Rajesh Kumar Tyagi et al. 2019, 7:6
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology

An Open Access Journal

A new tool to Monitor Network with LAMP Architecture

Rajesh Kumar Tyagi
Department of Computer Science and Engineering,
Amity University, Manesar Haryan, India

Abstract

Network Monitoring Tool enables you to log and monitor data owing in and out of your network. The program runs as a local service. It helps the administrator to check for congestion in the network and the root cause for it. The explained tool has been designed for ISO-OSI model 4th layer called as transport layer i.e. Specifically for two important protocol TCP (Transmission Control Protocol), UDP (User Datagram Protocol) and third layer protocol is ARP (Address Resolution Protocol).It traces timestamp, source IP, destination IP, source port, destination port and packet length. To keep the track record of all the communication and all the data transfer taking place, we analyzed these tools and showed the results in this paper. This technique is very effective for analyzing network traffic.

Keywords: ISO-OSI reference model,TCP,ARP and UDP,WAN.

I. INTRODUCTION

Network monitoring is the use of a system that constantly monitors a computer network for slow or failing components and that notices the network administrator in case of outages. Network Monitoring (NM) is part of network management [1-2].A network monitoring tool (NMT) is one which carries out the work of network monitoring and measures a few parameters of the functioning network to give the network administrator an idea of how the network is performing [3].

Commonly measured metrics are source, destination information, availability packet info, response-time, while consistency and reliability parameters are initial to gain reputation. The prevalent of Wide Area Network (WAN) optimization devices is having an undesirable effect on most network monitoring tools {particularly when it comes to measuring precise end-to-end reply time because they bound round trip visibility [4].

1. Current Working System

Currently there are numerous network monitoring tools accessible for usage. One such is Wire shark an online monitoring tool [5].

Wire shark is a open-source and free to use packet analyzer. Now a day's experts are used this tool for network troubleshooting, analyze, software and innovate protocol development, and education. Wire shark is a platform independent, using the GTK+ widget toolkit to implement its client interface, and using p-cap to capture packets; it runs on various plate forms such Unix-like operating systems including Linux, Microsoft Windows etc [6-7].

2. Our objective

The main-aim of this project is to widen a network monitoring tool usage. A network monitoring tool is dedicated for checking the operations of a computer/mobile networks. This tool is used to make sure ease of use and overall performance of the connecting hosts and system services. It monitors the network for the troubles caused by congested and/or gone down servers, network connections and other devices.

It notices the network administrator in case of outages so that he can handle the problem immediately. This Network Monitoring Tool essentially detects abnormal behavior of any host and in case of such abnormality checks for its activities by decrypting data (packets) sent across the network by this host machine. Thus, by reading this packet information we can analyze the nature of its conversation. Then we can take necessary action against that host in order to make it function properly.

Network Monitoring Tool (NMT) works with the help of tcp-dump. Tcp-dump can be used to confine some or all the communicating packets received by a device interface. The range of packets captured can be particular by the using a mixture of coherent operators and parameters such as source and destination Media Access Control(48 bits Mac Address) or Internet Protocol Addresses, types of protocol (IP and LAN) and TCP/UDP socket numbers.

II. PROPOSED ALGORITHM

1. Background work and Proposed Algorithm

Our main background algorithm is based on the Command Line Terminal interface in the Linux environment. Our commands issued in Linux Terminal will deal with the working part of our project while the user interface part that is the web pages encoded in HTML will be done in php/html architecture. The following are the steps of our proposed algorithm:

Initiate the packet sniffing phase.

1. First, capture all the packets propagating in the network using the tcpdump function. The number of packets can be explicitly specified.
2. Store the captured data in text files on the Linux OS.
3. Group all different packet types like UDP, TCP and ARP with their relevant information in different files. This is done using the grep command.
4. Now taking the TCP packet file as an example, we proceed with the packet processing phase.
5. Cut different fields from the TCP packet file segregating them into different text files for each relevant parameter.
6. The first field is stored in a new file consisting of the timestamp.
7. The second field is stored in a new file consisting of the source mac address.
8. The next relevant field that is the fourth field is again stored in a new file consisting of the destination mac address.
9. The same steps are carried out to store all the other different relevant information in separate text files. They are :
   1. Packet length
   2. Source IP address
   3. Destination IP address
   4. Source port number
   5. Destination port number

Now all 8 IP fields are combined into a new file using the paste command. Steps (5-11) are repeated for the UDP and ARP packets. Once done with extraction of all packet information of all the packet types, all of them are combined into a single file.

Now to identify the source IP address which is sending maximum no. of packets and possibly trying to congest the network, only the source IP addresses field of all packet files are placed in a single file. Now we initiate our result phase. We first sort all source IP addresses in a single file. Then we place a unique clause which displays all IP addresses along with the no. of packets. Then taking no. of packets as the condition, we sort all the source IP addresses again in the descending order of the no. of packets transmitted.

Finally, we implement the head function to display the top 5 source IP addresses along with descending packet instances. All the data which were stored in text files in the above steps are all sorted accordingly in MySQL RDBMS in order to make it available for the need for the network administrator. The UI (html/php web pages) display the final result to the administrator extracting it from MySQL through the Apache Server.

III. LAMP ARCHITECTURE

1. LAMP Working

LAMP is an acronym for Linux, Apache, MySQL, and PHP. LAMP is especially paying attention towards Web type applications. The structural design of LAMP is very straight forward, as illustrate in given below figure 3.1.

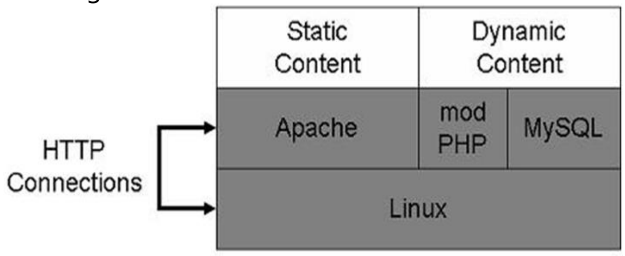


Figure 1: LAMP Architecture.

Ubuntu Linux 18.10 directly forwards Hyper Text Transfer Protocol links to Apache, which serves static content directly from the ubuntu Linux kernel 18.10. Dynamic web-pages are directly being forwarded by

Apache Server to PHP platform, which runs the PHP code to for designing the page. Database queries are forwarded to MySQL via PHP. Administration is commonly handled through PHP-My Admin platform, and every major project management system can Apache and Linux platform. There are plenty resources available on the Web for explaining how to swiftly build a LAMP application. Most software-developers are more creative within hours of installing the LAMP stack. Ubuntu Linux18.10 is not a necessary standard because software advancement often occurs on a Windows based platform running Apache, My-SQL, and either PHP and Python, or Perl (this configuration is called "WAMP"). LAMP is consisted of several separate free open-source projects, each with its own owners, software-packages, and distributors. Each of the core components of LAMP has a foremost commercial or non-profit foundation behind it i.e Linux: Red Hat and Ubuntu etc.

IV. METHODOLOGY

In designing this Network Monitoring Tool (NMT), following steps are used in sequence:-

1. Packet Sniffing
2. Packet Processing
3. Data Storage
4. User Interface

1. Packet Sniffing

First and most basic step is sniffing packets passing throughout the network. For this, we have used command-line packet capture tool tcp-dump. Tcp-dump is a common packet sniffer that execute under the command-line interface. It allows users to capture and exhibit TCP/IP and other packets being send or received over a computer network to which the computer is connected. Tcp-dump works on most of the UNIX/LINUX like operating systems: Linux/WINDOWS/MacOS X etc.

There is also a port of tcp-dump for windows called Win-dump. In various operating systems like Fedora, a user must have super-user privileges to use tcp-dump because the packet capturing mechanism on those systems requires high privileges. However, the -z option may be used to discard privileges to a specific un-privileged user after capturing has been set-up. TCP-DUMP can make available all the detailed information about any network dialogues that runs across the communicating channels. An advantage of TCP-DUMP is that it can be run remotely through a SSL or TEL-NET session, and the machine running TCP-DUMP doesn't have to be running x-windows. The function uses very little overhead, as it is a not friendly user- interface.

2. Packet Processing

Second step is to process the captured packets so that they can be written in database. Hence to move further we will process the packets by writing shell script on gedit editor. This shell script is the same script that also contains the tcpdump command in the beginning. As stated earlier, this script is invoked at runtime when the user signs in to use the tool.

For packet processing following UNIX commands have been used: edit traffic.

sh :: open new vi editor for writing shell script.Before executing we will be required to change file permission using 'chmod'. Then to execute write 'bash traffic.sh' on terminal (assuming file name to be trafffic.sh)

grep: (searching for a pattern) to search for all IP packets and put them into a separate file.

cut: (slitting a file vertically) to appropriately cut the data into file so that they can be written into database.

paste: (pasting files) to paste all cut files to one so that they can be loaded directly to database.

sort : (ordering a file) for use in toptalker to order the ip packets Option n compare according to string numeric value and option r sort in reverse order.

uniq: (locate repeated and non-repeated lines) to count number of packets for toptalker and remove duplicate entries.

head: (displaying the beginning of a file) to display first ten top talkers.

rm: (permanently deletes a file) to remove all the files when the user logs out for privacy purpose.

3. Data Storage

After processing, data is needed to be stored into MySQL database. It does so by using the following sql commands.

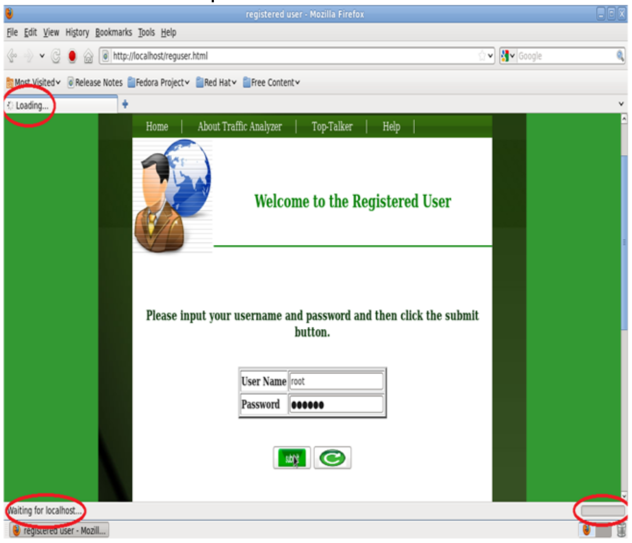
1. CREATE DATABASE database name //(to create the database)
2. USE database name //(to use the database)
3. CREATE TABLE table name (table attributes) //(to create any table in database)
4. LOAD DATA LOCAL INFILE filename INTO TABLE table name //(to load the data of a file into a table in the database)

Clearly database is first created then used in which table is created and then the file is loaded to that table. The database commands are stored in the same shell script that is called when the user signs in to use the network monitoring tool. Another shell script is written to delete all the packets information stored in files and MySQL database.

This script is invoked when the user logs out after using the services of the tool. Thus no outsider other than the administrator can access the captured packets information without signing in. This script is simply meant for security purpose.

Results

This is next and very essential part of monitoring tool. It is implemented using php and HTML.We can have a look in snapshots.

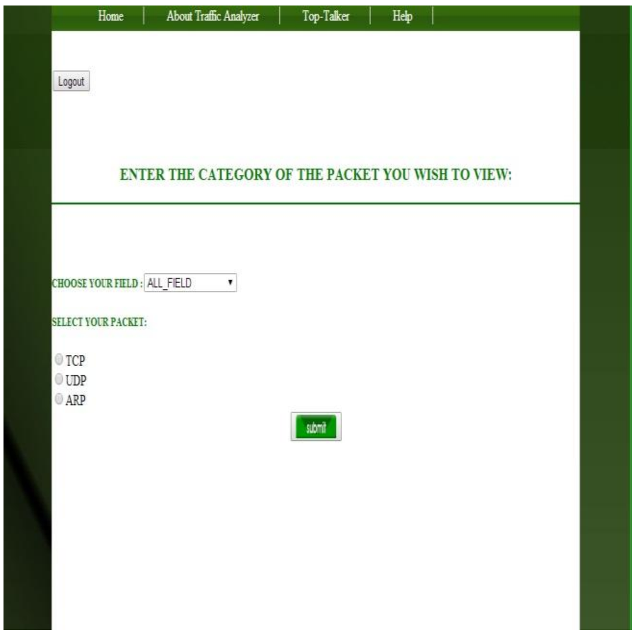

registered user - Mozilla Firefox
http://localhost/reguser.html
Loading...
Home | About Traffic Analyzer | Top-Talker | Help
Waiting for localhost...

Welcome to the Registered User

Please input your username and password and then click the submit button.

User Name	root
Password	●●●●●●

Figure 2: Capturing packets after login.



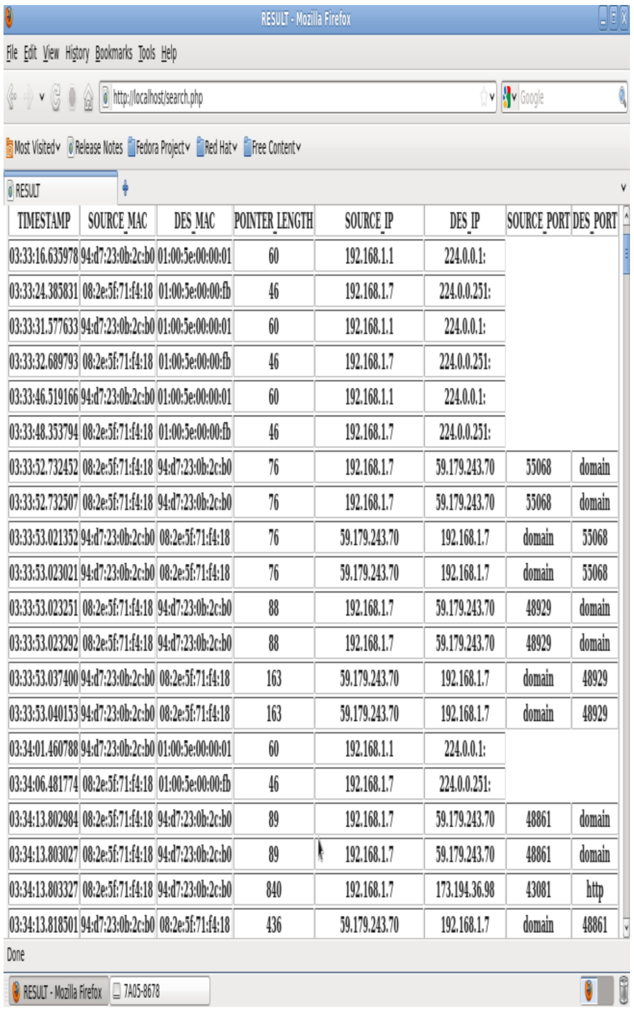
ENTER THE CATEGORY OF THE PACKET YOU WISH TO VIEW:

CHOOSE YOUR FIELD : ALL_FIELD

SELECT YOUR PACKET:

- TCP
- UDP
- ARP

Figure 3: Selecting type of packet.



TIMESTAMP	SOURCE_MAC	DES_MAC	POINTER_LENGTH	SOURCE_IP	DES_IP	SOURCE_PORT	DES_PORT
03:33:16.635978	94:d7:23:0b:2c:b0	01:00:5e:00:00:01	60	192.168.1.1	224.0.0.1:		
03:33:24.385831	08:2e:5f:71:f4:18	01:00:5e:00:00:fb	46	192.168.1.7	224.0.0.251:		
03:33:31.577633	94:d7:23:0b:2c:b0	01:00:5e:00:00:01	60	192.168.1.1	224.0.0.1:		
03:33:32.689793	08:2e:5f:71:f4:18	01:00:5e:00:00:fb	46	192.168.1.7	224.0.0.251:		
03:33:46.519166	94:d7:23:0b:2c:b0	01:00:5e:00:00:01	60	192.168.1.1	224.0.0.1:		
03:33:48.353794	08:2e:5f:71:f4:18	01:00:5e:00:00:fb	46	192.168.1.7	224.0.0.251:		
03:33:52.732452	08:2e:5f:71:f4:18	94:d7:23:0b:2c:b0	76	192.168.1.7	59.179.243.70	55068	domain
03:33:52.732507	08:2e:5f:71:f4:18	94:d7:23:0b:2c:b0	76	192.168.1.7	59.179.243.70	55068	domain
03:33:53.021352	94:d7:23:0b:2c:b0	08:2e:5f:71:f4:18	76	59.179.243.70	192.168.1.7	domain	55068
03:33:53.023021	94:d7:23:0b:2c:b0	08:2e:5f:71:f4:18	76	59.179.243.70	192.168.1.7	domain	55068
03:33:53.023251	08:2e:5f:71:f4:18	94:d7:23:0b:2c:b0	88	192.168.1.7	59.179.243.70	48929	domain
03:33:53.023292	08:2e:5f:71:f4:18	94:d7:23:0b:2c:b0	88	192.168.1.7	59.179.243.70	48929	domain
03:33:53.037400	94:d7:23:0b:2c:b0	08:2e:5f:71:f4:18	163	59.179.243.70	192.168.1.7	domain	48929
03:33:53.040153	94:d7:23:0b:2c:b0	08:2e:5f:71:f4:18	163	59.179.243.70	192.168.1.7	domain	48929
03:34:01.460788	94:d7:23:0b:2c:b0	01:00:5e:00:00:01	60	192.168.1.1	224.0.0.1:		
03:34:06.481774	08:2e:5f:71:f4:18	01:00:5e:00:00:fb	46	192.168.1.7	224.0.0.251:		
03:34:13.802984	08:2e:5f:71:f4:18	94:d7:23:0b:2c:b0	89	192.168.1.7	59.179.243.70	48861	domain
03:34:13.803027	08:2e:5f:71:f4:18	94:d7:23:0b:2c:b0	89	192.168.1.7	59.179.243.70	48861	domain
03:34:13.803327	08:2e:5f:71:f4:18	94:d7:23:0b:2c:b0	840	192.168.1.7	173.194.36.98	43081	http
03:34:13.818501	94:d7:23:0b:2c:b0	08:2e:5f:71:f4:18	436	59.179.243.70	192.168.1.7	domain	48861

Figure 4: Details of each packet as captured by tcp-dump (TCP only).

V. CONCLUSION AND FUTURE WORK

We have created a utility tool and can be easily installed on any application server and there it can be analyzed as on what time which packet was communicating with the server, what data has being transferred, whether the application server is not being hacked or misused, and which of the terminals is unable to communicate with server due to crash of the system or any fault on any system terminal. The scope also includes the help this project will provide to the users or the administrator. After being installed on any terminal a variety of checks can be done. To every illegal or any interrupted service occurs, alter is generated and is sent to the administrator to recover the faults. A complete search report is maintained and generated according to the search option.

In future work we need to extend the project so that it can decode the cipher and use that cipher to decode the data so that along with knowing infected IP using TOPTALKER we can also know what the infection is actually. Also telnet send data in plain text. But it is not in human readable form it still needs cipher to be decoded. In future we should be able to decode the data sent between two IPs in proper human readable format so that the centralized server has full power on network through this tool.

REFERENCES

[1]. http://network.spravcesite.net/subdom/network/ last retrieved on 14 March 2015

[2]. Tcp-dump tutorial and primer. [Online]. Available: http://www.danielmiessler.com/study/tcpdump/ last retrieved on 14th March 2015. [3] A-Z index of Bash Command Line for Linux. [Online]. Available: http://ss64.com/bash/ .

[3]. IT Business Edge. Understanding LAMP. [Online]. Available: http://www.serverwatch.com/tutorials/article.php/3567741/Understanding-LAMP.htm last retrieved on 26th February 2015.

[4]. W3schools. PHP 5 Tutorial. [Online]. Available: http://www.w3schools.com/PHP/ last retrieved on 17th February 2015.

[5]. The Apache Software Foundation. Apache HTTP Server Project. [Online] Available: http://httpd.apache.org/ last retrieved on 1st February 2014.

[6]. RS Shirbhate and PA Patil, Network Traffic Monitoring Using Intrusion Detection System in International Journal of Advanced Research in Computer Science and Software Engineering. Volume 2, Issue 1, January 2012.

[7]. Z. Zhao, W. Huangfu and L. Sun, NSSN: Network monitoring and packet sniffing tool for wireless networks in Wireless Communications and Mobile Computing Conference (IWCMC), 2012 8th International, pg. 537-542, 27 August 2012.

[8]. Hanaa Nafea,Kashif Kifayat,Qi Shi, Kashif Naseer Qureshi,Bob Askwith Efficient Non-Linear Covert Channel Detection in TCP Data Streams,10.1109/ACCESS.2019.2961609,IEEE Access,2019.