS placed outside of a firewall can detect attacks intended for resources behind the firewall, even though the firewall may be rejecting these attempts, whereas an HIDS do not see rejected attacks that never hit a host inside the firewall. So, a Hybrid IDS can make use of this information in evaluating and refining security policies.

4. Since an HIDS uses logs containing events that have actually occurred, they can measure whether an attack was successful or not with greater accuracy and fewer false positives than network-based systems. In this respect, an HIDS makes an excellent complement to network-based intrusion detection, with the network component providing early warning and the host component providing verification of whether an attack was successful or not. Hence, a Hybrid IDS would be able to give an early warning as well as verify success or failure of an attack.

5. Host-based systems can monitor changes to key system files and executables. Attempts to overwrite vital system files, or to install Trojan horses or backdoors, can be detected and stopped. Network-based systems sometimes miss this kind of activity. Therefore, a Hybrid IDS won't miss such activities.

6. Attacks from the keyboard of a critical server do not cross the network, and so cannot be seen by a network-based intrusion detection system but are detected by an HIDS. Also, certain types of encryption also present challenges to network-based intrusion detection. Depending where the encryption resides within the protocol stack, it may leave a network based system blind to certain attacks. Host-based IDS do not have this limitation. So, such kind of problems can also be handled by a Hybrid IDS.

7. Also, our proposed model would be variable, i.e, if our network/system has more of host based attacks or more of network based attacks, then it could be able to set to those conditions/requirements itself.

## **4.1 FUNCTIONALITY**

### **4.1.1 Jpcap**

Jpcap has been used in this project for capturing as well as generating traffic.

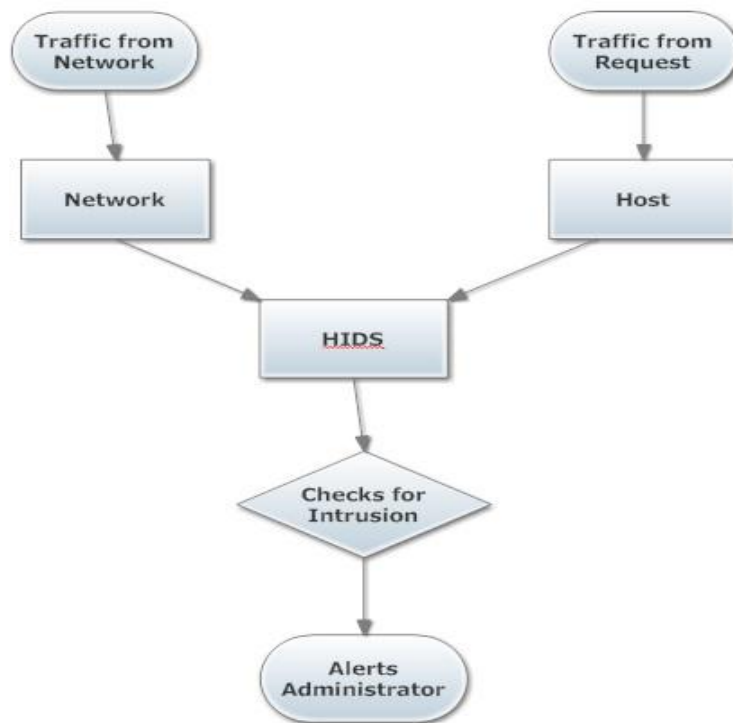
Jpcap is a set of Java classes which provide an interface and system for network packet capture. A protocol library and tool for visualizing network traffic is included. Jpcap hides the low-level details of network packet capture by abstracting many network packet types and protocols into Java classes. Internally, jpcap implements bindings to the libpcap system library through JNI (Java Native Interface). Therefore, for using Jpcap, we need to install libpcap (Winpcap in Windows) first.

Using Jpcap, you can develop applications to capture packets from a network interface and visualize/analyze them in Java. You can also develop Java applications to send arbitrary packets through a network interface.

Jpcap has been tested on Microsoft Windows (98/2000/XP/Vista), Linux (Fedora, Mandriva, Ubuntu), Mac OS X (Darwin), FreeBSD, and Solaris. Jpcap can capture Ethernet, IPv4, IPv6, ARP/RARP, TCP, UDP, and ICMPv4 packets. Jpcap is open source, and is licensed under GNU LGPL.

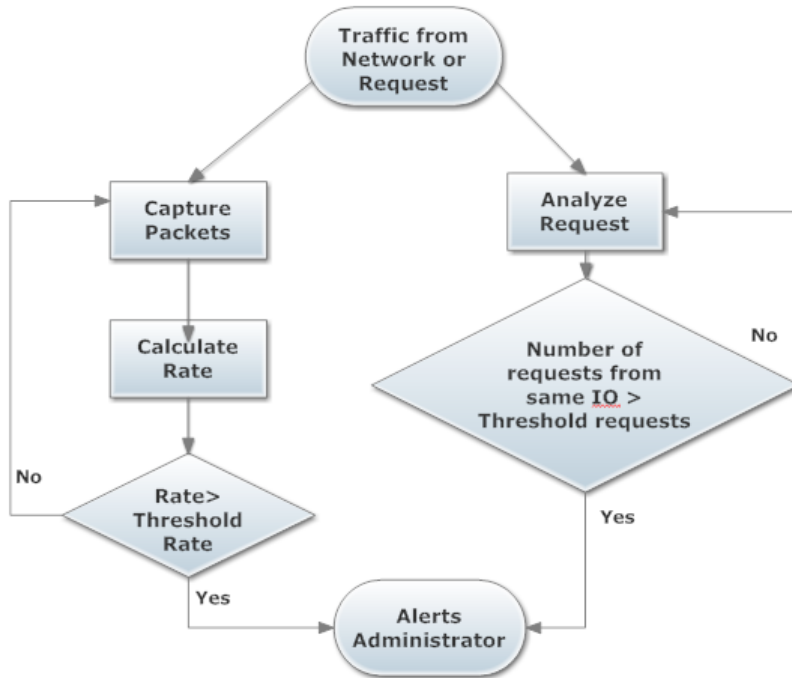
Jpcap captures and sends packets *independently* from the host protocols (e.g., TCP/IP). This means that Jpcap does not (cannot) block, filter or manipulate the traffic generated by other programs on the same machine: it simply "sniffs" the packets that transit on the wire. Therefore, it does not provide the appropriate support for applications like traffic shapers, QoS schedulers and personal firewalls.

## 4.1.2 Flowcharts



**Fig 8 – Flowchart showing the larger view of path that traffic follows**

The above flowchart the basic overall functioning of our Hybrid Intrusion Detection System. The traffic generated on a Network is captured as well as the traffic generated from an HTTP request is taken by the host, and both the types of request are analyzed by the HIDS which further after checking for any Intrusion alerts the administrator in case any Intrusion is detected. The working of HIDS is explained in the next flowchart.



**Fig 9 – Flowchart showing how the traffic is handled by HIDS**

This flowchart is the larger version of the previous flowchart. Here it is clearly shown how our HIDS works. Side by side, traffic from network as well as the HTTP requests is captured. For testing purposes, we have written a program that sends packets from a defined IP Address to another given IP Address stated that both are online. The network packets that are captured are bifurcated and all their information is displayed on the console. This information is stored in the database along with the arrival time of the packets. In another table, we keep a track of the number of packets arriving in every 10 seconds. This can be treated as rate of packets per 10 seconds. If this rate is greater than the predefined threshold, then an alert message is shown to the administrator. Alongside, the HTTP requests are entertained. The browser Information of client, Client IP Address, Server IP Address as well as Client MAC address is displayed to the administrator. If the request is coming from an IP Address that has not been listed earlier in the database, then it is listed first. If the request comes from an already existing IP Address and the requests keep on coming within an interval of 2 seconds each and they reach a threshold, then again an alert is sent to the administrator regarding Intrusion.

### 4.1.3 Pseudo Code

```
1. //Generate Packets (TCP)
2.   Send_packet()
3. While (Capture Packets)
4. {
5.     //Display packet information on console
6.     receivePacket()
7.     //Insert all the packet information in a table
8.
9.     insert packets captured in every 10 seconds
11->}
```

A.

**Fig 10 – Pseudo code of sending and capturing packets**

This is the pseudo code of the programs that helps in sending as well as capturing the network packets. First of all, from class Send\_packet.java, we send a variable number of TCP packets from a defined source to a defined destination. Then in class packet\_capture.java, we open connection to our database named “project” and truncate the tables’ packet and rate so that they don’t contain any previously recorded values. After that, we get the list of available Network Interfaces and open any one of them. Then, we calculate the initial time and set the capture filter to “TCP”. As the packets get captured, all their information gets bifurcated and printed on the console. This information is also inserted into a table called ‘packet’ and in another table named ‘rate’, number of packets captured in every 10 seconds are recorded.

```

1. // Input username and password in a login form
2. index.html
3. //Open connection with database
4. DatabaseAccess.java
5. If( username or password not correct)
6. {
7.     Go back to login page
8. }
9. Else
10. {
11.     //Open the connection with database
12.     openConnection()
13.     Get the IP Address of the client
14.     If( IP Address already not there)
15.     { Insert IP address ,the time of request as well as set i=0 in table
16.     }
17.     Else
18.     {
19.         Check the time when this IP was used last time
20.         Check the current request time
21.         Calculate the difference between the 2 times
22.         Update the current time into the table
23.         If(The difference in time > 2 seconds)
24.         {
25.             i=0
26.         }
27.         Else
28.         {
29.             Retrieve value of I from database and increment it
30.             Update this value of I in table
31.             If(value of I exceeds 5)
32.             {
33.                 i =0
34.                 Dispatch a request to a servlet that gives an INTRUSION ALERT
35.             }
36.         }
37.     }
38.     Get Browser information of Client and print
39.     Get Server IP Address and print
40.     In table rate, if(number of packets captured in any 10 seconds >800)
41.     {
42.         r=1
43.         Give an alert that Intrusion due to NIDS may be there
44.     }
45. }

```

**B**

**Fig 11 – Pseudo code showing how the request from host and network are handled**



In this pseudo code, the working of HIDS is explained. First of all, from an HTML page, 'index.html' the user is required to fill in his/her username and password. If the information is wrong, the user is asked to fill in the details again, but if the details are right, he/she is directed to a Servlet called 'DemoServlet.java'. Here, firstly, the IP Address of client is retrieved. After opening the connection with the database, it is checked that whether any request has come from that IP in the past. If not, details of that IP are inserted into the table 'ipaddress' along with the time when request was made and a variable i is initialized to 0. If request has come from an IP Address already existing in the table, then difference between the 2 requests is calculated. If the time difference is less than 2 seconds, then the variable i is incremented by 1. If the value of i reaches a predefined threshold value, then an alert is sent to the administrator regarding an Intrusion. Alongside, the MAC address and Browser Information of client as well as IP Address information of server are also retrieved and shown to the user.

# **CHAPTER 5**

## **EXPERIMENTS AND RESULTS**

## EXPERIMENTS AND RESULTS

For generating a stream of packets (here TCP ), we have used the following code with the help of Jpcap, that sends packets from a particular Source IP Address and MAC Address to a given IP Address and MAC Address

```
//open a network interface to send a packet to
NetworkInterface [] devices = JpcapCaptor.getDeviceList();

try{
JpcapSender sender=JpcapSender.openDevice(devices[2]);

//create a TCP packet with specified port numbers, flags, and
other parameters
TCPPacket p=new
TCPPacket(12,34,56,78,false,false,false,false,true,true,true,true
,10,10);

//specify IPv4 header parameters

p.setIPv4Parameter(0,false,false,false,0,false,false,false,0,1010
101,100,IPPacket.IPPROTO_TCP,InetAddress.getByName(Source IP
Address),InetAddress.getByName(Destination IP Address));

//set the data field of the packet
p.data=("data").getBytes();

//create an Ethernet packet (frame)
EthernetPacket ether=new EthernetPacket();

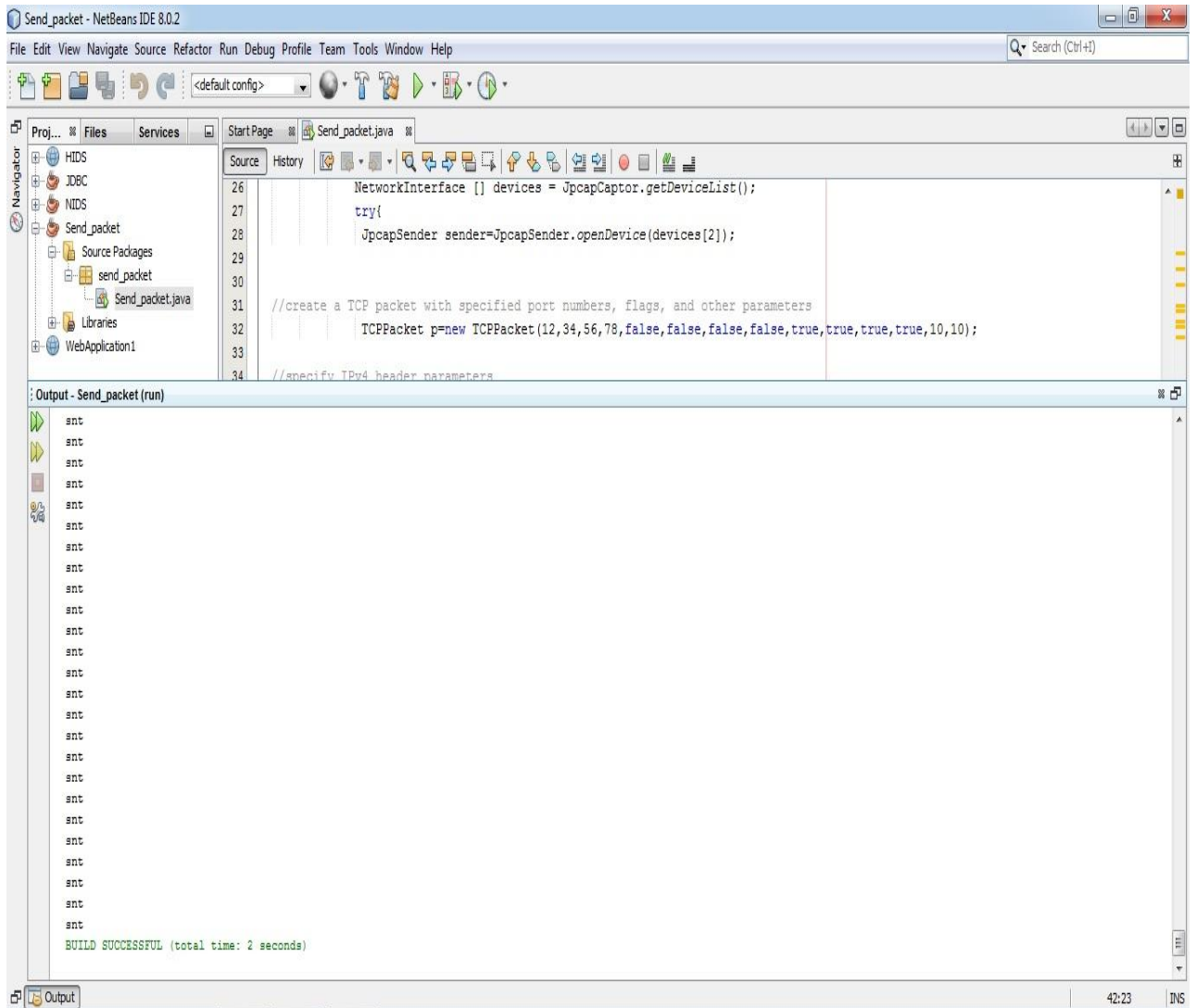
//set frame type as IP
```

```
ether.frameType=EthernetPacket.ETHERTYPE_IP;

//set source and destination MAC addresses
String strdst = new String(Destination MAC Address);
ether.dst_mac = strdst.getBytes();
String strsrc = new String(Source MAC Address);
ether.src_mac = strsrc.getBytes();

//set the datalink frame of the packet p as ether
p.datalink=ether;

//send the packet p
int i;
for(i=0;i<=2000;i++)
{
sender.sendPacket(p);
System.out.println("snt");
}
sender.close();
}
catch(IOException e){}
```

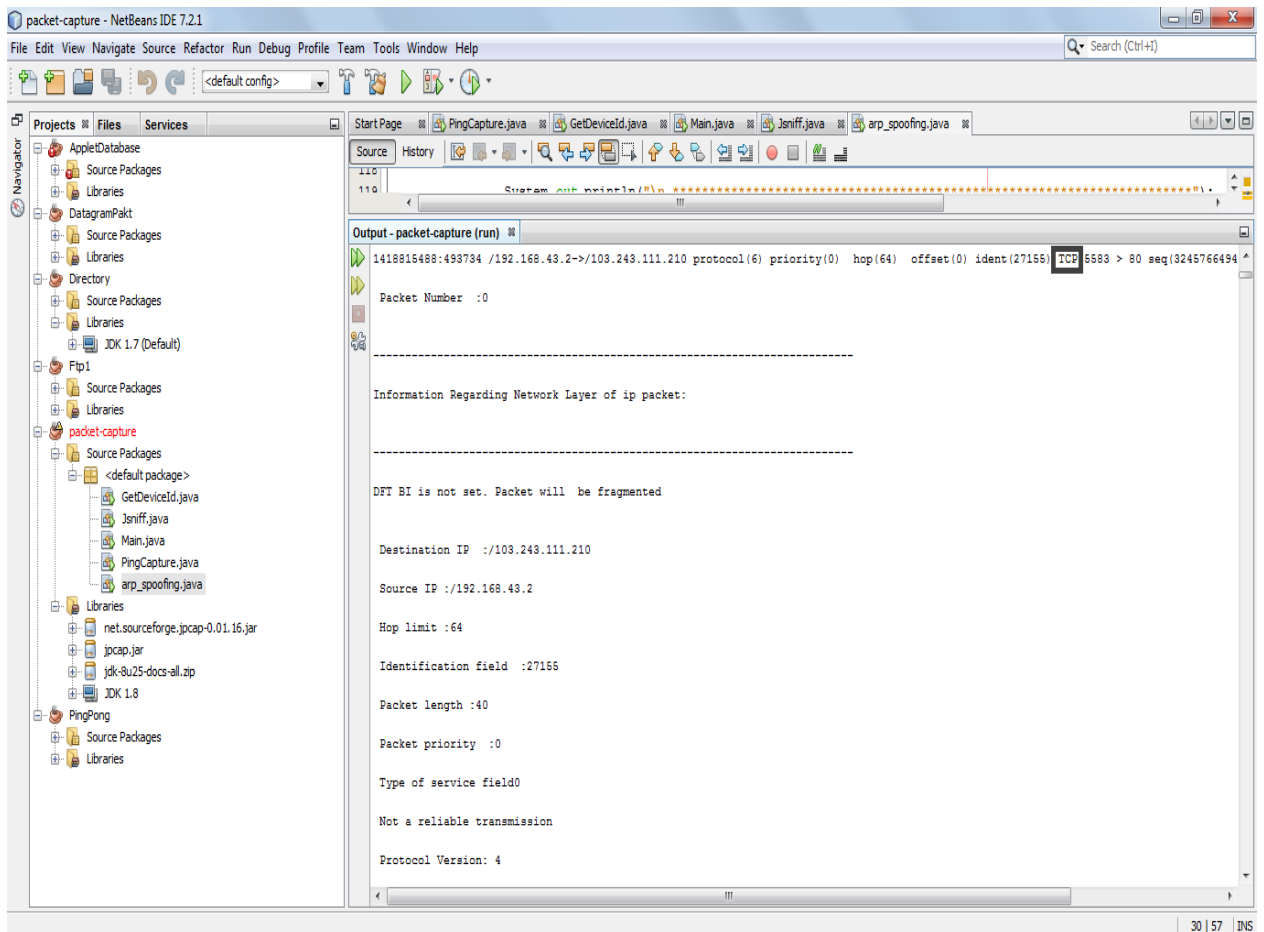


**Fig 12 – Log showing packets sent**

Here, we can see the log that shows the packets that were sent with the help of class `Send_packet.java`. Variable number of packets can be sent from a defined source to a defined destination each time this program is run. It helps in testing and debugging of the HIDS that we have made.

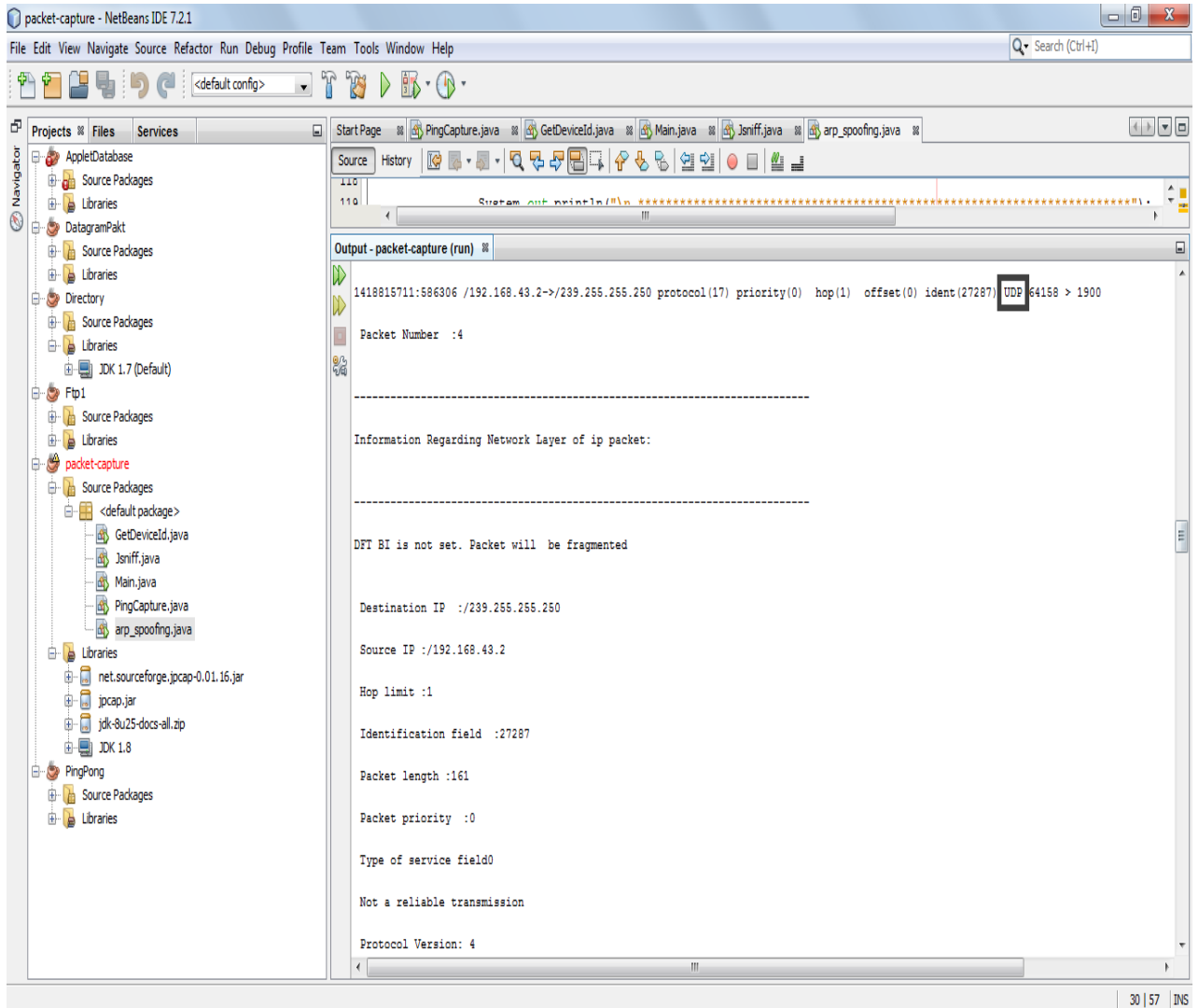
## 5.1 OUTPUTS

### NIDS-



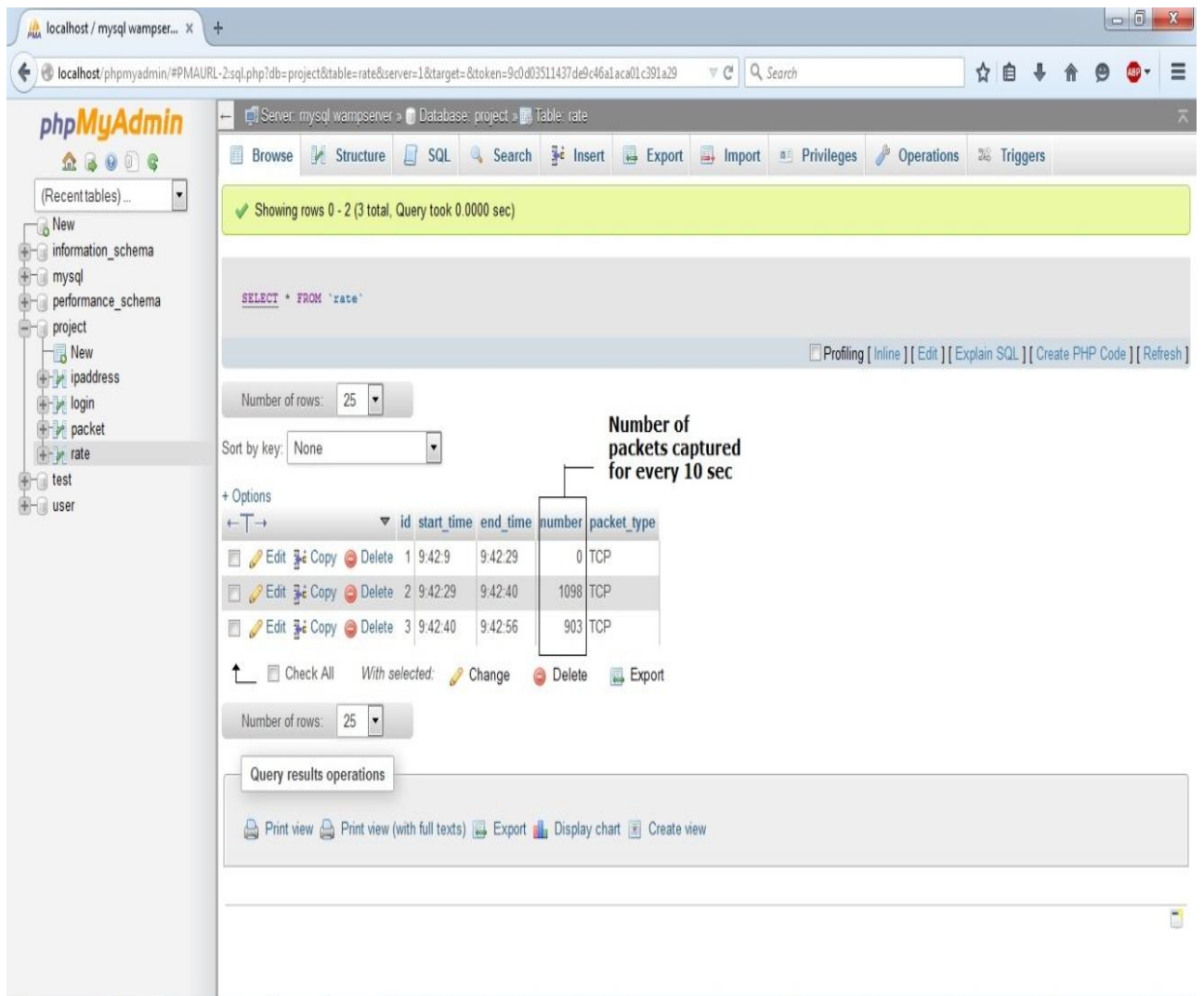
**Fig 13 – TCP packets were captured from the network with the help of jpcap library**

This output shows that how TCP packets are captured with the help of packet\_capture.java (can be the packets sent from class send\_packet.java) and their information is bifurcated and printed on the console. Side by side this information is also inserted in a table named ‘packets’.



**Fig14 – UDP packets were captured from the network with the help of jpcap library**

This output shows that how UDP packets are captured with the help of packet\_capture.java (can be the packets sent from class send\_packet.java) and their information is bifurcated and printed on the console. Side by side this information is also inserted in a table named ‘packets’.

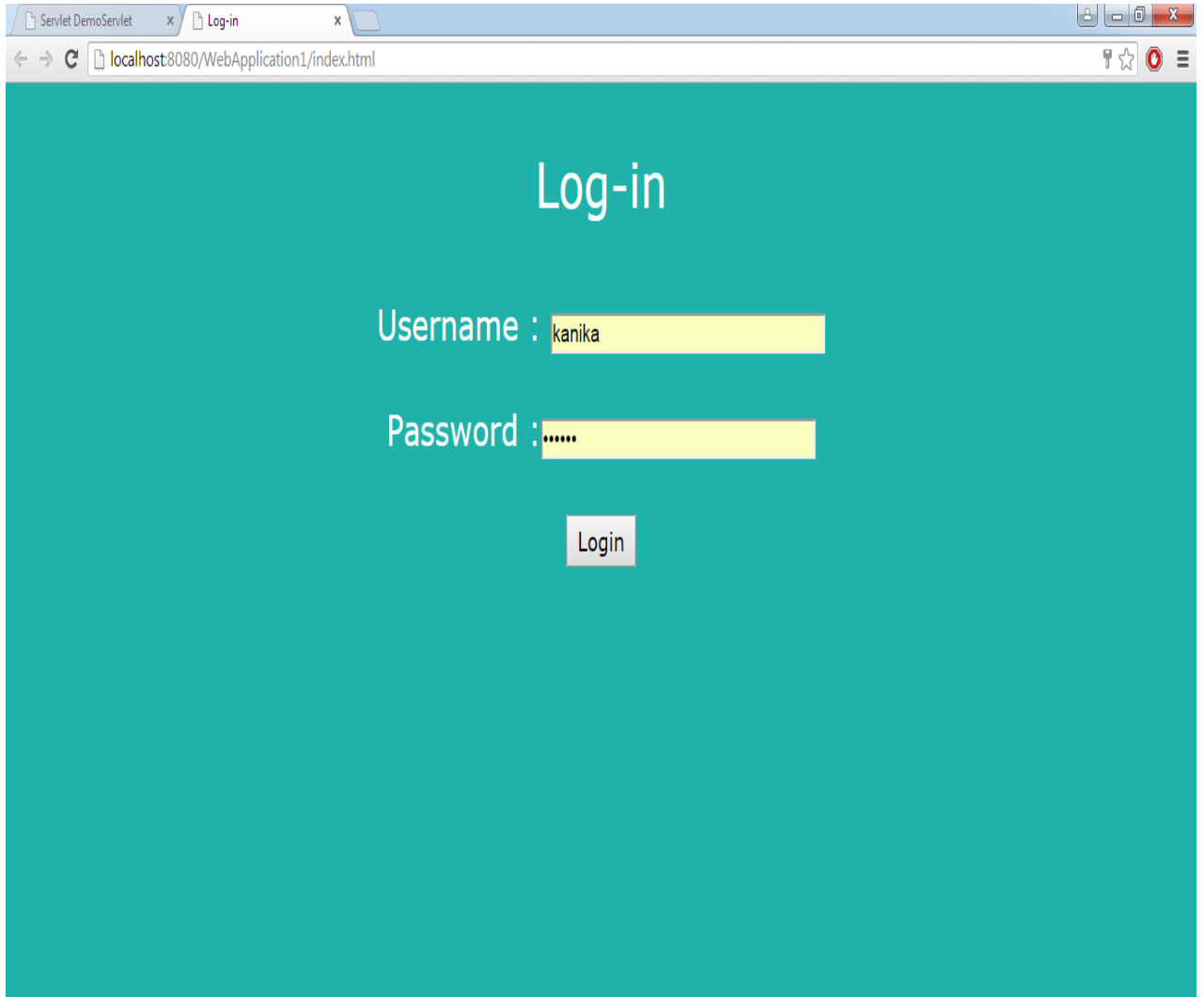


**Fig15 – Rate of packets captured every 10 seconds**

When the packets are captured by the class packet\_capture.java, their information is side by side stored in the database as well the in every 10 seconds, the number of packets captured is also recorded in a table called 'rate'. This output shows the tabular view of the table 'rate' in phpMyAdmin.



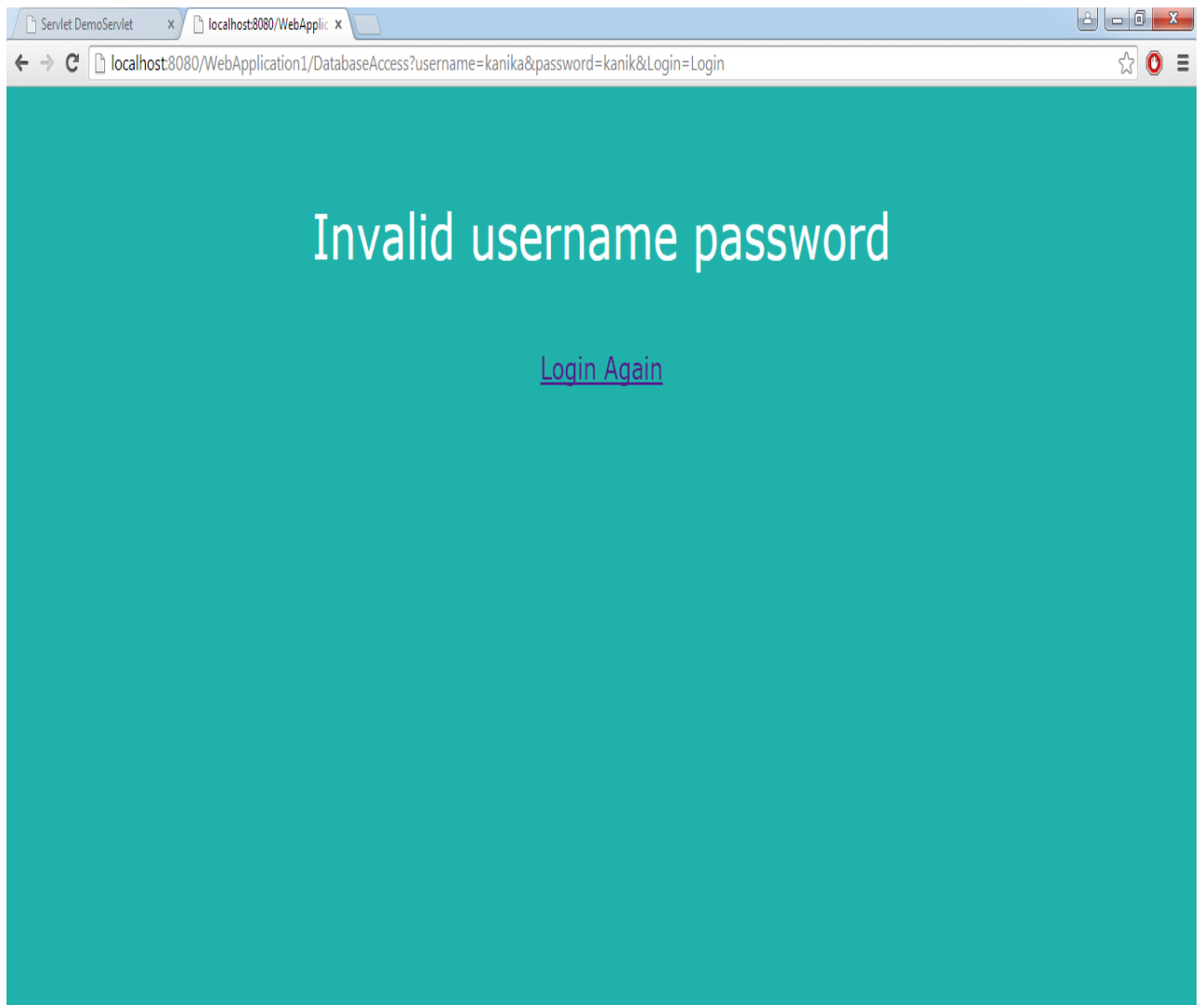
## HIDS-



The image shows a web browser window with two tabs: 'Servlet DemoServlet' and 'Log-in'. The address bar displays 'localhost:8080/WebApplication1/index.html'. The main content area has a teal background with the text 'Log-in' in white. Below this, there are two input fields: 'Username : kanika' and 'Password : .....'. A 'Login' button is positioned below the password field.

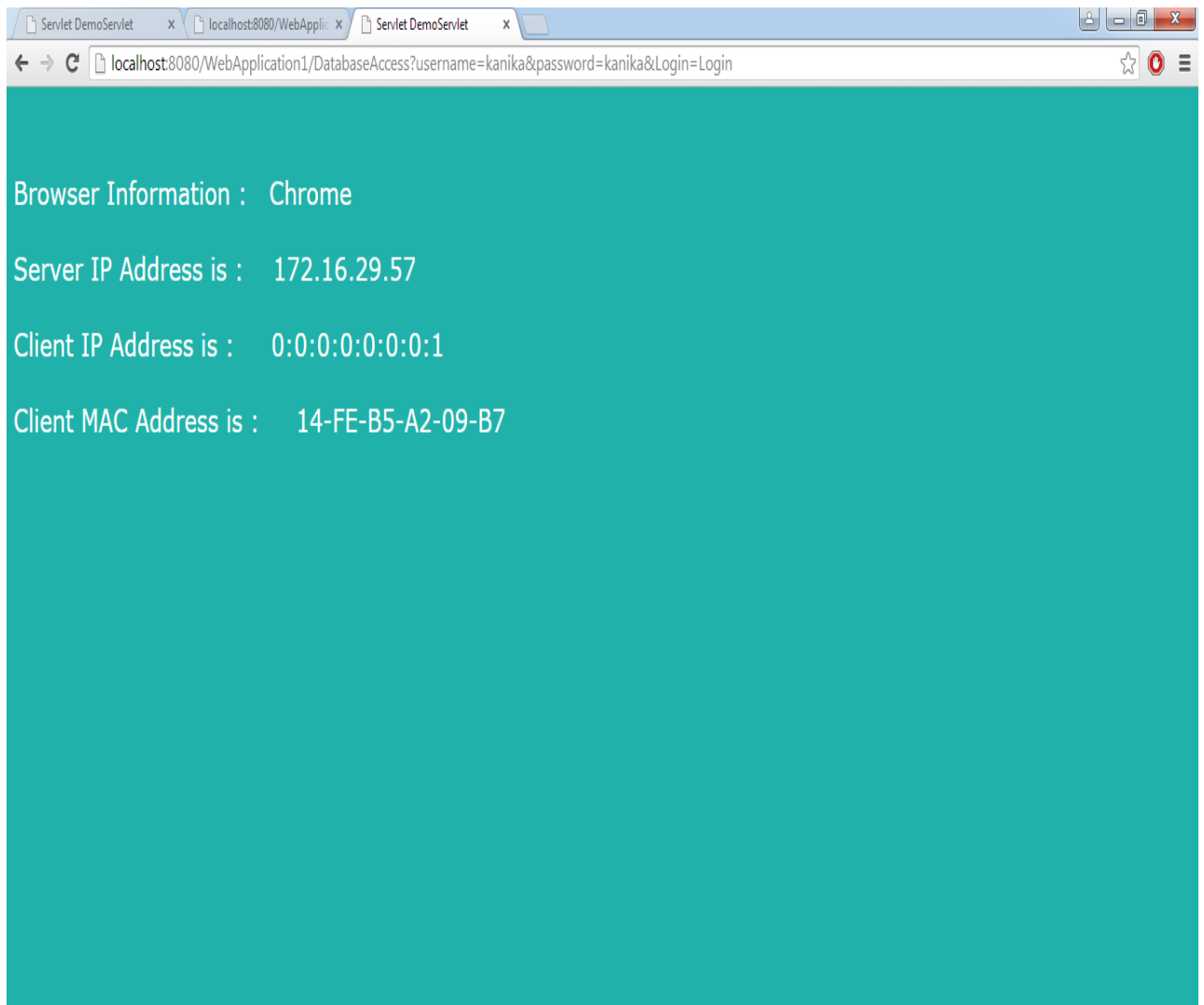
**Fig 16 – HTML Form used for sending HTTP Request so that client information can be retrieved by the server**

This is the HTML form where the user/administrator needs to login so as to check the working of HIDS and analyze the requests. The username and password are checked at the backend. If they are correct, the user is forwarded to the page showing information of requests, and if not, the user is asked to login again.



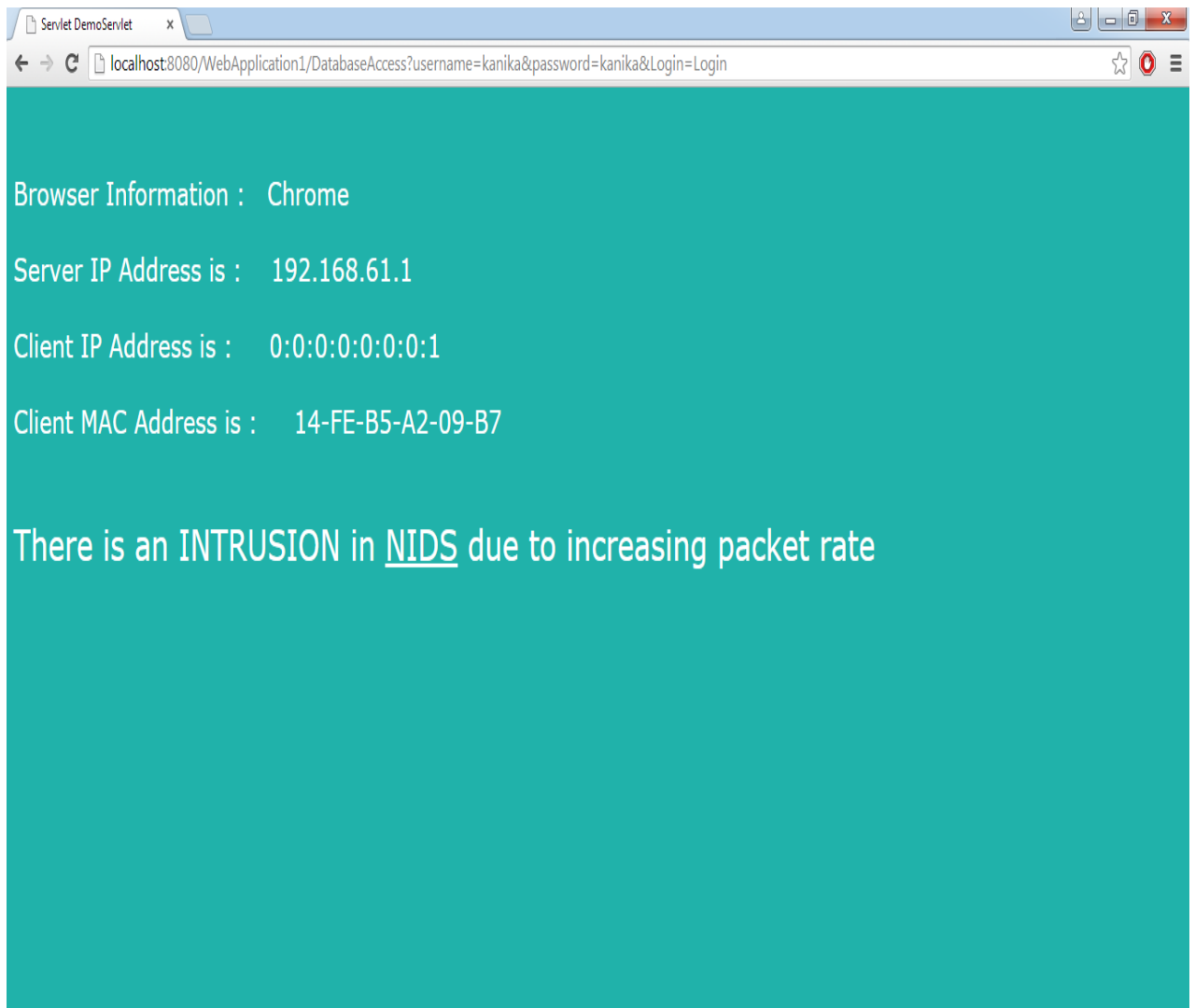
**Fig 17 – Page displaying that username or password input by user are wrong and he needs to login again**

When the user is asked to input his/her username and password, and if even one of them is wrong, then the user is directed to this page where he is asked to login again so that he can go further and analyze the incoming requests and check for any intrusions.



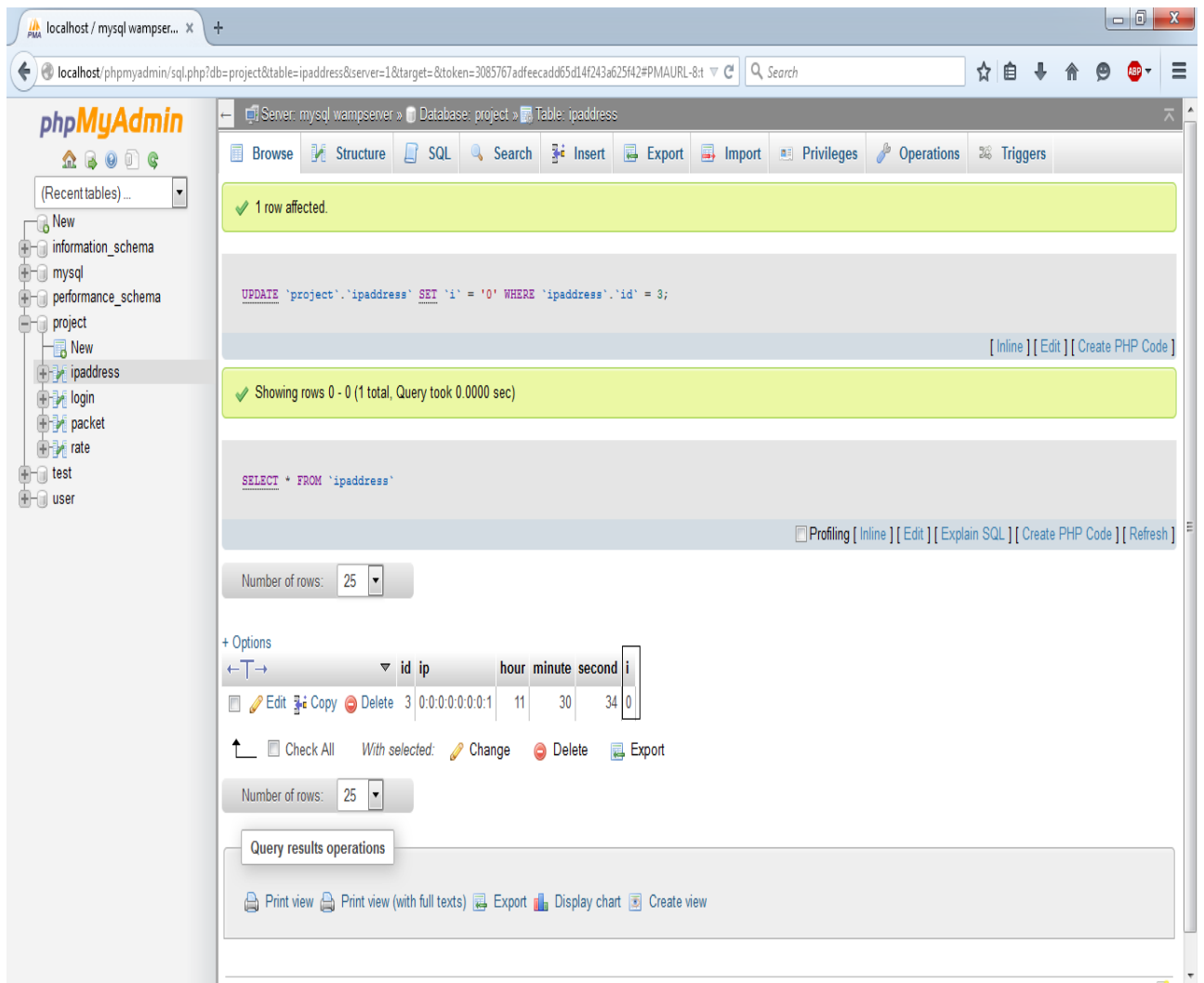
**Fig 18 – Browser Information, Server side and Client side IP Addresses retrieved from an HTTP request by the browser and displayed in Chrome**

When the user logs in with correct username and password, he/she is directed to this page. Here, the Browser information of Client. Server IP Address, Client IP Address as well as client MAC address can be seen. Whenever an HTTP request comes, all this information is displayed.



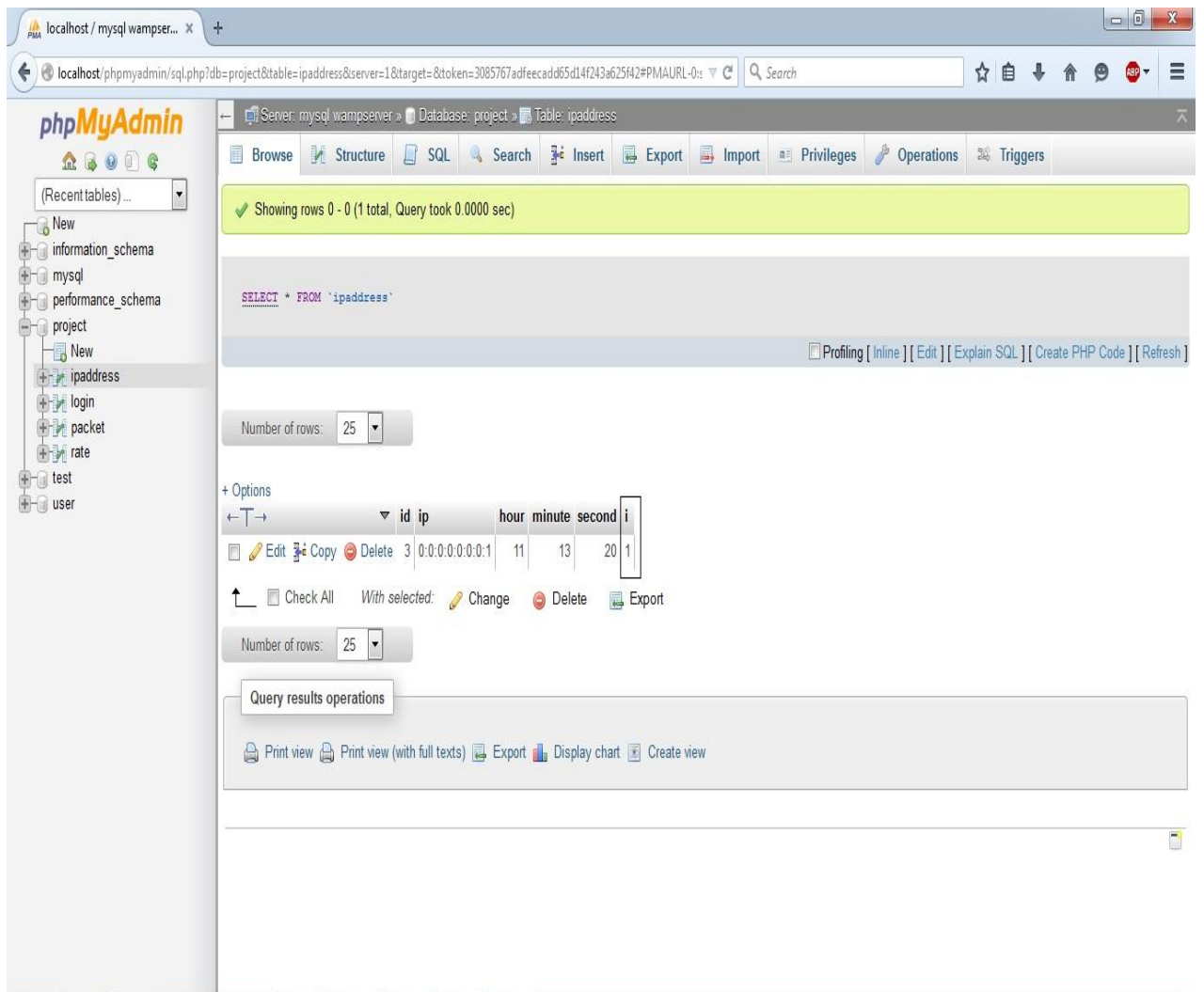
**Fig 19 – HIDS showing NIDS Intrusion Alert**

This is the same page as that of the previous one. It shows all the already discussed information except one. As we know, that in HIDS, our NIDS also run simultaneously. So if in NIDS, the packet rate increases, an alert is shown to the administrator via this page.



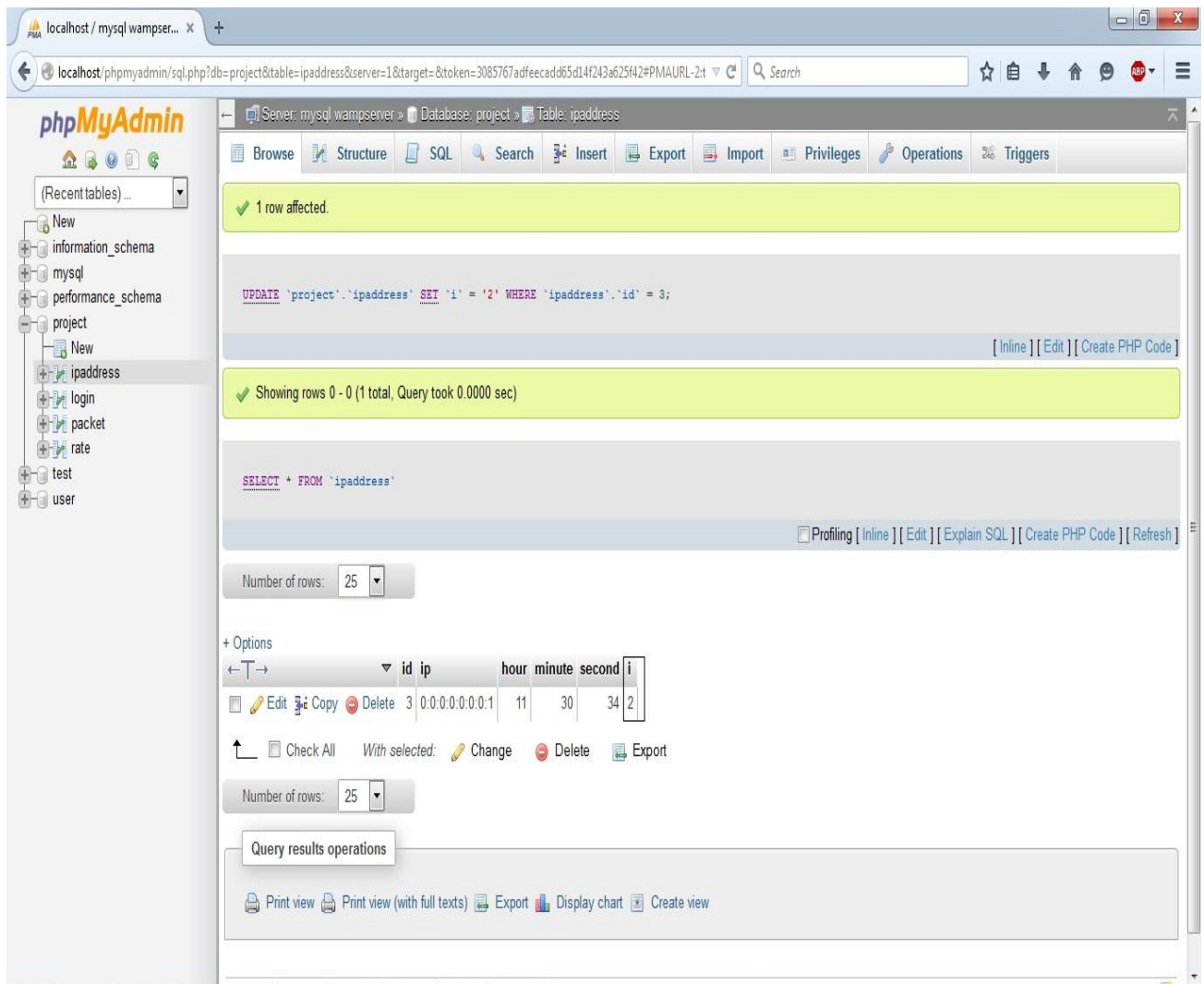
**Fig 20 – When no request from an IP is send, value of i is 0**

When an HTTP request from an IP Address is sent for the first time, then in our table 'ipaddress' it's information is shown along with a variable i which is initialized to 0. It is this variable with the help of which we are able to detect if any IP Address sends requests continuously or not.



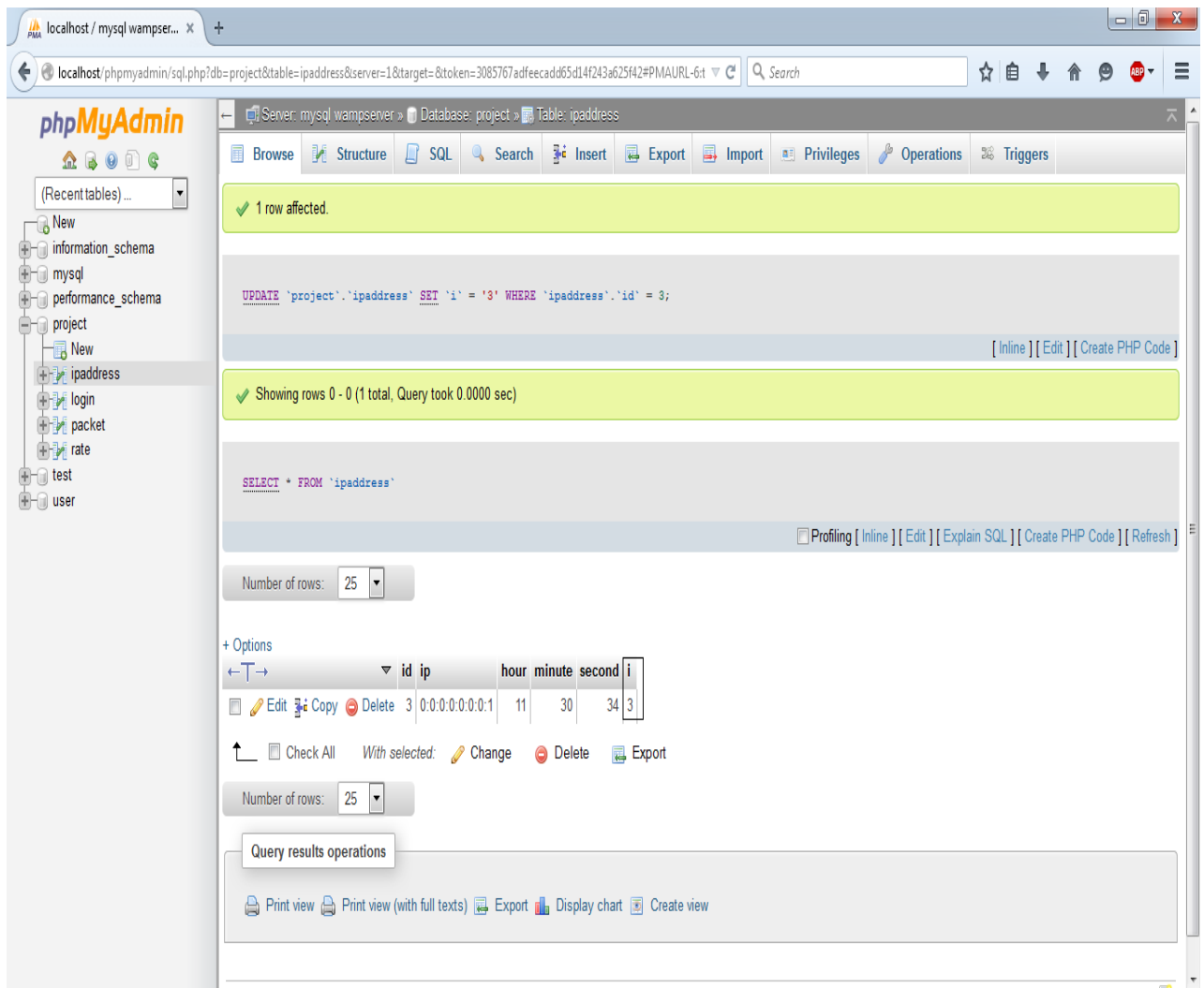
**Fig 21 – After sending first request within 2 seconds, i becomes 1**

When an HTTP request comes from an already existing IP Address within 2 seconds of previous request, then the value of i gets incremented by 1. Here, the value of i was initially 0, but after this request, it has become 1.



**Fig 22 – After sending second request within another 2 seconds, i becomes 2**

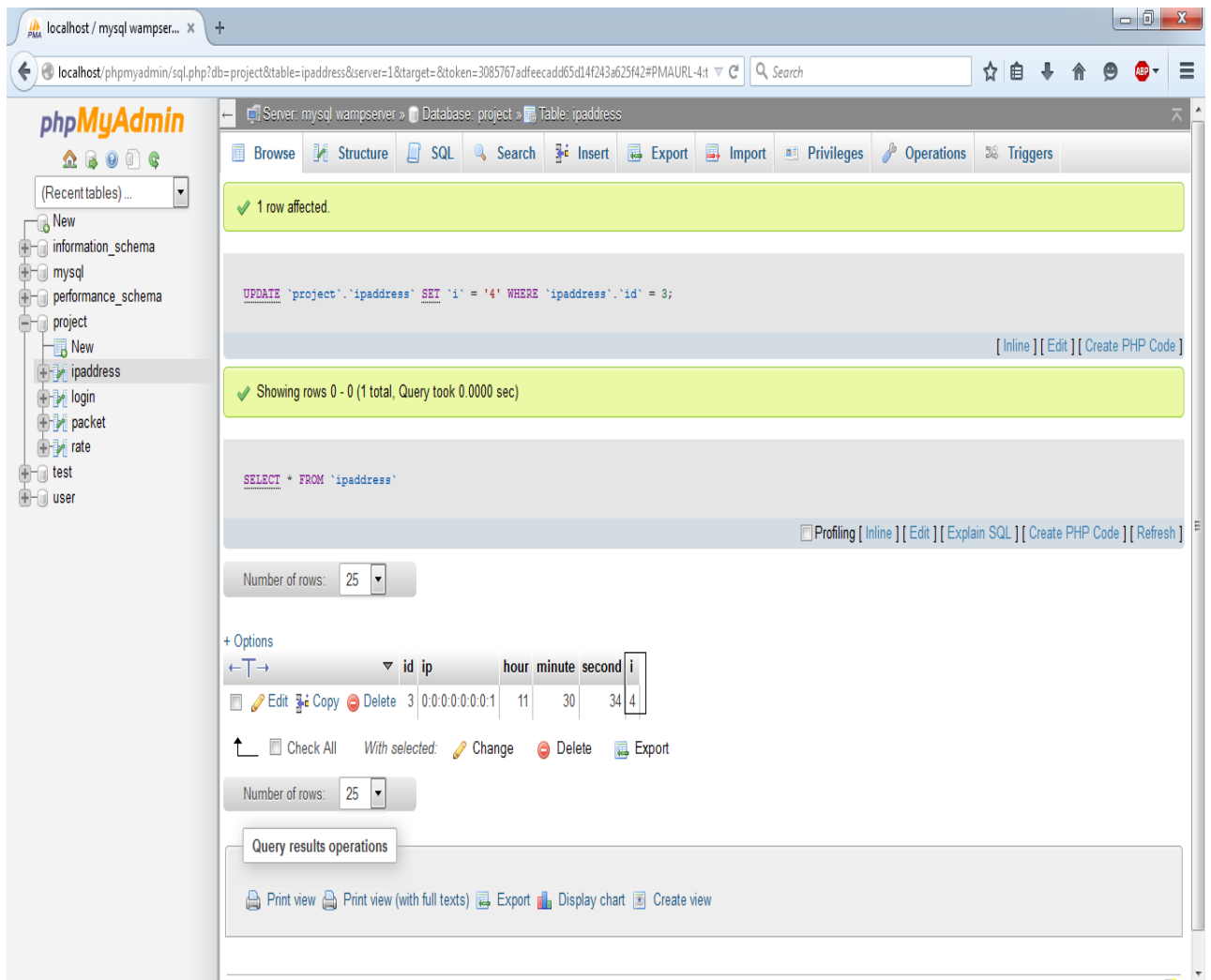
When an HTTP request comes from an already existing IP Address within 2 seconds of previous request, then the value of i gets incremented by 1. Here, the value of i was initially 1, but after this request, it has become 2.



**Fig 23 – After sending third request within another 2 seconds, i becomes 3**

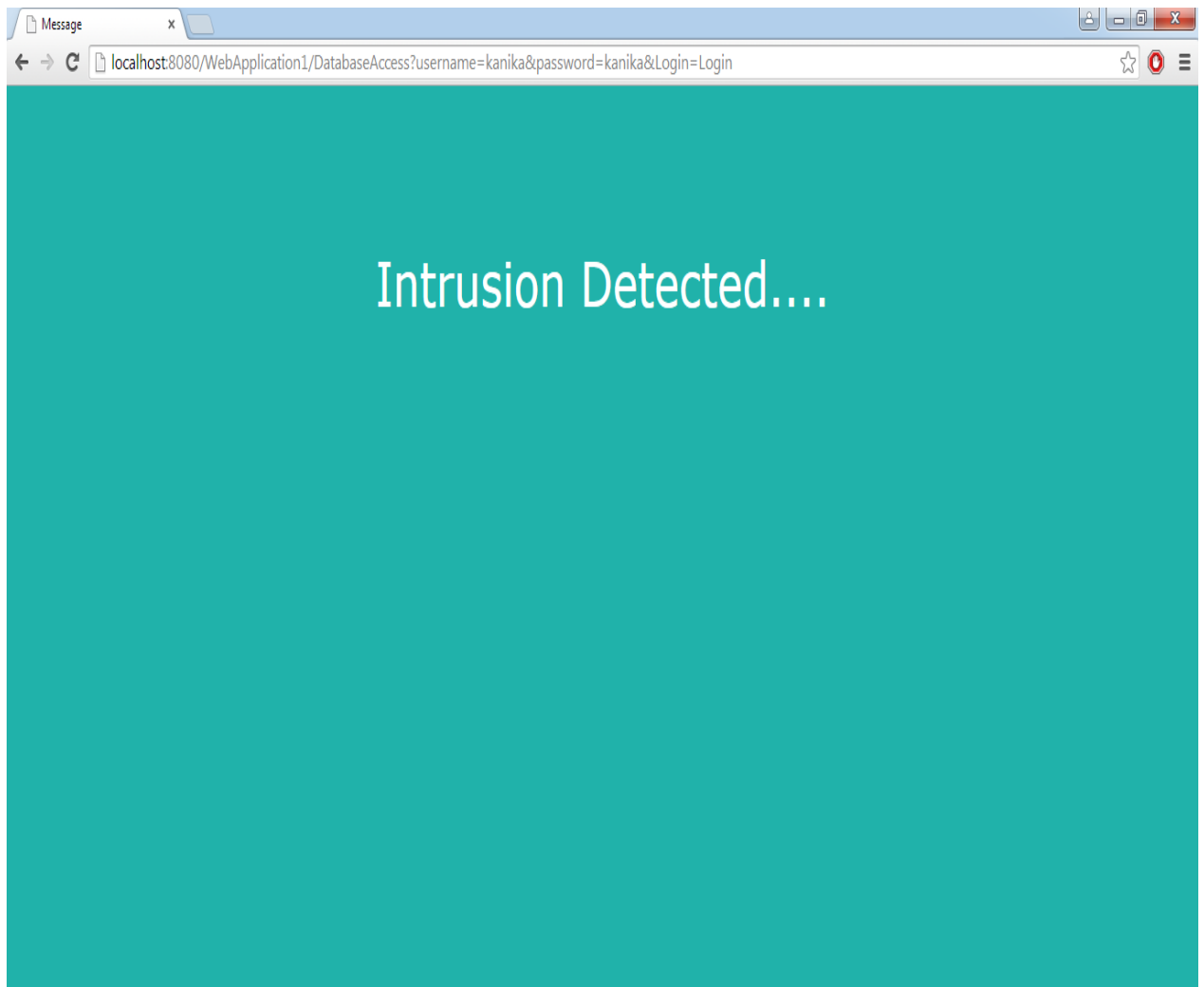
When an HTTP request comes from an already existing IP Address within 2 seconds of previous request, then the value of *i* gets incremented by 1. Here, the value of *i* was initially 2, but after this request, it has become 3.





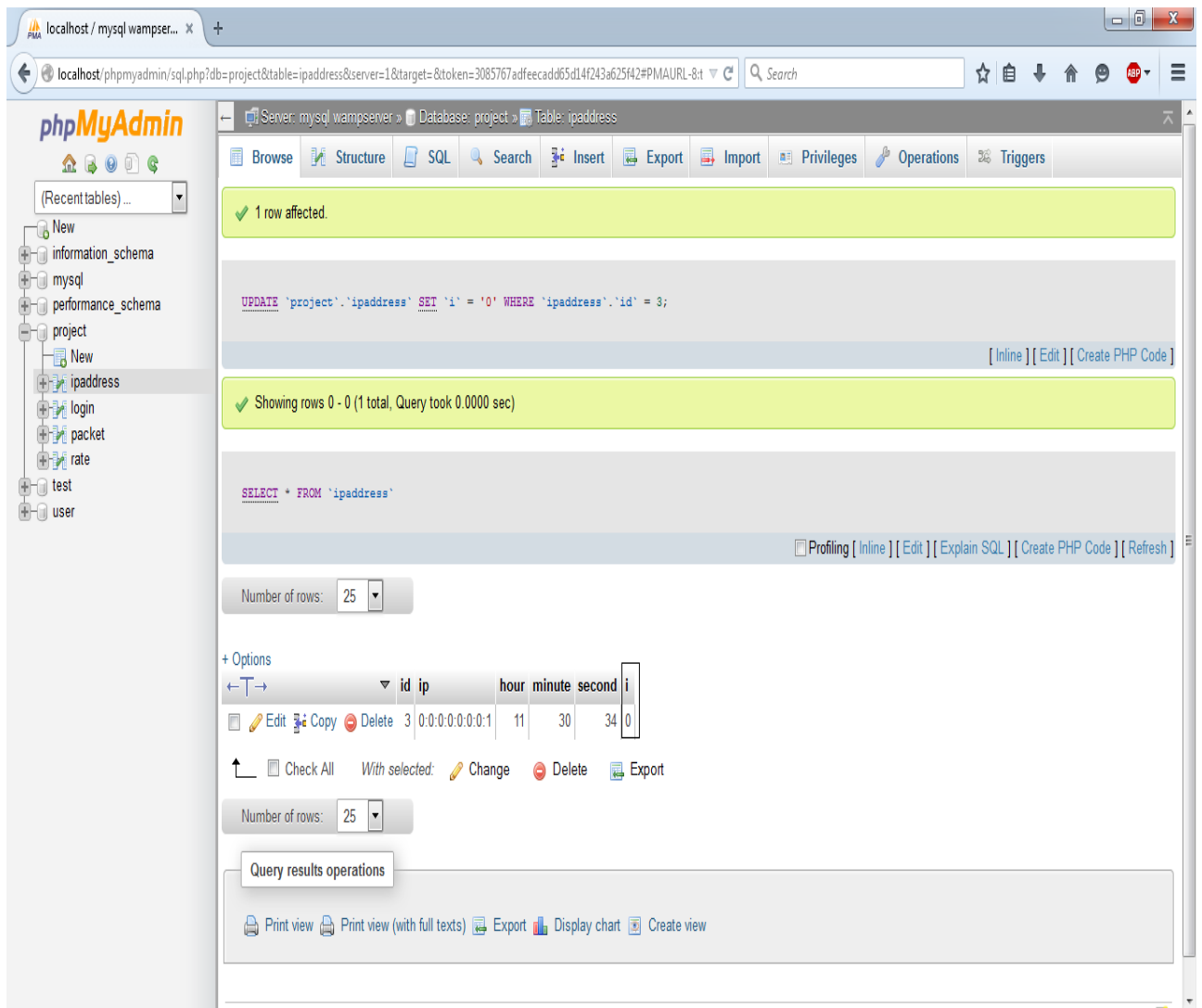
**Fig 24 – After sending forth request within another 2 seconds, i becomes 4**

When an HTTP request comes from an already existing IP Address within 2 seconds of previous request, then the value of i gets incremented by 1. Here, the value of i was initially 3, but after this request, it has become 4.



**Fig 25 – As soon as fifth request is sent within another 2 seconds, i becomes  $\geq 5$  and alert is sent to administrator regarding Intrusion**

When the value of this variable  $i$  exceeds a predefined value, then it's the indication of an intrusion. In this case, the user is directed towards a page containing an alert message for the administrator that an intrusion has occurred and he/she might look into the matter.



**Fig 26 – After fifth request, i becomes 0 again**

As soon as an intrusion is detected, the value of I for that particular IP Address is set to 0 again. It gets incremented if the same case happens as discussed before.

## **CHAPTER 6**

# **CONCLUSION AND FUTURE WORK**

In our project, I have designed a Hybrid Intrusion Detection System which targets users of banks and such other organizations that require client server interaction and continuous updating of certain data. The areas that we have covered in our project are as follows:

1. Detection of DOS attacks
2. Detection of Phishing

There is still much room for future development that would enhance the system and increase its value. The following items are some suggestions:

1. It can stop IP requests if they come from a particular location
2. It can stop IP requests if they come from a location that is from a far off place (may be another country) and can be an intrusion to network
3. It can stop requests if they come from a different browser that the user typically doesn't use
4. It can make an alert if any kind of anomaly is detected

# REFERENCES

## PAPERS -

- [1] Meisam S.A. Najjar, Mohammad Abdollahi Azgomi, “**A Distributed Multi-Approach Intrusion Detection System for Web Services**”, School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran, *SIN’10*, Sept. 7–11, 2010, Taganrog, Rostov-on-Don, Russian Federation.
- [2] A.S. Aneetha, T.S. Indhu, Dr. S. Bose, “**Hybrid Network Intrusion Detection System using Expert Rule based Approach**”, Anna University, Chennai, India, *CCSEIT-12*, October 26-28, 2012, Coimbatore [TamilNadu, India]
- [3] Muhammad Jamshed, Jihyung Lee, Sangwoo Moon, Insu Yun, Deokjin Kim, Sungryoul Lee, Yung Yi, KyoungSoo Park, “**Kargus: A Highly-scalable Software-based Intrusion Detection System**”, Department of Electrical Engineering, KAIST, *CCS’12*, October 16–18, 2012, Raleigh, North Carolina, USA.
- [4] Xinli Wang, Alex Kordas, Lihui Hu, Matt Gaedke, Derrick Smith, “**Administrative Evaluation of Intrusion Detection System**”, School of Technology, Michigan Tech University Houghton, USA, *RIIT’13*, October 10-12, 2013, Orlando, Florida, USA.
- [5] SunWoo Kim, TaeGuen Kim, Eul Gyu Im, “**Real-time Malware Detection Framework in Intrusion Detection Systems**”, Department of Computer and Software, Hanyang University Seoul, Korea, *RACS’13*, October 1–4, 2013, Montreal, QC, Canada.
- [6] P Kaur (RIMT-MAEC, Mandi Gobindgarh, Punjab, India), D Rattan(BBSBEC, Fatehgarh Sahib, Punjab, India), A K Bhardwaj (Thapar University, Patiala, Punjab, India), “**Enhancement of Fault Tolerance of Intrusion Detection System using AES and DES based Heart Beat Events**”, *ICWET’11*, February 25–26, 2011, Mumbai, Maharashtra, India.
- [7] Chun Guo, Yajian Zhou, Yuan Ping, Zhongkun Zhang, Guole Liu, Yixian Yang, “**A distance sum-based hybrid method for intrusion detection**”, 13 June 2013

[8] K. V. Arya, Hemant Kumar, ABV-Indian Institute of Information Technology & Management, Gwalior, India, “**A Clustering based Algorithm for Network Intrusion Detection**”, SIN’ 12, October 25-27, 2012, Jaipur, India

[9] Nawal A. Elfeshawy , Osama S. Faragallah, Department of Computer Science and Engineering, Faculty of Electronic Engineering, Minufiya University, Menouf 32952, Egypt, “**Divided two-part adaptive intrusion detection system**”, Wireless Netw (2013)

### **WEB LINKS -**

[1] [http://www.cis.syr.edu/~wedu/Teaching/cis758/LectureNotes/Intrusion\\_Detection.pdf](http://www.cis.syr.edu/~wedu/Teaching/cis758/LectureNotes/Intrusion_Detection.pdf)

[2] <http://www.sans.org/reading-room/whitepapers/detection/intrusion-detection-systems-definition-challenges-343>

[3] [http://www.cis.syr.edu/~wedu/Teaching/cis758/LectureNotes/Intrusion\\_Detection.pdf](http://www.cis.syr.edu/~wedu/Teaching/cis758/LectureNotes/Intrusion_Detection.pdf)

[4] <http://www.sans.org/security-resources/idfaq/ipe.php>

[5] [http://insecure.org/stf/secnet\\_ids/secnet\\_ids.html](http://insecure.org/stf/secnet_ids/secnet_ids.html)

[6] [http://en.wikipedia.org/wiki/Host-based\\_intrusion\\_detection\\_system](http://en.wikipedia.org/wiki/Host-based_intrusion_detection_system)