

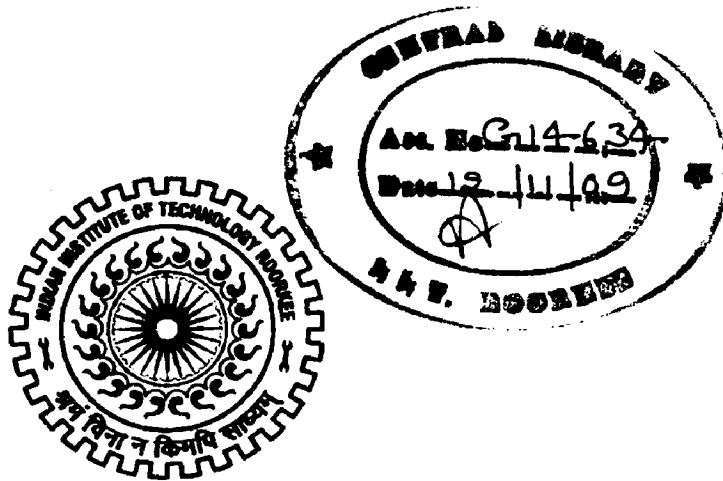
# IDENTIFICATION OF SOURCE MACHINE USING ATTRIBUTION BY NETWORK FORENSIC CAPTURING

## A DISSERTATION

*Submitted in partial fulfillment of the  
requirements for the award of the degree  
of*  
**MASTER OF TECHNOLOGY**  
in  
**COMPUTER SCIENCE AND ENGINEERING**

By

**SANJEEV SHUKLA**



**DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING  
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE  
ROORKEE -247 667 (INDIA)  
JUNE, 2009**

## CANDIDATE'S DECLARATION

---

I hereby declare that the work, which is being presented in this dissertation, entitled **“IDENTIFICATION OF SOURCE MACHINE USING ATTRIBUTION BY NETWORK FORENSIC CAPTURING ”**, towards partial fulfillment of the requirements for the award of the degree of MASTER OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING, submitted in the department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, Roorkee (India) is an authentic record of my own work carried out from June 2008 to June 2009, under the guidance and supervision of Dr. R. C. JOSHI, Professor, Department of Electronics and Computer Engineering, Indian Institute of Technology, Roorkee.

I have not submitted the matter embodied in this dissertation for the award of any other degree or diploma.

Date : 19/06/09

Place : Roorkee



( Sanjeev Shukla )

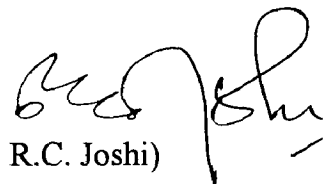
---

## CERTIFICATE

This is to certify that the above statement made by the candidate is correct to the best of knowledge and belief.

Date : 19/06/09

Place : Roorkee



( Dr. R.C. Joshi )

Professor

Department of Electronics and Computer Engineering

IIT-Roorkee - 247667

## ACKNOWLEDGEMENT

---

I would like to take this opportunity to extend my heartfelt gratitude to my guide and mentor Dr. R. C. JOSHI, Professor, Department of Electronics and Computer Engineering, Indian Institute of Technology, Roorkee, for his trust in my work, his able guidance, regular source of encouragement and for providing me constant support and thoughtful insights during the progress of my work. I would state that the dissertation work would not have been in the present shape without his inspirational support and I consider myself fortunate to have done my dissertation under him.

It also gives me immense pleasure to mention the encouragement I received from my Head of Department, Prof. Padam Kumar. He was kind enough to motivate me in times of need with both his advice and suggestions. I would also thank the support i received from the staff members and colleges of my department (ICC).

I would also like to express my gratitude to my parents for everything that they have done for me without complaining, to my wife for putting upto my tantrums but still supporting & encouraging me and finally to my daughter who sacrificed her quota of playtime with me. All of this would have been impossible without their constant support.



Sanjeev Shukla

## ABSTRACT

---

Networks have been an essential part of our information infrastructure which enables us to perform various critical operations. The vast amount of data traveling is a potential source which can be examined & investigated to find crucial evidence for e-crimes. Network forensic is an investigation technique which looks at network traffic to find substantial evidence in support of the dubious incidents.

The work presented here takes a specific problem of identifying source machine after network address translation is done. Network address translation (NAT) is the process of modifying network address information in datagram packet headers while in transit across a traffic routing device for the purpose of remapping a given address space into another. This poses a great challenge for forensic analysis because it is difficult to attribute observed traffic into discrete hosts. The algorithm thus developed relies on the combination of number of unique characteristics (attributes) specific to a source offered by each layer of the OSI model, allowing identification of source machines. The attribution method used is much better than other approaches like IP Traceback where IP is at the epicenter for all the processes. Here identification does not take IP into consideration & hence its possible for addresses to dynamically change ( by using DHCP) without effecting the algorithm.

The program developed follows the process framework of forensic analysis. In a step-by- step process it captures the network traffic which is then segregated to extract relevant information pertaining to the event concerned. It is then analyzed using attribution algorithm to find the source of each packet streams and the results are displayed in graphical form which is easy to represent and understand.

# CONTENTS

---

Candidates Declaration and Certificate.....	i
Acknowledgments.....	ii
Abstract.....	iii
Table of Contents.....	iv
List of Figures.....	vii
<b>CHAPTER 1</b>	<b>Introduction &amp; Problem Statement.....1</b>
1.1	Introduction.....1
1.2	Motivation.....2
1.3	Problem Statement.....3
1.4	Organization of the Report.....3
<b>CHAPTER 2</b>	<b>Background and Literature Review.....5</b>
2.1	Critical Review.....5
2.2	Research Gaps.....6
<b>CHAPTER 3</b>	<b>Forensic Process Framework for Source Attribution.....8</b>
3.1	Digital Forensic Process.....8
3.1.1	Traffic Capturing.....9
3.1.2	Data Segregation.....10
3.1.3	Traffic Analysis.....10
3.1.4	Visualization and Report Generation.....11

<b>CHAPTER 4</b>	<b>Source Attribution Components.....</b>	<b>12</b>
4.1	Attribution Method.....	12
4.1.1	Stream Testing.....	15
4.1.2	Source Model.....	16
4.2	Types of Attributes for Distinguishability.....	17
4.3	Energy Function Design.....	20
4.3.1	Internet Protocol Identification.....	20
4.3.2	Timestamp and Clocks.....	23
4.3.3	HTTP Referrer .....	24
<b>CHAPTER 5</b>	<b>Program Implementation.....</b>	<b>26</b>
5.1	System Setup.....	26
5.1.1	System requirement.....	26
5.1.2	Network Setup.....	26
5.2	Implementation of Traffic Capturing Module.....	26
5.3	Module Displaying the Contents.....	28
5.4	Logging Module and Statistical Information generation.....	29
5.5	Module for Source Attribution.....	30
<b>CHAPTER 6</b>	<b>Results &amp; Discussions.....</b>	<b>32</b>
6.1	Display and Discussion of results obtained.....	32
6.1.1	Internet Protocol Identification.....	32
6.1.2	Timestamp.....	34
6.1.3	HTTP Referrer.....	36
6.2	Limitations.....	37
6.3	Validation of Results.....	38

<b>CHAPTER 7</b>	<b>Conclusion &amp; Future Work.....</b>	<b>43</b>
7.1	Conclusion.....	43
7.2	Scope for Future Work.....	43
<b>REFERENCES.....</b>		<b>45</b>
<b>PUBLICATIONS.....</b>		<b>48</b>

## List of Figures

---

Figure 3.1	Components of DFRWS Digital Forensic Process.....	9
Figure 4.1	NAT Architecture.....	13
Figure 4.2	IP IDs Plotted Vs packet number (for Windows Machine).....	21
Figure 4.3	IP IDs plotted vs. packet number (for Linux Machine).....	22
Figure 4.4	HTTP Request Tree.....	24
Figure 5.1	Traffic Capturing Interface.....	28
Figure 5.2	Content Display both header and payload.....	29
Figure 5.3	Analysis based on algorithm.....	30
Figure 5.4	Flow Chat of the source identification algorithm.....	31
Figure 6.1	IP PD plotted against packet number for 1 windows & 1 Linux host.....	32
Figure 6.2	Packets Belonging to Windows machine.....	33
Figure 6.3	No timestamp value for Windows system.....	34
Figure 6.4	Timestamp value indicated in Linux systems.....	35
Figure 6.5	Timestamp plotted against packet capture time.....	36
Figure 6.6	HTTP Referrer header.....	37
Figure 6.7	Network Setup for Validating the results.....	39
Figure 6.8	Data Captured by capturing system inside private LAN.....	40
Figure 6.9	Data Captured and Analyzed with Results .....	41
Figure 6.10	Graph of Validation data.....	42



## **1.1 Introduction**

In this era of computers, almost each aspect of life is touched by computers or digital devices in some way or other. Therefore any crime is bound to have a digital footprint. For these reasons, cyber forensics has become a key investigative component of law enforcement and businesses, and crucial for companies and their economic growth. Computer forensics is a branch of forensic science pertaining to legal evidence found in computers and digital storage mediums [1]. It is basically collection of tools and techniques used to find evidence in a computer. This is a very significant place in the crime scene investigation which reveals a lot of evidence about the action taken & steps performed by the attacker. Another very potential source of information is networks. Cybercrimes are committed over internet and the network can be viewed as a virtual “crime scene” that holds critical evidence based on events before, during, and after a crime. Attacker uses network infrastructure to reach the crime site or victim & by looking into the network traffic one can extract vital information.

Network forensic is an investigation technique which looks at network traffic to find substantial evidence in support of the dubious incidents. It is an important sub-discipline of cyber forensic. Network forensics is defined as to capture, record, and analysis of network events in order to discover the source of security attacks or other problem incidents [2]. Analysts can use data from network traffic to reconstruct and analyze network based attacks and inappropriate network usage, as well as to troubleshoot various types of operational problems. The content of communication carries over the networks, such as email messages or audio, can also be collected in support of an investigation. The term network traffic [3] refers to computer network communications that are carried over wired or wireless network between hosts.

The greater usage of internet by academicians, industry & business class along with proliferation of interconnected systems and network has spurred the adoption of network

address translation gateways. Network address translation (NAT) is basically the process of modifying network address information in datagram packet headers while in transit across a traffic routing device for the purpose of remapping a given address space into another. This is done to deploy lot of systems in a small private LAN behind NAT gateway to use internet through the NAT gateways public IP interface. The machines behind gateway use private IP addresses for intranet activities & use gateways public address for internet. NAT came into existence because of the fear of IP addresses (IPv4) being exhausted by the rapid expansion of internet. NAT thus provided a solution that by using a single public IP as the gateway, a whole set of PCs in private LAN could be connected to internet.

Identifying the real source of attack is a big challenge in network security paradigm. IP traceback is one of the techniques used to determining the origin of a packet on the Internet. Commonly IP traceback is associated with Denial of Service (or Distributed DOS) where it tries to find the source of attack which has spoofed IP address [4]. To find the host which is the source of packet generation is a problem specific to DOS based attacks & a similar kind of problem is to find the source behind a NAT gateway. Network forensic analysis tries to find significant information from the captured network traffic which is relevant to the concerned event [5]. At times this information is obscured because the source of relevance is deployed behind NAT gateway. This makes attribution a significant problem because all the network traffic emanating from the NAT gateway appears to have same source IP as it is of the gateway's public interface. It is difficult to segregate the relevant source from the irrelevant source from the network traffic [6].

## **1.2 Motivation**

E-crime watch survey - 2006 which was conducted in cooperation with the U.S. Secret Service, Carnegie Mellon University Software Engineering Institute's CERT@Coordination Center and Microsoft Corp has shown some interesting information. The survey does reflect that 95 % of criminal activity leaves some sort of digital trace. This means that if a person performs any sort of activity with digital/electronic devices, there is very high probability that he might leave some sort of evidence or trace. It also states

that a whopping 70% of cases are not reported for various reasons. Of these approx 50 % cases were not considered due to lack of information or difficulty in finding the attacker identity.

All the techniques that are used to find the originating source of the packet (ex IP Traceback ) take IP address at their focal point. The immense significance given to the IP address in tracing back to the attacking node does pose some inherent challenges. In DHCP environment where the host IP address is changing dynamically or in IP spoofing where IP packets are created with a forged (spoofed) source IP address with the purpose of concealing the identity of the sender or impersonating another computing system, schemes based on IP will find an additional difficulty in finding the original source.

The motivation is to have a technique which is free of IP address based trace & yet tries to give results which is efficient, effective & less time consuming.

### **1.3 Problem Statement**

The aim is to identify the source machine from the data captured after the network addresses translated gateway. This problem can be divided into following sub problems :

- Implement a program which performs network traffic capturing so that the data acquired could be used for analysis.
- Apply attribution technique to write an algorithm for identification of source machine from where the packets are originating.
- Devise graphical display representation of results.

### **1.4 Organization of Report**

The complete work on the use of source attribution to find the originating packet source is presented in this report in the following format :

Chapter 2 contains the background and literature review. In this the existing techniques and the proposed methods used for forensic analysis & attribution are discussed. The existing research gaps are also highlighted.

In Chapter 3 Digital Forensic process framework is stated and a step by step explanation of the each process in terms of network forensic & source attribution is done.

Chapter 4 talks about the source attribution methods & its components involved in the identification process. It contains the proposed attribution method for our traffic stream, energy function design for the optimization of the right configuration for the system and types of unique attributes of source which helps us to identify them.

Chapter 5 explains the implementation details and issues of the proposed strategy. The various modules which perform certain function are explained.

Chapter 6 discusses the results. The various attributes used for experimentation and corresponding results obtained are shown in figures and explained. Validation of results is also done with issues specific to the topic & its limitation are also spelt out.

Chapter 7 concludes by summarizing the work and discusses its applicability. Areas where future work needs to be done are also listed out in this chapter.

## **2.1 Critical Review**

Digital forensic analysis came into existence way back in 2001 when it was formally put in first Digital forensic research workshop[7 & 8] conducted by Defensive information warfare branch of US. It mainly had proposed a framework for forensic processes. It also dwelled upon trustworthiness of digital evidence and talked about computer forensic. Initial years work were mainly concentrated in computer forensic analysis, finding evidence in PC/laptop/servers/PDA's by looking into memory, hidden places & storage media. Network forensic analysis came after few years when the usage of internet has grown & need was felt to study network traffic to find potential information [5].

Jung, Dong & Bong [9] applied fuzzy logic based approach to have an effective and automated analyzing system for network forensics. It proposed a fuzzy logic based expert system for network forensics that can analyze computer crimes in networked environments and make digital evidences automatically. It basically matched and analyzed patterns & based on that made rule to decide whether the packets are attacks or not. The problem with this approach was its pattern matching & frequency of changing rules, which was high & hence detection of packet was less. Also it worked more like an intrusion detection system.

Desmond & Cho [10] used source attribution technique based on fingerprinting to differentiate between unique devices over a wireless local area network (WLAN). Fingerprinting is essentially a process by which a machine or the software the machine is running can be uniquely identified due to its externally observable characteristic. This approach though has used physical layer of the OSI model to find this specific characteristic. Other used clock skew & jitter as the unique resemble identifier of a source to fingerprint a device at the physical layer [11]. Liberatore & Levine [12] on the other hand used transport layer of the OSI model to evaluate traffic analysis that infer the source of a web page retrieved under the cover of an encrypted tunnel. These techniques

identify sources by comparing observed traffic to profiles of known sites created from packet lengths.

Mchugh, McLead and Nagaonkar [13] used passive network forensic by looking at behavioral classification of hosts based on connection patterns. They used behavioral changes to identify role shifts and traced malicious and unintentional propagation of that change to other machines. The methodology of profiling host behavior on connection patterns utilizing the network traffic is subtle but it work more for monitoring purpose and will have issues in detecting any real time crimes.

Significant amount of researchers has taken software tools and suggested there effective usability or had performed some experimentation. Eg M.I. Cohen[14] published his paper in Elsevier “PyFlag- An advanced network forensic framework” in 2008 signifying the use of network traffic to be used as evidence by using PyFlag, an open source, forensic package which merges disk , memory and network forensics. Other like Nikkel [15] developed a small portable network forensic evidence collector device using inexpensive embedded hardware and open source software. The problem though is that the focus of device is limited to data acquisition & not analysis hence limiting its scope.

## **2.2 Research Gaps**

There are number of research gaps in this area that need attention. The main gap has been in the area of analysis where there is hardly any common ground on the methods applied. The techniques are merely proposed and no followed up of it has been done by applying and testing it thoroughly. There is also no symmetry in various forms of methods applied and the approaches used are vastly different.

*There is also no standardization of the forensic analysis processes. It is essential to have a framework well defined and standardized as the work involves legal implication. A step by step strategy is necessary in it as every method need to be verified or else can be challenged by defense team in court of law. The other challenge is in software used for automation. These software’s are essential for fast retrieval of information from massive*

data tombs. There are lots of software available in market neither of them are compatible with each other nor do they have any common format. Data collected by any tool should be in common format so that it can be exported by any analysis system & worked upon.

## **Forensic Process Framework for Source Attribution Chapter 3**

### **3.1 Digital Forensic Process**

The Digital Forensic Process as defined by Digital Forensic Research Workshop (DFRWS) in [7] and [8] has the following investigating process:

- (i) **Collection** : Data related to a specific event is identified, labeled, recorded, and collected, and its integrity is preserved. Here media is transformed into data.
- (ii) **Examination** : Identification and extraction of relevant information from the collected data while protecting its integrity using a combination of automated tools and manual processes. Here data is transformed into information.
- (iii) **Analysis** : This involves analyzing the results of the examination to derive useful information that helps the investigation. Here information is transformed into evidence.
- (iv) **Reporting** : Reporting the results of the analysis, describing the actions performed, determining what other actions need to be performed, and recommending improvements to policies, guidelines, procedures, tools, and other aspects. Generated evidence is used to formulate reports, prepare charts and support decisions.

Figure 3.1 depicts the framework in terms of activities and the inputs & outputs associated with each of the activity.



### **3.1.2 Data Segregation**

This phase of forensic process has its own significance and here traffic relevance is of primary concern. Traffic relevance refers to the fact that the concern data is related to the event of investigation. It extracts the relevant data from the vastly captured traffic streams and it is given to the next phase for analysis. The conversion from data to useful information specific to an event is achieved in data segregation.

### **3.1.3 Traffic Analysis**

The penultimate phase of the forensic process deals with the analysis of the results of the previous step. The analysis would try to link various bits and pieces of the evidence together to give a large and clear picture of the criminal/illegal activity using legally justifiable methods & techniques. The information generated would address the questions that were the impetus for performing the data collection & data segregation. Network forensics can generate enormous quantities of information. Its analysis ranges from the trivial (When did host A ask host B for a network time protocol update) to the grandiose (When are the demand peaks, and which protocols contribute to them) to the obscure (What implementations of RFC 868 violate the published standard). Hence the objective of analysis with reference to a specific event has to be clearly stated.

In our problem the network traffic is treated as streams and a method of attribution & source are applied. The algorithm based on the energy design is to find the best configuration for the system. Best configuration refers to the case where a stream is associated with the correct source. Optimization of energy function to detect the source for a stream is the basis of the algorithm. The algorithm thus developed relies on the combination of number of unique characteristics (attributes) specific to a source offered by each layer of the OSI model, allowing identification of source machines.

### **3.1.4 Visualization and Report Generation**

The final phase of the forensic investigation process within the incident response cycle involves reporting the results of the analysis. This activity would lead to describing the actions used, explaining how tools and procedures were selected, determining what other actions need to be performed (e.g. forensic examination of additional data sources, securing identified vulnerabilities, improving existing security controls) and providing recommendations for improvement to policies, guidelines, procedures, tools & other aspects of the forensic process. The formality of the reporting step varies greatly depending on the situation, organization of the investigating entity, legal framework being referenced, primary objective of the digital system which was being investigated etc.

Visualization basically facilitates traffic and content inspection in graphical form which is easy to represent and understand. It is to present the analyzed data in a manner which is suitable for viewing like in tabular forms or graphs etc.

### **4.1 Attribution Method**

To address the issue of the identification of source based on attribution [16], it is imperative to consider how NAT devices work. Network Address Translation (NAT) is the process of modifying network address information in datagram packet headers while in transit across a traffic routing device for the purpose of remapping a given address space into another. NAT is a technique that hides an entire address space, usually consisting of private network addresses (RFC 1918), behind a single IP address in another, often public address space. This mechanism is implemented in a routing device that uses connection state tables to map the "hidden" addresses into a single address and then rewrites the outgoing Internet Protocol (IP) packets on exit so that they appear to originate from the router. In the reverse communications path, responses are mapped back to the originating IP address using the rules ("state") stored in the translation tables. The translation is done for both TCP & UDP packets.

When a host sitting behind the NAT sends a packet, it is received by the NAT gateway which in turn looks into its connection state table to check whether the packet is part of an already going session or a new session. If new connection it rewrites the packet modifying its source IP address with the IP address of the external interface of NAT and its port number is the new assigned port number by the gateway. Similarly, for incoming packets they are consulted with translation table & now the destination IP is modified to the original source IP and the destination port is again the original port number which was stored in the translation table.

A typical architecture of source behind NAT is shown in Figure 4.1. Here hosts in a private LAN having illegal IP's are connected to NAT gateway for all their outbound traffic connections. A capturing device is placed after gateway to acquire network traffic data and hence has problem of identifying the source host of the originating packet.

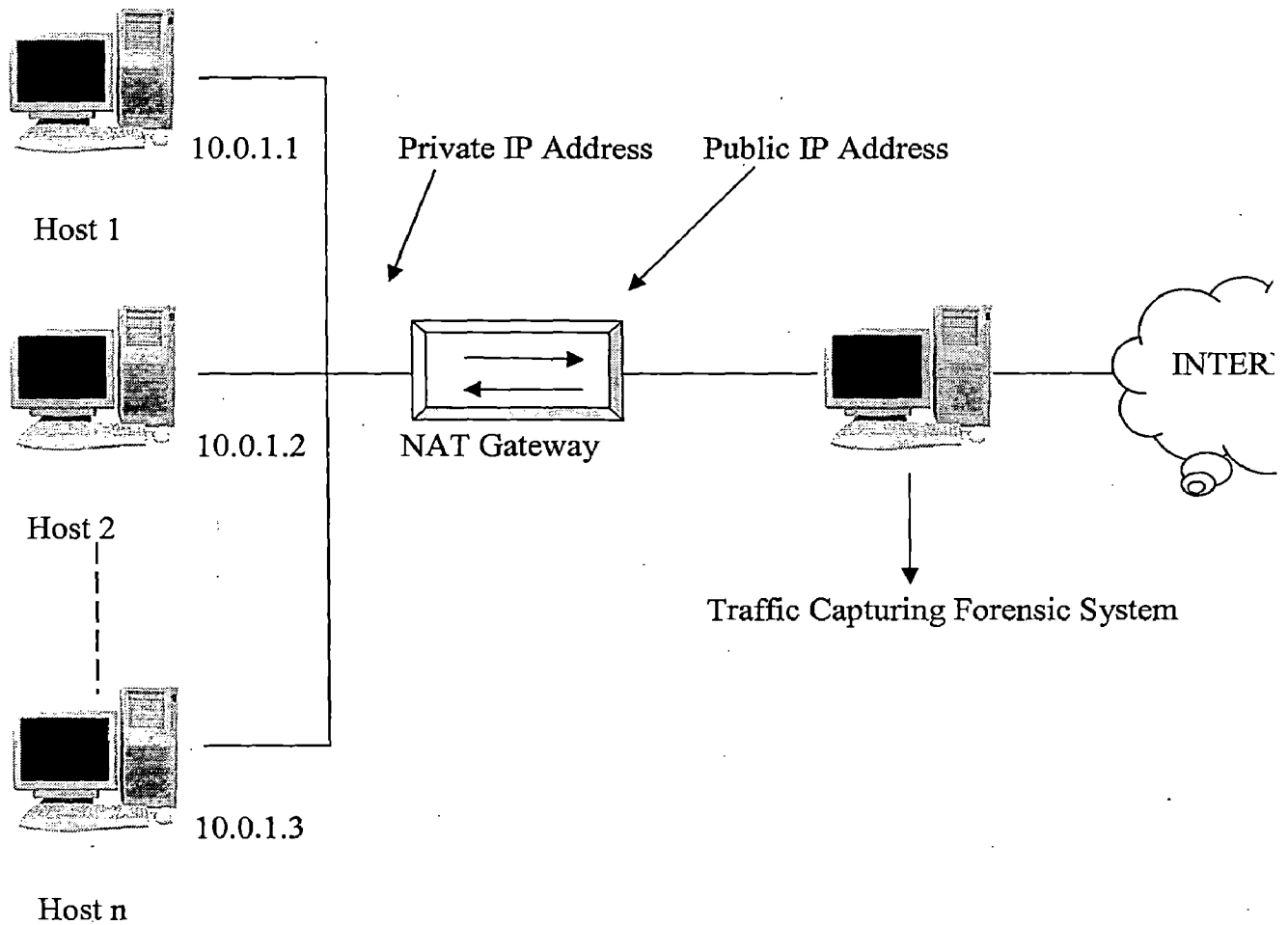


Figure 4.1 : NAT Architecture

Each translation can be loosely referred as streams. A connection state table can be summarized as a sequence of stream entries:

$$\text{Stream} = \{ S_{\text{Add}}, S_{\text{Port}}, GW_{\text{Add}}, GW_{\text{Port}}, D_{\text{Add}}, D_{\text{Port}} \} \quad (4.1.1)$$

$S_{\text{Add}}, S_{\text{Port}}$  : Non – Observable Properties

$GW_{\text{Add}}, GW_{\text{Port}}, D_{\text{Add}}, D_{\text{Port}}$  : Observable Properties

Where S represents the originating source, D destination and GW represent gateway.  $_{\text{Add}}$  and  $_{\text{Port}}$  represents IP address & port number. Address & port number after the gateway are only observable properties while those behind gateway (ex Source address & port number) are unobservable entries. It is also evident that  $GW_{\text{Add}}$  is fixed for all the packets while  $GW_{\text{Port}}$  changes sequentially in a specific defined range.

A stream is defined as a set of outgoing packets with a source having  $(GW_{Add}, GW_{Port})$  and a destination  $(D_{Add}, D_{Port})$ . Similarly in incoming stream is reverse of it by having source  $(D_{Add}, D_{Port})$  of & destination  $(GW_{Add}, GW_{Port})$ . The problem then becomes to deduce the connection state table and in particular assign unique sources for each stream given the observable properties and streams themselves.

Definition : A Source (denoted by  $S$ ) is a set of streams which are attributed to the same host. It forms a set of packets which is the union of each stream attributed to that source:

$$S = \{ s_1 , s_2 , s_3 \dots \} = s_1 \cup s_2 \cup s_3 \dots \dots \dots \quad (4.1.2)$$

Streams are attributed to exactly one source at a time.

The source attribution method thus developed contains a finite set of sources each of which contains a finite number of streams. The system is configured in such a way that a stream is assigned to a particular host say stream 1 ( $s_1$ ) is assigned to source 1( $S_1$ ). The accuracy of the system thus depends on the correct configuration which in turn is to assign right stream to right source. The correct configuration is taken to be the one which is most likely based on specific stream properties. The chosen configuration is tested for the streams internal consistency. If it is plausible for the configuration to be correct we retain the configuration as a possible configuration else we try different configuration. The problem of finding the best configuration is now reduced to a discrete multivariate optimization problem.

Optimization refers to the study of problems in which one seeks to minimize or maximize a real function by systematically choosing the values of real or integer variables from within an allowed set. Optimization problems typically contain a function "f" called an "objective function", "energy function", or "cost function". The energy function reflects how far away from an optimal state the current system is. A feasible solution that minimizes (or maximizes, if that is the goal) the objective function is called an "optimal solution" i.e. the system is re-configured in such a way as to minimize the energy function and thereby bringing the system to more optimizes state. The optimization

algorithms aim is to bring this energy function to either a local minimum or global minimum (an absolute optimal configuration).

The energy of system is defined as the sum of energy factors of each source :

$$E_{\text{Total}} = \sum E(S_i) \quad (4.1.3)$$

The correct configuration is the one which minimizes the total energy of the system.

#### 4.1.1 Stream Testing

The process of stream testing begins with each stream being assigned to a source assuming it is the best system configuration. This assigning of stream is done either sequentially or based on source model developed. The stream under test (s) is assigned to the most likely source (S) to start with and then is tested & energy calculated to know which stream is part of which source. Energy of stream is calculated based on their energy terms :

$$E(P_i) = E(S_i \cup s) + \sum_{j=0}^N E(S_j) \quad (4.1.1.1)$$

$$\Delta E(P_i) = E(S_i \cup s) - E(S_i) \quad (4.1.1.2)$$

Where N is the total number of sources & n is the total number of distinct stream captured and  $0 < i < N$ . P is the hypothesis that stream s belongs to source S. The energy associated with this hypothesis is simply the energy term for source  $S_i$  coupled with stream s in addition to the energy terms of all the other sources by themselves. The change of energy  $\Delta E(P)$  for each hypothesis is the amount the system's energy changed by introducing the stream to that configuration.

The hypothesis with the lowest possible overall energy term is chosen as the most likely configuration.

### Algorithm :

- 1). Take a stream  $s_i$  (where  $i = 1 \dots n$ ,  $n$  is total number of distinct streams)
- 2). Assign it to a Source  $S_j$  (where  $j = 1 \dots N$  and  $N$  is total number of sources). The assignment is either sequential or based on source model.
- 3). Calculate Energy  $E(P)$ .
- 4). Assign the same stream to next source.
- 5). Calculate energy  $E(P)$ .
- 6). Finally the source producing less energy after assigning stream to it i.e. the one with  $\min E(P)$  is the source to which stream is assigned.
- 7). Take next stream.
- 8) Repeat step 2) to 7).

### Example :

Total number of Source  $\implies N = 2 = \{ S_1, S_2 \}$

Total distinct streams  $\implies n = 3 = \{ s_1, s_2, s_3 \}$

Calculating Energy :  $s_1 \longrightarrow S_1 = E(P_1)$  }  $\min E(P_i) = s_1$   
 $s_1 \longrightarrow S_2 = E(P_2)$  }

Similarly for  $s_2$  &  $s_3$

$s_2 \longrightarrow S_1 = E(P_1)$  }  $\min E(P_i) = s_2$   
 $s_2 \longrightarrow S_2 = E(P_2)$  }

$s_3 \longrightarrow S_1 = E(P_1)$  }  $\min E(P_i) = s_3$   
 $s_3 \longrightarrow S_2 = E(P_2)$  }

#### 4.1.2 Source Model

The source model is used to make prediction and deductions about the source based on information collected about all the inferences that have been made for a particular source. It is quite useful to build a source model for each source in the system or for all the

sources together. A probabilistic model for each source can be build based on statistical techniques to detect deviations from the normal activities of the source [17]. The browser used for surfing or sending request can reveal information about the client program which generated the request and also information about the specific software version. This is achieved by user agent's HTTP header. By building a probability model of the occurrence of each User Agent string within the source it is possible to make an estimate of the probability that the request came from the source S :

$$p(\text{UserAgent}, S) = \frac{\text{Total User agent request}}{\text{Total request}} \quad (4.1.2.1)$$

Another useful attribute to include in this model is the frequency of requests to certain URLs. For example, setting a particular web site as a browser's home page will result in a request to that site each time the browser is started. Hence the frequency of assessing a URL by source having same URL as its browsers home page is maximum. This can thus be taken into account to device our model.

## 4.2 Types of Attributes for Distiguishability

An attribute is a property which defines an entity uniquely by it characteristics. In networking also each host has some of unique attributes which can be used to identify it specifically. These attributes can be found in each layer of OSI model when we move from application layer to physical layer. Some of these attributes are listed below :

### 4.2.1 HTTP Cookies

A cookie is a small string of text stored on a user's computer by a web browser. A cookie consists of one or more name-value pairs containing bits of information such as user preferences, shopping cart contents, and the identifier for a server-based session, or other data used by websites. HTTP is inherently a stateless protocol. However, many web applications rely on the user maintaining state throughout their use of the application. This state is maintained by use of HTTP cookies. A cookie is a bit of information which



the server requests the client to present in future interactions with the site. Cookies are used to track users and systems.

Attribution based on session cookies is considered very strong, that is we have a high degree of confidence that the two streams have come from the same source. This is because the session cookie is random and designed to be difficult to guess. Even if the user logged into the same site from two different machines, the session cookie will be different.

#### **4.2.2 Login Id's, Email Address, chats etc**

Once the traffic is captured, the sources of interest can usually be isolated through specific traffic attributes such as email address, chat session or login names etc. If any of these information is captured it can be correlated to provides strong hints about the source from where it had originated. Chat session containing text of communication can also be used as evidence for legal proceeding. Similarly email addresses or login names do point towards the attacker's identity.

#### **4.2.3 Timestamps**

A timestamp is a sequence of characters, denoting the date and/or time at which a certain event occurred. Timestamp appear in a number of layers of OSI model. TCP timestamp option (part of OSI Transport Layer) in TCP header has been found to be a reliable source identification technique [11]. TCP timestamp can also distinguish between different types of operating systems since it is off by default on windows OS and on in linux operating systems. Timers provide high degree of confidence in attribution based on clock sources.

Another useful source of timestamp is through HTTP (part of Application layer of OSI). The HTTP protocol itself does not specify for a client to transmit its time. However, many web applications do send the timestamp from the client clock and this can be used to estimate the client's clock drift.

#### **4.2.4 Proprietary Protocols**

Proprietary protocols also provide useful information which can be used for identification. Some of the protocols such as online game communication or VOIP (Voice over Internet Protocol) communication can have strongly attributable information. Some protocols may even divulge the source's internal IP address for example SIP ( Session Initiation Protocol widely used for setting up and tearing down multimedia communication sessions such as voice and video calls over Internet Protocol ) passes internal IP addresses for SDP ( Session Description Protocol which is a format for describing streaming media initialization parameters like session announcement, session invitation, and parameter negotiation in an ASCII string ) negotiated end points when the gateway does not support SIP NAT fixups.

Protocols which deal with usernames or nicks may be used for attribution, and even the contents of the communication itself can be very valuable. For example, identifying a suspect's voice on a VOIP call, or seeing their picture on a video conference stream make for very strong source attribution regardless of the protocols. Once some streams are strongly attributed, these can be used to merge logically distinct sources.

#### **4.2.5 HTTP Referrer**

HTTP referrer basically identifies, from the point of view of an internet webpage or resource, the address of the webpage of the resource which links to it. By checking the referrer, the new page can see where the request came from. When visiting a webpage, the referrer or referring page is the URL of the previous webpage from which a link was followed.

This property of HTTP referrer is part of application layer of OSI model. If a stream is a HTTP stream containing a Referrer header, it is likely that the request for the originating page also came from the same source. Searching for the source which in the recent past requested the referred URL allows us to attribute the present stream to the source.

#### **4.2.6 Internet Protocol Identification (IP ID)**

The IP header contains an identification field termed as IP ID field which is commonly 16 bit wide. The "identification" field in the IP header is used to identify the fragments of a single IP datagram. The value of this field is set by the originating system. It is unique for that source-destination pair and protocol for the duration in which the datagram will be active. The exact format of the IPID is not specified and its implementation is operating system specific. This is part of Network Layer of OSI model.

### **4.3 Energy function Design**

Some of the internal consistency measures which are examined for the construction of energy function are discussed here :

#### **4.3.1 Internet Protocol Identification**

Considering the OSI model, protocols below the network layer are typically invisible to our capture device since the sources of interest are not on the same physical broadcast domain (as shown in Figure. 4.1). We therefore begin our quest for attributable property at the OSI Network Layer, commonly represented by the IP protocol.

The IP header contains an identification field termed the IP ID field. This field is used to ensure each packet is unique in the event it needs to be fragmented during routing. The exact format of the IP ID is not specified and its implementation is operating system specific. Some operating systems simply assign integers which increment by one for each packet sent. The field is commonly 15 or 16 bit wide, and starts at 0 at boot time. Some operating systems write the IP ID in little endian format, while others write the IPID in big endian format. Since the IP ID behavior is an attribute of the source, we need to account for this behavior in our method. Modern operating systems such as Linux actually generate secure IP IDs by randomizing these for each stream in order to defeat the following analysis. This can be seen by the function `secure_ip_id ()` in the linux source tree. The following analysis is most useful for Windows based sources which use a simple sequential generator. IP ID based analysis cannot be performed on Linux based

systems hence we can differentiate between Operating Systems by this analysis

Most NAT implementations do not update the IP ID field at all, hence we usually find the captured packets have the same IP ID as was set by the host which generated the packet, even though source IP addresses and ports may have been rewritten by the NAT implementation [18]. For each packet the source sends, the IP ID increments by one, however, we may not see all the packets the source sends, since not all packets were routed through our capture point. A useful forensic analysis therefore is to plot the IP IDs of all outbound packets against the packet number [18]. Such an example plot is shown in Figure. 4.2.

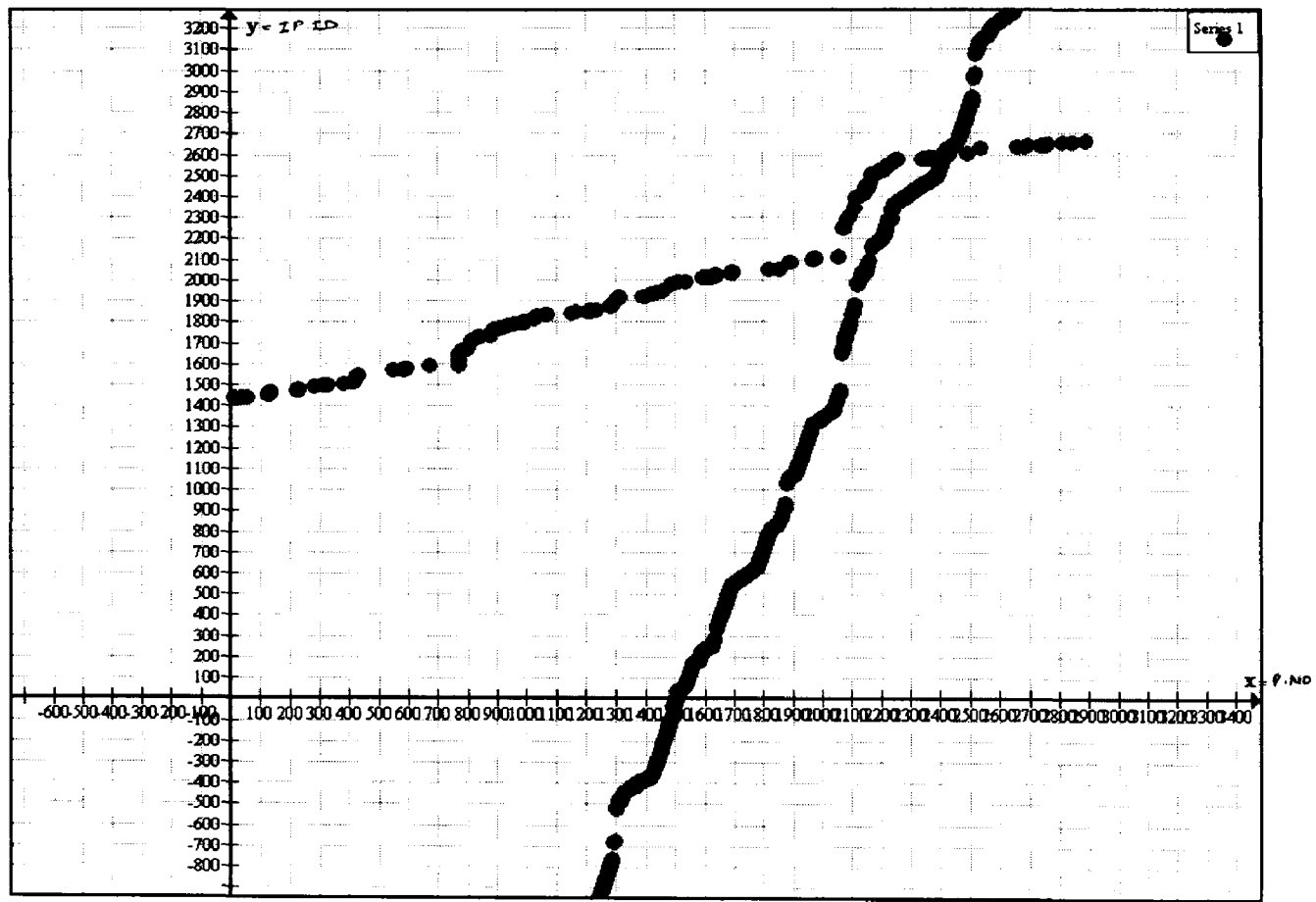


Figure 4.2 : IP IDs Plotted Vs packet number (for Windows Machine)

The Figure 4.2 shows IP ID's plotted against packet number for packets outbound from a

NAT gateway. The 2 lines clearly point to the fact that there are 2 sources which have windows as there operating system.

Similarly Figure 4.3 again shows the IP IDs plotted against packet number. But here we don't get a clear line symbolizing a source generating packets. This random outburst of IP ID's is part of Linux operating system which does not produce packet sequentially. It rather uses a random generator to produce secure IP IDs.

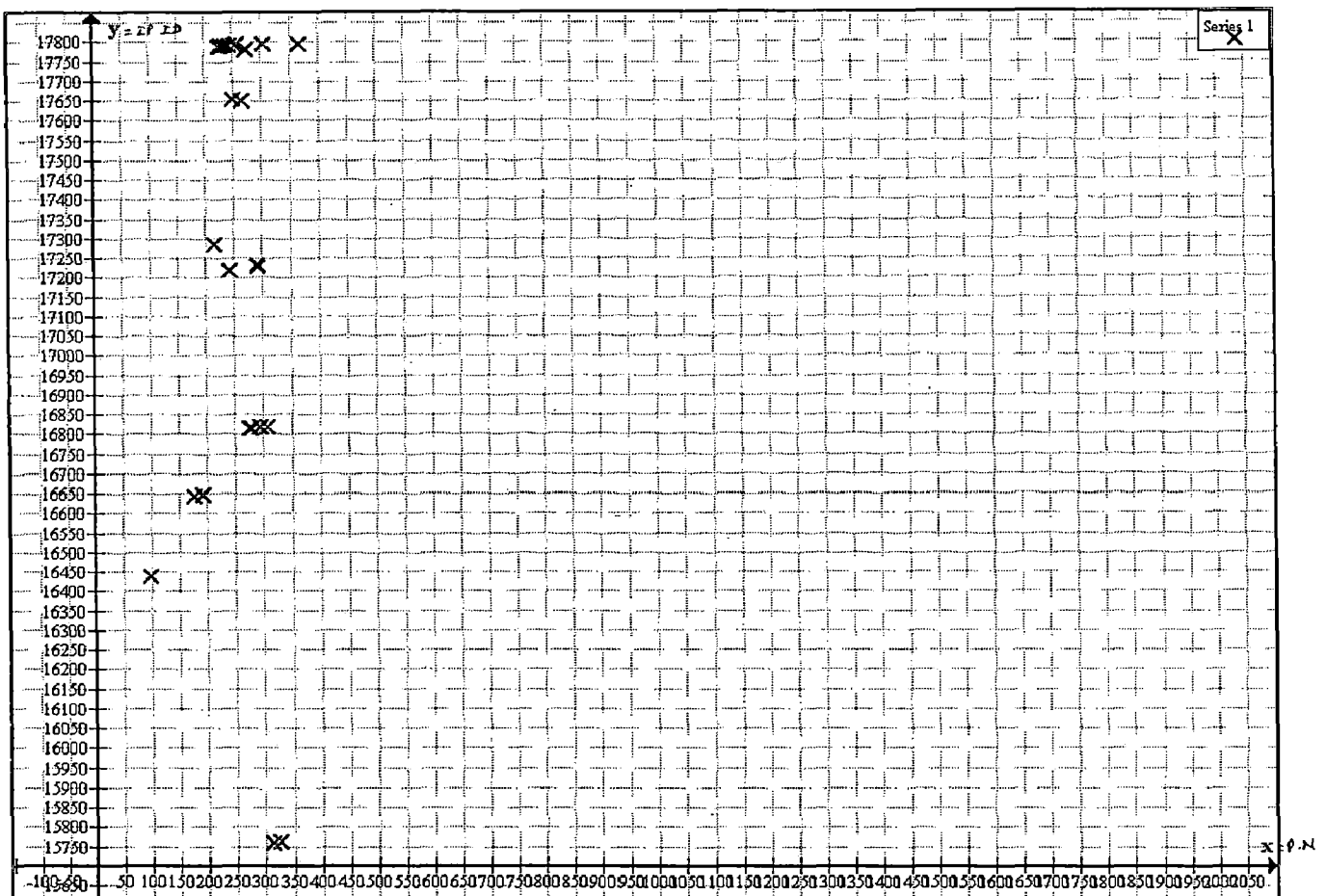


Figure 4.3 : IP IDs plotted vs. packet number (for Linux Machine)

A useful measure of internal consistency of the source is to observe all the packets of the source, with the sequence of their IP ID . For example, assuming a source which generates IP IDs in big endian format and increments IPIDs by one for each packet (e.g. Windows XP):

$$E(S) = \sum \text{mod} (p_{j+1} - p_j - 1, m) \quad (4.3.1.1)$$

where  $m$  is the width of the IP ID field. While  $p_j$  is the IP ID for the  $j$ th packet in the source sequence obtained by ordering packets in time order. Clearly if we are able to observe all packets from this source, the energy for the source will be zero since  $p_{j+1} = p_j + 1$ . If not zero the source producing minimum energy is assigned to be the source of the stream.

### 4.3.2 Timestamp and clocks

Time is a unique attribute of a source, both for its absolute value and for any clock drift we may encounter. Clock drift is a phenomenon where a clock does not run at the exact right speed compared to another clock. That is, after some time the clock "drifts apart" from the other clock. Timestamps may appear in a number of layers of the OSI model. For example, the TCP timestamp option (OSI Transport layer) has been shown to be a reliable source identification technique [11]. Although the TCP timestamp option is off by default on Windows OS's it is on by default on Linux OS's. Another useful source of timestamps is through HTTP. The HTTP protocol itself does not specify for a client to transmit its time (the server however, must send its clock in the Date header). However, many web applications do send the time stamp from the client's clock and this can be used to estimate the client's clock drift. This is an example of unique attribute introduced by the OSI Application layer.

The clock properties are set to be part of the method and an energy function contribution can be taken as the difference between any timestamp and the capturing time. Although clock tests can only be done on some of the connections (e.g. specific HTTP connections), we have a high degree of confidence in attribution based on clock sources.

### 4.3.3 HTTP Referrer

Another OSI Application layer protocol is the HTTP protocol one of the most common protocols on the Internet forming the basis for the World Wide Web. HTTP is often used to transmit HTML (Hyper Text Markup Language) documents. These documents rely heavily on cross linking to other documents, as well as embedding images, script and other multimedia content. HTTP typically uses the Referrer header to indicate that the current request was referred to from another page.

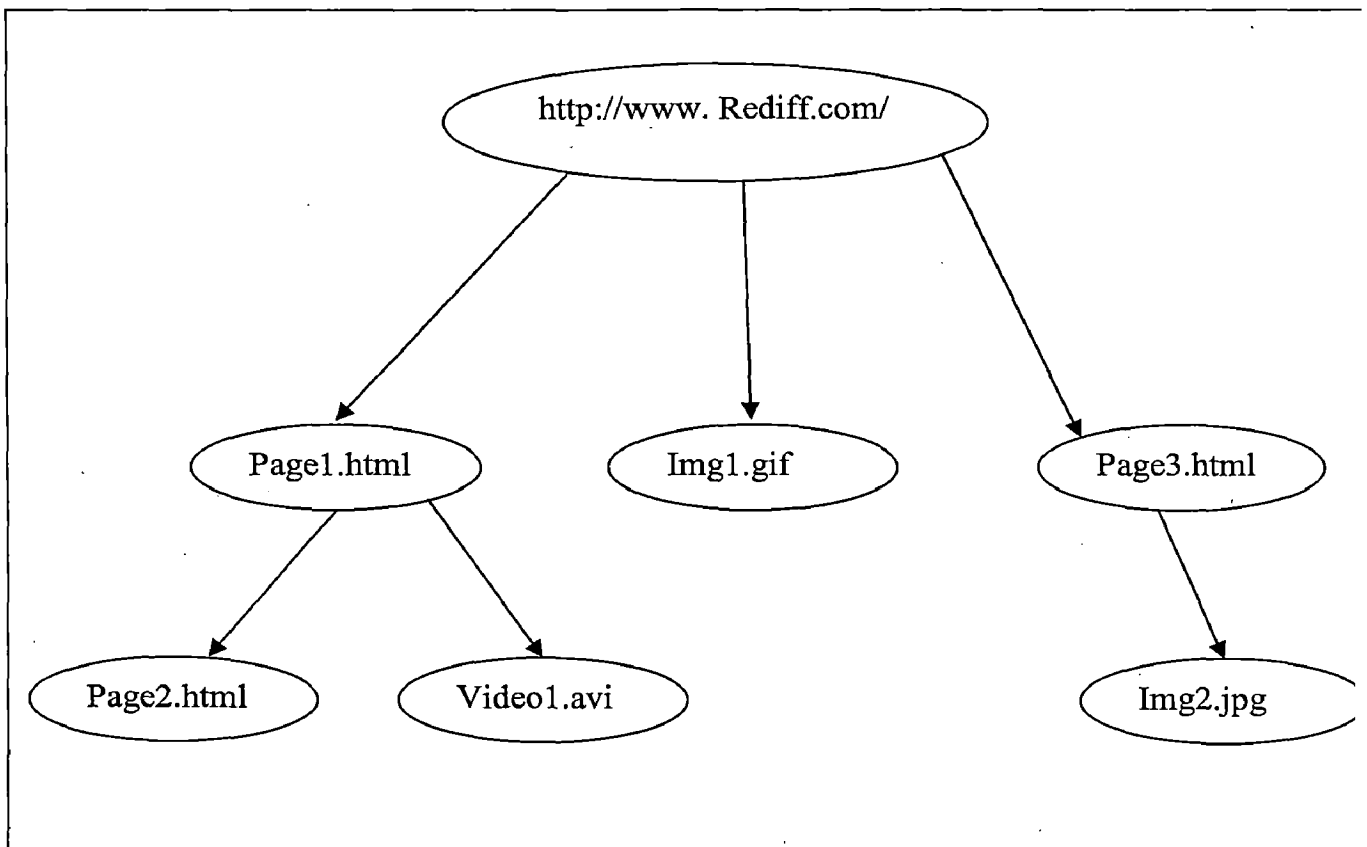
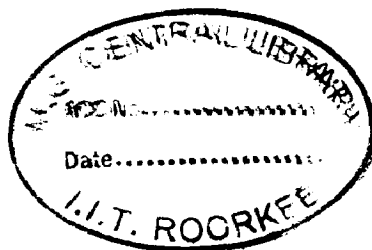


Figure 4.4 : HTTP Request Tree

Figure 4.4 illustrates a typical HTTP request tree. This tree is formed by tracking each object's Referrer header. The initial request fetches an image and 2 links to another page in the form of HTML file. The 1<sup>st</sup> page (page1.html) fetches a video file and again a new link to another page. Similarly 2<sup>nd</sup> page (page2.html) fetches a new image. Each item fetched includes a Referrer header indicating the URL of the page it was fetched from.

If a stream is a HTTP stream containing a Referrer header, it is likely that the request for the originating page also came from the same source. Searching for the source which in the recent past requested the referred URL allows us to attribute the present stream to the source. We can use this information to reduce the energy term for the relevant source.





## **5.1 System Setup**

The requirements of systems used and the test network setup details are as follows :

### **5.1.1 System Requirements**

The hardware used was a standard desktop with a Pentium IV and 562 MB RAM. The operating system used on the desktop PC was windows XP with SP3 pack. The program was developed in visual basic 6.0. The database used to store the captured data is MS Access 2007. There were no special resources or tweaking used because more than performance aspect it was validation of the concept that was the primary focus of the experiments as part of this work.

### **5.1.2 Network Setup**

To test the algorithm it was essential to setup a network pertaining to our requirement. It consists of a small private LAN of 2-3 computers which were connected to a switch. Any number of hosts can be added to the LAN by connecting it physically to this switch. Logically a private IP address space was used to assign static IP's to each host. This switch in turn was connected to the gateway (as shown in Figure 4.1). The gateway here used is a simple windows PC with 2 NIC cards, one to be connected with the private network and other with the public IP (here again a private address was taken) is connected to Internet. The capturing device is placed between the Gateways public interface and Internet. This capturing is done on a windows PC with the program developed running in order to capture the network traffic.

## **5.2 Implementation of Traffic Capturing Module**

This is primarily the first step in network forensic process. This component of the program takes network traffic as data. In traditional methods this data is logged or stored on a secondary device. The system keeps storing the data until some crime has occurred.

At that particular time this data is pulled out & is given as input to the forensic analysis system to trace & find evidence of crime & the identity of the person behind the crime. In our program we design a different model, in which real time data is taken. Instead of having the data captured in the first place and using it at time of crime as input for analysis, real time data is taken directly. As the application runs, the data is picked up from the network. Storage requirements in this kind of design is less as compared to traditional ones. The factors that were explored are :

- Storage requirement of capturing machine
- Data capture rate
- Capture data and analysis done later when crime is committed
- Capture data by Filtering, specific to an incidence

Since the intension is to capture all the traffic in the LAN, a raw socket is created using windows socket API. RAW is a special type of socket that gives access to packet headers also along with the data. This socket is then binded to the interface which needs to be monitored. This binding is an essential process as then only the program can take data from the binded interface. In case of multiple NIC (network interface card) in the system, the card used specifically for data capturing need to be mentioned in the bind. Finally the socket needs to be set in promiscuous mode.

Promiscuous mode or *promisc mode* is a configuration of a network card that makes the card pass all traffic it receives to the central processing unit rather than just packets addressed to it — a feature normally used for packet sniffing. When a network card receives a packet, it normally drops it unless the packet is addressed to that card. In promiscuous mode, however, the card allows all packets through, thus allowing the computer to read packets intended for other machines or network devices. Figure 5.1 shown the screenshot of the source identification program. When start button is pressed then the capturing of network traffic starts.

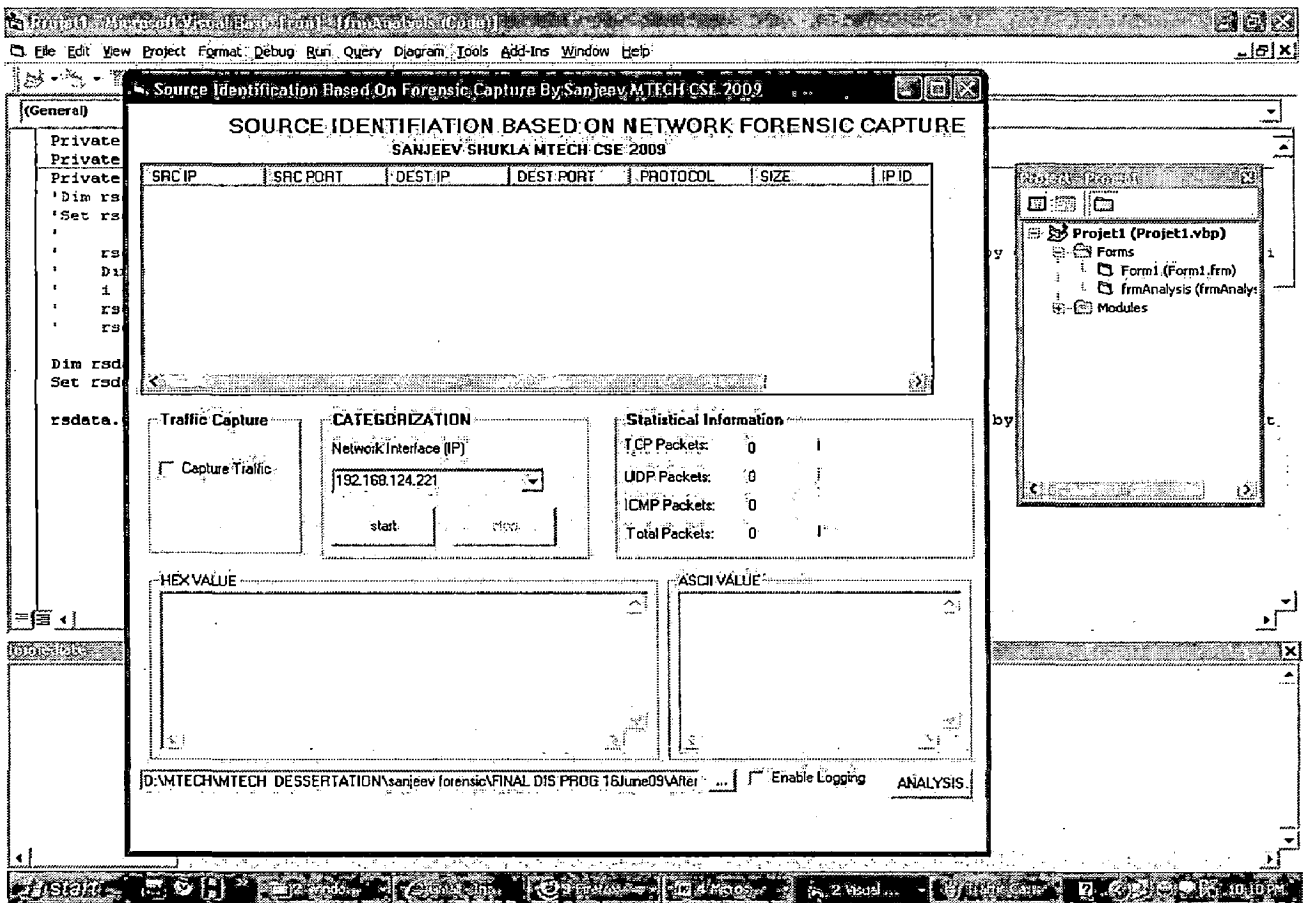


Figure 5.1 : Traffic Capturing Interface

### 5.3 Module displaying the contents

This module is build to display the content of the payload and header of the packet to forensic analyst. This kind of capture & display is though an invasion of privacy but can be performed after having permission from competent authorities. The legal aspect of this needs to be looked into before moving forward but this part (privacy) has been ignored in our dissertation. The header is displayed in a tabular format after categorizing various section of it. This includes showing source & destination IP address, source & destination port number, size of packet, type of protocol etc as shown in Figure 5.2. The payload is also captured and displayed in HEX form and in ASCII format. It gets displayed when a packet is clicked as highlighted by blue color.

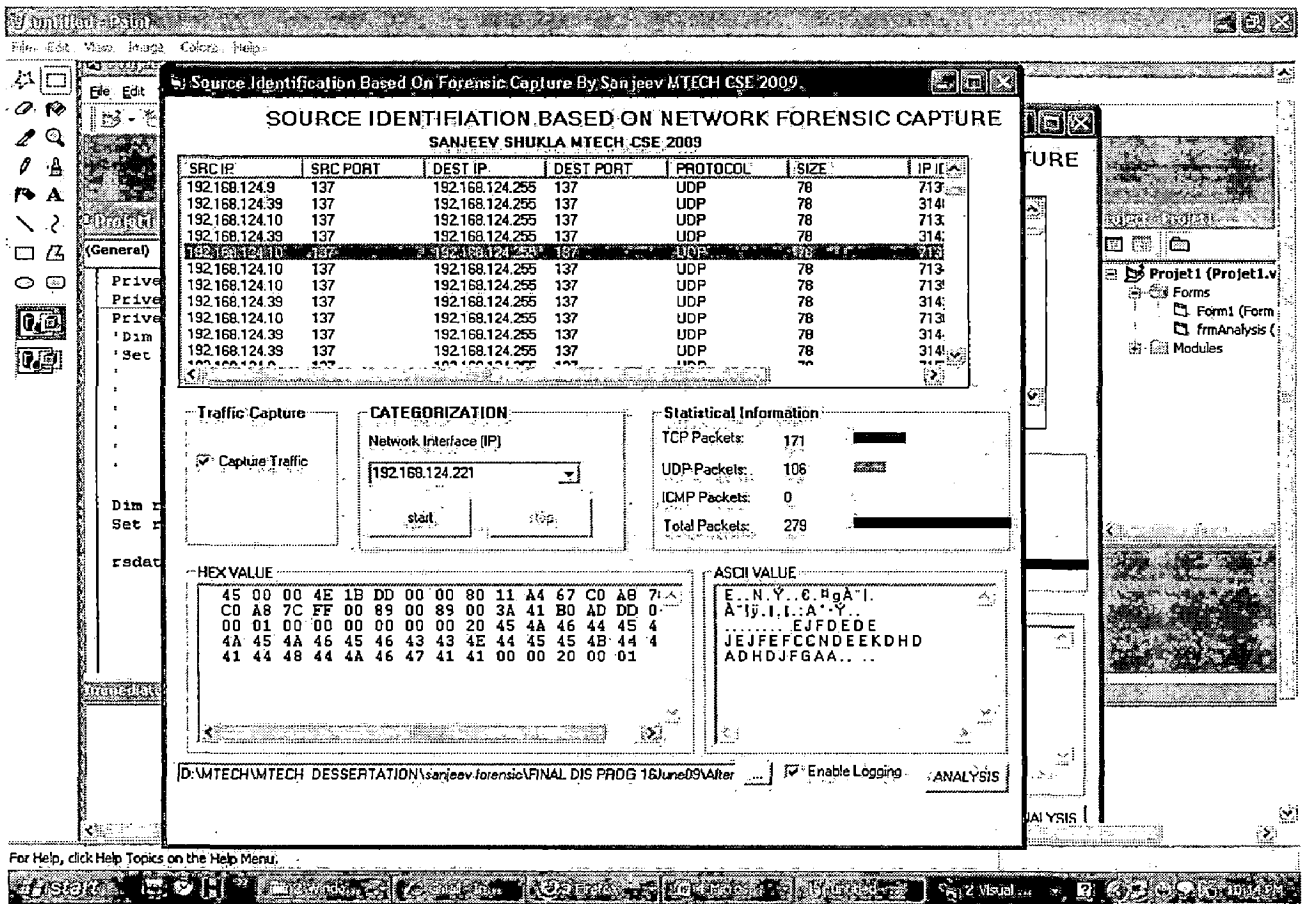


Figure 5.2 : Content Display both header and payload

## 5.4 Logging Module and Statistical Information generation

This is a significant part of program which essential records and saves all the data acquired over the network. The network traffic saved is not a raw data rather is a surmised view which is stored by performing some methods likes categorization. This saved information is later used for analysis purpose. The program saves the data in two different formats. One function logs all the record in txt format while the other functions save the records in MS access 07. Different formats help in having a backup of data and also the application can use them in different ways. For example the tabular form of MS Access helps us to pick up columns of a field which is not possible in txt format.

Statistical information is generated for providing assistance to analyst to have an idea of the type of protocols is used by each packet at a glance. This type of information can

have multiple utilities from network troubleshooting (ex a specific protocol is used to much etc) to identifying host, as we may know which type of protocol is used.

### 5.5 Module for source attribution

This is the part of module where streams are applied to the algorithm. Each stream containing information based on certain attributable property is evaluated and based on it the energy function decides the correct source. The information generated by each type of unique attribute is then correlated to identify the sources. The algorithm is shown in section 4.1.1 and the flow chat in figure 5.4. After identifying a stream to be part of a source it is assigned as S1 (source 1), S2 etc in our final analysis. This way we know the total number of host in the system and also the stream or packets associated with them.

Finally this complete information of analysis is shown in tabular format by populating the data from the database. Also graph of this data is generated to have a correct picture of the outcome as shown in figure 5.3.

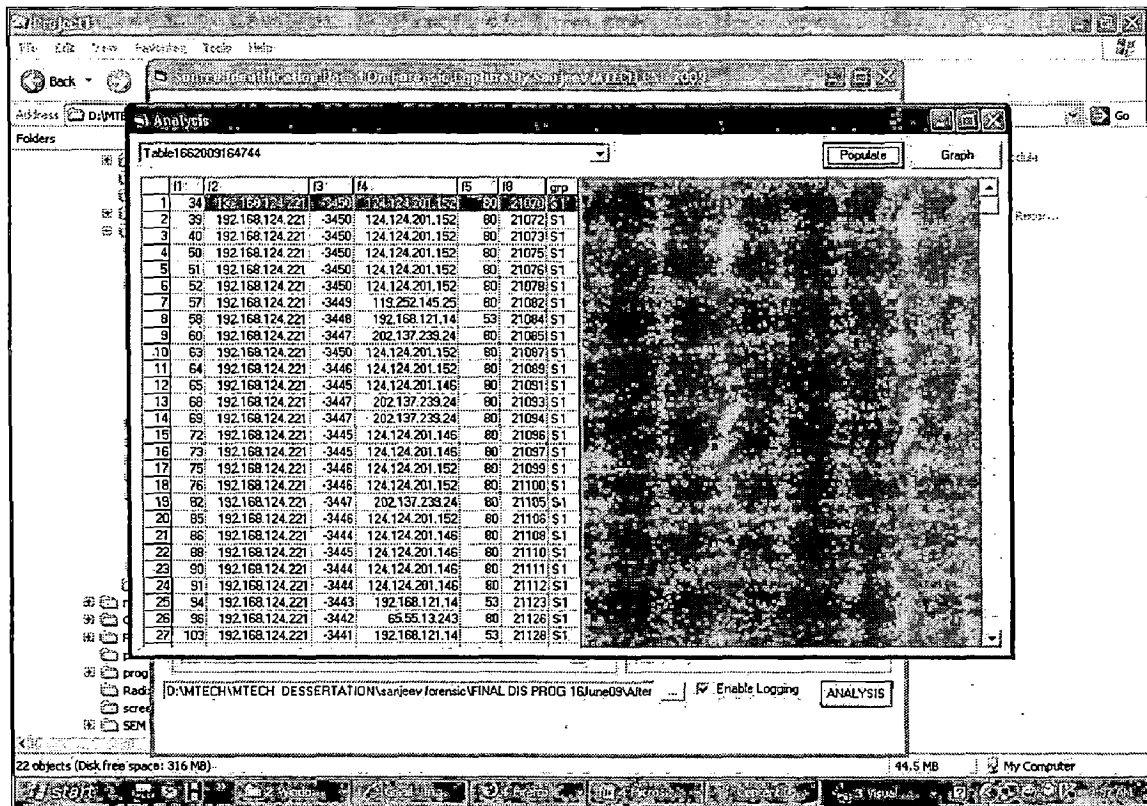


Figure 5.3 : Analysis based on algorithm

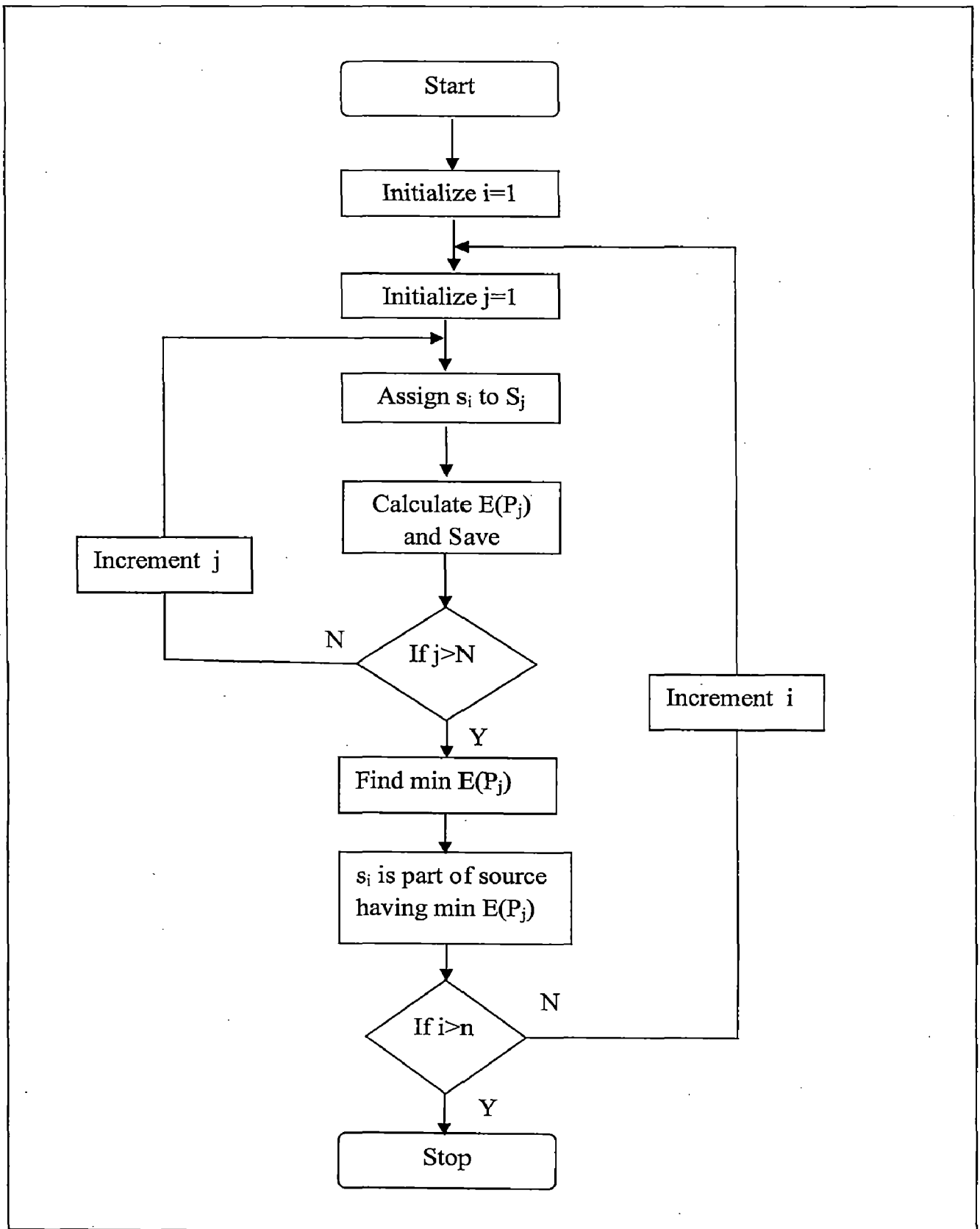


Figure 5.4 : Flow Chat of the source identification algorithm

### 6.1 Display and Discussion of Results Obtained

The results so far obtained from our program are displayed and discussed with respect to each unique attribute. These attributes which identify source based on its unique characteristics are discussed one by one.

#### 6.1.1 Internet Protocol Identification

The results of IP ID attribute is shown in Figure 6.1 by a plot of it with all the outbound packets from the network. Here we had taken 1 windows PC & 1 linux PC which were used to browse internet for 5 min. The large number of points scattered around in the graph is the result of linux host using secure random IP ID generator.

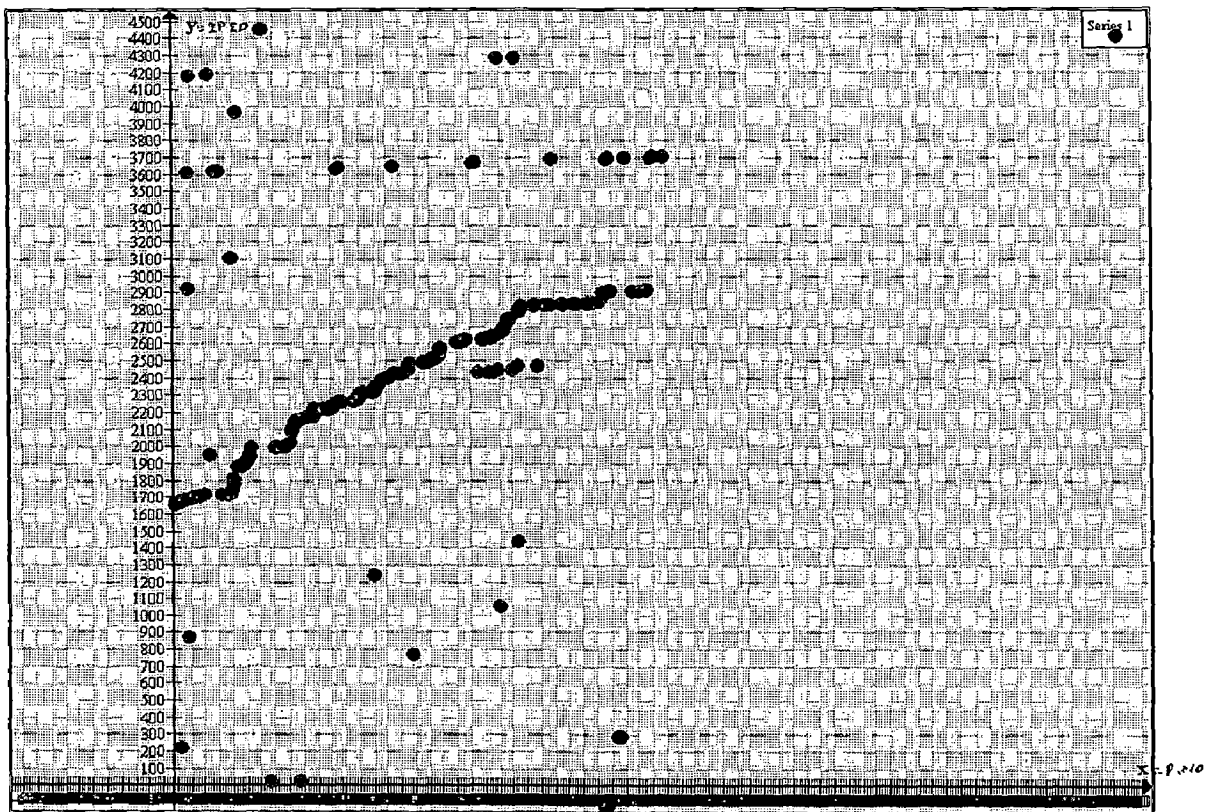


Figure 6.1: IP PD plotted against packet number for 1 windows & 1 Linux host

When processing a stream originating from the Linux system, the algorithm tended to place the stream into a new source because the stream the stream increases the energy of system. However, for packets from the Windows system, the algorithm showed a decrease of energy (negative  $\Delta E$ ) when placing the stream into the correct configuration. Thus the algorithm shows strong convergence. The final result was that for the Linux system there were lot of separate sources, most containing one stream. For the Windows system, all streams were correctly matched into the same source resulting in an IP ID plot shown in Figure. 6.2. The source shows a continuous increase of IPIDs with few missed packets.

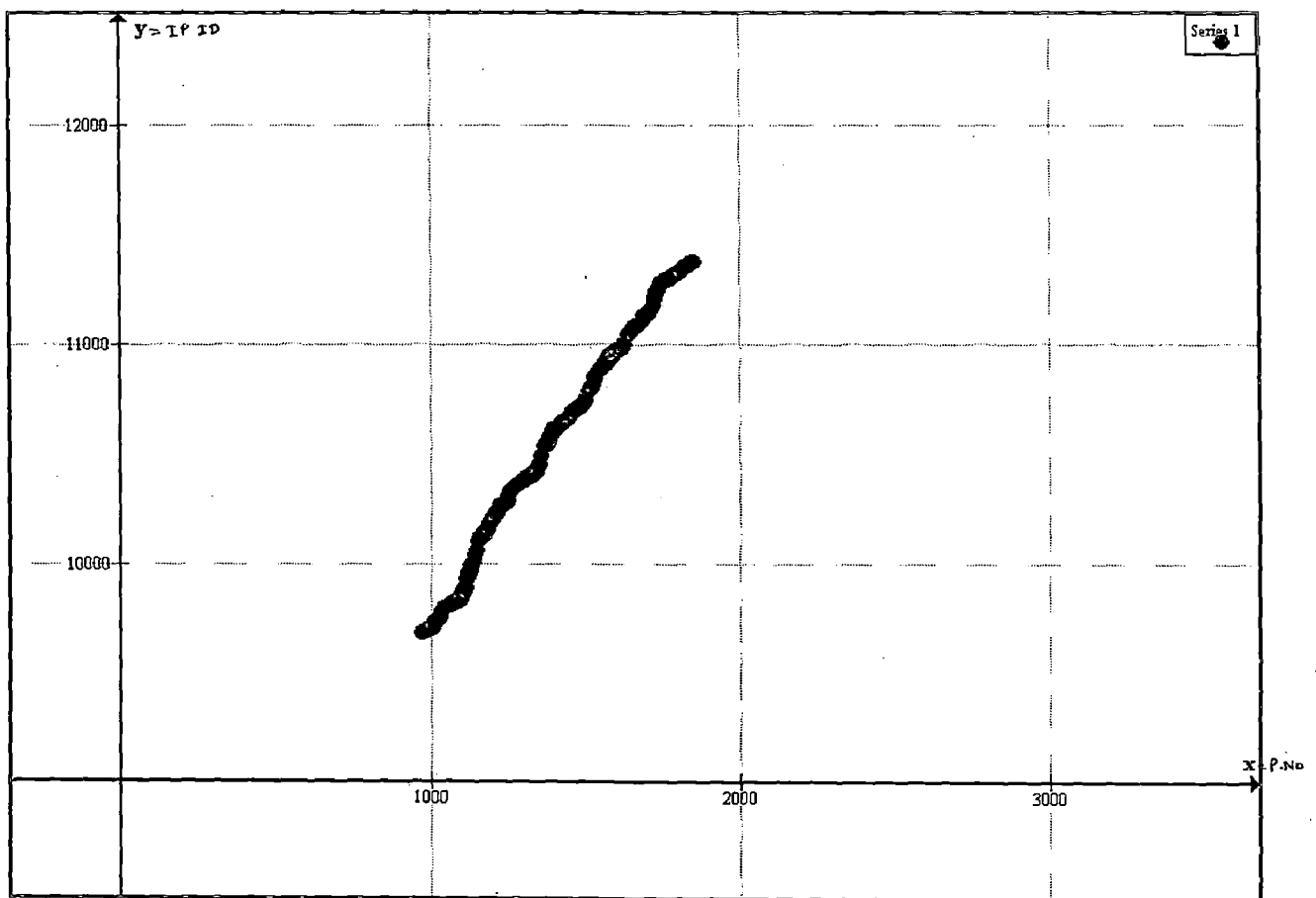


Figure 6.2 : Packets Belonging to Windows machine



## 6.1.2 Timestamp

It is clearly evident from above results that IP ID attribution techniques are not suitable for linux host which use secure IP ID generation algorithms. By applying TCP timestamp options from TCP header, linux based hosts can be determined. This is because windows XP does not send TCP timestamp option by default hence we don't get a value for it. Figure 6.3 clearly shows for the packet selected in blue color line the details in the below window indicated under TCP in Options filed (in blue again) that it does not have any TsVal (Timestamp value) field indicating the value of timestamp.

The screenshot shows the Wireshark interface with a list of network packets. Packet 61 is highlighted in blue. The detailed view for packet 61 shows the following information:

- Frame 61 (62 bytes on wire (62 bytes captured))
- Ethernet II, Src: Hewlett-Packard (00:15:00:0e:8c:6e), Dst: Cisco\_2c:af:fc (00:08:e2:2c:af:fc)
- Internet Protocol Version 4, Src: 192.168.124.221, Dst: 203.199.74.8
- Transmission Control Protocol, Src Port: 62761, Dst Port: http (80), Seq: 0, Len: 0
  - Source port: 62761 (62761)
  - Destination port: http (80)
  - [Stream index: 4]
  - Sequence number: 0 (relative sequence number)
  - Header length: 28 bytes
  - Flags: 0x02 (SYN)
  - Window size: 65535
  - Checksum: 0x7141 [validation disabled]
  - Options: (8 bytes)
    - Maximum segment size: 1460 bytes
    - NOP
    - NOP
    - SACK permitted

The TCP Options field is highlighted in blue, indicating that no timestamp value (TsVal) is present in the options.

Figure 6.3 : No timestamp value for Windows system

Linux sends TCP timestamp options by default which is shown in Figure 6.4 where the Tsvval is shown in blue color line under TCP in options fields.

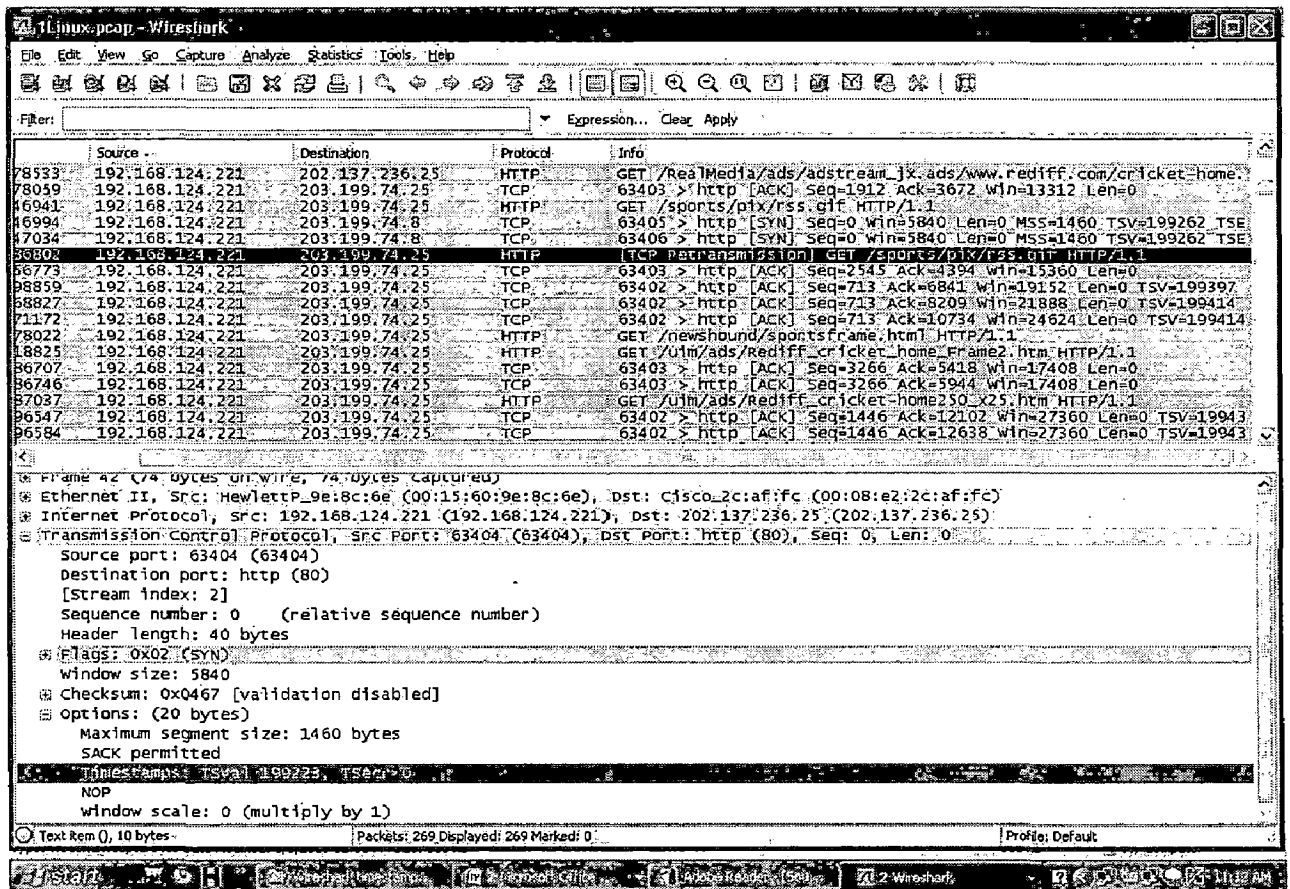


Figure 6.4 : Timestamp value indicated in Linux systems

Finally plotting the values of source timestamp obtained from the TCP header option field with the real time of packet captured is shown in Figure 6.5. The graph does not shown any windows machine because it does not send timestamp and only shows a linux machine (single line).

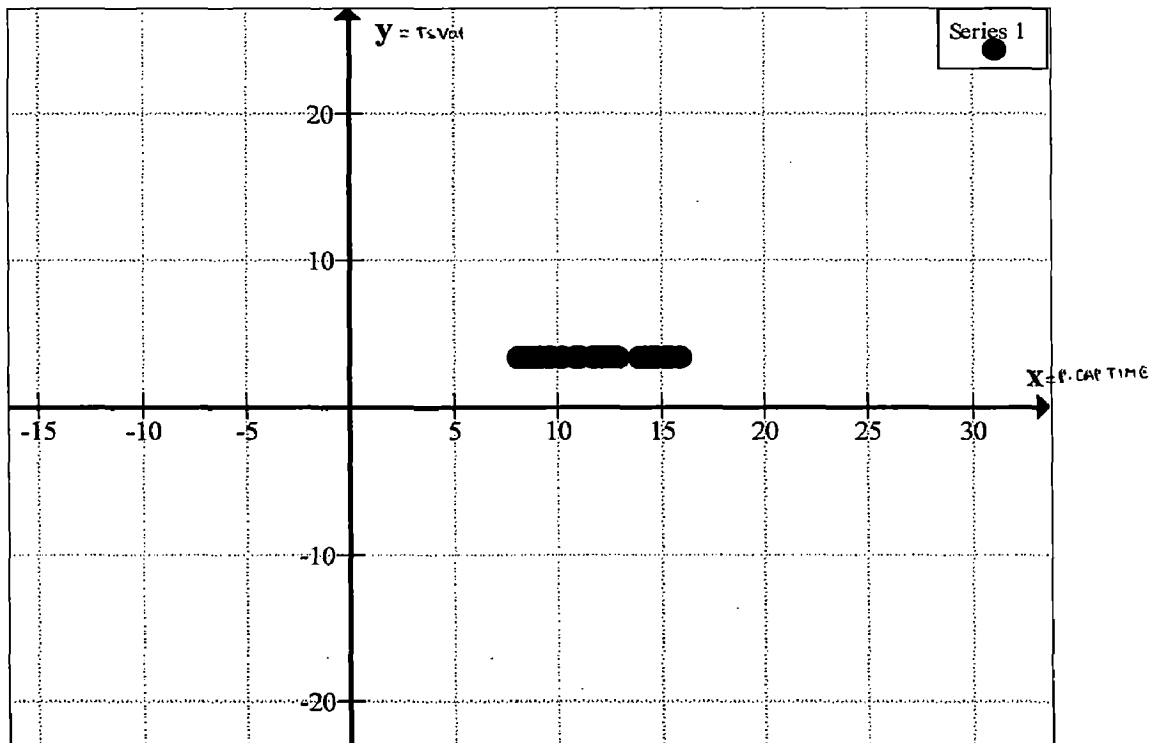


Figure 6.5 : Timestamp plotted against packet capture time

### 6.1.3 HTTP Referrer

HTTP Referrer is part of HTTP protocol which indicates that the current request was referred to from another page. The snapshot in Figure 6.6 shows the highlighted packet in blue color using HTTP protocol having a referrer header. Here the URL shown by referrer is [www.rediff.com](http://www.rediff.com) indicating if we search for sources which in the recent past requested this URL, it's quite likely that the request for originating page also came from same source.

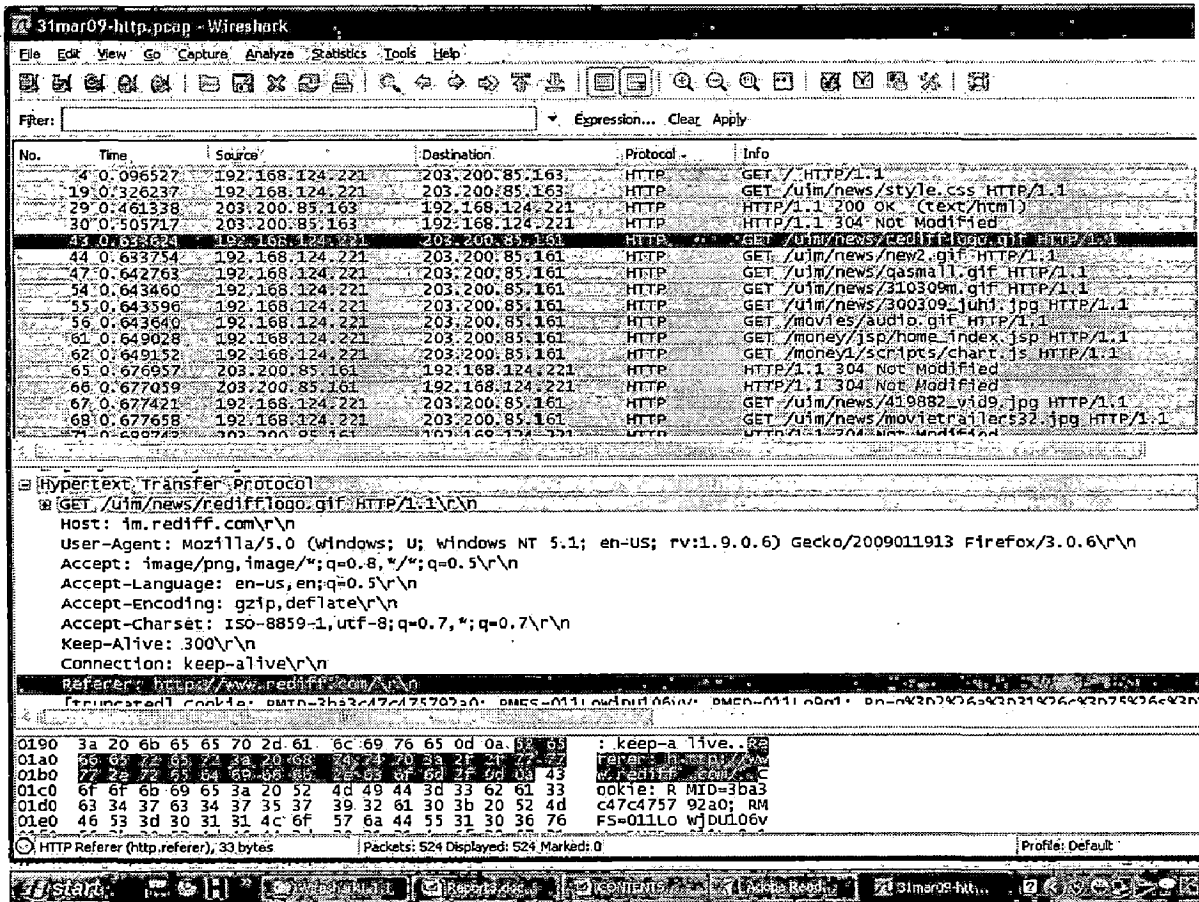


Figure 6.6 : HTTP Referrer header

## 6.2 Limitations

The advantages of the technique used have been enumerated in sufficient details. However it is essential to know the weaknesses of the program as well. The limitation which became apparent during this dissertation are :

- Limitation of attributes for some situation and there strong confidence level for other. For example timing analysis may be a less reliable for sources with NTP ( Network Time Protocol) synchronized clocks since the clock skew is very small in that case. But in other case this attributes shows high level of confidence for source identification.

- A single attribute may not be able to completely identify the source. So we need combination of them and they might also get nullified based on some network or system configuration. Ex if we use Linux machine, then our IP ID based analysis fails similarly if we used windows then our timestamp based analysis fails.
- Traffic captured data should be in some common format well defined so that it can be exported by any analysis system. Absence of which causes to write the capturing program as a whole to have the data in a format which can easily be analyzed by analysis program developed.

### **6.3 Validation of Results**

The results so far discussed about the program and its identification process needs to be validated. This is done by placing another traffic capturing machine inside the private LAN as shown in Figure 6.7. This helps us to capturing the network traffic inside the LAN which contains the source & destination IP address including other useful information. Inside the LAN private IP addresses are not hidden unlike in NAT and we exactly know which machine is the originating system of the packets which we capture. This information captured inside the LAN is used to match with the results obtained by our algorithm.

Only single host having windows operating system is used in the experimental setup. The other host with Linux is not used as it will fails in our IP ID analysis because of random IP ID generation. The capturing starts at both the capturing systems simultaneously. The host is allowed to browse for some time and all the packet send in that LAN are captured by both traffic capturing machines.

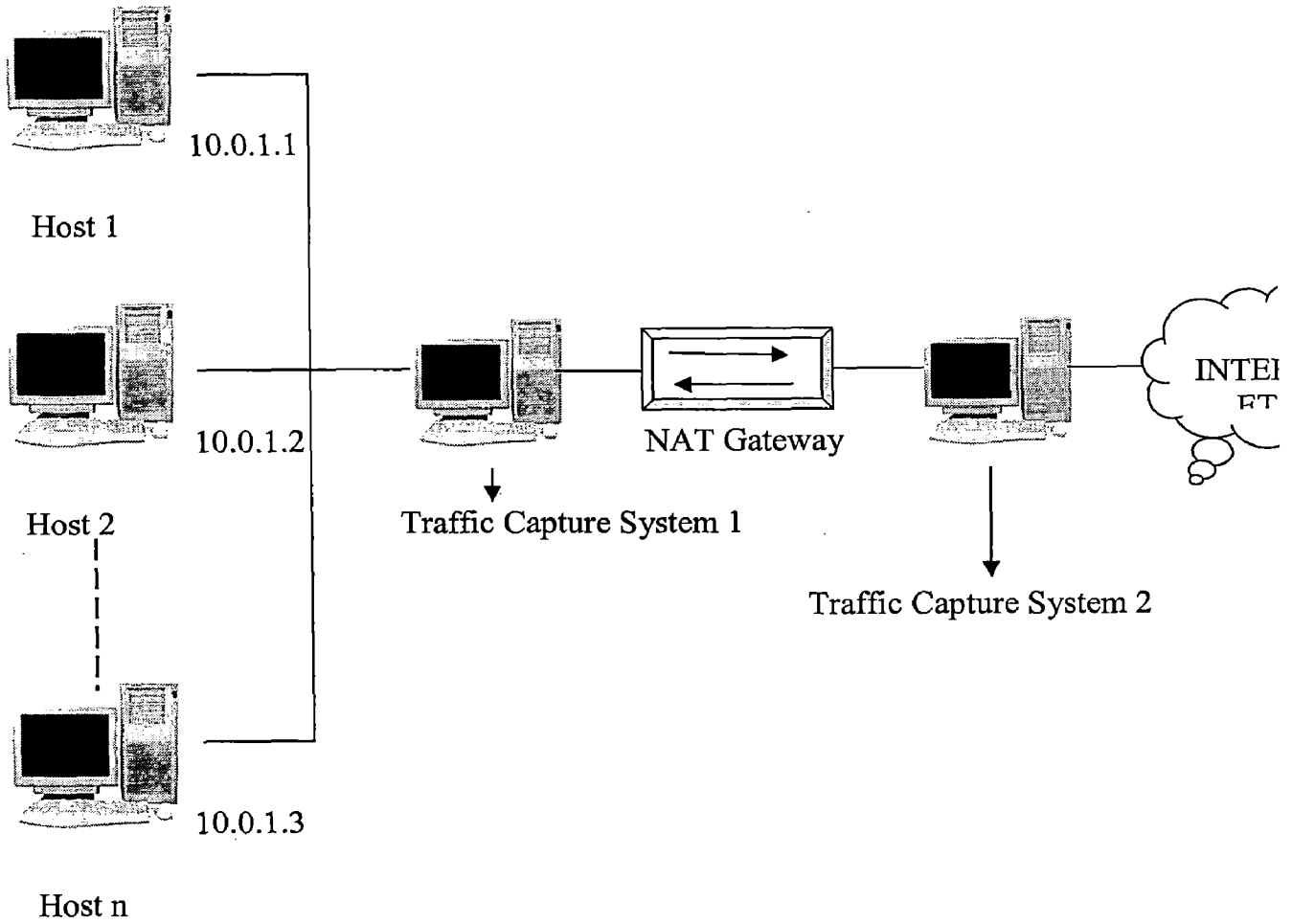


Figure 6.7 : Network Setup for validating the results

The capturing machine inside the LAN has correct information about the identity of the source sending the packets whereas the traffic capturing system outside the LAN placed after NAT uses the attribution method implemented in the algorithm to find the source. Figure 6.8 shown below displays the capturing system inside the private LAN configured for our test purpose. It can be seen from the screenshot that the total number of packets obtained after traffic capturing is saved in the database for further validation with the results obtained by other capturing machine.

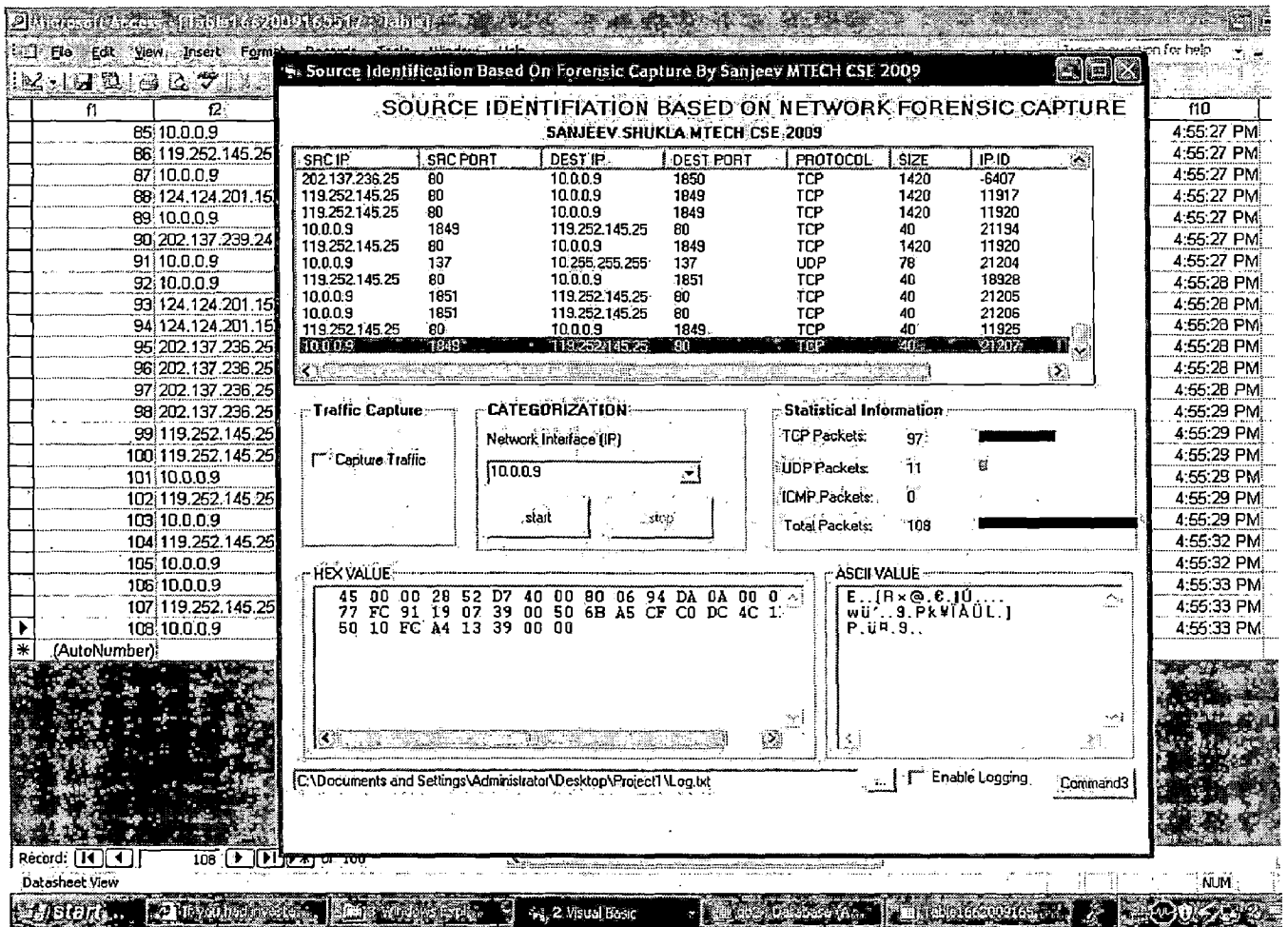


Figure 6.8 : Data Captured by capturing system inside private LAN

Similarly network data is captured by other system as well. Figure 6.9 shows it along with the results of the analysis. As displayed the analysis identifies the packets and indicate it by S1 (symbolizing source 1) S2 etc based on number of sources it has found. Since we have only 1 host which is transmitting packets, we should only get 1 source. The same is shown in the last column.

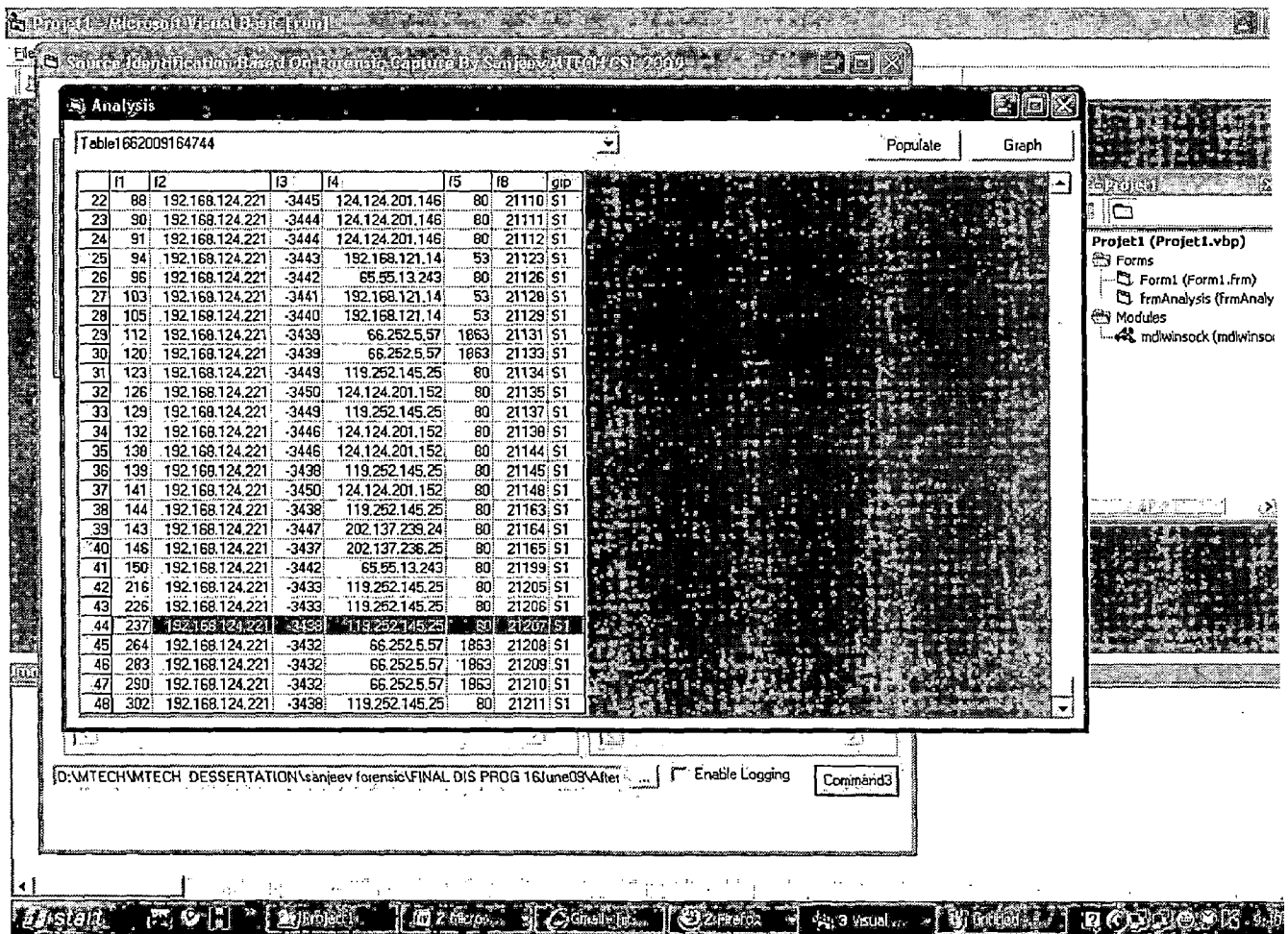


Figure 6.9 : Data Captured and Analyzed with Results

It's interesting to see the blue line in both the figures which is deliberately done to highlight that the source (10.0.0.9) generating IP ID (21207) in Figure 6.8 has same IP ID (21207) with only the source IP changed to the NAT public interface (here it is 192.168.124.221). Similarly the other IP ID's can also be matched to validate that it was the same source which has generated the packet with IP ID that has been identified by the our program as well.

The Graph also developed with the data acquired points to the same results of single host. The plotting of IP ID's with packet number for outbound traffic for this data acquired is shown in Figure 6.10 which indeed is a single line symbolizing a single host.



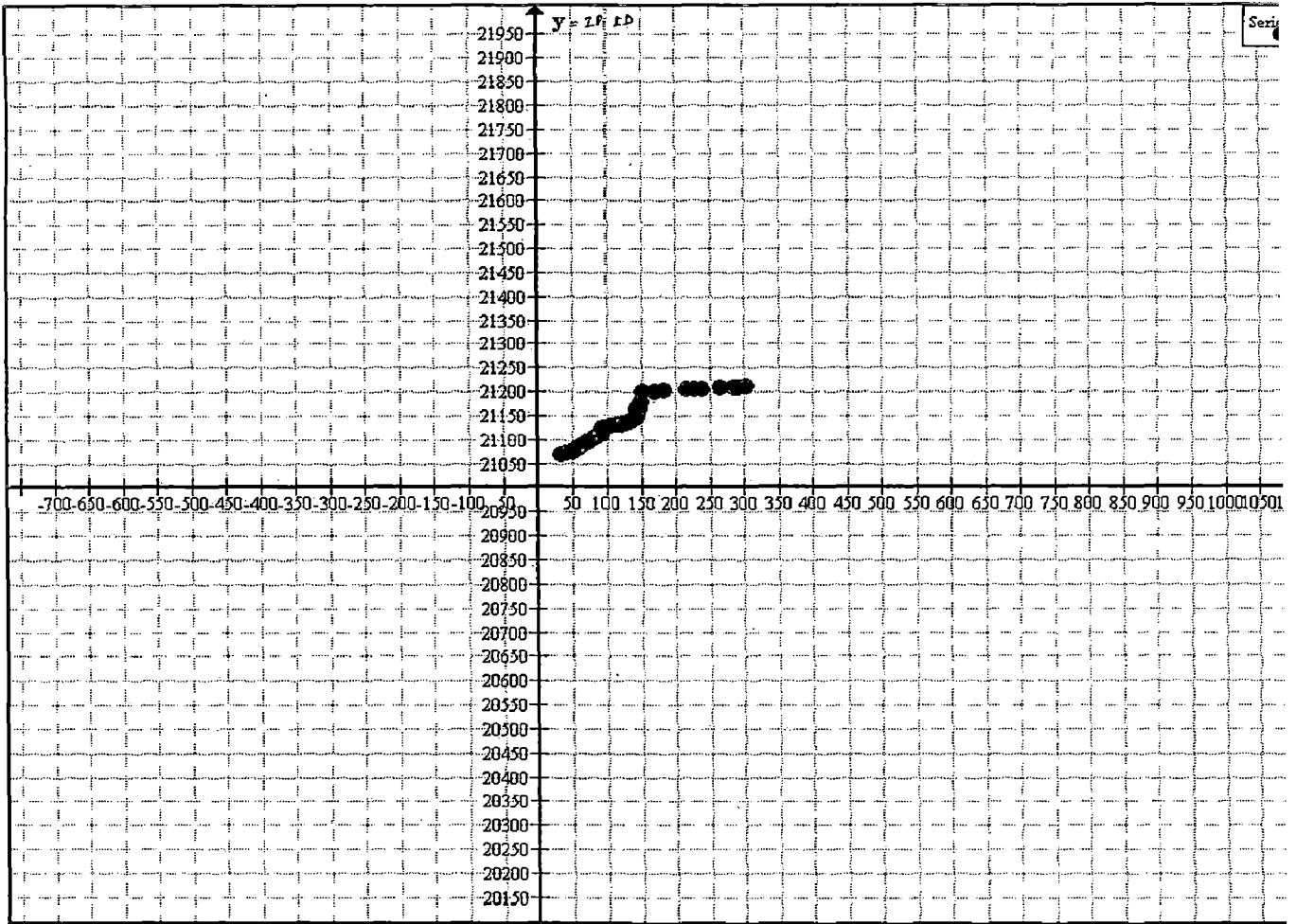


Figure 6.10 : Graph of Validation data

## **7.1 Conclusion**

Identifying source of interest hidden behind a network address translated gateway using attribution method will be an important factor of digital investigation in future. Techniques such as these would address some of the many challenges that are faced in cyber crime investigations. In this work it is shown that by using certain unique characteristic of the source identified at each layer of OSI, one can trace back or ascertain the source of the originating packet. The algorithm thus developed relies on the combination of number of unique characteristics (attributes) specific to a source offered by each layer of the OSI model, allowing identification of source machines. The results obtained clearly vindicate the properties of the attributes in identification of sources. IP ID attribute based analysis helps us to differentiate based on operating system. Here host having windows OS is identified because of its sequential packet generator whereas Linux hosts fail because of random packet generator. Similarly timestamp attribute of TCP layer helps us to identify host having Linux as it send TSVal (timestamp value) which are ON by default and fails with windows OS as its timestamp Option filed is OFF by default. Combination of both these attribute also tells us total number of hosts in the network.

## **7.2 Scope for Future work**

Before using this technique in field, further extensive experiments needs to be done. The system can be enhanced by adding more number of unique attributes & then co-relating all to form a formidable system. Also the bigger the size of LAN the more traffic needs to be captured and processed. This can be compensated by using cluster computing, grid computing or multi-threading to speed up the tasks. It will also be interesting to use distributed approach if there are multiple capturing points and the data thus acquired can be reduced/compressed and send for processing.

With cyber crimes increasing and government & business community becoming sensitive to security aspect there is lot of activity & scope in this area in near future.

Details for future work are highlighted as below:

- Identification algorithm can be enhanced by adding more number of attribute. Here we have applied two attributes only but by adding more number of attribute (like HTTP Cookies or Proprietary Protocols), it will widen the scope and bring more information pertaining to host identity.
- The size of LAN is another factor which needs deliberation. The present work has a very small LAN for testing (of 2-3 nodes) but if LAN has large number of nodes, then to cater to the enhanced processing of large number of packets or can use cluster, grid or multi-threading based approach to speed up the process. In this the capturing machine receiving large number of packet from the LAN acts as a single job and sends it to a cluster which divides it into multiple jobs and assign it to each node for processing. Hence the processing is become faster.
- Another way for big sized LAN having multiple VLAN'S is to use distributed approach which can also be applied and tested for identification. In this each VLAN can have a small program(agent) to capture data. Thus in a whole network we will be having multiple agents which will capture data and send them back to main server. All the information pertaining to a specific VLAN will be handled by these agents and they can query to each other and also to server for further information.

## References

---

- [1]. Wikipedia, the free encyclopedia, [http://en.wikipedia.org/wiki/Computer\\_forensics](http://en.wikipedia.org/wiki/Computer_forensics)
- [2]. Ian Walden, Computer Crime & Digital Investigation, Oxford University Press, 2007.
- [3]. Vincent J.M DiMaio and Suzanna E. Dana, Handbook of Forensic Pathology, 2<sup>nd</sup> edition, Taylor & Francis group, October 31, 2006.
- [4]. Shanmugasundaram K, Bronnimann H, Memon N. “ Payload attribution via hierarchical bloom filters “, In: Proceedings of the 11th ACM conference on computer and communications security, New York: ACM, 2004, pp. 31–41.
- [5]. Casey E. “ Network traffic as a source of evidence: tool strengths, weaknesses, and future needs ”, Digit Invest 2004, pp. 28–43.
- [6]. Shanmugasundaram K, Memon N, “ Network monitoring for security and forensics ”, In: Information systems security, ser. lecture notes in computer science. Berlin/Heidelberg: Springer, 2006, pp. 56–70.
- [7]. Report from the First Digital Forensic Research Workshop (DFRWS), “A Road Map for Digital Forensic Research” , DTR - T001-01 FINAL DFRWS TECHNICAL REPORT, Information warfare branch US, November , 2001.
- [8]. Karen Kent, Suzanne, Tim Grance, Hung Dang “Guide to Integrating Forensic Techniques into Incident Response” National Institute of Standards and Technology, US Special Publication 800-86, August 2006.

- [9]. Jung-Sun Kim, Dong-Geun Kim & Bong-Nam Noh, “A Fuzzy Logic Based Expert System as a Network Forensics” Fuzz- IEEE, 2004.
- [10]. Desmond LCC, Yuan CC, Pheng TC, Lee RS, “Identifying unique devices through wireless fingerprinting”, In: WiSec '08: proceedings of the first ACM conference on wireless network security. New York, NY, USA: ACM, 2008, pp. 46–55.
- [11]. Kohno T, Broido A, Claffy K, “Remote physical device fingerprinting”, IEEE Transaction on Dependable and Secure Computing 2005, vol. 2, Issue 2, pp. 93–108.
- [12]. Liberatore M, Levine BN, “Inferring the source of encrypted http connections”, In: CCS'06: proceedings of the 13th ACM conference on computer and communications security, New York, NY, USA: ACM, 2006, pp. 25–63.
- [13]. McHugh J, McLeod R, Nagaonkar V, “Passive network forensics: behavioral classification of network hosts based on connection patterns”, SIGOPS 2008, vol. 42, Issue 3, pp. 99–111.
- [14]. Cohen MI. “Pyflag – An advanced network forensic framework” In : The proceedings of the eighth annual DFRWS conference, vol.5, September 2008. Suppl. 1, pp. S112–120.
- [15]. Bruce J. Nikkel , “ A portable network forensic evidence collector”, published by Elsevier, The International Journal of Digital Forensics and Incident Response , Vol. 3, No. 3 (10.1016/j.diin.2006.08.012) , 2006.
- [16]. M.I Cohen , “ Source attribution for network address translated forensic capture “, published by Elsevier, Digital Investigation Vol. 5, Issue 3-4, pp. 138-145, doi:10.1016/j.diin.2008.12.002 , 2008.

- [17]. Goonatilake R, Herath A, Herath S, Herath S, Herath J, “ Intrusion detection using the chi-square goodness-of-fit test for information assurance, network, forensics and software security” , Journal of Computing Sciences in Colleges 2007, vol. 23, Issue 1, pp. 255–263.
- [18]. Bellovin SM, “ A technique for counting NATed hosts” , In: IMW '02: proceedings of the 2nd ACM SIGCOMM workshop on Internet measurement. New York, NY, USA: ACM.

## ***List of Publications***

---

- [1]. Sanjeev Shukla and R.C. Joshi, “ Designing a Framework for Network Forensic Analysis System ”, In the Proc. Of National Conference on Communication Networks (NCOCN’09), Kerala, India, Mar 2009, pp. 148-154.
- [2]. Sanjeev Shukla and R.C. Joshi, “ Distributed Agent based Integrated Network Forensic Analysis System ”, 4th International Conference on Information Technology (ICIT’09), Amman, Jordan, June 2009. [ Accepted in April 2009]
- [3]. Sanjeev Shukla and R.C. Joshi, “ Identifying source machine using attribution method for NAT data ”, The First International Conference on Networks & Communications (NetCoM-2009), Chennai, India, December 2009. [Communicated]