

Deep Malware Audit and Performance Evaluation Technologies for Network Deployments

Mr. D. D. Solomon Raj¹, Dr. Prasadu peddi², Dr. Devasish Pal³

Research Scholar¹, Dept of CSE, J.J.T. University, Rajasthan

Asst Professor², Dept of CSE, J.J.T. University, Rajasthan

Professor³, Dept of IT, Mufakham Jah Engineering College, Hyderabad

Abstract

Detection of malware and invalid traffic from the network environment is one of the key jobs with the cyber forensic administrator to keep the overall scenario very performance aware. Now days, the task of monitoring and administration has become a very important as well as difficult task due to huge amount of information is flowing in different transmission channels. In every organization it is very challenging job of the network administrators to analyze the traffic flowing in their network whether it is dealing with financial, military, education or social information. The network crackers are very active which are very curious to access the confidential data running inside the opponents' networks. At this instance, there is the need of very effective tools that can analyze the hacking or cracking attempts. Generally, the crackers analyze other's network and capture the information in their own records. This task is classically known as network sniffing in which repeatedly the network is analyzed for the information flowing in the network infrastructure. To perform this task, there is need to integrate enormous tools and techniques with the deep learning based tools which can detect and test the performance of web server with higher degree of performance. The process of software development integrates the tasks of rigorous testing on multiple parameters and dimensions so that the overall performance of the network suite can be highly effective. The process of network environment testing is mandatory for every type of suite whether it is local network software or cloud integrated web based application. Without proper testing, the network environment can behave abnormally or can crash with specific inputs that cause the failure of the software applications. In this manuscript, the assorted factors and technologies are specified for network forensic and audit.

Keywords: Malware Detection, Load Testing, Network Performance Testing

Introduction

The malware detection and load penetration testing is required in the network based applications so that the overall performance and effectiveness of the network [1, 2]. There are assorted tools and techniques with the integration of deep learning for network testing and audit [3, 4].

There are number of software products available in the technology market that provides the modules of network sniffer using which the system administrator can analyze the packets. Packet Capturing is the procedure of capturing and logging movement. The packet analyzer is also referred to as a network analyzer, protocol analysis tool or protocol analyzer, packet sniffer, Ethernet sniffer or simply a wireless sniffer. Such software is technically a software program that intercepts, seize and log the traffic passing through a network infrastructure. As information streams over the system, the sniffer catches every packet and, if required, translates the packet's crude information, demonstrating the qualities of different fields in the parcel, and investigates its substance consistent with the suitable RFC or different determinations.

Active and Passive Sniffing of Network Environment

Sniffing is a technique for fetching network information by capturing network packets. There are two types of packet sniffing in the networks: Active Sniffing and Passive Sniffing. In active sniffing, the packet sniffing tool or software send the requests over the network and then in response calculates the packets passing through the network. Passive sniffing does not rely on sending requests. This technique scans the network traffic without being detected on the network. It can be useful in places where networks are running critical systems like process control, radar systems, medical equipment or telecommunication, etc.

Features of Packet Tracing and Analysis Tools

There are number of applications and uses where the packet analyzers or sniffers can be used in a constructive way. Following is the list of the positive aspects of packets tracking tools :

- Analyze network problems
- Detect network intrusion attempts
- Detect network misuse by internal and external users
- Documenting regulatory compliance through logging all perimeter and endpoint traffic
- Gain information for effecting a network intrusion
- Isolate exploited systems
- Monitor WAN bandwidth utilization
- Monitor network usage (including internal and external users and systems)
- Monitor data-in-motion
- Monitor WAN and endpoint security status
- Gather and report network statistics
- Filter suspect content from network traffic
- Serve as primary data source for day-to-day network monitoring and management
- Spy on other network users and collect sensitive information such as login details or users cookies (depending on any content encryption methods that may be in use)

- Reverse engineer proprietary protocols used over the network
- Debug client/server communications
- Debug network protocol implementations
- Verify adds, moves and changes
- Verify internal control system effectiveness (firewalls, access control, Web filter, spam filter, proxy)

Libraries for Network Server Evaluation and Audit for Malware Detection

Wireshark: Wireshark is a free and open-source network packet analysis tool. This tool is utilized for network troubleshooting, dissection, programming and communications protocol research, development and training. Initially it was named as Ethereal, in May 2006 the venture was renamed Wireshark because of trademark issues. Wireshark is cross-platform, utilizing the Gtk+ widget toolkit within present discharges, and Qt in the improvement adaptation, to actualize its client interface, and utilizing pcap to catch parcels; it runs on different Unix-like working frameworks incorporating Gnu/linux, OS X, BSD, and Solaris, and on Microsoft Windows.

There is likewise a terminal-based (non-GUI) form called Tshark. Wireshark, and alternate projects distributed with this, for example, Tshark, are free programming, discharged under the terms of the GNU General Public License.

Wireshark has also won some industry awards and recognitions over the years

- eWeek
- Infoworld
- Insecure.org system security devices survey
- Sourceforge Project of the Month in August 2010
- McAfee SiteAdvisor
- Network Protocol Analysis Award
- VoIP Monitoring Award

Wireshark is specialized tool that automatically understanding the structure and format of different networking protocols. It can intelligently parse and show the fields, along with their description specified by assorted networking protocols. Wireshark makes use of pcap to capture the packets. This tool is able to capture packets on the types of networks that pcap supports.

Wireshark is having rich set of features including -

- Detailed as well as deep inspection of hundreds of protocols

- Live capturing of packets as well as offline investigation
- Cross-platform tool that can run on Windows, Linux, OS X, Solaris, FreeBSD, NetBSD, and many others without any specific configuration
- The captured network packets and data can be viewed via a GUI, or via the TTY-mode TShark utility
- VoIP Support and Analysis. VoIP calls in the captured traffic can be analyzed and detected.
- Capture files compressed with gzip can be decompressed on the fly
- Live data can be read from Ethernet, IEEE 802.11, PPP/HDLC, ATM, Bluetooth, USB, Token Ring, Frame Relay, FDDI, and others (depending on your platform)
- Decryption support for many protocols, including IPsec, ISAKMP, Kerberos, SNMPv3, SSL/TLS, WEP, and WPA/WPA2
- Coloring rules can be applied to the packet list for quick, intuitive analysis
- Output Export Feature to XML, PostScript, CSV or plain text
- Data can be captured Wired from a live network connection or can be read from a file of already-captured packets.
- Live data can be analyzed from a number of types of networks including Ethernet, IEEE 802.11, PPP, and loopback.
- The captured data can be edited or converted via command-line switches to the "editcap" tool.
- The refinement in the data display can be implemented using the display filter.
- Plug-ins can be implemented and developed for new protocols.
- Raw traffic related to USB can be captured easily.

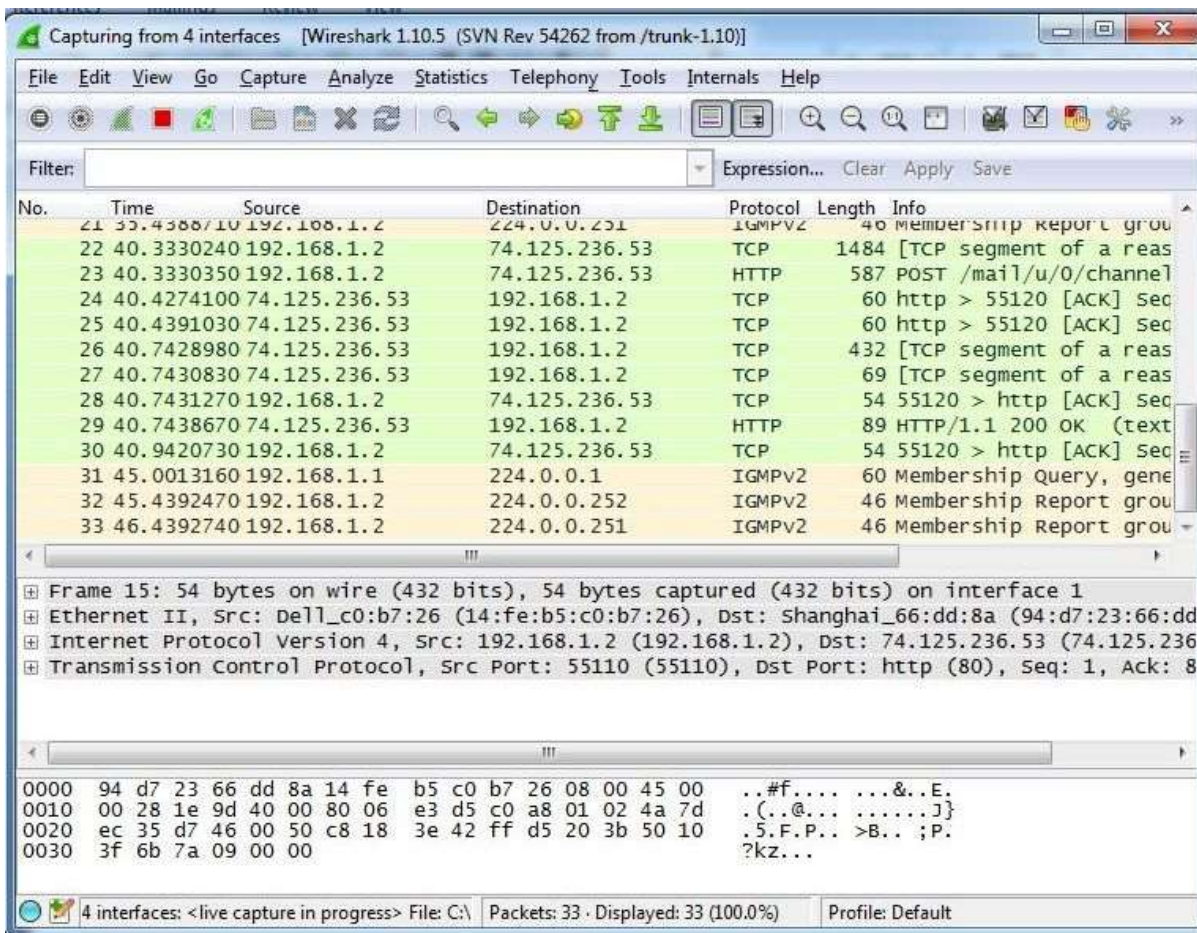


Figure 1: List of Packets and related information analyzed by Wireshark

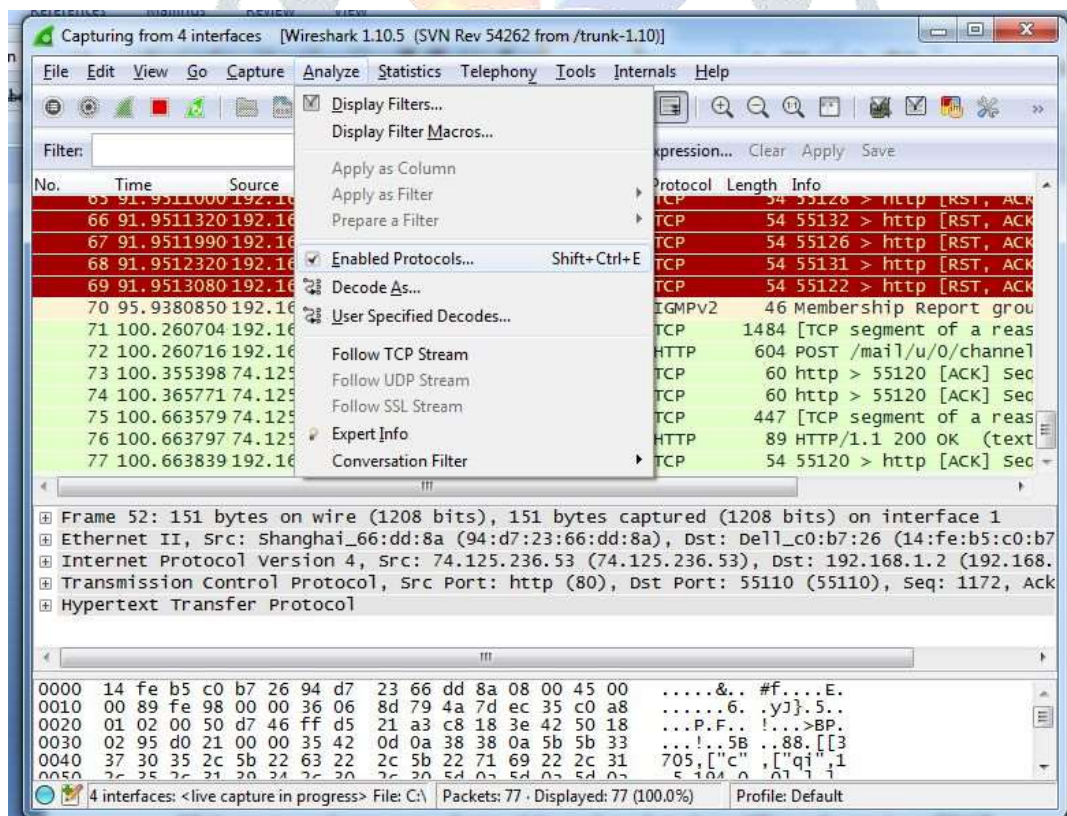


Figure 2: View Enabled Protocols for Analysis

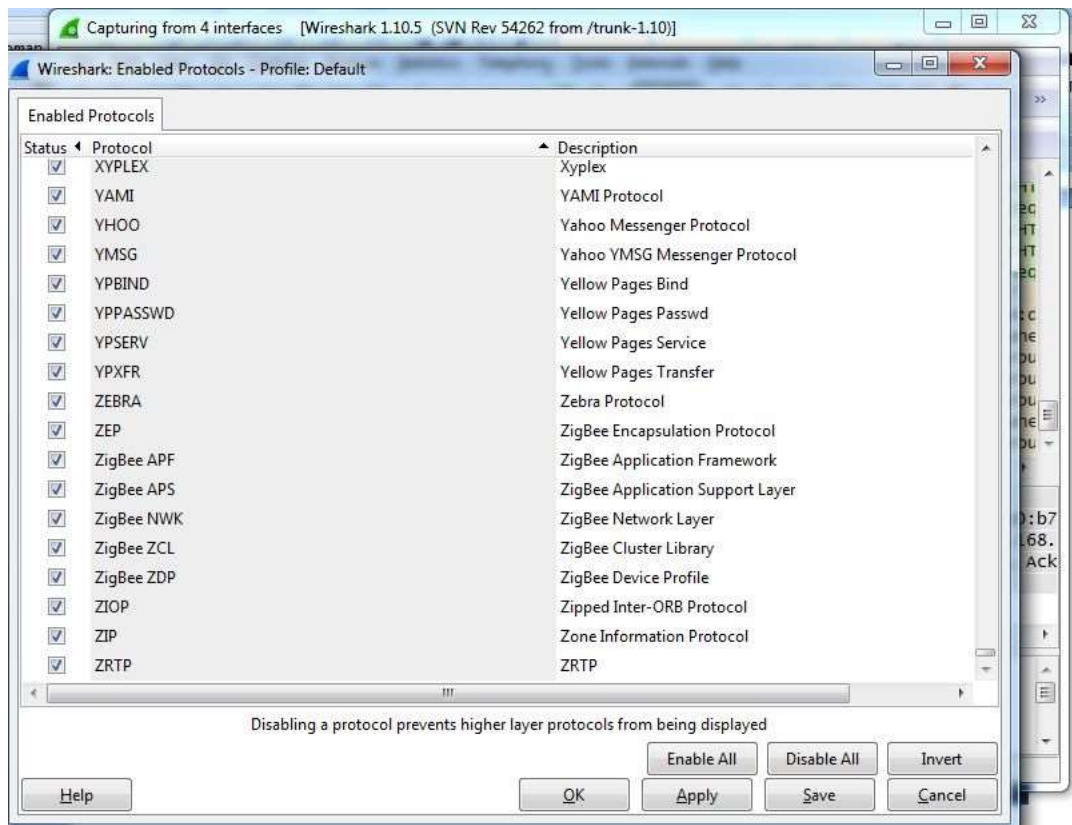


Figure 3: List of Protocols with the options displayed by Wireshark

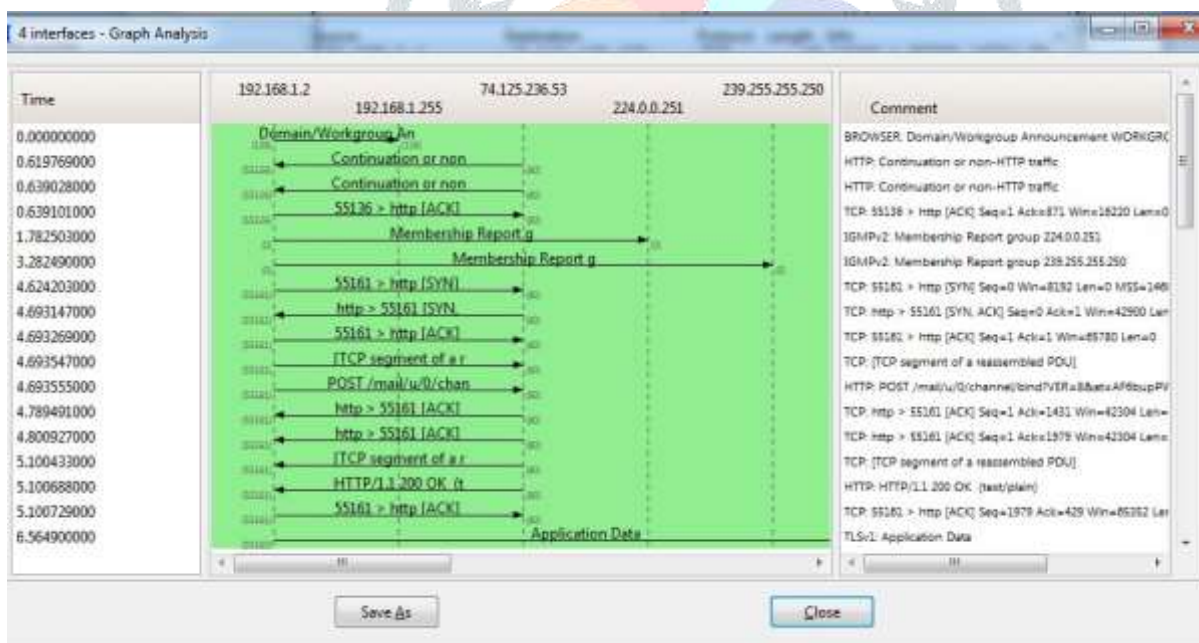


Figure 4: Analysis of all packets

High Performance IDS and IPS based Library of Snort: Snort is an open source tool written in C used as network intrusion prevention and detection system (IDS/IPS) developed by Sourcefire. It is having excellent combination of the benefits of signature, protocol, and anomaly-based inspection. The tool is associated

with millions of downloads and nearly 400,000 registered users, Snort has become the de facto standard for IPS.

Snort can implement protocol analysis and content investigation with number of other features including detection of a variety of attacks and probes such as buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts, and much more. Snort makes use of a flexible rules language to explain the traffic that it should collect or pass, as well as a detection engine that utilizes a modular plug-in architecture.

Snort tool can be configured in three different modes: sniffer, packet logger, and network intrusion detection. In sniffer mode, the tool read the network packets and displays them on the console. In packet logger mode, the tool implements the logging of the packets to the disk. In intrusion detection mode, the tool monitors the network traffic and analyzes it against a rule set defined by the user.

The main configuration file is /etc/snort/snort.conf. In this configuration file, the actual information of the network or system is specified that is under investigation. All values and parameters are commented in the file so that the changes can be done very easily.

Some of the extracts from the configuration file are :

```
# Setup the network addresses you are protecting
```

```
ipvar HOME_NET any
```

```
# Set up the external network addresses. Leave as "any" in most situations
```

```
ipvar EXTERNAL_NET any
```

```
# List of DNS servers on your network
```

```
ipvar DNS_SERVERS $HOME_NET
```

```
# List of SMTP servers on your network
```

```
ipvar SMTP_SERVERS $HOME_NET
```

```
# List of web servers on your network
```

```
ipvar HTTP_SERVERS $HOME_NET
```

```
portvar HTTP_PORTS
```

```
[36,80,81,82,83,84,85,86,87,88,89,90,311,383,555,591,593,631,801,808,818,901,972,1158,1220,1414,1533  
,1741,1830,2231,2301,2381,2809,3029,3037,3057,3128,3443,3702,4000,4343,4848,5117,5250,6080,6173,6  
988,7000,7001,7144,7145,7510,7770,7777,7779,8000,8008,8014,8028,8080,8081,8082,8085,8088,8090,81  
18,8123,8180,8181,8222,8243,8280,8300,8500,8509,8800,8888,8899,9000,9060,9080,9090,9091,9111,944  
3,9999,10000,11371,12601,15489,29991,33300,34412,34443,34444,41080,44449,50000,50002,51423,5333  
1,55252,55555,56712]
```

```
# List of ports you want to look for SHELLCODE on.  
portvar SHELLCODE_PORTS !80  
  
# List of ports you might see oracle attacks on  
portvar ORACLE_PORTS 1024:  
  
# List of ports you want to look for SSH connections on:  
portvar SSH_PORTS 22  
  
# List of ports you run ftp servers on  
portvar FTP_PORTS [21,2100,3535]  
  
# List of ports you run SIP servers on  
portvar GTP_PORTS [2123,2152,3386]  
  
# other variables, these should not be modified  
ipvar AIM_SERVERS  
[64.12.24.0/23,64.12.28.0/23,64.12.161.0/24,64.12.163.0/24,64.12.200.0/24,205.188.3.0/24,205.188.5.0/24,  
205.188.7.0/24,205.188.9.0/24,205.188.153.0/24,205.188.179.0/24,205.188.248.0/24]
```

Locust: Open Source Tool for Load Evaluation

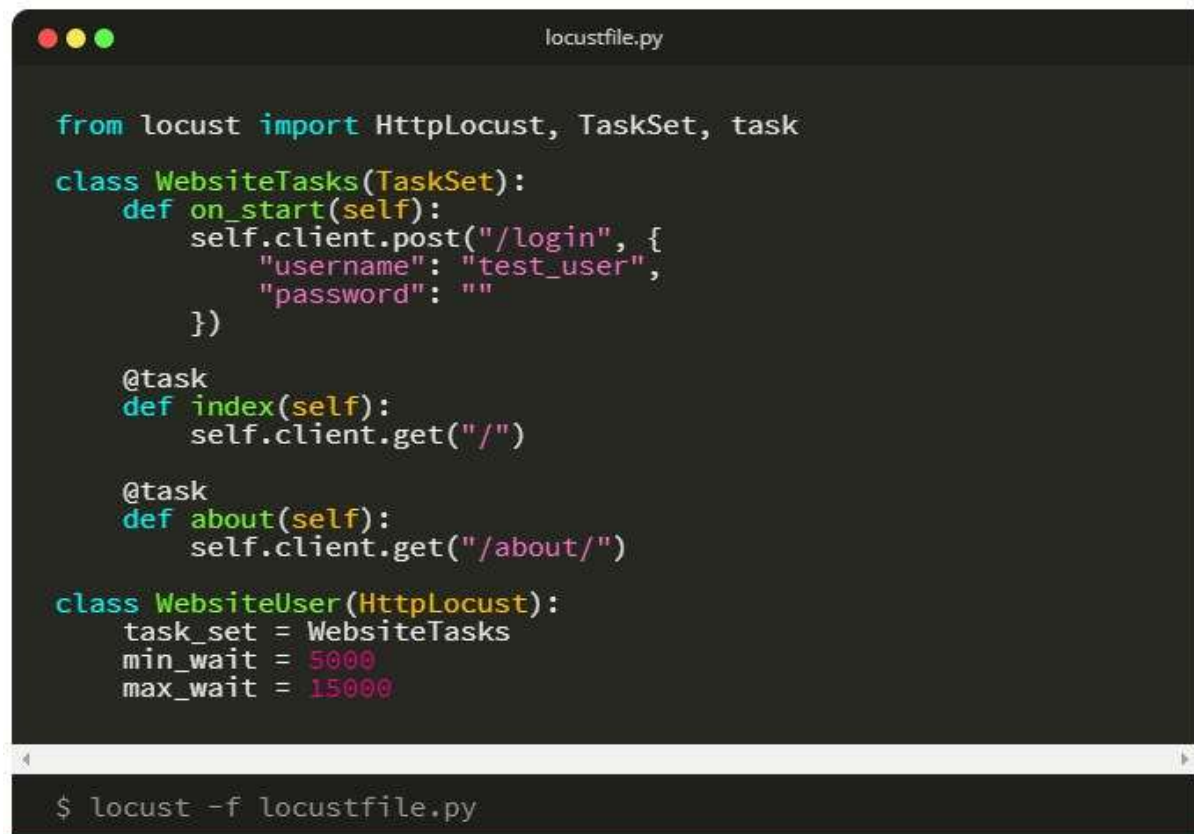
Locust is a Python based tool for evaluation of the behavior of a network server application with multiple concurrent users. Locust can generate the traffic that connects to the web application with the objectives to test the behavior with huge load. It is one of the high performance tools for the load testing under free and open source distribution. Locust defines the behavior and load of virtual traffic on the web application or server. It creates the enormous swarm based traffic so that the random load can be put on the web application or cloud server application [5].

For example, to test the behavior of a website on concurrent 1000 users, there is need to get 1000 users. Practically it is difficult to hire 1000 users to open a website and checking whether the website is working fine. Using Locust, any number of user traffic can be generated and finally the report is presented on the multiple parameters including number of requests, content size, failure attempts and many others.

Locust is having the plain code without any complex settings.

The easiest way to install Locust is from PyPI, using pip:

```
> pip install locustio
```

```
locustfile.py

from locust import HttpLocust, TaskSet, task

class WebsiteTasks(TaskSet):
    def on_start(self):
        self.client.post("/login", {
            "username": "test_user",
            "password": ""
        })

    @task
    def index(self):
        self.client.get("/")

    @task
    def about(self):
        self.client.get("/about/")

class WebsiteUser(HttpLocust):
    task_set = WebsiteTasks
    min_wait = 5000
    max_wait = 15000

$ locust -f locustfile.py
```

Figure 5: The View of locustfile.py and Instruction to Run

The code of Python locustfile.py can be directly executed in the command prompt to test the web application.

Apache JMeter: Tool for Web Server Audit for Multiple Evaluations

Apache JMeter is another Free and Open Source Tool that is used by the network based suite quality and testing experts for the load testing, performance testing and functional behavior of the web and cloud based network integrated applications. Initially, it was developed for the audit of web applications, but now days it is also used for core network functions and deep evaluation of the protocols [6].

Apache JMeter is able to test and present the deep audit of following

- File Transfer Protocol (FTP)
- Directory Services Lightweight Directory Access Protocol (LDAP)
- Database Connections
- TCP
- Mail Protocols
 - SMTP
 - POP

- IMAP
- Web Protocols
 - HTTP
 - HTTPS
- Java Objects
- Shell Scripts
- Native Commands

Apache JMeter is loaded with the full feature rich IDE for testing, debugging and test plan recordings so that the cavernous analytics of the web applications can be done. The detailed documentation and tutorials are available on the official website of Apache JMeter which can be used for assorted case studies.

Apache JMeter can be downloaded from [http:// JMeter.apache.org/download_Apache JMeter.cgi](http://JMeter.apache.org/download_Apache_JMeter.cgi) for different platforms and flavors as it is cross-platform software. From this link, the Apache JMeter is available in compressed format. Apache JMeter is extracted with the executable files which are available in the *bin* directory.

The GUI of Apache JMeter is having many features and options for creating and evaluating the test plans.

There is no specific naming convention or compulsion regarding the title of test plan. The testing engineer or audit expert can specify the name or title as per their own convenience.

The thread group is used so that the huge enormous users or traffic are required for the testing of application that integrates the threads.

A number of users are created virtually without physical presence so that the web application can be tested and thereby the cumulative performance can be investigated by the webmasters and web administrators.

The option of HTTP Request is selected because in case of web application testing, the protocol of HTTP works to present the outcome on web browser or HTTP client. The HTTP client opens the web page or URL only if that particular link is responding as without any issues [7].

The web application under testing can be specified in the suite of Apache JMeter in terms of official link or IP address so that the direct connectivity with the web server can be evaluated for further investigations [8].

The option to select “View Results Tree” enable the web administrator to visualize whether the load testing parameters and factors are successful with the association of assorted as well as enormous load on the web application.

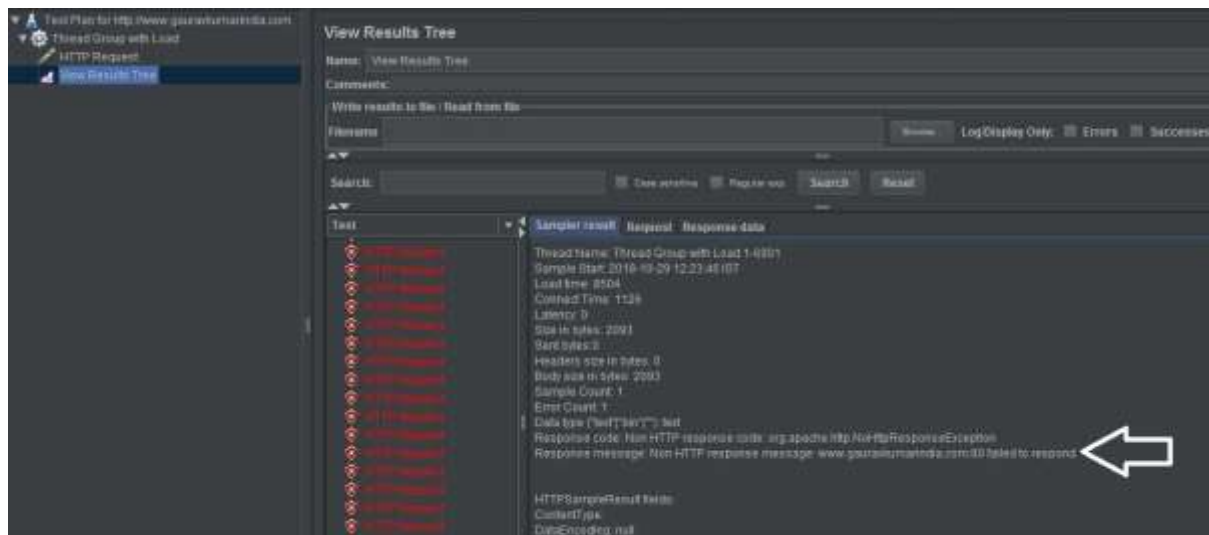


Figure 6: Analysis of Logs and Response from Server

From the analysis of logs and response outcome from the server, it can be checked whether the request to the web application was successful or not. The response code presents the cumulative outcome from the server from the initial point of contact by the client.

HTTP Unbearable Load King (HULK) Automation for Load Testing

Python is one of the powerful and multi-featured programming languages that is used for multiple applications including cloud computing, cyber security, digital forensics, Internet of Things (IoT), Fog Computing and many others [9].

Generally, the penetration testers are dependent on the third party tools and network applications for digital forensic and audit of the network applications. This approach of using third party software is not reliable and secured always.

The implementations of different testing strategies can be done using core programming scripts written in Python. As there are many tools and libraries for load testing, stress testing and performance testing but the Python Programs can be directly used for these implementations. Using unreliable tools for penetration testing and load testing can be dangerous because own system information can be shared to the other locations and it will not be secured.

HTTP Unbearable Load King (HULK) is a type of DDoS Attack on the server but it can be used for the testing of web application. Using HULK implementation, the performance of web application can be tested to check whether the web application is able to handle the DDoS or HULK based traffic [10, 11]. If the web application is getting down or having more delay in opening, the preventive measures can be taken. HULK script generates the unbearable huge load to the web application and by this way the web administrator can check whether the web application is working fine after implementation of HULK [12, 13].

The webmasters and network administrators can analyze the error log files and access log files to check the attempts and performance of web application after implementation of HULK or similar implementations [14].

Conclusion and Scope of Future Directions

In the domain of security towards the web based applications, there exist enormous perspectives which are required to be addressed and worked out. Following are few of the aspects and research points which can be solved by the researchers, corporate organizations and the academicians in the segment of network audit, evaluation and forensic based audit of network modules

- Biometric integrated Access for Secured Web Portals
- Development of Trust Architecture for Secured Network and Cloud Applications
- Integrity Aware Vulnerability Recognition in Secured Network Applications
- Deep Learning based Identification of Suspects in the Access of Web Applications

The packet tracing or sniffers are also used by the hacking community to analyze the data packets but as far as constructive purpose is there, such tools are very useful for the network administrators. The network administrators and engineers can analyze the type of packets flowing in their network infrastructure, bandwidth issues, port and protocols using such tools.

References

- [1] Xue Y, Meng G, Liu Y, Tan TH, Chen H, Sun J, Zhang J. Auditing Anti-Malware Tools by Evolving Android Malware and Dynamic Loading Technique. *IEEE Trans. Information Forensics and Security*. 2017 Jul 1;12(7):1529-44.
- [2] Basu A, Aydin A. Predicting uniaxial compressive strength by point load test: significance of cone penetration. *Rock Mechanics and Rock Engineering*. 2006 Nov 1;39(5):483-90.
- [3] Kolosnjaji B, Demontis A, Biggio B, Maiorca D, Giacinto G, Eckert C, Roli F. Adversarial Malware Binaries: Evading Deep Learning for Malware Detection in Executables. *arXiv preprint arXiv:1803.04173*. 2018 Mar 12.

- [4] Mohan AK, Sethumadhavan M. Wireless Security Auditing: Attack Vectors and Mitigation Strategies. *Procedia Computer Science*. 2017 Dec 31;115:674-82.
- [5] Düllmann TF, Heinrich R, van Hoorn A, Pitakrat T, Walter J, Willnecker F. CASPA: A Platform for Comparability of Architecture-based Software Performance Engineering Approaches.
- [6] Halili EH. *Apache JMeter: A practical beginner's guide to automated testing and performance measurement for your websites*. Packt Publishing Ltd; 2008 Jun 27.
- [7] Rawal BS, Karne RK, Wijesinha AL. Mini Web server clusters for HTTP request splitting. In *High Performance Computing and Communications (HPCC), 2011 IEEE 13th International Conference on 2011 Sep 2* (pp. 94-100). IEEE.
- [8] Srisuresh P, Holdrege M. IP network address translator (NAT) terminology and considerations. 1999.
- [9] Bonomi F, Milito R, Zhu J, Addepalli S. Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing 2012 Aug 17* (pp. 13-16). ACM.
- [10] Mirkovic J, Reiher P. A taxonomy of DDoS attack and DDoS defense mechanisms. *ACM SIGCOMM Computer Communication Review*. 2004 Apr 1;34(2):39-53.
- [11] Braga R, Mota E, Passito A. Lightweight DDoS flooding attack detection using NOX/OpenFlow. In *Local Computer Networks (LCN), 2010 IEEE 35th Conference on 2010 Oct 10* (pp. 408-415). IEEE.
- [12] Jin S, Yeung DS. A covariance analysis model for DDoS attack detection. In *Communications, 2004 IEEE International Conference on 2004 Jun 20* (Vol. 4, pp. 1882-1886). IEEE.
- [13] Li L, Lee G. DDoS attack detection and wavelets. *Telecommunication Systems*. 2005 Mar 1;28(3-4):435-51.
- [14] Bhuyan MH, Bhattacharyya DK, Kalita JK. An empirical evaluation of information metrics for low-rate and high-rate DDoS attack detection. *Pattern Recognition Letters*. 2015 Jan 1;51:1-7.