Grenze International Journal of Computer Theory and Engineering , Oct 2015

Network Security and Behavior Analysis of an Institute using Wireshark

Amanpreet Kaur[1] and Monika Sachdeva[2]
[1]Student Of M.Tech , SBSSTC/ Computer Engg. , Ferozepur ,India
e-mail - aman2s2@yahoo.co.in
[2]A.P, SBSSTC/Computer Engg. , Ferozepur , India
e-mail - monika.sal@rediffmail.com

***Abstract*—Security has become an important requisite due to the prevalent attacks and various other security issues that have made networks vulnerable to a great extent. There's a requirement to analyze the networks and diagnose the malicious packets travelling through it. This lead to the development of a number of packet analyzers that will monitor the network assets to detect their anomalous behavior and misuse. In our dissertation work, we use Wireshark as a packet analyzer which observed the communicating nodes and gathered data from them. Wireshark is an open source packet analyzer ,which was formerly known as Ethereal.**

Here we have monitored and analyzed the traffic of an institute using various protocols like TCP/IP, HTTP, ARP and ICMP. Wireshark observed data coming from certain IP addresses and captured packets that were exchanged by those nodes.. The outputs are shown in graphs namely Time Sequence graph, Round Trip Time graph and Throughput Graph. Protocol hierarchies are built which shows low , medium and peak loads. HTTP Statistics are built and Expert analysis is done. Certain attacks are observed on ARP, DHCP, DDOS and HTTP Spidering and they are shown through graphs as well. In order to resolve network problems, an exhaustive analysis of those areas or segments that are lower in performance is required. The graphs obtained here using wireshark help to interpret the efficiency and performance of the network of an institute taken.

***Index Terms*— Intrusion Detection System, Network Security , Wireshark.**

I. INTRODUCTION

Network security means to secure the electronic data while stored in networked systems or transmitted through networks from various vulnerabilities, attacks and threats [1]. The main goal of network security is to give people the freedom of using computer networks without fear of compromising their rights and interests. Network security involves a number of activities that protect the network and the network accessible resources from unauthorized access usually by the outsiders. Another feature is Intrusion Detection System (IDS) ,it is a process of detecting intrusion in database, network or any other device for providing secure data transmission. Intrusion detection system (IDS) is a device or software application that monitors network and system activities for malicious activities or policy violations and produces report to a management station [2].When you run the Wireshark program, the wireshark graphical user interface shown in Figure 1. will be displayed.

In our work, we have analyzed the network traffic of an institute from 30/01/2014 to 06/02/2014 for around 8 days for different durations and captured traffic using Wireshark, which is an open source packet analyzer. It provides facility named TCP Stream for reading data from source to destination. The results are obtained for six Traces by using the Wireshark tool, results are visualized with protocol usage at Low ,Medium and

Grenze ID: 01.GIJCTE.1.1.546

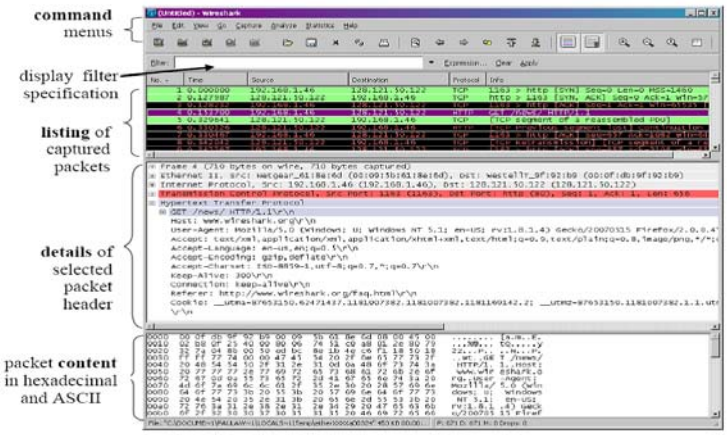


Figure1. Wireshark window

Peak loads, Request and Response analysis is done, Errors, Warnings are detected through Expert Info analysis, Time Sequence graphs, Round Trip Time (RTT) graphs and Throughput Graph are also analysed. While using wireshark some captured traces are too large, so graphs are drawn packet by packet. So that's why some of the graphs have been reduced to capture only the important details. As we analyzed the traffic , the following table shows the values of various parameters that we observed.

TABLE I. SUMMARY OF TRACES CAPTURED

	Capture Time	Duration	Captured Pkts.	Avg. Pkt /sec	Avg. Pkt Size	Bytes	Avg. Bytes/sec	Avg. Mbits/sec
Trace 1	12:21 – 12:28	7 min	14039	423.319	104.824	1471621	3476.385	0.028
Trace 2	12:43 - 12:50	7 min	1101	2.531	533.480	587362	1350.193	0.011
Trace 3	10:34 – 10:55	21 min	68016	53.973	191.975	13057369	10361.474	0.0831
Trace 4	12:01 – 12:22	21 min	87524	68.133	445.808	39018880	30374.230	0.243
Trace 5	09:01 – 09:54	53 min	120091	37.203	131.720	15818378	4900.425	0.039
Trace 6	09:32 – 10:25	53 min	104274	32.590	143.920	15007070	4690.401	0.038

II. LITERATURE SURVEY

The proposals common goal is to study the network traffic and analyze it by using some network security tool in order to have better understanding about the various threats and attacks that can affect the network. For this it is very important to go through certain research papers that deeply discuss the network tools and their results. A few papers enumerated are

Shilpi Gupta, et.al, explained about Intrusion Detection System which is a process of detecting intrusion in database, network or any other device for providing secure data transmission. The author purposed an IDS which detects intrusion in network to provide safe and intrusion free network by using Wireshark. Aamir Hassan discussed about all the possible tools and techniques that attackers use to compromise the network. The purpose for exploring these tools will help an administrator to find the security holes before an attacker can. It is important to note that most of the attention in network security is given to the router**,** but far less attention is given to securing a switch. Usha Banerjeein,.et.al illustrated the functionality of Wireshark as a sniffing tool in networks. Testing has been achieved through experimentation on a real time network analyzed by Wireshark. This paper highlights the working of Wireshark as a network protocol analyzer and also accentuates its flexibility as an open source utility to allow developers to add possible functionalities of intrusion detection devices in it. Inferences have been made which clearly depict Wireshark's capabilities highlighting it as a strong candidate for future development into a robust intrusion detection system. Joshua

L. Davis has discussed about capturing the traffic using wireshark and producing network usage baselines.The paper has proved that despite limitations in Wireshark for handling large capture files , there is a way to manipulate data to create comprehensive network-usage baselines. Through the development of this methodology, the author hopes to begin some open source projects to help fill this void while also intending on improving Wireshark's capabilities. Mohsin Khan investigated how DHCP Client/Server request and reply messages work and what values and parameters are considered during this whole process. In this research we capture DHCP packets by using wireshark to deeply investigate and analyze them. On a network, when data is transferred between the hosts, it is passed through several stages. Data is actually passed through a very complex process at the sender and receiver than it apparently looks to be. During transmission data is broken down into smaller chunks of data so that they can be carried on the wire. These chunks are given appropriate headers, encapsulated and then passed through several layers to reach the destination. Justin Jay Lister gave an introduction to computer security by identifying the confidentiality, integrity and availability issues of information security. He also examined many of problems and vulnerabilities. Some statistics of intrusions is presented to show that there is still need for more effective security mechanisms.Emilie Lundin done research in the intrusion detection area. He described the design and implementation of specific intrusion detection systems. His survey focused on presenting the different issues that must be addressed to build fully functional and practically usable intrusion detection systems (IDSs). He stressed on more work in field of privacy enhancing techniques such as third party analysis of log files and detection output.

III. Problem Formulation

To analyze traffic behavior pattern of an institute under peek , medium and low loads and to find out various Errors ,Warnings , and Malformed packets to indicate possible attacks.

IV. Methodology and Experimental Setup

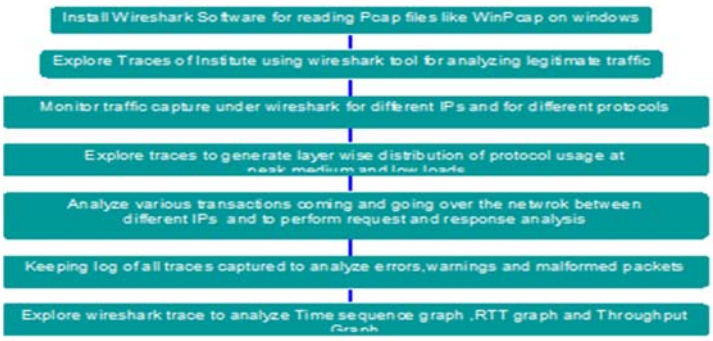


Figure 2. Methodology For Using Wireshark

a. Various cable taps,hubs,switches, etc. can be used to attach a sniffer to a network
b. Use laptop to run wireshark and a small hub attached to it and some network cables for troubleshooting.
c. Install a small hub between server and the switch and connect the wireshark laptop to it.Wireshark will then see all the traffic going to and coming from the server.

Figure 3. Wireshark placements using a Hub in an Institute

V. TRAFFIC PER PROTOCOL

By identifying the protocol distribution of captured traces, the following results are obtained shown in the table below. These tables depicts the values of various parameters namely percentage of packets, number of packets, percentage of bytes, bytes and Mbit/s in TCP Protocol which are observed in different traces as Low, Medium and Peak Load. Each row contains the statistical values of one protocol.[8] The table shown below displays the statistics for different traces that we obtained with respect to the protocols used.

TABLE II . SUMMARY OF PROTOCOL DISTRIBUTION ON THE BASIS OF MBITS/S

	Trace1	Trace 2	Trace 3	Trace 4	Trace 5	Trace 6
	Duration 7 min (Low Load)		Duration 21 min (Peak Load)		Duration 53 min (Med. Load)	
IPv4	0.007	0.011	0.054	0.000	0.020	-
UDP	0.011	0.000	0.029	0.014	0.008	0.007
NetBIOS Name Service	0.003	-	0.008	0.003	0.003	0.004
Domain Name Service	0.001	-	-	0.003	0.002	0.001
Data	0.001	-	-	-	0.001	0.005
HTTP	0.000	-	0.006	0.056	0.000	0.004
Dropbox LAN Discovery Protocol	0.000	-	0.000	0.001	0.001	-
NetBIOS Datagram Service	0.000	-	0.000	0.001	0.000	-
SMB	0.000	0.000	0.000	0.001	0.000	-
SMB Mail Slot Protocoln	0.000	0.000	0.000	0.001	0.000	-
Microsoft Window Browser Protocol	0.000	0.000	0.000	0.001	0.000	-
Data					0.000	
BOOTP	0.001	-	0.001	0.001	0.001	0.001
Teredo IPv6 over UDP Tunneling	-	-	-	-	0.000	-
IPv6	0.011	-	0.023	-	0.000	0.015
Open VPN Protocol	-	-	-	-	0.000	-
Malformed Packet	0.000	0.000	0.000	0.000	0.000	0.000
Network Time Protocol	0.000	0.000	0.000	0.000	0.000	0.000
Packet Cable	0.000	0.000	0.000	0.000	0.000	0.000
SEBEK-Kernel Data Capture	-	-	-	-	0.000	-
Data	-	-	-	-	0.000	-
Licklider Transmission Protocol	-	-	-	-	0.000	-
Data	-	-	-	-	0.000	-
Canon BJNP	-	-	-	-	0.000	-
IGMP	0.000	0.000	0.007	0.008	0.000	-
TCP	0.001	-	-	0.211	0.011	-
SSL	0.000	0.002	0.000	0.008	0.003	0.000
HTTP	0.000	-	0.006	0.056	0.000	0.004
Online Certificate Status Protocol	-	-	-	-	0.000	-
Media Type	-	-	-	-	0.000	-
Line Based Text Data	0.000	0.000	0.000	0.000	0.000	0.000
Data	-	-	-	-	0.000	-
NetBIOS Session Service	-	-	-	-	0.000	-
SMB	-	-	-	-	0.000	-
SMB Pipe	-	-	-	-	0.000	-

Protocol						
Microsoft Win Lanman Remote APIProtocol	-	-	-	-	0.000	-
SMB2	-	-	-	-	0.000	-
ICMP	0.000	-	0.004	0.000	0.000	-
ARP	0.009	0.000	0.006	0.008	0.007	0.005
IPv6	0.011	-	0.023	-	0.002	0.015
TCP					0.000	
HTTP	0.004	-	0.003	-	0.000	0.003
Logical Link Control	0.000	-	0.001	0.000	0.000	0.001
Spanning Tree Protocol	-	-	-	-	0.000	-
Data	-	-	-	-	0.000	-
Nortel Discovery Protocol	-	-	-	-	0.000	-
IPv4	0.03	0.025	0.063	0.517	0.000	0.105
Data	-	-	-	-	0.000	-
Total	0.09	0.038	0.234	0.89	0.059	0.17

From this summary we conclude that for traces of 21 mins (Trace 3 and 4) we have more values of Mbits/s than Traces for 7mins and 53 mins i.e Trace 1,2,5,6 resp.

A. Request And Response Analysis of HTTP Traffic

HTTP Packet Counter with Filter TCP

Wireshark can also present a tree-like view of HTTP activity .It identifies the types of request and response packets. Also the quantities of each type, data rates, and overall percentages of all request and response types .This feature is also helpful at identifying how a Web server is being used, and can even identify potentially malicious activity with unsupported or broken HTTP requests or responses. HTTP Request statistics identify all the HTTP request URLs for each HTTP server in the packet capture, including the number of frames, data rate, and request percentage. This is useful to identify popular requests for a specific server. [9].

TABLE III. HTTP STATISTICS FOR TRACE 1

HTTP/Load Distribution:

Topic / Item	Count	Rate (ms)	Percent	Burst rate	Burst start
HTTP Requests by Server	5	0.0056	100%	0.0100	49.090
HTTP Requests by Server Address	5	0.0056	100.00%	0.0100	49.090
173.194.36.78	3	0.0034	60.00%	0.0100	49.572
safebrowsing-cache.google.com	3	0.0034	100.00%	0.0100	49.572
74.125.236.33	1	0.0011	20.00%	0.0100	49.090
safebrowsing.clients.google.com	1	0.0011	100.00%	0.0100	49.090
173.194.36.64	1	0.0011	20.00%	0.0100	49.357
safebrowsing-cache.google.com	1	0.0011	100.00%	0.0100	49.357
HTTP Requests by HTTP Host	5	0.0056	100.00%	0.0100	49.090
safebrowsing-cache.google.com	4	0.0045	80.00%	0.0100	49.357
173.194.36.78	3	0.0034	75.00%	0.0100	49.572
173.194.36.64	1	0.0011	25.00%	0.0100	49.357
safebrowsing.clients.google.com	1	0.0011	20.00%	0.0100	49.090
74.125.236.33	1	0.0011	100.00%	0.0100	49.090
HTTP Responses by Server Address	5	0.0056	100%	0.0100	49.333
173.194.36.78	3	0.0034	60.00%	0.0100	49.687
OK	3	0.0034	100.00%	0.0100	49.687
74.125.236.33	1	0.0011	20.00%	0.0100	49.333
OK	1	0.0011	100.00%	0.0100	49.333
173.194.36.64	1	0.0011	20.00%	0.0100	49.539
OK	1	0.0011	100.00%	0.0100	49.539

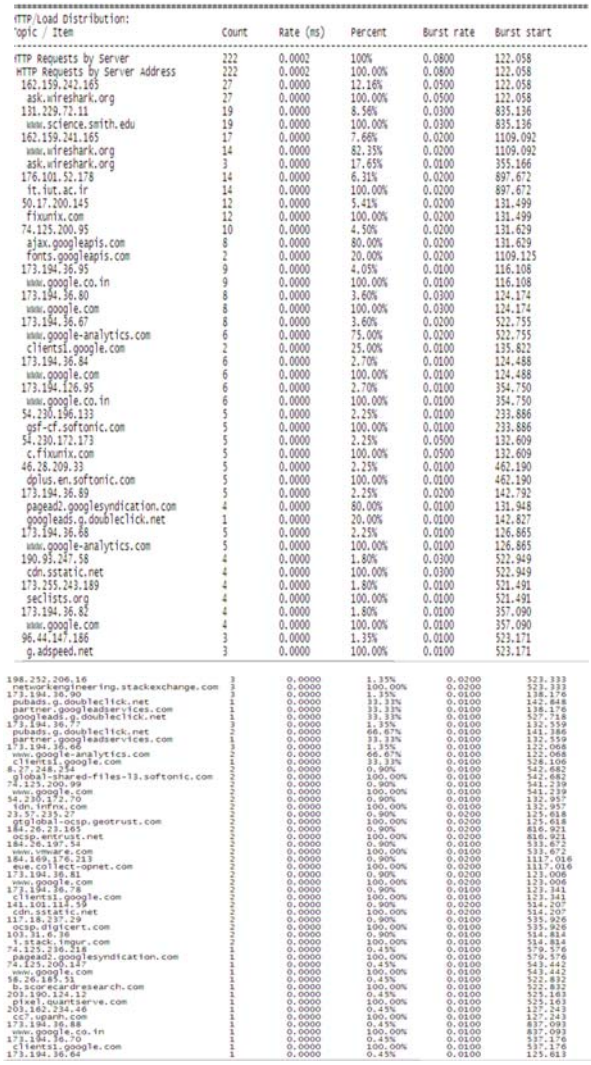
TABLE IV. HTTP STATISTICS FOR TRACE 2

HTTP/Load Distribution: Topic / Item	Count	Rate (ms)	Percent	Burst rate	Burst start
HTTP Requests by Server	33	0.0001	100%	0.0200	286.320
HTTP Requests by Server Address	33	0.0001	100.00%	0.0200	286.320
173.194.117.6	13	0.0000	39.39%	0.0200	286.320
s.ytimg.com	13	0.0000	100.00%	0.0200	286.320
65.55.11.179	5	0.0000	15.15%	0.0200	379.807
sa.windows.com	5	0.0000	100.00%	0.0200	379.807
49.200.255.209	3	0.0000	9.09%	0.0100	0.240
www.download.windowsupdate.com	3	0.0000	100.00%	0.0100	0.240
173.194.117.9	3	0.0000	9.09%	0.0100	269.183
www.youtube.com	2	0.0000	66.67%	0.0100	270.381
youtube.com	1	0.0000	33.33%	0.0100	269.183
65.55.206.229	2	0.0000	6.06%	0.0100	230.361
home.microsoft.com	2	0.0000	100.00%	0.0100	230.361
64.4.11.42	2	0.0000	6.06%	0.0100	227.924
www.microsoft.com	2	0.0000	100.00%	0.0100	227.924
207.46.61.29	2	0.0000	6.06%	0.0100	237.503
in.msn.com	2	0.0000	100.00%	0.0100	237.503
131.253.13.140	2	0.0000	6.06%	0.0100	233.584
www.msn.com	2	0.0000	100.00%	0.0100	233.584
74.125.200.94	1	0.0000	3.03%	0.0100	243.802
www.google.co.in	1	0.0000	100.00%	0.0100	243.802
HTTP Responses by Server Address	30	0.0001	100%	0.0100	0.642
173.194.117.6	12	0.0000	40.00%	0.0100	283.984
OK	12	0.0000	100.00%	0.0100	283.984
65.55.11.179	5	0.0000	16.67%	0.0100	380.446
OK	5	0.0000	100.00%	0.0100	380.446
49.200.255.209	3	0.0000	10.00%	0.0100	0.642
OK	3	0.0000	100.00%	0.0100	0.642
173.194.117.9	3	0.0000	10.00%	0.0100	269.684
OK	3	0.0000	100.00%	0.0100	269.684
65.55.206.229	2	0.0000	6.67%	0.0100	231.583
OK	2	0.0000	100.00%	0.0100	231.583
64.4.11.42	2	0.0000	6.67%	0.0100	228.963
OK	2	0.0000	100.00%	0.0100	228.963
131.253.13.140	2	0.0000	6.67%	0.0100	235.100
OK	2	0.0000	100.00%	0.0100	235.100
74.125.200.94	1	0.0000	3.33%	0.0100	247.184
OK	1	0.0000	100.00%	0.0100	247.184

TABLE V. HTTP STATISTICS FOR TRACE 3

HTTP/Load Distribution: Topic / Item	Count	Rate (ms)	Percent	Burst rate	Burst start
HTTP Requests by Server	259	0.0004	100%	0.1800	277.299
HTTP Requests by Server Address	259	0.0004	100.00%	0.1800	277.299
68.232.44.121	153	0.0003	59.07%	0.1800	277.299
1.gravatar.com	60	0.0001	39.22%	0.0600	277.229
0.gravatar.com	49	0.0001	32.03%	0.0700	280.468
2.gravatar.com	44	0.0001	28.76%	0.0800	277.300
68.232.44.111	27	0.0000	10.42%	0.0500	273.383
s0.wp.com	13	0.0000	48.15%	0.0300	273.383
s2.wp.com	11	0.0000	40.74%	0.0200	273.426
widgets.wp.com	1	0.0000	3.70%	0.0100	285.638
s1.wp.com	1	0.0000	3.70%	0.0100	282.537
i0.wp.com	1	0.0000	3.70%	0.0100	282.535
58.27.124.202	19	0.0000	7.34%	0.0100	530.424
download.windowsupdate.com	19	0.0000	100.00%	0.0100	530.424
58.26.185.57	10	0.0000	3.86%	0.0100	127.042
download.windowsupdate.com	10	0.0000	100.00%	0.0100	127.042
173.194.36.69	6	0.0000	2.32%	0.0100	305.209
safebrowsing-cache.google.com	6	0.0000	100.00%	0.0100	305.209
58.27.124.154	5	0.0000	1.93%	0.0100	524.939
ds.download.windowsupdate.com	5	0.0000	100.00%	0.0100	524.939
74.125.236.198	4	0.0000	1.54%	0.0100	264.853
clients1.google.com	4	0.0000	100.00%	0.0100	264.853
124.124.204.200	4	0.0000	1.54%	0.0100	30.621
crl.microsoft.com	4	0.0000	100.00%	0.0100	30.621
58.26.185.65	3	0.0000	1.16%	0.0100	130.122
ds.download.windowsupdate.com	3	0.0000	100.00%	0.0100	130.122
182.50.136.239	3	0.0000	1.16%	0.0100	286.445
ocsp.godaddy.com	3	0.0000	100.00%	0.0100	286.445
74.125.200.94	2	0.0000	0.77%	0.0100	270.726
www.google.co.in	2	0.0000	100.00%	0.0100	270.726
58.27.124.163	2	0.0000	0.77%	0.0100	524.748
download.windowsupdate.com	2	0.0000	100.00%	0.0100	524.748
54.246.174.85	2	0.0000	0.77%	0.0200	285.707
p.skimresources.com	2	0.0000	100.00%	0.0200	285.707
50.18.52.222	2	0.0000	0.77%	0.0100	286.960
t.skimresources.com	2	0.0000	100.00%	0.0100	286.960
23.47.235.27	2	0.0000	0.77%	0.0100	79.856
ocsp.thawte.com	1	0.0000	50.00%	0.0100	79.856
gtglobal-ocsp.geotrust.com	1	0.0000	50.00%	0.0100	265.295
23.41.75.27	2	0.0000	0.77%	0.0100	122.068
gtssl-ocsp.geotrust.com	2	0.0000	100.00%	0.0100	122.068
192.0.80.247	2	0.0000	0.77%	0.0200	285.049
stats.wordpress.com	2	0.0000	100.00%	0.0200	285.049
76.74.254.120	1	0.0000	0.39%	0.0100	271.487
srackham.wordpress.com	1	0.0000	100.00%	0.0100	271.487
74.125.236.199	1	0.0000	0.39%	0.0100	304.892
68.232.44.251	1	0.0000	0.39%	0.0100	282.643
s.stats.wordpress.com	1	0.0000	100.00%	0.0100	282.643
65.54.82.145	1	0.0000	0.39%	0.0100	30.833
nscrl.microsoft.com	1	0.0000	100.00%	0.0100	30.833
65.54.51.252	1	0.0000	0.39%	0.0100	133.695
update.microsoft.com	1	0.0000	100.00%	0.0100	133.695
58.26.185.66	1	0.0000	0.39%	0.0100	176.460
ds.download.windowsupdate.com	1	0.0000	100.00%	0.0100	176.460
58.26.185.42	1	0.0000	0.39%	0.0100	31.159
ctldl.windowsupdate.com	1	0.0000	100.00%	0.0100	31.159
58.26.185.35	1	0.0000	0.39%	0.0100	150.181
download.windowsupdate.com	1	0.0000	100.00%	0.0100	150.181
23.58.43.27	1	0.0000	0.39%	0.0100	481.506
ocsp.thawte.com	1	0.0000	100.00%	0.0100	481.506
199.27.77.192	1	0.0000	0.39%	0.0100	282.650
s.skimresources.com	1	0.0000	100.00%	0.0100	282.650
184.72.54.69	1	0.0000	0.39%	0.0100	287.098
r.skimresources.com	1	0.0000	100.00%	0.0100	287.098
HTTP Responses by Server Address	236	0.0004	100%	0.1000	282.732

TABLE VI. HTTP STATISTICSVFOR TRACE 4

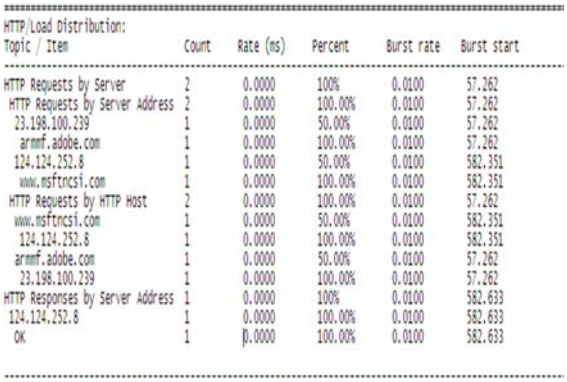


Topic / Item	Count	Rate (ms)	Percent	Burst rate	Burst start
HTTP Requests by Server	222	0.0002	100%	0.0800	122.058
HTTP Requests by Server Address	222	0.0002	100.00%	0.0800	122.058
162.159.242.165	27	0.0000	12.16%	0.0500	122.058
ask.wireshark.org	27	0.0000	100.00%	0.0500	122.058
131.229.72.11	19	0.0000	8.56%	0.0300	835.136
www.science.smith.edu	19	0.0000	100.00%	0.0300	835.136
162.159.241.165	17	0.0000	7.66%	0.0200	1109.092
www.wireshark.org	14	0.0000	82.35%	0.0200	1109.092
ask.wireshark.org	3	0.0000	17.65%	0.0100	355.166
176.101.52.178	14	0.0000	6.31%	0.0200	897.672
it.iut.ac.ir	14	0.0000	100.00%	0.0200	897.672
50.17.200.145	12	0.0000	5.41%	0.0200	131.499
fixunix.com	12	0.0000	100.00%	0.0200	131.499
74.125.200.95	10	0.0000	4.50%	0.0200	131.629
ajax.googleapis.com	8	0.0000	80.00%	0.0200	131.629
fonts.googleapis.com	2	0.0000	20.00%	0.0200	1109.125
173.194.36.95	9	0.0000	4.05%	0.0100	116.108
www.google.co.in	9	0.0000	100.00%	0.0100	116.108
173.194.36.80	8	0.0000	3.60%	0.0300	124.174
www.google.com	8	0.0000	100.00%	0.0300	124.174
173.194.36.67	8	0.0000	3.60%	0.0200	522.755
www.google-analytics.com	6	0.0000	75.00%	0.0200	522.755
clients1.google.com	2	0.0000	25.00%	0.0100	135.822
173.194.36.84	6	0.0000	2.70%	0.0100	124.488
www.google.com	6	0.0000	100.00%	0.0100	124.488
173.194.126.95	6	0.0000	2.70%	0.0100	354.750
www.google.co.in	6	0.0000	100.00%	0.0100	354.750
54.230.196.133	5	0.0000	2.25%	0.0100	233.886
gsf-cf.softonic.com	5	0.0000	100.00%	0.0100	233.886
54.230.172.173	5	0.0000	2.25%	0.0500	132.609
c.fixunix.com	5	0.0000	100.00%	0.0500	132.609
46.28.209.33	5	0.0000	2.25%	0.0100	462.190
dplus.en.softonic.com	5	0.0000	100.00%	0.0100	462.190
173.194.36.89	5	0.0000	2.25%	0.0200	142.792
pagead2.googlesyndication.com	4	0.0000	80.00%	0.0100	131.948
googleads.g.doubleclick.net	1	0.0000	20.00%	0.0100	142.827
173.194.36.68	5	0.0000	2.25%	0.0100	126.865
www.google-analytics.com	5	0.0000	100.00%	0.0100	126.865
190.93.247.58	4	0.0000	1.80%	0.0300	522.949
cdn.sstatic.net	4	0.0000	100.00%	0.0300	522.949
173.255.243.189	4	0.0000	1.80%	0.0100	521.491
seclists.org	4	0.0000	100.00%	0.0100	521.491
173.194.36.82	4	0.0000	1.80%	0.0100	357.090
www.google.com	4	0.0000	100.00%	0.0100	357.090
96.44.147.186	3	0.0000	1.35%	0.0100	523.171
g.adspeed.net	3	0.0000	100.00%	0.0100	523.171
198.252.206.16	3	0.0000	1.35%	0.0200	523.333
networkengineering.stackexchange.com	3	0.0000	100.00%	0.0200	523.333
173.194.36.90	3	0.0000	1.35%	0.0100	138.176
pubads.g.doubleclick.net	1	0.0000	33.33%	0.0100	142.848
partner.googleadservices.com	1	0.0000	33.33%	0.0100	138.176
googleads.g.doubleclick.net	1	0.0000	33.33%	0.0100	527.718
173.194.36.77	3	0.0000	1.35%	0.0100	132.559
pubads.g.doubleclick.net	2	0.0000	66.67%	0.0100	141.386
partner.googleadservices.com	1	0.0000	33.33%	0.0100	132.559
173.194.36.66	3	0.0000	1.35%	0.0100	122.068
www.google-analytics.com	2	0.0000	66.67%	0.0100	122.068
clients1.google.com	1	0.0000	33.33%	0.0100	528.106
8.27.248.254	2	0.0000	0.90%	0.0100	542.682
global-shared-files-13.softonic.com	2	0.0000	100.00%	0.0100	542.682
74.125.200.99	2	0.0000	0.90%	0.0100	541.239
www.google.com	2	0.0000	100.00%	0.0100	541.239
54.230.172.70	2	0.0000	0.90%	0.0100	132.957
cdn.imfnx.com	2	0.0000	100.00%	0.0100	132.957
23.57.235.27	2	0.0000	0.90%	0.0200	125.618
gtglobal-ocsp.geotrust.com	2	0.0000	100.00%	0.0200	125.618
184.26.23.165	2	0.0000	0.90%	0.0200	816.921
ocsp.entrust.net	2	0.0000	100.00%	0.0200	816.921
184.26.197.54	2	0.0000	0.90%	0.0100	533.672
www.vmware.com	2	0.0000	100.00%	0.0100	533.672
184.169.176.213	2	0.0000	0.90%	0.0200	1117.016
eue.collect-opnet.com	2	0.0000	100.00%	0.0200	1117.016
173.194.36.81	2	0.0000	0.90%	0.0200	123.006
www.google.com	2	0.0000	100.00%	0.0200	123.006
171.194.36.78	2	0.0000	0.90%	0.0100	123.341
clients1.google.com	2	0.0000	100.00%	0.0100	123.341
141.101.114.59	2	0.0000	0.90%	0.0200	514.207
cdn.sstatic.net	2	0.0000	100.00%	0.0200	514.207
117.18.237.29	2	0.0000	0.90%	0.0100	535.926
ocsp.digicert.com	2	0.0000	100.00%	0.0100	535.926
103.31.6.36	2	0.0000	0.90%	0.0100	514.814
i.stack.imgur.com	2	0.0000	100.00%	0.0100	514.814
74.125.236.218	1	0.0000	0.45%	0.0100	579.576
pagead2.googlesyndication.com	1	0.0000	100.00%	0.0100	579.576
74.125.200.147	1	0.0000	0.45%	0.0100	543.442
www.google.com	1	0.0000	100.00%	0.0100	543.442
58.26.185.51	1	0.0000	0.45%	0.0100	522.832
b.scorecardresearch.com	1	0.0000	100.00%	0.0100	522.832
203.190.124.12	1	0.0000	0.45%	0.0100	525.163
pixel.quantserve.com	1	0.0000	100.00%	0.0100	525.163
203.162.234.46	1	0.0000	0.45%	0.0100	127.243
cc7.uparh.com	1	0.0000	100.00%	0.0100	127.243
173.194.36.88	1	0.0000	0.45%	0.0100	837.093
www.google.co.in	1	0.0000	100.00%	0.0100	837.093
173.194.36.70	1	0.0000	0.45%	0.0100	537.176
clients1.google.com	1	0.0000	100.00%	0.0100	537.176
173.194.36.64	1	0.0000	0.45%	0.0100	125.613

TABLE VII. HTTP STATISTICS FOR TRACE 5

Topic / Item	Count	Rate (ms)	Percent	Burst rate	Burst start
HTTP Requests by Server	90	0.0000	100%	0.0200	1689.467
HTTP Requests by Server Address	90	0.0000	100.00%	0.0200	1689.467
124.124.252.25	32	0.0000	35.56%	0.0100	526.785
download.windowsupdate.com	32	0.0000	100.00%	0.0100	526.785
124.124.252.9	14	0.0000	15.56%	0.0100	111.753
download.windowsupdate.com	14	0.0000	100.00%	0.0100	111.753
173.194.36.65	10	0.0000	11.11%	0.0100	1361.975
safebrowsing-cache.google.com	10	0.0000	100.00%	0.0100	1361.975
173.194.36.72	5	0.0000	5.56%	0.0200	3226.500
safebrowsing-cache.google.com	3	0.0000	60.00%	0.0200	3226.500
clients1.google.com	2	0.0000	40.00%	0.0100	1220.715
199.7.55.72	4	0.0000	4.44%	0.0200	1689.467
ocsp.verisign.com	4	0.0000	100.00%	0.0200	1689.467
124.124.252.99	4	0.0000	4.44%	0.0100	526.998
ds.download.windowsupdate.com	4	0.0000	100.00%	0.0100	526.998
58.27.124.219	3	0.0000	3.33%	0.0100	111.297
ds.download.windowsupdate.com	3	0.0000	100.00%	0.0100	111.297
23.47.235.27	3	0.0000	3.33%	0.0100	1210.232
gtssl-ocsp.geotrust.com	2	0.0000	66.67%	0.0100	1210.232
gtglobal-ocsp.geotrust.com	1	0.0000	33.33%	0.0100	1221.122
117.18.237.29	3	0.0000	3.33%	0.0200	1809.782
ocsp.digicert.com	3	0.0000	100.00%	0.0200	1809.782
23.51.43.27	2	0.0000	2.22%	0.0100	1569.364
ocsp.thawte.com	2	0.0000	100.00%	0.0100	1569.364
fe80::517a:7a24:b76a:d546	1	0.0000	1.11%	0.0100	2249.402
[fe80::517a:7a24:b76a:d546]:2869	1	0.0000	100.00%	0.0100	2249.402
65.55.58.195	1	0.0000	1.11%	0.0100	317.541
go.microsoft.com	1	0.0000	100.00%	0.0100	317.541
65.55.192.94	1	0.0000	1.11%	0.0100	111.888
update.microsoft.com	1	0.0000	100.00%	0.0100	111.888
65.54.62.138	1	0.0000	1.11%	0.0100	319.222
az307127.vo.msecnd.net	1	0.0000	100.00%	0.0100	319.222
65.52.33.27	1	0.0000	1.11%	0.0100	318.553
www.legallery.com	1	0.0000	100.00%	0.0100	318.553
58.27.124.202	1	0.0000	1.11%	0.0100	108.835
download.windowsupdate.com	1	0.0000	100.00%	0.0100	108.835
23.41.75.27	1	0.0000	1.11%	0.0100	76.286
ocsp.thawte.com	1	0.0000	100.00%	0.0100	76.286
199.7.51.72	1	0.0000	1.11%	0.0100	99.816
ocsp.verisign.com	1	0.0000	100.00%	0.0100	99.816
173.194.36.71	1	0.0000	1.11%	0.0100	1361.571
safebrowsing.clients.google.com	1	0.0000	100.00%	0.0100	1361.571
173.194.36.66	1	0.0000	1.11%	0.0100	3226.167
safebrowsing.clients.google.com	1	0.0000	100.00%	0.0100	3226.167
HTTP Responses by Server Address	76	0.0000	100%	0.0200	1690.186
124.124.252.25	29	0.0000	38.16%	0.0100	548.948

TABLE VIII. HTTP STATISTICS FOR TRACE 6



HTTP/Load Distribution:

Topic / Item	Count	Rate (ms)	Percent	Burst rate	Burst start
HTTP Requests by Server	2	0.0000	100%	0.0100	57.262
HTTP Requests by Server Address	2	0.0000	100.00%	0.0100	57.262
23.198.100.239	1	0.0000	50.00%	0.0100	57.262
armmf.adobe.com	1	0.0000	100.00%	0.0100	57.262
124.124.252.8	1	0.0000	50.00%	0.0100	582.351
www.msftncsi.com	1	0.0000	100.00%	0.0100	582.351
HTTP Requests by HTTP Host	2	0.0000	100.00%	0.0100	57.262
www.msftncsi.com	1	0.0000	50.00%	0.0100	582.351
124.124.252.8	1	0.0000	100.00%	0.0100	582.351
armmf.adobe.com	1	0.0000	50.00%	0.0100	57.262
23.198.100.239	1	0.0000	100.00%	0.0100	57.262
HTTP Responses by Server Address	1	0.0000	100%	0.0100	582.633
124.124.252.8	1	0.0000	100.00%	0.0100	582.633
OK	1	0.0000	100.00%	0.0100	582.633

From above analysis we conclude that Trace 3 and Trace 4 contains more amount of packets captured as compared to other traces. Which depicts that at peek load we have more amount of communication between sender and receiver or between two nodes.

B. Expert Analysis Summary

TABLE IX. EXPERT INFO. FOR TRACES CAPTURED

	Errors	Count	Warnings	Count	Notes	Count
Trace 1	Bad checksum	1(41)	Duplicate IP addr. Ack no. broken TCP	5(47)	Malformed BOOTP/DHCP	5(65)
Trace 2	Malformed Packet	1(1)	Ack segment not captured	1(3)	Retransmission Duplicate Ack Keep Alive	4(64)
Trace 3	Bad Checksum Malformed Pkt	4(2987)	Duplicate IP addr Ack no. broken TCP Out of order segment	10(57)	Malformed BOOTP/DHCP Duplicate ACK Fast Retransmission	44(417)
Trace 4	Bad checksum Retransmission	2(19854)	Duplicate IP addr Previous segment not captured Ack no. broken TCP Out of order segment	16(2574)	Malformed BOOTP/DHCP Duplicate Ack Retransmission Fast Retransmission	61(11405)
Trace 5	Bad Checksum Malformed Packet	2(2263)	Duplicate IP addr Previous segment not captured Ack no. broken TCP	18(146)	Malformed BOOTP/DHCP Duplicate Ack Retransmission	47(805)
Trace 6	Bad Checksum Malformed Packet	3(767)	Duplicate IP addr Previous segment not captured Ack no. brokenTCP	12(1093)	Malformed BOOTP/DHCP Duplicate Ack Retransmission	10(668)

This Expert info table summarizes various errors coming during capturing as Bad Checksum, Malfomed Packets , all the warnings that comes on the way of network as Duplicate IP addresses, Previous segment not captured. Acknowledgement no. broken TCP, Out of order segment and also various notes which give us information about malformed packets, Duplicate acknowledgments and retransmissions. If we have to filter out abnormal traffic we use expert info.

VI. RESULTS

Wireshark offers numerous graphs to depict traffic flow trends. Some graphs are directional, focusing on traffic flowing in a specific direction. In our work , we have analyzed the traffic and obtained the following graphs.

- Time Sequence Graph- The time-sequence graph shows the TCP sequence numbers vs. time. It conveys a lot more information about the TCP stream.
- Round Trip Time Graph- The RTT graph shows the RTT vs. the sequence number.
- Throughput Graph - The throughput graph shows the throughput of the TCP stream vs. time

A. Analyzing graphs

On per packet basis we can visualize packet rate on different intervals In Time sequence graph**,** discontinuity in the graph leads to packet loss , throughput fell off dramatically during retransmission. Also these graphs have even slope after every 0.3 sec for approximately 3 seconds. When there is a major disruption, the gap in the graphs suggests TCP retransmission .Round Trip Time graph is meant for establishing the connection. When a packet exceeds RTT value, packet is considered to be lost and thus it is retransmitted in a TCP connection. TCP Throughput graphs are created based on the packet which is selected in the Packet List pane. Graphs can be easily created for any conversation in the trace file.
We have obtained graphs for peak load traces.

B. Case 1. Trace 3

For graph analysis we have to look at the Flow graph of the trace and from there we plot RTT for each TCP segment sent .Also from the trace we can calculate Throughput of it.

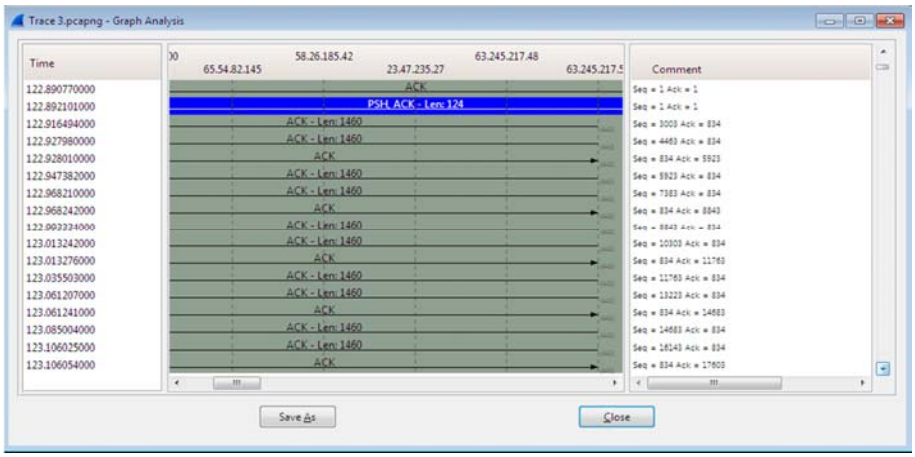


Figure 4. Flow Graph

From this Flow graph RTT is calculated for each of the first six segments shown in the table below

TABLE X. RTT CALCULATION FOR TRACE 3

Segment	Relative segment no.	Time sent	Acknowledgement received	RTT
1	1	122.892101000	122.927980000	0.035879
2	834	122.928010000	122.947382000	0.019372
3	5923	122.947382000	122.968242000	0.02086
4	8843	122.993334000	123.013276000	0.019942
5	11763	123.035503000	123.061207000	0.025704
6	14683	123.085004000	123.106025000	0.021021

RTT is calculated as , RTT = Acknowledge received – Time sent

Generally the TCP segment will have standard maximum length of 1500 bytes (40 bytes TCP/IP header data and 1460 bytes of TCP payload).This trace shows TCP length greater than 1500 bytes then wireshark is reporting the wrong TCP segment length .It shows one large TCP segment than multiple smaller segments .This inconsistency is due to interaction between Ethernet driver and wireshark software .My results shows too long TCP segments.

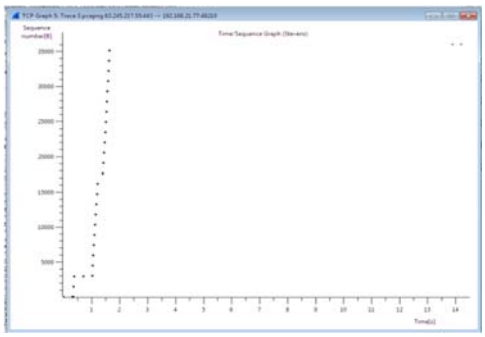


Figure 5. Time Sequence Graph

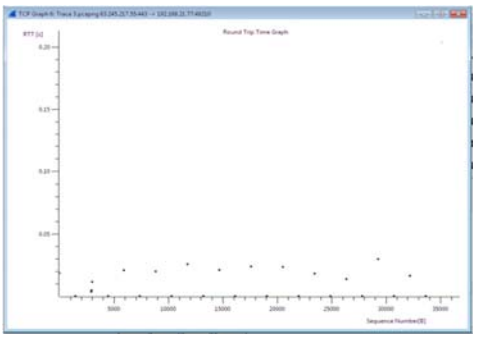


Figure 6. Round Trip Time Graph

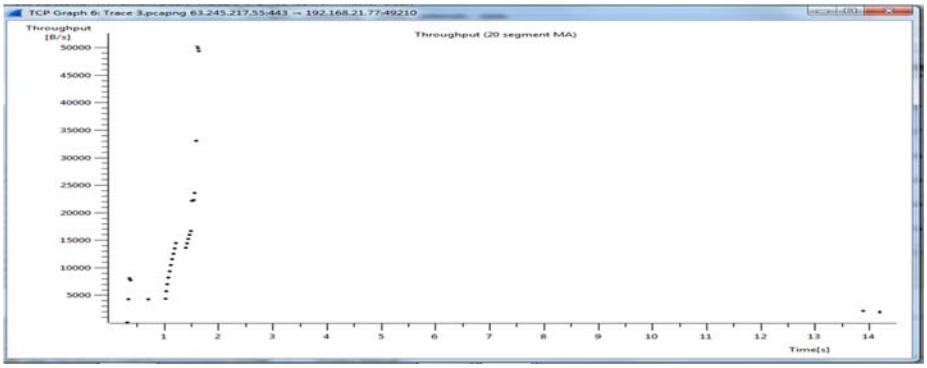


Figure 7. Throughput graph

C. Case2 : Trace 4

Flow Graph

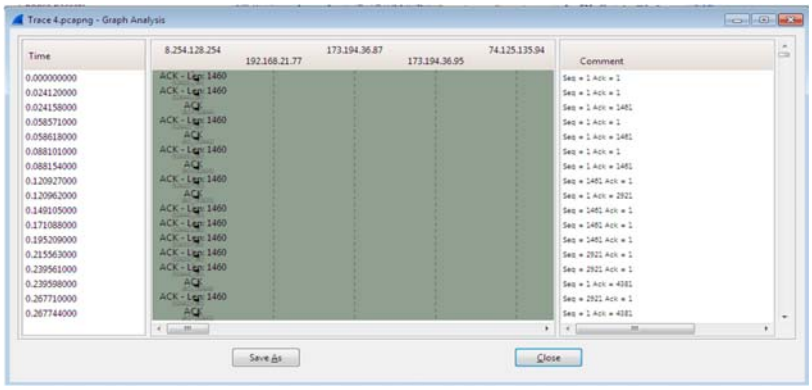


Figure 8. Flow Graph For Trace 4

TABLE XI. RTT CALCULATION FOR TRACE 4

Segment	Relative segment number	Time sent	Acknowledgement received	RTT
1	1	0.024120000	0.088154000	0.064034
2	1461	0.120927000	0.195209000	0.074282
3	2921	0.215563000	0.294916000	0.079353
4	4381	0.318974000	0.408527000	0.089553
5	5841	0.446301000	0.493136000	0.046835
6	7301	0.522220000	0.597098000	0.074878

From this RTT calculation we see that the ACK numbers increase in the sequence 1461,2921,4381,5841….ACK number increases by 1460 each time ,indicates that the receiver is acknowledging 1460 bytes.
By this throughput can also be calculated as

Throughput = Bytes Acknowledge / Time in secs.

As I looked to FINACK packet which shows a acknowledgement no. of 452,meaning that 452 bytes were acknowledged .The time on this message is 118.501677000.So approximate average throughput can be calculated as 452/118.501677000 ≈ 3.814 bytes/sec .Screen shot is as below.

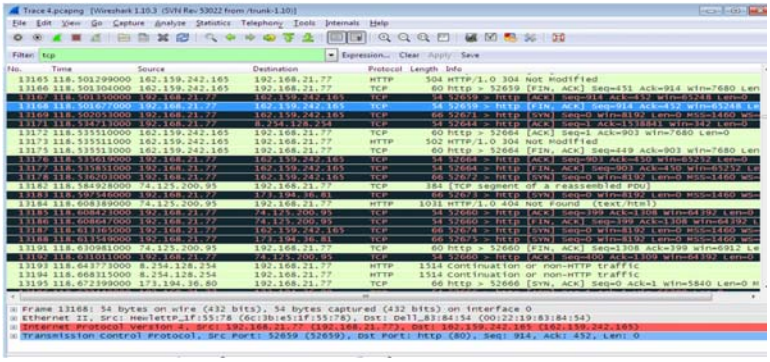


Figure 9. Screenshot of wireshark screen of Trace 4

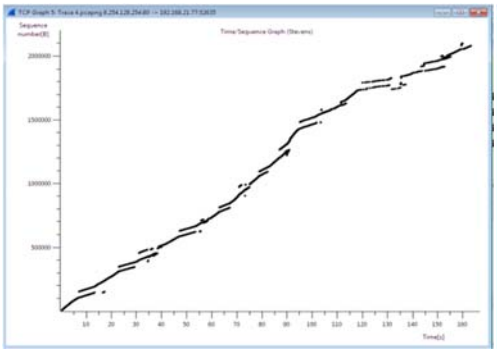


Figure 10. Time Sequence Graph

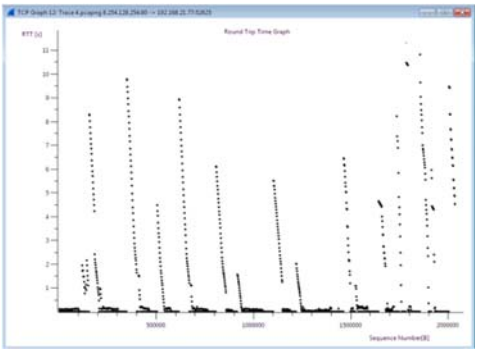


Figure 11. Round Trip Time Graph

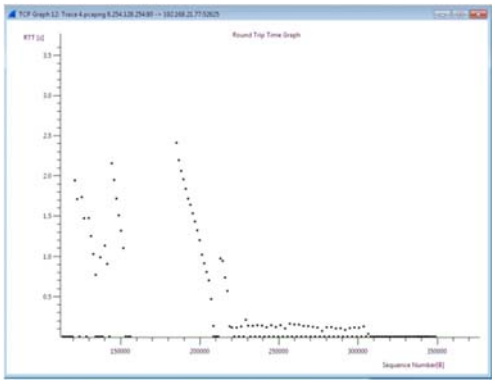


Figure 12. RTT graph (Zoom)

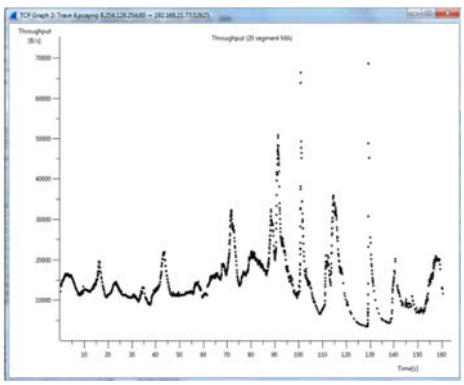


Figure 13. Throughput Graph

- Note that a set of dots stacked above each other represents a series of packets that were sent back-to-back by the sender.

(I) Anomalies

DHCP SPOOF

A DHCP attack consists of falsifying DHCP packets.In this, attacker install a false DHCP such that it responds to DHCP DISCOVER client request. When a computer is connected to a network and requests an IP address,it sends DHCP DISCOVER to broadcast address and waits for the response of a DHCP server.

The server then replies to this request by sending DHCP OFFER.The client can receive offers from various DHCP as if offer is corresponding to a previously assigned address the client selects this and if proposal is not related to the previous address,the client acquires the first offer received.Then in response DHCP REQUEST is sent for authorization with DHCPACK or with DHCPNAK.

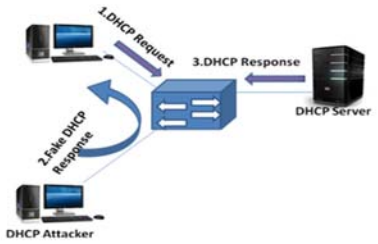


Figure 14. DHCP Spoofing

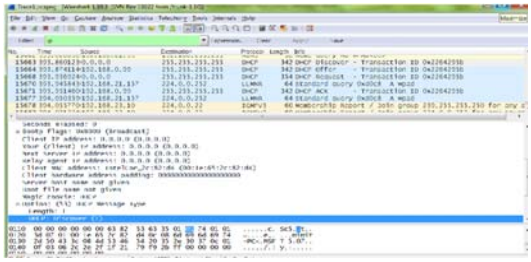


Figure 15. DHCP protocol session from packet no 15663-15671(Negotiating DHCP)

To provide warning of these situations we can use filters in Wireshark to fastly search for ACK responses with a DNS different from the one configured on DHCP server: bootp.option.value == 05 && (frame[309:6] != 03:04:c0:a8:fe:fe || frame[315:6] == 06:04:c0:a8:fe:d3)

In this way we can configure it to display the segments sent by DHCP server that do not contain the IP gateway.

One more type of attack consists of sending multiple DHCP DISCOVER packets with the objective of finishing-up the range of IP available in the DHCP server.

To get out of this type of problems many tools are available for free.

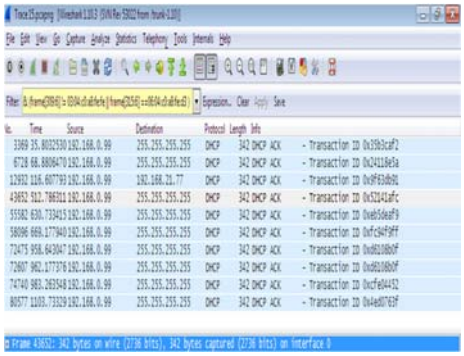


Figure 16. DHCP Filter

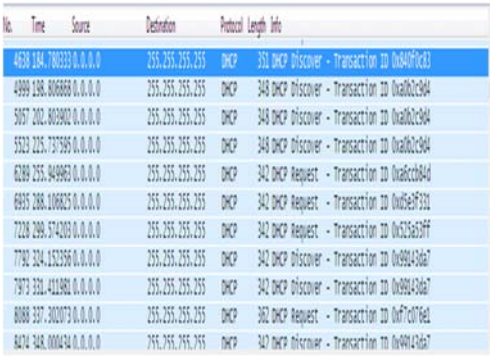


Figure 17. DHCP Exhaustion

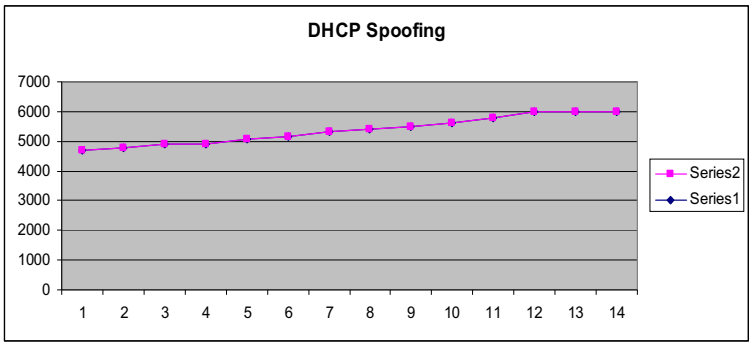


Figure 18. Graph for DHCP Spoofing

DDOS Attack

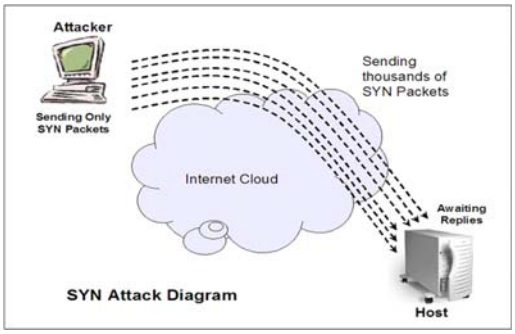


Figure 19 . DDOS SYN Attack

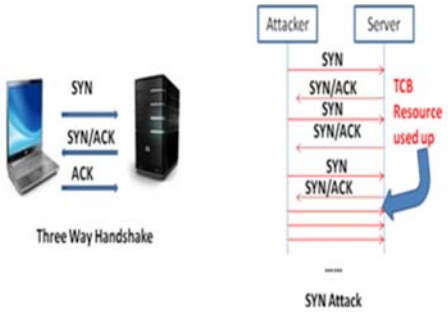


Figure 20. Showing Three way handshake process and SYN Attack

Figure 21. is an example of DDOS attacks on a small scale, that stands out as soon as the capture process starts. In this process a large number of TCP segments with the SYN flag activated from the same IP that do not receive a response from the web service. You can see the packet sequence graphically by selecting from the menu *Statistics, >>Flow Graph*. By this we can track the behaviour of TCP connections, arrows shows the source and target of each packet. There are a number of attempts at one address, this an unusual situation.

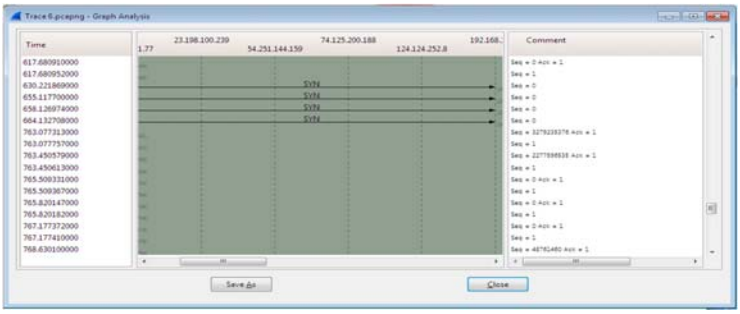


Figure 21. Flow Graph

When no response is received ,it cannot send an ACK-SYN to the same to continue with the three step connection.TCP/IP stack has to wait for a set of time for each connection. More packets keep arriving that create new connections and to identify these connection Transmission Control Block is created so that machine stops responding to more connection requests.

ARP SPOOF

ARP SPOOF is used by attacker to get in between one or more machine to intercept or capture packets.

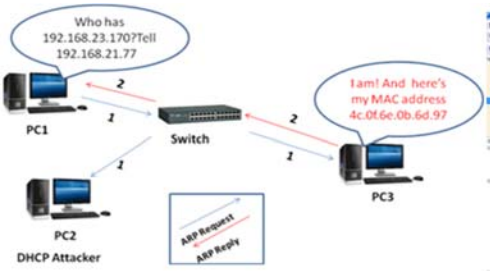


Figure 22 . ARP Request /Reply

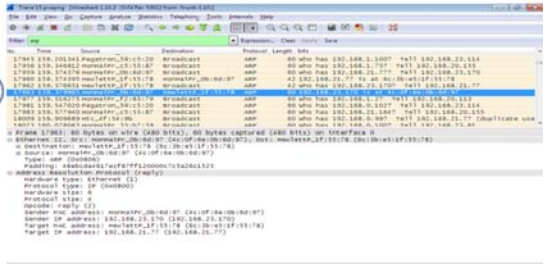


Figure 23. Wireshark Areas Of ARP Packets

where you can quickly see that something suspect is occurring due to the large quantity of ARP traffic that is being received. If you take a more detailed look at the behaviour of the protocol, you will realize that the server is being attacked. In packet number 17963, you can see how the machine with IP 192.168.21.77, and a Message Authentication Code (MAC) HonHaiPr_0b:6d:97, has launched an ARP request to the broadcast address asking for the MAC of the IP 192.168.23.170 Immediately afterwards, the router responds with an

ARP reply indicating the MAC address. Then the same IP repeats the process and requests the MAC of the IP using another broadcast diffusion.[7] The server responds with its MAC address. Everything is going normal till. Problem occurs when machine repeatedly sends to server false ARP packets both with its own MAC.This way traffic transmitted between local network and server goes through the attacking machine
The raw data format of an ARP reply generated by your machine to an ARP request is then shown. You can look for these packets with the following filters arp.opcode = = 0x0002 (ARP *reply*):[7]

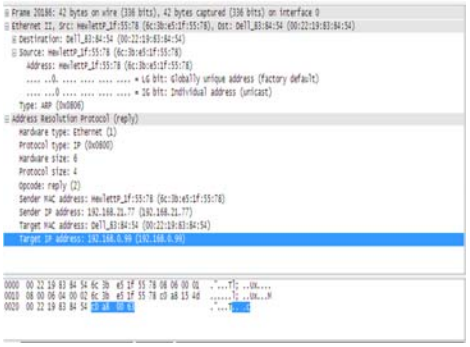


Figure 24. ARP Spoof raw data format

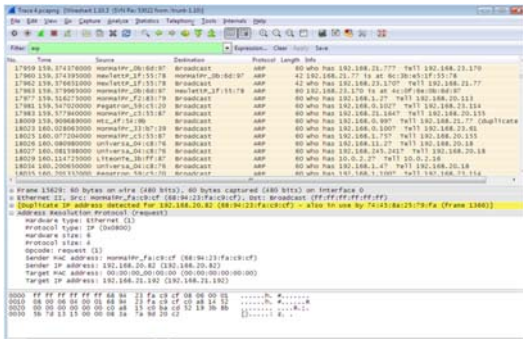


Figure 25. Arp capturing Duplicate IP address which is first used in frame no.1366

The hexadecimal text shown in the lower portion corresponds to the segment transmitted by the network. Therefore, anyone can take those values.He can modify them and resend them. To do this, right-click "*Frame 20186*" and select *"Export Selected Packet Bytes"* and save the segment in a file. At a later stage you can modify the segment creating an ARP reply with any kind of Hexadecimal Editor. If there is any other device using the same IP which is already in use by another, it sends ARP Reply with it's MAC address. Thus the Windows comes to know that the same IP address is being used again as in Figure 25.
There might be another situation when number of packets are coming from same IP address continuously as shown in Figure 26. And this is for attacking purpose.Graph is shown in Figure 27.

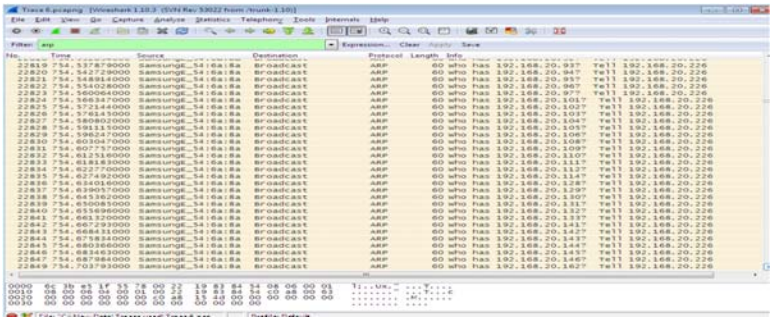


Figure 26. ARP spoofing window

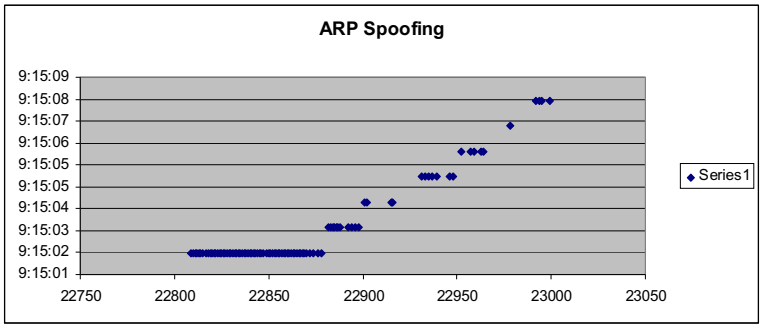


Figure 27. Graph of ARP Spoofing

HTTP Spidering - In HTTP a client sends a request message to the server and then in return a response message back to client.When sending malicious requests to the application, the web client will send a request for a specific resource. In this case is 192.168.21.77.The GET method is used to request a web page and it passes any parameters in the URL field .Some applications just requests many web pages in a short period of time. There's over 13 different requests made under 1 sec from the same address shown in Figure 28. And graph is shown in Figure 29.

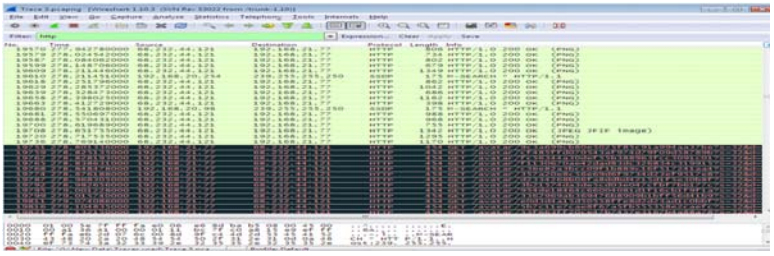


Figure 28. HTTP Spidering

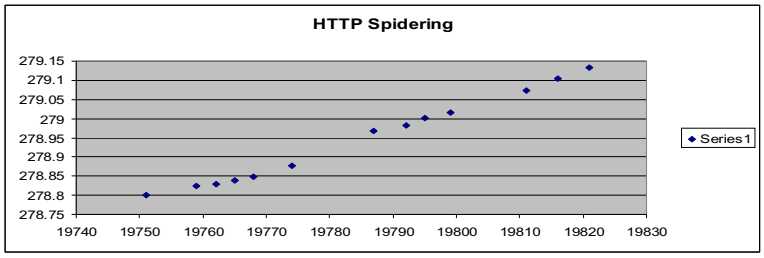


Figure 29. Graph of HTTP SpIdering

VIII Conclusions

In our work we analyzed and captured the data which is done with a tool named Wireshark which is the best packet analyzer. All the options in this tool were studied and experimented by obtaining traces from the conversations among nodes from specific IP addresses in an institute. The traces thus obtained from the traffic analysis were analysed as protocol usage in all traces for Low ,Medium and Peak loads and HTTP Statistics i.e Request and response from one address to another. Expert analysis is also taken which shows errors ,warnings , notes of all the information coming under capturing. These are then graphed into Time sequence graph, Round Trip Time graph and Throughput graph. The tool also takes into account the possible attacks such as DHCP SPOOFING, DDOS Attack, ARP spoofing, HTTP Spidering.

Future Work

There are some bandwidth limitations on wireshark which lead to performance degradation while traffic analysis is carried by it. Moreover the processing load at the monitoring device is very high because during traffic analysis it captures the irrelevant data also which is of no use and thus increasing the load on the device. So there should be some special filters installed at the monitoring device to capture the data not more than the data which is actually needed for the analysis. So we suggest more research should be done by considering these parameters also.

Acknowledgement

I am grateful to God Almighty with whose blessings this study could be successfully completed. Words won't suffice to express my extreme indebtedness and deep sense of gratitude for my respected teacher Dr. Monika Sachdeva, Dept of Computer Science, SBSSTC , Ferozepur, for her constant inspiration, consistent encouragement, personal interest, and invaluable guidance of this study. I also find myself short of words to express my gratefulness to Dr.Krishan Saluja, who rendered me help, moral support, cooperation and encouragement in shaping up this paper.

REFERENCES

[1] Shilpi Gupta,et.al "*Intrusion Detection System Using Wireshark",* Software engineering, ITM University Gurgaon, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 11, November 2012 ISSN: 2277 128X

[2] Aamir Hassan "*Network Security Analysis*" , School of Information Science, Computer and Electrical Engineering Halmstad University, Technical report, IDE 1004, February 2010

[3] Usha Banerjee, et.al , "*Evaluation of the Capabilities of WireShark as a tool for Intrusion Detection"* , Department of Computer Science & Engineering College of Engineering Roorkee, International Applications (0975 – 8887) Volume 6– No.7, September Journal of Computer 2010

[4] Joshua L. Davis "*Using Wireshark to Create Network-Usage Baselines"* ,Georgia Tech Research Institute Georgia Institute of Technology Atlanta, GA 30332, Copyright © 2007 Georgia Tech Research Corporation, June 2007

[5] Ulf Lamping, Richard Sharpe, NS Computer S/W And Services P/L, Ed Warnicke" *Wireshark User's Guide",* Copyright © 2004-2014 Ulf Lamping, Richard Sharpe, Ed Warnicke

[6] Mohsin Khan, et.al, "*Investigation of DHCP Packets using Wireshark"*, Volume 63- Number 4, Published by Foundation of Computer Science, New York, USA , *International Journal of Computer Applications* 63(4):1-9, February 2013.

[7] Inteco-Cert ,"Traffic Analysis With Wireshark", Borja Merino Febero, February 2011

[8] Sanders,Chris (May 23,2007),"*Practical Packet Analysis Using Wireshark to solve Real World Network Problems"*, No starch Press p.192 ISBN 1-59327-149-2

[9] Orebaugh ,Angela ; Ramirez ,Gilbert ; Beale , Jay(February 14,2007) " Wireshark & Ethereal Network Protocol Analyzer Toolkit" by Angela Orebaugh ,Gilbert Ramirez ,Josh Burke, by Syngress Publishing.

[10] Justin Jay Lister *,"Intrusion Detection Systems: An introduction to the detection and prevention of computer abuse",* Computer Security Research, Department of Computer Science, University of Wollongong.

[11] Emilie Lundin, et.al, "*Survey of Intrusion Detection Research",* Department of Computer Engineering Chalmers University of Technology, Technical Report nr. 02-04.