LP IV Lab Manual 24-25 Sem I

SITRC

Sandip Institute of Technology & Reasearch Centre
Nashik

LABORATORY MANUAL

Laboratory Practice IV

Department of Computer Engineering

BE (2019 Course)

SANDIP FOUNDATIONS

SANDIP INSTITUTE OF TECHNOLOGY & RESEARCH CENTRE, NASHIK

Department of Computer Engineering

LABORATORY MANUAL

AY: 2024-25

Laboratory Practice IV

BE Computer
EngineeringSemester –I
Subject Code - 410247

TEACHING SCHEME	CREDIT	EXAMINATION
Practical: 2 Hrs / Week	01	TW: 50 Marks

Prepared By:

Prof.Pradeep A Patil
(Assistant Professor, Department of Computer Engineering)

Prof.Dr.Ankita V Karale
(HOD, Department of Computer Engineering)

CONTENTS

Cyber Security Manual

Answers **(5)**	Coding Efficiency **(5)**	Viva **(5)**	Timely Completion **(5)**	Total **(20)**	Dated Sign of Subject Teacher

Expected Date of Completion:-------------------------

Actual Date of Completion:----------------------------

Experiment No: Group A-1

Problem Definition:

Write a program for Tracking Emails and Investigating Email Crimes. i.e. Write a program to analyze e–mail header

1.1 Prerequisite:

Application Layer Protocols

1.2 Learning Objective:

1. To understand how Mails are transferred from Sender to Receiver.
2. To Understand Email related Parameter.

1.3 Theory:

1.3.1 Introduction

Analysis of email is especially important not just because email may be used to communicate about things that we might be interested in for an investigation, but because it is a comparatively permanent and public record of those communications. In the case of a phone call, there is only the record that a call took place; in a spoken conversation, there may be no record at all. Conventional mail can be virtually untraceable, and paper documents are easily destroyed. Email, however, is unique; when a message is sent, the entire message is stored for both the sender and the receiver, and records of the mail being sent are stored on dozens of servers that the message passes through before arriving at its destination. There are a number of ways to analyze email, including: data mining techniques, which may be applied to large or small data sets; straightforward searching of a user's email for certain content; and in-depth analysis of an individual email's lineage.

E-mail system comprises of various hardware and software components that include sender’s client and server computers and receiver’s client and server computers with required software and services installed on each. Besides these, it uses various systems and services of the Internet. The sending and receiving servers are always connected to the Internet but the sender's and receiver's client connects to the Internet as and when required.

An e-mail communication between a sender '*Alice'* having e-mail address '*alice@a.com*' and recipient '*Bob'* having e-mail address '*bob@b.com*' is shown in figure 1.'*Alice'* composes an e-mail message on her computer called client for '*Bob'* and sends it to her sending server 'smtp.a.org' using *SMTP* protocol. Sending server performs a lookup for the mail exchange record of receiving server '*b.org*' through Domain Name System (*DNS)* protocol on *DNS* server '*dns.b.org*'. The *DNS* server responds with the highest priority mail exchange server '*mx.b.org'* for the domain 'b.org'. Sending server establishes *SMTP* connection with the receiving server and delivers the e-mail message to the mailbox of '*Bob'* on the receiving server. '*Bob'* downloads the message from his mailbox on receiving server to local mailbox on his client computer using *POP3* or *IMAP* protocols. Optionally, '*Bob'* can also read the message stored in his server mailbox without downloading it to the local mailbox by using a Webmail program.

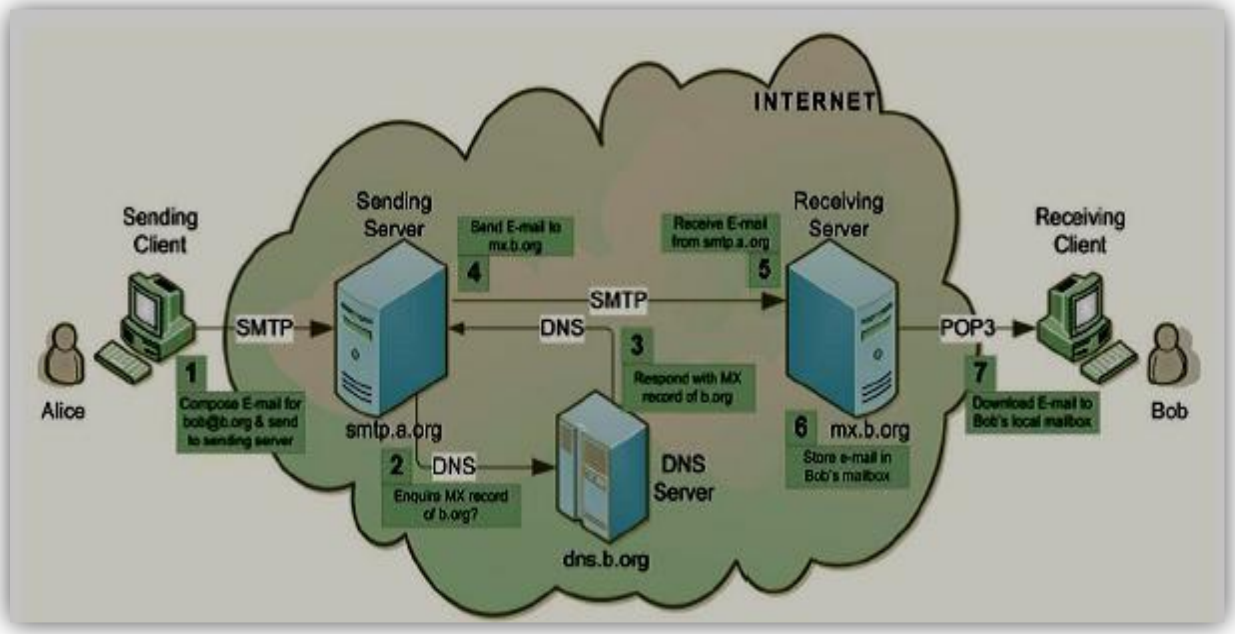
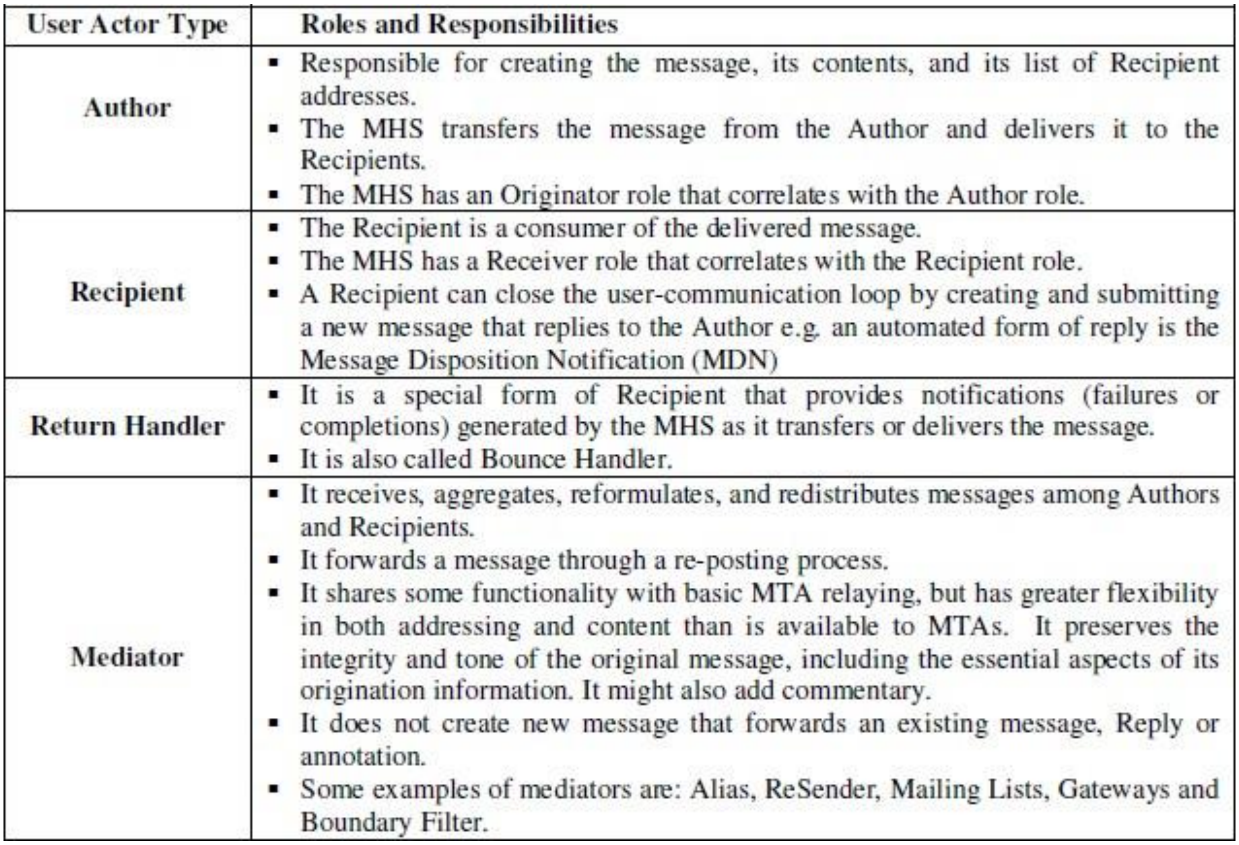

INTERNET
Sending Client
Sending Server
Send E-mail to mx.b.org
4
Receive E-mail from smtp.a.org
5
Receiving Server
Receiving Client
SMTP
SMTP
DNS
3
Respond with MX record of b.org
POP3
7
Download E-mail to Bob's local mailbox
Alice
1
Compose E-mail for bob@b.org & send to sending server
smtp.a.org
2
DNS
Enquire MX record of b.org?
DNS Server
dns.b.org
6
mx.b.org
Store e-mail in Bob's mailbox
Bob

Figure 1*: E-mail communication between a sender 'Alice' and recipient 'Bob'*

1.3.2 E-MAIL ACTORS, ROLES AND RESPONSIBILITIES

E-mail is a highly distributed service involving several actors that play different roles to accomplish end-to-end mail exchange. These actors fall under "User Actors", "Message Handling Service (*MHS*) Actors" and "Administrative Management Domain (*ADMD*) Actors" groups.

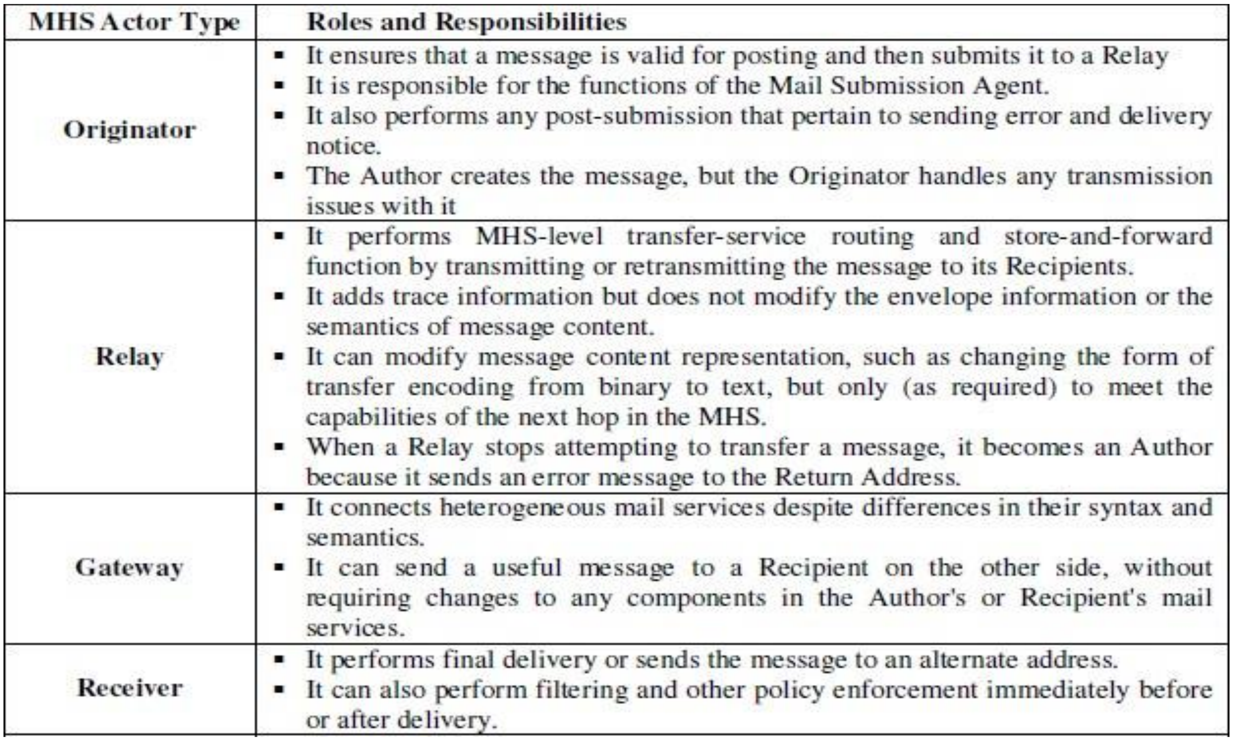
User Actors are people, organizations or processes that serve as sources or sinks of messages. They can generate, modify or look at the whole message. User Actors can be of following four types (Table 1):



User Actor Type	Roles and Responsibilities
Author	▪ Responsible for creating the message, its contents, and its list of Recipient addresses. ▪ The MHS transfers the message from the Author and delivers it to the Recipients. ▪ The MHS has an Originator role that correlates with the Author role.
Recipient	▪ The Recipient is a consumer of the delivered message. ▪ The MHS has a Receiver role that correlates with the Recipient role. ▪ A Recipient can close the user-communication loop by creating and submitting a new message that replies to the Author e.g. an automated form of reply is the Message Disposition Notification (MDN)
Return Handler	▪ It is a special form of Recipient that provides notifications (failures or completions) generated by the MHS as it transfers or delivers the message. ▪ It is also called Bounce Handler.
Mediator	▪ It receives, aggregates, reformulates, and redistributes messages among Authors and Recipients. ▪ It forwards a message through a re-posting process. ▪ It shares some functionality with basic MTA relaying, but has greater flexibility in both addressing and content than is available to MTAs. It preserves the integrity and tone of the original message, including the essential aspects of its origination information. It might also add commentary. ▪ It does not create new message that forwards an existing message, Reply or annotation. ▪ Some examples of mediators are: Alias, ReSender, Mailing Lists, Gateways and Boundary Filter.

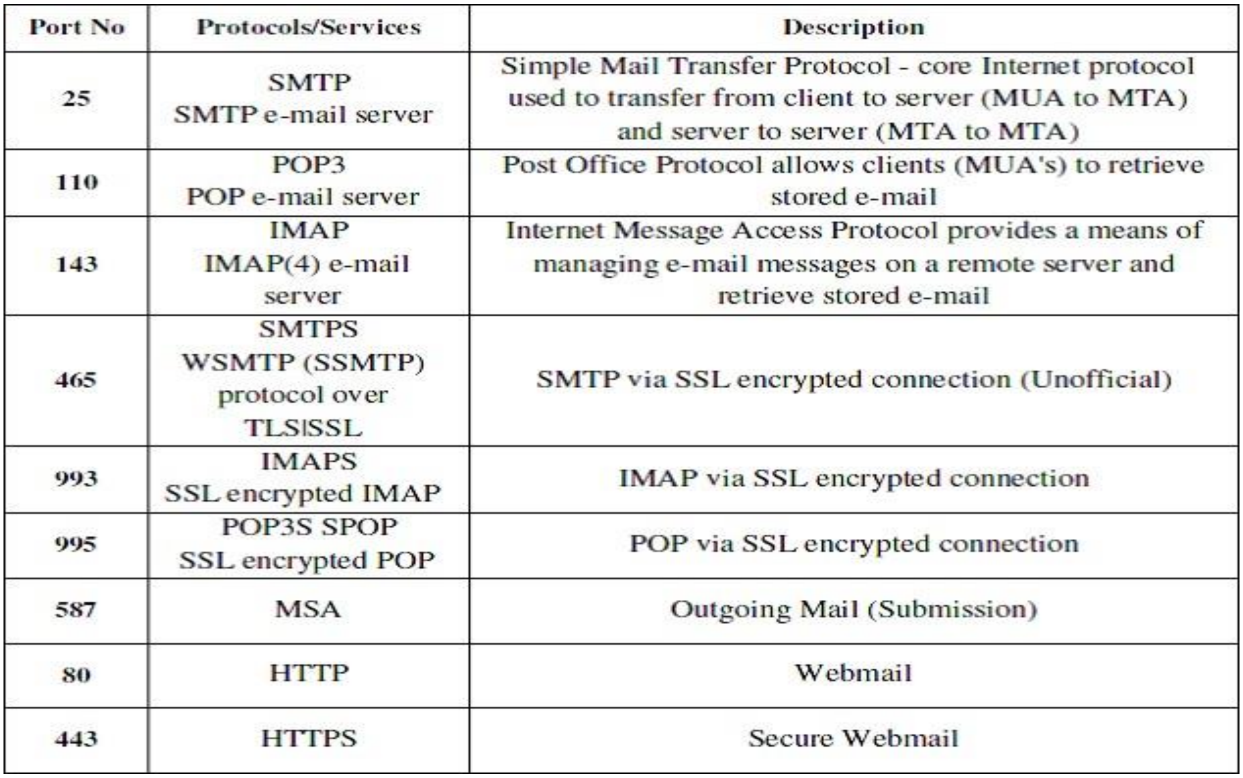
All types of Mediator user actors set HELO/EHLO, ENVID, RcptTo and Received fields. Alias actors also typically change To/CC/BCC and MailFrom fields. Identities relevant to ReSender are: From, Reply-To, Sender, To/CC/BCC, Resent-From, Resent-Sender, Resent-To/CC/BCC and MailFrom fields. Identities relevant to Mailing List processor are: List-Id, List-*, From, Reply-To, Sender, To/CC and MailFrom fields. Identities relevant to Gateways are: From, Reply-To, Sender, To/CC/BCC and MailFrom fileds.

Message Handling Service (MHS) Actors are responsible for end-to-end transfer of messages. These Actors can generate, modify or look at only transfer data in the message. *MHS* Actors can be of following four types (Table 2):



MHS Actor Type	Roles and Responsibilities
Originator	▪ It ensures that a message is valid for posting and then submits it to a Relay ▪ It is responsible for the functions of the Mail Submission Agent. ▪ It also performs any post-submission that pertain to sending error and delivery notice. ▪ The Author creates the message, but the Originator handles any transmission issues with it
Relay	▪ It performs MHS-level transfer-service routing and store-and-forward function by transmitting or retransmitting the message to its Recipients. ▪ It adds trace information but does not modify the envelope information or the semantics of message content. ▪ It can modify message content representation, such as changing the form of transfer encoding from binary to text, but only (as required) to meet the capabilities of the next hop in the MHS. ▪ When a Relay stops attempting to transfer a message, it becomes an Author because it sends an error message to the Return Address.
Gateway	▪ It connects heterogeneous mail services despite differences in their syntax and semantics. ▪ It can send a useful message to a Recipient on the other side, without requiring changes to any components in the Author's or Recipient's mail services.
Receiver	▪ It performs final delivery or sends the message to an alternate address. ▪ It can also perform filtering and other policy enforcement immediately before or after delivery.

For networks, a port means an endpoint to a logical connection. The port number identifies what type (application/service offered) of port it is. The commonly used default port numbers used in e-mail are shown in Table 3. A complete list of default port numbering assignment is given in



Port No	Protocols/Services	Description
25	SMTP SMTP e-mail server	Simple Mail Transfer Protocol - core Internet protocol used to transfer from client to server (MUA to MTA) and server to server (MTA to MTA)
110	POP3 POP e-mail server	Post Office Protocol allows clients (MUA's) to retrieve stored e-mail
143	IMAP IMAP(4) e-mail server	Internet Message Access Protocol provides a means of managing e-mail messages on a remote server and retrieve stored e-mail
465	SMTPS WSMTP (SSMTP) protocol over TLS|SSL	SMTP via SSL encrypted connection (Unofficial)
993	IMAPS SSL encrypted IMAP	IMAP via SSL encrypted connection
995	POP3S SPOP SSL encrypted POP	POP via SSL encrypted connection
587	MSA	Outgoing Mail (Submission)
80	HTTP	Webmail
443	HTTPS	Secure Webmail

1.3.3 Analyzing an Individual Email

Although webmail will feature prominently in this section, the analysis of a particular email's lineage is much broader and can be applied to any email. A simple view of the path of an email from a sender to a client is presented in Figure 2. The email originates from the sender, whether from a local email client or a webmail application. When the email is sent, it is first sent to a Simple Mail Transfer Protocol (SMTP) server. That server forwards it to other SMTP servers until it finally reaches the destination server. On reaching its destination, the email is sent to a Post Office Protocol (POP) server, or any number of similar mail-delivery servers (IMAP is another, and webmail services may use their own servers for this purpose). The receiving client then connects to that server, retrieves the message, and allows the recipient to read it.

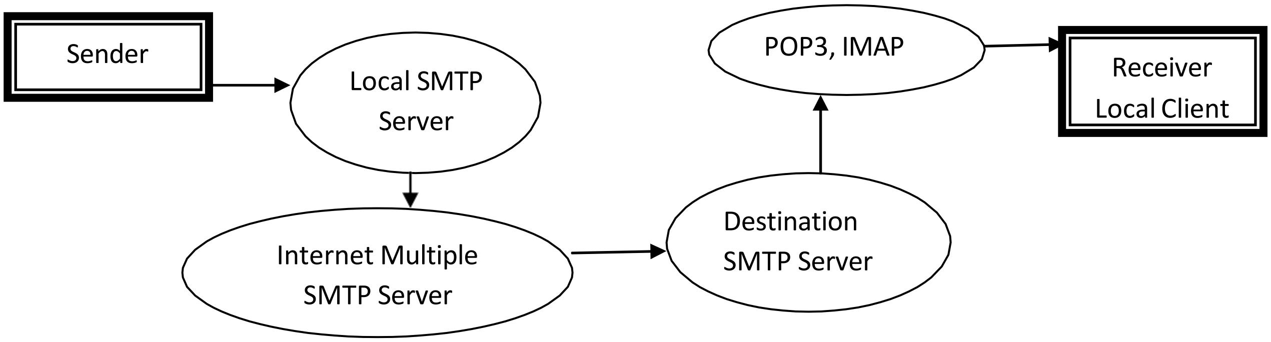


***Figure 2:** Path of an email from a sender to a client*

When the email is sent and when it is received, those respective servers add their own information to the email's header, and most likely log the action. Access to those logs may be required for much analysis, but specifics are outside of the scope of this paper. Considerable information can be gleaned from the header alone.

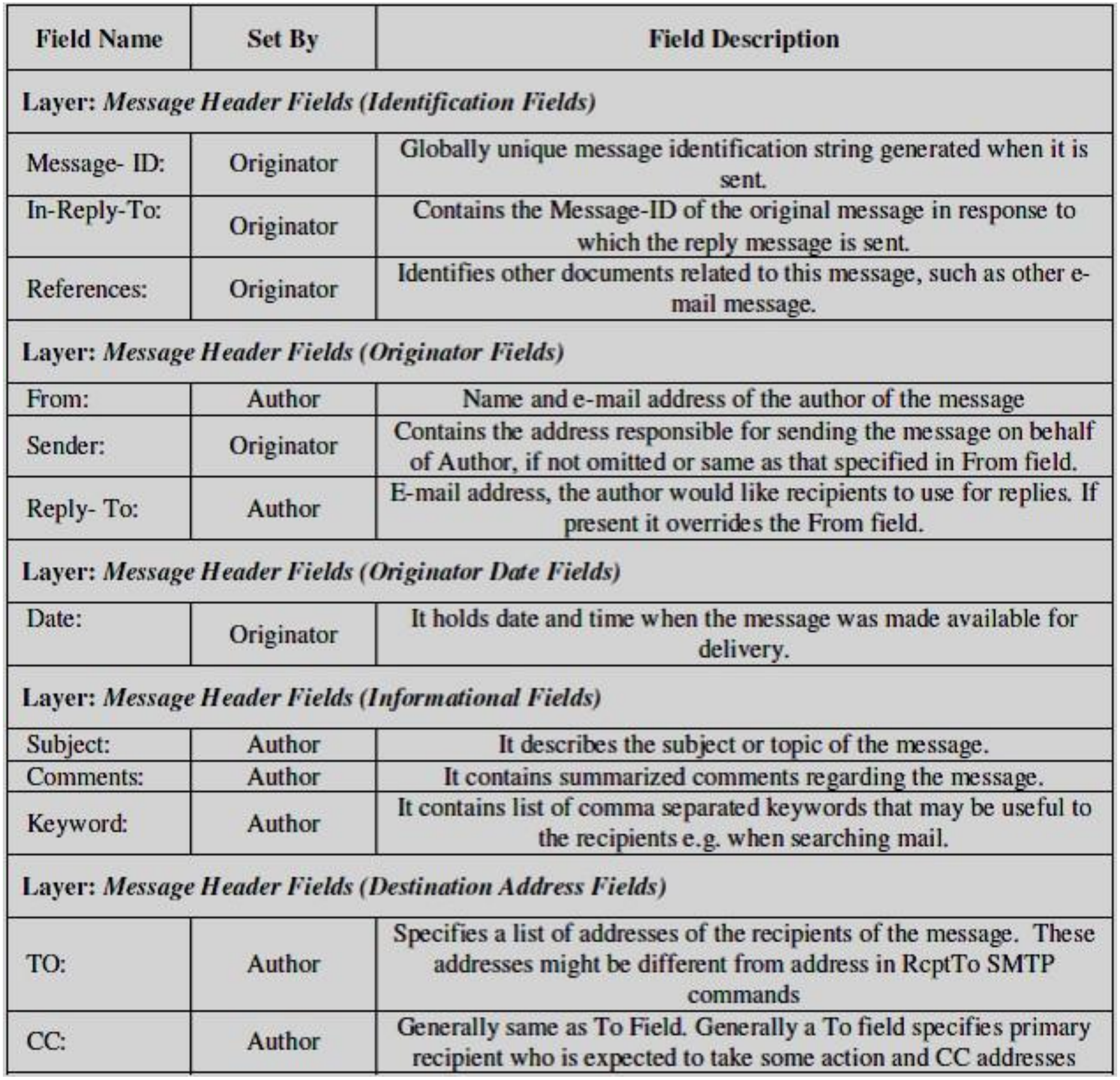
Suppose Moses, with the address moses@nmt.edu, sends an email from his office on the New Mexico Tech campus to his similarly named friend, with the email address thenewmoses@gmail.com. The subject of this email is "Snakes," and the content "Fish."

Below is the entire theoretical email, including all headers.

```
From:
moses@nmt.edu
Subject: Snakes
Date: September 25, 2021 9:35:29 PM
MDT To: thenewmoses@gmail.com
X-Gmail-Received:
ca493ed685a8e9ae77165ab2ce345127e5b310b4 Delivered-
To: thenewmoses@gmail.com
Received: by 10.90.33.15 with SMTP id g15cs279684agg; Mon, 25 Sep
2021 20:35:32 0700 (PDT)
Received: by 10.35.113.12 with SMTP id q12mr526602pym; Mon, 25 Sep 2021
20:35:32 -0700 (PDT)
Received: from mailhost.nmt.edu (mailhost.NMT.EDU
[129.138.4.52]) by mx.gmail.com with ESMTP id 36si2059018nza.
2021.09.25.20.35.32; Mon, 25 Sep 2021 20:35:32 -0700 (PDT)
Received: from localhost (localhost.localdomain [127.0.0.1]) by
localhost.localdomain (Postfix) with ESMTP id 09FF4436164 for
<thenewmoses@gmail.com>; Mon, 25 Sep 2021 21:35:32 -0600
(MDT) Received: from mailhost.nmt.edu ([127.0.0.1]) by localhost
(mailhost.nmt.edu [127.0.0.1]) (amavisd-new, port 10024) with
ESMTP id 11225-05 for <thenewmoses@gmail.com>; Mon, 25 Sep 2021
21:35:30 -0600 (MDT)
Received: from [192.168.1.2] (cs-fitch017.nmt.edu [129.138.21.110])
by mailhost.nmt.edu (Postfix) with ESMTP id 6FD4B436030 for
<thenewmoses@gmail.com>; Mon, 25 Sep 2021 21:35:30 -0600
(MDT) Return-Path: <moses@nmt.edu>
Received-Spf: pass (gmail.com: best guess record for
domain of moses@nmt.edu designates 129.138.4.52 as
permitted sender) Mime-Version: 1.0 (Apple Message
framework v752.2)
Content-Transfer-Encoding: 7bit
Message-Id: <77E313EF-271F-4AD0-A8D3-81263BF7B083@nmt.edu>
```

```
Content-Type: text/plain; charset=US-ASCII;
format=flowed X-Mailer: Apple Mail (2.752.2)
X-Virus-Scanned: by amavisd-new-2.3.1 (20050509) (RHEL AS) at
nmt.edu Fish
```

E-MAIL IDENTITIES:



Field Name	Set By	Field Description
Layer: ***Message Header Fields (Identification Fields)***		
Message- ID:	Originator	Globally unique message identification string generated when it is sent.
In-Reply-To:	Originator	Contains the Message-ID of the original message in response to which the reply message is sent.
References:	Originator	Identifies other documents related to this message, such as other e-mail message.
Layer: ***Message Header Fields (Originator Fields)***		
From:	Author	Name and e-mail address of the author of the message
Sender:	Originator	Contains the address responsible for sending the message on behalf of Author, if not omitted or same as that specified in From field.
Reply- To:	Author	E-mail address, the author would like recipients to use for replies. If present it overrides the From field.
Layer: ***Message Header Fields (Originator Date Fields)***		
Date:	Originator	It holds date and time when the message was made available for delivery.
Layer: ***Message Header Fields (Informational Fields)***		
Subject:	Author	It describes the subject or topic of the message.
Comments:	Author	It contains summarized comments regarding the message.
Keyword:	Author	It contains list of comma separated keywords that may be useful to the recipients e.g. when searching mail.
Layer: ***Message Header Fields (Destination Address Fields)***		
TO:	Author	Specifies a list of addresses of the recipients of the message. These addresses might be different from address in RcptTo SMTP commands
CC:	Author	Generally same as To Field. Generally a To field specifies primary recipient who is expected to take some action and CC addresses

1.4 Execution Steps

- Open the Email which you want to analyze header
- Click on the right side three vertical dot(more) and select show original.
- New tab will be open then copy header information which you want to analyze.
- Open **https://www.whatismyip.com/email-header-analyzer** website.
- Copy the header information and click on analyze button.
- Then You will see the header analysis on screen.

1.5 Assignment Question:

1. Why to Analyze Email Header?
2. What Fields are analyzed during Email Analysis Header?
3. Which Readymade Tools are Available for Analyzing E-Mail Header?
4. Explain Email Architecture in Detail?
5. What is POP3, IMAP, SMTP, and MIME?

1.6 Conclusion:

E-mail is a widely used and highly distributed application involving several actors that play Different roles. These actors include hardware and software components, services and protocols which provide interoperability between its users and among the components along the path of transfer. Cybercriminals forge e-mail headers or send it anonymously for illegitimate purposes which lead to several crimes and thus make e-mail forensic investigation crucial.

Answers **(5)**	Coding Efficiency **(5)**	Viva **(5)**	Timely Completion **(5)**	Total **(20)**	Dated Sign of Subject Teacher

Expected Date of Completion:-------------------------

Actual Date of Completion:----------------------------

Assignment Group A-2

Problem Definition:
Implement a program to generate and verify CAPTCHA image.

2.1 Prerequisite:

Basics of PYTHON

2.2 Learning Objectives:

1. Understand the use of CAPTCHA Image.
2. Generation and Verification of it.

2.3New Concepts:

1. CAPTCHA generation
2. Functions used like RANDOM

2.4 Theory

2.4.1 Introduction

A **CAPTCHA** (an acronym for "**C**ompletely **A**utomated **P**ublic **T**uring test to tell **C**omputers and **H**umans **A**part") is a type of challenge-response test used in computing to determine whether or not the user is human. The most common type of CAPTCHA was first invented by Mark D. Lillibridge, Martin Abadi, Krishna Bharat and Andrei Z. Broder. This form of CAPTCHA requires that the user type the letters of a distorted image, sometimes with the addition of an obscured sequence of letters or digits that appears on the screen. Because the test is administered by a computer, in contrast to the standard Turing test that is administered by a human, a CAPTCHA is sometimes described as a reverse Turing test. Actually CAPTCHA is used as a simple puzzle hurdle, which restricts various automated programs to sign-up E-mail accounts, cracking passwords, spam sending, privacy violation

etc. This CAPTCHA actually challenges a particular automated program, which is trying to access some private zone. So, CAPTCHA helps in preventing access of personal mail accounts by some un-authorized automated spamming programs

2.4.2 Characteristics:-

CAPTCHAs are by definition fully automated, requiring little human maintenance or intervention to administer. This has obvious benefits in cost and reliability.

By definition, the algorithm used to create the CAPTCHA must be made public, though it may be covered by a patent. This is done to demonstrate that breaking it requires the solution to a difficult problem in the field of artificial intelligence (AI) rather than just the discovery of the (secret) algorithm, which could be obtained through reverse engineering or other means.

Modern text-based CAPTCHAS are designed such that they require the simultaneous use of three separate abilities—invariant recognition, segmentation, and parsing—to correctly complete the task with any consistency.

1. Invariant recognition refers to the ability to recognize the large amount of variation in the shapes of letters. There are nearly an infinite number of versions for each character that a human brain can successfully identify. The same is not true for a computer, and teaching it to recognize all those differing formations is an extremely challenging task.
2. Segmentation, or the ability to separate one letter from another, is also made difficult in CAPTCHAs, as characters are crowded together with no white space in between.
3. Context is also critical. The CAPTCHA must be understood holistically to correctly identify each character. For example, in one segment of a CAPTCHA, a letter might

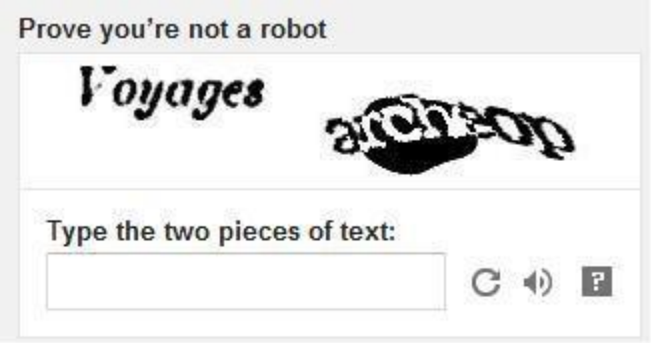
look like an “m.” Only when the whole word is taken into context does it become clear that it is a “u” and an “n.”

Each of these problems pose a significant challenge for a computer, even in isolation. The presence of all three at the same time is what makes CAPTCHAs difficult to solve.

2.4.3 Why we Prefer Captcha rather other security measures?

1. To Protect Website’s Registration Forms –

Many Websites like Hotmail, Gmail, Yahoo, Facebook etc. offers free free registration. It is necessary to protect these website’s registrations so that it ensures the registered user is a human not a program or bot. Captcha Code is used to protect the Registration Form Submission Programmatically.



Prove you’re not a robot

Type the two pieces of text:

Figure 1:- *Gmail Registration Form Captcha Code Image Screen Capture*

2. To Prevent Comment Spams in Blogs –

Captcha Code is used in the comment form so that only human can comment on a post otherwise spammers can flood hundreds of comments to a single post.

3. To Protect Email Address Scrapping –

Spammers crawl the web in the search of Email address posted in the clear text (e.g. **email@website.com**). You can protect your email address either by using Captcha to hide the email address, one can solve the Captcha before showing the Email address or by using alternative trick to post Email Address in the format of email at website dot com.

4. To Protect from Search Engine Bots –

Many Html tags are available to specifying indexing condition to Search engine bots. To prevent a website or specific webpage from search engine crawling, it is desirable to use **html meta tag** but sometimes it is not completely sure that the webpage is fully protected from search engine crawlers and large companies who needs a high security uses Captcha rather than to use only meta tags to protect their highly public protected and confidential Web Pages.

2.4.4 Application of CAPTCHA:-

Applications of CAPTCHAs

CAPTCHAs have several applications for practical security:

- **Preventing Comment Spam in Blogs.** Most bloggers are familiar with programs that submit bogus comments, usually for the purpose of raising search engine ranks of some website (e.g., "buy penny stocks here"). This is called comment spam. By using a CAPTCHA, only humans can enter comments on a blog. There is no need to make users sign up before they enter a comment, and no legitimate comments are ever lost!
- **Protecting Website Registration.** Several companies (Yahoo!, Microsoft, etc.) offer free email services. Up until a few years ago, most of these services suffered from a specific type of attack: "bots" that would sign up for thousands of email accounts every minute. The solution to this problem was to use CAPTCHAs to ensure that only humans obtain free accounts. In general, free services should be protected with a CAPTCHA in order to prevent abuse by automated scripts.
- **Protecting Email Addresses From Scrapers.** Spammers crawl the Web in search of email addresses posted in clear text. CAPTCHAs provide an effective mechanism to hide your email address from Web scrapers. The idea is to require users to solve a CAPTCHA before showing your email address.
- **Online Polls.** In November 1999, *http://www.slashdot.org* released an online poll asking which was the best graduate school in computer science (a dangerous question to ask over the web!). As is the case with most online polls, IP addresses of voters were recorded in order to prevent single users from voting more than once.

However, students at Carnegie Mellon found a way to stuff the ballots using programs that voted for CMU thousands of times. CMU's score started growing rapidly. The next day, students at MIT wrote their own program and the poll became a contest between voting "bots." MIT finished with 21,156 votes, Carnegie Mellon with 21,032 and every other school with less than 1,000. Can the result of any online poll be trusted? Not unless the poll ensures that only humans can vote.

- **Preventing Dictionary Attacks.** CAPTCHAs can also be used to prevent dictionary attacks in password systems. The idea is simple: prevent a computer from being able to iterate through the entire space of passwords by requiring it to solve a CAPTCHA after a certain number of unsuccessful logins. This is better than the classic approach of locking an account after a sequence of unsuccessful logins, since doing so allows an attacker to lock accounts at will.

- **Search Engine Bots.** It is sometimes desirable to keep WebPages unindexed to prevent others from finding them easily. There is an html tag to prevent search engine bots from reading web pages. The tag, however, doesn't guarantee that bots won't read a web page; it only serves to say "no bots, please." Search engine bots, since they usually belong to large companies, respect web pages that don't want to allow them in. However, in order to truly guarantee that bots won't enter a web site, CAPTCHAs are needed.
- **Worms and Spam.** CAPTCHAs also offer a plausible solution against email worms and spam: "I will only accept an email if I know there is a human behind the other computer." A few companies are already marketing this idea.

2.4.5 Advantages:

1. Distinguishes between a human and a machine
2. Makes online polls more legitimate
3. Reduces spam and viruses
4. Makes online shopping safer
5. Diminishes abuse of free email account services

2.4.6 Disadvantages:

1. Sometimes very difficult to read
2. Are not compatible with users with disabilities
3. Time-consuming to decipher
4. Technical difficulties with certain internet browsers
5. May greatly enhance Artificial Intelligence

2.5 Algorithm:

1. Start
2. Install captcha in python by this command : **pip install captcha**
3. Import **image.captcha** from ImageCaptcha
4. Generate captcha by **generate(captcha_text)** function
5. For randomly captcha generation use function **random.randint()**
6. Create an image to write a captcha text
7. End

5.7 Assignment Questions:

1. What is Full Form of CAPTCHA?
2. Write down different forms of CAPTCHA?
3. Why CAPTCHA is needed?
4. Explain the uses of different captcha as per Requirement?

Conclusion:

Hence we conclude that CAPTCHA is used to distinguished Human and Machine and Provide Security to Programs.

Answers **(5)**	Coding Efficiency **(5)**	Viva **(5)**	Timely Completion **(5)**	Total **(20)**	Dated Sign of Subject Teacher

Expected Date of Completion:-------------------------

Actual Date of Completion:----------------------------

Assignment Group A-3

Problem Definition: Write a Computer Forensics Application Program in Java/Python/C++ for recovering Deleted Files and Deleted Partitions.

3.1 Prerequisite:

a) Knowledge about Partitions in Ubuntu. b) Path of Trash folder.

3.2 Learning Objectives:

- Understand the concept of Recovery of deleted files.
- Implementation of recovery of deleted Partitions.

3.3New Concepts:

a. Recovery of files in LINUX OS.

3.4 Theory

3.4.1 Introduction

Have you accidentally deleted an important file because you are in a habit of using “Shift+Del” rather than delete only?? Well don't panic.There are many utilities in Ubuntu and other Linux distributions which helps you in recovering the so called “permanently deleted” files. Actually when you delete a file permanently (accidentally or intentionally), It doesn't get removed from your hard disk.It get stored in certain blocks of the storage device and they continue to exist in the blocks unless you overwrite them with newer files. There are many Tools available to recover permanently deleted files Scalpel.

Scalpel is a platform independent command based tool which is small yet very powerful.

But, if the file is deleted i.e. by just pressing Delete button the file is stored in Trash folder in Ubuntu OS. So it is easy to recover the deleted files from Trash Folder. Just we need to know the path of trash folder.

Path is: ="/home/gurukul/.local/share/Trash/files"

There are sub-Folders in Trash Folder namely :

1. files- contains files which are deleted
2. info- contains information of files deleted
3. expunged

3.4.2 Introduction to file systems:

File systems are one of the things any newcomer to linux must become acquainted with. In the world of Microsoft you never really have to worry about it, the default being NTFS. Linux however, being built on a world of open source and differing opinions, is not limited in this way and so the user should have an understanding of what a file system is, and how it affects the computer.

At the core of a computer, it's all 1s and 0s, but the organization of that data is not quite as simple. A *bit* is a 1 or a 0, a *byte* is composed of 8 bits, a kilobyte is 1024 (i.e. 2) bytes, a megabyte is 1024 kilobytes and so on and so forth. All these *bits* and *bytes* are permanently stored on a Hard Drive. A hard drive stores all your data, any time you save a file, you're writing thousands of 1s and 0s to a metallic disc, changing the magnetic properties that can later be read as 1 or 0. There is so much data on a hard drive that there has to be some way to organize it, like a library of books and the old card drawers that indexed all of them, without that index, we'd be lost. Libraries, for the most part, use the Dewey Decimal System to organize their books, but

there exist other systems to do so, none of which have attained the same fame as Mr. Dewey's invention. File systems are the same way. The ones most users are aware of are the ones Windows uses, the vFat or the NTFS systems, these are the Windows default file systems.

Ubuntu (like all UNIX-like systems) organizes files in a hierarchical tree, where relationships are thought of in teams of children and parent. *Directories* can contain other directories as well as *regular files*, which are the "leaves" of the tree. Any element of the tree can be references by a *path name*; an *absolute path name* starts with the character / (identifying the *root directory*, which contains all other directories and files), then every child directory that must be traversed to reach the element is listed, each separated by a / sign.

3.4.3 Main directories

The standard Ubuntu directory structure mostly follows the File system Hierarchy Standard, which can be referred to for more detailed information.

Here, only the most important directories in the system will be presented.
/bin is a place for most commonly used terminal commands, like ls, mount, rm, etc.

/boot contains files needed to start up the system, including the Linux kernel, a RAM disk image and bootloader configuration files.

/dev contains all *device files*, which are not regular files but instead refer to various hardware devices on the system, including hard drives.

/etc contains system-global configuration files, which affect the system's behavior for all users. **/home** home sweet home, this is the place for users' home directories.

/lib contains very important dynamic libraries and kernel modules

/media is intended as a mount point for external devices, such as hard drives or removable media (floppies, CDs, DVDs).

/mnt is also a place for mount points, but dedicated specifically to "temporarily mounted" devices, such as network filesystems.

/opt can be used to store addition software for your system, which is not handled by the package manager.

/proc is a virtual filesystem that provides a mechanism for kernel to send information to processes.

/root is the superuser's home directory, not in /home/ to allow for booting the system even if /home/ is not available.

/sbin contains important administrative commands that should generally only be employed by the superuser.

/srv can contain data directories of services such as HTTP (/srv/www/) or FTP.

/sys is a virtual filesystem that can be accessed to set or obtain information about the kernel's view of the system.

/tmp is a place for temporary files used by applications.

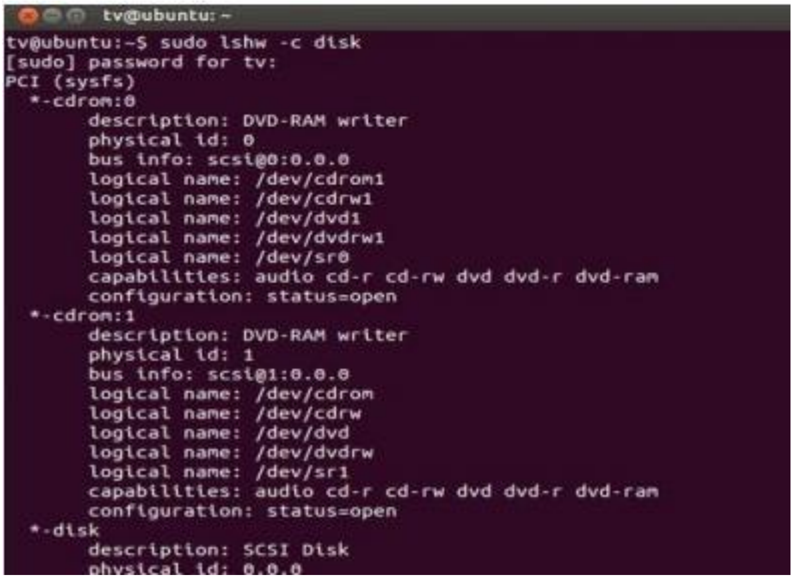
/usr contains the majority of user utilities and applications, and partly replicates the root directory structure, containing for instance, among others, /usr/bin/ and /usr/lib.

/var is dedicated variable data that potentially changes rapidly; a notable directory it contains is /var/log where system log files are kept. **Steps to Partition HardDisk Drive in Ubuntu:-**

Step 1. If you are trying to format or partition your hard drive it is assumed that bios is able to detect the device. To determine the path and other specific information about your drive open a terminal window and enter this command:

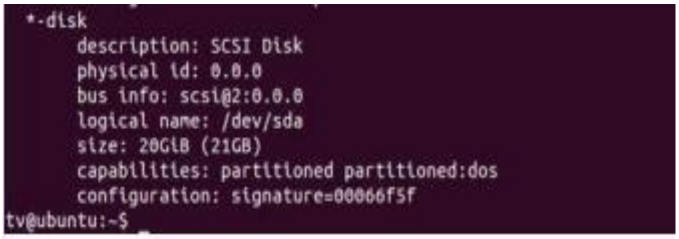
sudo lshw -C disk

Step 2. After entering this command Ubuntu should return something similar to this. Take note of the "logical name" because this will be used throughout the partitioning process if done via terminal window.



```
tv@ubuntu:~$ sudo lshw -c disk
[sudo] password for tv:
PCI (sysfs)
  *-cdrom:0
       description: DVD-RAM writer
       physical id: 0
       bus info: scsi@0:0.0.0
       logical name: /dev/cdrom1
       logical name: /dev/cdrw1
       logical name: /dev/dvd1
       logical name: /dev/dvdrw1
       logical name: /dev/sr0
       capabilities: audio cd-r cd-rw dvd dvd-r dvd-ram
       configuration: status=open
  *-cdrom:1
       description: DVD-RAM writer
       physical id: 1
       bus info: scsi@1:0.0.0
       logical name: /dev/cdrom
       logical name: /dev/cdrw
       logical name: /dev/dvd
       logical name: /dev/dvdrw
       logical name: /dev/sr1
       capabilities: audio cd-r cd-rw dvd dvd-r dvd-ram
       configuration: status=open
  *-disk
       description: SCSI Disk
       physical id: 0.0.0
```

Step 3. The part we will be most concerned with will be the hard drive information that is displayed in the terminal window.

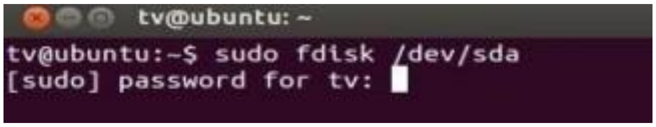


```
  *-disk
       description: SCSI Disk
       physical id: 0.0.0
       bus info: scsi@2:0.0.0
       logical name: /dev/sda
       size: 20GiB (21GB)
       capabilities: partitioned partitioned:dos
       configuration: signature=00066f5f
tv@ubuntu:~$
```

If you plan on using the hard drive only for Ubuntu then the recommended file system to use is either ext3/ext4 depending on whether or not you need backwards compatibility with previous versions of Linux. If you will need to share files between Ubuntu and Windows machines fat 32 is the recommended file system to use, but NTFS will also work well also.

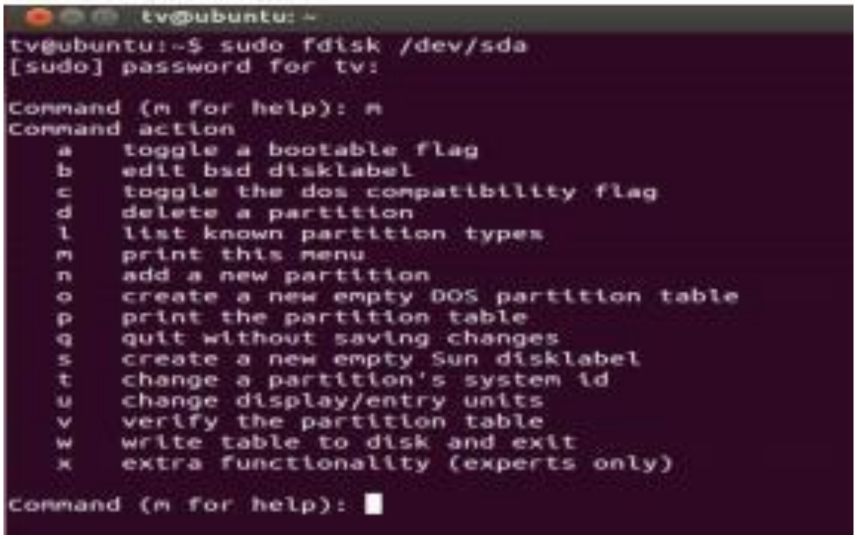
3.4.4 Partition using command line in Terminal: Step

1. Start fdisk with this command



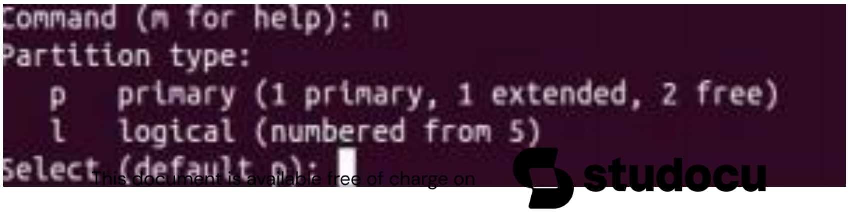
```
tv@ubuntu:~$ sudo fdisk /dev/sda
[sudo] password for tv:
```

Step 2. Press **"m"** then hit **enter**. This will return a menu like the one below showing all of the available commands for the fdisk program.



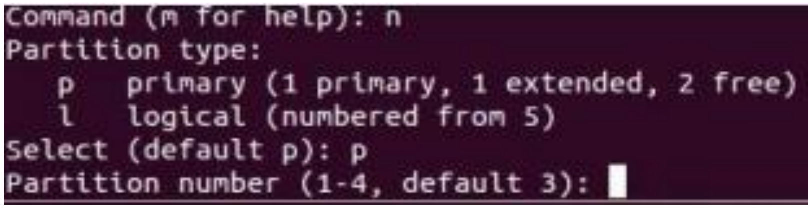
```
tv@ubuntu:~$ sudo fdisk /dev/sda
[sudo] password for tv:

Command (m for help): m
Command action
   a   toggle a bootable flag
   b   edit bsd disklabel
   c   toggle the dos compatibility flag
   d   delete a partition
   l   list known partition types
   m   print this menu
   n   add a new partition
   o   create a new empty DOS partition table
   p   print the partition table
   q   quit without saving changes
   s   create a new empty Sun disklabel
   t   change a partition's system id
   u   change display/entry units
   v   verify the partition table
   w   write table to disk and exit
   x   extra functionality (experts only)

Command (m for help):
```

Step 3. Since we want to add a new partition press **"n"** and then **enter**.



```
Command (m for help): n
Partition type:
   p   primary (1 primary, 1 extended, 2 free)
   l   logical (numbered from 5)
Select (default p):
```

Step 4. To create a primary partition (what we want) press **“p”** and then hit **enter**.



```
Command (m for help): n
Partition type:
   p   primary (1 primary, 1 extended, 2 free)
   l   logical (numbered from 5)
Select (default p): p
Partition number (1-4, default 3):
```

Step 5.

If you only want 1 partition press **“1”** and hit **enter**. You may be provided with a default response, you may choose this as the Partition number if you would like. Next you will be prompted for the locations of where you would like the first and last sectors of the partition to be. You may again be provided with default responses choose these if you want.

Step 6.

Now choose w to write the partition to the disk. Type “w” then press enter.Your drive is now partitioned. Now we need to format it. By default Linux will recognize this partition as dev/sdb1.

Step 7. To format the partition with an ext3 filesystem.

sudo mkfs -t ext3 /dev/sdb1

3.5 Algorithm:

1. Start
2. Initialize variables as path="/home/gurukul/.local/share/Trash/files" infopath="/home/gurukul/.local/share/Trash/info"
3. Check the list of files present in files folder.
4. Find the path of file to restore it using info folder.
5. Copy the contents of file which is deleted and is in Trash folder into new file at original location.
6. Delete the file from Trash folder.
7. End

3.6 Mathematical Model:

I= P (path of trash folder) **Functions:**
re.findall(r'/.*',line)
destipath.lstrip('[') destipath.rstrip(']')
destipath[:-1]
destipath[1:] **Output:**
R- Recovered file

3.7 Assignment Questions:

1. What is Path of Trash folder in Ubuntu and what are the different folders?
2. How to see hidden files and filesystem of ubuntu?
3. What are different file systems in Ubuntu also state main directories of it??
4. How to list different files and what are the various options of ls used for file related function?

Conclusion:

Hence we conclude that using **Forensics Application Program in Python we can recover Deleted Files.**

Answers **(5)**	Coding Efficiency **(5)**	Viva **(5)**	Timely Completion **(5)**	Total **(20)**	Dated Sign of Subject Teacher

Expected Date of Completion:-------------------------

Actual Date of Completion:----------------------------

Assignment Group A-4

Write a program for Log Capturing and Event Correlation

Prerequisite:

- Latest version of Squid should be used.(version 2.5 or greater)
- A web server for testing purpose which can be used instead of Internet.
- Squid Version greater than 2.6 is required for Transparent squid proxy configuration in this lab.
-

Learning Objectives:

- To understand how Log Records are generated for Further Analysis.

New Concepts:

- Squid and Sarg

Theory

4.1. Introduction:

- During the period of development of internet, users are allowed for unlimited access to the resources due to less number of users. So there were less issues related to accessing speed over internet.
- With the increase in internet usage, many issues raised related to accessing speed, effective bandwidth utilization etc. One method of overcoming these issues is, maintaining a copy of webpage visited by a user in the cache so that the other user who visits the same webpage will access the same website within a short period of time. This method not only increases the accessing speed but also helps in utilizing the bandwidth effectively.
- The above said functionality can be achieved by maintaining a proxy server through which all the users in the organization or a group access the internet. The most widely used proxy server in Linux is Squid Proxy, which is free software released General Public License.

- Squid provides proxy and cache services for **H**yper **T**ext **T**ransfer **P**rotocol (HTTP),

  File **T**ransfer **P**rotocol (FTP), and various other protocols.

To configure a system as a proxy server, one should have a sufficient amount of memory for maintaining the cache which in turn increases the performance.

- In case if the internet connection is not available, setup one host as a web server in place of internet and assign the IP address to the proxy server network interface in the network, used by web server instead of public IP address assigned to that interface.

4.2. Steps to Configure Squid Proxy:

4.2.1. Installation of Squid Package

A Squid proxy server is generally installed on a separate server than the Web server with the original files. Squid works by tracking object use over the network. Squid will initially act as an intermediary, simply passing the client's request on to the server and saving a copy of the requested object. If the same client or multiple clients request the same object before it expires from Squid's cache, Squid can then immediately serve it, accelerating the download and saving bandwidth.

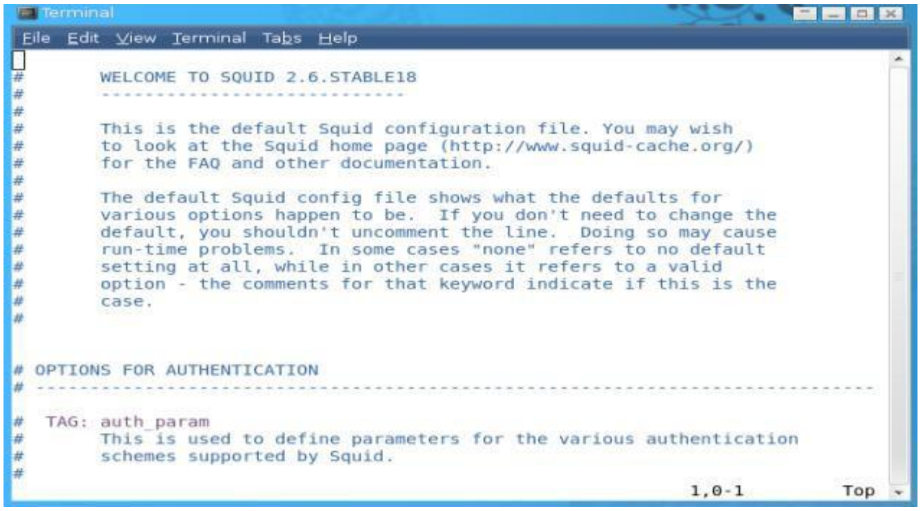
```
sudo apt update
sudo apt -y install squid
```

4.2.2. Accessing the Proxy Server configuration file

To configure squid proxy server we need to edit the *sudo gedit /etc/squid/squid.conf* file and the default location of squid.conf file varies from distribution to distribution and from version to version. We can edit the configuration file using vi editor through command prompt.

```
sudo gedit /etc/squid/squid.conf
```

Then the content of the configuration file can be viewed as shown below in the figure.



```
#
#       WELCOME TO SQUID 2.6.STABLE18
#       ----------------------------
#
#       This is the default Squid configuration file. You may wish
#       to look at the Squid home page (http://www.squid-cache.org/)
#       for the FAQ and other documentation.
#
#       The default Squid config file shows what the defaults for
#       various options happen to be.  If you don't need to change the
#       default, you shouldn't uncomment the line.  Doing so may cause
#       run-time problems.  In some cases "none" refers to no default
#       setting at all, while in other cases it refers to a valid
#       option - the comments for that keyword indicate if this is the
#       case.
#


# OPTIONS FOR AUTHENTICATION
# -----------------------------------------------------------------------------

#  TAG: auth_param
#       This is used to define parameters for the various authentication
#       schemes supported by Squid.
#
                                                         1,0-1          Top
```

Editing the squid configuration file

`sudo gedit /etc/squid/squid.conf`

Search the TAG: auth_param and paste the following acl

```
auth_param basic program /usr/lib/squid/basic_ncsa_auth /etc/squid/passwd
auth_param basic children 5
auth_param basic realm Squid Basic Authentication
auth_param basic credentialsttl 2 hours
acl auth_users proxy_auth REQUIRED
http_access allow auth_users

 #search localnet and paste line in last

acl localnet src 192.168.60.70
      >sudo service squid3 restart
```

Specifying the interface and port number on which the proxy server should listen.

By default, the proxy server will listen on all the available network interfaces on the system for requests. For Example, if one interface card is assigned a public ip from which it is connected to internet and the other interface card is assigned an ip address which belongs to your local area network. Then in order to make you proxy server to listen for requests from your Local Area Network through a particular port, then change the variable http_port 3128 in the squid configuration file to desired ip address and port number in the format shown below.

```
http_port   <ip address belonging to LAN>:<port number>
```

Example: For example, if your proxy server has an ip address 192.168.60.70 which belongs to the local area network 192.168.60.0/24 and you want the server to listen for requests from your LAN through a particular port say 3456, then you can change the variable http_port as shown.

```
http_port 192.168.60.70:3456
```

Assigning Access Controls

By default, no user machine is allowed to connect to the proxy server except the localhost. To allow the local machines access your proxy server, locate the acl section in the squid configuration file starting with acl and at the end of the last acl line specify your access

control. For example to allow local area network 192.168.60.0/24 machines to access your proxy server, specify the acl as

```
acl  mylan src 192.168.60.0/255.255.255.0
```

In the above example, mylan specifies the name of my access control. We can specify any name other than my lan for access control. src specifies the source network.

Allow or Deny based on Access Control.

After specifying the access control for your local LAN, we need to provide allow permission for the specified LAN using http_access variable in the squid configuration file as shown in the example below.

Example: To allow the above specified access control (i.e acl mylan src 192.168.60.0/255.255.255.0), we need to specify the http_access variable as

Copyright © 2009, Centre for Development of Advanced Computing,
Hyderabad http_access allow mylan

Here mylan specifies the access control used. Suppose if we want to allow all the networks except the 192.168.60.0/24 network to access the proxy then we can specify the http_access variable as

```
http_access deny !mylan
```

In the above line, !mylan specifies except mylan network.

Note:

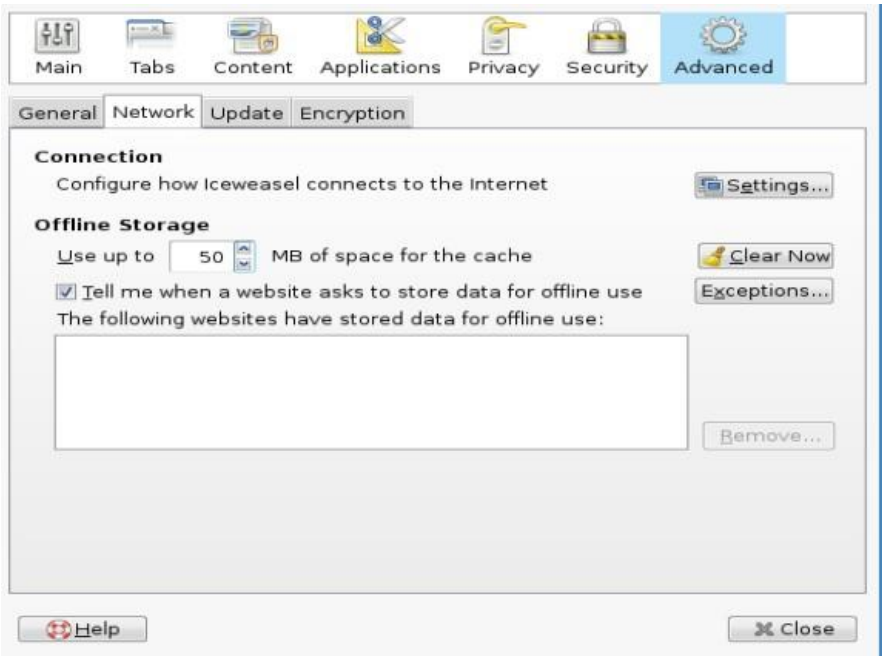
The above specified http_access variable should be specified before the line http_access deny all in the configuration file.

Saving the changes and exit the gedit Editor

After making appropriate changes to your configuration file exit the vi editor window by pressing Esc followed by :wq!. Here wq specifies save changes and exit the configuration file.

4.3 Testing the Squid configuration

To test the squid configuration, open a browser in any one of the pc in local area network or on the proxy server and specify the proxy settings as the ipaddress of the proxy server and port on which it is listening for requests. For example, in firefox web browser if we want to set the proxy settings in the browser window goto **Edit -->Preferences** and window similar to shown below will be displayed.



Connection

Configure how Iceweasel connects to the Internet

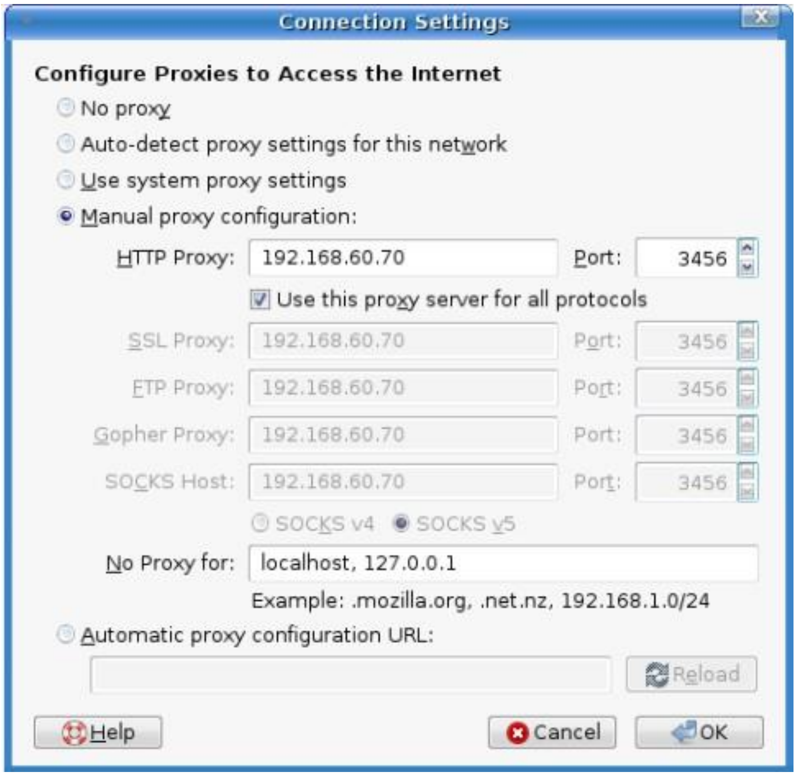
Offline Storage

Use up to 50 MB of space for the cache

☑ Tell me when a website asks to store data for offline use

The following websites have stored data for offline use:

Now select Advanced tab, and under advanced tab click on Network tab and click on Settings option under Connection field. Then a window similar to the shown below will be displayed.



Connection Settings

Configure Proxies to Access the Internet

○ No proxy
○ Auto-detect proxy settings for this network
○ Use system proxy settings
● Manual proxy configuration:

HTTP Proxy: 192.168.60.70 Port: 3456
☑ Use this proxy server for all protocols
SSL Proxy: 192.168.60.70 Port: 3456
FTP Proxy: 192.168.60.70 Port: 3456
Gopher Proxy: 192.168.60.70 Port: 3456
SOCKS Host: 192.168.60.70 Port: 3456
○ SOCKS v4 ● SOCKS v5
No Proxy for: localhost, 127.0.0.1
Example: .mozilla.org, .net.nz, 192.168.1.0/24

○ Automatic proxy configuration URL:

4.4 SARG – Squid Analysis Report Generator and Internet Bandwidth Monitoring Tool

SARG is an open source tool that allows you to analyze the squid log files and generates beautiful reports in HTML format with information’s about users, IP addresses, top accessed sites, total bandwidth usage, elapsed time, downloads, access denied websites, daily reports, weekly reports and monthly reports.

The SARG is very handy tool to view how much internet bandwidth is utilized by individual machines on the network and can watch on which websites the network’s users are accessing.

Installing Sarg from Source

The ‘**sarg**‘package by default not included in **RedHat** based distributions, so we need to manually compile and install it from source tar ball. For this, we need some additional pre- requisites packages to be installed on the system before compiling it from source.

$ sudo apt-get install sarg

Configuring Sarg

Now it's time to edit some parameters in SARG main configuration file. The file contains lots of options to edit, but we will only edit required parameters like:

Access logs

path

Output

directory

Date

Format

Overwrite report for the same date.

Open sarg.conf file with your choice of editor and make changes as shown below.

```
# vi /usr/local/etc/sarg.conf       [On RedHat based systems]
```

Now Uncomment and add the original path to your squid access
log file. # sarg.conf

```
# TAG:  access_log file
#       Where is the access.log file
#       sarg
-l file
access_log /var/log/squid/access.log
```

Next, add the correct Output directory path to save the generate squid reports in that directory. Please note, under Debian based distributions the Apache web root directory is '/var/www'. So, please be careful while adding correct web root paths under your Linux distributions.

```
# TAG:  output_dir
#       The reports will be saved in that
directory #   sarg -o dir

output_dir  /var/www/html/squid-reports
```

Set the correct date format for reports. For example, 'date_format e' will

display reports in ‘dd/mm/yy‘ format.

```
# TAG:  date_format
#           Date format in reports: e (European=dd/mm/yy), u
(American=mm/dd/yy), w (Weekly=yy.ww)
#
date_format e
```

Next, uncomment and set Overwrite report to
‘Yes’. # TAG: overwrite_report yes|no

```
#  yes - if report date already exist then will be overwritten.
#  no - if report date already exist then will be renamed to filename.n,
filename.n+1 #
overwrite_report yes
```

That’s it! Save and close the file.

Step 3: Generating Sarg Report

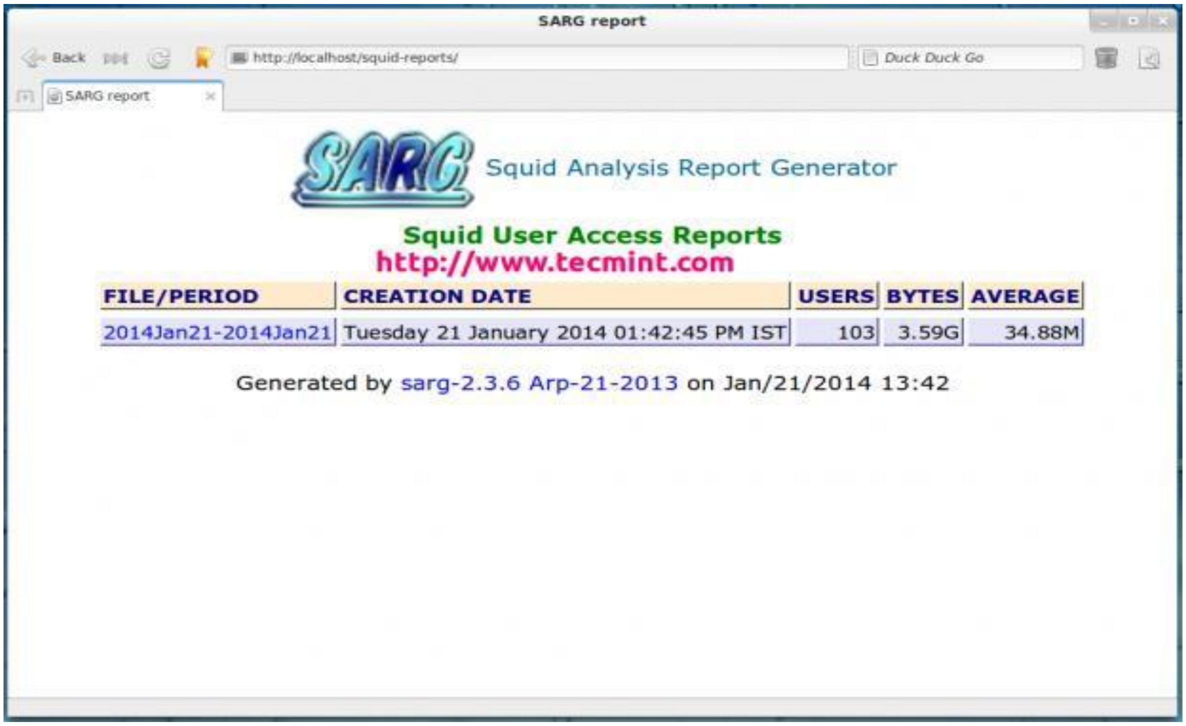
Once, you’ve done with the configuration part, it’s time to generate the squid log report using the following command.

```
# sarg -x       [On RedHat based systems]
```

Assessing Sarg Report

The generated reports placed under ‘/var/www/html/squid-reports/‘ or ‘/var/www/squid-reports/‘ which can be accessed from the web browser using the address.

http://localhost/squid-
reports OR
http://ip-address/squid-reports


SARG report
Back
http://localhost/squid-reports/
Duck Duck Go
SARG report
SARG Squid Analysis Report Generator

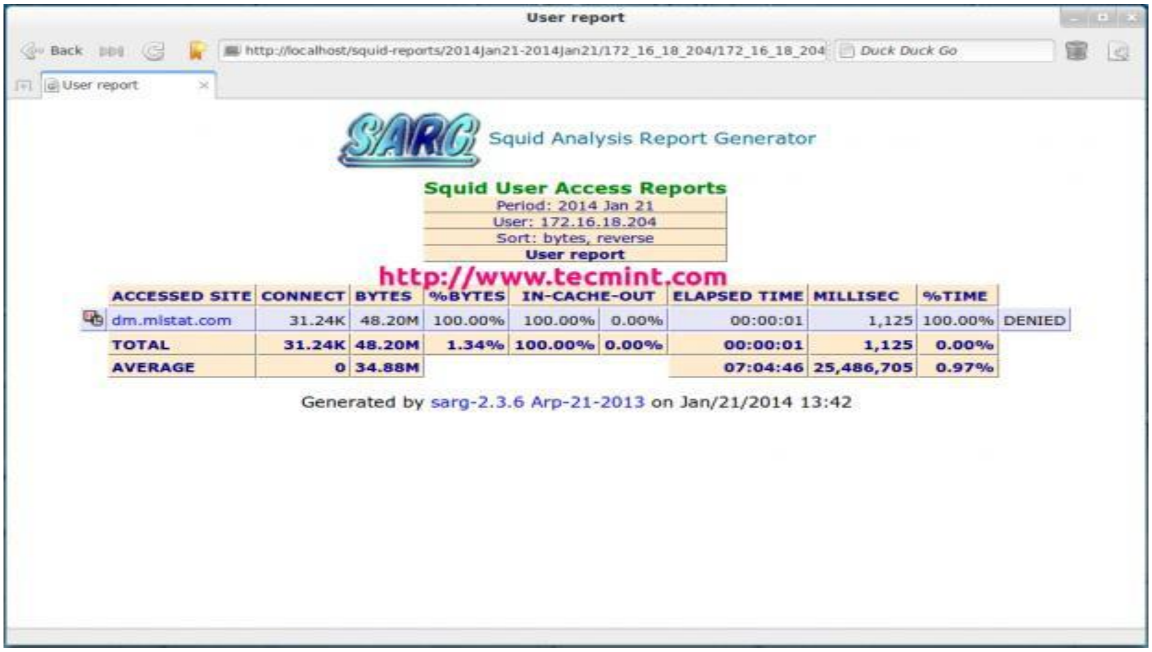
Squid User Access Reports

http://www.tecmint.com

FILE/PERIOD	CREATION DATE	USERS	BYTES	AVERAGE
2014Jan21-2014Jan21	Tuesday 21 January 2014 01:42:45 PM IST	103	3.59G	34.88M

Generated by sarg-2.3.6 Arp-21-2013 on Jan/21/2014 13:42

Fig.1 Sarg Main Window


User report
Back
http://localhost/squid-reports/2014Jan21-2014Jan21/172_16_18_204/172_16_18_204
Duck Duck Go
User report
SARG Squid Analysis Report Generator

Squid User Access Reports

Period: 2014 Jan 21

User: 172.16.18.204

Sort: bytes, reverse

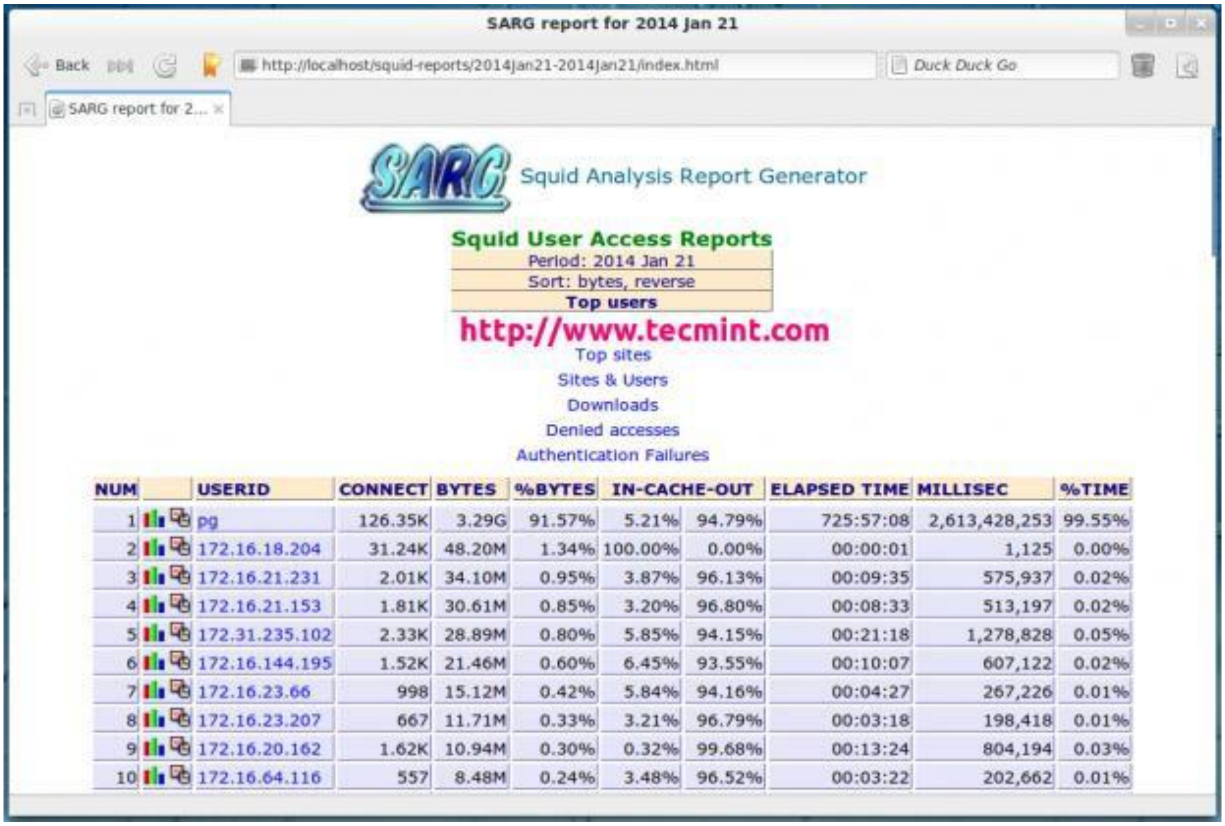
User report

http://www.tecmint.com

ACCESSED SITE	CONNECT	BYTES	%BYTES	IN-CACHE-OUT		ELAPSED TIME	MILLISEC	%TIME	
dm.mistat.com	31.24K	48.20M	100.00%	100.00%	0.00%	00:00:01	1,125	100.00%	DENIED
TOTAL	31.24K	48.20M	1.34%	100.00%	0.00%	00:00:01	1,125	0.00%	
AVERAGE	0	34.88M				07:04:46	25,486,705	0.97%	

Generated by sarg-2.3.6 Arp-21-2013 on Jan/21/2014 13:42

Fig2.User Report


SARG report for 2014 Jan 21
Back
http://localhost/squid-reports/2014Jan21-2014Jan21/index.html
Duck Duck Go
SARG report for 2...
SARG

Squid Analysis Report Generator

Squid User Access Reports

Period: 2014 Jan 21

Sort: bytes, reverse

Top users

http://www.tecmint.com

Top sites

Sites & Users

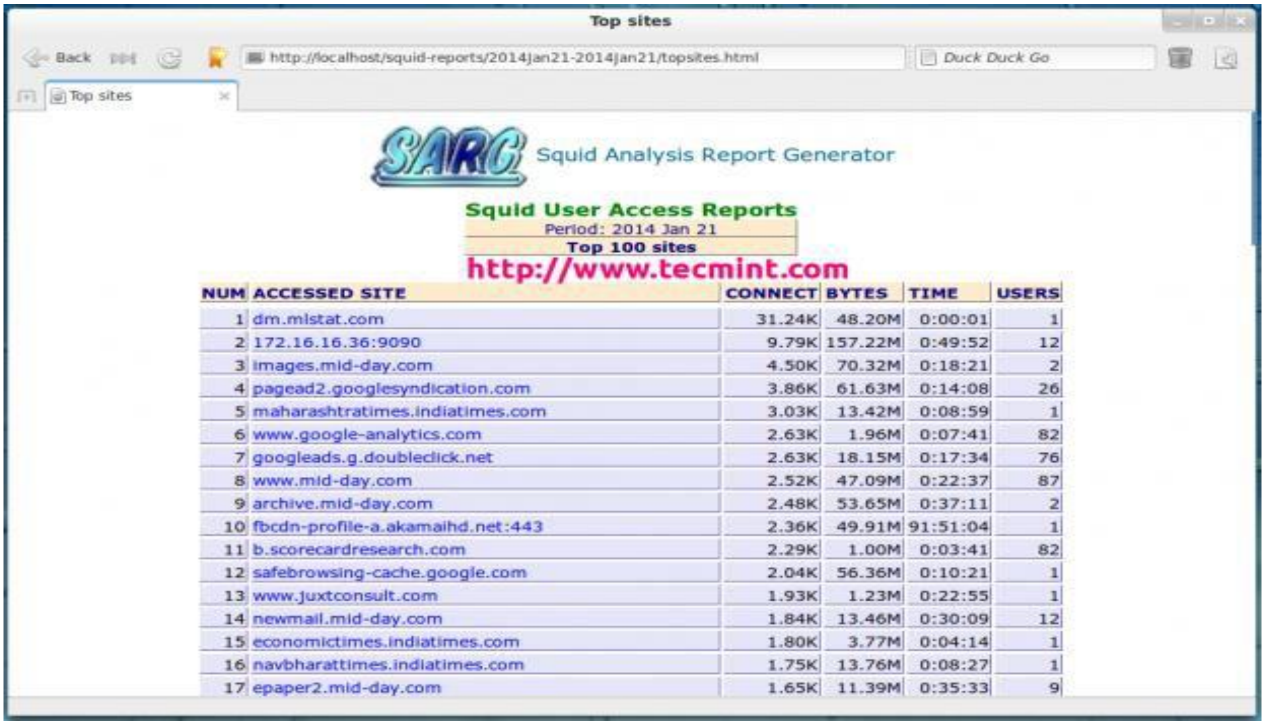
Downloads

Denied accesses

Authentication Failures

NUM		USERID	CONNECT	BYTES	%BYTES	IN-CACHE-OUT		ELAPSED TIME	MILLISEC	%TIME
1		pg	126.35K	3.29G	91.57%	5.21%	94.79%	725:57:08	2,613,428,253	99.55%
2		172.16.18.204	31.24K	48.20M	1.34%	100.00%	0.00%	00:00:01	1,125	0.00%
3		172.16.21.231	2.01K	34.10M	0.95%	3.87%	96.13%	00:09:35	575,937	0.02%
4		172.16.21.153	1.81K	30.61M	0.85%	3.20%	96.80%	00:08:33	513,197	0.02%
5		172.31.235.102	2.33K	28.89M	0.80%	5.85%	94.15%	00:21:18	1,278,828	0.05%
6		172.16.144.195	1.52K	21.46M	0.60%	6.45%	93.55%	00:10:07	607,122	0.02%
7		172.16.23.66	998	15.12M	0.42%	5.84%	94.16%	00:04:27	267,226	0.01%
8		172.16.23.207	667	11.71M	0.33%	3.21%	96.79%	00:03:18	198,418	0.01%
9		172.16.20.162	1.62K	10.94M	0.30%	0.32%	99.68%	00:13:24	804,194	0.03%
10		172.16.64.116	557	8.48M	0.24%	3.48%	96.52%	00:03:22	202,662	0.01%

Fig 3. Specific Date


Top sites
Back
http://localhost/squid-reports/2014Jan21-2014Jan21/topsites.html
Duck Duck Go
Top sites
SARG

Squid Analysis Report Generator

Squid User Access Reports

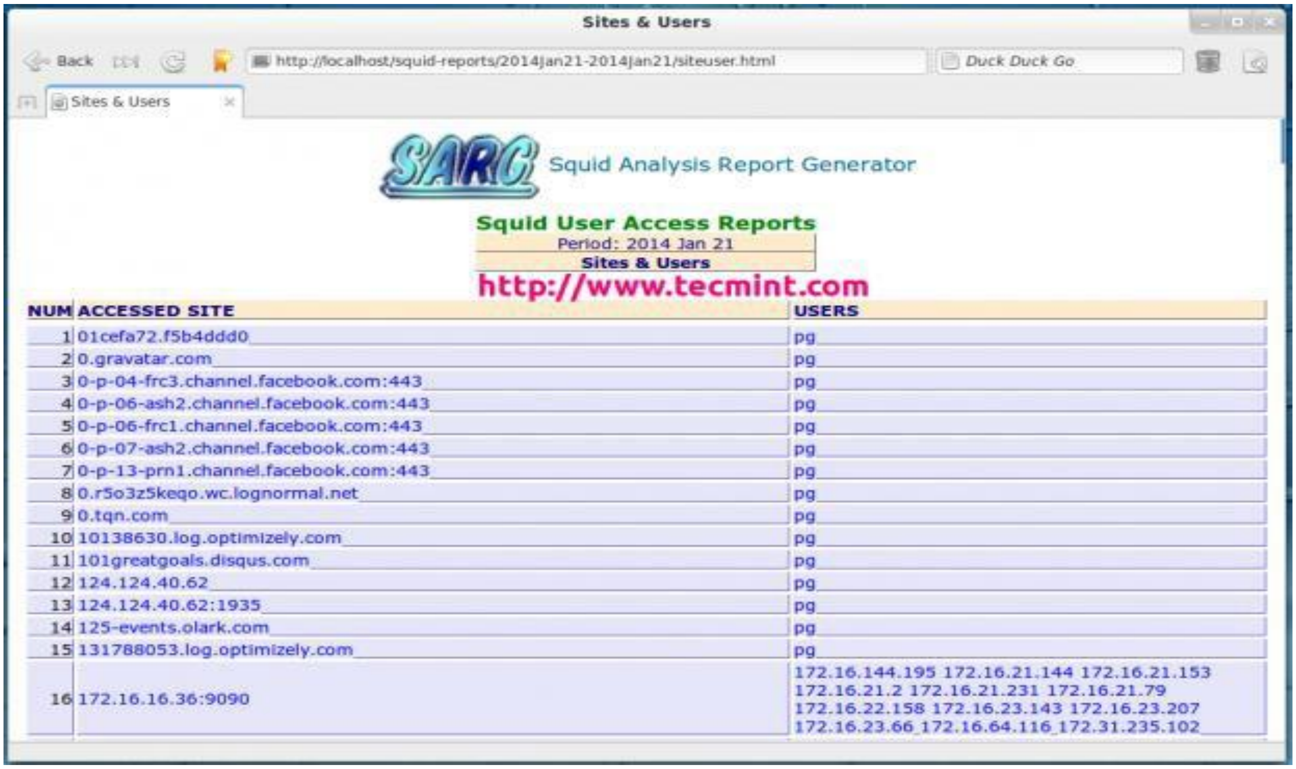
Period: 2014 Jan 21

Top 100 sites

http://www.tecmint.com

NUM	ACCESSED SITE	CONNECT	BYTES	TIME	USERS
1	dm.mlstat.com	31.24K	48.20M	0:00:01	1
2	172.16.16.36:9090	9.79K	157.22M	0:49:52	12
3	images.mid-day.com	4.50K	70.32M	0:18:21	2
4	pagead2.googlesyndication.com	3.86K	61.63M	0:14:08	26
5	maharashtratimes.indiatimes.com	3.03K	13.42M	0:08:59	1
6	www.google-analytics.com	2.63K	1.96M	0:07:41	82
7	googleads.g.doubleclick.net	2.63K	18.15M	0:17:34	76
8	www.mid-day.com	2.52K	47.09M	0:22:37	87
9	archive.mid-day.com	2.48K	53.65M	0:37:11	2
10	fbcdn-profile-a.akamaihd.net:443	2.36K	49.91M	91:51:04	1
11	b.scorecardresearch.com	2.29K	1.00M	0:03:41	82
12	safebrowsing-cache.google.com	2.04K	56.36M	0:10:21	1
13	www.juxtconsult.com	1.93K	1.23M	0:22:55	1
14	newmail.mid-day.com	1.84K	13.46M	0:30:09	12
15	economictimes.indiatimes.com	1.80K	3.77M	0:04:14	1
16	navbharattimes.indiatimes.com	1.75K	13.76M	0:08:27	1
17	epaper2.mid-day.com	1.65K	11.39M	0:35:33	9

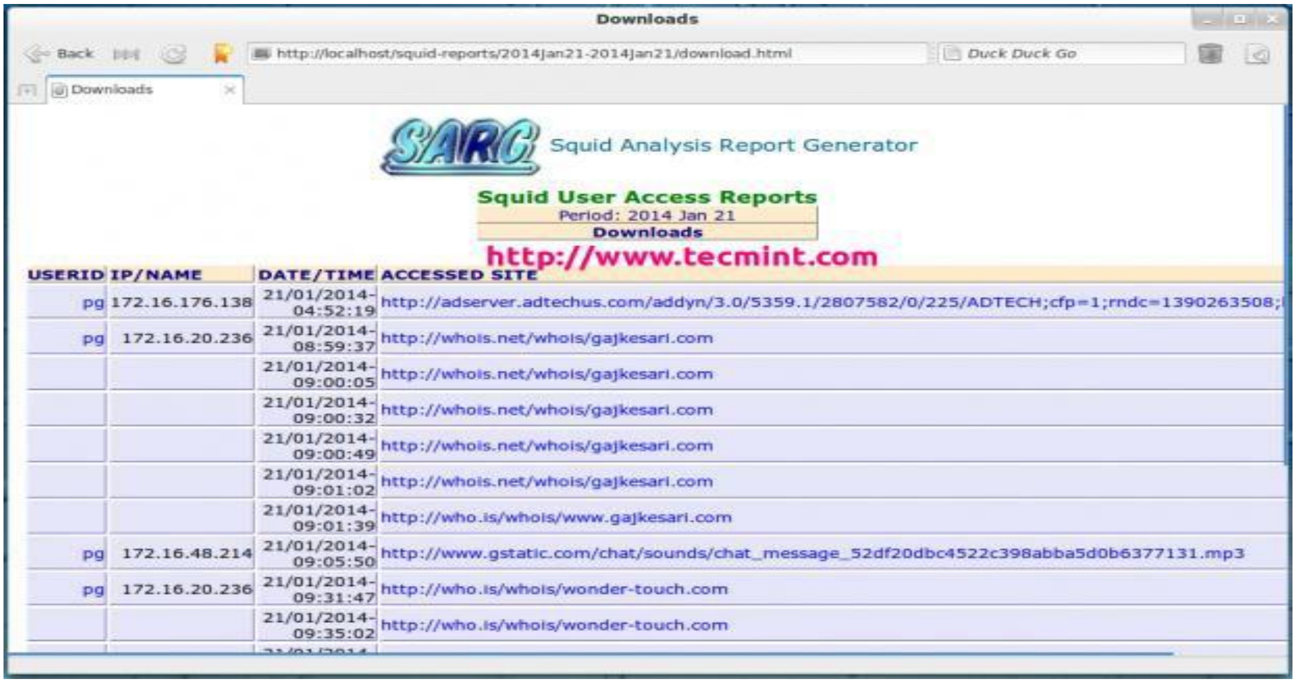
Fig 4. Top Accessed Sites


Sites & Users
Back
http://localhost/squid-reports/2014Jan21-2014Jan21/siteuser.html
Duck Duck Go
Sites & Users
SARG
Squid Analysis Report Generator
http://www.tecmint.com

Squid User Access Reports
Period: 2014 Jan 21
Sites & Users

NUM	ACCESSED SITE	USERS
1	01cefa72.f5b4ddd0	pg
2	0.gravatar.com	pg
3	0-p-04-frc3.channel.facebook.com:443	pg
4	0-p-06-ash2.channel.facebook.com:443	pg
5	0-p-06-frc1.channel.facebook.com:443	pg
6	0-p-07-ash2.channel.facebook.com:443	pg
7	0-p-13-prn1.channel.facebook.com:443	pg
8	0.r5o3z5keqo.wc.lognormal.net	pg
9	0.tqn.com	pg
10	10138630.log.optimizely.com	pg
11	101greatgoals.disqus.com	pg
12	124.124.40.62	pg
13	124.124.40.62:1935	pg
14	125-events.olark.com	pg
15	131788053.log.optimizely.com	pg
16	172.16.16.36:9090	172.16.144.195 172.16.21.144 172.16.21.153 172.16.21.2 172.16.21.231 172.16.21.79 172.16.22.158 172.16.23.143 172.16.23.207 172.16.23.66 172.16.64.116 172.31.235.102

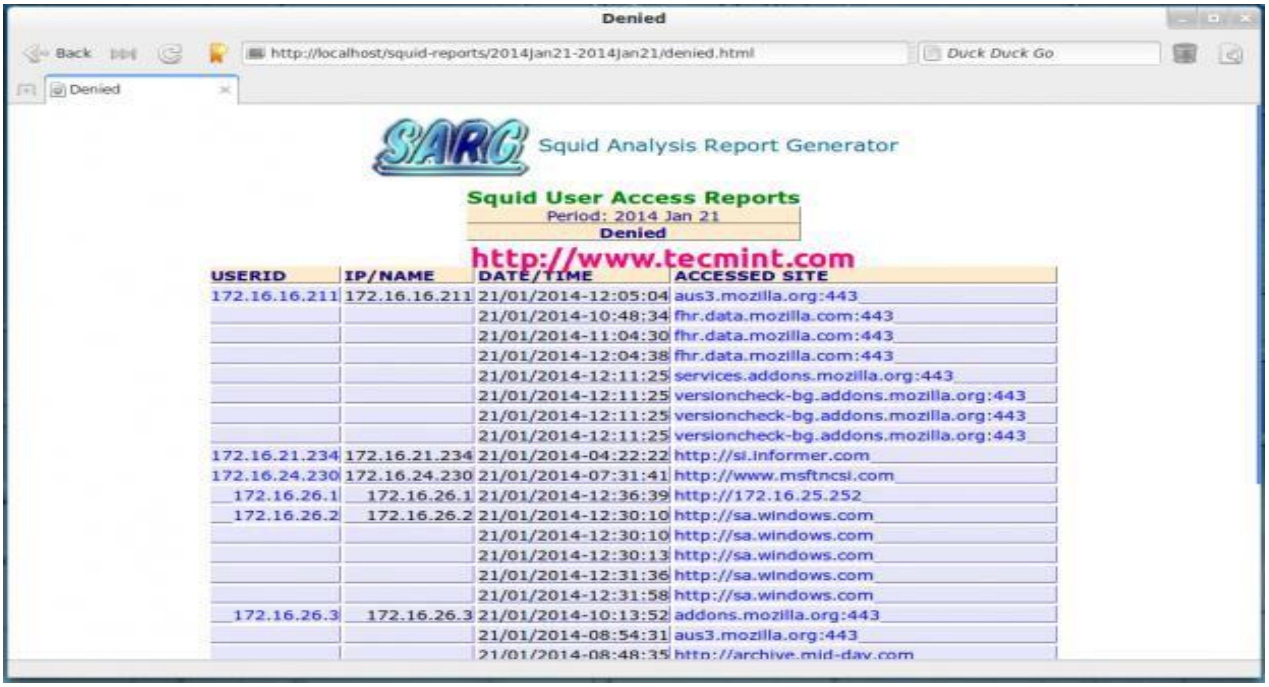
Fig 5. Top Sites and Users


Downloads
Back
http://localhost/squid-reports/2014Jan21-2014Jan21/download.html
Duck Duck Go
Downloads
SARG
Squid Analysis Report Generator
http://www.tecmint.com

Squid User Access Reports
Period: 2014 Jan 21
Downloads

USERID	IP/NAME	DATE/TIME	ACCESSED SITE
pg	172.16.176.138	21/01/2014-04:52:19	http://adserver.adtechus.com/addyn/3.0/5359.1/2807582/0/225/ADTECH;cfp=1;rndc=1390263508;
pg	172.16.20.236	21/01/2014-08:59:37	http://whois.net/whois/gajkesari.com
		21/01/2014-09:00:05	http://whois.net/whois/gajkesari.com
		21/01/2014-09:00:32	http://whois.net/whois/gajkesari.com
		21/01/2014-09:00:49	http://whois.net/whois/gajkesari.com
		21/01/2014-09:01:02	http://whois.net/whois/gajkesari.com
		21/01/2014-09:01:39	http://who.is/whois/www.gajkesari.com
pg	172.16.48.214	21/01/2014-09:05:50	http://www.gstatic.com/chat/sounds/chat_message_52df20dbc4522c398abba5d0b6377131.mp3
pg	172.16.20.236	21/01/2014-09:31:47	http://who.is/whois/wonder-touch.com
		21/01/2014-09:35:02	http://who.is/whois/wonder-touch.com

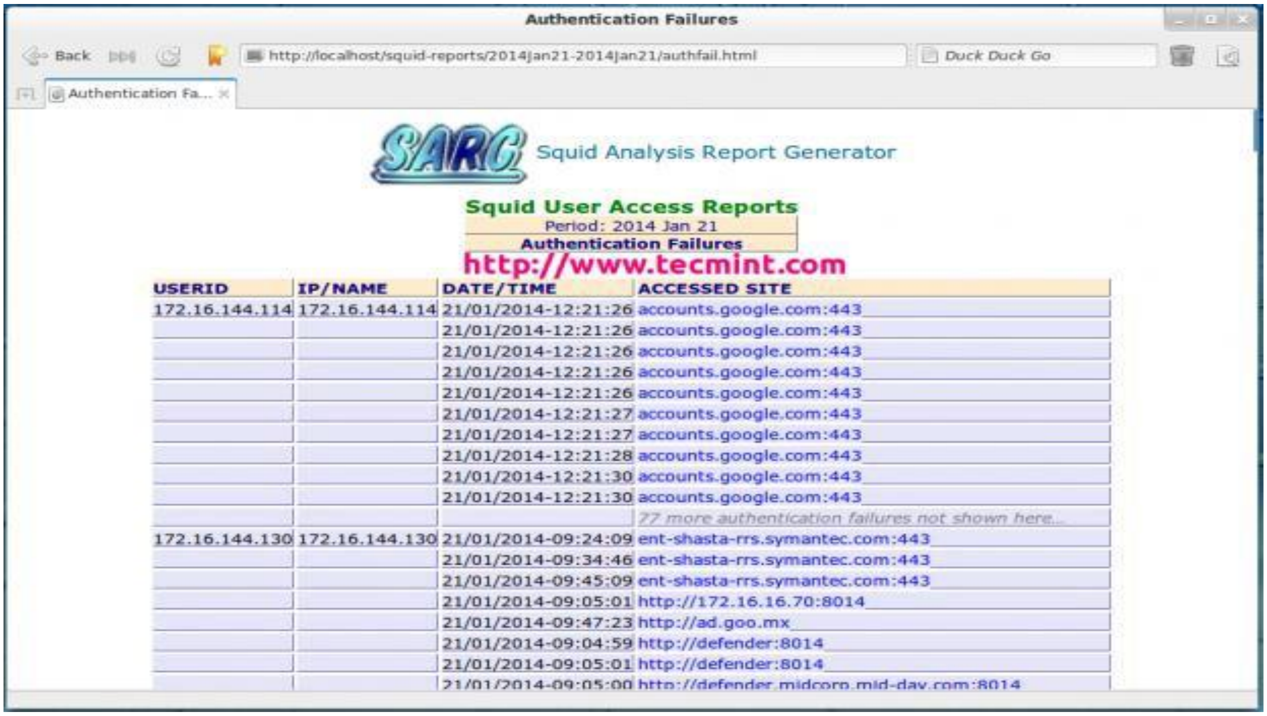
Fig 6. Top Downloads


Denied
Back
http://localhost/squid-reports/2014Jan21-2014Jan21/denied.html
Duck Duck Go
Denied
SARG
Squid Analysis Report Generator

Squid User Access Reports
Period: 2014 Jan 21
Denied
http://www.tecmint.com

USERID	IP/NAME	DATE/TIME	ACCESSED SITE
172.16.16.211	172.16.16.211	21/01/2014-12:05:04	aus3.mozilla.org:443
		21/01/2014-10:48:34	fhr.data.mozilla.com:443
		21/01/2014-11:04:30	fhr.data.mozilla.com:443
		21/01/2014-12:04:38	fhr.data.mozilla.com:443
		21/01/2014-12:11:25	services.addons.mozilla.org:443
		21/01/2014-12:11:25	versioncheck-bg.addons.mozilla.org:443
		21/01/2014-12:11:25	versioncheck-bg.addons.mozilla.org:443
		21/01/2014-12:11:25	versioncheck-bg.addons.mozilla.org:443
172.16.21.234	172.16.21.234	21/01/2014-04:22:22	http://si.informer.com
172.16.24.230	172.16.24.230	21/01/2014-07:31:41	http://www.msftncsi.com
172.16.26.1	172.16.26.1	21/01/2014-12:36:39	http://172.16.25.252
172.16.26.2	172.16.26.2	21/01/2014-12:30:10	http://sa.windows.com
		21/01/2014-12:30:10	http://sa.windows.com
		21/01/2014-12:30:13	http://sa.windows.com
		21/01/2014-12:31:36	http://sa.windows.com
		21/01/2014-12:31:58	http://sa.windows.com
172.16.26.3	172.16.26.3	21/01/2014-10:13:52	addons.mozilla.org:443
		21/01/2014-08:54:31	aus3.mozilla.org:443
		21/01/2014-08:48:35	http://archive.mid-day.com

Fig 7. Denied Access


Authentication Failures
Back
http://localhost/squid-reports/2014Jan21-2014Jan21/authfail.html
Duck Duck Go
Authentication Fa...
SARG
Squid Analysis Report Generator

Squid User Access Reports
Period: 2014 Jan 21
Authentication Failures
http://www.tecmint.com

USERID	IP/NAME	DATE/TIME	ACCESSED SITE
172.16.144.114	172.16.144.114	21/01/2014-12:21:26	accounts.google.com:443
		21/01/2014-12:21:26	accounts.google.com:443
		21/01/2014-12:21:26	accounts.google.com:443
		21/01/2014-12:21:26	accounts.google.com:443
		21/01/2014-12:21:26	accounts.google.com:443
		21/01/2014-12:21:27	accounts.google.com:443
		21/01/2014-12:21:27	accounts.google.com:443
		21/01/2014-12:21:28	accounts.google.com:443
		21/01/2014-12:21:30	accounts.google.com:443
		21/01/2014-12:21:30	accounts.google.com:443
			77 more authentication failures not shown here...
172.16.144.130	172.16.144.130	21/01/2014-09:24:09	ent-shasta-rrs.symantec.com:443
		21/01/2014-09:34:46	ent-shasta-rrs.symantec.com:443
		21/01/2014-09:45:09	ent-shasta-rrs.symantec.com:443
		21/01/2014-09:05:01	http://172.16.16.70:8014
		21/01/2014-09:47:23	http://ad.goo.mx
		21/01/2014-09:04:59	http://defender:8014
		21/01/2014-09:05:01	http://defender:8014
		21/01/2014-09:05:00	http://defender.midcorp.mid-day.com:8014

Fig 8. Authentication Failures

4.5 Assignment Questions:

1. Why to Configure Proxy Server?
2. What is SARG?
3. Which Parameter is there in SARG Report?
4. What do you mean by Log and Event Co-relation?

Conclusion:

By configuring this Network Administrator can easily analyze the Network Traffic and Bandwidth Utilization.

Answers **(5)**	Coding Efficiency **(5)**	Viva **(5)**	Timely Completion **(5)**	Total **(20)**	Dated Sign of Subject Teacher

Expected Date of Completion:-------------------------

Actual Date of Completion:----------------------------

Assignment Group A-5

Problem Definition:

Study and Implementation of Honeypot.

5.1 Learning Objectives:

1. To learn the concept of Honeypot
2. To study the representation, implementation of Honeypot

5.2 Learning outcome:

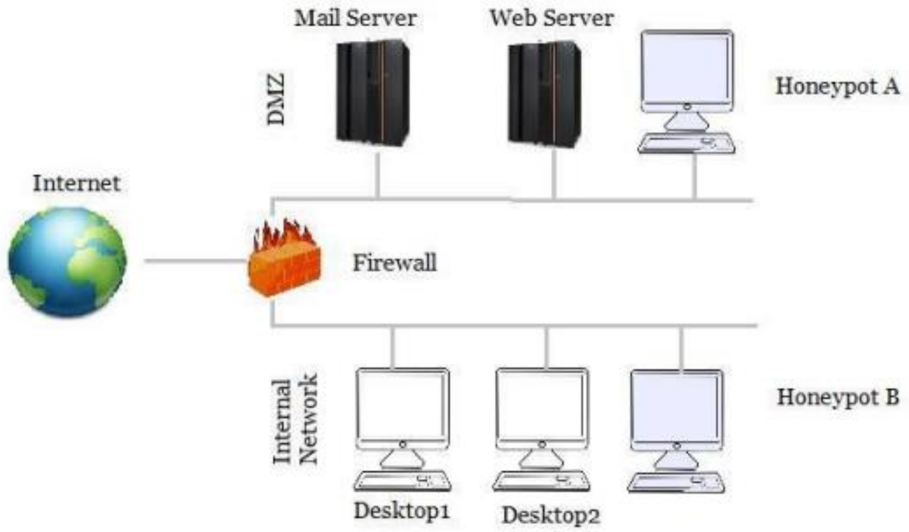
Use honeybot tool to capture packets and configure tools and systems to enter unknown unauthenticated IP.

5.3 Theory

5.3.1 Honeypot

It is a computer system. There are files, directories in it just like a real computer. However, the aim of the computer is to attract hackers to fall into it to watch and follow their behavior. So we can define it as a fake system which looks like a real system. They are different than other security systems since they are not only finding one solution to a particular problem, but also they are eligible to apply variety of security problems and finding several approaches for them. For example, they can be used to log Malicious activities in a compromised system; they can be also used to learn new threats for users and creating ideas how to get rid of those problems.

Honeypots are security resources that have no production value; no person or resource should be communicating with them. Any activity sent their way is suspect. Any traffic initiated by the honeypot means the system has most likely been compromised. Any traffic sent to the honeypot is most likely a probe, scan, or attack. With a honeypot, nothing is expected. To better understand the concepts of honeypots, let's take a look at the following example of honeypot deployments refer the figure.



The purpose here is to demonstrate to you that honeypots can come in many different flavors, and they can achieve different things. However, they are both honeypots because they share the same definition and concepts. With the intent using systems as a honeypots, to determine if there is any unauthorized activity happening within your DMZ. Honeypots passively capture any traffic or activity that interacts with them .

5.1.1Types of Honeypots:

There are two general types of honeypots:

5.1.1.1 Production honeypots are easy to use, capture only limited information, and are used primarily by companies or corporations. They are capturing a limited amount of information; mostly low interaction honeypots are used. security administrator watches the hacker’s movements carefully and tries to lower the risks that may come from it towards the company.

5.1.1.2 Research honeypots are complex to deploy and maintain, capture extensive information, and are used primarily by research, military, or government organizations. The objective is to learn how to Protect a system better, they do not bring any direct value to the security of an organization Honeypots are increasingly used to provide early warning of potential intruders, identify flaws in security strategies, and improve an organization's overall security awareness. “Honeypots can simulate a variety of internal and external devices, including Web servers, mail servers, database servers, application servers, and even firewalls. As a software development manager, we can regularly use honeypots to gain insight into vulnerabilities in both the software my team writes and the OS upon which we depend.” A honeypot is a security resource whose value lies in being probed, attacked, or compromised. This means that whatever we designate as a honeypot, it is our expectation and goal to have the system probed, attacked, and potentially exploited.

5.2 Legal issues with honeypots:

While deploying and start using a honeypot, there are some legal issues that a person should know about. Every country has different laws regarding to honeypot usage and information capturing. These regulations are related to data security, collection of data and finally how to use honeypots. All these different laws are based on the quality of the data that a honeypot can capture and a person who is deploying it. Privacy and data leads us to confidentiality term in network security. Our example is being a network administrator in a company.

5.3 Practical implementation:

We are starting with low interaction honeypot and then continue on a middle level of interaction to finally conclude with a high level of interaction.

- **Starting to honeypot:**

  We started with Honeyd as low level interaction honeypot and then we will move on medium level interaction honeypots. Every honeypot has specific and different attitudes. We will explain them one by one.

- **HoneyBOT is a medium interaction honeypot for windows.**

  A honeypot creates a safe environment to capture and interact with unsolicited and often malicious traffic on a network. HoneyBOT is an easy to use solution ideal for network security research or as part of an early warning IDS. The logging capability of a honeypot is far greater than any other network security tool and captures raw packet level data even including the keystrokes and mistakes made by hackers. The captured information is highly valuable as it contains only malicious traffic with little to no false positives. Honeypots are becoming one of the leading security tools used to monitor the latest tricks and exploits of hackers by recording their every move so that the security community can more quickly respond to new exploits.

- **How does it work?**

HoneyBOT works by opening a range of listening sockets on your computer which are designed to mimic vulnerable services. When an attacker connects to these services they are fooled into thinking they are attacking a real server. The honeypot safely captures all communications with the attacker and logs these results for future analysis. Should an attacker attempt an exploit or upload a rootkit or trojan to the server the honeypot environment can safely store these files on your computer for malware collection and analysis purposes. Following figure shows implementation of honeypot.

- **Installing and Securing Your Honeypot:**

A honeypot is intentionally put in harms way so it is critical to carry out some security precautions on your honeypot computer before deployment on any network. Install HoneyBOT on a dedicated computer or virtual machine. Update the operating system with security updates and use an antivirus product. You want your honeypot to be as free as possible from legitimate traffic so in broad terms we can consider any traffic to the honeypot to be malicious in nature. Remember that we are attracting attackers to intrude into this system so precautions are important.

- **Network Placement:**

If you place HoneyBOT inside the internal network where it is secured by perimeter defences it should never to be attacked. Any malicious traffic captured in this situation would indicate that another computer inside the network is already compromised or that the perimeter defences have been breached. In this configuration HoneyBOT is acting as an intrusion detection system. If you place

HoneyBOT on an external network or internet you will attract higher volumes of unsolicited network traffic. Direct internet placement is the most common setup with HoneyBOT being on the network DMZ.

- **Windows Services, SMB and NetBIOS**

You should disable any Windows services that are not required for the machine to operate as they offer an attacker a possible avenue of attack. HoneyBOT cannot listen on a port that is already in use by a Windows service. Some of the services that you may choose to disable include Messenger, ClipBook, COM+, FTP Publishing, SMTP, SNMP, TCP/IP NetBIOS Helper, Telnet, WWW Publishing. SMB (CIFS) provides name resolution, network browsing and printing services over TCP/IP. To disable SMB open the Network Connections window, right click the adapter and select Properties and uninstall Client For Microsoft Networks and File And Printer Sharing. SMB services may also be provided over NetBIOS (NBT). To disable NetBIOS open the Device Manager window, select Show Hidden Devices, expand Non-Plug And Play Drivers and disable NetBios Over Tcpip. If you are monitoring your honeypot via a remote desktop tool then you should change the default listening port to a random high numbered port.

Finally, before starting HoneyBOT take a baseline of the current listening services by opening a command shell and launching netstat with the -ano option. Any listening services that you are unable to disable need to be blocked at the firewall.

- **Firewall:**

A firewall will prevent unsolicited connections from reaching your computer. In order for HoneyBOT to communicate you need to customise your firewall rules to allow incoming connections. If you are using a software firewall you should create an exception for HoneyBOT.

- **HoneyBOT Options:**

Select Options from the View menu to configure HoneyBOT. Automatically Start Engine: The server engine will start automatically when the application is started. Enable Sound Alert: Plays a short sound each time an event occurs. Capture Binaries: If this option is enabled HoneyBOT will attempt to capture malware and other files and save them to the \HoneyBOT\Captures\ folder. If this option is enabled you should add an exception in your antivirus software to exclude this folder from its scan. Automatically Rotate Log: Each day at midnight HoneyBOT will save the current log file and start a new log file. Server Name: The alias name of the HoneyBOT server given to the remote machine.

- **Email Alerts** :

Enter your email address and SMTP server information to receive daily email updates from HoneyBOT.

- **Exports:**
  Select the Export Logs to CSV option to create a daily extract of your log file as a CSV file. Exported logs are saved in the \HoneyBOT\Logs\ folder. You can also choose to participate in the centralised log program and have your log files uploaded to the HoneyBOT website.
- **Syslog :**
  Select to send connection events to a Syslog server. Enter the Syslog server IP address and port.
- **Bindings:**
  Only applicable to multihomed machines. Provides support for multiple networks so HoneyBOT can bind to one or all detected networks. Enter the IP address that you want HoneyBOT to bind to. If the IP address is not valid and more than one IP address is available you will be prompted to select an address when the server engine starts.
- **Updates:**
  Select to have HoneyBOT check for updates on startup. There are two update types that may occur. A service update is a minor update to the server listening services, if a service update is available you will be prompted to install the update. An application update notification will occur if a new version of HoneyBOT is available.
- **Services and Profiles :**
  Select to edit the TCP and UDP services started by the HoneyBOT engine. You can add a new port, edit and disable an existing port, or delete the port configuration entirely. By default HoneyBOT will open more listening ports than a typical computer and this may alert an attacker to its presence. You can choose to limit your honeypot exposure to just a handful of ports that more closely resembles a real operating system. By loading a profile you can quickly emulate common operation system setups like an SQL Server, IIS Server, Exchange Server, etc.
- **Whitelist :**
  You may find HoneyBOT is interacting with services on your network that are

legitimate and not a cause for alarm. You can whitelist the source machine by adding the IP and port to the whitelist settings. When a machine is whitelisted HoneyBOT will no longer accept connections from that machine.

- **Debug :**

  The debug window will display application messages and socket events that occur during typical application operation.

- **Event Navigation:**

  The event tree on the left shows the ports that have been probed and remote addresses that have connected to HoneyBOT. The event list at the top right will display all connection attempts including the attributes of the connection. The packet list at the bottom displays each packet transmitted and received between the remote machine and the HoneyBOT server. You can expand the event tree and filter the events displayed by selecting an item in the list.

5.4 Advantages of honeypots:

There are many security solutions available in the market. Anyone can browse the variety of choices through internet and find the most suitable solution for their needs. Honeypots can capture attacks and give information about the attack type and if needed, thanks to the logs, it is possible to see additional information about the attack. New attacks can be seen and new security solutions can be created by looking at them. More examinations can be obtained by looking at the type of the malicious behaviors. It helps to understand more attacks that may happen. Honeypots are not bulky in terms of capturing data. They are only dealing with the incoming malicious traffic. Therefore, the

information that has been caught is not as much as the whole traffic. Focusing only on the malicious traffic makes the investigation far easier.

5.5 Disadvantages of honey pots:

We can only capture data when the hacker is attacking the system actively. If he does not attack the system, it is not possible to catch information. If there is an attack occuring in another system, our honeypot will not be able to identify it. So, attacks not towards our honeypot system may damage other systems and cause big problems. There is fingerprinting disadvantage of honeypots. It is easy for an experienced hacker to understand if he is attacking a honeypot system or a real system. Fingerprinting allows us to distinguish between these two. It is a not a wanted result of our experiment. The honeypot may be used as a zombie to reach other systems and compromise them. This can be very dangerous.

5.6 Assignment Questions:

Q1.What is Honey Pot?

Q2.What are different types of Honey Pot?

Q3.What is Malware Honey Pot?

Q4. What is Database honey pot?

Q5.What is Honey nets?

Q6. Which are two popular reasons or goals behind setting up a Honey Pot?

Conclusion: Hence, we have successfully studied concept of Honeypot in which we have set different network setting and set different drivers to identify unauthenticated access in our system

Assignment 2
Captcha number.py

```
import random
username=input("Enter Username:")
pwd=input("Enter Password:")
if(username=='test' and pwd=='1234'):
    num=random.randint(1000,9999)
    print("Enter this 4 digit number:",num)
    user_num=int(input())
    if(num==user_num):
        print("Login Successfully")
    else:
        print("Login Failed")
else:
     print("Login Failed")
```

Captcha.py

```
from captcha.image import ImageCaptcha
image_info=ImageCaptcha(width=250,height=100)
captcha_text='snjb'
source=image_info.generate(captcha_text)
image_info.write(captcha_text,'abc.png')
```

Generate_verify_image_captcha.py

```
from tkinter import *
def addNumbers():
    a=u.get()
    b=p.get()
    ca=c.get()
    from captcha.image import ImageCaptcha
    image_info=ImageCaptcha(width=250,height=100)
    captcha_text='snjb'
    source=image_info.generate(captcha_text)
    image_info.write(captcha_text,'abc.png')
    if(a=='test' and b=='1234'):
         if(ca==captcha_text):
           messagebox.showinfo("showinfo", "login sucessufully")
         else:
             messagebox.showinfo("showinfo", "login Failed")
    else:
            messagebox.showinfo("showinfo", "login Failed")
root = Tk()
canvas = Canvas(root, width = 300, height = 300)
canvas.pack()
img = PhotoImage(file="abc.png")
canvas.create_image(20,20, anchor=NW, image=img)
l1 = Label(root, text = "username")
l1.config(font =("Courier", 14))
u= Entry(root)
l2 = Label(root, text = "password")
l2.config(font =("Courier", 14))
p = Entry(root)
```

```
l3 = Label(root, text = "captcha")
l3.config(font =("Courier", 14))
c =Entry(root)
b2 = Button(root, text = "Login",command = addNumbers)
l1.pack()
u.pack()
l2.pack()
p.pack()
l3.pack()
c.pack()
b2.pack()
mainloop()
```

Assignment 3
Tr.py

```
import os
import re

path="/home/lenovo/.local/share/Trash/files"
infopath="/home/lenovo/.local/share/Trash/info"
dirlist=os.listdir(path)
#list of file which present in file folder
directory=[]
popis=""
for fname in dirlist:
    directory.append(fname)
    popis=popis + " " + fname
print popis
fname=raw_input("\nEnter the file name which toyou want to recover")

a=open(infopath+"/"+fname+".trashinfo","r")

print a
#To open file which user have selected
#To read the path which user have selected
for line in a:
      if "Path=" in line:
               ab=re.findall(r'/.*',line)

print ab
#To remove the extra elements frm path
destipath=str(ab)
destipath= destipath.lstrip('[')
destipath=destipath.rstrip(']')
destipath=destipath[:-1]
destipath=destipath[1:]
print("destination path is"+ destipath)

#To open the file from trash
file1 = open(path+"/"+fname,"r")
print file1
#To create the file in home folder
file2 = open(destipath,"w")
```

```
print file2
#write data into file which is stored at home folder
file2.write(file1.read())
file1.close()
file2.close()
print "files is recovered to desination"
#remove file from trash
os.remove(path+"/"+fname)
print("remove from"+path+":"+fname)
#remove file from trashinfo
os.remove(infopath+"/"+fname+".trashinfo")
print("remove from"+infopath+":"+fname)
```

Assignment 4

Squidguard Install

```
sudo apt update
sudo apt -y install squid
sudo systemctl start squid
sudo systemctl enable squid
sudo systemctl status squid
```

```
sudo apt install apache2-utils

sudo touch /etc/squid/passwd
sudo chown proxy: /etc/squid/passwd
ls -l /etc/squid/passwd

sudo htpasswd /etc/squid/passwd tecmint

sudo vim /etc/squid/squid.conf
```

```
#search auth_param and pest lines

auth_param basic program /usr/lib/squid/basic_ncsa_auth
/etc/squid/passwd
auth_param basic children 5
auth_param basic realm Squid Basic Authentication
auth_param basic credentialsttl 2 hours
acl auth_users proxy_auth REQUIRED
http_access allow auth_users
```

```
#search localnet and pest line in last

acl localnet src 10.1.100.222
```

```
sudo systemctl restart squid
```

```
browser setting
10.1.100.222  port 3128

sudo cat /var/log/squid/access.log


    Squid configuration file: /etc/squid/squid.conf
    Squid Access log: /var/log/squid/access.log
    Squid Cache log: /var/log/squid/cache.log
```

Assignment 5

Honey Pot

Blog for ref.txt

https://blog.own.sh/low-interaction-honeypots-with-python/

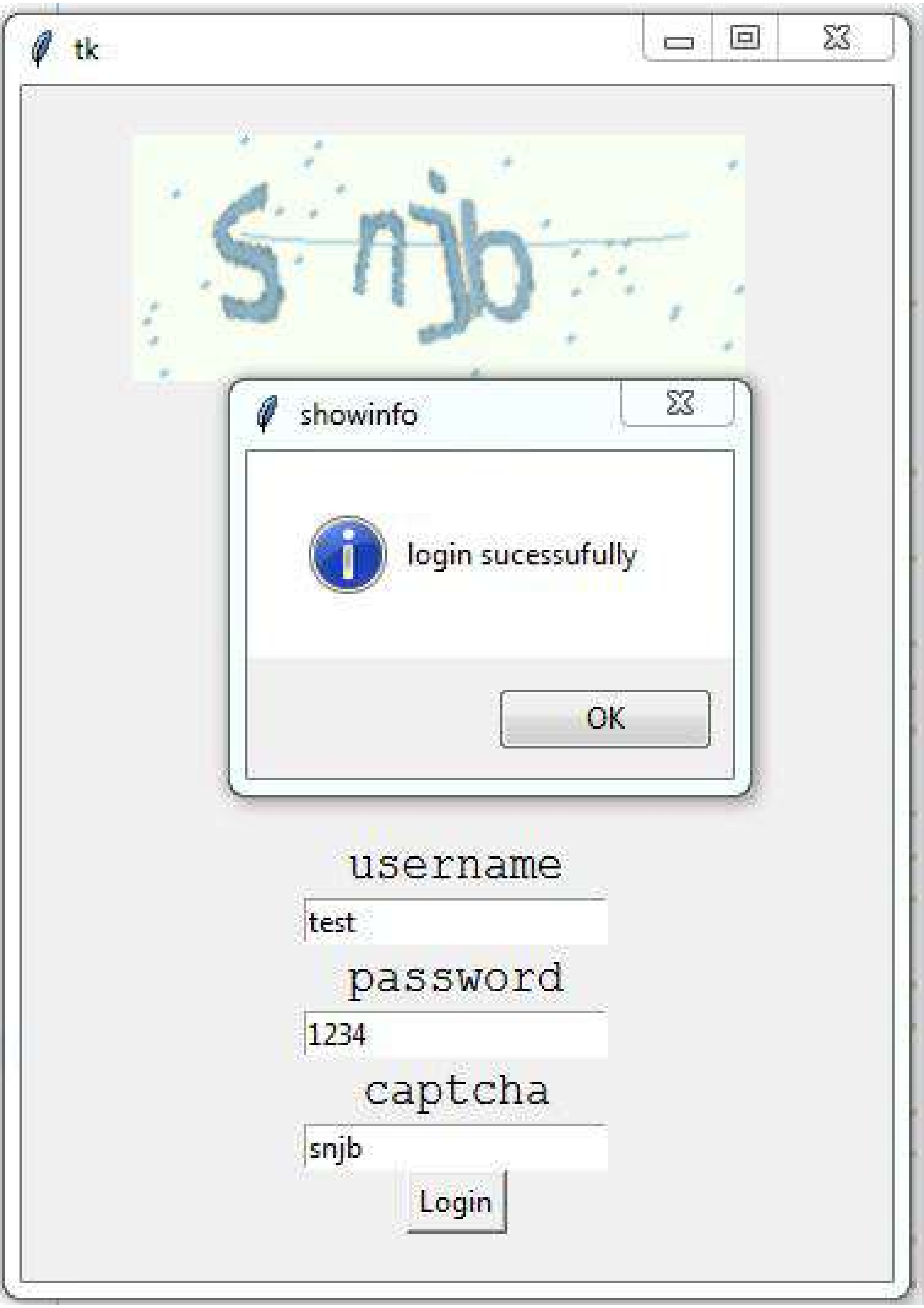
honey_pot.py

```
import socket
import atexit

# Local IP/Port for the honeypot to listen on (TCP)
LHOST = '0.0.0.0'
LPORT = 23

# Remote IP/Port to send the log data to (TCP)
RHOST = '192.168.1.210'
RPORT = 9000

# Banner displayed when connecting to the honeypot
BANNER = b'Ubuntu 14.04 LTS\nlogin:\n username: '

# Socket timeout in seconds
TIMEOUT = 10

def main():
    print ('[*] Honeypot starting on ' + LHOST + ':' + str(LPORT))
    atexit.register(exit_handler)
    listener.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    listener.bind((LHOST, LPORT))
    listener.listen(5)
    while True:
        (insock, address) = listener.accept()
        insock.settimeout(TIMEOUT)
        print ('[*] Honeypot connection from ' + address[0] + ':' +
str(address[1]) + ' on port ' + str(LPORT))
        try:
            insock.send(BANNER)
            data=[]
            user=""
            password=""
            while 1:
                user1=insock.recv(1024)
                user+=str(user1,"utf8")
                if str(user1,"utf8")=="\\":
                    break
            insock.send(b'\npassword: ')

            while 1:

                password1=insock.recv(1024)
                password+=str(password1,"utf8")
                if str(password1,"utf8")=="\\":
                    break
            string='\nAlert!!! Your action has been reported {} |
{}'.format(user,password)
            msg=bytes(string, 'utf-8')
            insock.send(msg)
            print("Username: {} \n Password:{}".format(user,password))

        except socket.error as e:
            sendLog(address[0],'Error: ' + str(e))
        else:
            sendLog(address[0],data)
        finally:
            insock.close()

def sendLog(fromip, message):
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect((RHOST, RPORT))
    s.send('IP:' + fromip + ' Port:' + str(LPORT) + ' | ' +
message.replace('\r\n', ' '))
    s.close()

def exit_handler():
    print ('\n[*] Honeypot is shutting down!')
    listener.close()

listener = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
if __name__ == '__main__':
    try:
        main()
    except KeyboardInterrupt:
        pass
```

steps_program_output.txt

```
Honey Pot

Steps for Execution

1.Run the prgram on anaconda spider
2.Open windows command window
3.Type C:\Users\lenovo> telnet localhost
4.Then remote user will be connect with localhost
5. In banner login: username:rajnor\
6.password:snjbcomp\
7.then user what is enter on banner are recorded at host side
```

```
#Program
import socket
import atexit

# Local IP/Port for the honeypot to listen on (TCP)
LHOST = '0.0.0.0'
LPORT = 23

# Remote IP/Port to send the log data to (TCP)
RHOST = '192.168.1.210'
RPORT = 9000

# Banner displayed when connecting to the honeypot
BANNER = b'Ubuntu 14.04 LTS\nlogin:\n username: '

# Socket timeout in seconds
TIMEOUT = 10

def main():
    print ('[*] Honeypot starting on ' + LHOST + ':' + str(LPORT))
    atexit.register(exit_handler)
    listener.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    listener.bind((LHOST, LPORT))
    listener.listen(5)
    while True:
        (insock, address) = listener.accept()
        insock.settimeout(TIMEOUT)
        print ('[*] Honeypot connection from ' + address[0] + ':' +
str(address[1]) + ' on port ' + str(LPORT))
        try:
            insock.send(BANNER)
            data=[]
            user=""
            password=""
            while 1:
                user1=insock.recv(1024)
                user+=str(user1,"utf8")
                if str(user1,"utf8")=="\\":
                    break
            insock.send(b'\npassword: ')

            while 1:

                password1=insock.recv(1024)
                password+=str(password1,"utf8")
                if str(password1,"utf8")=="\\":
                    break
            string='\nAlert!!! Your action has been reported {} |
{}'.format(user,password)
            msg=bytes(string, 'utf-8')
            insock.send(msg)
            print("Username: {} \n Password:{}".format(user,password))

        except socket.error as e:
            sendLog(address[0],'Error: ' + str(e))
        else:
            sendLog(address[0],data)
        finally:
            insock.close()

def sendLog(fromip, message):
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect((RHOST, RPORT))
    s.send('IP:' + fromip + ' Port:' + str(LPORT) + ' | ' +
message.replace('\r\n', ' '))
    s.close()

def exit_handler():
    print ('\n[*] Honeypot is shutting down!')
    listener.close()

listener = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
if __name__ == '__main__':
    try:
        main()
    except KeyboardInterrupt:
        pass

#Out Put

runfile('D:/LP IV 2022-23/5Honey Pot/Honey_Pot.py', wdir='D:/LP IV
2022-23/5Honey Pot')
[*] Honeypot starting on 0.0.0.0:23
[*] Honeypot connection from 127.0.0.1:50677 on port 23
Username: rajnor\
 Password:snjbcomp\
```



login sucessufully

username

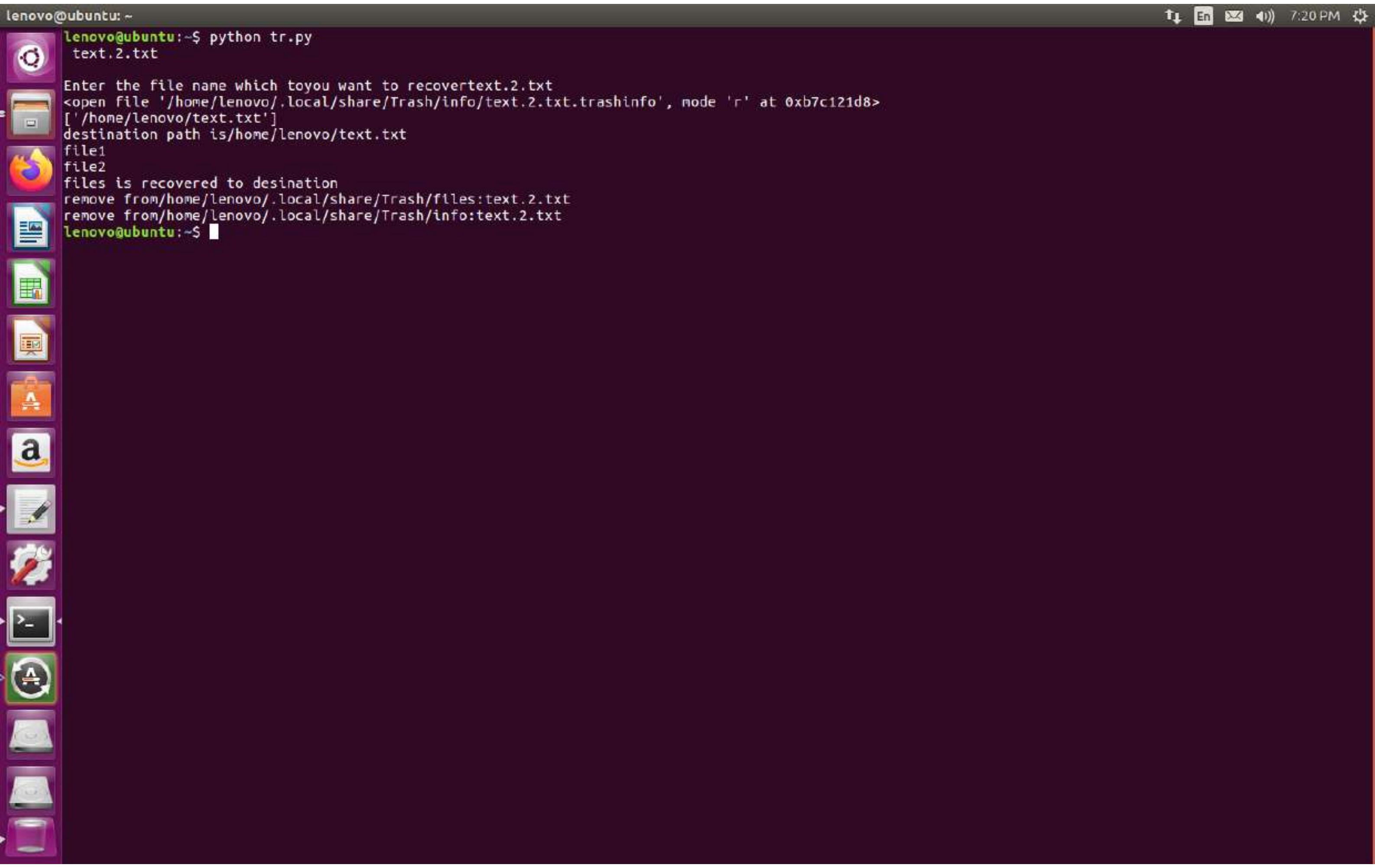
test

password

1234

captcha

snjb



```
lenovo@ubuntu:~$ python tr.py
 text.2.txt

Enter the file name which toyou want to recovertext.2.txt
<open file '/home/lenovo/.local/share/Trash/info/text.2.txt.trashinfo', mode 'r' at 0xb7c121d8>
['/home/lenovo/text.txt']
destination path is/home/lenovo/text.txt
file1
file2
files is recovered to desination
remove from/home/lenovo/.local/share/Trash/files:text.2.txt
remove from/home/lenovo/.local/share/Trash/info:text.2.txt
lenovo@ubuntu:~$
```

410245(D) Software Testing and Quality Assurance

Group 1

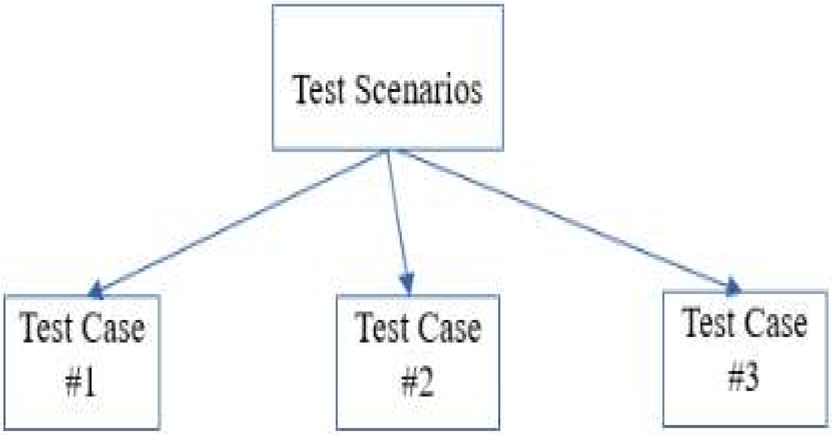
EXPERIMENT NO: 1

Problem Statement:

Write TEST Scenario for Gmail Login Page.

Theory:

Test Scenario is the function that can be tested. It is also called **Test Condition** or **Test Possibility**. A single Test Scenario can cover one or more Test Cases.



Test Scenario** is **'What to be tested'** and **Test Case** is **'How to be tested'.

As a tester, you can put yourself as an end-user and find real situations and possible functionalities of the application being tested.

Why Write Test Scenarios?

Test Scenarios are created for the following reasons:

- To ensure the completion of test coverage.
- Validate that the software is working properly for each Use Case.
- Improve the User Experience.
- Identify the most important end-to-end transactions.
- Study the terminal function and help to build test cases

Who and When Write Test Scenarios?

Typically, Testers are the ones who are responsible for creating Test Scenarios. In case of some complex and critical applications like Banking Software's, Business Analysts or Test Leads provide the Test Scenarios to Testers. But again depends, and vary from organization to organization.

As, Test Scenarios tell us what needs to be tested, therefore always written before test cases.

Steps to writing Test Scenarios:

As a tester, you can follow these five steps to create a test scenario:

1. Go through all the requirement documents available like BRD, SRS, and FSD to understand the functionalities of the application to be tested.
2. For each requirement, find out possible actions and goals of the user.
3. List down all the possible functionalities/scenarios that need to consider for each requirement. (Test Scenarios Template is provided to download)
4. Once all possible test scenarios are listed, create a Traceability Matrix to ensure that all requirements have a corresponding test scenario.
5. Review the Test Scenario document and Traceability Matrix with Test Lead / Business Analyst

Conclusion:

Test Scenario for Gmail Login page written successfully.

FAQ's

1. What is test scenarios examples?
2. What is scenario based questions in manual testing?
3. What are test scenarios for test cases?
4. How many test scenarios are there?
5. What do you understand by software testing?
6. When should you stop the testing process?
7. What do verification and validation mean in software testing?
8. Explain Test scenarios for Gmail Functionality

EXPERMIENT NO: 2

Problem Statement:

Write TEST Scenario for Inbox Functionality

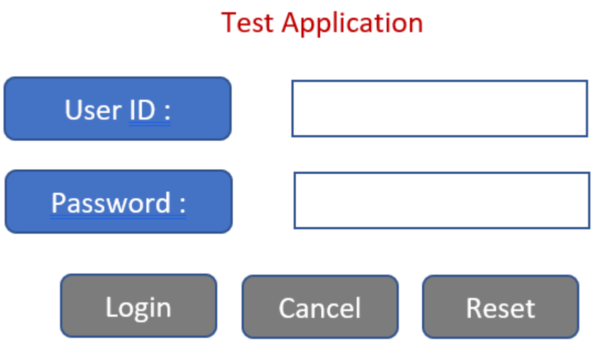
Theory:

Difference between Test Scenario & Test Case

TEST CASE	TEST SCENARIO
Answer "How to Test"	Answer "What to Test"
Test case includes test case name, precondition, test steps, expected result, and the actual result	A test scenario is a high-level documentation which will be associated with multiple test cases
Execute a set of steps to validate the test scenario	Validate functionality of a software application
Hard to maintain due to more in numbers	Easy to maintain due to its high-level design
Derived from Test Scenarios	Derived from Use Cases or requirement documents like BRD, SRS
Low-level actions	High-level actions
Written by Testers	Written by Test Leads, BAs or Testers
Details test cases will require more time and resource	If the test scenario is not detailed, may take time to discuss and understand

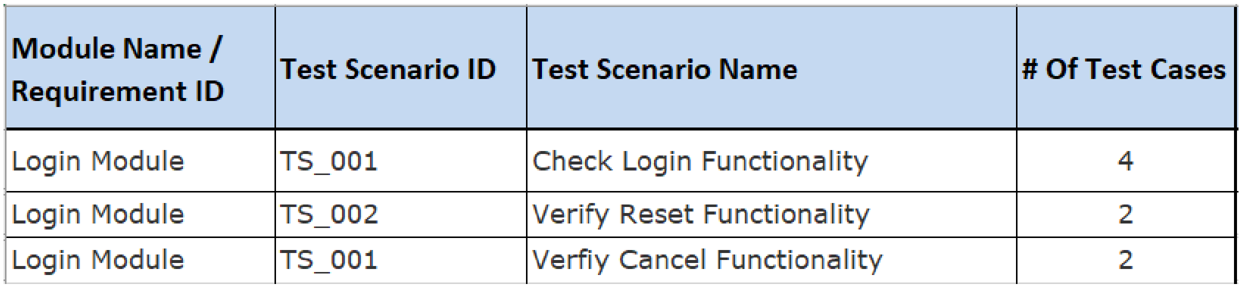
Sample Test Application:

Let's take an example of a Login page of a Test Application:



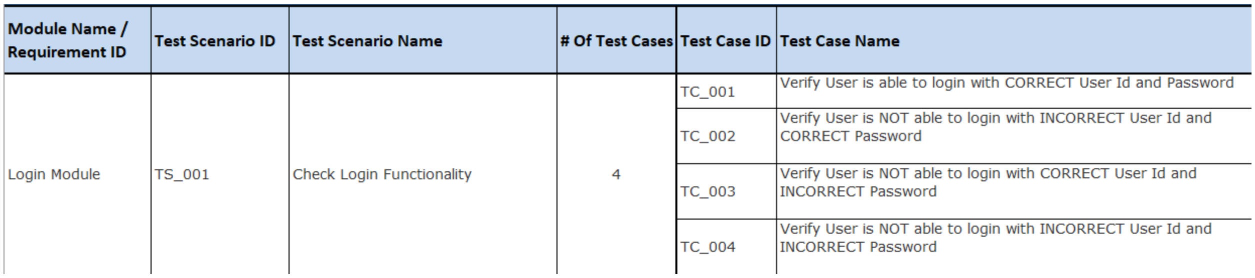
Here we can see **Login**, **Reset** and **Cancel** buttons on the page which will perform the respected functions when clicked.

So below can be the possible Test Scenarios for it:



Module Name / Requirement ID	Test Scenario ID	Test Scenario Name	# Of Test Cases
Login Module	TS_001	Check Login Functionality	4
Login Module	TS_002	Verify Reset Functionality	2
Login Module	TS_001	Verfiy Cancel Functionality	2

Here you can figure out the high-level Test Cases and provide the numbers AND/OR actual Test Case Names.



Module Name / Requirement ID	Test Scenario ID	Test Scenario Name	# Of Test Cases	Test Case ID	Test Case Name
Login Module	TS_001	Check Login Functionality	4	TC_001	Verify User is able to login with CORRECT User Id and Password
				TC_002	Verify User is NOT able to login with INCORRECT User Id and CORRECT Password
				TC_003	Verify User is NOT able to login with CORRECT User Id and INCORRECT Password
				TC_004	Verify User is NOT able to login with INCORRECT User Id and INCORRECT Password

Conclusion:

Test Scenario for Inbox Functionality written successfully.

FAQ's

1. Explain Test scenarios for Inbox Functionality.
2. What is test scenarios examples?
3. What is scenario based questions in manual testing?
4. What are test scenarios for test cases?
5. What is static testing?
6. Define black-box testing.

EXPERIMENT NO 3:

Problem Statement:

Write Test cases in excel sheet for Social Media application or website

Theory:

What is Test Case?

The test case is a set of actions to take to verify if the function of the software application is working as expected.

TCs are designed and executed to find defects in the software application in the early stages. So the defects can be fixed and the quality of software application can be improved. For this reason, preparing the test case as early as possible in the software development process is preferred.

Usually, testers write the test cases (except Unit Tests which are written and executed by developer). In a typical scenario when developers are developing a software application, testers are designing Test Scenarios, Test Cases and creating Test Data.

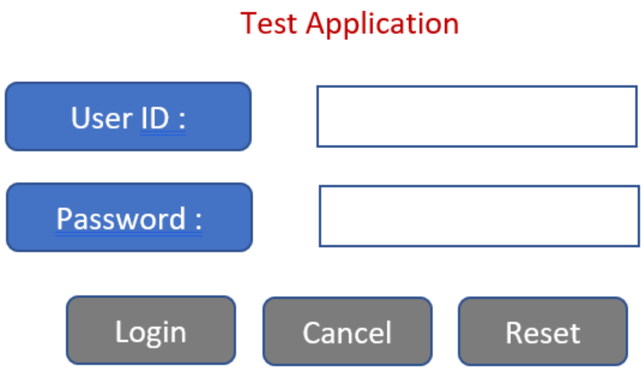
Once Development is completed, testers execute the TCs to test software applications.

Main Components of Test Case:

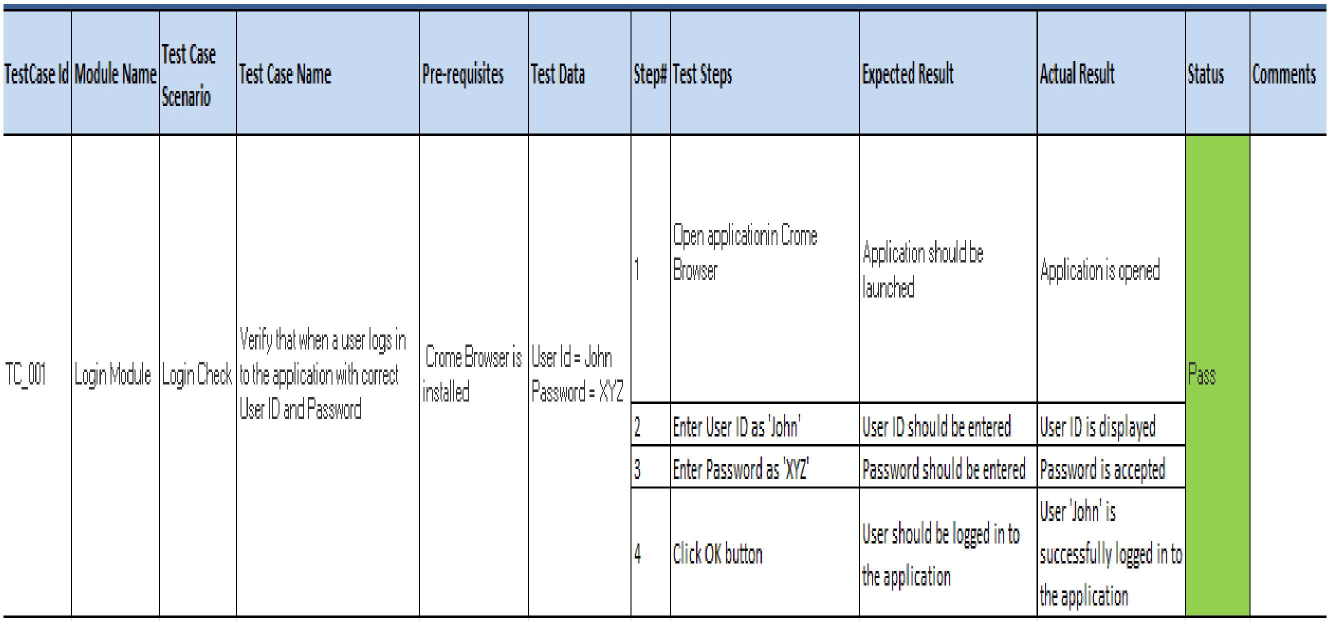
- **Test Case ID:** the unique value to identify the TC or sometimes to determine the number of TCs.
- **Module Name:** Module or Function Name which will be tested by TC.
- **TC Scenario:** A functionality of module to test
- **Test Case Name:** Unique name for TC to know what it verifies.
- **Pre-requisite:** any special setup is needed to be done before running TC
- **Test Data:** the data that needs to be prepared for testing
- **Test Steps:** describe the steps to perform tests
- **Expected Results:** expected results from the steps performed above
- **Actual Result:** usually it will be pass or fail
- **Status:** After execution, if TC is Pass / Failed or On Hold for some reason
- **Comments:** This column is used to note related information when performing test cases.
- **Test Executed By:** Name of the tester who executed the TC

Sample Test Application:

Let’s take a below Test Application



And below is how we can write TCs for Test Scenario of ‘Login Functionality’



TestCase Id	Module Name	Test Case Scenario	Test Case Name	Pre-requisites	Test Data	Step#	Test Steps	Expected Result	Actual Result	Status	Comments
TC_001	Login Module	Login Check	Verify that when a user logs in to the application with correct User ID and Password	Crome Browser is installed	User Id = John Password = XYZ	1	Open applicationin Crome Browser	Application should be launched	Application is opened	Pass	
						2	Enter User ID as 'John'	User ID should be entered	User ID is displayed		
						3	Enter Password as 'XYZ'	Password should be entered	Password is accepted		
						4	Click OK button	User should be logged in to the application	User 'John' is successfully logged in to the application		

Some Sample TCs:

Here are some sample TCs we can think of for Login Functionality:

1. Check the case of only entering the [User ID] field but not the password
2. Check the case of only entering the [Password] field but not the [User ID]
3. Verify login function for correct input [User ID] and [Password]
4. Check the login function in case of incorrect entry 3 times [User ID] and [Password]
5. Check the [User ID]/ [Password] input for more than allowed characters
6. Verify the case of entering special characters into [User ID]
7. Check the input field [User ID] with leading spaces.
8. Check the case of [Password] with leading spaces.
9. Verify the data displayed in the Password textbox must be a star or dot
10. Test if the OK button works as expected
11. Verify if Cancel button works as expected
12. Verify if Reset button works as expected

What is Test Script

In Automated Testing, testers write code/test scripts to automatically execute tests. Testers will use automation tools to build test scenarios and verify that the software works properly. The goal is to complete the test in a shorter time.

So test script is a set of instructions that will be performed on the application using an automation tool to verify if application functionalities work as expected.

Difference between Test Cases & Test Scripts

Test Case is a term used in manual testing and a test script used in an automation Testing.

The only difference between these two concepts is, Test cases are written and executed manually, while Test Scripts are full script/code that QA/testers write to execute using a test automation tool.

Conclusion:

Test Cases for Social Media Application written successfully.

FAQ’s

1. What are some examples of test cases?
2. How many test cases can execute per day?
3. What is a good test case in QA?
4. Who writes test cases in Agile?
5. What are test case quality factors?
6. What makes a test case a good test case?
7. Can we write test cases without test scenarios?
8. How many test cases are possible?
9. Which test cases consider as good test cases?

EXPERIMENT NO 4:

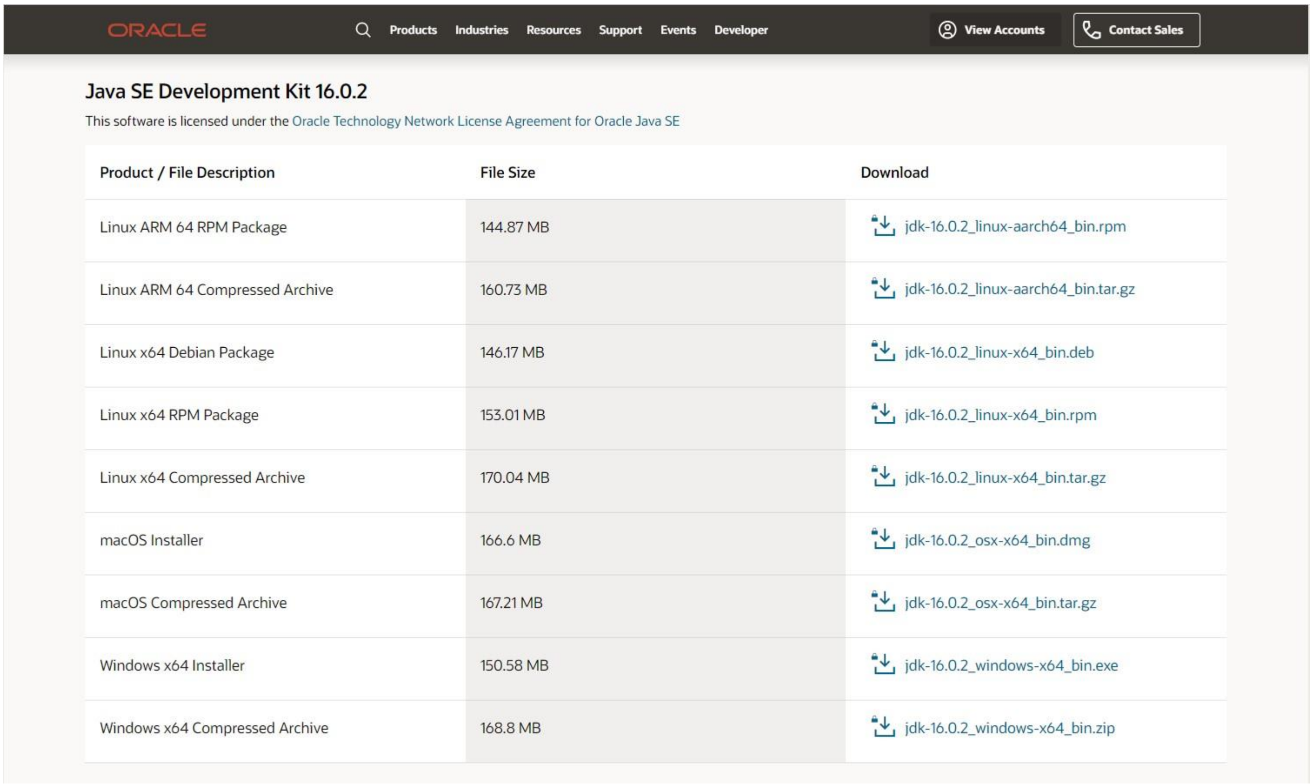
Problem Statement

Installation of Selenium Grid and Selenium Web driver & Java Eclipse (Automation Tools).

Theory

Prerequisites for configuring Selenium in Eclipse

Install java: Download Java SE Development Kit 16.0.2 according to the Windows, Linux, or macOS platform



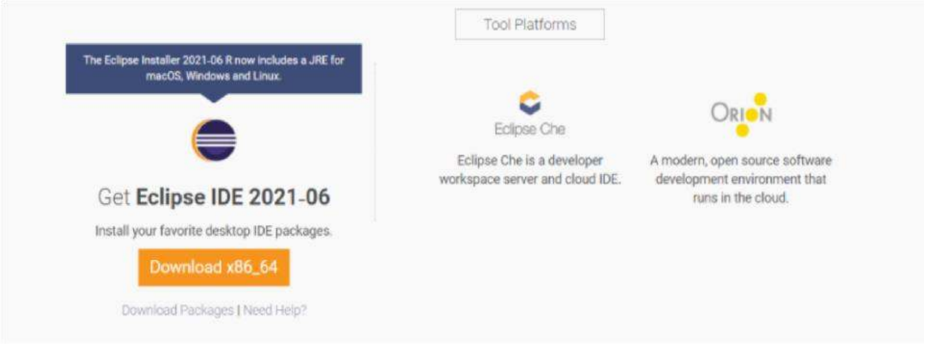
Java SE Development Kit 16.0.2

This software is licensed under the Oracle Technology Network License Agreement for Oracle Java SE

Product / File Description	File Size	Download
Linux ARM 64 RPM Package	144.87 MB	jdk-16.0.2_linux-aarch64_bin.rpm
Linux ARM 64 Compressed Archive	160.73 MB	jdk-16.0.2_linux-aarch64_bin.tar.gz
Linux x64 Debian Package	146.17 MB	jdk-16.0.2_linux-x64_bin.deb
Linux x64 RPM Package	153.01 MB	jdk-16.0.2_linux-x64_bin.rpm
Linux x64 Compressed Archive	170.04 MB	jdk-16.0.2_linux-x64_bin.tar.gz
macOS Installer	166.6 MB	jdk-16.0.2_osx-x64_bin.dmg
macOS Compressed Archive	167.21 MB	jdk-16.0.2_osx-x64_bin.tar.gz
Windows x64 Installer	150.58 MB	jdk-16.0.2_windows-x64_bin.exe
Windows x64 Compressed Archive	168.8 MB	jdk-16.0.2_windows-x64_bin.zip

Run the JDK Installer by double clicking on the file name in the download location and following the instruction wizard. Alternatively, silently install JDK by entering the following command: **jdk.exe /s**

Install Eclipse IDE



The Eclipse Installer 2021-06 R now includes a JRE for macOS, Windows and Linux.

Get **Eclipse IDE 2021-06**

Install your favorite desktop IDE packages.

Eclipse Che

Eclipse Che is a developer workspace server and cloud IDE.

ORION

A modern, open source software development environment that runs in the cloud.

Install Selenium Grid

Download and install Selenium to be set up in Eclipse.

Install Selenium Driver

For Cross Browser Testing, download the relevant Browser Driver – Chrome Driver (for Chrome), Gecko Driver (for Firefox), Safari Driver(for Safari), and Internet Explorer Driver and MS Edge Driver (IE and Edge respectively). Place these Browser Driver files in a directory that is part of the environment PATH. This will allow a command-line call to the programs to execute them irrespective of the working directory.

How to configure Selenium in Eclipse

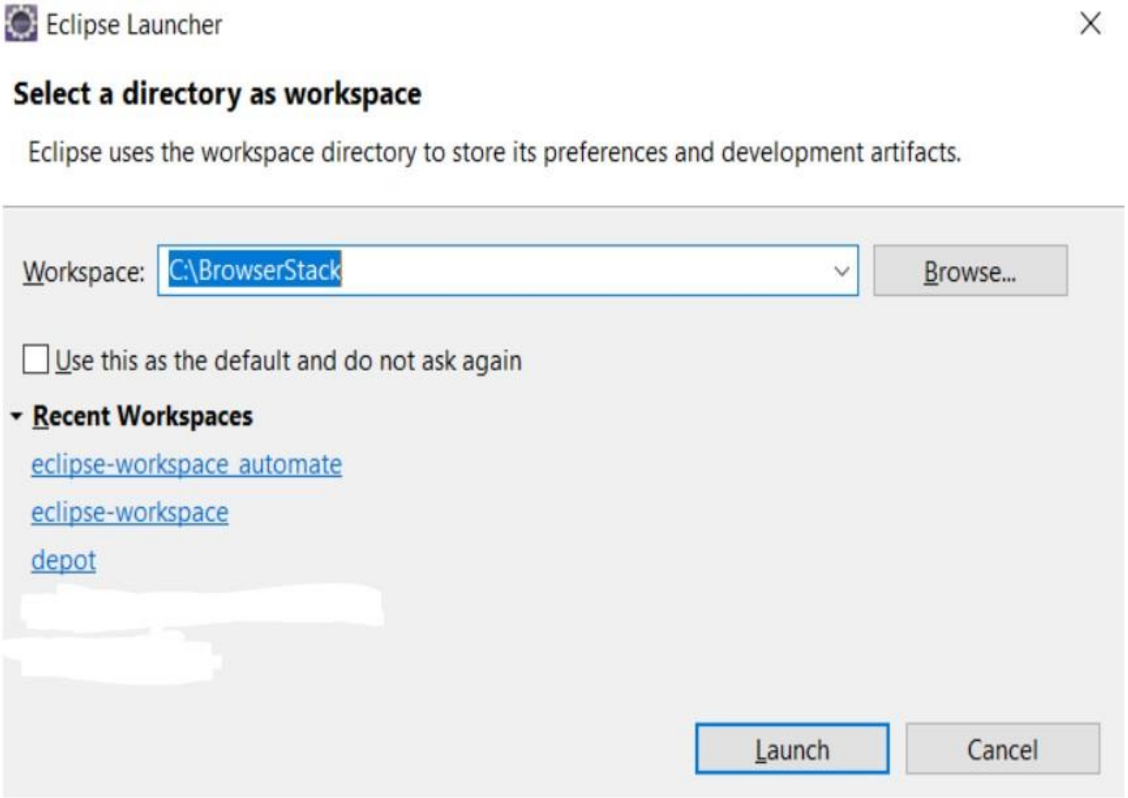
Here are the steps to configure Selenium Web driver with Eclipse:

Step 1: Launch Eclipse

To launch Eclipse double click on the **eclipse.exe** file in the download location.

Step 2: Create Workspace in Eclipse

This workspace named "**C:\BrowserStack**" is like any other folder, which will store all the test scripts. Launch the BrowserStack workspace.



Eclipse Launcher

Select a directory as workspace

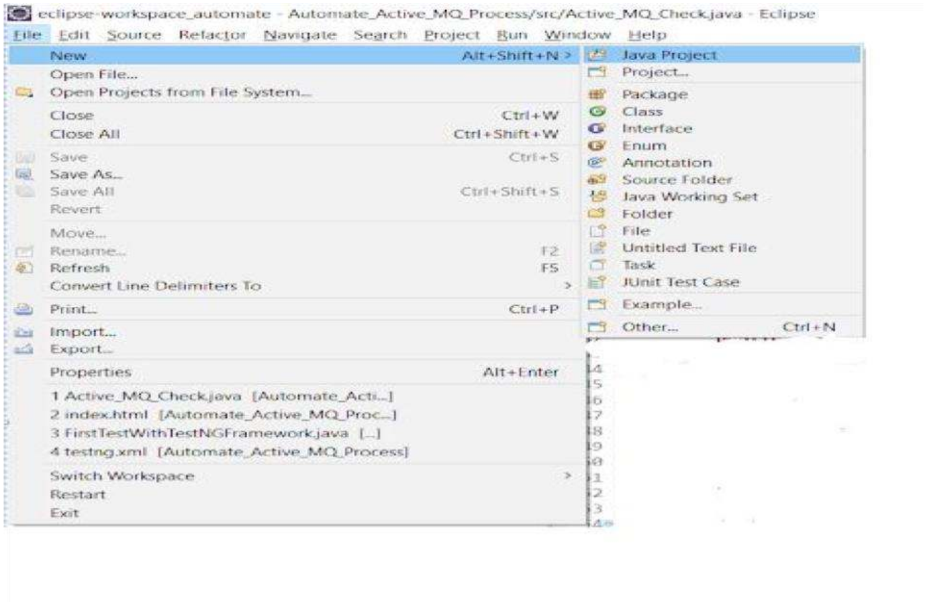
Eclipse uses the workspace directory to store its preferences and development artifacts.

Workspace: C:\BrowserStack

□ Use this as the default and do not ask again

Step 3: Create New Java Project in the BrowserStack Workspace

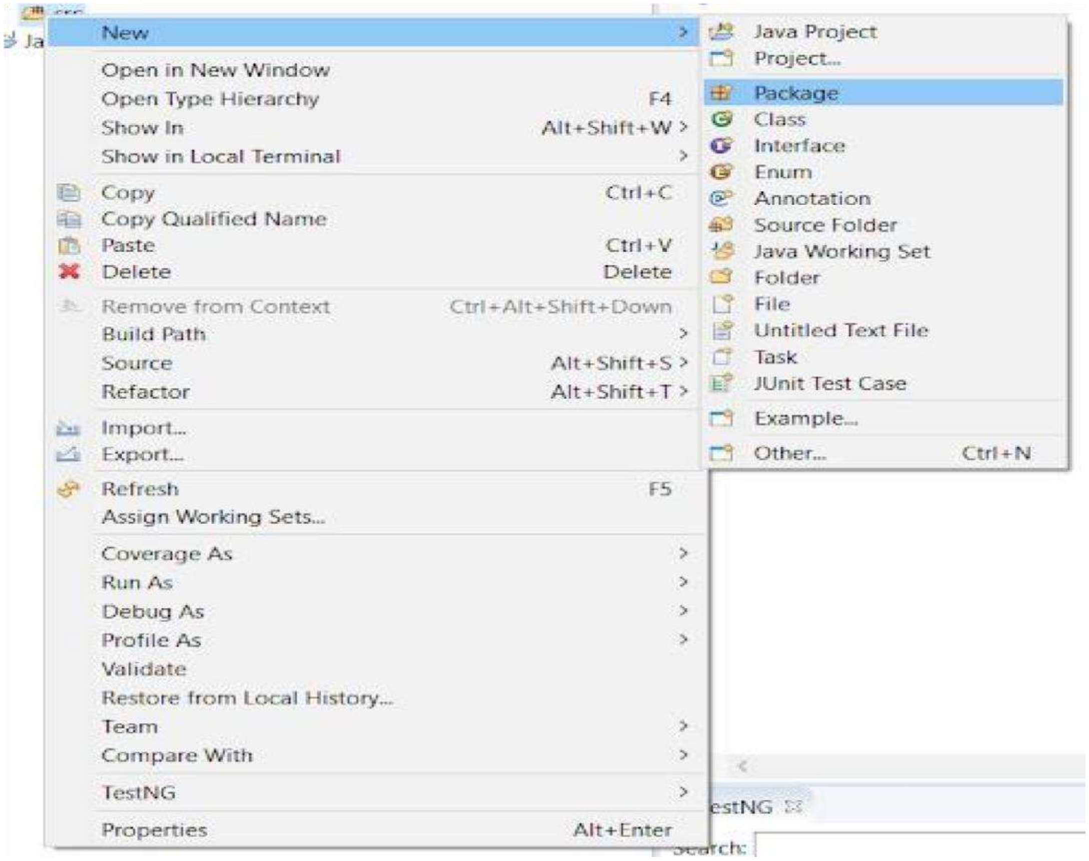
Create a new Java Project by clicking on **File > New > Java Project** and name it.


eclipse-workspace_automate - Automate_Active_MQ_Process/src/Active_MQ_Check.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
New Alt+Shift+N >
Open File...
Open Projects from File System...
Close Ctrl+W
Close All Ctrl+Shift+W
Save Ctrl+S
Save As...
Save All Ctrl+Shift+S
Revert
Move...
Rename... F2
Refresh F5
Convert Line Delimiters To >
Print... Ctrl+P
Import...
Export...
Properties Alt+Enter
1 Active_MQ_Check.java [Automate_Acti...]
2 index.html [Automate_Active_MQ_Proc...]
3 FirstTestWithTestNGFramework.java [...]
4 testng.xml [Automate_Active_MQ_Process]
Switch Workspace >
Restart
Exit
Java Project
Project...
Package
Class
Interface
Enum
Annotation
Source Folder
Java Working Set
Folder
File
Untitled Text File
Task
JUnit Test Case
Example...
Other... Ctrl+N

Creating a new Java Project

Step 4: Create Package and Class under the Java Project

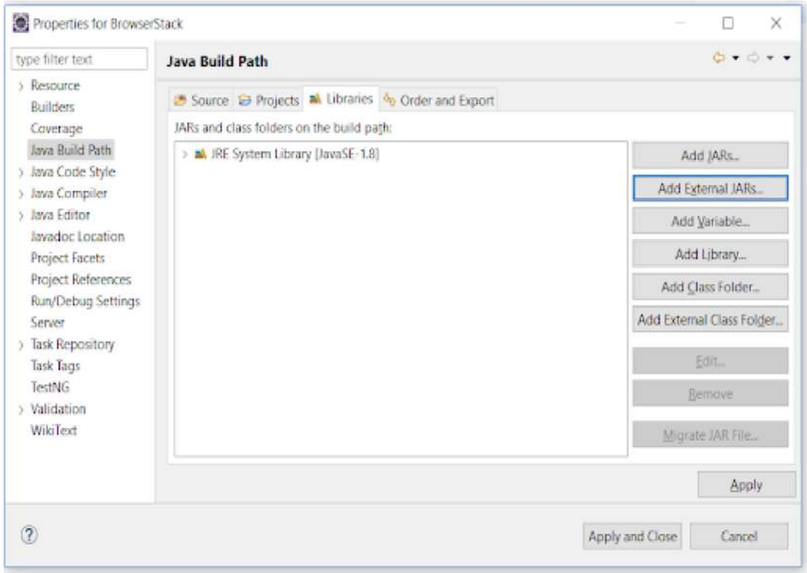
By clicking on the **src folder** (which is the source folder), create a new package and name it (BrowserStack). Then right-click on the package name and create a class.



Creating Class in the BrowserStack Package

Step 5: Add Selenium JARs to the Java Project in Eclipse

To add the Selenium Jars to the BrowserStack Java right click on the BrowserStack Project folder and select the **Properties** option. In the properties window, click on the **Java Build Path** and **Add External JARs**. Browse and add the downloaded Selenium JARs i.e. Client Combined JAR and all the JARs under the Libs folder, then click **Apply and Close**.



Adding Selenium JARs in the BrowserStack Project

This configures Selenium with Eclipse, making it ready to execute the first test script.

<u>Conclusion</u>

Installation of automation tools is done successfully.

<u>FAQ’s</u>

1. What is automation testing?
2. What are the types of automation testing?
3. What's the difference between manual testing and automated testing?
4. When is a good time to automate a test?
5. When will you avoid automated testing?
6. How do you choose a tool/framework for automated testing?
7. Which tool is mostly used in automation testing?

8. How many types of tools are used in automation testing?

9. What is the crucial requirement for test automation tool?

10. When not to use automation tools for testing?

11. What we Cannot automate using Selenium?

12. What makes Selenium unique from other automation tools?

13. What are the tools required for Selenium automation?

14. What is the limitation of Selenium automation?

EXPERIMENT NO 5:

Problem Statement

Prepare Software requirement specification for any project or problem statement

Theory:

Software Requirement Specification (SRS) Format:

as name suggests, is complete specification and description of requirements of software that needs to be fulfilled for successful development of software system. These requirements can be functional as well as non-functional depending upon type of requirement. The interaction between different customers and contractor is done because its necessary to fully understand needs of customers. Depending upon information gathered after interaction, SRS is developed which describes requirements of software that may include changes and modifications that is needed to be done to increase quality of product and to satisfy customer's demand.

1. **Introduction:**
   - Purpose of this Document – At first, main aim of why this document is necessary and what's purpose of document is explained and described.
   - Scope of this document – In this, overall working and main objective of document and what value it will provide to customer is described and explained. It also includes a description of development cost and time required.
   - Overview – In this, description of product is explained. It's simply summary or overall review of product.
2. **General description :** In this, general functions of product which includes objective of user, a user characteristic, features, benefits, about why its importance is mentioned. It also describes features of user community.
3. **Functional Requirements:** In this, possible outcome of software system which includes effects due to operation of program is fully explained. All functional requirements which may include calculations, data processing, etc. are placed in a ranked order.
4. **Interface Requirements :** In this, software interfaces which mean how software program communicates with each other or users either in form of any language, code, or message are fully Described and explained. Examples can be shared memory, data streams, etc.

5. **Performance Requirements:** In this, how a software system performs desired functions under specific condition is explained. It also explains required time, required memory, maximum error rate, etc.
6. **Design Constraints:** In this, constraints which simply means limitation or restriction are specified and explained for design team. Examples may include use of a particular algorithm, hardware and software limitations, etc
7. **Non-Functional Attributes:** In this, non-functional attributes are explained that are required by software system for better performance. An example may include Security, Portability, Reliability, Reusability, Application compatibility, Data integrity, Scalability capacity, etc.
8. **Preliminary Schedule and Budget:** In this, initial version and budget of project plan are explained which include overall time duration required and overall cost required for development of project.
9. **Appendices :** In this, additional information like references from where information is gathered, definitions of some specific terms, acronyms, abbreviations, etc. are given and explained

Sample Software Requirements Specification points:

1) Introduction:

2) Overall description:

 2.1 Product Perspective

 2.2 Product Functions

 2.3 User Classes and Characteristics

 2.4 Operating Environment

 2.5 Design and Implementation Constraints

 2.6 User Documentation

 2.7 Assumptions and Dependencies

3) External Interface Requirements:

 3.1 User Interfaces

 3.2 Hardware Interfaces

 3.3 Software Interfaces

 3.4 Communication Interfaces

4) Functional Requirement Specifications (FRS):

 4.1 System Features

 4.2 Functional Requirements

5) Non Functional Requirements:

5.1 Usability Requirements

5.2 Performance Requirements

5.3 Compatibility Requirements

6) Other Requirements

7) Glossary

Conclusion: Prepared the software requirement specification document successfully.

FAQ's

1. Which part of SRS is more important?
2. What is the role of SRS in software life cycle?
3. Why SRS is called black box?
4. What is the need for SRS give significant components of SRS?
5. What questions to ask for requirements gathering?
6. In which phase SRS is prepared?
7. What are the qualities of a good SRS?
8. What are the major components of SRS?
9. What are functional requirements in SRS?