Institute of Distance and Open Learning

Vidya Nagari, Kalina, Santacruz East – 400098.

A Practical Journal Submitted in fulfillment

of the degree of

MASTER OF SCIENCE

IN

COMPUTER SCIENCE

YEAR 2023-24

Part II

Semester-3

Subject code – 90985 R

Subject Name - Cyber Information Security II

BY

Mr. Mohammed Maaz Shaikh

Application ID- 41775
Seat No - 4100058

Institute of Distance and Open Learning
(IDOL)

University of Mumbai

Certificate

This is to certify that **Mr. Mohammed Maaz Shaikh** student of Masters of Computer Science, Part 2, Semester 3 has completed the specified term work in the subject of **Cyber Information Security II** in satisfactorily manner within this institute as laid down by University of Mumbai during the academic year 2024 to 2025.

M.Sc. - CS Coordinator Examiner

Date: Guide

INDEX

Practical 1

Aim:- Write a program to take backup of mysql database

Code:

```
import java.io.IOException;

public class MySQLBackup {

    public static void main(String[] args) {
        String host = "localhost";
        String port = "3306";
        String databaseName = "your_database_name";
        String username = "your_username";
        String password = "your_password";
        String backupPath = "path_to_backup_directory/backup.sql";

        backupMySQLDatabase(host, port, databaseName, username, password, backupPath);
    }

    public static void backupMySQLDatabase(String host, String port, String databaseName,
String username, String password, String backupPath) {         try {
            String command = "mysqldump -h " + host + " -P " + port + " -u " + username +
" -p" + password + " " + databaseName + " -r " + backupPath;

            ProcessBuilder processBuilder = new ProcessBuilder("cmd.exe", "/c", command);
Process process = processBuilder.start();

            int exitCode = process.waitFor();
if (exitCode == 0) {
                System.out.println("Backup completed successfully.");
            } else {
                System.out.println("Backup failed. Please check the error message.");
            }
```

```
        } catch (IOException | InterruptedException e) {
    e.printStackTrace();
        }
      }
    }
```

Note :

Make sure to replace placeholders like "your_database_name", "your_username", "your_password", and "path_to_backup_directory" with your actual MySQL database details.

Output

If the backup is successful:

Backup completed successfully.

If there is an error during the backup process:

Backup failed. Please check the error message.

Practical 2

Aim:- Write a program to restore mysql database

Code :

```
import java.io.IOException;

public class MySQLRestore {

    public static void main(String[] args) {
        String host = "localhost";
        String port = "3306";
        String databaseName = "your_database_name";
        String username = "your_username";
        String password = "your_password";
        String backupPath = "path_to_backup_directory/backup.sql";

        restoreMySQLDatabase(host, port, databaseName, username, password, backupPath);
    }

    public static void restoreMySQLDatabase(String host, String port, String databaseName,
String username, String password, String backupPath) {         try {
            String command = "mysql -h " + host + " -P " + port + " -u " + username +
" -p" + password + " " + databaseName + " < " + backupPath;

            ProcessBuilder processBuilder = new ProcessBuilder("cmd.exe", "/c", command);
Process process = processBuilder.start();

            int exitCode = process.waitFor();
if (exitCode == 0) {
                System.out.println("Database restore completed successfully.");
            } else {
                System.out.println("Database restore failed. Please check the error message.");
            }
```

```
        } catch (IOException | InterruptedException e) {
    e.printStackTrace();
        }
      }
    }
```

Replace placeholders like "your_database_name", "your_username", "your_password", and "path_to_backup_directory" with your actual MySQL database details.

Output:

If the restore is successful:

Database restore completed successfully

If there is an error during the restore process:

Database restore failed. Please check the error message

Practical 3

Use DriveImage XML to image a hard drive

Note : This program assumes you have already installed DriveImage XML on your system.

```
import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;


public class DriveImageXMLExample {


   public static void main(String[] args) {

      // Replace these values with your actual DriveImage XML installation path and desired image file path

      String driveImageXMLPath = "C:\\Program Files\\Runtime Software\\DriveImage XML\\dixml.exe";

      String sourceDrive = "C:";  // Replace with the drive you want to image

      String destinationImagePath = "D:\\backup_image.xml";  // Replace with the desired image file path


      // Create a disk image         createDiskImage(driveImageXMLPath,

sourceDrive, destinationImagePath);
```

```
        // Restore from the disk image (optional)
        // restoreFromDiskImage(driveImageXMLPath, destinationImagePath, "E:"); // Replace with the destination drive
    }

    public static void createDiskImage(String driveImageXMLPath, String sourceDrive, String destinationImagePath) {
        try {
            String command = driveImageXMLPath + " /b /v /c /s" + sourceDrive + " /f" + destinationImagePath;

            ProcessBuilder processBuilder = new ProcessBuilder("cmd.exe", "/c", command);
            Process process = processBuilder.start();

            int exitCode = process.waitFor();
if (exitCode == 0) {
                System.out.println("Disk image created successfully.");
} else {
                System.out.println("Disk image creation failed. Please check the error message.");
printErrorStream(process);
            }

        } catch (IOException | InterruptedException e) {
            e.printStackTrace();
        }
    }

    public static void restoreFromDiskImage(String driveImageXMLPath, String sourceImagePath,
```

```
String destinationDrive) {
        try {
            String command = driveImageXMLPath + " /r /f" + sourceImagePath + " /s" + destinationDrive;

            ProcessBuilder processBuilder = new ProcessBuilder("cmd.exe", "/c", command);
            Process process = processBuilder.start();

            int exitCode = process.waitFor();
if (exitCode == 0) {
                System.out.println("Restoration from disk image completed successfully.");
            } else {
                System.out.println("Restoration from disk image failed. Please check the error message.");
printErrorStream(process);
            }

        } catch (IOException | InterruptedException e) {
            e.printStackTrace();
        }
    }

    private static void printErrorStream(Process process) throws IOException {
        try (BufferedReader reader = new BufferedReader(new
InputStreamReader(process.getErrorStream()))) {
            String line;
            System.out.println("Error message:");
while ((line = reader.readLine()) != null) {
                System.out.println(line);
            }
```

```
        }
    }
}
```

OUTPUT :

Disk image created successfully.

Practical 4

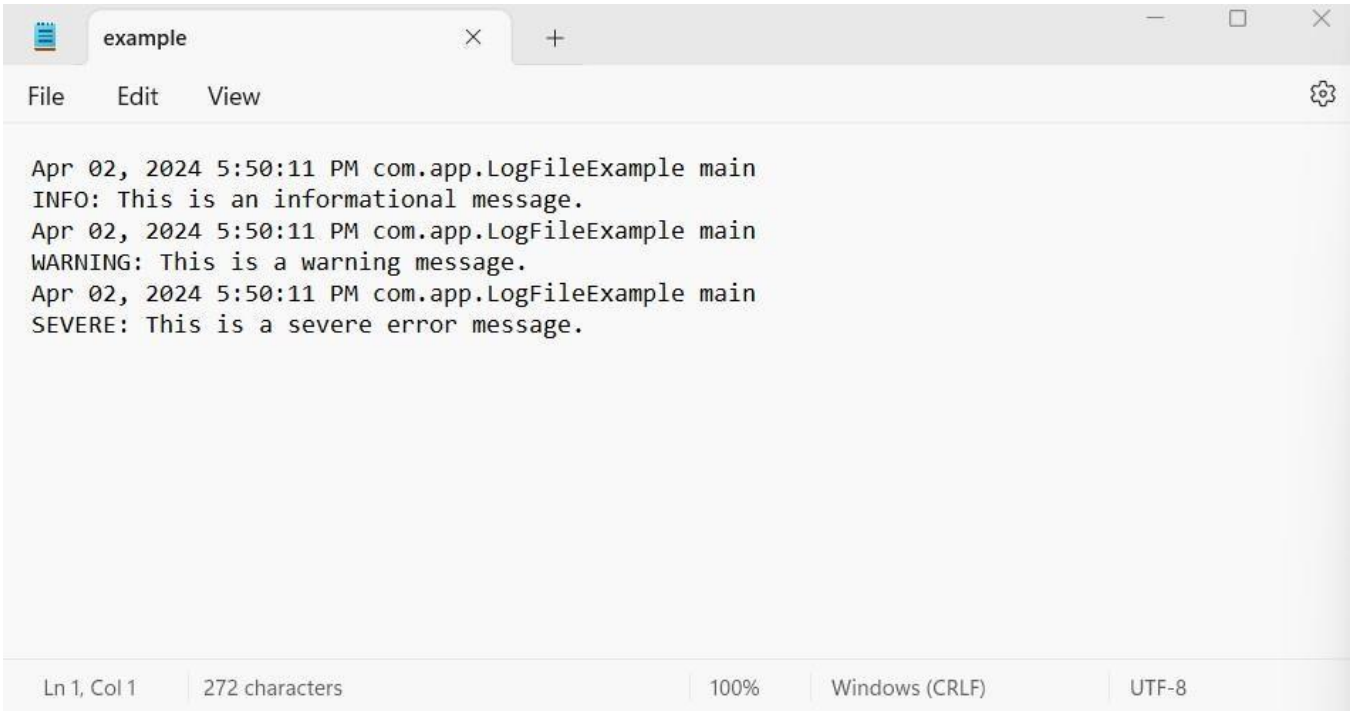
Write a program to create a log file.

Code :

```
import java.io.IOException; import
java.util.logging.FileHandler; import
java.util.logging.Level; import
java.util.logging.Logger; import
java.util.logging.SimpleFormatter;

public class LogFileExample {

   public static void main(String[] args) {
      Logger logger = Logger.getLogger(LogFileExample.class.getName());

      try {
         // Create a file handler to write log messages to a file
FileHandler fileHandler = new FileHandler("example.log");

         // Create a simple text formatter for the log records
SimpleFormatter simpleFormatter = new SimpleFormatter();
fileHandler.setFormatter(simpleFormatter);


         // Add the file handler to the logger
logger.addHandler(fileHandler);


         // Set the logging level (FINEST, FINER, FINE,
INFO, WARNING, SEVERE)
logger.setLevel(Level.INFO);

         // Log some sample messages logger.info("This is an
informational message."); logger.warning("This is a warning
message."); logger.severe("This is a severe error message.");

         } catch (IOException e) { e.printStackTrace();
         }
         }
         }
```

Note : Compile and run this program, and you should see a new file (`example.log`) created in the project directory with the logged messages.

OUTPUT :



Practical 5

Write a program to find a file in a directory

Code :

a)

```
// Java Program to Search for a File in a Directory  import
java.io.*;

// MyFilenameFilter class implements FilenameFilter
// interface
class MyFilenameFilter implements FilenameFilter {

	String initials;

	// constructor to initialize object
	public MyFilenameFilter(String initials)
	{
		this.initials = initials;
	}

	// overriding the accept method of FilenameFilter
	// interface
	public boolean accept(File dir, String name)
	{
		return name.startsWith(initials);
	}
}

public class Main {

	public static void main(String[] args)
	{
		// Create an object of the File class
		// Replace the file path with path of the directory
	File directory = new File("/home/user/");

		// Create an object of Class MyFilenameFilter
		// Constructor with name of file which is being
		// searched
		MyFilenameFilter filter
```

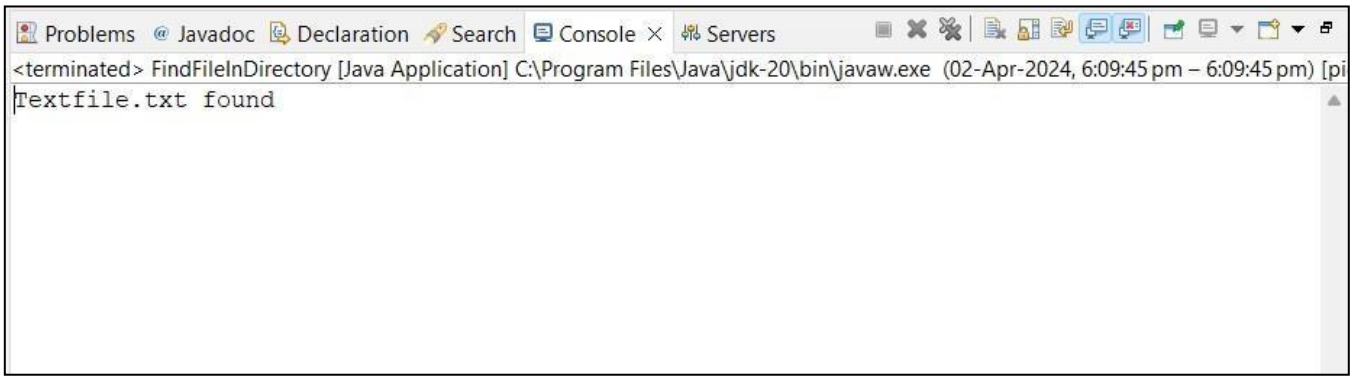
```
                        = new MyFilenameFilter("file.cpp");

                // store all names with same name
                // with/without extension
                String[] flist = directory.list(filter);

                // Empty array
                if (flist == null) {
                        System.out.println(
                                "Empty directory or directory does not exists.");
                }
                else {

                        // Print all files with same name in directory
                        // as provided in object of MyFilenameFilter
                        // class
                       for (int i = 0; i < flist.length; i++) {
    System.out.println(flist[i]+" found");
                        }
                }
        }
}
```

b)

```
// Java Program to Search for a File in a Directory  import
java.io.File;

public class Main {

        public static void main(String[] argv) throws Exception
        {
                // Create an object of the File class
               // Replace the file path with path of the directory
    File directory = new File(""D:/New folder")

                // store all names with same name
                // with/without extension
String[] flist = directory.list();    int flag =
0;    if (flist == null) {
                        System.out.println("Empty directory.");
```

```
                }
                        else {// Linear search in the
                arrayfor (int i = 0; i < flist.length;
                i++) {
                                        String filename = flist[i];
        if (filename.equalsIgnoreCase("Textfile.txt")) {
                System.out.println(filename + " found");
                                                flag = 1;
                                        }
                                }
                }

                if (flag == 0) {
                        System.out.println("File Not Found");
                }
        }
}
```

OUTPUT



Practical 6

Write a program to find a word in a

file Code : import

java.io.IOException; import

java.nio.file.Files; import

java.nio.file.Path; import

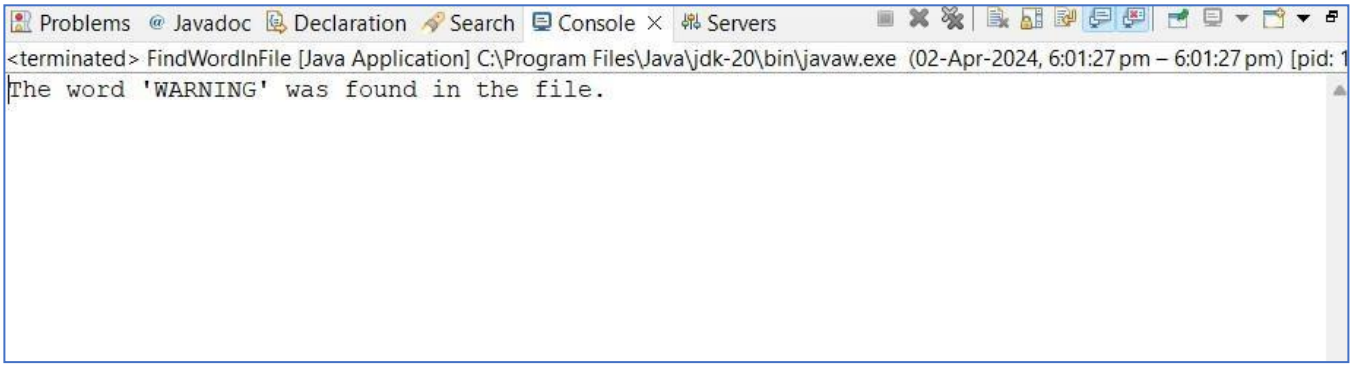
java.nio.file.Paths; import

java.util.Scanner;

```
public class FindWordInFile {

    public static void main(String[] args) {
        // Replace "path/to/your/file.txt" with the actual path to your text file
        String filePath = "D:/eclipse-workspace/MyJavaApp/example.log";
String targetWord = "WARNING";
        try {
            boolean wordFound = searchWordInFile(filePath, targetWord);

            if (wordFound) {
                System.out.println("The word '" + targetWord + "' was found in the file.");
} else {
                System.out.println("The word '" + targetWord + "' was not found in the file.");
            }
        } catch (IOException e) {
            e.printStackTrace();
```

```
        }

    }

    public static boolean searchWordInFile(String filePath, String targetWord) throws IOException {
Path path = Paths.get(filePath);
        // Check if the file exists

if (!Files.exists(path)) {

            System.out.println("File not found: " + filePath);

return false;

        }
        try (Scanner scanner = new Scanner(path)) {

// Iterate through each line of the file            while

(scanner.hasNextLine()) {

                String line = scanner.nextLine();

                // Check if the target word is present in the line

if (line.contains(targetWord)) {                    return true;

// Word found

                }

            }

        }

        return false;  // Word not found

    }

}
```

OUTPUT



Practical 7

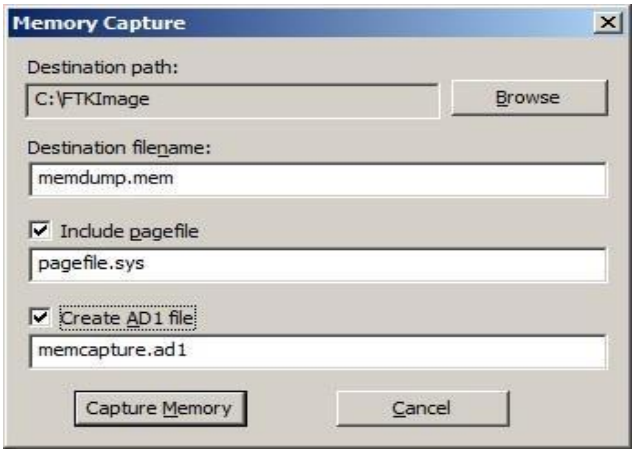
Create forensic images of digital devices from volatile data such as memory using Imager for: (i) Computer System; (ii) Server; (iii) Mobile Device

Using FTK Imager:

FTK Imager is a graphical tool that provides a user-friendly interface for creating forensic images. It's available for Windows and Linux.

Download and install FTK Imager: AccessData FTK Imager.

Open FTK Imager and go to File > Capture Memory.



Choose the target device, set the destination path, and click Start to create a memory image.

Practical 10

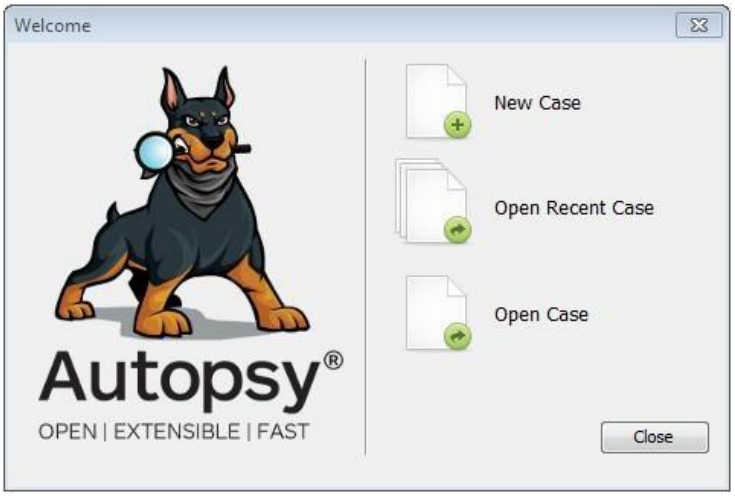
Create a new investigation case using Forensic Tool: (i) Computer System; (ii) Computer Network; (iii) Mobile Device ;(iv) Wireless Network.

(i) Computer System:

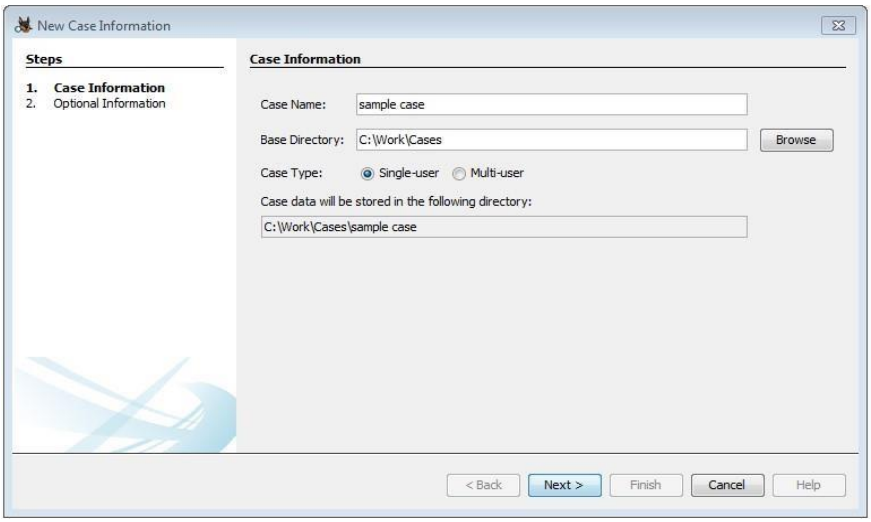
Forensic Tool: Autopsy

Download and Install Autopsy: Autopsy Open

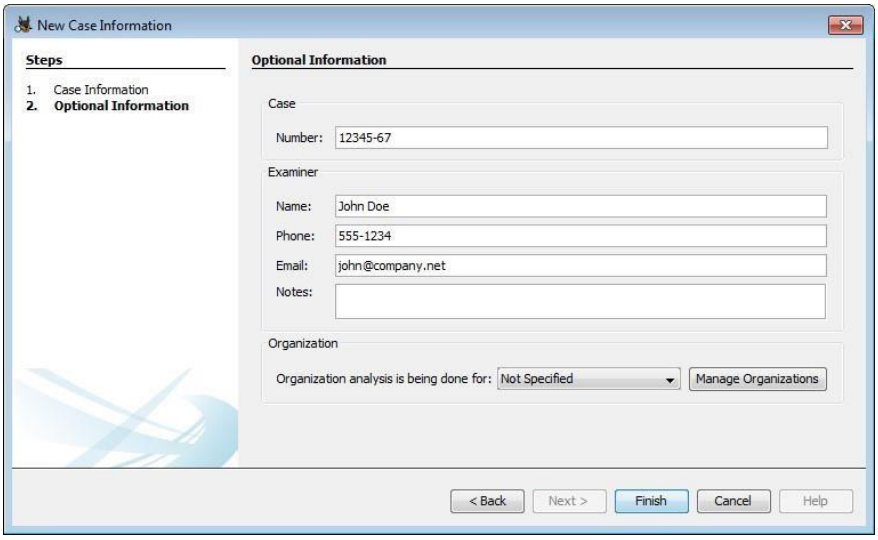
Autopsy and create a new case.



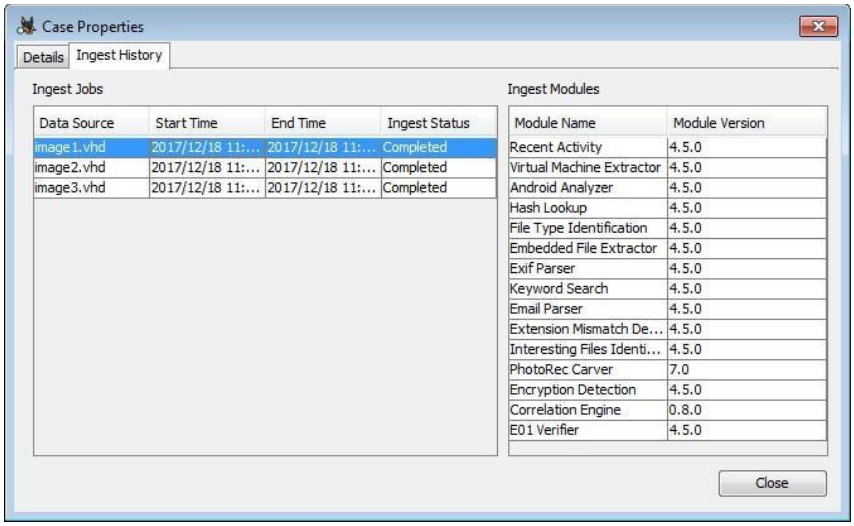
Enter case details, such as case name, case number, and investigator information.



Add a data source for the computer system (e.g., a disk image).



Start the analysis to examine file systems, recover deleted files, and explore system artifacts.



(ii) Computer Network:

Forensic Tool: Wireshark

Download and Install Wireshark: Wireshark

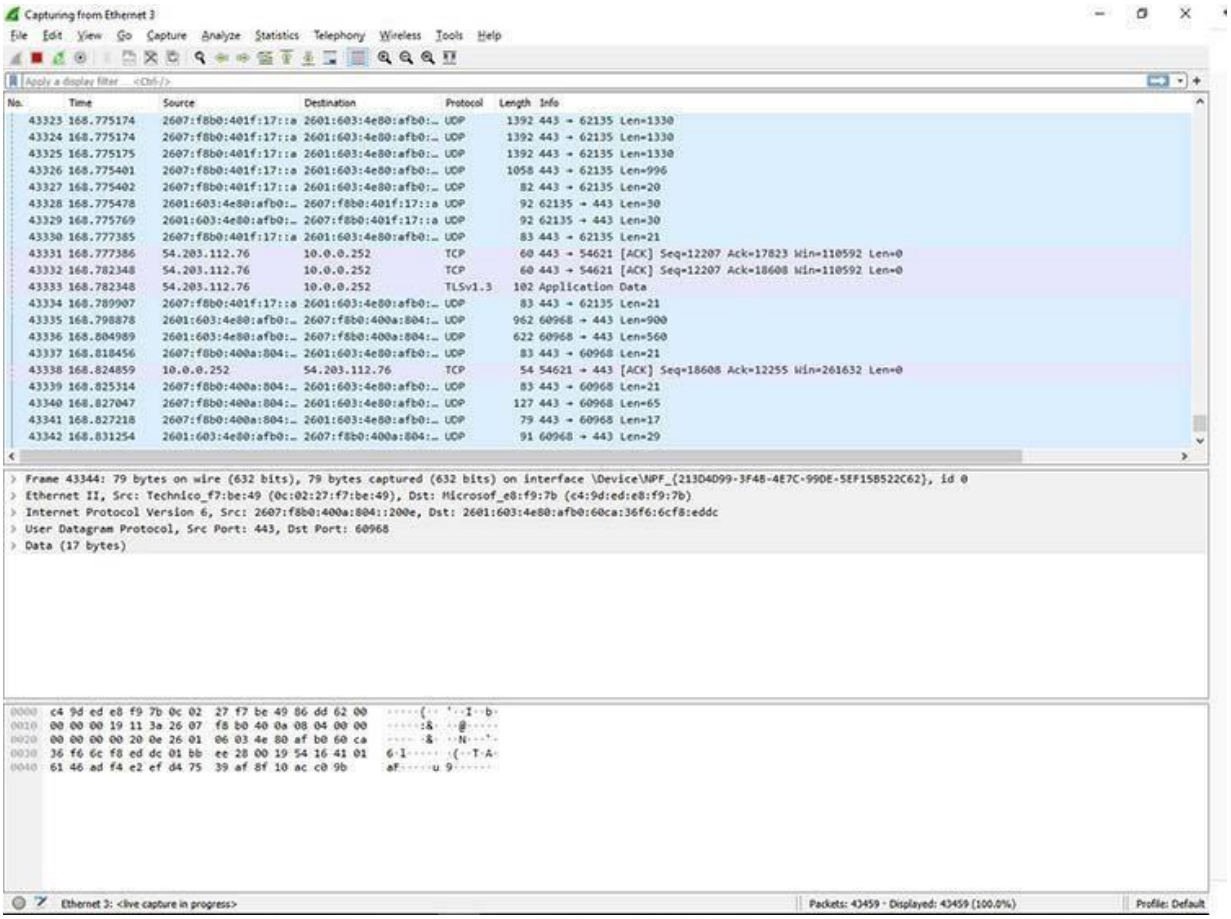
Capture network traffic using Wireshark on the suspect system.

Save the captured data to a file (PCAP format).

Open Wireshark and load the captured file.

Analyze the network traffic for any suspicious activities, such as unauthorized access or data

exfiltration.



(iii) Mobile Device:

Forensic Tool: Cellebrite UFED (Universal Forensic Extraction Device) Acquire

a Cellebrite UFED device or use the software version.

Connect the mobile device to the UFED device or software.

Follow the steps to create a new case in UFED.

Perform a forensic extraction of the mobile device's data.

Analyze the extracted data for evidence related to the investigation.

(iv) Wireless Network:

Forensic Tool: Aircrack-ng

Download and Install Aircrack-ng: Aircrack-ng

Capture wireless network traffic using Aircrack-ng tools like airodump-ng.
Save the captured data to a file.

Use tools like aircrack-ng to analyze the captured data and crack WEP/WPA keys if necessary.

Investigate any suspicious wireless network activities.