STUDY OF NETWORK TRAFFIC BEHAVIOR AND DETECTION OF ATTACKS IN WIRESHARK

AMANPREET KAUR
Student Of M.Tech , SBSSTC/ Computer Engg. , Ferozepur ,India
e-mail - aman2s2@yahoo.co.in

MONIKA SACHDEVA
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 is 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 this paper we use wireshark as a packet analyzer which observed the communicating nodes and gathered data from them on an institute network . Wireshark is an open source packet analyzer , which was formerly known as Ethereal. Protocol usage distribution is built which shows low , medium and peak loads of traffic . HTTP Statistics are built for request and response analysis and Expert analysis is done to detect warnings and malformed packets . The outputs are shown in graphs namely time Sequence graph, round trip time graph , throughput graph and flow graph . Certain attacks are observed namely DHCP Spoofing , DDOS attack, ARP Spoofing , HTTP Spidering and they are shown through graphs as well. The graphs obtained here using wireshark help to interpret the efficiency and performance of the network of an institute taken.

KEYWORDS: Intrusion Detection System, Network Security, Wireshark.

INTRODUCTION TO NETWORK SECURITY AND INTRUSION DETECTION SYSTEM

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.

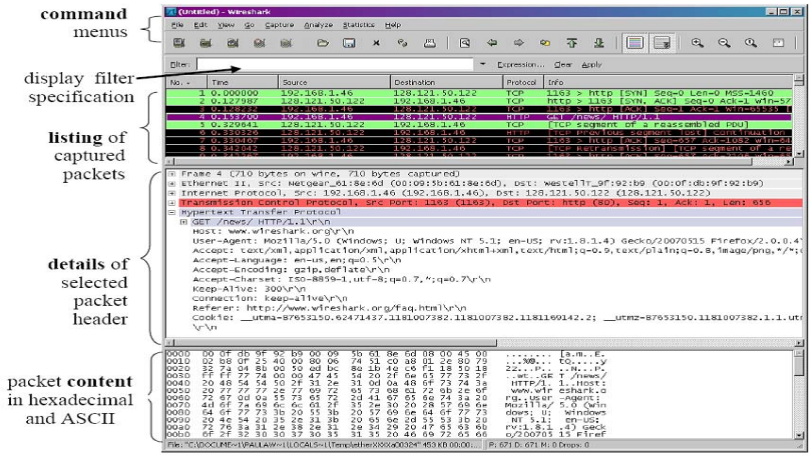


Figure1. Wireshark window

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 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 ,Table I shows the values of various parameters that we observed.

Table 1. 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

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, 2012 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 in 2010 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, 2010 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, 2007 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, et.al, **2013** 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, 1995 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. Lundin, E. ; Jonsson, E. ,**2002** 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.

METHODOLOGY AND EXPERIMENTAL SETUP

For real-time packet capturing,we use following methodology for packet capturing as shown in Figure 2.

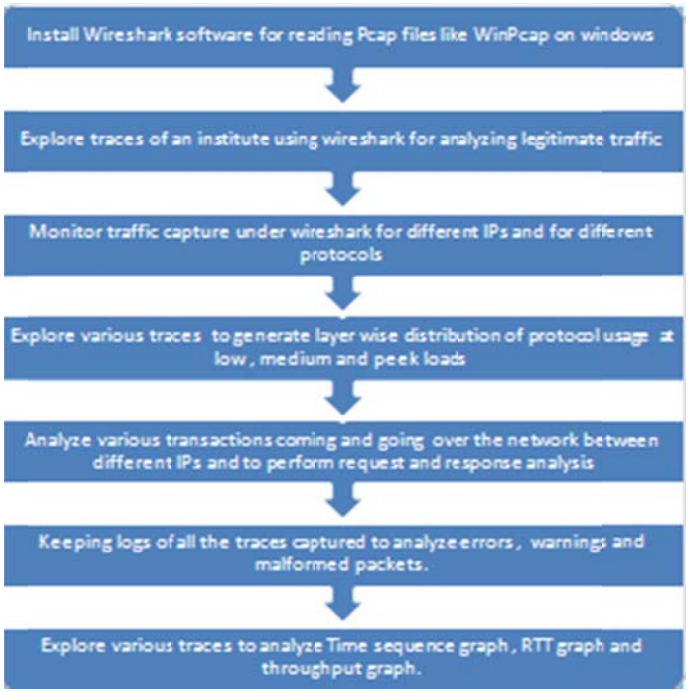


Figure 2. Methodology For Using Wireshark

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 as shown in Figure 3.

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] Table II. shown below displays the statistics for different traces that we obtained with respect to the protocols used.

Figure 3. Wireshark placements using a Hub in an Institute

Table 2 . Summary Of Protocol Distribution On The Basis Of Mbits/s

	Trace 1	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 Protocol	-	-	-	-	0.000	-
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.This means we have more number of conversation between sender and receiver.

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] HTTP Statisticts for all the six traces captured are shown in Tables III,IV,V,VI,VII,VIII .

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] HTTP Statisticts for all the six traces captured are shown in Tables III,IV,V,VI,VII,VIII .

Table 3. Http Statistics For Trace 1

Topic/Item	Count	Rate (ms)	%	Topic/Item	Count	Rate (ms)	%
HTTP Request By Server	5	0.005630					
a)HTTP Requests by	5	0.005630	100%	b)HTTP Responses by	5	0.005630	

server address				server address			
74.125.236.33	1	0.001126	20.00%	74.125.236.33	1	0.001126	20.00%
173.194.36.64	1	0.001126	20.00%	173.194.36.64	1	0.001126	20.00%
173.194.36.78	3	0.003378	60.00%	173.194.36.78	3	0.003378	60.00%
HTTP Requests by HTTP Host	5	0.005630	100%				

Table 4. Http Statistics For Trace 2

Topic/Item	Count	Rate (ms)	%	Topic/Item	Count	Rate (ms)	%
HTTP Request By Server	33	0.000082					
a)HTTP Requests by server address	33	0.000082	100%	b)HTTP Responses by server address	30	0.000075	
49.200.255.209	3	0.000007	9.09 %	49.200.255.209	3	0.000007	10.00 %
64.4.11.42	2	0.000005	6.06 %	64.4.11.42	2	0.000005	6.67 %
65.55206.229	2	0.000005	6.06 %	65.55.206.229	2	0.000005	6.67 %
131.253.13.140	2	0.000005	6.06 %	131.253.13.140	2	0.000005	6.67 %
207.46.61.29	2	0.000005	6.06 %	74.125.200.94	1	0.000002	3.33 %
74.125.200.94	1	0.000002	3.03 %	173.194.117.9	3	0.000007	9.09 %
173.194.117.9	3	0.000007	9.09 %	173.194.117.6	12	0.000030	40.00 %
173.194.117.6	13	0.000032	39.39 %	65.55.11.179	5	0.000012	16.67 %
65.55.11.1793	5	0.000012	15.15 %				
HTTP Requests by HTTP Host	33	0.000082	100%				

TABLE 5. HTTP STATISTICS FOR TRACE 3

Topic/Item	Count	Rate (ms)	%	Topic/Item	Count	Rate (ms)	%
HTTP Request By Server	259	0.000450					

HTTP Requests by server address	259	0.000450	100%	b) HTTP Responses by server address	236	0.000410	100%
124.124.201.200	4	0.000007	1.54%	124.124.201.200	4	0.000007	1.69%
65.54.82.145	1	0.000002	0.39%	65.54.82.145	1	0.000002	0.42%
58.26.185.42	1	0.000002	0.39%	58.26.185.42	1	0.000002	0.42%
23.47.235.27	2	0.000003	0.77%	23.47.235.27	2	0.000003	0.85%
23.41.75.27	2	0.000003	0.77%	23.41.75.27	2	0.000003	0.85%
58.26.185.57	10	0.000017	3.86%	58.26.185.57	7	0.000012	2.97%
58.26.185.65	3	0.000005	1.16%	58.26.185.65	1	0.000002	0.42%
65.54.51.252	1	0.000002	0.39%	65.54.51.252	-	-	
58.26.185.35	1	0.000002	0.39%	58.26.185.35	1	0.000002	0.42%
58.26.185.66	1	0.000002	0.39%	58.26.185.66	-	-	
74.125.236.198		0.000007	1.54%	74.125.236.198	4	0.000007	1.69%
74.125.200.94	2	0.000003	0.77%	74.125.200.94	2	0.000003	0.85%
76.74.254.120	1	0.000002	0.39%	76.74.254.120	1	0.000002	0.42%
68.232.44.111		0.000047	10.42%	68.232.44.111	24	0.000042	10.17%
68.232.44.121	53	0.000266	59.07%	68.232.44.121	46	0.000254	61.86%
68.232.44.251	1	0.000002	0.39%	68.232.44.251	1	0.000002	0.42%
199.22.77.192	1	0.000002	0.39%	199.22.77.192	1	0.000002	0.42%
192.0.80.247	2	0.000003	0.77%	192.0.80.247	2	0.000003	0.85%
54.246.174.85	2	0.000003	0.77%	54.246.174.85	2	0.000003	0.85%
182.50.136.239	3	0.000005	1.93%	182.50.136.239	3	0.000005	1.27%
50.18.52.222	2	0.000003	0.77%	50.18.52.222	2	0.000003	0.85%
184.72.54.6	1	0.000	0.39	184.72.54.69	1	0.000002	0.42

9		002	%				%
74.125.236.199	1	0.000002	0.39 %	74.125.236.199	1	0.000002	0.42 %
173.194.36.69	6	0.000010	2.32 %	173.194.36.69	6	0.000010	2.54 %
23.58.43.27	1	0.000002	0.39 %	23.58.43.27	1	0.000002	0.42 %
58.27.124.163	2	0.000003	0.77 %	58.27.124.163	1	0.000002	0.42 %
58.27.124.154	5	0.000009	1.93 %	58.27.124.154	1	0.000002	0.42 %
58.27.124.202	19	0.000033	7.34 %	58.27.124.202	18	0.000031	7.63 %
HTTP Request By Host	259	0.000450	100%				

Table 6. Http Statisticsvfor Trace 4

Topic/Item	Count	Rate (ms)	%	Topic/Item	Count	Rate (ms)	%
HTTP Request By Server	222	0.000199					
a)HTTP Requests by server address	2	0.000199	100%	b)HTTP Responses by server address	99	0.000410	100%
173.194.36.95	9	0.000008	4.05	173.194.36.95	8	0.000007	4.02
162.159.242.165	7	0.000024	12.16	162.159.242.165	7	0.000024	13.57
74.125.200.95	10	0.000009	4.50	74.125.200.95	0	0.000009	5.03
173.194.36.80	8	0.000009	3.60	173.194.36.80	8	0.000007	4.02
149.101.116.148	1	0.000001	0.45	149.101.116.148	1	0.000001	0.50
173.194.36.66	3	0.000003	1.35	173.194.36.66	2	0.000002	1.01
173.194.36.81	2	0.000002	0.90	173.194.36.81	2	0.000002	1.01
173.194.36.78	2	0.000002	0.90	173.194.36.78	2	0.000002	1.01
173.194.36.84	6	0.000005	2.70	173.194.36.84	6	0.000005	3.02
173.194.36.64	1	0.000001	0.45	173.194.36.64	1	0.000001	0.50

23.57.235.27	2	0.000002	0.90	23.57.235.27	2	0.000002	1.01
173.194.36.68	5	0.000004	2.25	173.194.36.68	5	0.000004	2.51
123.30.6.20	1	0.000001	0.45	123.30.6.20	1	0.000001	0.50
203.162.234.46	1	0.000001	0.45	203.162.234.46	1	0.000001	0.50
50.17.200.145	12	0.000011	5.41	50.17.200.145	12	0.000011	6.03
173.194.36.89	5	0.000004	2.25	173.194.36.89	4	0.000004	2.01
173.194.36.77	3	0.000003	1.35	173.194.36.77	3	0.000003	1.51
54.230.172.173	5	0.000004	2.25	54.230.172.173	5	0.000004	2.51
54.230.172.70	2	0.000002	0.90	54.230.172.70	2	0.000002	1.01
173.194.36.67	8	0.000007	3.60	173.194.36.67	8	0.000007	4.02
173.194.36.90	3	0.000003	1.35	173.194.36.90	3	0.000003	1.51
54.230.196.133	5	0.000004	2.25	54.230.196.133	-	-	-
173.194.126.95	6	0.000005	2.70	173.194.126.95	5	0.000004	2.01
162.159.241.165	17	0.000015	7.66	162.159.241.165	17	0.000015	8.54
173.194.36.82	4	0.000004	1.80	173.194.36.82	4	0.000004	2.01
46.28.209.33	5	0.000004	2.25	46.28.209.33	-	-	-
198.255.206.16	3	0.000003	1.35	198.255.206.16	3	0.000003	1.51
141.101.114.59	2	0.000002	0.90	141.101.114.59	2	0.000002	1.01
103.31.6.36	2	0.000002	0.90	103.31.6.36	2	0.000002	1.01
173.255.243.189	4	0.000004	1.80	173.255.243.189	3	0.000004	1.51
58.26.185.51	1	0.000001	0.45	58.26.185.51	2	0.000002	1.01
190.93.247.58	4	0.000004	1.80	190.93.247.58	4	0.000004	2.01
96.44.147.186	3	0.000003	1.35	96.44.147.186	2	0.000002	1.01
203.190.12	1	0.000	0.45	203.190.124.12	1	0.000001	0.50

4.12		001					
184.26.197.54	2	0.000002	0.90	184.26.197.54	1	0.000001	0.50
117.18.237.29	2	0.000002	0.90	117.18.237.29	2	0.000002	1.01
173.194.36.70	1	0.000001	0.45	173.194.36.70	1	0.000001	0.50
74.125.200.99	2	0.000002	0.90	74.125.200.99	2	0.000002	1.01
8.27.248.254	2	0.000002	0.90	8.27.248.254	-	-	-
74.125.200.147	1	0.000001	0.45	74.125.200.147	1	0.000001	0.50
74.125.236.218	1	0.000001	0.45	74.125.236.218	1	0.000001	0.50
131.229.72.11	19	0.000017	8.56	131.229.72.11	17	0.000015	8.54
184.26.23.165	2	0.000002	0.90	184.26.23.165	2	0.000002	1.01
173.194.36.88	1	0.000001	0.45	173.194.36.88	1	0.000001	0.50
176.101.52.178	14	0.000013	6.31	176.101.52.178	12	0.000011	6.03
184.169.176.213	2	0.000002	0.90	184.169.176.213	2	0.000002	1.01
HTTP Request By Host	2	0.000199	100%				

Table 7. Http Statistics For Trace 5

Topic/Item	Count	Rate (ms)	%	Topic/Item	Count	Rate (ms)	%
HTTP Request By Server	2	0.000004					
a)HTTP Requests by server address	2	0.000004	100%	b)HTTP Responses by server address	1	0.000002	
23.198.100.239	1	0.000002	50	23.198.100.239		-	-
124.124.252.8		0.000002	50	124.124.252.8	1	0.000002	100
HTTP Request By Host	2	0.000004	100%				

Table 8. Http Statistics For Trace 6

Topic/Item	Count	Rate (ms)	%	Topic/Item	Count	Rate (ms)	%
HTTP Request By Server	90	0.000082					
HTTP Requests by server address	90	0.000082	100%	b)HTTP Responses by server address	77	0.000075	
23.41.75.21	1	0.000000	1.11	23.41.75.21	1	0.000000	1.30
199.7.51.72	1	0.000000	1.11	199.7.51.72	1	0.000000	1.30
58.27.124.202	1	0.000000	1.11	58.27.124.202	-	-	-
58.27.124.219	3	0.000001	3.33	58.27.124.219	-	-	-
124.124.252.9	14	0.000004	15.56	124.124.252.9	2	0.000004	15.58
65.55.192.94	1	0.000000	1.11	65.55.192.94		-	-
65.55.58.195	1	0.000000	1.11	65.55.58.195	1	0.000000	1.30
65.52.33.27	1	0.000000	1.11	65.52.33.27	1	0.000000	1.30
65.54.82.158	1	0.000000	1.11	65.54.82.158	1	0.000000	1.30
124.124.255.25	32	0.000010	35.56	124.124.255.25		0.000010	38.96
124.124.252.99	4	0.000001	4.44	124.124.252.99	-	-	-
23.47.235.27	3	0.000001	3.33	23.47.235.27		0.000001	3.90
173.194.36.72	5	0.000002	5.56	173.194.36.72		0.000002	6.49
173.194.36.71	1	0.000000	1.11	173.194.36.71	1	0.000000	1.30
173.194.36.65	10	0.000003	11.11	173.194.36.65		0.000003	12.99
23.51.43.27	2	0.000001	2.22	23.51.43.27		0.000001	2.60
199.7.55.72	4	0.000001	4.44	199.7.55.72		0.000001	5.19
177.18.237.29	3	0.000001	3.33	177.18.237.29	3	0.000001	3.90
175.43.124.	1	0.000	1.11	175.43.124.200	1	0.000000	1.30

200		000					
173.194.36.66	1	0.000000	1.11	173.194.36.66	1	0.000000	1.30
HTTP Request By Host	90	0.000029	100%				

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.

Expert Analysis Summary

Table 9. 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)

The Expert info table shown in Table IX. 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.

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

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.

Case 1. Trace 3

For graph analysis we have to look at the Flow graph of the trace shown in Figure 4. and from there we plot RTT for each TCP segment sent .Also from the trace we can calculate Throughput of it.

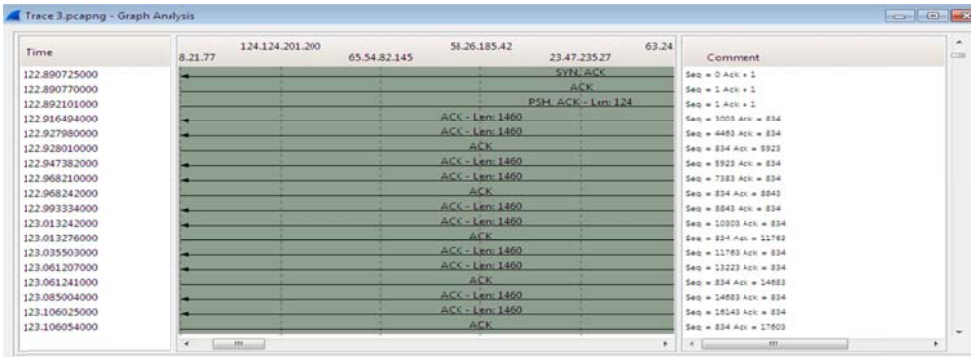


Figure 4. Flow Graph

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

Table 10. 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. Time sequence graph of these segments is shown in Figure 5. RTT graph in Figure 6. and Throughput graph in Figure 7.

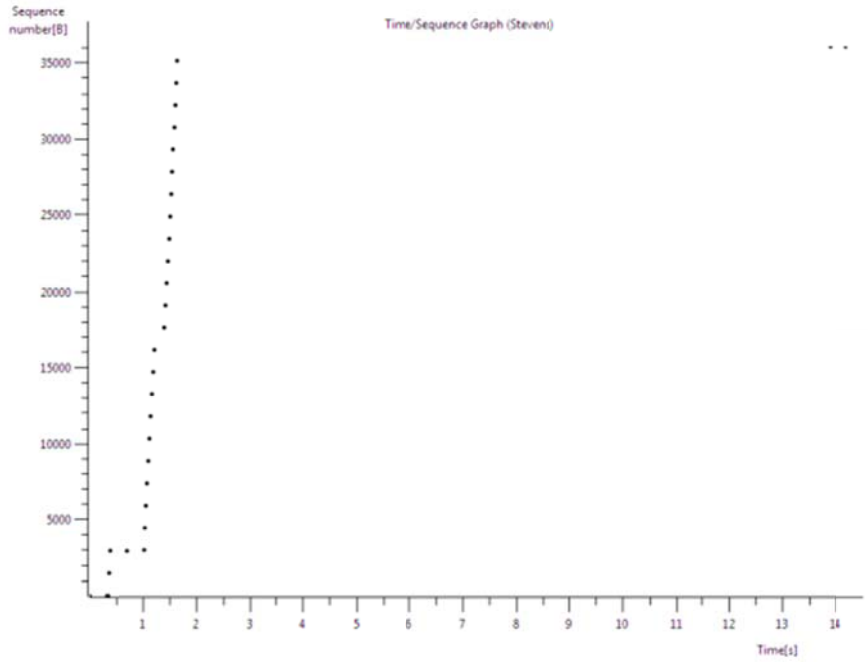


Figure 5. Time Sequence Graph

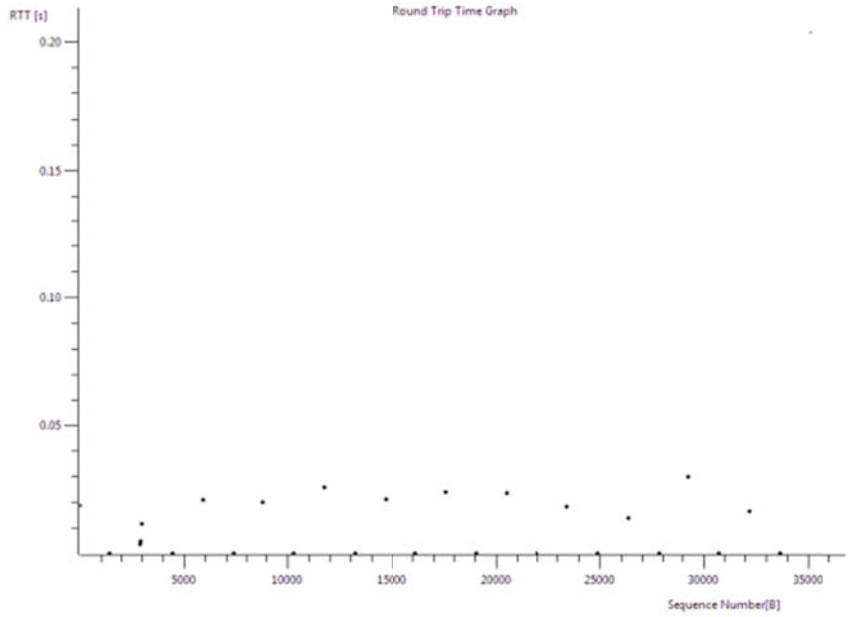


Figure 6. Round Trip Time Graph

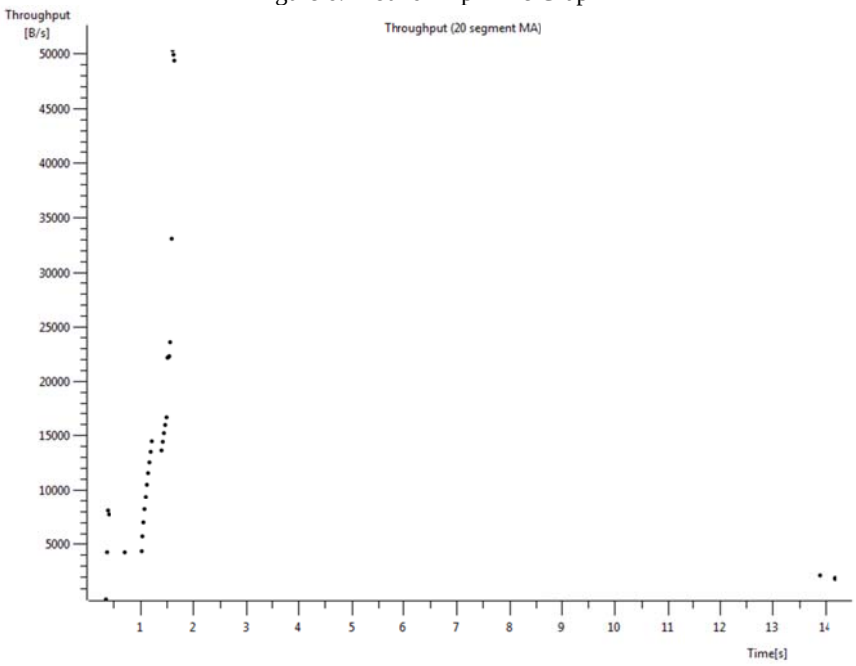


Figure 7. Throughput graph

Case2 : Trace 4

For graph analysis of Trace 4 Flow Graph is shown in Figure 8. Below by which we can calculate RTT of first six segments.

Figure 8. Flow Graph For Trace 4

Table 11. 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 the RTT calculation shown in Table XI. 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 in Figure 9.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 below in Figure 9. Is of throughput calculation And Time sequence graph for this in Figure 10. RTT in Figure 11. Zoomed RTT in Figure 12. and Throughput Graph in Figure 13.

No.	Time	Source	Destination	Protocol	Length	Info
13165	118.501299000	162.159.242.165	192.168.21.77	HTTP	504	HTTP/1.0 304 Not Modified
13166	118.501304000	162.159.242.165	192.168.21.77	TCP	60	http > 52659 [FIN, ACK] Seq=451 Ack=914 Win=7680 Len
13167	118.501350000	192.168.21.77	162.159.242.165	TCP	54	52659 > http [ACK] Seq=914 Ack=452 Win=65248 Len=0
13168	118.501677000	192.168.21.77	162.159.242.165	TCP	54	52659 > http [FIN, ACK] Seq=914 Ack=452 Win=65248 Le
13169	118.502053000	192.168.21.77	162.159.242.165	TCP	66	52671 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=
13171	118.534713000	192.168.21.77	8.254.128.254	TCP	54	52644 > http [ACK] Seq=1 Ack=1538841 Win=342 Len=0
13172	118.535510000	162.159.242.165	192.168.21.77	TCP	60	http > 52664 [ACK] Seq=1 Ack=903 Win=7680 Len=0
13173	118.535511000	162.159.242.165	192.168.21.77	HTTP	502	HTTP/1.0 304 Not Modified
13175	118.535513000	162.159.242.165	192.168.21.77	TCP	60	http > 52664 [FIN, ACK] Seq=449 Ack=903 Win=7680 Len
13176	118.535619000	192.168.21.77	162.159.242.165	TCP	54	52664 > http [ACK] Seq=903 Ack=450 Win=65252 Len=0
13177	118.535851000	192.168.21.77	162.159.242.165	TCP	54	52664 > http [FIN, ACK] Seq=903 Ack=450 Win=65252 Le
13178	118.536203000	192.168.21.77	162.159.242.165	TCP	66	52672 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=
13182	118.584928000	74.125.200.95	192.168.21.77	TCP	384	[TCP segment of a reassembled PDU]
13183	118.597546000	192.168.21.77	173.194.36.81	TCP	66	52673 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=
13184	118.608389000	74.125.200.95	192.168.21.77	HTTP	1031	HTTP/1.0 404 Not Found (text/html)
13185	118.608423000	192.168.21.77	74.125.200.95	TCP	54	52660 > http [ACK] Seq=399 Ack=1308 Win=64392 Len=0
13186	118.608647000	192.168.21.77	74.125.200.95	TCP	54	52660 > http [FIN, ACK] Seq=399 Ack=1308 Win=64392 L
13187	118.613365000	192.168.21.77	162.159.242.165	TCP	66	52674 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=
13188	118.613549000	192.168.21.77	173.194.36.81	TCP	66	52675 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=
13191	118.630981000	74.125.200.95	192.168.21.77	TCP	60	http > 52660 [FIN, ACK] Seq=1308 Ack=399 Win=6912 Le

Figure 9. Screenshot of wireshark screen of Trace 4

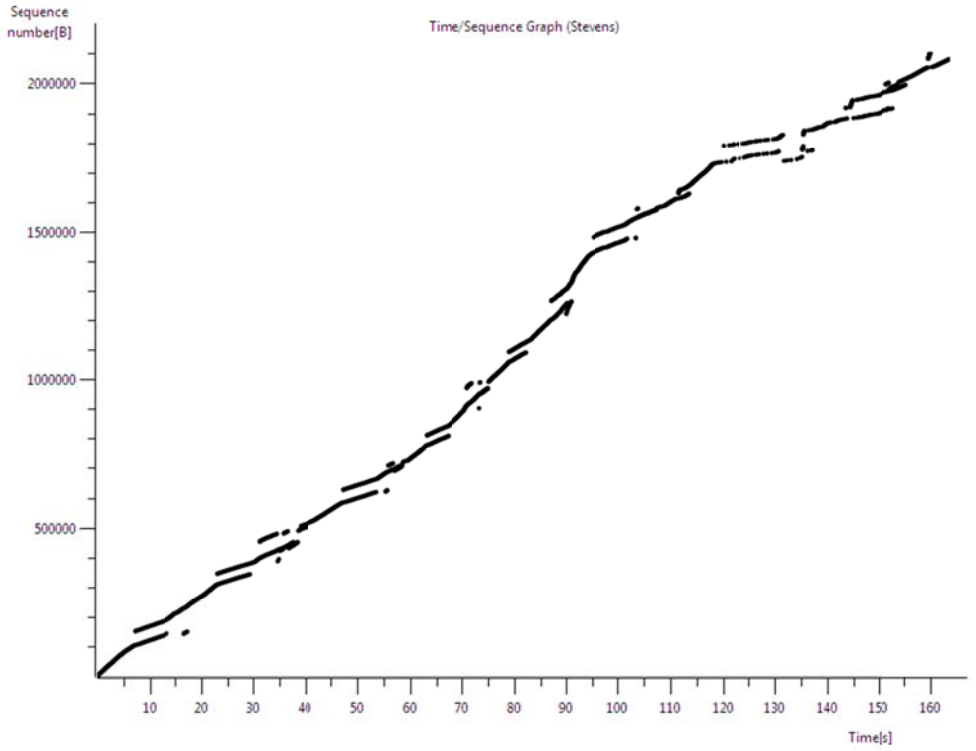


Figure 10. Time Sequence Graph

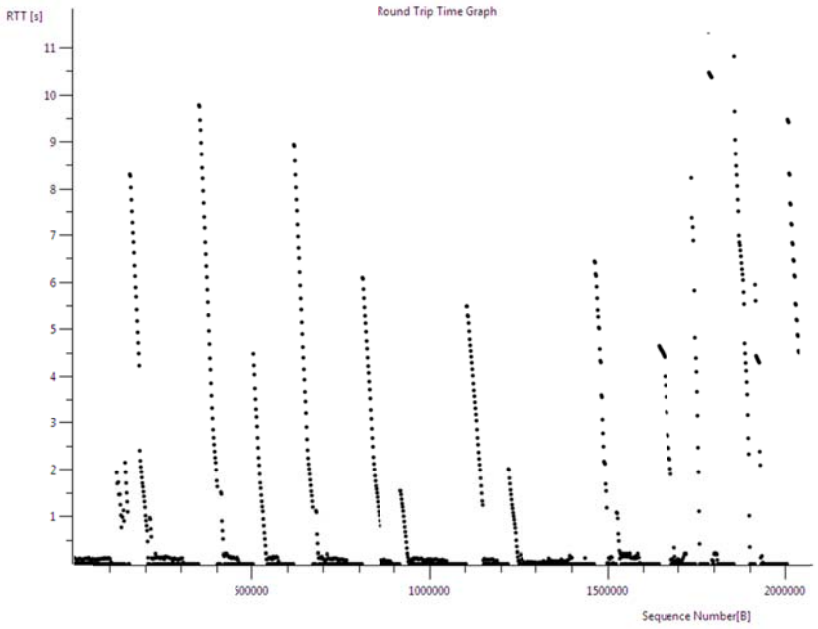


Figure 11. Round Trip Time Graph

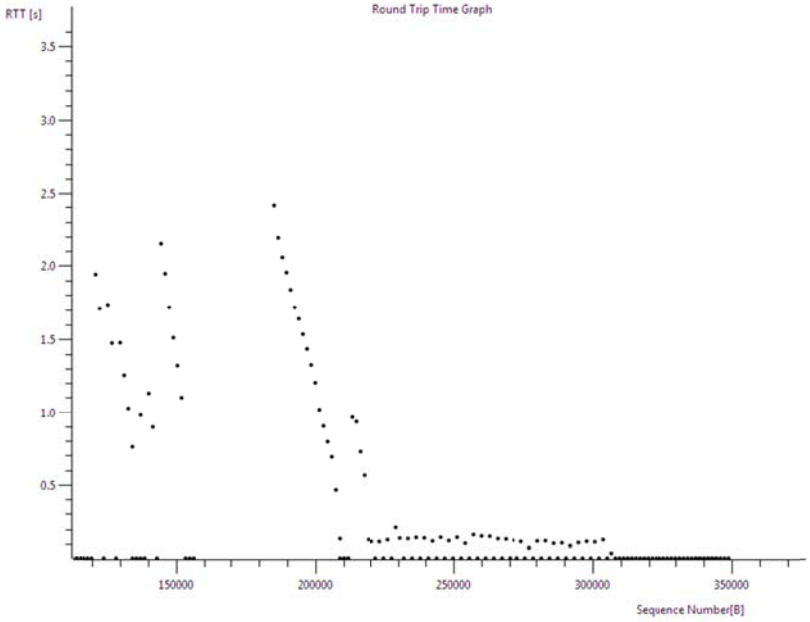


Figure 12. RTT graph (Zoom)

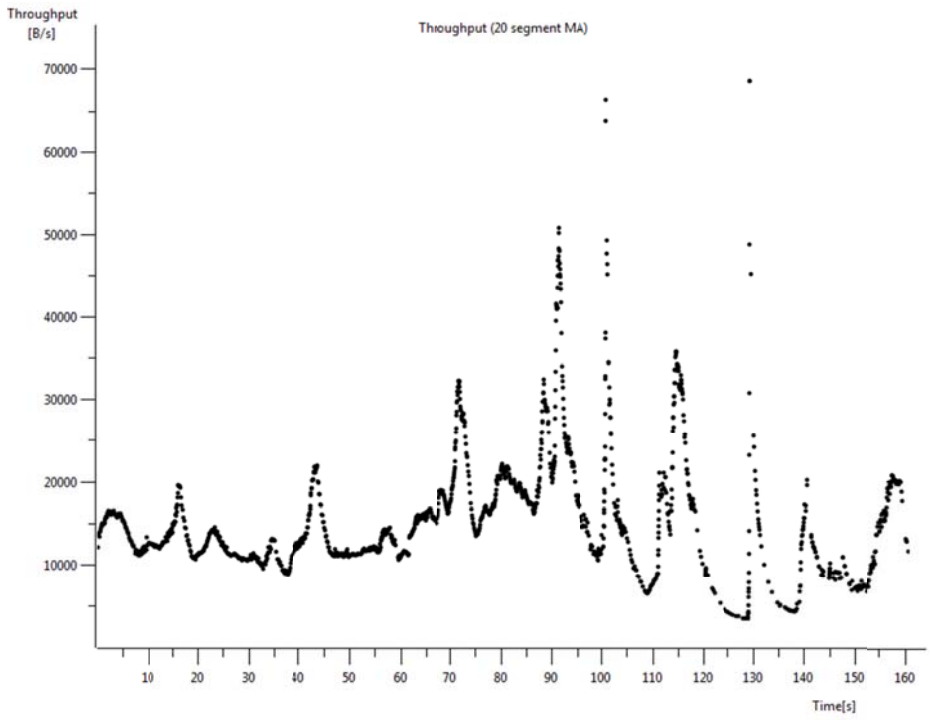


Figure 13. Throughput Graph

- 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 as shown in Figure 14.

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.

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) as shown in Figure 15.

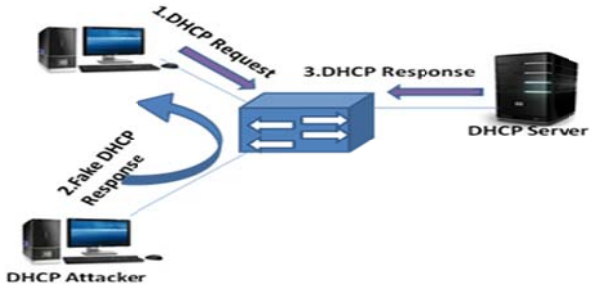


Figure 14. DHCP Spoofing

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 as shown in Figure 16.with the objective of finishing-up the range of IP available in the DHCP server. Graph for DHCP Spoofing is shown in Figure 17. which shows how multiple packets are coming in small time span. To get out of this type of problems many tools are available for free.

No.	Time	Source	Destination	Protocol	Length	Info
3369	35.803253000	192.168.0.99	255.255.255.255	DHCP	342	DHCP ACK - Transaction ID 0x35b3caf2
6728	68.880647000	192.168.0.99	255.255.255.255	DHCP	342	DHCP ACK - Transaction ID 0x24118e5a
12932	116.607793000	192.168.0.99	192.168.21.77	DHCP	342	DHCP ACK - Transaction ID 0x9f63db91
43652	512.786311000	192.168.0.99	255.255.255.255	DHCP	342	DHCP ACK - Transaction ID 0x52141afc
55582	630.733415000	192.168.0.99	255.255.255.255	DHCP	342	DHCP ACK - Transaction ID 0xeb5deaf9
58096	669.177940000	192.168.0.99	255.255.255.255	DHCP	342	DHCP ACK - Transaction ID 0xfc94f9ff
72475	958.643047000	192.168.0.99	255.255.255.255	DHCP	342	DHCP ACK - Transaction ID 0xd6108b0f
72607	962.177376000	192.168.0.99	255.255.255.255	DHCP	342	DHCP ACK - Transaction ID 0xd6108b0f
74740	983.263548000	192.168.0.99	255.255.255.255	DHCP	342	DHCP ACK - Transaction ID 0xcfe04452
80577	1103.733290000	192.168.0.99	255.255.255.255	DHCP	342	DHCP ACK - Transaction ID 0x4ed0763f

Figure 15. DHCP Filter

No.	Time	Source	Destination	Protocol	Length	Info
18145	161.477629000	192.168.23.47	255.255.255.255	DHCP	342	DHCP Inform - Transaction ID 0xf9861d59
18258	163.254898000	0.0.0.0	255.255.255.255	DHCP	351	DHCP Discover - Transaction ID 0xdb6866c4
18265	163.403255000	0.0.0.0	255.255.255.255	DHCP	363	DHCP Request - Transaction ID 0xdb6866c4
18640	167.608708000	0.0.0.0	255.255.255.255	DHCP	345	DHCP Request - Transaction ID 0x39a6fe09
18793	170.596705000	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x2857dd97
18794	170.617814000	0.0.0.0	255.255.255.255	DHCP	351	DHCP Request - Transaction ID 0x2857dd97
18937	174.256376000	0.0.0.0	255.255.255.255	DHCP	342	DHCP Decline - Transaction ID 0x2857dd97
18953	174.508168000	0.0.0.0	255.255.255.255	DHCP	354	DHCP Request - Transaction ID 0xd08ff5f9
19129	177.574967000	192.168.23.19	255.255.255.255	DHCP	342	DHCP Inform - Transaction ID 0x5269b76a
19471	180.573559000	192.168.23.19	255.255.255.255	DHCP	342	DHCP Inform - Transaction ID 0x5269b76a
19890	184.221952000	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xcb415e7b
20275	188.347628000	0.0.0.0	255.255.255.255	DHCP	345	DHCP Request - Transaction ID 0xc9176296
20380	191.478739000	0.0.0.0	255.255.255.255	DHCP	342	DHCP Decline - Transaction ID 0xc9176296
20549	196.332450000	0.0.0.0	255.255.255.255	DHCP	343	DHCP Request - Transaction ID 0xfd6a9aac

Figure 16. DHCP Exhaustion

Figure 20. 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

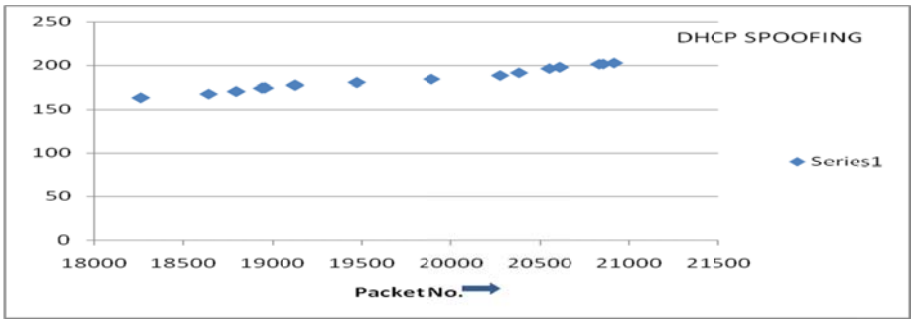


Figure 17. Graph for DHCP Spoofing

DDOS Attack

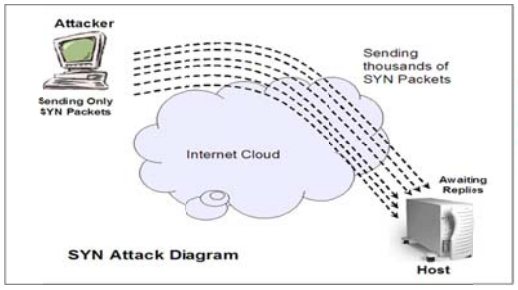


Figure 18 . DDOS SYN Attack

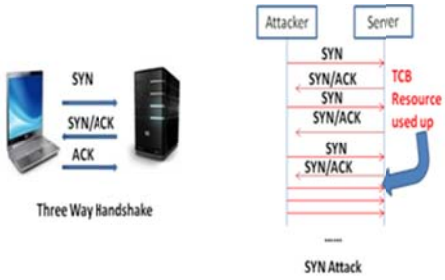


Figure 19. Showing Three way handshake process and SYN Attack

the same IP also shown in Figure 18. 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.
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 shown in Figure 19. 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. 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 shown in Figure 21. In Figure 22. 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

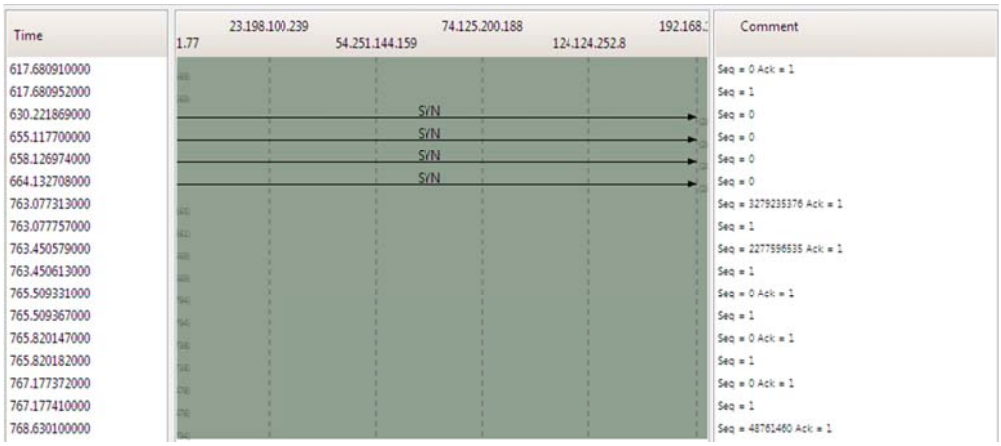


Figure 20. Flow Graph

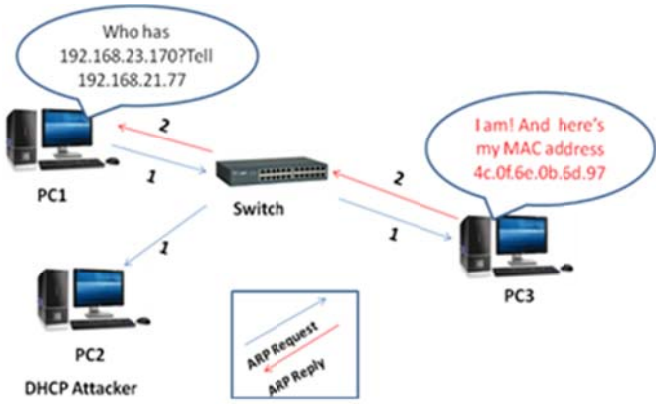


Figure 21 . ARP Request /Reply

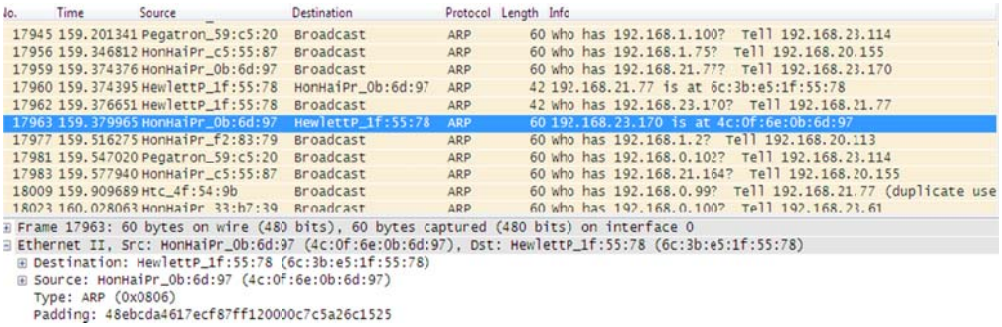


Figure 22. Wireshark Areas Of ARP Packets

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.

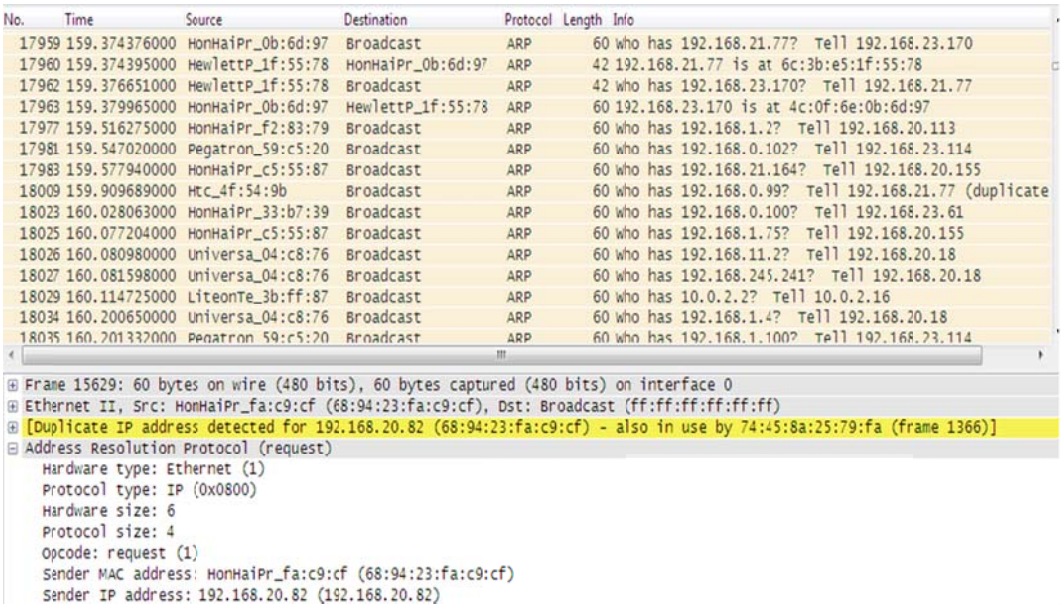


Figure 23. Arp capturing Duplicate IP address which is first used in frame no.1366

The hexadecimal text 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 1366*" 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 23.

There might be another situation when number of packets are coming from same IP address continuously as shown in Figure 24. And this is for attacking purpose. Graph in Figure 25 shows at time interval near 9.15 there are continous packets coming from same address.

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 26. And graph is shown in Figure 27 which shows multiple requests in a short period.

No.	Time	Source	Destination	Protocol	Length	Info
22819	754.537879000	SamsungE_54:6a:8a	Broadcast	ARP	60	who has 192.168.20.93? Tell 192.168.20.226
22820	754.542729000	SamsungE_54:6a:8a	Broadcast	ARP	60	who has 192.168.20.94? Tell 192.168.20.226
22821	754.548914000	SamsungE_54:6a:8a	Broadcast	ARP	60	who has 192.168.20.95? Tell 192.168.20.226
22822	754.554028000	SamsungE_54:6a:8a	Broadcast	ARP	60	who has 192.168.20.96? Tell 192.168.20.226
22823	754.560064000	SamsungE_54:6a:8a	Broadcast	ARP	60	who has 192.168.20.97? Tell 192.168.20.226
22824	754.566347000	SamsungE_54:6a:8a	Broadcast	ARP	60	who has 192.168.20.101? Tell 192.168.20.226
22825	754.572144000	SamsungE_54:6a:8a	Broadcast	ARP	60	who has 192.168.20.102? Tell 192.168.20.226
22826	754.576145000	SamsungE_54:6a:8a	Broadcast	ARP	60	who has 192.168.20.103? Tell 192.168.20.226
22827	754.580802000	SamsungE_54:6a:8a	Broadcast	ARP	60	who has 192.168.20.104? Tell 192.168.20.226
22828	754.591115000	SamsungE_54:6a:8a	Broadcast	ARP	60	who has 192.168.20.105? Tell 192.168.20.226
22829	754.596247000	SamsungE_54:6a:8a	Broadcast	ARP	60	who has 192.168.20.106? Tell 192.168.20.226
22830	754.603047000	SamsungE_54:6a:8a	Broadcast	ARP	60	who has 192.168.20.108? Tell 192.168.20.226
22831	754.607757000	SamsungE_54:6a:8a	Broadcast	ARP	60	who has 192.168.20.109? Tell 192.168.20.226
22832	754.612516000	SamsungE_54:6a:8a	Broadcast	ARP	60	who has 192.168.20.110? Tell 192.168.20.226
22833	754.618183000	SamsungE_54:6a:8a	Broadcast	ARP	60	who has 192.168.20.111? Tell 192.168.20.226
22834	754.622770000	SamsungE_54:6a:8a	Broadcast	ARP	60	who has 192.168.20.112? Tell 192.168.20.226
22835	754.627492000	SamsungE_54:6a:8a	Broadcast	ARP	60	who has 192.168.20.114? Tell 192.168.20.226
22836	754.634016000	SamsungE_54:6a:8a	Broadcast	ARP	60	who has 192.168.20.128? Tell 192.168.20.226
22837	754.639057000	SamsungE_54:6a:8a	Broadcast	ARP	60	who has 192.168.20.129? Tell 192.168.20.226
22838	754.645362000	SamsungE_54:6a:8a	Broadcast	ARP	60	who has 192.168.20.130? Tell 192.168.20.226
22839	754.650085000	SamsungE_54:6a:8a	Broadcast	ARP	60	who has 192.168.20.131? Tell 192.168.20.226
22840	754.655696000	SamsungE_54:6a:8a	Broadcast	ARP	60	who has 192.168.20.132? Tell 192.168.20.226
22841	754.661320000	SamsungE_54:6a:8a	Broadcast	ARP	60	who has 192.168.20.133? Tell 192.168.20.226
22842	754.667293000	SamsungE_54:6a:8a	Broadcast	ARP	60	who has 192.168.20.141? Tell 192.168.20.226
22843	754.668431000	SamsungE_54:6a:8a	Broadcast	ARP	60	who has 192.168.20.142? Tell 192.168.20.226
22844	754.675834000	SamsungE_54:6a:8a	Broadcast	ARP	60	who has 192.168.20.143? Tell 192.168.20.226
22845	754.680366000	SamsungE_54:6a:8a	Broadcast	ARP	60	who has 192.168.20.144? Tell 192.168.20.226
22846	754.683463000	SamsungE_54:6a:8a	Broadcast	ARP	60	who has 192.168.20.145? Tell 192.168.20.226
22847	754.687984000	SamsungE_54:6a:8a	Broadcast	ARP	60	who has 192.168.20.146? Tell 192.168.20.226
22849	754.703793000	SamsungE_54:6a:8a	Broadcast	ARP	60	who has 192.168.20.162? Tell 192.168.20.226

Figure 24. ARP spoofing window

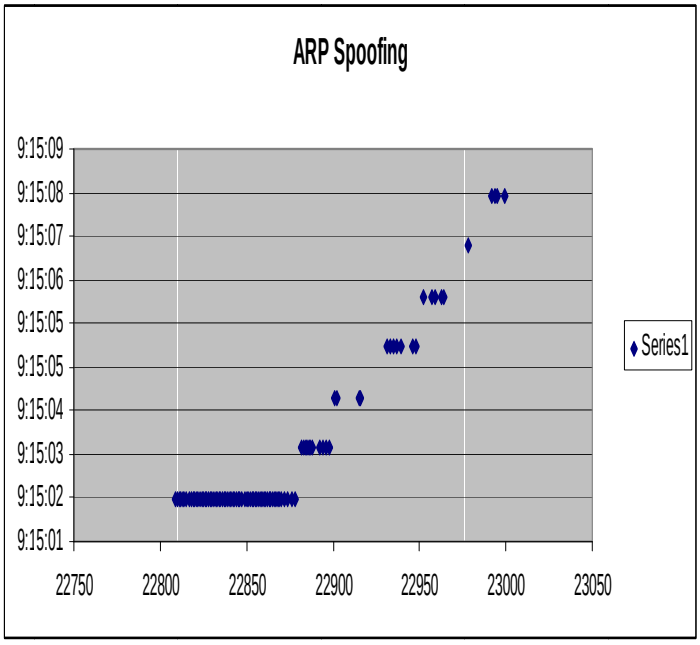
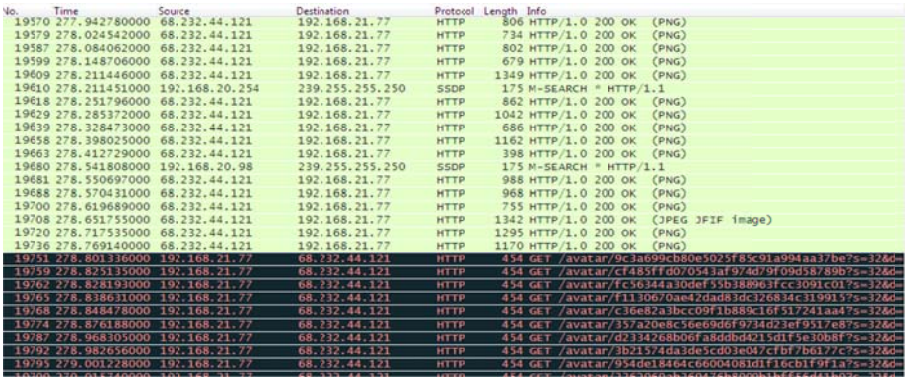


Figure 25. Graph of ARP Spoofing



No.	Time	Source	Destination	Protocol	Length	Info
19570	277.942780000	68.232.44.121	192.168.21.77	HTTP	806	HTTP/1.0 200 OK (PNG)
19579	278.024542000	68.232.44.121	192.168.21.77	HTTP	734	HTTP/1.0 200 OK (PNG)
19587	278.084062000	68.232.44.121	192.168.21.77	HTTP	802	HTTP/1.0 200 OK (PNG)
19599	278.148706000	68.232.44.121	192.168.21.77	HTTP	679	HTTP/1.0 200 OK (PNG)
19609	278.211446000	68.232.44.121	192.168.21.77	HTTP	1349	HTTP/1.0 200 OK (PNG)
19610	278.211451000	192.168.20.254	239.255.255.250	SSDP	175	M-SEARCH * HTTP/1.1
19618	278.251796000	68.232.44.121	192.168.21.77	HTTP	862	HTTP/1.0 200 OK (PNG)
19629	278.285372000	68.232.44.121	192.168.21.77	HTTP	1042	HTTP/1.0 200 OK (PNG)
19639	278.328473000	68.232.44.121	192.168.21.77	HTTP	686	HTTP/1.0 200 OK (PNG)
19658	278.398025000	68.232.44.121	192.168.21.77	HTTP	1162	HTTP/1.0 200 OK (PNG)
19663	278.412729000	68.232.44.121	192.168.21.77	HTTP	398	HTTP/1.0 200 OK (PNG)
19680	278.541808000	192.168.20.98	239.255.255.250	SSDP	175	M-SEARCH * HTTP/1.1
19681	278.550697000	68.232.44.121	192.168.21.77	HTTP	988	HTTP/1.0 200 OK (PNG)
19688	278.570431000	68.232.44.121	192.168.21.77	HTTP	968	HTTP/1.0 200 OK (PNG)
19700	278.619689000	68.232.44.121	192.168.21.77	HTTP	755	HTTP/1.0 200 OK (PNG)
19708	278.651755000	68.232.44.121	192.168.21.77	HTTP	1342	HTTP/1.0 200 OK (JPEG JFIF image)
19720	278.717535000	68.232.44.121	192.168.21.77	HTTP	1295	HTTP/1.0 200 OK (PNG)
19736	278.769140000	68.232.44.121	192.168.21.77	HTTP	1170	HTTP/1.0 200 OK (PNG)
19751	278.801336000	192.168.21.77	68.232.44.121	HTTP	454	GET /avatar/9c3a699cb80e5025f85c91a994aa37be?s=32&d=
19759	278.825135000	192.168.21.77	68.232.44.121	HTTP	454	GET /avatar/cf485ffd07054af974d79f09d58789b?s=32&d=
19762	278.828193000	192.168.21.77	68.232.44.121	HTTP	454	GET /avatar/fc56344a30def55b388963fcc3091c01?s=32&d=
19765	278.838631000	192.168.21.77	68.232.44.121	HTTP	454	GET /avatar/f1130670ae42dad83dc326834c319915?s=32&d=
19768	278.848478000	192.168.21.77	68.232.44.121	HTTP	454	GET /avatar/c36e82a3bcc09f1b889c16f517241aa4?s=32&d=
19774	278.876188000	192.168.21.77	68.232.44.121	HTTP	454	GET /avatar/357a20e8c56e69d6f9734d23ef9517e8?s=32&d=
19787	278.968305000	192.168.21.77	68.232.44.121	HTTP	454	GET /avatar/d2334268b06fa8ddbd4215d1f5e30b8f?s=32&d=
19792	278.982656000	192.168.21.77	68.232.44.121	HTTP	454	GET /avatar/3b21574da3de5cd03e047cfbf7b617c?s=32&d=
19795	279.001228000	192.168.21.77	68.232.44.121	HTTP	454	GET /avatar/954de18464c66004081d1f16cb1f9f1a?s=32&d=

Figure 26. HTTP Spidering

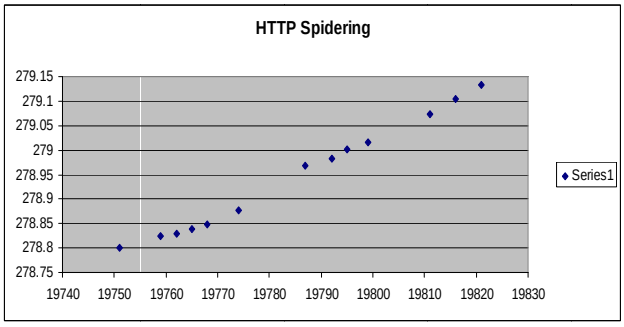


Figure 27. Graph of HTTP SpIdering

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

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

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

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

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

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

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.

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

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

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.

Justin Jay Lister (January , 1995) ,"Intrusion Detection Systems: An introduction to the detection and prevention of computer abuse", Computer Security Research, Department of Computer Science, University of Wollongong Copyright © 1994 by Justin Jay Lister

Emilie Lundin, et.al, "Survey of Intrusion Detection Research", Department of Computer Engineering Chalmers University of Technology, Technical Report nr. 02-04 ,2002.