

Certificate

Name: Aditi Indoori

Class: H1 . (IT-1)

Roll No: 160118737001

Exam No:

Institution Chaitanya Bharathi Institute of Technology.

This is certified to be the bonafide work of the student in the
Information Security Laboratory during the academic
year 20 / 20 .

No. of practicals certified _____ out of _____ in the
subject of _____

.....
Teacher In-charge

.....
Examiner's Signature

.....
Principal

Date:

Institution Rubber Stamp

(N.B: The candidate is expected to retain his/her journal till he/she passes in the subject.)

Index

S. No.	Name of the Experiment	Page No.	Date of Experiment	Date of Submission	Remarks
(1)	Program to implement encryption & decryption using the following:	1			
(a)	Substitution Cipher - Polyalpha Cipher	1			
(b)	Intransposition Cipher - Railfence Cipher	4			
(c)	Product Cipher	7			
(2)	Program to implement Diffie Hellman key exchange algorithm.	9			
(3)	Program to implement AES Algorithm	10			
(4)	Program to implement SHA-1	12			
(5)	Program to implement MD5	14			
(6)	Program to implement digital signature Algorithm	16			
(7)	Blowfish Algorithm	19			
(8)	Wireshark	20			
(9)	Nmap	22			
(10)	KF Sensor	24			
(11)	Rootkits	26			
(12)	TCP Dump & Dumper	28			
(13)	Snort Tool	30			

(1) Program to implement encryption & decryption using the following

(a) Substitution cipher — Polyalphabetic cipher.

```
import java.util.*;
import java.util.Scanner;
class Polyalpha {
    // This function generates the key in a cyclic
    // manner until its length isn't equal to
    // the length of original text.
```

```
static String generateKey(String str, String key) {
    int n = str.length();
    for (int i=0; ; i++) {
        if (2n == i)
            i = 0;
        if (key.length() == str.length())
            break;
        key += (key.charAt(i));
    }
    return key;
}
```

// This function generates the encrypted text
// Generated with the help of the key

```
static String cipherText(String str, String key) {
    String ciphertext = "";
    for (int i=0; i<str.length(); i++) {
        // Converting in range 0-25
```

```

int x = (str.charAt(i) + key.charAt(i)) % 26;
//convert into alphabets (ASCII)
x += 'A';
cipher-text += (char)(x);
}

return cipher-text;
}

// This function decrypts the encrypted text
// and returns the original text
static String originalText (String cipher-text, String key) {
    String orig-text = "";
    for (int i=0; i < cipher-text.length() && i < key.length(); i++) {
        //converting in range 0-25
        int x = (cipher-text.charAt(i) - key.charAt(i) + 26) % 26;
        //convert into alphabets (ASCII)
        x += 'A';
        orig-text += (char)(x);
    }

    return orig-text;
}

```

```

public static void main (String[] args) {
    Scanner myObj = new Scanner (System.in); //create a scanner obj
    System.out.println ('Enter username');
    String str = myObj.nextLine();
    System.out.println ('Enter key');
    String keyword = myObj.nextLine();
    String key = generateKey (str, keyword);
    String cipher-text = cipherText (str, key);
}

```

```
System.out.println("Ciphertext : " + cipher_text + 'h');
System.out.println("Original/Decrypted Text : "
originalText(cipher_text, key));
```

{}

Description

- Polyalphabetic cipher is any cipher based on substitution, using multiple substitution alphabets.
- The relationship between a character in the plain text & the characters in the cipher text is one-to-many.
- Polyalphabetic cipher is described as substitution cipher in which plain text letters in different positions are enciphered using different cryptalphabets.

Output

Substitution cipher - Poly alphabetic cipher

Enter username

ADITI

Enter Key

SHH

Ciphertext : SKPLP

Original / Decrypted Text : ADITI

(b) Transposition Cipher - Rail fence cipher.

```

import java.util.Scanner;
class rail fence CipherHelper {
    int depth;
    String encode (String msg, int depth) throws Exception {
        int r = depth;
        int l = msg.length();
        int c = l / depth;
        int k = 0;
        char mat [j][] = new char[r][c];
        String enc = "";
        for (int i=0; i<c; i++) {
            for (int j=0; j<r; j++) {
                if (k == l) {
                    mat[j][i] = msg.charAt(k);
                } else {
                    mat[j][i] = 'X';
                }
            }
        }
        for (int i=0; i<r; i++) {
            for (int j=0; j<c; j++) {
                enc += mat[i][j];
            }
        }
        return enc;
    }
}

```

```

String decode(string encmsg, int depth) throws Exception {
    int r = depth;
    int L = encmsg.length();
    int c = l / depth;
    int k = 0;
    char mat[r][c] = new char[r][c];
    String dec = "";
    for (int i=0; i<r; i++) {
        for (int j=0; j<c; j++) {
            mat[i][j] = encmsg.charAt(k++);
        }
    }
    for (int i=0; i<c; i++) {
        for (int j=0; j<r; j++) {
            dec += mat[j][i];
        }
    }
    return dec;
}

```

```

class railfencecipher {
    public static void main(string[] args) throws java.lang.Exception {
        railfencecipherhelper rf = new railfencecipherhelper();
        string enc, dec;
        Scanner myObj = new Scanner (System.in); // create a scanner obj
        System.out.println ("Enter Username");
        string msg = myObj.nextLine();
        System.out.println ("Enter depth");

```

```
int depth = myObj.nextInt();
enc = rf.encode(msg, depth);
dec = rf.decode(enc, depth);
System.out.println("Simulating Railfence Cipher.....");
System.out.println("Input Message: " + msg);
System.out.println("Encrypted message: " + enc);
System.out.println("Decrypted Message: " + dec);
```

{

Description

- The rail fence cipher is a form of transposition cipher.
- In a transposition cipher, the order of the alphabets is rearranged to obtain the cipher-text.
- Railfence cipher is a transposition cipher consisting in writing a text in zig-zag & read it from left to right.

Encryption :

- Rail fence encryption uses an integer for the number of levels.
- The encoded message is written in zig-zag (like a railfence) along a path with N levels.

Eg: Encrypt DECODEZIGZAG with $N=3$ is writing

D	E	Z		
C	O	Z	G	A
O	I	G		

The cipher is read by rows.

Encrypted message is : DEZCDOZGAI

Decryption

- Railfence decryption requires to know the number of levels N .

Output

Transposition cipher - Railfence cipher

Enter Username

ADITI

Enter depth

2

Simulating the Railfence Cipher

Input message : ADITI

Encrypted Message : AIDT

Decrypted Message : ADIT

(c) Product Cipher

```

import java.util.*;
class ProductCipher {
    public static void main (String args[]) {
        System.out.println ("Enter the input to be encrypted:");
        String substitutionInput = new Scanner (System.in).nextLine();
        System.out.println ("Enter the key value");
        int k = new Scanner (System.in).nextInt();
        System.out.println ("Enter the depth of the transposition
                           cipher");
        int n = new Scanner (System.in).nextInt();
        // Substitution encryption
        StringBuffer substitutionOutput = new StringBuffer();
        for (int i=0; i<substitutionInput.length(); i++) {
            char c = substitutionInput.charAt(i);
            SubstitutionOutput.append ((char)(i+k));
        }
        System.out.println ("In Substituted text:");
        System.out.println (SubstitutionOutput);
        // Transposition encryption
        String transpositionInput = SubstitutionOutput.toString();
        int modulus;
        if ((modulus = transpositionInput.length() % n) != 0) {
            modulus = n - modulus;
        }
        // 'modulus' is now the number of blanks/padding (x)
        // to be appended.
        for (; modulus != 0; modulus--) {
            transpositionInput += "/";
        }
    }
}

```

```

String Buffer transpositionOutput = new String Buffer();
System.out.println("In Transposition Matrix:");
for (int i=0; i<n; i++) {
    for (int j=0; j<transpositionInput.length()/n; j++) {
        char c = transpositionInput.charAt(i + (j*n));
        System.out.print(c);
        transpositionOutput.append(c);
    }
}
System.out.println();
}

```

```

System.out.println("In final encrypted text:");
System.out.println(transpositionOutput);

```

//transposition decryption

```

n = transpositionOutput.length() / n;
String Buffer transpositionPlainText = new String Buffer();
for (int i=0; i<n; i++) {
    for (int j=0; j<transpositionOutput.length()/n; j++) {
        char c = transpositionOutput.charAt(i + (j*n));
        transpositionPlainText.append(c);
    }
}

```

//substitution decry

```

StringBuffer plaintext = new StringBuffer();
for (int i=0; i<transpositionPlainText.length(); i++) {
    char c = transpositionPlainText.charAt(i);
    plaintext.append((char)(c - k));
}

```

```

System.out.println("In PlainText:");

```

```

System.out.println(plaintext);

```

Description

- Product cipher, data encryption scheme in which the ciphertext produced by encrypting a plaintext document is subjected to further encryption.
- By combining two or more simple transposition ciphers, or substitution ciphers, a more secure encryption may result.

Output

Product cipher

Enter the input to be encrypted :

ADITI

Enter the key value

1

Enter the depth for transposition cipher -

1

Substituted text :

BEJUS

Transposition Matrix :

BEJUS

Final encrypted text -

BEJUS

Plain text :

ADITI

22.

Program to implement Diffie-Hellman key exchange algorithm.

```

import java.io.*;
import java.math.BigInteger;
public class DEFFIE_HELLMAN {
    public static void main (String [] args) throws IOException {
        BufferedReader br = new BufferedReader (new InputStreamReader
            (System.in));
        System.out.println ("Enter prime number:");
        BigInteger p = new BigInteger(br.readLine ());
        System.out.print ("Enter primitive root of " + p + ":");
        BigInteger g = new BigInteger(br.readLine ());
        System.out.print ("Enter value for x less than " + p + ":");
        BigInteger x = new BigInteger(br.readLine ());
        BigInteger R1 = g.modPow (x, p);
        System.out.println ("R1 = " + R1);
        System.out.print ("Enter value for y less than " + p + ":");
        BigInteger y = new BigInteger(br.readLine ());
        BigInteger R2 = g.modPow (y, p);
        System.out.println ("R2 = " + R2);
        BigInteger k1 = R2.modPow (x, p);
        System.out.println ("Key calculated at Alice's side: " + k1);
        BigInteger k2 = R1.modPow (y, p);
        System.out.println ("Key calculated at Bob's side: " + k2);
        System.out.println ("Diffie Hellman Secret Key Encryption has
            taken");
    }
}

```

Output

Enter prime number :

13

Enter primitive root of 13 : 5

Enter value for x less than 13 :

6

R₁ = 12

Enter value for y less than 13 : 3

R₂ = 8

key calculated at Alice's side : 12

key calculated at Bob's side : 12

Diffie Hellman Secret Key Encryption has taken

Description

- Diffie-Hellman (DH) Key exchange algorithm is a method for securely exchanging cryptographic keys over a public communications channels.
- Keys are not actually exchanged. They are jointly derived; named after their inventors Whitfield Diffie & Martin Hellman

(3) Program to Implement AES Algorithm.

```

import java.util.Base64;
import java.util.Scanner;
import javax.crypto.Cipher;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;

public class EncryptionDecryptionAES {
    static Cipher cipher;
    public static void main(String[] args) throws Exception {
        KeyGenerator keyGenerator = KeyGenerator.getInstance("AES");
        keyGenerator.init(128);
        SecretKey secretKey = keyGenerator.generateKey();
        Cipher cipher = Cipher.getInstance("AES");
        Scanner myobj = new Scanner(System.in);
        System.out.println("Enter plaintext");
        String plaintext = myobj.nextLine();
        System.out.print("Plain text before encryption: " + plaintext);
        String encryptedText = encrypt(plaintext, secretKey);
        System.out.print("Encrypted text after encryption: " + encryptedText);
        String decryptedText = decrypt(encryptedText, secretKey);
        System.out.print("Decrypted text after decryption: " + decryptedText);
    }

    public static String encrypt(String plaintext, SecretKey secretKey)
            throws Exception {
        byte[] plainTextByte = plaintext.getBytes();
        cipher.init(Cipher.ENCRYPT_MODE, secretKey);
        byte[] encryptedByte = cipher.doFinal(plainTextByte);
    }
}

```

```

Base64Encoder encoder = Base64.getEncoder();
String encryptedText = encoder.encodeToString(encryptedByte);
return encryptedText;
}

```

```

public static String decrypt(String encryptedText, SecretKey secretKey)
throws Exception {
}

```

```

Base64Decoder decoder = Base64.getDecoder();
byte[] encryptedTextByte = decoder.decode(encryptedText);
cipher.init(Cipher.DECRYPT_MODE, secretKey);
byte[] decryptedByte = cipher.doFinal(encryptedTextByte);
String decryptedText = new String(decryptedByte);
return decryptedText;
}
}

```

Description

The AES algorithm (also known as Rijndael algorithm) is a symmetrical block cipher algorithm that takes plain text in blocks of 128 bits & converts them to cipher text using keys of 128, 192, 256 bits.

Output

Enter plain text

aditi

Plain text before encryption : aditi

Encrypted Text after Encryption : w2dqssnbfxInotpIgI8HQ==

Decrypted Text after Decryption : aditi

(4) Program to calculate the message digest of a text using the SHA-1 algorithm.

```

import java.math.BigInteger;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

public class SHA1 {
    public static String encryptThisString(String input)
    {
        try {
            //getInstances() method is called with algorithm SHA-1
            MessageDigest md = MessageDigest.getInstance("SHA-1");

            //digest() method is called to calculate message digest
            //of the input String returned as array of byte
            byte[] messageDigest = md.digest(input.getBytes());

            //convert byte array into signum representation,
            BigInteger no = new BigInteger(1, messageDigest);

            //Convert message digest into hex value
            String hashText = no.toString(16);

            //Add preceding 0s to make it 32 bit
            while (hashText.length() < 32) {
                hashText = "0" + hashText;
            }
        }
    }
}

```

//return the hash text

return hashText;

}

//for specifying wrong message digest algorithms

catch (NoSuchAlgorithmException e) {

throw new RuntimeException(e);

}

}

public static void main(String args[]) throws NoSuchAlgorithmException

{

System.out.println("HashCode generated by SHA-1 for : ");

String s1 = "Aditi Indoors";

System.out.println ("ln" + s1 + ":" + encryptThisString(s1));

}

}

Description

SHA-1 or Secure Hash Algorithm 1 is a cryptographic hash function which takes an input & produces a 160-bit (20 byte) hash-value (message digest).

Output

HashCode generated by SHA-1 for :

AditiEndoori : 64c34f6446a25c9cd52ad703a537cf80049c5d6e

(5) Program to calculate the message digest of a text using the MD5 algorithm.

```

import java.math.BigInteger;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

public class MD5 {
    public static String getMD5(String input) {
        try {
            // Static getInstance method is called with
            // hashing MD5
            MessageDigest md = MessageDigest.getInstance("MD5");
            // digest() method is called to calculate message
            // digest of an input digest() return array of byte
            byte[] messageDigest = md.digest(input.getBytes());
            // Convert byte array into signum representation
            BigInteger no = new BigInteger(1, messageDigest);
            // Convert message digest into hex value
            String hashText = no.toString(16);
            while (hashText.length() < 32) {
                hashText = "0" + hashText;
            }
            return hashText;
        }
    }
}

```

//For specifying wrong message digest algorithms
catch (NoSuchAlgorithmException e) {
 throw new RuntimeException(e);
}

public static void main(String args[]) throws NoSuchAlgorithmException

String s = "AditiIndoor";

System.out.println("Your Hash Code Generated by MD5 is : "+
getMD5(s));
}

{

Description

The MD5 hashing algorithm is a one-way cryptographic function that accepts a message of any length as input & returns as output a fixed-length : digest value to be used for authenticating the original message.

Output

Your hash code generated by MD5 is : b9f620e9b3cb069af045a6----e6

(6) Program to implement Digital Signature Algorithm.

```
import java.security.KeyPair;
import java.security.KeyPairGenerator;
import java.security.PrivateKey;
import java.security.PublicKey;
import java.security.SecureRandom;
import java.security.Signature;
import java.util.Scanner;
import javax.xml.bind.DatatypeConverter;
```

```
public class DigitalSignature {
    // Signing Algorithm.
    private static final String SIGNING_ALGORITHM = "SHA256withRSA";
    private static final String RSA = "RSA";
    private static Scanner sc;
```

```
// function to implement Digital signature, using
// SHA256 & RSA algorithm, by passing private key
public static byte[] createDigitalSignature(
    byte[] input,
    PrivateKey key)
throws Exception {
    Signature signature = Signature.getInstance(
        SIGNING_ALGORITHM);
    signature.initSign(key);
    signature.update(input);
    return signature.sign();
}
```

//generating asymmetric key pair using SecureRandom
//class functions & RSA algorithm

public static KeyPair generateRSAKeyPair() throws
Exception {

SecureRandom secureRandom = new SecureRandom();

KeyPairGenerator keyPairGenerator = KeyPairGenerator.getInstance("RSA");

keyPairGenerator.initialize(2048, secureRandom);

return keyPairGenerator.generateKeyPair();

}

//function for verification of the digital signature using
//public key

public static boolean verifyDigitalSignature(

byte[] input,

byte[] signatureToVerify,

PublicKey key)

throws Exception.

}

Signature signature = Signature.getInstance(

SIGNATURE

SIGNING_ALGORITHM);

signature.initVerify(key);

signature.update(input);

return signature.verify(signatureToVerify);

}

public static void main(String args[]) throws Exception {

Scanner myobj = new Scanner(System.in);

System.out.println("Enter plaintext");

String input = myobj.nextLine();

keyPair keyPair = generate_RSA_KeyPair();

byte[] signature = create_Digital_Signature(
input.getBytes(),
keyPair.getPrivate());

System.out.println("Signature Value: " + datatypeConverter
.printHexBinary(signature));

System.out.println("Verification: " + verify_Digital_Signature(
input.getBytes(),
signature, keyPair.getPublic()));

}

}

Description

Digital signatures are the public-key primitives of message authentication. A digital signature is a technique that binds a person/entity to the digital data. It is a cryptographic value that is calculated from the data & a secret key only known by the signer.

Output

Enter plain text : Hi

Signature value 90006256847E06D5-----C65

Verification : True

(7) Blowfish Algorithm.

```

import javax.crypto.cipher;
import javax.crypto.spec.SecretKeySpec;
import java.util.Base64;

class Blowfish {
    private static final String Key = '123';
    public static String encrypt(String password) {
        try {
            byte[] keyData = (key).getBytes();
            SecretKeySpec secretKeySpec = new SecretKeySpec(
                keyData, "Blowfish");
            Cipher cipher = cipher.getInstance("Blowfish");
            cipher.init(Cipher.ENCRYPT_MODE, secretKeySpec);
            byte[] hasil = cipher.doFinal(password.getBytes());
            return new String(Base64.getEncoder().encode(hasil));
        }
        catch (Exception e) {
            e.printStackTrace();
            return null;
        }
    }

    public static void main(String[] args) {
        System.out.println("Your encrypted word is" + encrypt("Hello"));
    }
}

```

O/P:

Your encrypted word is cdwix@b51as

Description

Blowfish is an encryption technique designed as an alternative to DES encryption technique. It is significantly faster than DES & provides good encryption rate with no effective cryptanalysis technique found to date.

WIRESHARK:

Wireshark is an opensource packet analyser which is used for education, analysis, software development, communication protocol development & network troubleshooting.

features :

- It is a multiplatform software i.e. it can run on Linux, Windows, OS X, NetBSD etc.
- It is a standard three-pane packet browser.
- It performs deep inspection of the hundreds of protocols.
- It can also capture raw USB traffic
- It is also useful in VoIP analysis.

Installation :

- (1) Open the web browser
- (2) Search for 'Download wireshark'
- (3) Select the windows installer according to your system configuration, either 32-bit or 64-bit. Save the program in browser.
- (4) Now open the software & follow the install instruction by accepting the license.
- (5) Wireshark is ready for use.

why wireshark ?

- Wireshark is used to track the packets so that each one is filtered to meet our specific needs
- It is commonly called as sniffer, network protocol analyser & network analyser.
- It is also used by network security engineers to examine security problems

Uses of Wireshark

- It is used by network security engineers to examine security problems
- It allows the users to watch all the traffic being passed over the network
- It is used by network engineers to troubleshoot network issues.
- It also helps to troubleshoot latency issues & malicious activities on your computer.

Filters in Wireshark

(1) HTTP

Time	Source	Destination	Protocol	Length
204624.966343	192.250.195.174	192.168.1.104	HTTP	137

(2) TCP

Time	Source	Destination	Protocol	Length
29.356180.03948	192.168.1.104	23.58.36.195	TCP	54 [FIN ACK] seq. = 1699 win = 131328 len = 0.

(3) UDP

Time	Source	Destination	Protocol	Length	Info.
10.000	192.168.1.104	74.125.250.13	UDP	120	51891 → 19305 len = 78

NMAP

NMAP, short for Network Mapper is a network discovery & security auditing tool.

Uses:

- NMAP is widely used by network administrators to scan for:
- Open ports & services
 - Discover services along with their versions
 - Guess the operating system running on a target machine
 - Monitoring tools.

Common NMAP features

- PING SCANNING
- PORT SCANNING
- HOST SCANNING
- OS SCANNING
- SCAN TOP PORTS
- OUTPUT TO FILES
- DISABLE DNS RESOLUTION.

Scans:

- (1) Intense Scan: A comprehensive scan. Contains operating system (os) detection, version detection, script scanning, traceroute & has aggressive scan timing. This is considered an intrusive scan.

- (3) Ping Scan : This Scan simply detects if the targets are online , it doesn't scan any ports .
- (3) Quick Scan : This is quicker than a regular scan due to aggressive timing & only scanning select ports .
- (4) Regular Scan : This is the standard Nmap Scan without any modifiers . It will return ping & return open ports on the target .

Ex: Target IP address : 192.168.1.1

HOST DETAILS :

open ports : 2

filtered ports : 0

closed ports : 998

Addresses

IPv4 : 192.168.1.1

IPv6 : Not available

MAC : 80:3F:5D:87:3F:66

Operating System

Name : Linux 2.6.9 - 2.6.33

KF Sensor

- KF Sensor is a windows based honeypot intrusion detection system (IDS). It acts as a honeypot to attract & detect hackers & worms by simulating vulnerable system services & trojans. By acting as a decoy server it can divert attacks from critical systems and provide a higher level of information than can be achieved by using firewalls & NIDS alone. KF Sensor is a system installed in a network in order to divert & study an attacker's behaviour. This is a new technique that is effective in detecting attacks.
- The main feature of KF Sensor is that every connection it receives is a suspect hence it results in very few false alerts. At the heart of KF Sensor sits a powerful internet daemon service that is built to handle multiple ports & IP addresses.
- It is written to resist denial of service & buffer overflow attacks. Building on this flexibility KF Sensor can respond to connections in a variety of ways from simple port listening & basic services such as echo, to complex simulations of standard system services.

- For HTTP protocol KFSensor accurately simulates the way Microsoft's web server (IIS) responds to both valid & invalid requests.
- As well as being able to host a website it also handles complexities such as range requests & client side cache negotiations. This is extremely difficult for an attacker to fingerprint, or identify KFSensor as a honeypot.

Rootkits

- A rootkit is a stealthy type of malicious software (malware) designed to hide the existence of certain processes or programs from normal method of detections & enables continued privileged access to a computer.
- The term rootkit is a concatenation of "root" (the privileged account on Unix OS) & "kit" (software components that implement the tool).
- A rootkit is a collection of tools that enable administrator level access to a computer.
- A rootkit may consist of spyware & other programs that monitor traffic & keystrokes; create a "backdoor" into the system for hacker's use; alter log files; attack other mechanisms on the network & alter existing system tools to escape detection.
- Types of Rootkits
 - user-mode or application rootkit
 - kernel-mode
 - Bootkits
 - Firmware rootkits
 - Rootkit Hypervisors

→ Study about variety options

Steps:

- (1) Double click on rootkit folder
- (2) Double click on the CMER rootkit application.
- (3) Now the rootkit screen will be displayed.
- (4) Select any one of the drive which is shown at right side of the screen.
- (5) After selecting the drive click on Scan button
- (6) click on the option processes the screen will be displayed.
- (7) click on option services
- (8) Now click on different options to perform different actions.

TCP Dump & Dumpcap

- TCP dump is a data-network packet analyzer computer program that runs under a command line interface. It allows the user to display TCP/IP and other packets being transmitted or received over a network to which the computer is attached.
- TCP dump uses libpcap library to capture the network packets & is available on almost all Linux / unix flavors.
- It is used to capture, filter & analyze network traffic such as TCP/IP packets going through your system. It is mainly used as a security tool. It saves the captured information in a pcap file, these pcap files can be opened through wireshark or through the command tool itself.
- * To capture the packets of current network interface.
- Sudo tcpdump.
- * To display all available interfaces.
- Sudo tcpdump -D.
- * To capture packets from a specified network interface.
- Sudo tcpdump -i wlo1

- * To capture specific number of packets
 - sudo tcpdump -c 4 -i wlo1

- * To print captured packets in ASCII format .
 - sudo tcpdump -A -i wlo1

- * To capture only TCP packets
 - sudo tcpdump -i wlo1 tcp

- * Get all the packets based on the IP addresses whether source or destination or both, using the following command .
 - sudo tcpdump host 192.168.1.100 .

- * To get packets based on source or destination of an IP address, use .
 - sudo tcpdump src 192.168.1.100
 - sudo tcpdump dst 192.168.1.100

- Dumpcap is a network traffic dump tool. It captures packet data from a live network & writes the packets to a file.

- Dumpcap's native capture file format is pcapng, which is also the format used by Wireshark.

- By default dumpcap uses the pcap library to capture traffic from the first available network interface & writes the received raw packet data, along with the packets time stamps into a pcapng file .

Snort Tool

Intrusion detection using SNORT TOOL :

- A general thought is that if a firewall is protecting one's network, the network can be considered secure.
- Intrusion detection system is used to evaluate aggressive (or) unexpected packets & generate an alert before these programs harm the network.
- An advantage of host based intrusion detection systems is that it can also detect anomalies (or) malicious traffic, generated from the host itself.
- HIDS work by monitoring & analysing network traffic & compare it with an established ruleset.
- SNORT is a flexible, lightweight & popular system that can be deployed according to the needs of the network.

INFORMATION SECURITY

LAB RECORD – PROGRAM OUTPUTS

Aditi Indoori
160118737001
IT1 (H1)

1. Program to implement encryption and decryption using the following:

(a) Substitution Cipher – Polyalphabetic Cipher

```
D:\6th semester\Information Security Lab>javac Polyalpha.java
D:\6th semester\Information Security Lab>java Polyalpha
Enter username
ADITI
Enter key
SHH
Ciphertext : SKPLP
Original/Decrypted Text : ADITI
```

(b) Transposition Cipher – Railfence Cipher

```
D:\6th semester\Information Security Lab>javac railFenceCipher.java
D:\6th semester\Information Security Lab>java railFenceCipher
Enter username
ADITI
Enter depth
2
Simulating Railfence Cipher
-----
Input Message : ADITI
Encrypted Message : AIDT
Decrypted Message : ADIT
```

(c) Product Cipher

```
D:\6th semester\Information Security Lab>javac ProductCipher1.java

D:\6th semester\Information Security Lab>java ProductCipher1
Enter the input to be encrypted:
ADITI
Enter the key value
1
Enter the depth for transposition cipher
1

Substituted text:
BEJUJ

Transposition Matrix:
BEJUJ

Final encrypted text:
BEJUJ

Plaintext:
ADITI
```

2. Program to implement Diffie Hellman Key Exchange Algorithm

```
D:\6th semester\Information Security Lab>javac DEFFIE_HELLMAN.java

D:\6th semester\Information Security Lab>java DEFFIE_HELLMAN
Enter prime number:
13
Enter primitive root of 13:5
Enter value for x less than 13:
6
R1=12
Enter value for y less than 13:3
R2=8
Key calculated at Alice's side:12
Key calculated at Bob's side:12
deffie hellman secret key Encryption has Taken
```

3. Program to implement AES Algorithm

```
D:\6th semester\Information Security Lab>javac EncryptionDecryptionAES.java

D:\6th semester\Information Security Lab>java EncryptionDecryptionAES
Enter plaintext
aditi
Plain Text Before Encryption: aditi
Encrypted Text After Encryption: bNzgIh3d3t5aDRTLgICtWA==
Decrypted Text After Decryption: aditi
```

4. Program to implement SHA-1

```
D:\6th semester\Information Security Lab>javac SHA1.java

D:\6th semester\Information Security Lab>java SHA1
HashCode Generated by SHA-1 for:
AditiIndoori : 64c34f6446a25c9cd52ad703a537cf00a49c5d6e
```

5. Program to implement MD5

```
D:\6th semester\Information Security Lab>javac MD5.java

D:\6th semester\Information Security Lab>java MD5
Your HashCode Generated by MD5 is: b9f620e9b3cb069af045a6ab1ffa5aeb
```

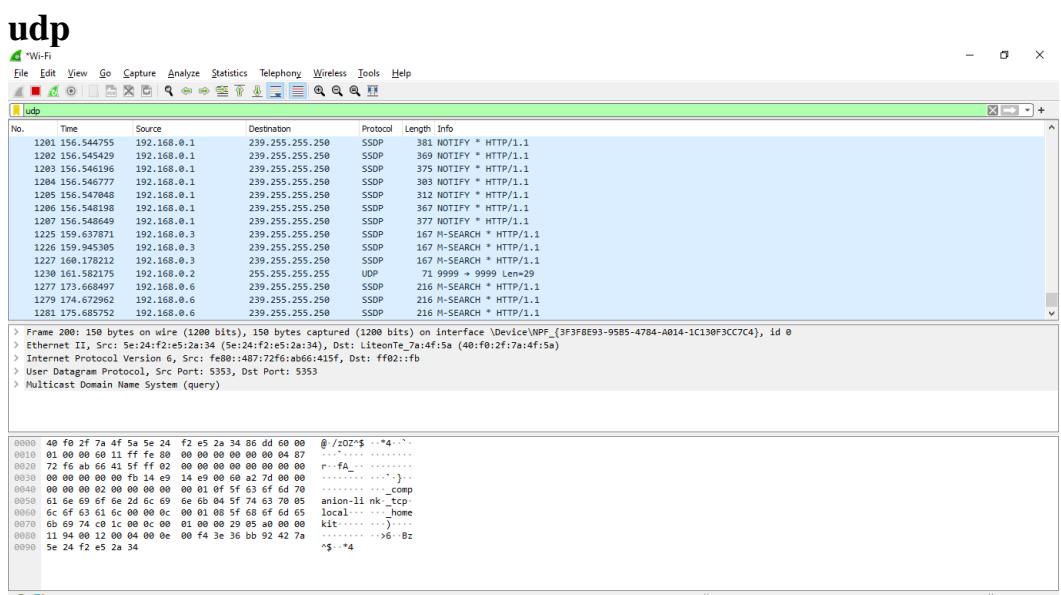
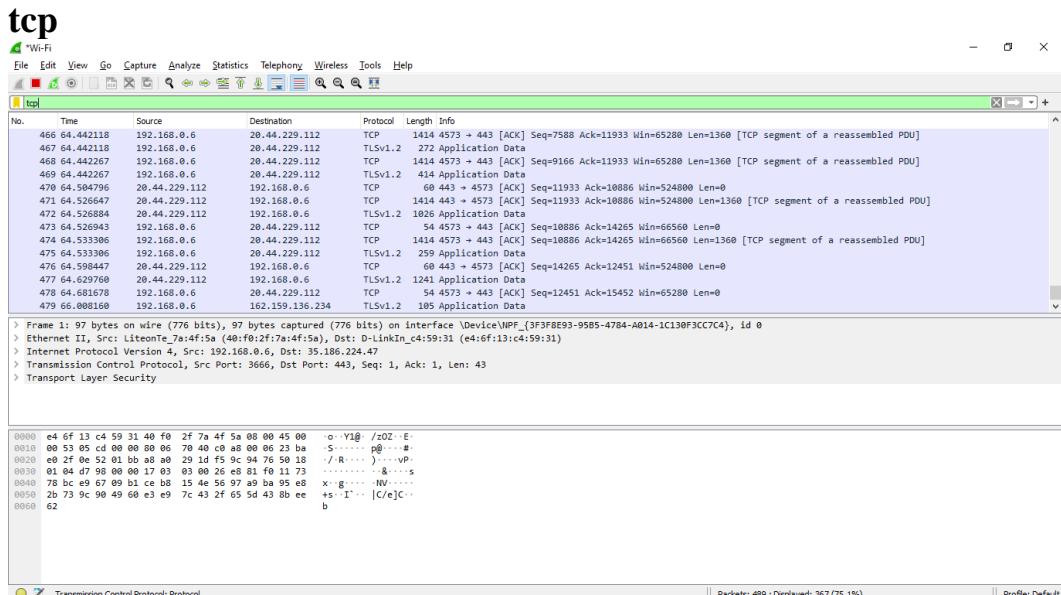
6. Program to implement Digital Signature Algorithm

```
Enter plaintext
hi
Signature Value:
885d717c9177137b88350440967145d5455de44089a4aff1e87915c5a2fdd1bce9dd5708b76bc1f5edd978242245fabe0c44c1ce2e0cb679e7b146
40bb6288194bb5a46c01dbe97aap3d67b7e4a4f64012be5002d08a03be14aaef86a77a75db69002a31df251403a30fbb11f400fc030b9ef2bb7527c
b2bdabbaacdadead1f673277ca2376df9160a6ee933c05c1829b58c8be1c49ceb4776a328coaa1ae004eb2a358ca005bb583331feddf2338f6b6bc87
cb8b75ff07d814bb6290e1727e045d7bd865048d48b1e39489a54377abc783999cd9bf1914e8ccb46d4eb4ff86f065fb34bbff7a0963b62d7f9064d
6b92db05c4f67886db1d2ebe020fdd5a8d2f6
Verification: true
```

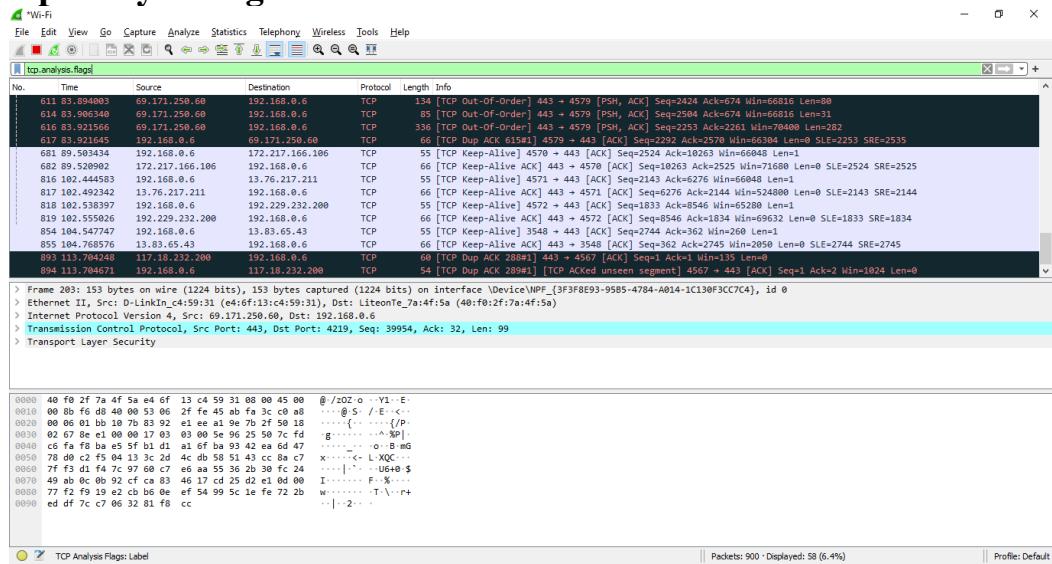
7. Blowfish Algorithm

```
D:\6th semester\Information Security Lab>javac Blowfish.java  
D:\6th semester\Information Security Lab>java Blowfish  
your encrypted word is aE9JrBKj0xk=
```

8. Wireshark

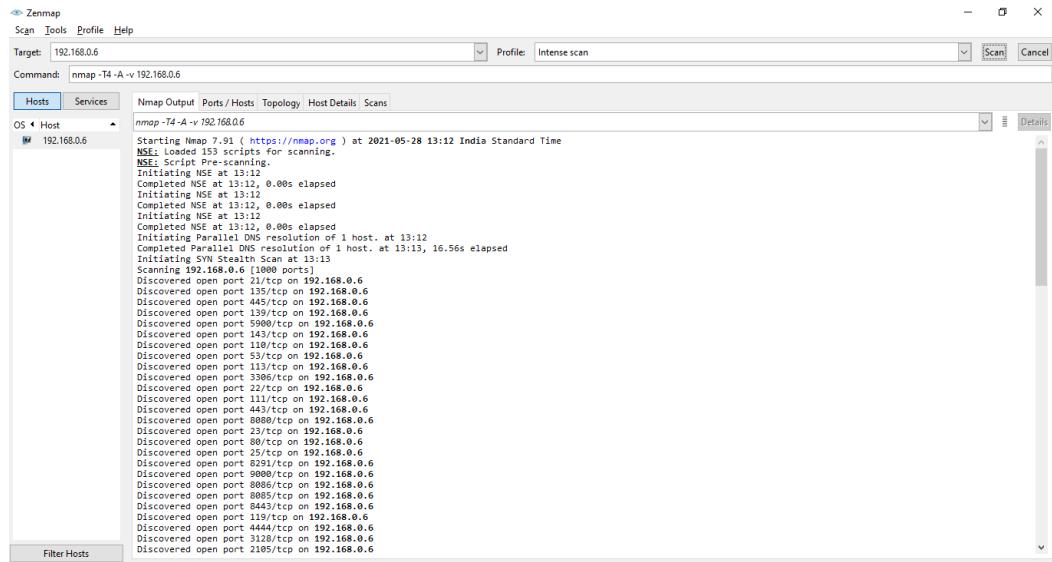


tcp.analysis.flags

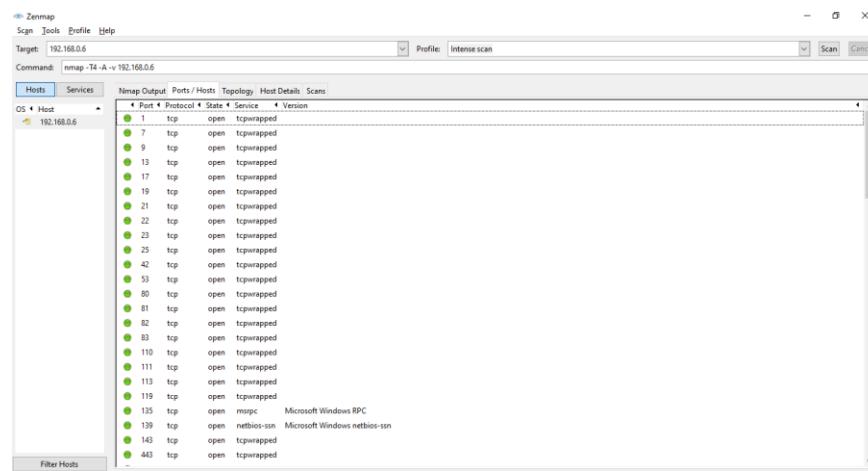


9. Nmap

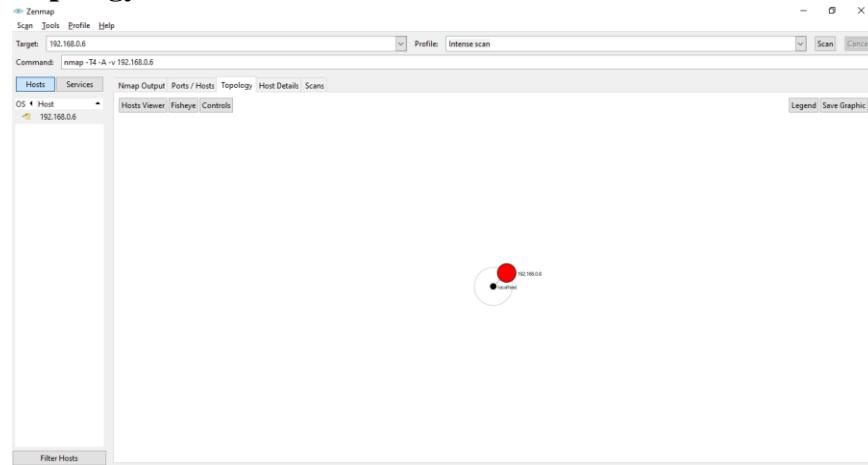
Intense scan



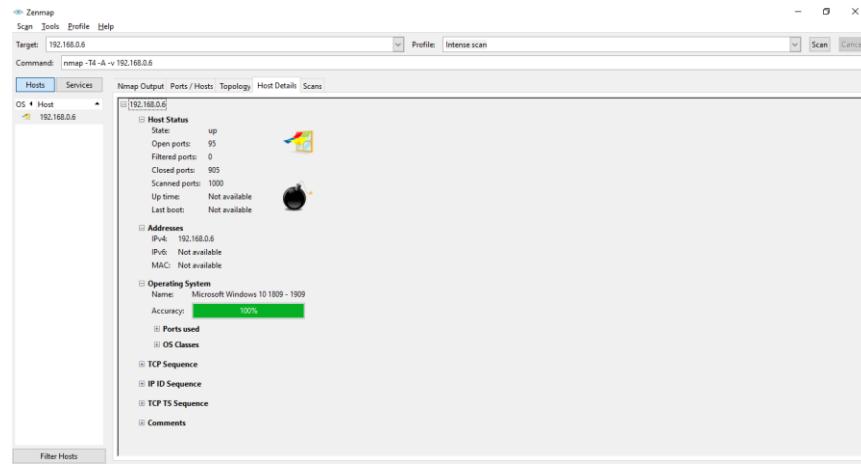
- Ports/Hosts



- Topology



- Host Details



Ping scan

Zenmap interface showing a ping scan of host 192.168.0.6. The command used is nmap -sn 192.168.0.6. The output shows the host is up.

```
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-28 13:13 India Standard Time
Nmap scan report for 192.168.0.6
Host is up.
Nmap done; 1 IP address (1 host up) scanned in 17.19 seconds
```

Quick scan

Zenmap interface showing a quick scan of host 192.168.0.6. The command used is nmap -T4 -F 192.168.0.6. The output shows various open ports including 80/tcp, 443/tcp, 8000/tcp, and 5443/tcp.

```
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-28 13:11 India Standard Time
Nmap scan report for 192.168.0.6
Host is up (0.0024s latency).
Not shown: 56 closed ports
PORT      STATE SERVICE
7/tcp      open  discard
9/tcp      open  discard
13/tcp     open  daytime
21/tcp     open  ftp
22/tcp     open  ssh
23/tcp     open  telnet
25/tcp     open  smtp
53/tcp     open  domain
80/tcp     open  http
80/https   open  http-2
110/tcp    open  pop3
111/tcp    open  rpcbind
113/tcp    open  ident
119/tcp    open  nntp
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
143/tcp    open  imap
443/tcp    open  https
445/tcp    open  microsoft-ds
548/tcp    open  afp
1028/tcp   open  unknown
125/tcp    open  sql-s
2000/tcp   open  cisco-ccp
3000/tcp   open  ppp
3128/tcp   open  squid-http
3306/tcp   open  mysql
4899/tcp   open  unknown
5000/tcp   open  upnp
5060/tcp   open  sip
5357/tcp   open  webmail
5432/tcp   open  postgresql
5631/tcp   open  picanyvheredata
5800/tcp   open  vnc-https
5900/tcp   open  vnc
```

10. KFSensor

Ports

KFSensor Professional - Evaluation Trial

File View Scenario Signatures Settings Help

Ports

ID	Start	Duration	Pro.	Sens.	Name	Visitor	Description	Received	Sig. Message
886	28-05-2021 13:32:05.638	21.043	TCP	6942	gmail.com		SendMail: Protocol Failure id:15 t...	421 Cannot connect to SMTP serv...	
885	28-05-2021 13:32:04.000	1.639	WIN	3389	Logon		auditType:Failure Audit Account:...	Audit type: Failure Audit[00 0A]...	
884	28-05-2021 13:32:04.000	1.636	WIN	3389	Logon		auditType:Failure Audit Account:...	Audit type: Failure Audit[00 0A]...	
883	28-05-2021 13:13:12.395	21.053	TCP	4795	gmail.com		SendMail: Protocol Failure id:14 t...	421 Cannot connect to SMTP serv...	
882	28-05-2021 13:13:06.276	21.067	TCP	4750	gmail.com		SendMail: Protocol Failure id:12 t...		
881	28-05-2021 13:13:06.302	21.041	TCP	4754	gmail.com		SendMail: Protocol Failure id:13 t...		
879	28-05-2021 13:13:12.388	6.038	TCP	443	IIS HTTPS	DESKTOP-L4P127N...	[IAC WILL TerminalType](IAC SB T...	secure connection;established, u...	GET / HTTP/1.0[00 0A 0D 0A]
878	28-05-2021 13:13:12.393	6.020	TCP	3000	Terminal Server	DESKTOP-L4P127N...			[00 0A 0D 0A 0A 0D 0A]
877	28-05-2021 13:13:12.366	5.031	TCP	2967	Symantec Anti...	DESKTOP-L4P127N...			
876	28-05-2021 13:13:12.347	5.034	TCP	2222	AMD exploit C...	DESKTOP-L4P127N...	Idle time out		
875	28-05-2021 13:13:12.342	5.039	TCP	2107	MS MQS	DESKTOP-L4P127N...	Idle time out		
874	28-05-2021 13:13:06.301	11.025	TCP	1433	SQL Server	DESKTOP-L4P127N...		TDS Packet: Num:1 Type:id:80 Typ...	
873	28-05-2021 13:13:06.290	11.034	TCP	510	POP3	DESKTOP-L4P127N...		[00 0A 0D 0A 0 0 0A]	
872	28-05-2021 13:13:10.574	5.059	TCP	2105	MS MQS	DESKTOP-L4P127N...			
871	28-05-2021 13:13:12.361	2.023	TCP	2869	MS UPNP Host	DESKTOP-L4P127N...			
870	28-05-2021 13:13:10.341	4.028	TCP	143	IMAP	DESKTOP-L4P127N...			
869	28-05-2021 13:13:10.345	4.025	TCP	1099	Java RMI Server	DESKTOP-L4P127N...		GET / HTTP/1.0[00 0A 0D 0A]	JRM[00 02]K
868	28-05-2021 13:13:10.341	4.024	TCP	22	SSH	DESKTOP-L4P127N...		[00 0A 0D 0A]	
867	28-05-2021 13:13:10.346	4.024	TCP	2000	iKettle	DESKTOP-L4P127N...		[16 03 00 00][01 00 00][03 00]G...	
866	28-05-2021 13:13:10.343	4.027	TCP	548	Apple Filing Pr...	DESKTOP-L4P127N...		[16 03 00 00][01 00 00][03 00]G...	
865	28-05-2021 13:13:10.344	4.026	TCP	636	LDAP SSL	DESKTOP-L4P127N...		[16 03 00 00][01 00 00][03 00]G...	
864	28-05-2021 13:13:10.340	4.029	TCP	119	NNTP	DESKTOP-L4P127N...		[00 0A 0D 0A]	
863	28-05-2021 13:13:10.341	4.028	TCP	42	WINS	DESKTOP-L4P127N...		[00 00 00 A4 FF]SMB[00 00 00 ...	
862	28-05-2021 13:13:10.335	4.034	TCP	53	DNS	DESKTOP-L4P127N...		[00 1E 00 05 01 00 00 01 00 00 0...	
861	28-05-2021 13:13:10.336	4.033	TCP	111	sunmpc	DESKTOP-L4P127N...		[80 00 00][FF 1D 13 00 00 00 00 ...	
860	28-05-2021 13:13:10.338	4.031	TCP	113	ident	DESKTOP-L4P127N...		[00 0A 0D 0A]	
859	28-05-2021 13:13:10.311	4.028	TCP	9	Discard	DESKTOP-L4P127N...		[00 0A 0D 0A]	
858	28-05-2021 13:13:10.314	4.024	TCP	1	port one	DESKTOP-L4P127N...		GET / HTTP/1.0[00 0A 0D 0A]	
857	28-05-2021 13:13:06.297	6.284	TCP	1028	MS CIS	DESKTOP-L4P127N...		[03 00 00 08 06 00 00 00 00 00]	
856	28-05-2021 13:13:06.295	6.286	TCP	593	CIS	DESKTOP-L4P127N...		[00 0A 0D 0A]	
855	28-05-2021 13:13:12.159	0.266	TCP	2103	MS MQS	DESKTOP-L4P127N...		OPTIONS / RTSP/1.0[00 0A 0D 0A]	
854	28-05-2021 13:13:12.162	0.263	TCP	1801	MS MQS	DESKTOP-L4P127N...		OPTIONS / RTSP/1.0[00 0A 0D 0A]	
853	28-05-2021 13:13:12.391	0.007	TCP	1080	SOCKS	DESKTOP-L4P127N...		[05 04 00 01 02 80 05 01 00 03 0A]g...	
852	28-05-2021 13:13:12.393	0.000	TCP	3000	Multi-port Scan	DESKTOP-L4P127N...		Port Scan,[00 0A 0D 0A]The visit...	
851	28-05-2021 13:13:12.373	0.012	TCP	443	IIS HTTPS	DESKTOP-L4P127N...			
850	28-05-2021 13:13:12.376	0.009	TCP	1080	SOCKS	DESKTOP-L4P127N...		GET / HTTP/1.0[00 0A 0D 0A]	
849	28-05-2021 13:13:06.283	6.056	TCP	23	Telnet	DESKTOP-L4P127N...		[00 0A 0D 0A]	

User Rights: Admin [78] Server: Attack Visitors: 14 Events: 886/886

Visitors

KFSensor Professional - Evaluation Trial

File View Scenario Signatures Settings Help

Visitors

ID	Start	Duration	Pro.	Sens.	Name	Visitor	Description	Received	Sig. Message
880	28-05-2021 13:13:12.353	11.030	TCP	2323	Telnet IoT	DESKTOP-L4P127N...	(IAC WILL TerminalType)(IAC SB T...		
878	28-05-2021 13:13:12.388	6.038	TCP	443	IIS HTTPS	DESKTOP-L4P127N...	secure connection;established, u...	GET / HTTP/1.0[00 0A 0D 0A]	
877	28-05-2021 13:13:12.366	6.020	TCP	3000	Terminal Server	DESKTOP-L4P127N...			
876	28-05-2021 13:13:12.347	5.034	TCP	2222	AMD exploit C...	DESKTOP-L4P127N...	Idle time out		
875	28-05-2021 13:13:12.342	5.039	TCP	2107	MS MQS	DESKTOP-L4P127N...	Idle time out		
874	28-05-2021 13:13:06.301	11.025	TCP	1433	SQL Server	DESKTOP-L4P127N...			TDS Packet: Num:1 Type:id:80 Typ...
873	28-05-2021 13:13:06.290	11.034	TCP	110	POP3	DESKTOP-L4P127N...		[00 0A 0D 0A 0 0 0A]	
872	28-05-2021 13:13:10.574	5.059	TCP	2105	MS MQS	DESKTOP-L4P127N...			
871	28-05-2021 13:13:12.361	2.023	TCP	2869	MS UPNP Host	DESKTOP-L4P127N...			
870	28-05-2021 13:13:10.341	4.028	TCP	143	IMAP	DESKTOP-L4P127N...		GET / HTTP/1.0[00 0A 0D 0A]	JRM[00 02]K
869	28-05-2021 13:13:10.345	4.025	TCP	1099	Java RMI Server	DESKTOP-L4P127N...			[00 0A 0D 0A 0 0 0A]
868	28-05-2021 13:13:10.341	4.029	TCP	22	SSH	DESKTOP-L4P127N...		[16 03 00 00][01 00 00][03 00]G...	
867	28-05-2021 13:13:10.346	4.024	TCP	2000	iKettle	DESKTOP-L4P127N...		[16 03 00 00][01 00 00][03 00]G...	
866	28-05-2021 13:13:10.343	4.027	TCP	548	Apple Filing Pr...	DESKTOP-L4P127N...		[16 03 00 00][01 00 00][03 00]G...	
865	28-05-2021 13:13:10.340	4.026	TCP	636	LDAP SSL	DESKTOP-L4P127N...		[16 03 00 00][01 00 00][03 00]G...	
864	28-05-2021 13:13:10.340	4.029	TCP	119	NNTP	DESKTOP-L4P127N...		[00 0A 0D 0A]	
863	28-05-2021 13:13:10.341	4.028	TCP	42	WINS	DESKTOP-L4P127N...		[00 00 00 A4 FF]SMB[00 00 00 ...	
862	28-05-2021 13:13:10.335	4.034	TCP	53	DNS	DESKTOP-L4P127N...		[00 1E 00 05 01 00 00 01 00 00 0...	
861	28-05-2021 13:13:10.336	4.033	TCP	111	sunmpc	DESKTOP-L4P127N...		[80 00 00][FF 1D 13 00 00 00 00 ...	
860	28-05-2021 13:13:10.338	4.031	TCP	113	ident	DESKTOP-L4P127N...		[00 0A 0D 0A]	
859	28-05-2021 13:13:10.311	4.028	TCP	9	Discard	DESKTOP-L4P127N...		[00 0A 0D 0A]	
858	28-05-2021 13:13:10.314	4.024	TCP	1	port one	DESKTOP-L4P127N...		GET / HTTP/1.0[00 0A 0D 0A]	
857	28-05-2021 13:13:06.297	6.284	TCP	1028	MS CIS	DESKTOP-L4P127N...		[03 00 00 08 06 00 00 00 00 00]	
856	28-05-2021 13:13:06.299	6.286	TCP	593	CIS	DESKTOP-L4P127N...		[00 0A 0D 0A]	
855	28-05-2021 13:13:12.159	0.266	TCP	2103	MS MQS	DESKTOP-L4P127N...		OPTIONS / RTSP/1.0[00 0A 0D 0A]	
854	28-05-2021 13:13:12.162	0.263	TCP	1801	MS MQS	DESKTOP-L4P127N...		OPTIONS / RTSP/1.0[00 0A 0D 0A]	
853	28-05-2021 13:13:12.391	0.007	TCP	1080	SOCKS	DESKTOP-L4P127N...		[05 04 00 01 02 80 05 01 00 03 0A]g...	
852	28-05-2021 13:13:12.393	0.000	TCP	3000	Multi-port Scan	DESKTOP-L4P127N...		Port Scan,[00 0A 0D 0A]The visit...	
851	28-05-2021 13:13:12.373	0.012	TCP	443	IIS HTTPS	DESKTOP-L4P127N...			
850	28-05-2021 13:13:12.376	0.009	TCP	1080	SOCKS	DESKTOP-L4P127N...		GET / HTTP/1.0[00 0A 0D 0A]	
849	28-05-2021 13:13:06.283	6.056	TCP	23	Telnet	DESKTOP-L4P127N...		[00 0A 0D 0A]	

User Rights: Admin [78] Server: Running Visitors: 14 Events: 98/886

Event Details

KFSensor Professional - Evaluation Trial

File View Scenario Signatures Settings Help

Event - 878

ID	Start	Dur.
886	28-05-2021 13:32:05.638	
885	28-05-2021 13:32:04.000	
884	28-05-2021 13:32:04.000	
883	28-05-2021 13:13:12.393	
882	28-05-2021 13:13:06.276	
881	28-05-2021 13:13:06.302	
880	28-05-2021 13:13:12.353	
879	28-05-2021 13:13:12.388	
878	28-05-2021 13:13:12.393	
877	28-05-2021 13:13:12.366	
876	28-05-2021 13:13:12.347	
875	28-05-2021 13:13:12.342	
874	28-05-2021 13:13:06.301	
873	28-05-2021 13:13:06.290	
872	28-05-2021 13:13:10.574	
871	28-05-2021 13:13:12.361	
870	28-05-2021 13:13:10.341	
869	28-05-2021 13:13:10.345	
868	28-05-2021 13:13:10.341	
867	28-05-2021 13:13:10.346	
866	28-05-2021 13:13:10.343	
865	28-05-2021 13:13:10.344	
864	28-05-2021 13:13:10.340	
863	28-05-2021 13:13:10.341	
862	28-05-2021 13:13:10.335	
861	28-05-2021 13:13:10.336	
860	28-05-2021 13:13:10.338	
859	28-05-2021 13:13:10.311	
858	28-05-2021 13:13:10.314	
857	28-05-2021 13:13:06.297	
856	28-05-2021 13:13:06.295	
855	28-05-2021 13:13:12.159	

Summary Details Signature Data

Event
Sensor ID: kfsensor Event ID: 878
Start Time: 28-05-2021 13:13:12.393 Severity: High
Description:
Visitor IP: 192.168.0.6 Port: 4793
Domain: DESKTOP-L4P127N.domain.name
Sensor Name: Terminal Server
Protocol: TCP Port: 3000
Signature Message:
Request Data - 4 Bytes

Received Sig. Message

protocol Failure id:15... 421 Cannot connect to SMTP serv...
failure Audit Account... Audit Type: Failure Audit [ID 0A]...
failure Audit Account... Audit Type: Failure Audit [ID 0A]...
protocol Failure id:14... 421 Cannot connect to SMTP serv...
protocol Failure id:13... [IAC WILL TerminalType][IAC SB T...
ection:established. | u... GET /HTTP/1.0/0D 0A 0D 0A
[0D 0A 0D 0A 0D 0A]

TDS Packet: Num:1 Type:id:80 Typ...
[0D 0A 0D 0A 0D 0A]

GET /HTTP/1.0/0D 0A 0D 0A]
JRM/[0D 02]K
[0D 0A 0D]
[16 03 00 00]S[01 00 00]O[03 00]T[G...
[16 03 00 00]S[01 00 00]O[03 00]T[G...
[16 03 00 00]S[01 00 00]O[03 00]T[G...
[0D 0A 0D]
[0D 00 A4 F7FBMB[00 00 00 ...
[00 1E 00 01 00 01 00 01 00 00 00 ...
[00 80]R[FTE 1D 13 00 00 00 00 ...
[0D 0A 0D]
[0D 0A 0D]
GET /HTTP/1.0/0D 0A 0D 0A]
[03 00 08 0B 60 00 00 00 00]
[0D 0A 0D]
OPTIONS /RTSP/1.0/D[0D 0A 0D]

User Rights: Admin [78] Server Running Visitors: 14 Events: 886/886

Edit Scenarios

KFSensor Professional - Evaluation Trial

File View Scenario Signatures Settings Help

KFSensor - localhost - Main Scenario

ID	Start	Duration	Pro...	Sens...	Name	Visitor	Description
886	28-05-2021 13:32:05.638	21,043	TCP	6942	gmail.com		SendMail: Protocol

1 port Edit Scenario

9 Discs

13 Day

17 Quo

19 char

21 FTP

22 SSH

23 Telnet

25 SMTP

42 WIN

53 DNS

68 DHC

80 IIS

81 IIS S

82 IIS S

83 IIS S

110 PO

111 sru

113 ide

119 NN

135 MS

139 NB

Name: Main Scenario **Domain Name:** networkforu.com

Listen

Name	Active	Hide	Proto...	Port	Sensor Bind	Severity	Action	Sim Server	Class
Closed TCP Ports	True	Hide	TCP	0	Native	High	ReadAndClose		Trojans and worms
port one	True	Hide	TCP	1	Medium	Close			Trojans and worms
Death, Trojan	True	Hide	TCP	2	Medium	Close			Trojans and worms
Echo	True	Hide	TCP	7	Medium	SimBanner	Echo		Linux (services no...)
Discard	True	Hide	TCP	13	Medium	ReadAndClose			Linux (services no...)
Daytime	True	Hide	TCP	17	Medium	SimBanner	Daytime		Linux (services no...)
Quote of the day	True	Hide	TCP	19	Medium	SimBanner	chargen		Linux (services no...)
chargen	True	Hide	TCP	21	High	SimServer	chargen		Windows Internet...
FTP	True	Hide	TCP	22	Medium	ReadAndClose			Linux (services no...)
SSH	True	Hide	TCP	23	High	SimServer	Telnet		Linux (services no...)
SMTP	True	Hide	TCP	25	High	SimServer	SMTP		Windows Internet...
WINS	True	Hide	TCP	42	High	ReadAndClose			Windows Server
DNS	True	Hide	TCP	53	Medium	ReadAndClose			Windows Server
Mail Transfer Pr...	True	Hide	TCP	57	Medium	Close			Linux (services no...)
DHCP	True	Hide	TCP	68	Medium	Close			Windows Server
ITS	True	Hide	TCP	80	Medium	SimServer	ITS		Windows Internet...

Edit Listen

Listen On: port one

Icon: Hadier

Class: Trojans and worms

Protocol: TCP

Port: 1

Bind Address:

Active: Hide if no events:

Action

Action Type: Close Read And Close Sim Banner Sim Std Server Native

Severity: Medium

Time Out: 0 Milliseconds

Sim Name:

Visitor DOS Attack Limits

Max connections per IP:

Action on max connections per IP:

OK **Cancel** **Help**

11. Rootkits

Services

Processes	Modules	Services	Files	Registry	Rootkit/Malware	CMD	Autostart
Name	Start	File name	Description				
NET CLR Prof...	ON	%systemroot%\system32\netprofperf.dll					
NET CLR Netwo...	ON	%systemroot%\system32\netprofperf.dll					
NET CLR Netwo...	ON	%systemroot%\system32\netprofperf.dll					
NET Data Provid...	ON	%systemroot%\system32\netprofperf.dll					
NET Data Provid...	ON	%systemroot%\system32\netprofperf.dll					
NET Framework Ca...	ON	%systemroot%\system32\netprofperf.dll					
NET Framework Ca...	ON	%systemroot%\system32\netprofperf.dll					
1394d.dll	MANUAL						
3ware	BOOT	System32\drivers\3ware.sys					
AacUpc	MANUAL	System32\drivers\AacUpc.sys					
AasVc_394d5	MANUAL	C:\Windows\System32\drivers\AasVc.dll					
ACPI	BOOT	System32\drivers\ACPI.sys					
AcpiPev	MANUAL	\SystemRoot\System32\drivers\AcpiPev.sys					
acpedit	ON	System32\drivers\Acpedit.exe					
acpigring	MANUAL	\SystemRoot\System32\drivers\acpigring.sys					
acpimline	MANUAL	\SystemRoot\System32\drivers\acpimline.sys					
AcpiT000	MANUAL	system32\drivers\AcpiT000.sys					
ADWCleaner.Pack...	BOOT	System32\drivers\ADWPB00.CSYS					
ADP89XX	BOOT	System32\drivers\ADP89XX.CSYS					
adsi							
AFS	SYSTEM	\SystemRoot\system32\drivers\afs.dll					
afunix	SYSTEM	\SystemRoot\system32\drivers\afunix.sys					
apache	SYSTEM	system32\IIS\Apache\apache					
AllRoute	MANUAL	\SystemRoot\System32\AllRouter.dll					
ALG	MANUAL	\SystemRoot\System32\alg.dll					
amdgpu0	MANUAL	\SystemRoot\System32\drivers\amdgpu0.sys					
amdgpu02	MANUAL	\SystemRoot\System32\drivers\amdgpu02.sys					
amds2	MANUAL	\SystemRoot\System32\drivers\amds2.sys					
AmkD8	MANUAL	\SystemRoot\System32\drivers\amkd8.sys					
AmnPDM	MANUAL	\SystemRoot\System32\drivers\amnpdm.sys					
amrude	BOOT	System32\drivers\amrude.sys					
amvba	BOOT	System32\drivers\amvba.sys					
andxwla	BOOT	System32\drivers\andxwla.sys					
ApplD	BOOT	System32\drivers\appld.sys					
ApplD\Scv	MANUAL	\SystemRoot\System32\drivers\appld\scv.dll					
ApplP	MANUAL	\SystemRoot\System32\drivers\applp.dll					
Apple Mobile Dev...	AUTO	"C:\Program Files\Common Files\Apple\Mobile ...	Provides the interface to Apple mobile devices.				
Apple\mdfilter	MANUAL	\SystemRoot\System32\drivers\Apple\mdfilter...					
Apple_lowFilter	MANUAL	\SystemRoot\System32\drivers\Apple\lowFilter...					
AppMgmt	MANUAL	\SystemRoot\System32\appmgmt.dll					
AppReadiness	MANUAL	\SystemRoot\System32\appreadiness.dll					
AppClient	DISABLED	\SystemRoot\System32\AppClient.exe					

Files

The screenshot shows the GMER 2.2.19802 interface. The left pane displays a file tree with the following structure:

- Processes
- Modules
- Services
- Files
- Registry
- Rootkit/Malware
- CMD
- Autostart

My Computer

- C:
 - DRIVERS
 - \$Recycle.Bin
 - \$WinREAgent
 - android
 - ADE
 - App
 - avast! sandbox
 - Config.Msi
 - Documents and Settings
 - Font
 - Java
 - ksensor
 - OneDriveTemp
 - PerfLogs
 - Program Files
 - Program Files (x86)
 - ProgramData
 - Recovery
 - Root
 - System Volume Information
 - temp
 - Users
 - Windows
 - xampp
 - D:

The right pane shows a detailed list of files and folders under C:\, including:

 - \$Recycle.Bin
 - \$WinREAgent
 - android
 - ADE
 - App
 - avast! sandbox
 - Config.Msi
 - Documents and Settings
 - DRIVERS
 - Intel
 - Java
 - ksensor
 - OneDriveTemp
 - PerfLogs
 - Program Files
 - Program Files (x86)
 - ProgramData
 - Recovery
 - Snort
 - System Volume Information
 - temp
 - User
 - Windows
 - xampp
 - \$WinRE_BACKUP_PARTITION_MARKER
 - DumpStack.log
 - DumpStack.log.tmp
 - hiberfil.sys
 - swappile.sys

On the far right, there are buttons for Delete, Copy, Kill, and a checkbox for Only hidden.

GMER 2.2.19802 WINDOWS 6.2.9200 x64 AntiVirus: http://www.avast.com

Registry



12. TCPDump & Dumpcap

Show all packets

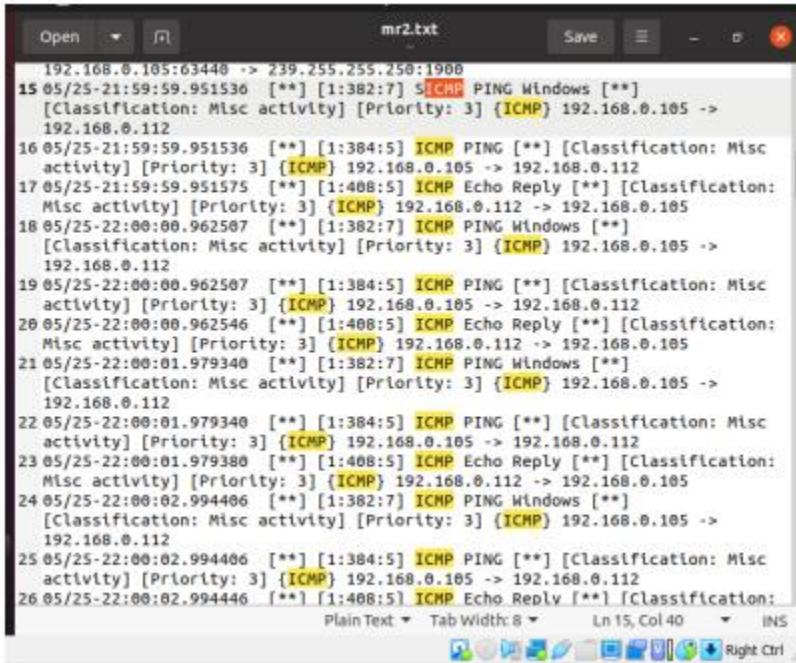
Display all interfaces and find packets of a specific interface

```
aditi@aditi-VirtualBox:~$ sudo tcpdump -D
1.eth0 [Up, Running]
2.lo [Up, Running, Loopback]
3.any [Pseudo-device that captures on all interfaces] [Up, Running]
4.bluetooth-monitor (Bluetooth Linux Monitor) [none]
5.nflog (Linux netfilter log (NFLOG) interface) [none]
6.nfqueue (Linux netfilter queue (NFQUEUE) interface) [none]
aditi@aditi-VirtualBox:~$ sudo tcpdump -l eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
17:50:39.039787 IP aditi-VirtualBox.42140 > alpyn.canonical.com.ntp: NTPv4, Client, length 48
17:50:39.043894 IP aditi-VirtualBox.43998 > 192.168.0.1.domain: 48357+ PTR? 15.2.0.10.in-addr.arpa. (40)
17:50:39.264435 IP alpyn.canonical.com.ntp > aditi-VirtualBox.42140: NTPv4, Server, length 48
17:50:42.125665 IP 192.168.0.1.domain > aditi-VirtualBox.43998: 48357 NXDomain* 0/1/0 (99)
17:50:42.128123 IP aditi-VirtualBox.42644 > 192.168.0.1.domain: 22235+ PTR? 1.0.168.192.in-addr.arpa. (42)
17:50:44.098656 ARP, Request who-has _gateway tell aditi-VirtualBox, length 28
17:50:44.099428 ARP, Reply _gateway is-at 52:54:00:12:35:02 (out Unknown), length 46
17:50:46.139995 IP 192.168.0.1.domain > aditi-VirtualBox.42644: 22235 NXDomain* 0/1/0 (101)
17:50:46.142758 IP aditi-VirtualBox.60237 > 192.168.0.1.domain: 46762+ PTR? 2.2.0.10.in-addr.arpa. (39)
17:50:47.268502 IP aditi-VirtualBox.38276 > 32.121.122.34.bc.googleusercontent.com.http: Flags [S], seq 3227714833, win 64240, options [mss 14
60,sackOK,TS val 1898936705 ecr 0,nop,wscale 7], length 0
17:50:48.298597 IP aditi-VirtualBox.38276 > 32.121.122.34.bc.googleusercontent.com.http: Flags [S], seq 3227714833, win 64240, options [mss 14
60,sackOK,TS val 1898937735 ecr 0,nop,wscale 7], length 0
17:50:50.314369 IP aditi-VirtualBox.38276 > 32.121.122.34.bc.googleusercontent.com.http: Flags [S], seq 3227714833, win 64240, options [mss 14
60,sackOK,TS val 1898939750 ecr 0,nop,wscale 7], length 0
17:50:50.879377 IP 192.168.0.1.domain > aditi-VirtualBox.60237: 46762 NXDomain* 0/1/0 (98)
17:50:50.881620 IP aditi-VirtualBox.50364 > 192.168.0.1.domain: 9225+ PTR? 32.121.122.34.in-addr.arpa. (44)
17:50:51.093747 IP 192.168.0.1.domain > aditi-VirtualBox.50364: 9225 1/0/0 PTR 32.121.122.34.bc.googleusercontent.com. (96)
17:50:54.511007 IP 32.121.122.34.bc.googleusercontent.com.http > aditi-VirtualBox.38276: Flags [S.], seq 14592001, ack 3227714834, win 65535,
options [mss 1460], length 0
17:50:54.511139 IP aditi-VirtualBox.38276 > 32.121.122.34.bc.googleusercontent.com.http: Flags [.], ack 1, win 64240, length 0
17:50:54.511598 IP aditi-VirtualBox.38276 > 32.121.122.34.bc.googleusercontent.com.http: Flags [P.], seq 1:88, ack 1, win 64240, length 87: HT
TP- GET / HTTP/1.1
```

Display packets information in ASCII format

```
aditi@aditi-VirtualBox:~$ sudo tcpdump -A -i eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
18:04:20.477440 IP 192.168.0.1.domain > aditi-VirtualBox.41659: 552 0/1/0 (108)
E.....@.....
....5....t....(.....connectivity-check.ubuntu.com.....1.ns1 canonical.&
hostmaster.7xI....*0.... :.....
18:04:20.482743 IP aditi-VirtualBox.47196 > 192.168.0.1.domain: 37962+ PTR? 15.2.0.10.in-addr.arpa. (40)
E..D..0..0..c
.....\5.0...J.....15.2.0.10.in-addr.arpa.....
18:04:20.484014 IP aditi-VirtualBox.57216 > 192.168.0.1.domain: 24604+ AAAA? connectivity-check.ubuntu.com. (47)
E..K..0..@..[.....
.....5.7... . ....connectivity-check.ubuntu.com.....
18:04:22.506175 ARP, Request who-has _gateway tell aditi-VirtualBox, length 28
'.....
'.....
18:04:22.506453 ARP, Reply _gateway is-at 52:54:00:12:35:02 (oui Unknown), length 46
.....RT..5.
'.....
'.....
18:04:24.475752 IP 192.168.0.1.domain > aditi-VirtualBox.57216: 24604 0/0/0 (47)
E...K...@.. .....
....5....t....(.....connectivity-check.ubuntu.com.....
18:04:24.475826 IP 192.168.0.1.domain > aditi-VirtualBox.47196: 37962 NXDomain* 0/1/0 (99)
E.....@.....
....5.\k...J.....15.2.0.10.in-addr.arpa.....*0./ localhost.nobody.invalid..... :...*0
18:04:24.480427 IP aditi-VirtualBox.59691 > 192.168.0.1.domain: 9904+ AAAA? connectivity-check.ubuntu.com.domain.name. (59)
E..W..30..@..^.
.....+..5..C..&.....connectivity-check.ubuntu.com.domain.name.....
18:04:24.481496 IP aditi-VirtualBox.57505 > 192.168.0.1.domain: 22715+ PTR? 1.0.168.192.in-addr.arpa. (42)
E..F..40..@..^.
```

13. Snort Tool



The screenshot shows a window titled "mr2.txt" containing a list of network log entries. The entries are timestamped and describe ICMP PING and Echo Reply events between two hosts, 192.168.0.105 and 192.168.0.112. The log entries are as follows:

```
192.168.0.105:63448 -> 239.255.255.250:1900
15 05/25-21:59:59.951530 [**] [1:382:7] 5 [ICMP] PING Windows [**]
[Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.105 ->
192.168.0.112
16 05/25-21:59:59.951536 [**] [1:384:5] ICMP PING [**] [Classification: Misc
activity] [Priority: 3] {ICMP} 192.168.0.105 -> 192.168.0.112
17 05/25-21:59:59.951575 [**] [1:408:5] ICMP Echo Reply [**] [Classification:
Misc activity] [Priority: 3] {ICMP} 192.168.0.112 -> 192.168.0.105
18 05/25-22:00:00.962507 [**] [1:382:7] ICMP PING Windows [**]
[Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.105 ->
192.168.0.112
19 05/25-22:00:00.962507 [**] [1:384:5] ICMP PING [**] [Classification: Misc
activity] [Priority: 3] {ICMP} 192.168.0.105 -> 192.168.0.112
20 05/25-22:00:00.962546 [**] [1:408:5] ICMP Echo Reply [**] [Classification:
Misc activity] [Priority: 3] {ICMP} 192.168.0.112 -> 192.168.0.105
21 05/25-22:00:01.979340 [**] [1:382:7] ICMP PING Windows [**]
[Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.105 ->
192.168.0.112
22 05/25-22:00:01.979340 [**] [1:384:5] ICMP PING [**] [Classification: Misc
activity] [Priority: 3] {ICMP} 192.168.0.105 -> 192.168.0.112
23 05/25-22:00:01.979380 [**] [1:408:5] ICMP Echo Reply [**] [Classification:
Misc activity] [Priority: 3] {ICMP} 192.168.0.112 -> 192.168.0.105
24 05/25-22:00:02.994406 [**] [1:382:7] ICMP PING Windows [**]
[Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.105 ->
192.168.0.112
25 05/25-22:00:02.994406 [**] [1:384:5] ICMP PING [**] [Classification: Misc
activity] [Priority: 3] {ICMP} 192.168.0.105 -> 192.168.0.112
26 05/25-22:00:02.994446 [**] [1:408:5] ICMP Echo Reply [**] [Classification:
```

At the bottom of the window, there are standard text editor controls: Plain Text, Tab Width: 8, Ln 15, Col 40, and INS.