Ex. No: 1(a) IMPLEMENTATION OF CAESAR CIPHER

AIM

To write a Java Program for implementing Caesar Cipher.

ALGORITHM

Step 1: Start the program.

Step 2: Define a class CaesarCipher.

Step 3: Declare a string ALPHABET.

Step 4: Define a function encrypt() to produce a cipher text and decrypt() to reproduce the plain text.

Step 5: Define a main(), get the string and call encrypt() to encrypt the string and decrypt() to reproduce the plain text and display it.

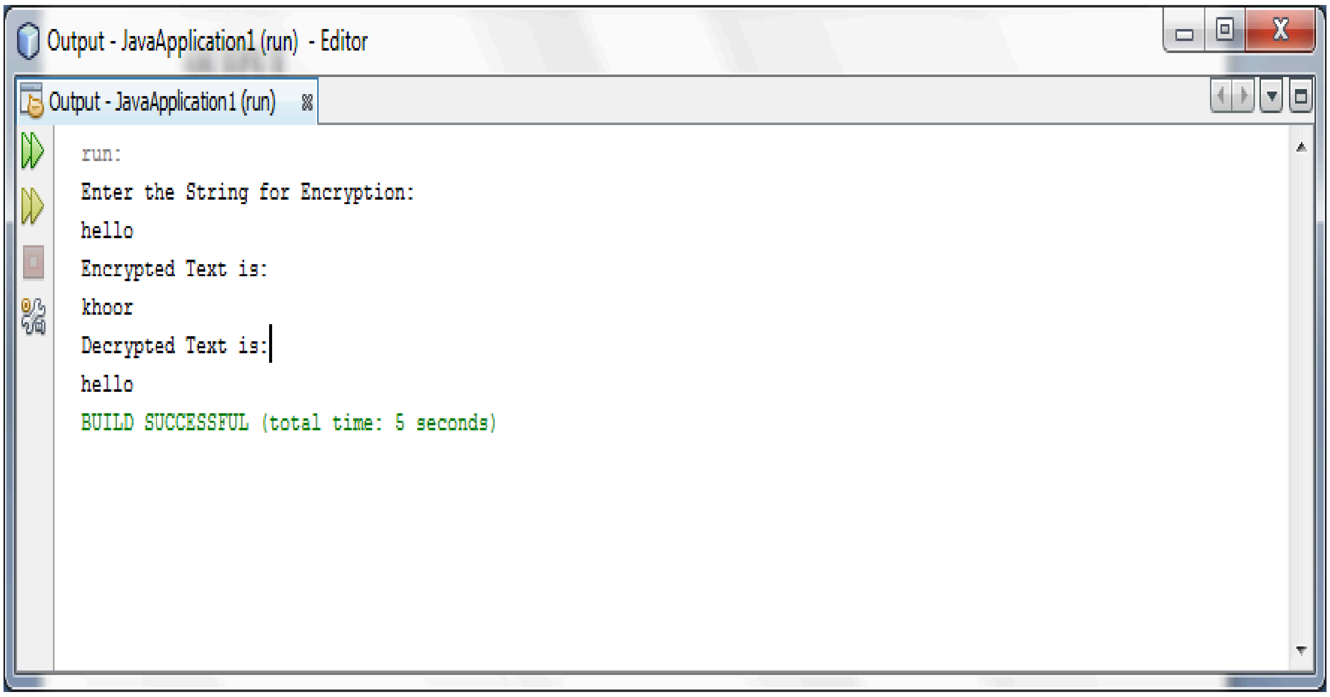
Step 6: Stop the program.

PROGRAM

```
package javaapplication1;
import java.util.Scanner;
public class CaesarCipher
{
	public static final String ALPHABET = "abcdefghijklmnopqrstuvwxyz";
	public static String encrypt(String plainText, int shiftKey)
	{
		plainText = plainText.toLowerCase();
		String cipherText = "";
		for (int i = 0; i < plainText.length(); i++)
		{
			int charPosition = ALPHABET.indexOf(plainText.charAt(i));
			int keyVal = (shiftKey + charPosition) % 26;
			char replaceVal = ALPHABET.charAt(keyVal);
			cipherText += replaceVal;
		}
		return cipherText;
	}

	public static String decrypt(String cipherText, int shiftKey)
	{
		cipherText = cipherText.toLowerCase();
		String plainText = "";
		for (int i = 0; i < cipherText.length(); i++)
		{
			int charPosition = ALPHABET.indexOf(cipherText.charAt(i));
			int keyVal = (charPosition - shiftKey) % 26;
			if (keyVal < 0)
			{
				keyVal = ALPHABET.length() + keyVal;
			}
```

```
                    char replaceVal = ALPHABET.charAt(keyVal);
                    plainText += replaceVal;
                }
                return plainText;
        }
        public static void main(String[] args)
        {
                try (Scanner sc = new Scanner(System.in))
                {
                        System.out.println("Enter the String for Encryption: ");
                        String message = new String();
                        message = sc.next();
                        System.out.println("Encrypted Text is:");
                        System.out.println(encrypt(message, 3));
                        System.out.println("Decrypted Text is:");
                        System.out.println(decrypt(encrypt(message, 3), 3));
                }
        }
}
```

OUTPUT


Output - JavaApplication1 (run) - Editor
Output - JavaApplication1 (run)

```
run:
Enter the String for Encryption:
hello
Encrypted Text is:
khoor
Decrypted Text is:
hello
BUILD SUCCESSFUL (total time: 5 seconds)
```

RESULT

Thus java program to implement Caesar Cipher was written, executed and output is verified successfully.

Ex.No: 1(b) IMPLEMENTATION OF PLAYFAIR CIPHER

AIM

To write a Java program for implementing Playfair Cipher.

ALGORITHM

Step 1: Start the program.

Step 2: Define a class Basic to find the index of a char.

Step 3: Define a class PlayFair to define the key matrix, find the row position, column position, encrypt the text and decrypt the text.

Step 4: Define a class PlayFairCipher, to get the plain text and to encrypt and decrypt text.

Step 5: Display the encrypted text and decrypted text.

Step 6: Stop the program.

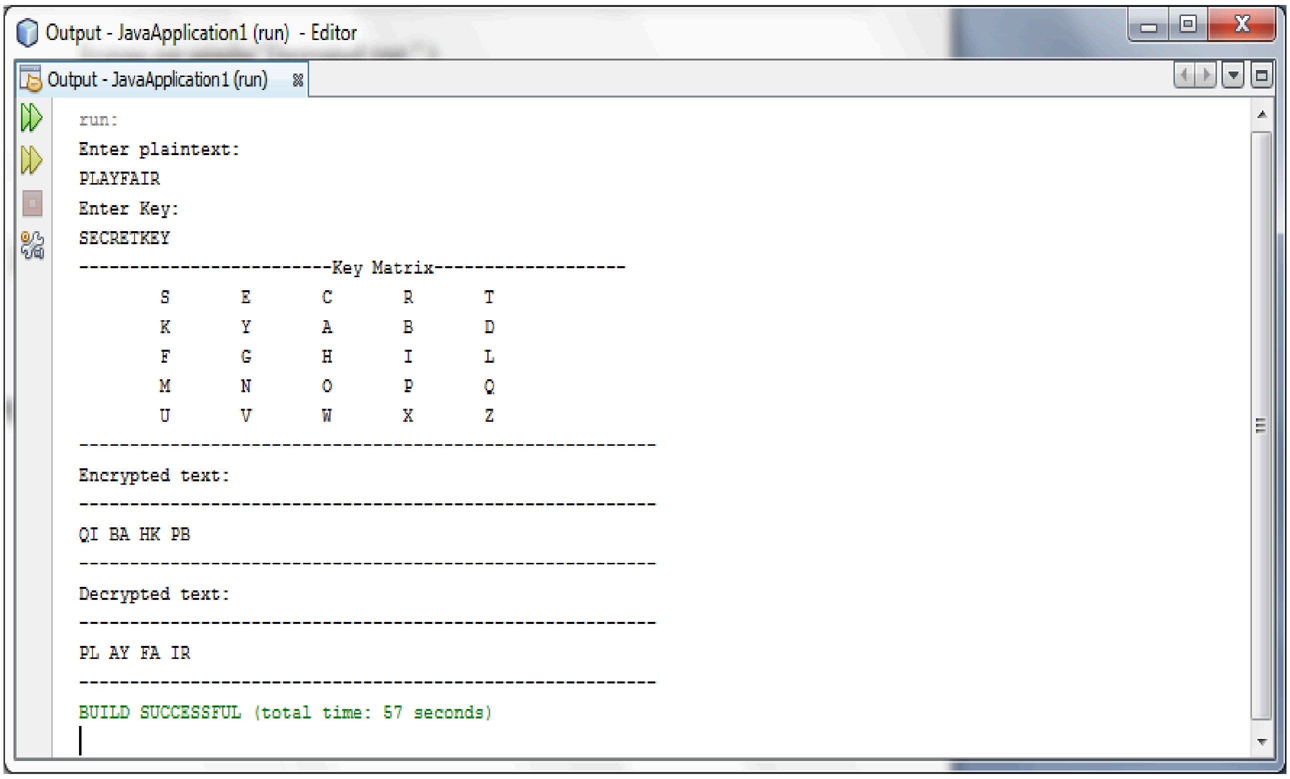
PROGRAM

```
package javaapplication1;
import java.util.*;
class Basic
{
        String allChar="ABCDEFGHIJKLMNOPQRSTUVWXYZ";
        boolean indexOfChar(char c)
        {
                for(int i=0;i < allChar.length();i++)
                {       if(allChar.charAt(i)==c)
                                return true;
                }return false;
        }
}
 class PlayFair
{       Basic b=new Basic();
        char keyMatrix[][]=new char[5][5];
        boolean repeat(char c)
        {       if(!b.indexOfChar(c))
                {       return true;
                }
                for(int i=0;i < keyMatrix.length;i++)
                {       for(int j=0;j < keyMatrix[i].length;j++)

                                        return true;
                        }
                } return false;
        }
        void insertKey(String key)
        {       key=key.toUpperCase();
                key=key.replaceAll("J", "I");
                key=key.replaceAll(" ", "");

                int a=0,b=0;
                for(int k=0;k < key.length();k++)
                        {       if(keyMatrix[i][j]==c || c=='J')
                {       if(!repeat(key.charAt(k)))
                        {       keyMatrix[a][b++]=key.charAt(k);
                                if(b>4)
                                {       b=0;
                                        a++;
                                }
                        }
                }
                char p='A';
                while(a < 5)
                {       while(b < 5)
                        {       if(!repeat(p))
                                {       keyMatrix[a][b++]=p;
                                }       p++;
                        }       b=0;
                                a++;
                }System.out.print("-------------------------Key Matrix-------------------");
                for(int i=0;i < 5;i++)
                {       System.out.println();
                        for(int j=0;j < 5;j++)
                        System.out.print("\t"+keyMatrix[i][j]);
                }
                System.out.println("\n-------------------------------------------------------");
        }int rowPos(char c)
        {       for(int i=0;i < keyMatrix.length;i++)
                {       for(int j=0;j < keyMatrix[i].length;j++)
                        {       if(keyMatrix[i][j]==c)
                                        return i;
                        }
                } return -1;
        }
        int columnPos(char c)
        {       for(int i=0;i < keyMatrix.length;i++)
                {       for(int j=0;j < keyMatrix[i].length;j++)
                        {       if(keyMatrix[i][j]==c)
                                        return j;
                        }
                }return -1;
                }
        String encryptChar(String plain)
        {       plain=plain.toUpperCase();
                char a=plain.charAt(0),b=plain.charAt(1);
                String cipherChar="";
                int r1,c1,r2,c2;
                r1=rowPos(a);
```

```
                c1=columnPos(a);
                r2=rowPos(b);
                c2=columnPos(b);
                if(c1==c2)
                {       ++r1;
                        ++r2;
        if(r1>4)
                r1=0;
        if(r2>4)
                r2=0;
cipherChar+=keyMatrix[r1][c2];
                        cipherChar+=keyMatrix[r2][c1];
                }
                else if(r1==r2)
                {       ++c1;
                        ++c2;
                        if(c1>4)
                                c1=0;
                        if(c2>4)
                                c2=0;
                        cipherChar+=keyMatrix[r1][c1];
                        cipherChar+=keyMatrix[r2][c2];
                }
                else
                {       cipherChar+=keyMatrix[r1][c2];
                        cipherChar+=keyMatrix[r2][c1];
                }   return cipherChar;
        }
        String Encrypt(String plainText,String key)
        {       insertKey(key);
                String cipherText="";
                plainText=plainText.replaceAll("j", "i");
                plainText=plainText.replaceAll(" ", "");
                plainText=plainText.toUpperCase();
                int len=plainText.length();
                if(len/2!=0)
                {       plainText+="X";
                        ++len;
                }
                for(int i=0;i < len-1;i=i+2)
                {       cipherText+=encryptChar(plainText.substring(i,i+2));
                        cipherText+=" ";
                }  return cipherText;       }
        String decryptChar(String cipher)
        {       cipher=cipher.toUpperCase();
                char a=cipher.charAt(0),b=cipher.charAt(1);
        String plainChar="";
        int r1,c1,r2,c2;
        r1=rowPos(a);
        c1=columnPos(a);
        r2=rowPos(b);
        c2=columnPos(b);
        if(c1==c2)
        {       --r1;
                --r2;
                if(r1 < 0)
                        r1=4;
                if(r2 < 0)
                        r2=4;
                plainChar+=keyMatrix[r1][c2];
                plainChar+=keyMatrix[r2][c1];
        }
        else if(r1==r2)
        {
                --c1;
                --c2;
                if(c1 < 0)
                        c1=4;
                if(c2 < 0)
                        c2=4;
                plainChar+=keyMatrix[r1][c1];
                plainChar+=keyMatrix[r2][c2];
        }
        else
        {       plainChar+=keyMatrix[r1][c2];
                plainChar+=keyMatrix[r2][c1];
        }       return plainChar;
}
String Decrypt(String cipherText,String key)
{       String plainText="";
        cipherText=cipherText.replaceAll("j", "i");
        cipherText=cipherText.replaceAll(" ", "");
        cipherText=cipherText.toUpperCase();
        int len=cipherText.length();
        for(int i=0;i < len-1;i=i+2)
        {       plainText+=decryptChar(cipherText.substring(i,i+2));
                plainText+=" ";
        }       return plainText;
}
```

```
class PlayfairCipher
{
        public static void main(String args[])throws Exception
        {
                PlayFair p=new PlayFair();
                Scanner scn=new Scanner(System.in);
                String key,cipherText,plainText;
                System.out.println("Enter plaintext:");
                plainText=scn.nextLine();
                System.out.println("Enter Key:");
                key=scn.nextLine();
                cipherText=p.Encrypt(plainText,key);
                System.out.println("Encrypted text:");
                System.out.println("-------------------------------------------------------\n"+cipherText);
                 System.out.println("-------------------------------------------------------");
                String encryptedText=p.Decrypt(cipherText, key);
                System.out.println("Decrypted text:" );
                System.out.println("-------------------------------------------------------\n"+encryptedTe
                xt);
                System.out.println("-------------------------------------------------------");
        }
}
```

OUTPUT


Output - JavaApplication1 (run) - Editor

Output - JavaApplication1 (run)

```
run:
Enter plaintext:
PLAYFAIR
Enter Key:
SECRETKEY
-------------------------Key Matrix------------------
        S       E       C       R       T
        K       Y       A       B       D
        F       G       H       I       L
        M       N       O       P       Q
        U       V       W       X       Z
---------------------------------------------------------
Encrypted text:
---------------------------------------------------------
QI BA HK PB
---------------------------------------------------------
Decrypted text:
---------------------------------------------------------
PL AY FA IR
---------------------------------------------------------
BUILD SUCCESSFUL (total time: 57 seconds)
```

RESULT

Thus java program to implement PlayFair Cipher was written, executed and output is verified successfully.

Ex.No: 1(c) IMPLEMENTATION OF HILL CIPHER

AIM

To write a Java Program for implementing Hill Cipher.

ALGORITHM

Step 1: Start the program.

Step 2: Define a class HillCipher, in that declare 2 array one for key, other for inverse key and define a string key.

Step 3: In this class, define a main(), get the choice to encrypt or decrypt.

Step 4: Based on the choice call encrypt() and decrypt() to find cipher and plain text.

Step 5: Display the result.

Step 6: Stop the program.

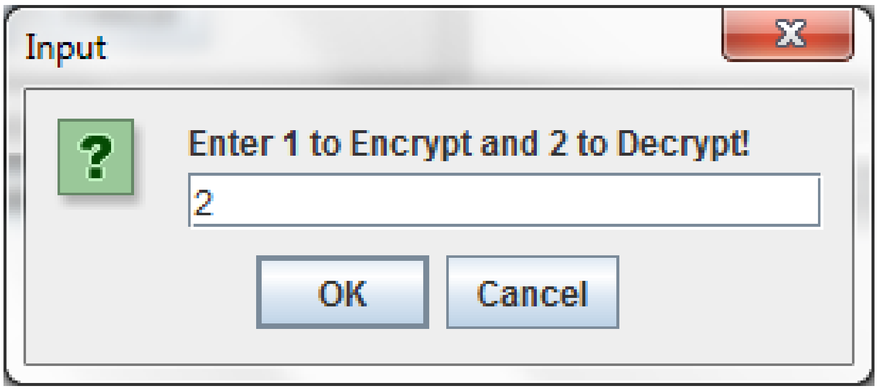
PROGRAM

```
package javaapplication1;
import javax.swing.JOptionPane;
public class HillCipher
{//the 3x3 key matrix for 3 characters at once
        public static int[][] keymat = new int[][]{
           { 1, 2, 1 },
           { 2, 3, 2 },
           { 2, 2, 1 },
        };
        public static int[][] invkeymat = new int[][]{
           { -1, 0, 1 },
           { 2, -1, 0 },
           { -2, 2, -1},
        };
        public static String key = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
        public static void main(String[] args)
        { // TODO code application logic here
                String text,outtext ="";
                int ch, n;
                ch = Integer.parseInt(JOptionPane.showInputDialog(null, "Enter 1 to Encrypt and 2
        to Decrypt!"));
                text = JOptionPane.showInputDialog(null, "Enter plain/cipher text to encrypt?");
                text = text.toUpperCase();
                text = text.replaceAll("\\s",""); //removing spaces
                n = text.length() % 3;
                if(n!=0)
                {       for(int i = 1; i<= (3-n);i++)
                { text+= 'X';
                        }
```

```
		}
		System.out.println("Padded Text:" + text);
		char[] ptextchars = text.toCharArray();
		switch(ch)
		{
			case 1:
			for(int i=0;i< text.length(); i+=3)
			{
				outtext += encrypt(ptextchars[i],ptextchars[i+1],ptextchars[i+2]);
			}
			break;
			case 2:
			for(int i=0;i< text.length(); i+=3)
			{
				outtext += decrypt(ptextchars[i],ptextchars[i+1],ptextchars[i+2]);
			}
			break;
			default: System.out.println("Invalid Choice!");
		}
		System.out.println("Output: " + outtext);
	}


	private static String encrypt(char a, char b, char c)
	{
		String ret = "";
		int x,y, z;
		int posa = (int)a - 65;
		int posb = (int)b - 65;
		int posc = (int)c - 65;
		x = posa * keymat[0][0] + posb * keymat[1][0] + posc * keymat[2][0];
		y = posa * keymat[0][1] + posb * keymat[1][1] + posc * keymat[2][1];
		z = posa * keymat[0][2] + posb * keymat[1][2] + posc * keymat[2][2];
		a = key.charAt(x%26);
		b = key.charAt(y%26);
		c = key.charAt(z%26);
		 ret = "" + a + b + c;
		return ret;
	}
	private static String decrypt(char a, char b, char c)
	{
		String ret = "";
		int x,y,z;
		int posa = (int)a - 65;
```

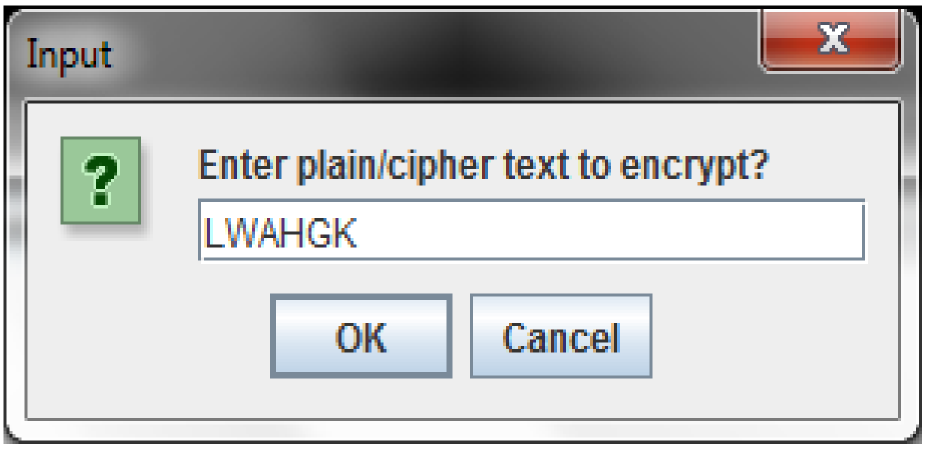
```
                int posb = (int)b - 65;
                int posc = (int)c - 65;
                x = posa * invkeymat[0][0]+ posb * invkeymat[1][0] + posc * invkeymat[2][0];
                y = posa * invkeymat[0][1]+ posb * invkeymat[1][1] + posc * invkeymat[2][1];
                z = posa * invkeymat[0][2]+ posb * invkeymat[1][2] + posc * invkeymat[2][2];
                a = key.charAt((x%26<0)?(26+x%26):(x%26));
                b = key.charAt((y%26<0)?(26+y%26):(y%26));
                c = key.charAt((z%26<0)?(26+z%26):(z%26));
                ret = "" + a + b + c;
                return ret;
        }
}
```

DECRYPT


Input

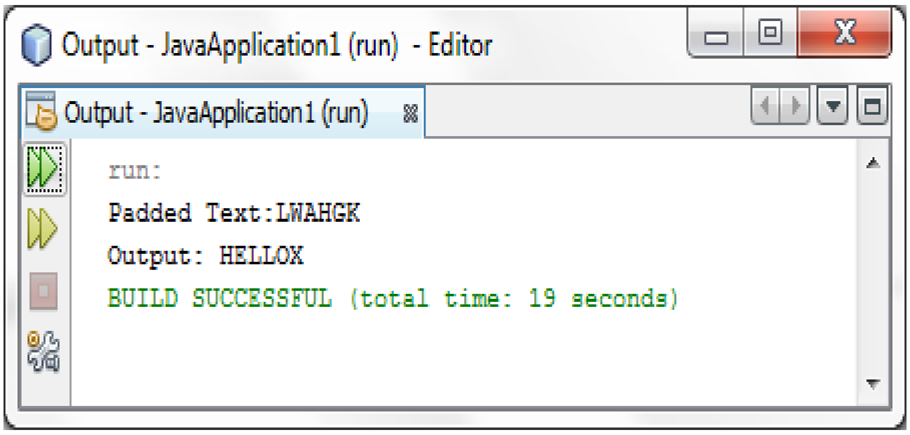
Enter 1 to Encrypt and 2 to Decrypt!

2


Input

Enter plain/cipher text to encrypt?

LWAHGK


Output - JavaApplication1 (run) - Editor
Output - JavaApplication1 (run)

```
run:
Padded Text:LWAHGK
Output: HELLOX
BUILD SUCCESSFUL (total time: 19 seconds)
```

RESULT

Thus java program to implement Hill Cipher was written, executed and output is verified successfully.

Ex.No: 1(d) **IMPLEMENTATION OF VIGENERE CIPHER**

AIM

To write a Java Program for implementing Vigenere Cipher.

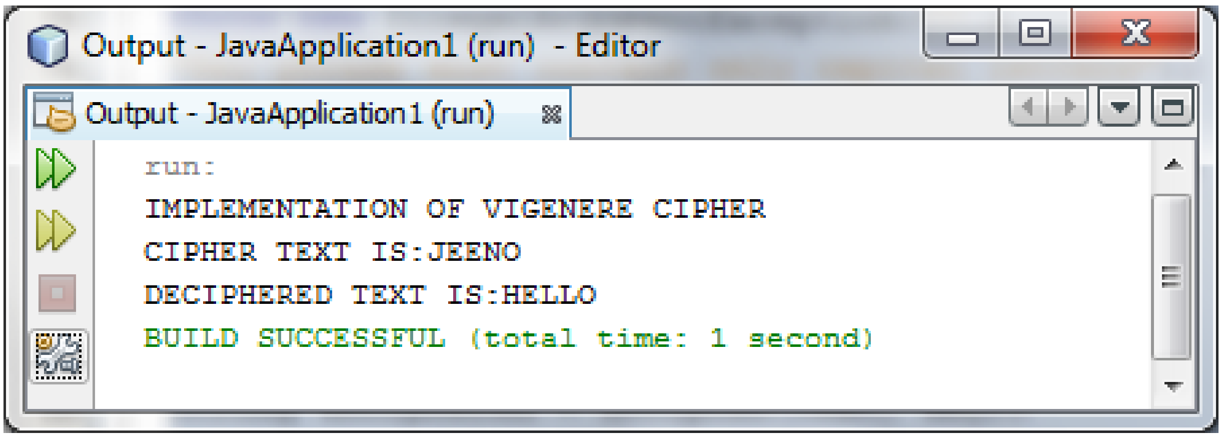
ALGORITHM

Step 1: Start the program.
Step 2: Define a class VC1, in that define encipher() to produce a cipher text.
Step 3: Define decipher() to reproduce the plain text.
Step 4: Define a shift() to shift the values.
Step 5: In main(), define the text and key values and call the encipher() to encrypt and decipher() to decrypt the encrypted text.
Step 6: Display the results.
Step 7: Stop the program.

PROGRAM

```
package javaapplication1;
public class VC1
{
        public static String encipher(String s, String key)
        {
                StringBuilder builder = new StringBuilder();
                for(int i = 0; i < s.length(); i ++)
                {
                        if(s.charAt(i) < 65 || s.charAt(i) > 90)
                        { //ASCII character (capital letter)
                                throw new IllegalArgumentException("" +"Open text must contain
                                only capital letters");
                        }
                        //add shift modularly
                        char encyphered = s.charAt(i) + getShift(key, i) > 90 ? (char)((s.charAt(i) +
                        getShift(key, i)) - 26) : (char)(s.charAt(i) + getShift(key, i));
                        builder.append(encyphered);
                }
                return builder.toString();
        }
        public static String decipher(String s, String key)
        {
                StringBuilder builder = new StringBuilder();
                for(int i = 0; i < s.length(); i ++)
                {
                        if(s.charAt(i) < 65 || s.charAt(i) > 90)
                        { //ASCII character (capital letter)
                                throw new IllegalArgumentException("" +"Ciphertext must contain
                                only capital letters");
                        }
```

```
                    //subtract shift modularly
                    char decyphered = s.charAt(i) - getShift(key, i) < 65 ? (char)((s.charAt(i) -
                    getShift(key, i)) + 26) : (char)(s.charAt(i) - getShift(key, i));
                    builder.append(decyphered);
            }
            return builder.toString();
      }
      private static int getShift(String key, int i)
      {
            if(key.charAt(i % key.length()) < 65 || key.charAt(i % key.length()) > 90)
            {
                    throw new IllegalArgumentException("" +"Key phrase must contain only
                    capital letters");
            }
            return ((int)key.charAt(i % key.length())) - 65;
      }
      public static void main(String[] args)
      {
            String text = "HELLO";
            String key = "CAT";
            String enciphered = encipher(text, key);
            System.out.println("IMPLEMENTATION OF VIGENERE CIPHER");
            System.out.println("CIPHER TEXT IS:"+enciphered);
            System.out.println("DECIPHERED TEXT IS:"+decipher(enciphered, key));
      }
}
```



Output - JavaApplication1 (run) - Editor
Output - JavaApplication1 (run)

```
run:
IMPLEMENTATION OF VIGENERE CIPHER
CIPHER TEXT IS:JEENO
DECIPHERED TEXT IS:HELLO
BUILD SUCCESSFUL (total time: 1 second)
```

RESULT

Thus java program to implement Vigenere Cipher was written, executed and output is verified successfully.

Ex.No: 2 **IMPLEMENTATION OF RAIL FENCE – ROW & COLUMN TRANSFORMATION**

AIM

To write a Java Program for implementing Rail Fence – Row & Column Transformation.

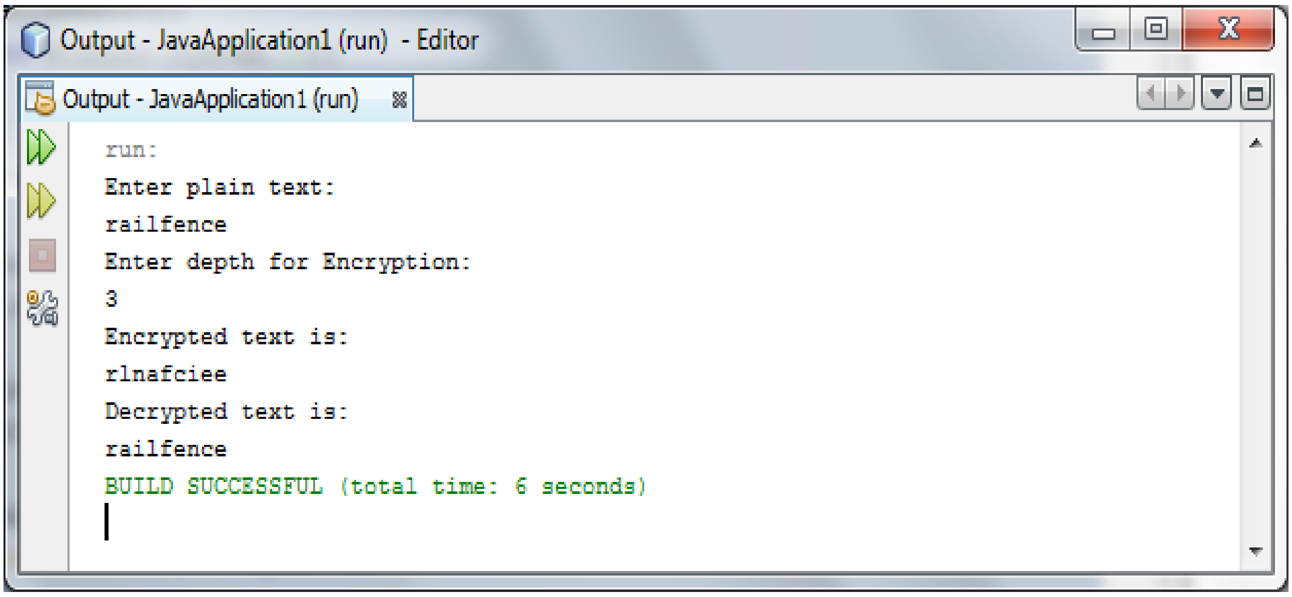
ALGORITHM

Step 1: Start the program.
Step 2: Define a class railfencebasic, in that define Encryption() to produce cipher text.
Step 3: Define Decryption() to produce decipher text.
Step 4: Define a class railfencecipher, in that define main() and input the text to cipher and decipher it.
Step 5: Display the results.
Step 6: Stop the program.

PROGRAM

```
package javaapplication1;
import java.util.*;
class RailFenceBasic
{
        int depth;
        String Encryption(String plainText,int depth)throws Exception
        {
                int r=depth,len=plainText.length();
                int c=len/depth;
                char mat[][]=new char[r][c];
                int k=0;
                String cipherText="";
                for(int i=0;i< c;i++)
                {       for(int j=0;j< r;j++)
                        {       if(k!=len)
                                        mat[j][i]=plainText.charAt(k++);
                                else    mat[j][i]='X';
                        }
                }
                for(int i=0;i< r;i++)
                {       for(int j=0;j< c;j++)
                        {       cipherText+=mat[i][j];
                        }
}       return cipherText;
        }

        String Decryption(String cipherText,int depth)throws Exception
        {
                int r=depth,len=cipherText.length();
                int c=len/depth;
                char mat[][]=new char[r][c];
                int k=0;
                String plainText="";
                for(int i=0;i< r;i++)
                {
                        for(int j=0;j< c;j++)
                        {
                                mat[i][j]=cipherText.charAt(k++);
                        }
                }
                for(int i=0;i< c;i++)
                {       for(int j=0;j< r;j++)
                        {       plainText+=mat[j][i];
                        }
                }       return plainText; }
}
 class railfencecipher
{
        public static void main(String args[])throws Exception
        {       RailFenceBasic rf=new RailFenceBasic();
                Scanner scn=new Scanner(System.in);
                int depth;
                String plainText,cipherText,decryptedText;
                System.out.println("Enter plain text:");
                plainText=scn.nextLine();
                System.out.println("Enter depth for Encryption:");
                depth=scn.nextInt();
                cipherText=rf.Encryption(plainText,depth);
                System.out.println("Encrypted text is:\n"+cipherText);
                decryptedText=rf.Decryption(cipherText, depth);
                System.out.println("Decrypted text is:\n"+decryptedText);
        }
}
```

OUTPUT


Output - JavaApplication1 (run) - Editor
Output - JavaApplication1 (run)

```
run:
Enter plain text:
railfence
Enter depth for Encryption:
3
Encrypted text is:
rlnafciee
Decrypted text is:
railfence
BUILD SUCCESSFUL (total time: 6 seconds)
```

RESULT

Thus java program to implement Rail Fence – Row & Column Transformation was written, executed and output is verified successfully.

Ex.No: 3 **IMPLEMENTATION OF DES**

AIM

To write a Java Program for implementing DES.

ALGORITHM

Step 1: Start the program.
Step 2: Define a class DES, in that define a constructor to display the encrypted and decrypted message.
Step 3: Define generateSymmetricKey(), to generate the key.
Step 4: Define encrypt() to encrypt the plain text.
Step 5: Define decrypt() to decrypt the ciphered text.
Step 6: Define main() to declare object for the class.
Step 7: Stop the program.

PROGRAM

```
import java.util.*;

class DES1 {
  private static class DES {
    // CONSTANTS
    // Initial Permutation Table
    int[] IP = { 58, 50, 42, 34, 26, 18,
            10, 2, 60, 52, 44, 36, 28, 20,
            12, 4, 62, 54, 46, 38,
            30, 22, 14, 6, 64, 56,
            48, 40, 32, 24, 16, 8,
            57, 49, 41, 33, 25, 17,
            9, 1, 59, 51, 43, 35, 27,
            19, 11, 3, 61, 53, 45,
            37, 29, 21, 13, 5, 63, 55,
            47, 39, 31, 23, 15, 7 };

    // Inverse Initial Permutation Table
    int[] IP1 = { 40, 8, 48, 16, 56, 24, 64,
             32, 39, 7, 47, 15, 55,
             23, 63, 31, 38, 6, 46,
             14, 54, 22, 62, 30, 37,
             5, 45, 13, 53, 21, 61,
             29, 36, 4, 44, 12, 52,
             20, 60, 28, 35, 3, 43,
             11, 51, 19, 59, 27, 34,
             2, 42, 10, 50, 18, 58,
             26, 33, 1, 41, 9, 49,
             17, 57, 25 };
```

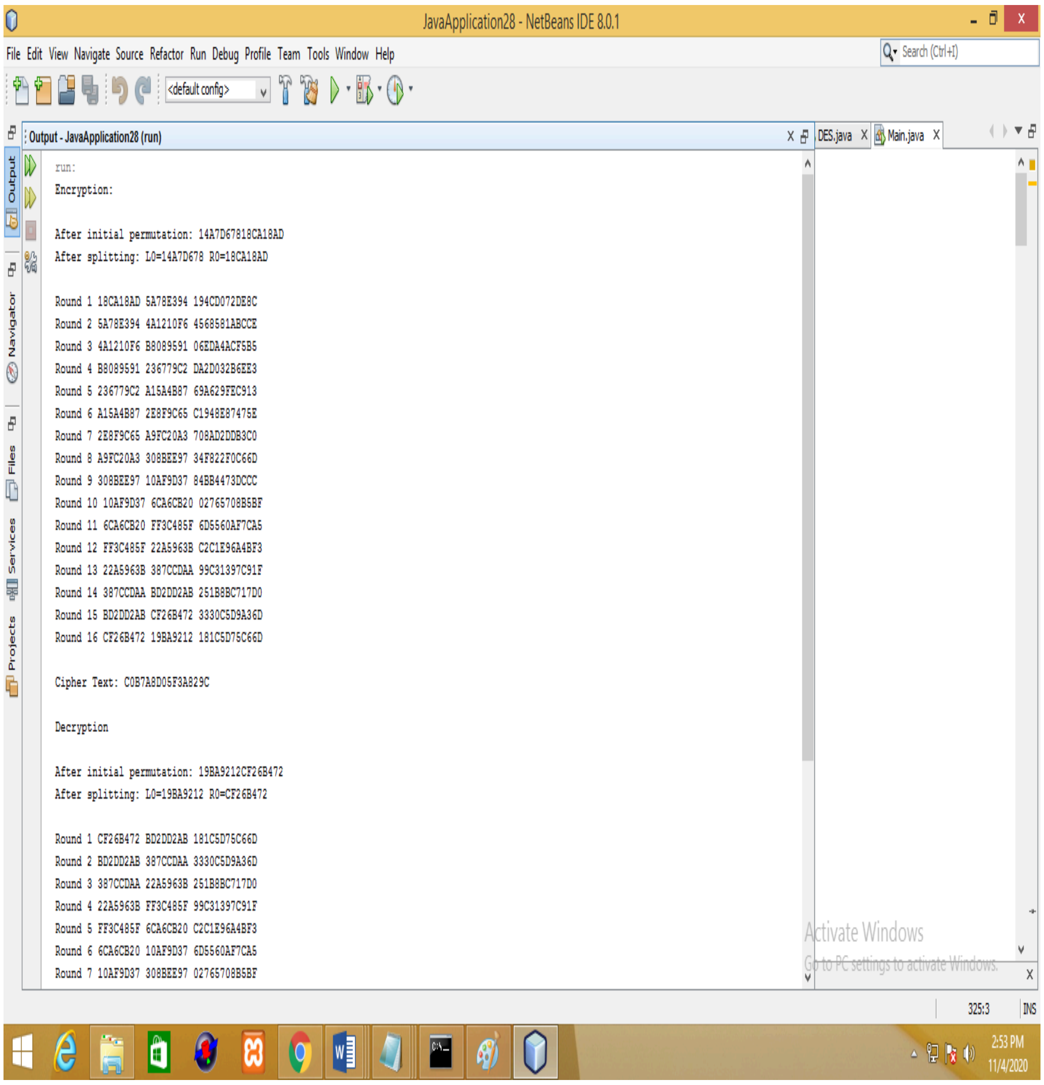
```
// first key-hePermutation Table
int[] PC1 = { 57, 49, 41, 33, 25,
            17, 9, 1, 58, 50, 42, 34, 26,
            18, 10, 2, 59, 51, 43, 35, 27,
            19, 11, 3, 60, 52, 44, 36, 63,
            55, 47, 39, 31, 23, 15, 7, 62,
            54, 46, 38, 30, 22, 14, 6, 61,
            53, 45, 37, 29, 21, 13, 5, 28,
            20, 12, 4 };

// second key-Permutation Table
int[] PC2 = { 14, 17, 11, 24, 1, 5, 3,
            28, 15, 6, 21, 10, 23, 19, 12,
            4, 26, 8, 16, 7, 27, 20, 13, 2,
            41, 52, 31, 37, 47, 55, 30, 40,
            51, 45, 33, 48, 44, 49, 39, 56,
            34, 53, 46, 42, 50, 36, 29, 32 };

// Expansion D-box Table
int[] EP = { 32, 1, 2, 3, 4, 5, 4,
           5, 6, 7, 8, 9, 8, 9, 10,
           11, 12, 13, 12, 13, 14, 15,
           16, 17, 16, 17, 18, 19, 20,
           21, 20, 21, 22, 23, 24, 25,
           24, 25, 26, 27, 28, 29, 28,
           29, 30, 31, 32, 1 };

// Straight Permutation Table
int[] P = { 16, 7, 20, 21, 29, 12, 28,
          17, 1, 15, 23, 26, 5, 18,
          31, 10, 2, 8, 24, 14, 32,
          27, 3, 9, 19, 13, 30, 6,
          22, 11, 4, 25 };

// S-box Table
int[][][] sbox = {
    { { 14, 4, 13, 1, 2, 15, 11, 8, 3, 10, 6, 12, 5, 9, 0, 7 },
      { 0, 15, 7, 4, 14, 2, 13, 1, 10, 6, 12, 11, 9, 5, 3, 8 },
      { 4, 1, 14, 8, 13, 6, 2, 11, 15, 12, 9, 7, 3, 10, 5, 0 },
      { 15, 12, 8, 2, 4, 9, 1, 7, 5, 11, 3, 14, 10, 0, 6, 13 } },

    { { 15, 1, 8, 14, 6, 11, 3, 4, 9, 7, 2, 13, 12, 0, 5, 10 },
      { 3, 13, 4, 7, 15, 2, 8, 14, 12, 0, 1, 10, 6, 9, 11, 5 },
      { 0, 14, 7, 11, 10, 4, 13, 1, 5, 8, 12, 6, 9, 3, 2, 15 },
      { 13, 8, 10, 1, 3, 15, 4, 2, 11, 6, 7, 12, 0, 5, 14, 9 } },
    { { 10, 0, 9, 14, 6, 3, 15, 5, 1, 13, 12, 7, 11, 4, 2, 8 },
      { 13, 7, 0, 9, 3, 4, 6, 10, 2, 8, 5, 14, 12, 11, 15, 1 },
      { 13, 6, 4, 9, 8, 15, 3, 0, 11, 1, 2, 12, 5, 10, 14, 7 },
      { 1, 10, 13, 0, 6, 9, 8, 7, 4, 15, 14, 3, 11, 5, 2, 12 } },
    { { 7, 13, 14, 3, 0, 6, 9, 10, 1, 2, 8, 5, 11, 12, 4, 15 },
      { 13, 8, 11, 5, 6, 15, 0, 3, 4, 7, 2, 12, 1, 10, 14, 9 },
      { 10, 6, 9, 0, 12, 11, 7, 13, 15, 1, 3, 14, 5, 2, 8, 4 },
      { 3, 15, 0, 6, 10, 1, 13, 8, 9, 4, 5, 11, 12, 7, 2, 14 } },
    { { 2, 12, 4, 1, 7, 10, 11, 6, 8, 5, 3, 15, 13, 0, 14, 9 },
      { 14, 11, 2, 12, 4, 7, 13, 1, 5, 0, 15, 10, 3, 9, 8, 6 },
      { 4, 2, 1, 11, 10, 13, 7, 8, 15, 9, 12, 5, 6, 3, 0, 14 },
      { 11, 8, 12, 7, 1, 14, 2, 13, 6, 15, 0, 9, 10, 4, 5, 3 } },
    { { 12, 1, 10, 15, 9, 2, 6, 8, 0, 13, 3, 4, 14, 7, 5, 11 },
      { 10, 15, 4, 2, 7, 12, 9, 5, 6, 1, 13, 14, 0, 11, 3, 8 },
      { 9, 14, 15, 5, 2, 8, 12, 3, 7, 0, 4, 10, 1, 13, 11, 6 },
      { 4, 3, 2, 12, 9, 5, 15, 10, 11, 14, 1, 7, 6, 0, 8, 13 } },
    { { 4, 11, 2, 14, 15, 0, 8, 13, 3, 12, 9, 7, 5, 10, 6, 1 },
      { 13, 0, 11, 7, 4, 9, 1, 10, 14, 3, 5, 12, 2, 15, 8, 6 },
      { 1, 4, 11, 13, 12, 3, 7, 14, 10, 15, 6, 8, 0, 5, 9, 2 },
      { 6, 11, 13, 8, 1, 4, 10, 7, 9, 5, 0, 15, 14, 2, 3, 12 } },
    { { 13, 2, 8, 4, 6, 15, 11, 1, 10, 9, 3, 14, 5, 0, 12, 7 },
      { 1, 15, 13, 8, 10, 3, 7, 4, 12, 5, 6, 11, 0, 14, 9, 2 },
      { 7, 11, 4, 1, 9, 12, 14, 2, 0, 6, 10, 13, 15, 3, 5, 8 },
      { 2, 1, 14, 7, 4, 10, 8, 13, 15, 12, 9, 0, 3, 5, 6, 11 } }
};
int[] shiftBits = { 1, 1, 2, 2, 2, 2, 2, 2,
                    1, 2, 2, 2, 2, 2, 2, 1 };

// hexadecimal to binary conversion
String hextoBin(String input)
{
    int n = input.length() * 4;
    input = Long.toBinaryString(
        Long.parseUnsignedLong(input, 16));
    while (input.length() < n)
        input = "0" + input;
    return input;
}

// binary to hexadecimal conversion
String binToHex(String input)
{
    int n = (int)input.length() / 4;
    input = Long.toHexString(
        Long.parseUnsignedLong(input, 2));
    while (input.length() < n)
        input = "0" + input;
    return input;
}
  // per-mutate input hexadecimal
// according to specified sequence
String permutation(int[] sequence, String input)
{
    String output = "";
    input = hextoBin(input);
    for (int i = 0; i < sequence.length; i++)
        output += input.charAt(sequence[i] - 1);
    output = binToHex(output);
    return output;
}
  // xor 2 hexadecimal strings
String xor(String a, String b)
{
    // hexadecimal to decimal(base 10)
    long t_a = Long.parseUnsignedLong(a, 16);
    // hexadecimal to decimal(base 10)
    long t_b = Long.parseUnsignedLong(b, 16);
    // xor
    t_a = t_a ^ t_b;
    // decimal to hexadecimal
    a = Long.toHexString(t_a);
    // prepend 0's to maintain length
    while (a.length() < b.length())
        a = "0" + a;
    return a;
}

// left Circular Shifting bits
String leftCircularShift(String input, int numBits)
{
    int n = input.length() * 4;
    int perm[] = new int[n];
    for (int i = 0; i < n - 1; i++)
        perm[i] = (i + 2);
    perm[n - 1] = 1;
    while (numBits-- > 0)
        input = permutation(perm, input);
    return input;
}
```

```
// preparing 16 keys for 16 rounds
String[] getKeys(String key)
{
   String keys[] = new String[16];
   // first key permutation
   key = permutation(PC1, key);
   for (int i = 0; i < 16; i++) {
      key = leftCircularShift(
              key.substring(0, 7), shiftBits[i])
           + leftCircularShift(key.substring(7, 14),
                          shiftBits[i]);
      // second key permutation
      keys[i] = permutation(PC2, key);
   }
   return keys;
}
 // s-box lookup
String sBox(String input)
{
   String output = "";
   input = hextoBin(input);
   for (int i = 0; i < 48; i += 6) {
      String temp = input.substring(i, i + 6);
      int num = i / 6;
      int row = Integer.parseInt(
         temp.charAt(0) + "" + temp.charAt(5), 2);
      int col = Integer.parseInt(
         temp.substring(1, 5), 2);
      output += Integer.toHexString(
         sbox[num][row][col]);
   }
   return output;
}

String round(String input, String key, int num)
{
   // fk
   String left = input.substring(0, 8);
   String temp = input.substring(8, 16);
   String right = temp;
   // Expansion permutation
   temp = permutation(EP, temp);
   // xor temp and round key
   temp = xor(temp, key);
   // lookup in s-box table
```

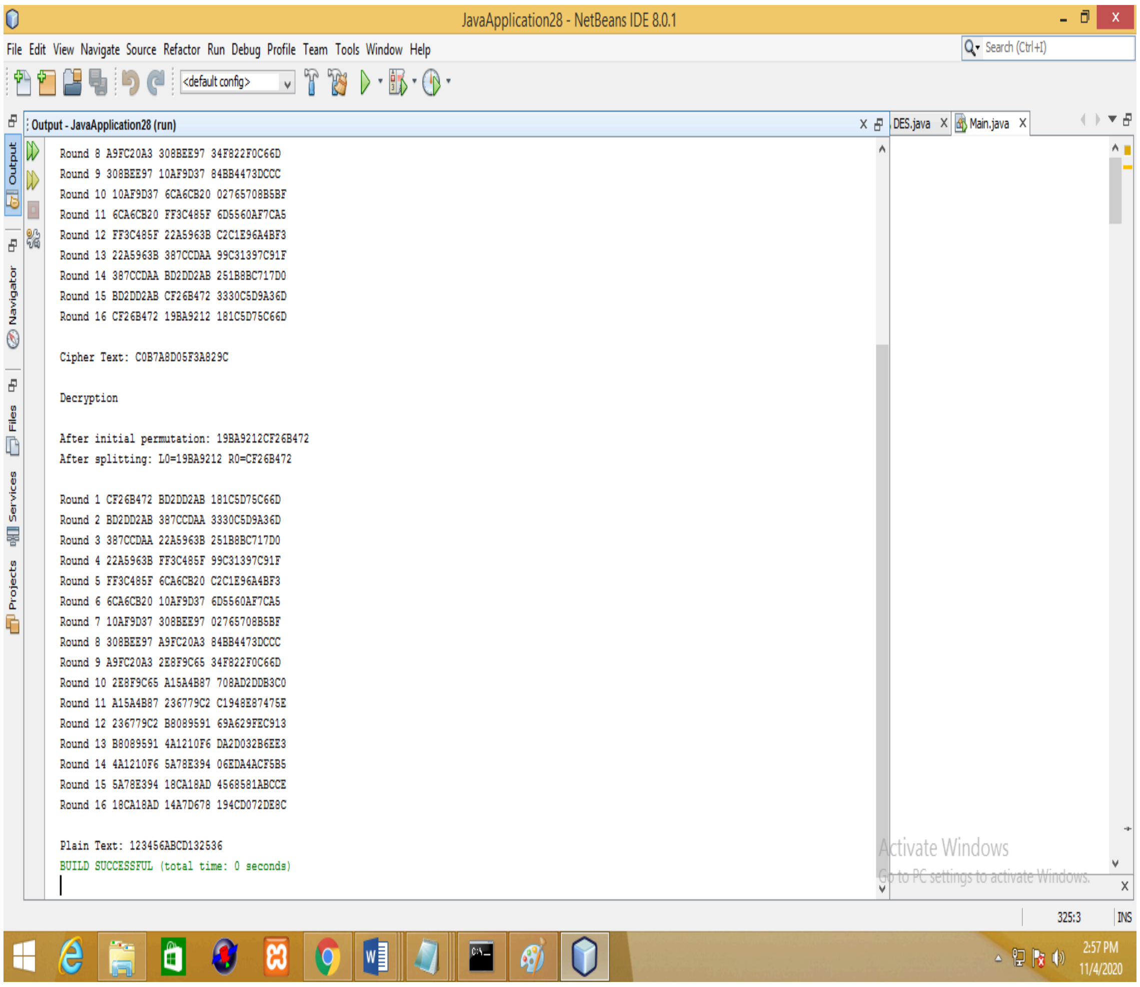
```
    temp = sBox(temp);
    // Straight D-box
    temp = permutation(P, temp);
    // xor
    left = xor(left, temp);
    System.out.println("Round "
                    + (num + 1) + " "
                    + right.toUpperCase()
                    + " " + left.toUpperCase() + " "
                    + key.toUpperCase());

    // swapper
    return right + left;
}
String encrypt(String plainText, String key)
{
    int i;
    // get round keys
    String keys[] = getKeys(key);

    // initial permutation
    plainText = permutation(IP, plainText);
    System.out.println(
        "After initial permutation: "
        + plainText.toUpperCase());
    System.out.println(
        "After splitting: L0="
        + plainText.substring(0, 8).toUpperCase()
        + " R0="
        + plainText.substring(8, 16).toUpperCase() + "\n");

    // 16 rounds
    for (i = 0; i < 16; i++) {
        plainText = round(plainText, keys[i], i);
    }

    // 32-bit swap
    plainText = plainText.substring(8, 16)
            + plainText.substring(0, 8);

    // final permutation
    plainText = permutation(IP1, plainText);
    return plainText;
}
```

```
    String decrypt(String plainText, String key)
    {
        int i;
        // get round keys
        String keys[] = getKeys(key);

        // initial permutation
        plainText = permutation(IP, plainText);
        System.out.println(
            "After initial permutation: "
            + plainText.toUpperCase());
        System.out.println(
            "After splitting: L0="
            + plainText.substring(0, 8).toUpperCase()
            + " R0=" + plainText.substring(8, 16).toUpperCase()
            + "\n");

        // 16-rounds
        for (i = 15; i > -1; i--) {
            plainText = round(plainText, keys[i], 15 - i);
        }

        // 32-bit swap
        plainText = plainText.substring(8, 16)
                    + plainText.substring(0, 8);
        plainText = permutation(IP1, plainText);
        return plainText;
    }
}
public static void main(String args[])
{
    String text = "123456ABCD132536";
    String key = "AABB09182736CCDD";

    DES cipher = new DES();
    System.out.println("Encryption:\n");

    text = cipher.encrypt(text, key);

    System.out.println(
        "\nCipher Text: " + text.toUpperCase() + "\n");
    System.out.println("Decryption\n");
    text = cipher.decrypt(text, key);
    System.out.println(
        "\nPlain Text: "
        + text.toUpperCase());     } }

                catch(Exception e)
                {       System.out.println(e);          }
        }
        void generateSymmetricKey()
        {       try  {  Random r = new Random();
                        int num = r.nextInt(10000);
                        String knum = String.valueOf(num);
                        byte[] knumb = knum.getBytes();
                        skey=getRawKey(knumb);
                        skeyString = new String(skey);
                        System.out.println("DES Symmetric key = "+skeyString);
                }
                catch(Exception e)
                {       System.out.println(e);                  }
        }
        private static byte[] getRawKey(byte[] seed) throws Exception
        {       KeyGenerator kgen = KeyGenerator.getInstance("DES");
                SecureRandom sr = SecureRandom.getInstance("SHA1PRNG");
                sr.setSeed(seed);
                kgen.init(56, sr);
                SecretKey skey = kgen.generateKey();
                raw = skey.getEncoded();
                return raw;
        }
        private static byte[] encrypt(byte[] raw, byte[] clear) throws Exception
        {       SecretKeySpec skeySpec = new SecretKeySpec(raw, "DES");
                Cipher cipher = Cipher.getInstance("DES");
                cipher.init(Cipher.ENCRYPT_MODE, skeySpec);
                byte[] encrypted = cipher.doFinal(clear);
                return encrypted;
        }
        private static byte[] decrypt(byte[] raw, byte[] encrypted) throws Exception
        {       SecretKeySpec skeySpec = new SecretKeySpec(raw, "DES");
                Cipher cipher = Cipher.getInstance("DES");
                cipher.init(Cipher.DECRYPT_MODE, skeySpec);
                byte[] decrypted = cipher.doFinal(encrypted);
                return decrypted;
        }
        public static void main(String args[])
        {       DES des = new DES();        }
}
```

OUTPUT


JavaApplication28 - NetBeans IDE 8.0.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
Output - JavaApplication28 (run)
DES.java
Main.java
Output
Navigator
Files
Services
Projects
Activate Windows
Go to PC settings to activate Windows.
325:3
INS
2:53 PM
11/4/2020

```
run:
Encryption:

After initial permutation: 14A7D67818CA18AD
After splitting: L0=14A7D678 R0=18CA18AD

Round 1 18CA18AD 5A78E394 194CD072DE8C
Round 2 5A78E394 4A1210F6 4568581ABCCE
Round 3 4A1210F6 B8089591 06EDA4ACF5B5
Round 4 B8089591 236779C2 DA2D032B6EE3
Round 5 236779C2 A15A4B87 69A629FEC913
Round 6 A15A4B87 2E8F9C65 C1948E87475E
Round 7 2E8F9C65 A9FC20A3 708AD2DDB3C0
Round 8 A9FC20A3 308BEE97 34F822F0C66D
Round 9 308BEE97 10AF9D37 84BB4473DCCC
Round 10 10AF9D37 6CA6CB20 02765708B5BF
Round 11 6CA6CB20 FF3C485F 6D5560AF7CA5
Round 12 FF3C485F 22A5963B C2C1E96A4BF3
Round 13 22A5963B 387CCDAA 99C31397C91F
Round 14 387CCDAA BD2DD2AB 251B8BC717D0
Round 15 BD2DD2AB CF26B472 3330C5D9A36D
Round 16 CF26B472 19BA9212 181C5D75C66D

Cipher Text: C0B7A8D05F3A829C

Decryption

After initial permutation: 19BA9212CF26B472
After splitting: L0=19BA9212 R0=CF26B472

Round 1 CF26B472 BD2DD2AB 181C5D75C66D
Round 2 BD2DD2AB 387CCDAA 3330C5D9A36D
Round 3 387CCDAA 22A5963B 251B8BC717D0
Round 4 22A5963B FF3C485F 99C31397C91F
Round 5 FF3C485F 6CA6CB20 C2C1E96A4BF3
Round 6 6CA6CB20 10AF9D37 6D5560AF7CA5
Round 7 10AF9D37 308BEE97 02765708B5BF
```



```
Round 8 A9FC20A3 308BEE97 34F822F0C66D
Round 9 308BEE97 10AF9D37 84BB4473DCCC
Round 10 10AF9D37 6CA6CB20 02765708B5BF
Round 11 6CA6CB20 FF3C485F 6D5560AF7CA5
Round 12 FF3C485F 22A5963B C2C1E96A4BF3
Round 13 22A5963B 387CCDAA 99C31397C91F
Round 14 387CCDAA BD2DD2AB 251B8BC717D0
Round 15 BD2DD2AB CF26B472 3330C5D9A36D
Round 16 CF26B472 19BA9212 181C5D75C66D

Cipher Text: C0B7A8D05F3A829C

Decryption

After initial permutation: 19BA9212CF26B472
After splitting: L0=19BA9212 R0=CF26B472

Round 1 CF26B472 BD2DD2AB 181C5D75C66D
Round 2 BD2DD2AB 387CCDAA 3330C5D9A36D
Round 3 387CCDAA 22A5963B 251B8BC717D0
Round 4 22A5963B FF3C485F 99C31397C91F
Round 5 FF3C485F 6CA6CB20 C2C1E96A4BF3
Round 6 6CA6CB20 10AF9D37 6D5560AF7CA5
Round 7 10AF9D37 308BEE97 02765708B5BF
Round 8 308BEE97 A9FC20A3 84BB4473DCCC
Round 9 A9FC20A3 2E8F9C65 34F822F0C66D
Round 10 2E8F9C65 A15A4B87 708AD2DDB3C0
Round 11 A15A4B87 236779C2 C1948E87475E
Round 12 236779C2 B8089591 69A629FEC913
Round 13 B8089591 4A1210F6 DA2D032B6EE3
Round 14 4A1210F6 5A78E394 06EDA4ACF5B5
Round 15 5A78E394 18CA18AD 4568581ABCCE
Round 16 18CA18AD 14A7D678 194CD072DE8C

Plain Text: 123456ABCD132536
BUILD SUCCESSFUL (total time: 0 seconds)
```

RESULT

Thus java program to implement DES was written, executed and output is verified successfully.

Ex.No:4 **IMPLEMENTATION OF AES**

AIM

To implement AES using java.

ALGORITHM

Step 1: Start the program.
Step 2: Define a class AES, in that assign values to plaintext, IV and key variables.
Step 3: Define main() to display the encrypted and decrypted text of plain text.
Step 4: Define encrypt() to generated cipher text and decrypt() to generate plain text.
Step 5: Stop the program.

PROGRAM

```
package javaapplication1;
import java.security.MessageDigest;
import java.util.Arrays;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;
import javax.crypto.spec.IvParameterSpec;

import javax.crypto.Cipher;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.SecretKeySpec;

public class AES
{
        static String IV = "AAAAAAAAAAAAAAAA";
        static String plaintext = "test text 123\0\0\0"; /*Note null padding*/
        static String encryptionKey = "0123456789abcdef";
        public static void main(String [] args)
        {
                try
                {
                        System.out.println("==Java==");
                        System.out.println("plain:   " + plaintext);
                        byte[] cipher = encrypt(plaintext, encryptionKey);
                        System.out.print("cipher:  ");
                        for (int i=0; i<cipher.length; i++)
                                System.out.print(new Integer(cipher[i])+" ");
                        System.out.println("");
                        String decrypted = decrypt(cipher, encryptionKey);
                        System.out.println("decrypt: " + decrypted);
                }
                catch (Exception e)
                {
                        e.printStackTrace();
                }
        }
        public static byte[] encrypt(String plainText, String encryptionKey) throws Exception
        {
                Cipher cipher = Cipher.getInstance("AES/CBC/NoPadding", "SunJCE");
                SecretKeySpec key = new SecretKeySpec(encryptionKey.getBytes("UTF-8"),
                "AES");
                cipher.init(Cipher.ENCRYPT_MODE,                          key,new
                IvParameterSpec(IV.getBytes("UTF-8")));
                return cipher.doFinal(plainText.getBytes("UTF-8"));
        }
        public static String decrypt(byte[] cipherText, String encryptionKey) throws Exception
        {
                Cipher cipher = Cipher.getInstance("AES/CBC/NoPadding", "SunJCE");
                SecretKeySpec key = new SecretKeySpec(encryptionKey.getBytes("UTF-8"),
                "AES");
                cipher.init(Cipher.DECRYPT_MODE,                          key,new
                IvParameterSpec(IV.getBytes("UTF-8")));
                return new String(cipher.doFinal(cipherText),"UTF-8");
        }
}
```

OUTPUT

```
Enter plaintext:
onetimepadalogrihtmsneedlongkey
Enter key:
longkeyisrequiredforonetimepad
Encrpted text is:zbrzsqcxsuebioimky jbriwt rvkhj
Decrypted text is:onetimepadalogrihtmsneedlongkey
```

RESULT

Thus AES was implemented using java and output is verified successfully.

Ex.No: 5 IMPLEMENTATION OF RSA

AIM

To write a Java Program for implementing RSA.

Algorithm

Step 1 : Choose two prime numbers p and q.

Step 2 : Calculate $n = p*q$

Step 3 : Calculate $\phi(n) = (p - 1) * (q - 1)$

Step 4 : Choose e such that $gcd(e , \phi(n)) = 1$

Step 5 : Calculate d such that $e*d \bmod \phi(n) = 1$

Step 6 : Public Key {e,n} Private Key {d,n}

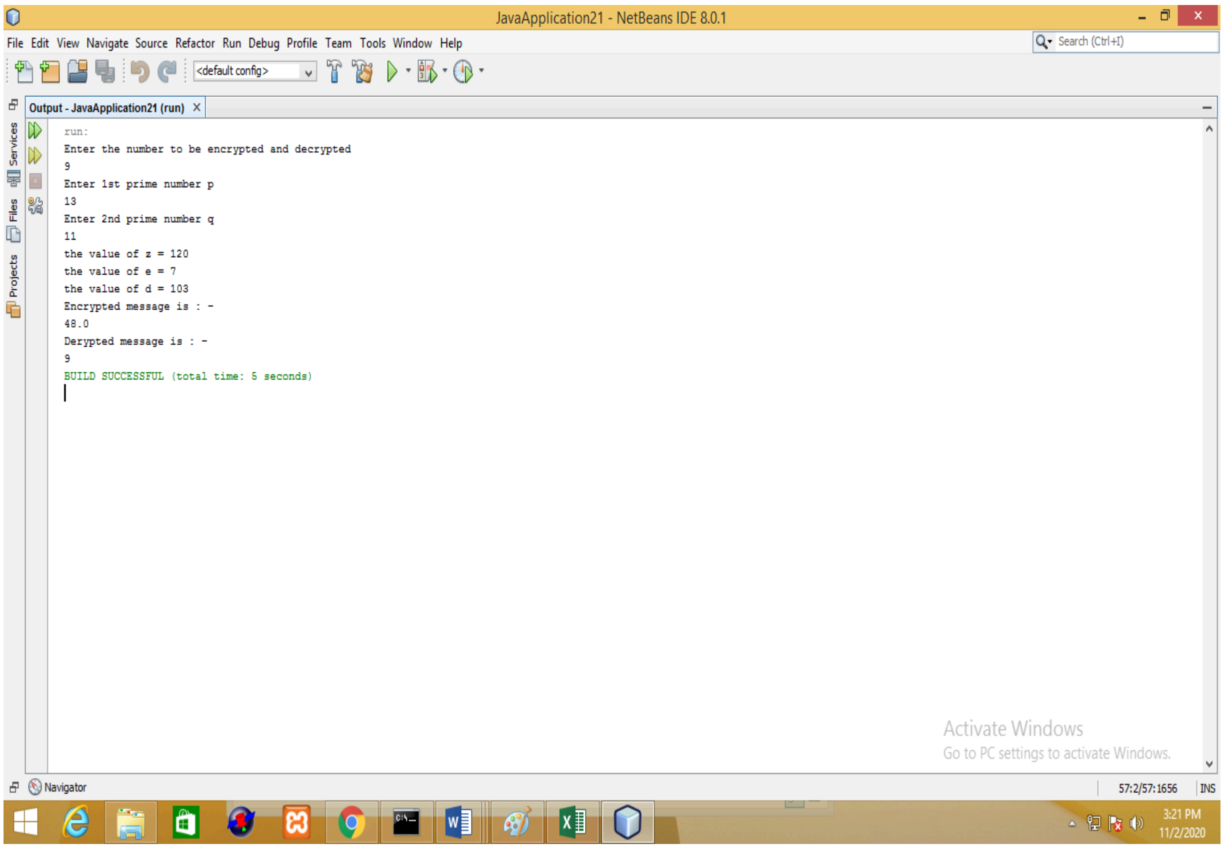
Step 7 : Cipher text $C = P^e \bmod n$ where P = plaintext

Step 8 : For Decryption $D = D^d \bmod n$ where D will give back the plaintext.

PROGRAM

```
import java.util.*;
import java.math.*;
class RSA
{
   public static void main(String args[])
   {
      Scanner sc=new Scanner(System.in);
      int p,q,n,z,d=0,e,i;
      System.out.println("Enter the number to be encrypted and decrypted");
      int msg=sc.nextInt();
      double c;
      BigInteger msgback;
      System.out.println("Enter 1st prime number p");
      p=sc.nextInt();
      System.out.println("Enter 2nd prime number q");
      q=sc.nextInt();
      n=p*q;
      z=(p-1)*(q-1);
      System.out.println("the value of z = "+z);

      for(e=2;e<z;e++)
      {
         if(gcd(e,z)==1)           // e is for public key exponent
         {
            break;
            }
        }
        System.out.println("the value of e = "+e);
        for(i=0;i<=9;i++)
        {
            int x=1+(i*z);
            if(x%e==0)      //d is for private key exponent
            {
                d=x/e;
                break;
            }
        }
        System.out.println("the value of d = "+d);
        c=(Math.pow(msg,e))%n;
        System.out.println("Encrypted message is : -");
        System.out.println(c);
                    //converting int value of n to BigInteger
        BigInteger N = BigInteger.valueOf(n);
//converting float value of c to BigInteger
        BigInteger C = BigDecimal.valueOf(c).toBigInteger();
        msgback = (C.pow(d)).mod(N);
        System.out.println("Derypted message is : -");
        System.out.println(msgback);
    }
    static int gcd(int e, int z)
    {
        if(e==0)
        return z;
        else
        return gcd(z%e,e);
    }
}
```

OUTPUT



```
run:
Enter the number to be encrypted and decrypted
9
Enter 1st prime number p
13
Enter 2nd prime number q
11
the value of z = 120
the value of e = 7
the value of d = 103
Encrypted message is : -
48.0
Derypted message is : -
9
BUILD SUCCESSFUL (total time: 5 seconds)
```

RESULT

Thus java program to implement RSA was written, executed and output is verified successfully.

Ex.No: 6 IMPLEMENTATION OF DIFFIEE - HELLMAN

AIM

To write a Java Program for implementing Diffiee - Hellman.

ALGORITHM

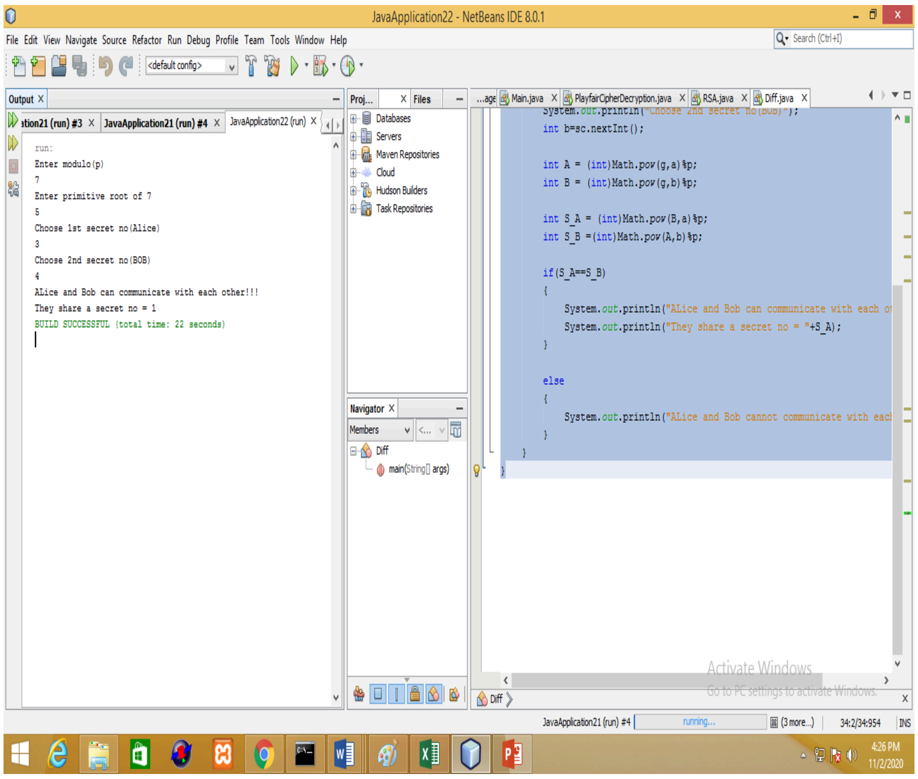
Step 1 : Choose two prime numbers **g(primitive root of p)** and **p.**
Step 2 : Alice selects a secret no(**a**) and computes **g^a** *mod* **p** , let's call it **A**. Alice sends **A** to Bob.
Step 3 : Bob selects a secret no(**b**) and computes **g^b** *mod* **p,** let's call it **B.** Bob sends **B** to Alice.
Step 4 : Alice computes ***S_A*** = $B^a \bmod p$
Step 5 : Bob computes ***S_B*** = $A^b \bmod p$
Step 6 : If **S_A=S_B** then Alice and Bob can agree for future communication.

PROGRAM

```
import java.util.*;

class Diff
{
   public static void main(String args[])
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter modulo(p)");
      int p=sc.nextInt();
      System.out.println("Enter primitive root of "+p);
      int g=sc.nextInt();
      System.out.println("Choose 1st secret no(Alice)");
      int a=sc.nextInt();
      System.out.println("Choose 2nd secret no(BOB)");
      int b=sc.nextInt();

      int A = (int)Math.pow(g,a)%p;
      int B = (int)Math.pow(g,b)%p;
      int S_A = (int)Math.pow(B,a)%p;
      int S_B =(int)Math.pow(A,b)%p;

      if(S_A==S_B)
      {
         System.out.println("ALice and Bob can communicate with each other!!!");
         System.out.println("They share a secret no = "+S_A);
      }
       else
      {
         System.out.println("ALice and Bob cannot communicate with each other!!!");
      }
   }
```

}

OUTPUT



```
run:
Enter modulo(p)
7
Enter primitive root of 7
5
Choose 1st secret no(Alice)
3
Choose 2nd secret no(BOB)
4
ALice and Bob can communicate with each other!!!
They share a secret no = 1
BUILD SUCCESSFUL (total time: 22 seconds)
```

```
        System.out.println("Choose 2nd secret no(BOB)");
        int b=sc.nextInt();

        int A = (int)Math.pow(g,a)%p;
        int B = (int)Math.pow(g,b)%p;

        int S_A = (int)Math.pow(B,a)%p;
        int S_B =(int)Math.pow(A,b)%p;

        if(S_A==S_B)
        {
            System.out.println("ALice and Bob can communicate with each o
            System.out.println("They share a secret no = "+S_A);
        }

        else
        {
            System.out.println("ALice and Bob cannot communicate with each
        }
    }
}
```

RESULT

Thus java program to implement Diffiee Hellman was written, executed and output is verified successfully.

Ex.No: 7 IMPLEMENTATION OF SHA-1

AIM

To write a Java Program for implementing SHA-1.

ALGORITHM

Step 1: Start the program.

Step 2: Define a class HashTextTest, in that main() call sha1() to display secured hash value.

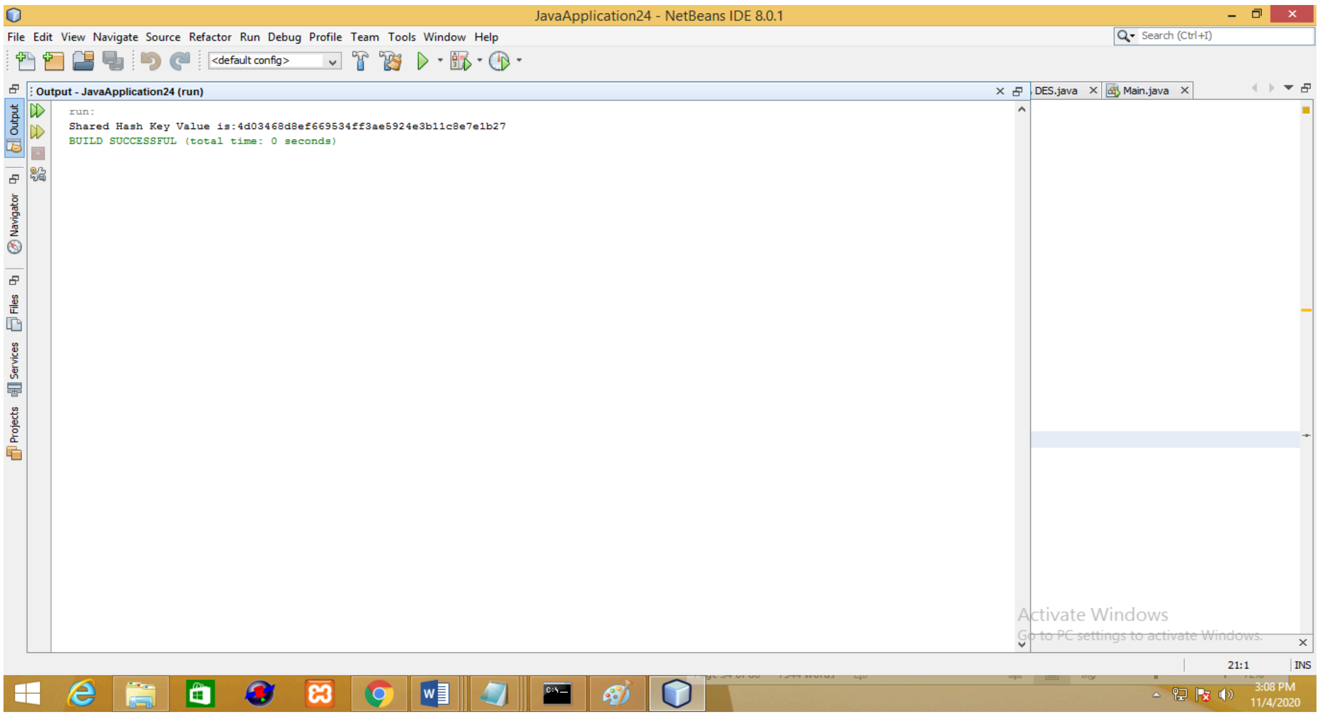
Step 3: Define sha1(), in that define the instances and generate the hash value.

Step 4: Stop the program.

PROGRAM

```
package javaapplication1;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
public class HashTextTest
{
        public static void main(String[] args) throws NoSuchAlgorithmException
        {
                System.out.println("Shared Hash Key Value is:"+sha1("shared Hashing"));
        }
        static String sha1(String input) throws NoSuchAlgorithmException
        {
                MessageDigest mDigest = MessageDigest.getInstance("SHA1");
                byte[] result = mDigest.digest(input.getBytes());
                StringBuffer sb = new StringBuffer();
                for (int i = 0; i < result.length; i++)
                {
                        sb.append(Integer.toString((result[i] & 0xff) + 0x100, 16).substring(1));
                }
                return sb.toString();
        }
}
```

OUTPUT


JavaApplication24 - NetBeans IDE 8.0.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Output - JavaApplication24 (run)
DES.java
Main.java

```
run:
Shared Hash Key Value is:4d03468d8ef669534ff3ae5924e3b11c8e7e1b27
BUILD SUCCESSFUL (total time: 0 seconds)
```

RESULT

Thus java program to implement SHA was written, executed and output is verified successfully.

Ex. No: 8 **IMPLEMENTATION OF SIGNATURE SCHEME DIGITAL SIGNATURE STANDARD**

AIM

To implement Signature Scheme of Digital Signature Standard using java.

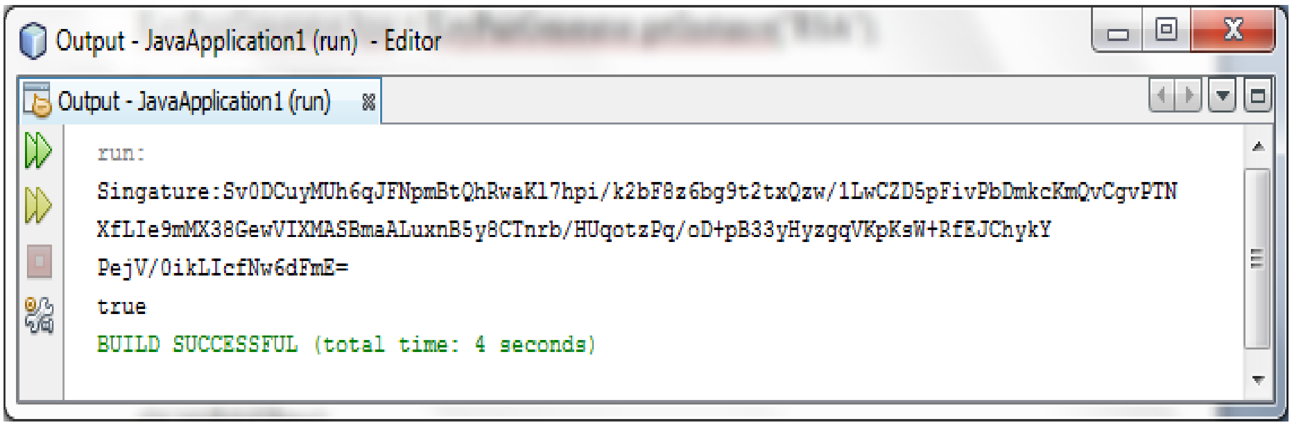
ALGORITHM

Step 1: Start the program.
Step 2: Define a class DSS, in that define a main().
Step 3: Create instance for KeypairGenerator class.
Step 4: Generate Key Pair using KeyPair Class.
Step 5: Send the data to encrypt and sign.
Step 6: Sign the data using Signature class.
Step 7: Display the signature and verification status.
Step 8: Stop the program.

PROGRAM

```
package javaapplication1;
import java.security.KeyPair;
import java.security.KeyPairGenerator;
import java.security.Signature;
import sun.misc.BASE64Encoder;
public class DSS
{
public static void main(String[] args) throws Exception
	{	KeyPairGenerator kpg = KeyPairGenerator.getInstance("RSA");
		kpg.initialize(1024);
		KeyPair keyPair = kpg.genKeyPair();
		byte[] data = "test".getBytes("UTF8");
		Signature sig = Signature.getInstance("MD5WithRSA");
		sig.initSign(keyPair.getPrivate());
		sig.update(data);
		byte[] signatureBytes = sig.sign();
		System.out.println("Singature:" + new BASE64Encoder().encode(signatureBytes));
		sig.initVerify(keyPair.getPublic());
		sig.update(data);
		System.out.println(sig.verify(signatureBytes));
	}
}
```

OUTPUT


Output - JavaApplication1 (run) - Editor
Output - JavaApplication1 (run)

```
run:
Singature:Sv0DCuyMUh6qJFNpmBtQhRwaKl7hpi/k2bF8z6bg9t2txQzw/1LwCZD5pFivPbDmkcKmQvCgvPTN
XfLIe9mMX38GewVIXMASBmaALuxnB5y8CTnrb/HUqotzPq/oD+pB33yHyzgqVKpKsW+RfEJChykY
PejV/0ikLIcfNw6dFmE=
true
BUILD SUCCESSFUL (total time: 4 seconds)
```

RESULT

Thus java program to implement Signature Scheme of Digital Signature Standard was written, executed and output is verified successfully.

Ex.No: 9 DEMONSTRATE INTRUSION DETECTION SYSTEM (IDS) USING SNORT

AIM

To demonstrate intrusion detection system (ids) using snort.

PROCEDURE

SNORT can be configured to run in three modes:

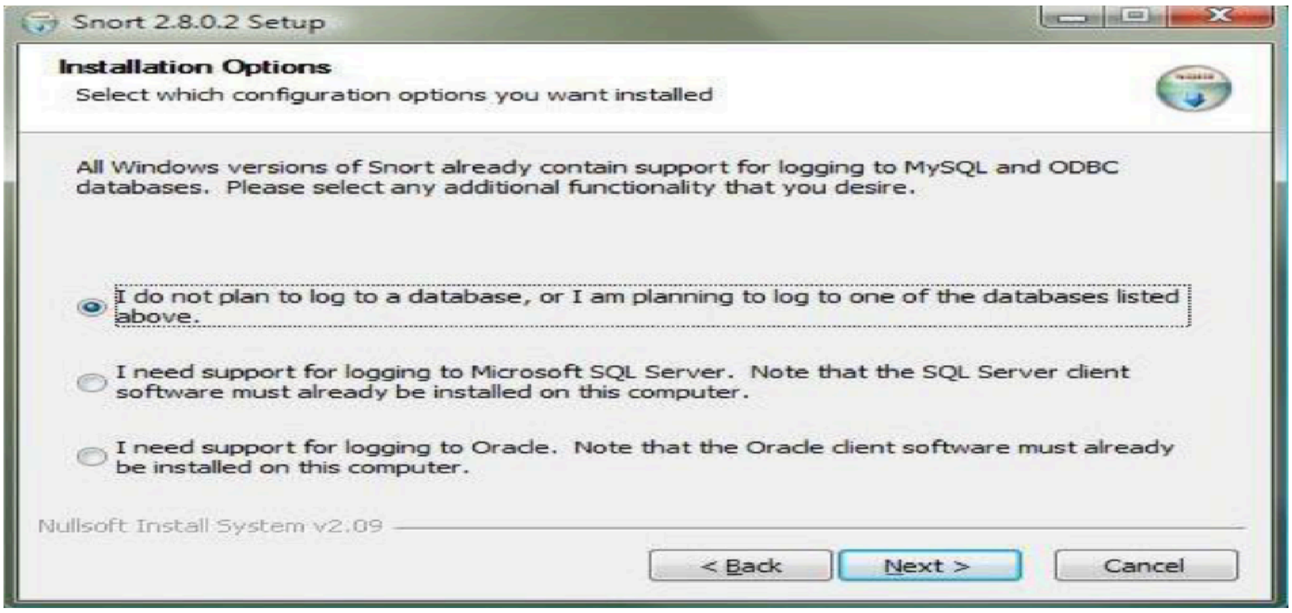
1. Sniffer mode
2. Packet Logger mode
3. Network Intrusion Detection System mode

Sniffer modesnort –v Print out the TCP/IP packets header on the screen
Snort –vd show the TCP/IP ICMP header with application data in transit.

Packet Logger modesnort –dev –l c:\log [create this directory in the C drive] and snort will automatically know to go into packet logger mode, it collects every packet it sees and places it in log directory. snort –dev –l c:\log –h ipaddress/24 This rule tells snort that you want to print out the data link and TCP/IP headers as well as application data into the log directory. snort –l c:\log –b This is binary mode logs everything into a single file.

Network Intrusion Detection System modesnort –d c:\log –h ipaddress/24 –c snort.conf - This is a configuration file applies rule to each packet to decide it an action based upon the rule type in the file. Snort –d –h ipaddress/24 –l c:\log –c snort.conf - This will configure snort to run in its most basic NIDS form, logging packets that trigger rules specifies in the snort.conf.

Step 1: Download SNORT from snort.org
Step 2: Install snort with or without database support.


Snort 2.8.0.2 Setup

Installation Options
Select which configuration options you want installed

All Windows versions of Snort already contain support for logging to MySQL and ODBC databases. Please select any additional functionality that you desire.

[✓ I do not plan to log to a database, or I am planning to log to one of the databases listed above.]

I need support for logging to Microsoft SQL Server. Note that the SQL Server client software must already be installed on this computer.

I need support for logging to Oracle. Note that the Oracle client software must already be installed on this computer.

Nullsoft Install System v2.09

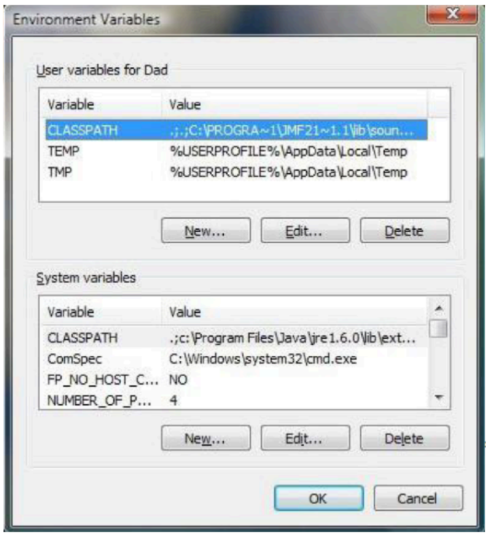
Step 3: Select all the components and Click Next.

Step 4: Install and Close.

Step 5: Skip the WinPcap driver installation

Step 6: Add the path variable in windows environment variable by selecting new classpath.

Step 7: Create a path variable and point it at snort.exe variable name ☾ path and variable value ☾ c:\snort\bin.



Environment Variables

User variables for Dad

Variable	Value
CLASSPATH	.;.;C:\PROGRA~1\JMF21~1.1\lib\soun...
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

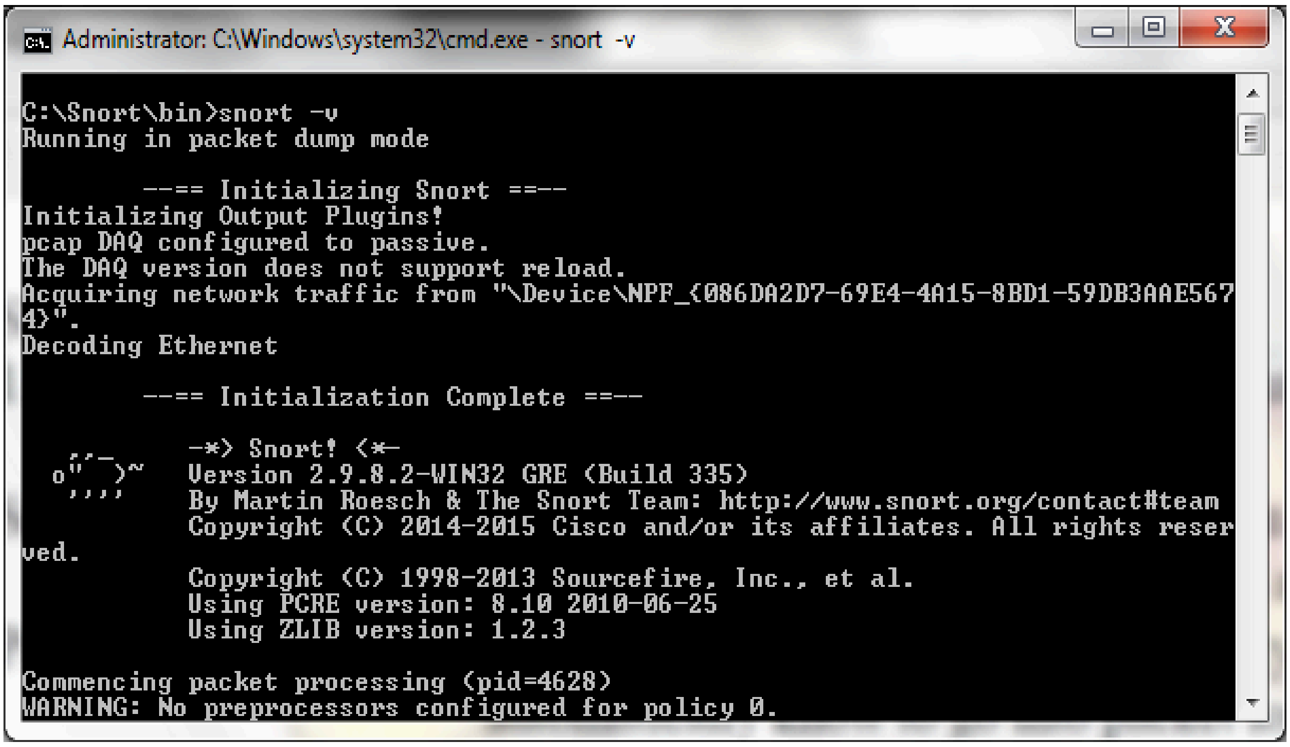
System variables

Variable	Value
CLASSPATH	.;c:\Program Files\Java\jre1.6.0\lib\ext...
ComSpec	C:\Windows\system32\cmd.exe
FP_NO_HOST_C...	NO
NUMBER_OF_P...	4

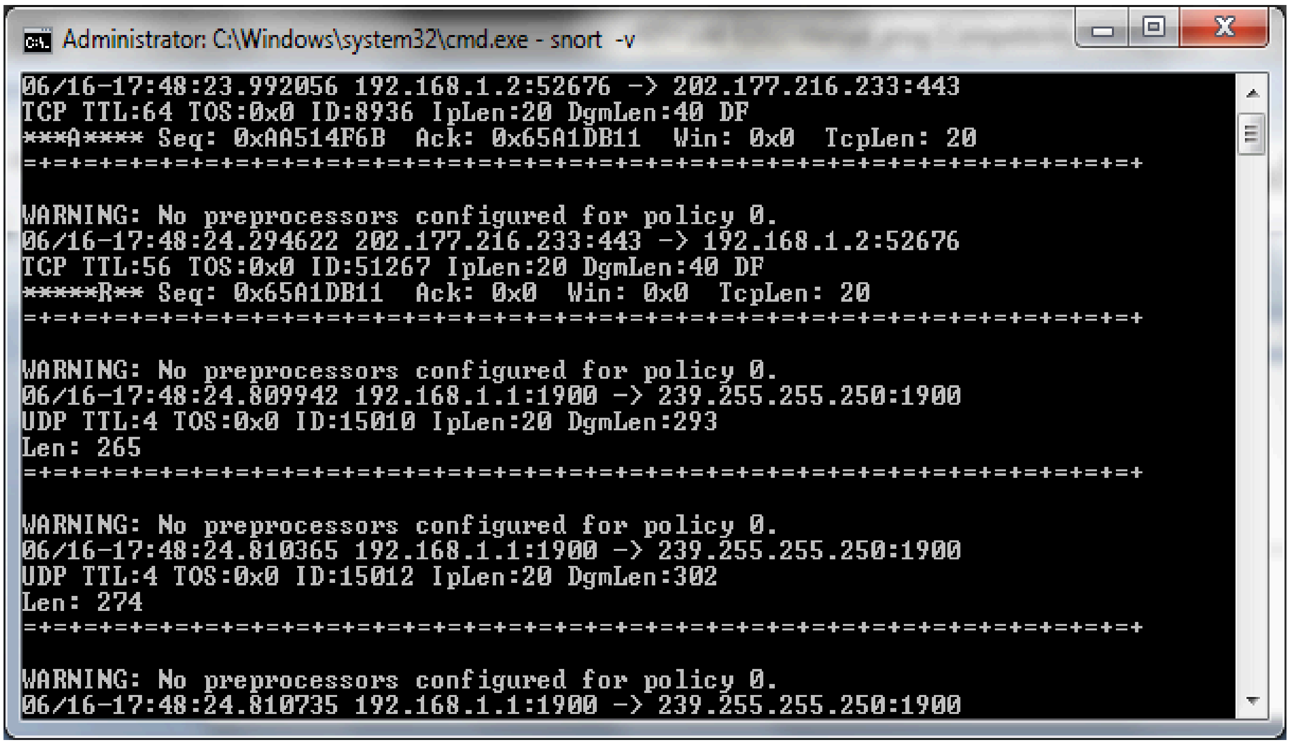
Step 8: Click OK button and then close all dialog boxes.

Step 9: Open command prompt and type the following commands:

C:\Snort\bin>Snort - v

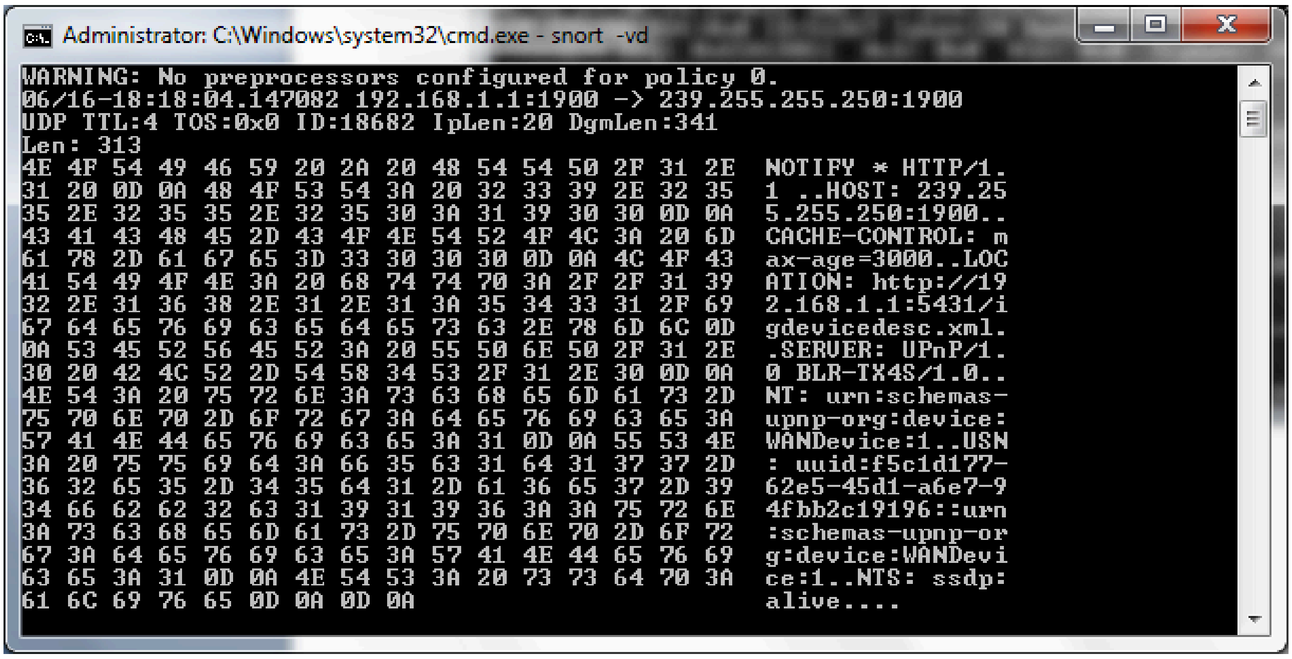


```
C:\Snort\bin>snort -v
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
The DAQ version does not support reload.
Acquiring network traffic from "\Device\NPF_{086DA2D7-69E4-4A15-8BD1-59DB3AAE567
4}".
Decoding Ethernet

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.8.2-WIN32 GRE (Build 335)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2015 Cisco and/or its affiliates. All rights reser
ved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using PCRE version: 8.10 2010-06-25
           Using ZLIB version: 1.2.3

Commencing packet processing (pid=4628)
WARNING: No preprocessors configured for policy 0.
```

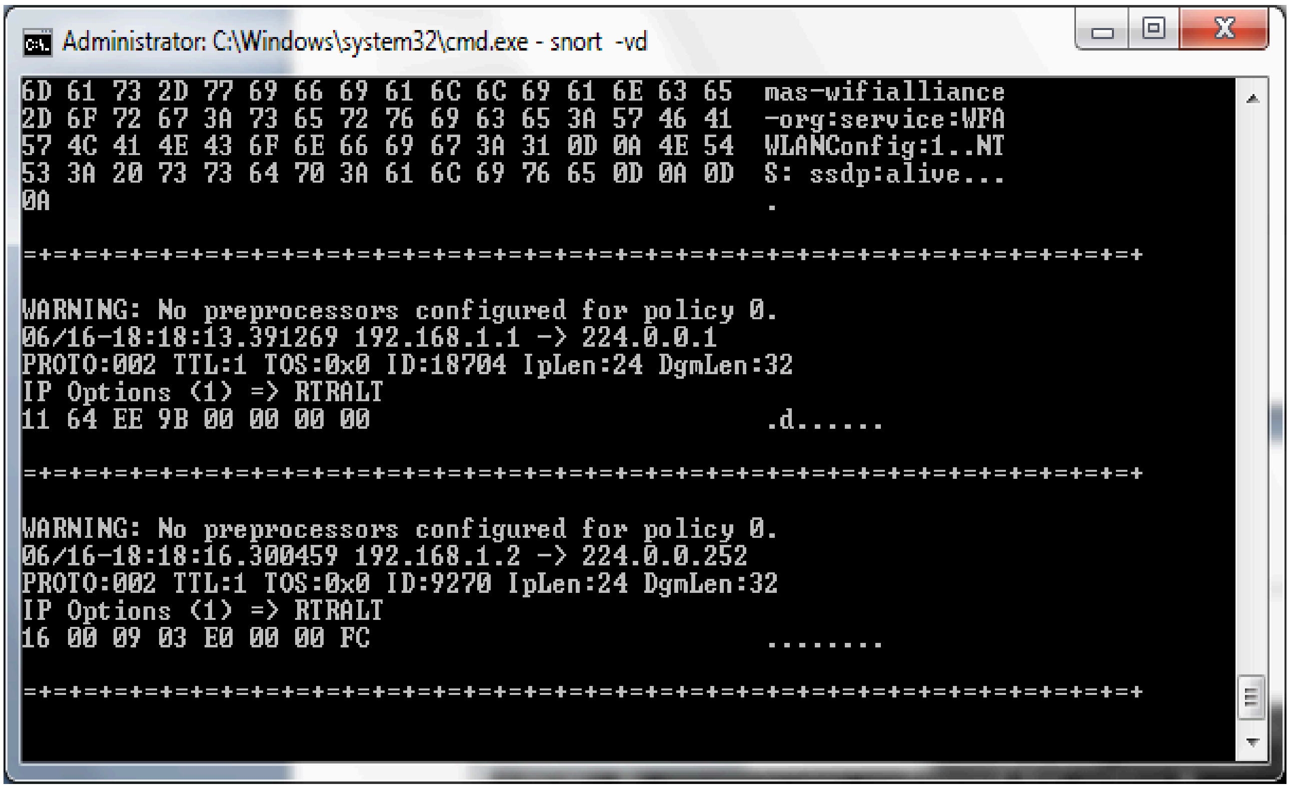


```
06/16-17:48:23.992056 192.168.1.2:52676 -> 202.177.216.233:443
TCP TTL:64 TOS:0x0 ID:8936 IpLen:20 DgmLen:40 DF
***A**** Seq: 0xAA514F6B  Ack: 0x65A1DB11  Win: 0x0  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
06/16-17:48:24.294622 202.177.216.233:443 -> 192.168.1.2:52676
TCP TTL:56 TOS:0x0 ID:51267 IpLen:20 DgmLen:40 DF
*****R** Seq: 0x65A1DB11  Ack: 0x0  Win: 0x0  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
06/16-17:48:24.809942 192.168.1.1:1900 -> 239.255.255.250:1900
UDP TTL:4 TOS:0x0 ID:15010 IpLen:20 DgmLen:293
Len: 265
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
06/16-17:48:24.810365 192.168.1.1:1900 -> 239.255.255.250:1900
UDP TTL:4 TOS:0x0 ID:15012 IpLen:20 DgmLen:302
Len: 274
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
06/16-17:48:24.810735 192.168.1.1:1900 -> 239.255.255.250:1900
```

C:\Snort\bin>Snort – vd



```
WARNING: No preprocessors configured for policy 0.
06/16-18:18:04.147082 192.168.1.1:1900 -> 239.255.255.250:1900
UDP TTL:4 TOS:0x0 ID:18682 IpLen:20 DgmLen:341
Len: 313
4E 4F 54 49 46 59 20 2A 20 48 54 54 50 2F 31 2E  NOTIFY * HTTP/1.
31 20 0D 0A 48 4F 53 54 3A 20 32 33 39 2E 32 35  1 ..HOST: 239.25
35 2E 32 35 35 2E 32 35 30 3A 31 39 30 30 0D 0A  5.255.250:1900..
43 41 43 48 45 2D 43 4F 4E 54 52 4F 4C 3A 20 6D  CACHE-CONTROL: m
61 78 2D 61 67 65 3D 33 30 30 30 0D 0A 4C 4F 43  ax-age=3000..LOC
41 54 49 4F 4E 3A 20 68 74 74 70 3A 2F 2F 31 39  ATION: http://19
32 2E 31 36 38 2E 31 2E 31 3A 35 34 33 31 2F 69  2.168.1.1:5431/i
67 64 65 76 69 63 65 64 65 73 63 2E 78 6D 6C 0D  gdevicedesc.xml.
0A 53 45 52 56 45 52 3A 20 55 50 6E 50 2F 31 2E  .SERVER: UPnP/1.
30 20 42 4C 52 2D 54 58 34 53 2F 31 2E 30 0D 0A  0 BLR-TX4S/1.0..
4E 54 3A 20 75 72 6E 3A 73 63 68 65 6D 61 73 2D  NT: urn:schemas-
75 70 6E 70 2D 6F 72 67 3A 64 65 76 69 63 65 3A  upnp-org:device:
57 41 4E 44 65 76 69 63 65 3A 31 0D 0A 55 53 4E  WANDevice:1..USN
3A 20 75 75 69 64 3A 66 35 63 31 64 31 37 37 2D  : uuid:f5c1d177-
36 32 65 35 2D 34 35 64 31 2D 61 36 65 37 2D 39  62e5-45d1-a6e7-9
34 66 62 62 32 63 31 39 31 39 36 3A 3A 75 72 6E  4fbb2c19196::urn
3A 73 63 68 65 6D 61 73 2D 75 70 6E 70 2D 6F 72  :schemas-upnp-or
67 3A 64 65 76 69 63 65 3A 57 41 4E 44 65 76 69  g:device:WANDevi
63 65 3A 31 0D 0A 4E 54 53 3A 20 73 73 64 70 3A  ce:1..NTS: ssdp:
61 6C 69 76 65 0D 0A 0D 0A                       alive....
```



```
6D 61 73 2D 77 69 66 69 61 6C 6C 69 61 6E 63 65  mas-wifialliance
2D 6F 72 67 3A 73 65 72 76 69 63 65 3A 57 46 41  -org:service:WFA
57 4C 41 4E 43 6F 6E 66 69 67 3A 31 0D 0A 4E 54  WLANConfig:1..NT
53 3A 20 73 73 64 70 3A 61 6C 69 76 65 0D 0A 0D  S: ssdp:alive...
0A                                               .

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
06/16-18:18:13.391269 192.168.1.1 -> 224.0.0.1
PROTO:002 TTL:1 TOS:0x0 ID:18704 IpLen:24 DgmLen:32
IP Options (1) => RTRALT
11 64 EE 9B 00 00 00 00                          .d......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
06/16-18:18:16.300459 192.168.1.2 -> 224.0.0.252
PROTO:002 TTL:1 TOS:0x0 ID:9270 IpLen:24 DgmLen:32
IP Options (1) => RTRALT
16 00 09 03 E0 00 00 FC                          ........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

RESULT

Thus Intrusion Detection System was demonstrated using Snort tool successfully.

Ex. No : 10

Date : **Exploring N-Stalker, a Vulnerability Assessment Tool**

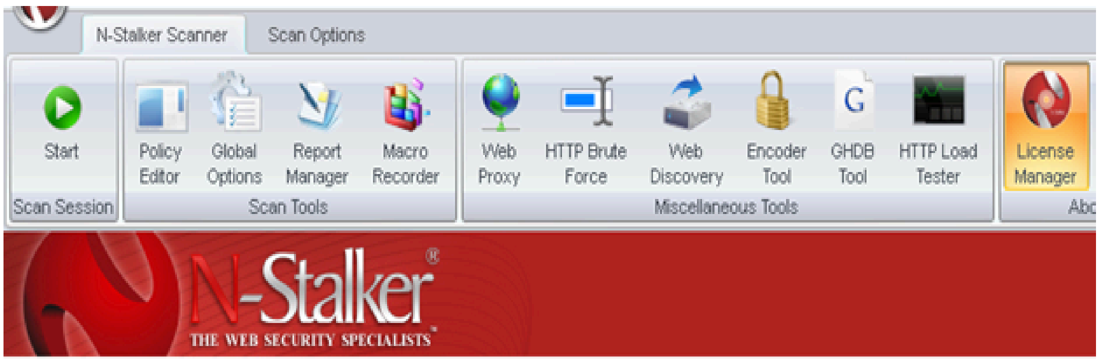
AIM:

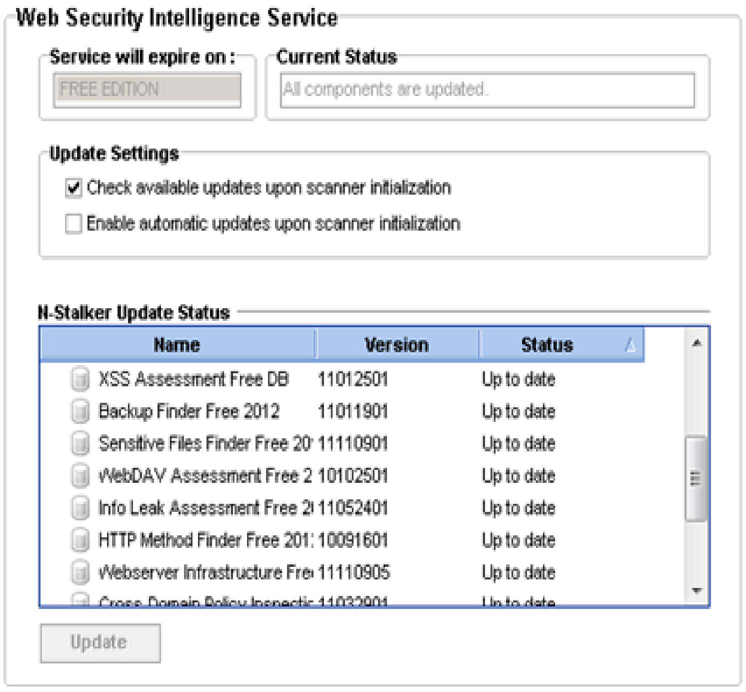
To download the N-Stalker Vulnerability Assessment Tool and exploring the features.

EXPLORING N-STALKER:

- ❖ N-Stalker Web Application Security Scanner is a Web security assessment tool.
- ❖ It incorporates with a well-known N-Stealth HTTP Security Scanner and 39,000 Web attack signature database.
- ❖ This tool also comes in both free and paid version.
- ❖ Before scanning the target, go to “License Manager” tab, perform the update.
- ❖ Once update, you will note the status as up to date.
- ❖ You need to download and install N-Stalker from www.nstalker.com.

1. Start N-Stalker from a Windows computer. The program is installed under Start ➪ Programs ➪ N-Stalker ➪ N-Stalker Free Edition.
2. Enter a host address or a range of addresses to scan.
3. Click Start Scan.
4. After the scan completes, the N-Stalker Report Manager will prompt
5. you to select a format for the resulting report as choose Generate HTML.
6. Review the HTML report for vulnerabilities.




Web Security Intelligence Service

Service will expire on : FREE EDITION

Current Status: All components are updated.

Update Settings

- [✓] Check available updates upon scanner initialization
- [ ] Enable automatic updates upon scanner initialization

N-Stalker Update Status

Name	Version	Status
XSS Assessment Free DB	11012501	Up to date
Backup Finder Free 2012	11011901	Up to date
Sensitive Files Finder Free 20	11110901	Up to date
WebDAV Assessment Free 2	10102501	Up to date
Info Leak Assessment Free 2	11052401	Up to date
HTTP Method Finder Free 201	10091601	Up to date
Webserver Infrastructure Fre	11110905	Up to date
Cross Domain Policy Inspectio	11032901	Up to date

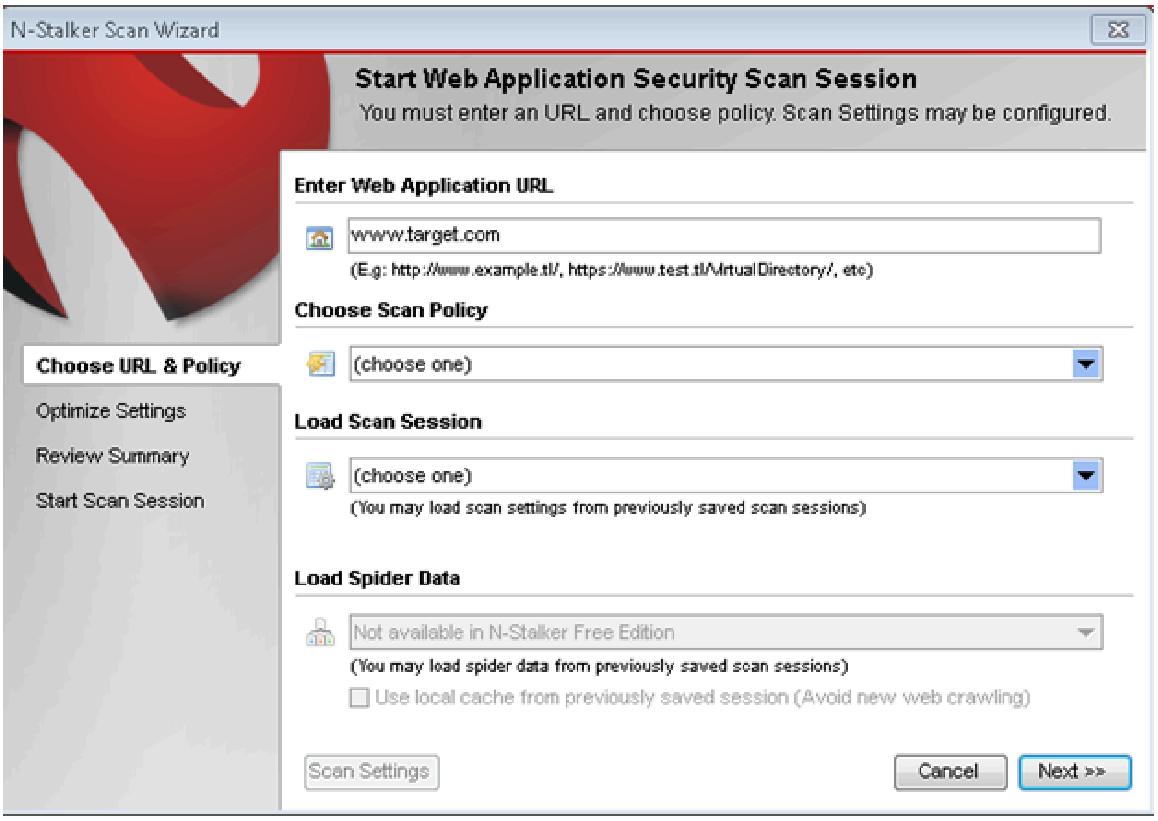
Update

Now goto "Scan Session", enter the target URL.
In scan policy, you can select from the four options,

- ❖ Manual test which will crawl the website and will be waiting for manual attacks.
- ❖ Full Cross Site Scripting (XSS) assessment
- ❖ Open Web Application Security Project (OWASP) policy
- ❖ Web server infrastructure analysis.

Once, the option has been selected, next step is "Optimize settings" which will crawl the whole website for further analysis.

In review option, you can get all the information like host information, technologies used, policy name, etc.



Start Web Application Security Scan Session

You must enter an URL and choose policy. Scan Settings may be configured.

Enter Web Application URL

www.target.com

(E.g: http://www.example.tl/, https://www.test.tl/VirtualDirectory/, etc)

Choose Scan Policy

(choose one)

Load Scan Session

(choose one)

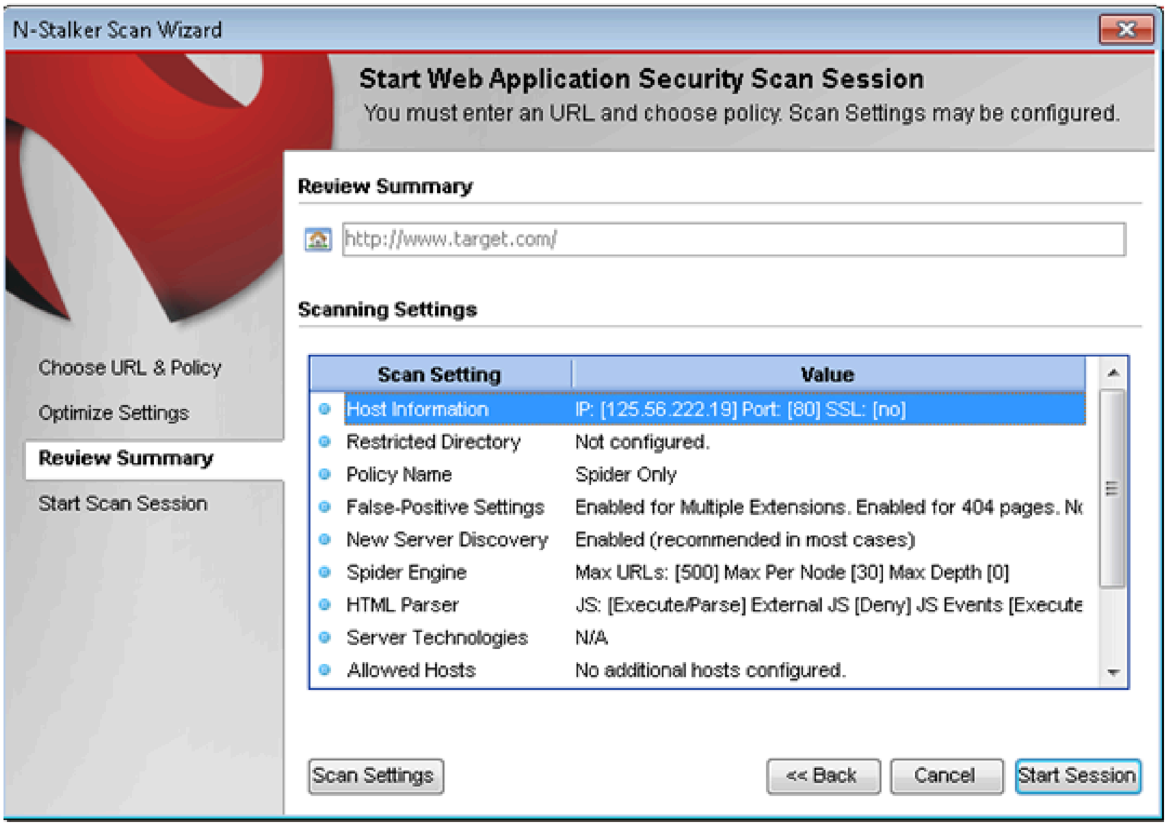
(You may load scan settings from previously saved scan sessions)

Load Spider Data

Not available in N-Stalker Free Edition

(You may load spider data from previously saved scan sessions)

☐ Use local cache from previously saved session (Avoid new web crawling)



Start Web Application Security Scan Session

You must enter an URL and choose policy. Scan Settings may be configured.

Review Summary

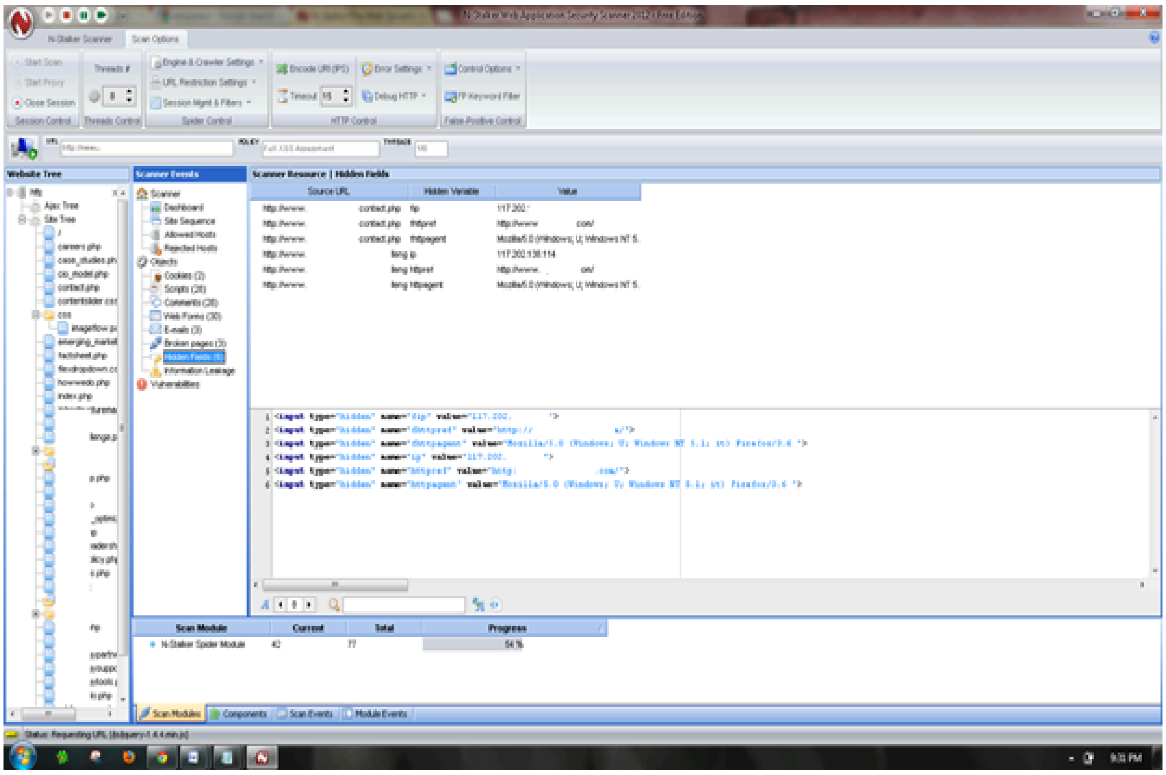
http://www.target.com/

Scanning Settings

Scan Setting	Value
Host Information	IP: [125.56.222.19] Port: [80] SSL: [no]
Restricted Directory	Not configured.
Policy Name	Spider Only
False-Positive Settings	Enabled for Multiple Extensions. Enabled for 404 pages. Nc
New Server Discovery	Enabled (recommended in most cases)
Spider Engine	Max URLs: [500] Max Per Node [30] Max Depth [0]
HTML Parser	JS: [Execute/Parse] External JS [Deny] JS Events [Execute
Server Technologies	N/A
Allowed Hosts	No additional hosts configured.

Once done, start the session and start the scan.

The scanner will crawl the whole website and will show the scripts, broken pages, hidden fields, information leakage, web forms related information which helps to analyze further.


Scanner Resource | Hidden Fields

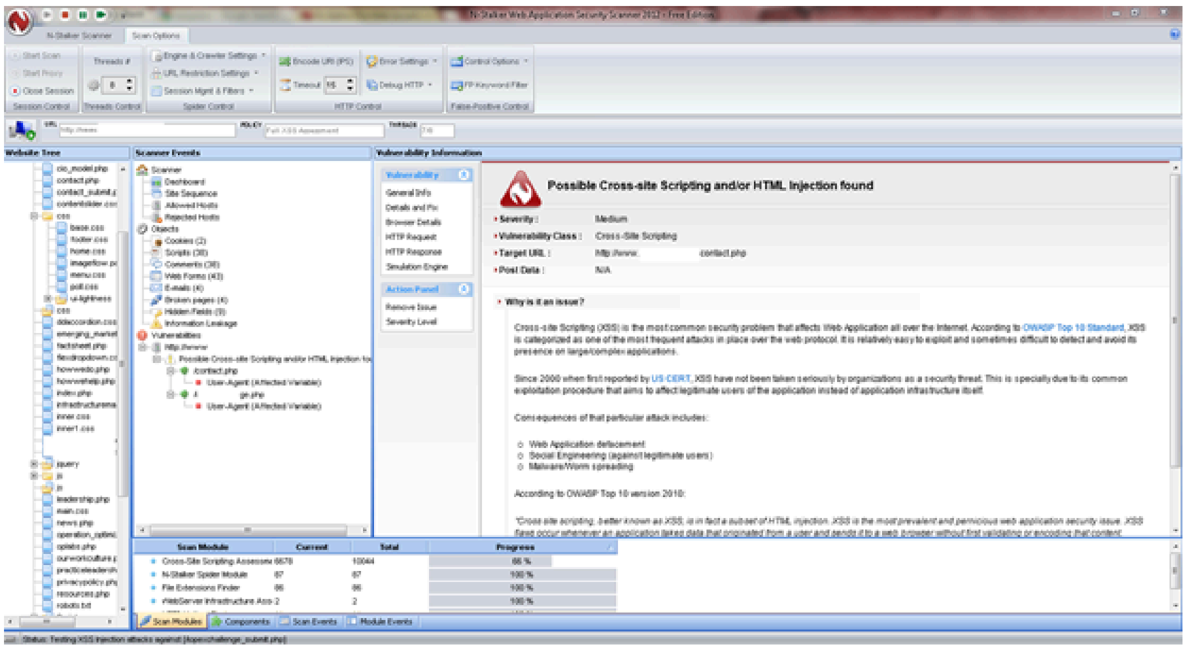
Source URL	Hidden Variable	Value
http://www. contact.php	fip	117.202.
http://www. contact.php	fhttpref	http://www .com/
http://www. contact.php	fhttpagent	Mozilla/5.0 (Windows; U; Windows NT 5.
http://www. feng	ip	117.202.138.114
http://www. feng	httpref	http://www. .om/
http://www. feng	httpagent	Mozilla/5.0 (Windows; U; Windows NT 5.

```
<input type="hidden" name="fip" value="117.202.      ">
<input type="hidden" name="fhttpref" value="http://              m/">
<input type="hidden" name="fhttpagent" value="Mozilla/5.0 (Windows; U; Windows NT 5.1; it) Firefox/3.6 ">
<input type="hidden" name="ip" value="117.202.      ">
<input type="hidden" name="httpref" value="http:              .com/">
<input type="hidden" name="httpagent" value="Mozilla/5.0 (Windows; U; Windows NT 5.1; it) Firefox/3.6 ">
```

Scan Module	Current	Total	Progress
N-Stalker Spider Module	42	77	54 %

Status: Requesting URL [/js/jquery-1.4.4.min.js]

Once the scan is completed, the N-Stalker scanner will show details like severity level, vulnerability class, why is it an issue, the fix for the issue and the URL which is vulnerable to the particular vulnerability?


Vulnerability Information

Possible Cross-site Scripting and/or HTML Injection found

- **Severity :** Medium
- **Vulnerability Class :** Cross-Site Scripting
- **Target URL :** http://www.   contact.php
- **Post Data :** N/A

Why is it an issue?

Cross-site Scripting (XSS) is the most common security problem that affects Web Application all over the Internet. According to OWASP Top 10 Standard, XSS is categorized as one of the most frequent attacks in place over the web protocol. It is relatively easy to exploit and sometimes difficult to detect and avoid its presence on large/complex applications.

Since 2000 when first reported by US-CERT, XSS have not been taken seriously by organizations as a security threat. This is specially due to its common exploitation procedure that aims to affect legitimate users of the application instead of application infrastructure itself.

Consequences of that particular attack includes:

- Web Application defacement
- Social Engineering (against legitimate users)
- Malware/Worm spreading

According to OWASP Top 10 version 2010:

"Cross site scripting, better known as XSS, is in fact a subset of HTML injection. XSS is the most prevalent and pernicious web application security issue. XSS flaws occur whenever an application takes data that originated from a user and sends it to a web browser without first validating or encoding that content.

Scan Module	Current	Total	Progress
Cross-Site Scripting Assessme	6678	10044	66 %
N-Stalker Spider Module	87	87	100 %
File Extensions Finder	86	86	100 %
WebServer Infrastructure Ass	2	2	100 %

Status: Testing XSS Injection attacks against [/opechallenge_submit.php]

RESULT:
Thus the N-Stalker Vulnerability Assessment tool has been downloaded, installed and the features has been explored by using a vulnerable website.

Ex. No : 11
Date : **Defeating Malware - Building Trojans**
AIM:
To build a Trojan and know the harmness of the trojan malwares in a computer system.

TROJAN:

❖ In computing, a Trojan horse, or Trojan, is any malware which misleads users of its true intent.

❖ Trojans are generally spread by some form of social engineering, for example where a user is duped into executing an email attachment disguised to appear not suspicious, (e.g., a routine form to be filled in), or by clicking on some fake advertisement on social media or anywhere else.

❖ Although their payload can be anything, many modern forms act as a backdoor, contacting a controller which can then have unauthorized access to the affected computer.

❖ Trojans may allow an attacker to access users' personal information such as banking information, passwords, or personal identity.

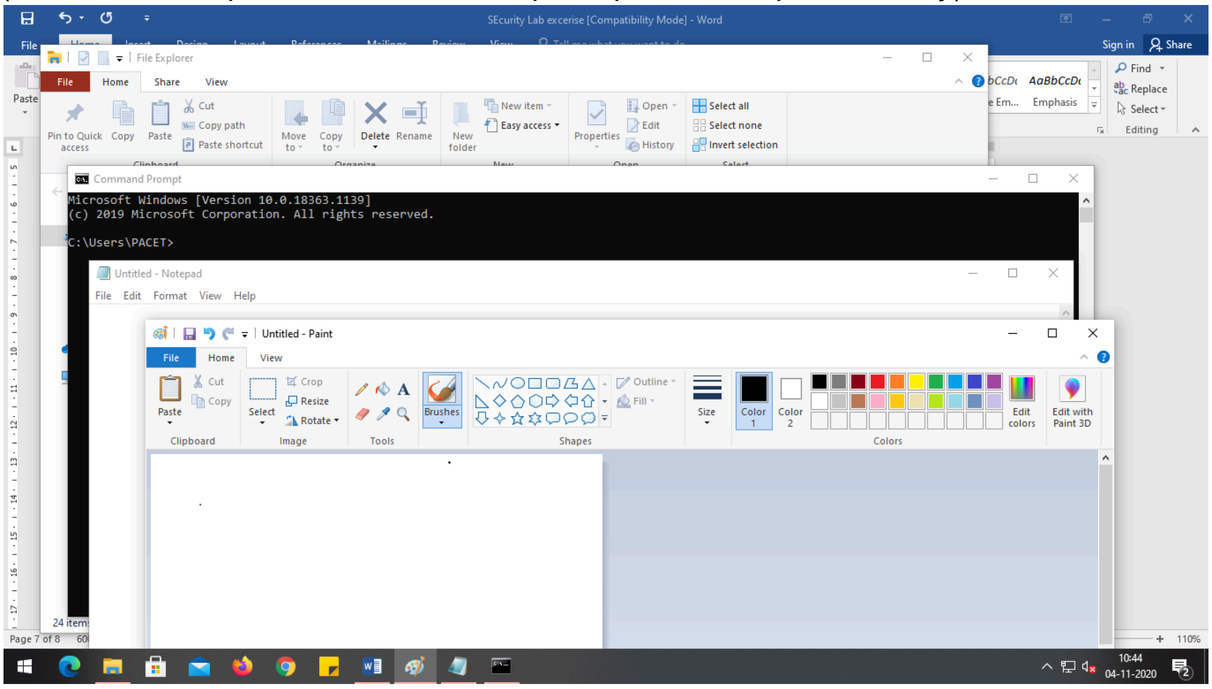
❖ Example: Ransomware attacks are often carried out using a trojan.

PROCEDURE:
1. Create a simple trojan by using Windows Batch File (.bat)
2. Type these below code in notepad and save it as Trojan.bat
3. Double click on Trojan.bat file.
4. When the trojan code executes, it will open MS-Paint, Notepad, Command Prompt, Explorer, etc., infinitely.
5. Restart the computer to stop the execution of this trojan.

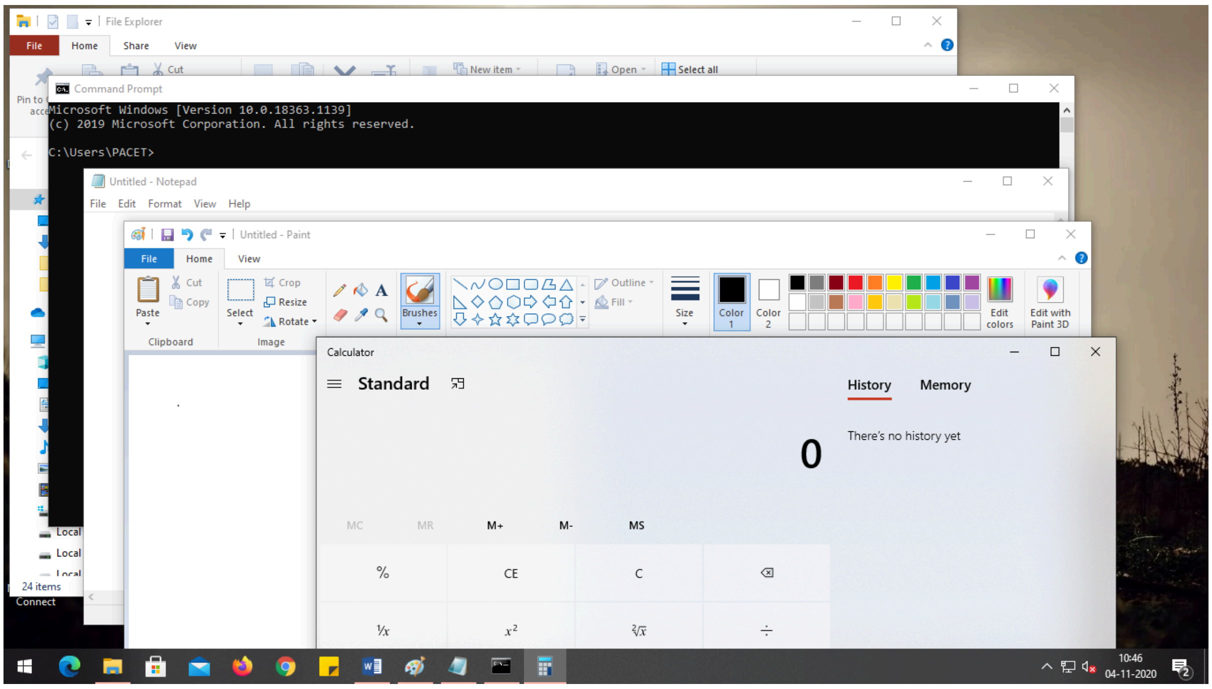
CODE:
<u>Trojan.bat</u>
```
@echo off
:x
start mspaint
start notepad
start cmd
start explorer
start control
start calc
goto x
```

OUTPUT

(MS-Paint, Notepad, Command Prompt, Explorer will open infinitely)



```
Microsoft Windows [Version 10.0.18363.1139]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\PACET>
```



```
Microsoft Windows [Version 10.0.18363.1139]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\PACET>
```

RESULT:
Thus a trojan has been built and the harmness of the trojan viruses has been explored.

Ex.No: 12 **INSTALLATION OF ROOTKITS AND STUDY ABOUT THE VARIETY OF OPTIONS**

AIM

To install rootkits and study about the variety of options.

PROCEDURE

Rootkit is a stealth type of malicious software designed to hide the existence of certain process from normal methods of detection and enables continued privileged access to a computer.

Step 1: Download Rootkit Tool from GMER website. www.gmer.net

Step 2: This displays the Processes, Modules, Services, Files, Registry, RootKit/Malwares, Autostart, CMD of local host.

Step 3: Select Processes menu and kill any unwanted process if any.

Step 4: Modules menu displays the various system files like .sys, .dll

Step 5: Services menu displays the complete services running with Autostart, Enable, Disable, System, Boot.

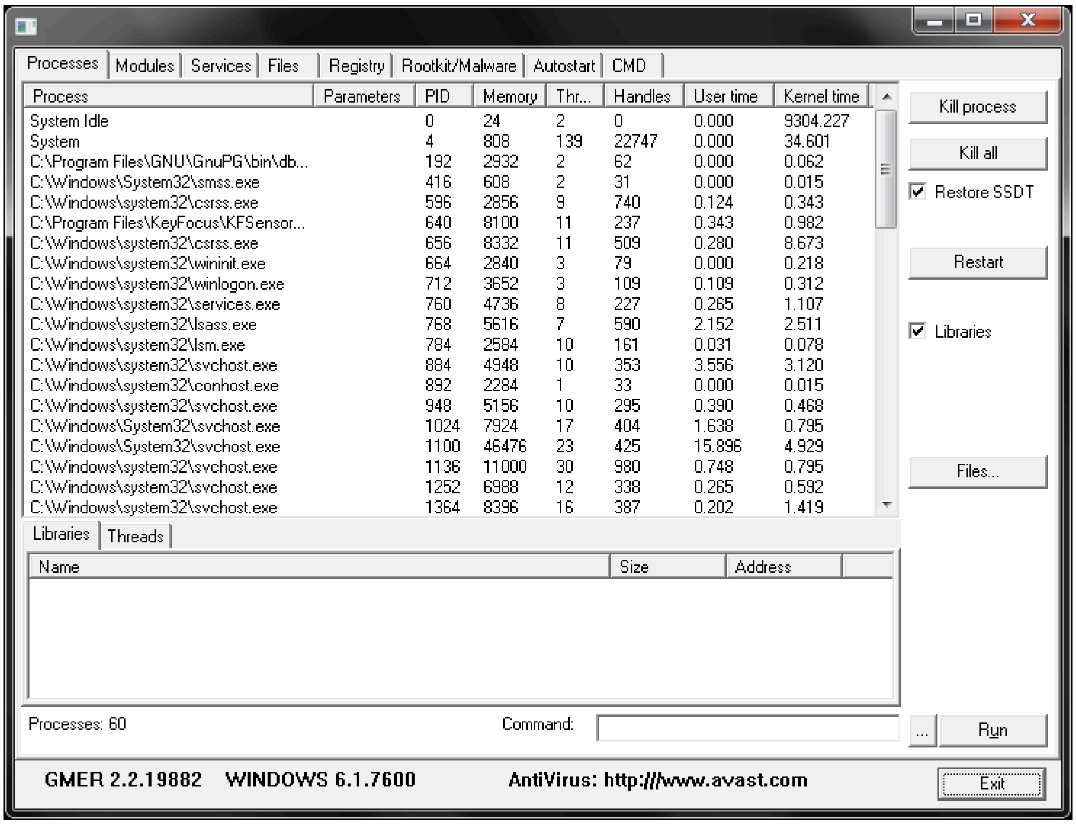
Step 6: Files menu displays full files on Hard-Disk volumes.

Step 7: Registry displays Hkey_Current_user and Hkey_Local_Machine.

Step 8: Rootkits/Malawares scans the local drives selected.

Step 9: Autostart displays the registry base Autostart applications.

Step 10: CMD allows the user to interact with command line utilities or Registry.

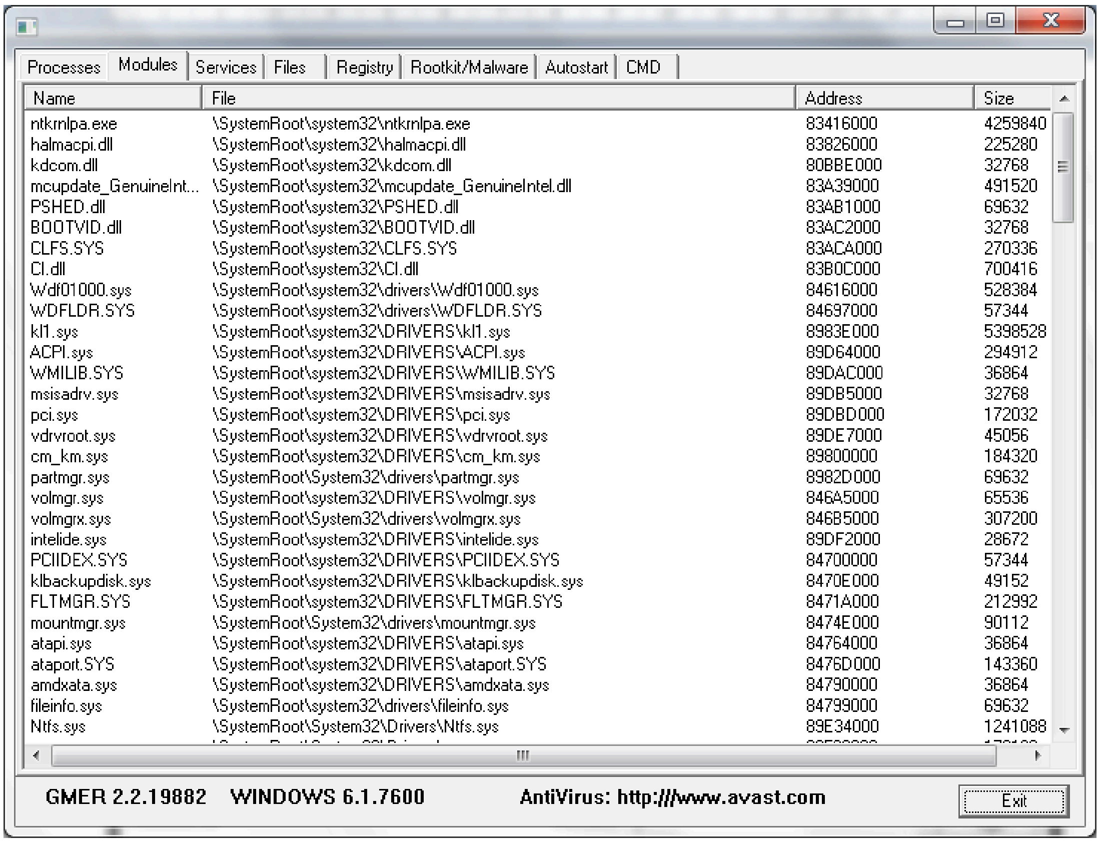


Process	Parameters	PID	Memory	Thr...	Handles	User time	Kernel time
System Idle		0	24	2	0	0.000	9304.227
System		4	808	139	22747	0.000	34.601
C:\Program Files\GNU\GnuPG\bin\db...		192	2932	2	62	0.000	0.062
C:\Windows\System32\smss.exe		416	608	2	31	0.000	0.015
C:\Windows\system32\csrss.exe		596	2856	9	740	0.124	0.343
C:\Program Files\KeyFocus\KFSensor...		640	8100	11	237	0.343	0.982
C:\Windows\system32\csrss.exe		656	8332	11	509	0.280	8.673
C:\Windows\system32\wininit.exe		664	2840	3	79	0.000	0.218
C:\Windows\system32\winlogon.exe		712	3652	3	109	0.109	0.312
C:\Windows\system32\services.exe		760	4736	8	227	0.265	1.107
C:\Windows\system32\lsass.exe		768	5616	7	590	2.152	2.511
C:\Windows\system32\lsm.exe		784	2584	10	161	0.031	0.078
C:\Windows\system32\svchost.exe		884	4948	10	353	3.556	3.120
C:\Windows\system32\conhost.exe		892	2284	1	33	0.000	0.015
C:\Windows\system32\svchost.exe		948	5156	10	295	0.390	0.468
C:\Windows\System32\svchost.exe		1024	7924	17	404	1.638	0.795
C:\Windows\System32\svchost.exe		1100	46476	23	425	15.896	4.929
C:\Windows\system32\svchost.exe		1136	11000	30	980	0.748	0.795
C:\Windows\system32\svchost.exe		1252	6988	12	338	0.265	0.592
C:\Windows\system32\svchost.exe		1364	8396	16	387	0.202	1.419

[✓] Restore SSDT
[✓] Libraries

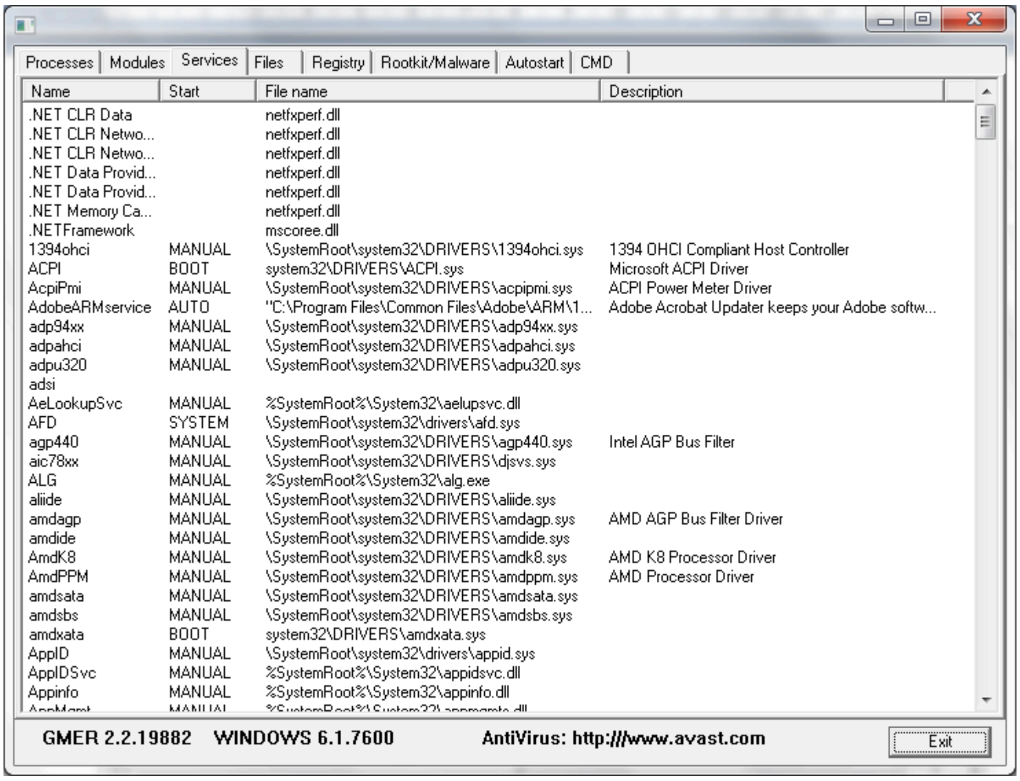
Processes: 60

GMER 2.2.19882 WINDOWS 6.1.7600 AntiVirus: http://www.avast.com



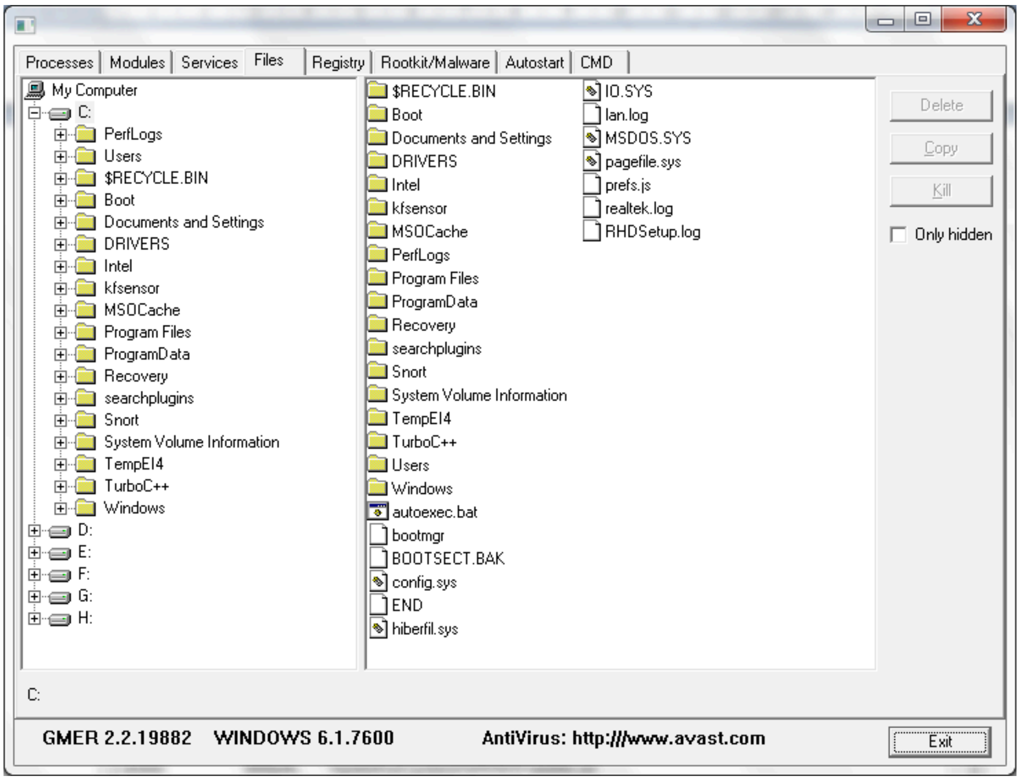
Name	File	Address	Size
ntkrnlpa.exe	\SystemRoot\system32\ntkrnlpa.exe	83416000	4259840
halmacpi.dll	\SystemRoot\system32\halmacpi.dll	83826000	225280
kdcom.dll	\SystemRoot\system32\kdcom.dll	80BBE000	32768
mcupdate_GenuineInt...	\SystemRoot\system32\mcupdate_GenuineIntel.dll	83A39000	491520
PSHED.dll	\SystemRoot\system32\PSHED.dll	83AB1000	69632
BOOTVID.dll	\SystemRoot\system32\BOOTVID.dll	83AC2000	32768
CLFS.SYS	\SystemRoot\system32\CLFS.SYS	83ACA000	270336
CI.dll	\SystemRoot\system32\CI.dll	83B0C000	700416
Wdf01000.sys	\SystemRoot\system32\drivers\Wdf01000.sys	84616000	528384
WDFLDR.SYS	\SystemRoot\system32\drivers\WDFLDR.SYS	84697000	57344
kl1.sys	\SystemRoot\system32\DRIVERS\kl1.sys	8983E000	5398528
ACPI.sys	\SystemRoot\system32\DRIVERS\ACPI.sys	89D64000	294912
WMILIB.SYS	\SystemRoot\system32\DRIVERS\WMILIB.SYS	89DAC000	36864
msisadrv.sys	\SystemRoot\system32\DRIVERS\msisadrv.sys	89DB5000	32768
pci.sys	\SystemRoot\system32\DRIVERS\pci.sys	89DBD000	172032
vdrvroot.sys	\SystemRoot\system32\DRIVERS\vdrvroot.sys	89DE7000	45056
cm_km.sys	\SystemRoot\system32\DRIVERS\cm_km.sys	89800000	184320
partmgr.sys	\SystemRoot\System32\drivers\partmgr.sys	8982D000	69632
volmgr.sys	\SystemRoot\system32\DRIVERS\volmgr.sys	846A5000	65536
volmgrx.sys	\SystemRoot\System32\drivers\volmgrx.sys	846B5000	307200
intelide.sys	\SystemRoot\system32\DRIVERS\intelide.sys	89DF2000	28672
PCIIDEX.SYS	\SystemRoot\system32\DRIVERS\PCIIDEX.SYS	84700000	57344
klbackupdisk.sys	\SystemRoot\system32\DRIVERS\klbackupdisk.sys	8470E000	49152
FLTMGR.SYS	\SystemRoot\system32\DRIVERS\FLTMGR.SYS	8471A000	212992
mountmgr.sys	\SystemRoot\System32\drivers\mountmgr.sys	8474E000	90112
atapi.sys	\SystemRoot\system32\DRIVERS\atapi.sys	84764000	36864
ataport.SYS	\SystemRoot\system32\DRIVERS\ataport.SYS	8476D000	143360
amdxata.sys	\SystemRoot\system32\DRIVERS\amdxata.sys	84790000	36864
fileinfo.sys	\SystemRoot\system32\drivers\fileinfo.sys	84799000	69632
Ntfs.sys	\SystemRoot\System32\Drivers\Ntfs.sys	89E34000	1241088

GMER 2.2.19882 WINDOWS 6.1.7600 AntiVirus: http:///www.avast.com



Name	Start	File name	Description
.NET CLR Data		netfxperf.dll	
.NET CLR Netwo...		netfxperf.dll	
.NET CLR Netwo...		netfxperf.dll	
.NET Data Provid...		netfxperf.dll	
.NET Data Provid...		netfxperf.dll	
.NET Memory Ca...		netfxperf.dll	
.NETFramework		mscoree.dll	
1394ohci	MANUAL	\SystemRoot\system32\DRIVERS\1394ohci.sys	1394 OHCI Compliant Host Controller
ACPI	BOOT	system32\DRIVERS\ACPI.sys	Microsoft ACPI Driver
AcpiPmi	MANUAL	\SystemRoot\system32\DRIVERS\acpipmi.sys	ACPI Power Meter Driver
AdobeARMservice	AUTO	"C:\Program Files\Common Files\Adobe\ARM\1...	Adobe Acrobat Updater keeps your Adobe softw...
adp94xx	MANUAL	\SystemRoot\system32\DRIVERS\adp94xx.sys	
adpahci	MANUAL	\SystemRoot\system32\DRIVERS\adpahci.sys	
adpu320	MANUAL	\SystemRoot\system32\DRIVERS\adpu320.sys	
adsi			
AeLookupSvc	MANUAL	%SystemRoot%\System32\aelupsvc.dll	
AFD	SYSTEM	\SystemRoot\system32\drivers\afd.sys	
agp440	MANUAL	\SystemRoot\system32\DRIVERS\agp440.sys	Intel AGP Bus Filter
aic78xx	MANUAL	\SystemRoot\system32\DRIVERS\djsvs.sys	
ALG	MANUAL	%SystemRoot%\System32\alg.exe	
aliide	MANUAL	\SystemRoot\system32\DRIVERS\aliide.sys	
amdagp	MANUAL	\SystemRoot\system32\DRIVERS\amdagp.sys	AMD AGP Bus Filter Driver
amdide	MANUAL	\SystemRoot\system32\DRIVERS\amdide.sys	
AmdK8	MANUAL	\SystemRoot\system32\DRIVERS\amdk8.sys	AMD K8 Processor Driver
AmdPPM	MANUAL	\SystemRoot\system32\DRIVERS\amdppm.sys	AMD Processor Driver
amdsata	MANUAL	\SystemRoot\system32\DRIVERS\amdsata.sys	
amdsbs	MANUAL	\SystemRoot\system32\DRIVERS\amdsbs.sys	
amdxata	BOOT	system32\DRIVERS\amdxata.sys	
AppID	MANUAL	\SystemRoot\system32\drivers\appid.sys	
AppIDSvc	MANUAL	%SystemRoot%\System32\appidsvc.dll	
Appinfo	MANUAL	%SystemRoot%\System32\appinfo.dll	

GMER 2.2.19882 WINDOWS 6.1.7600 AntiVirus: http:///www.avast.com

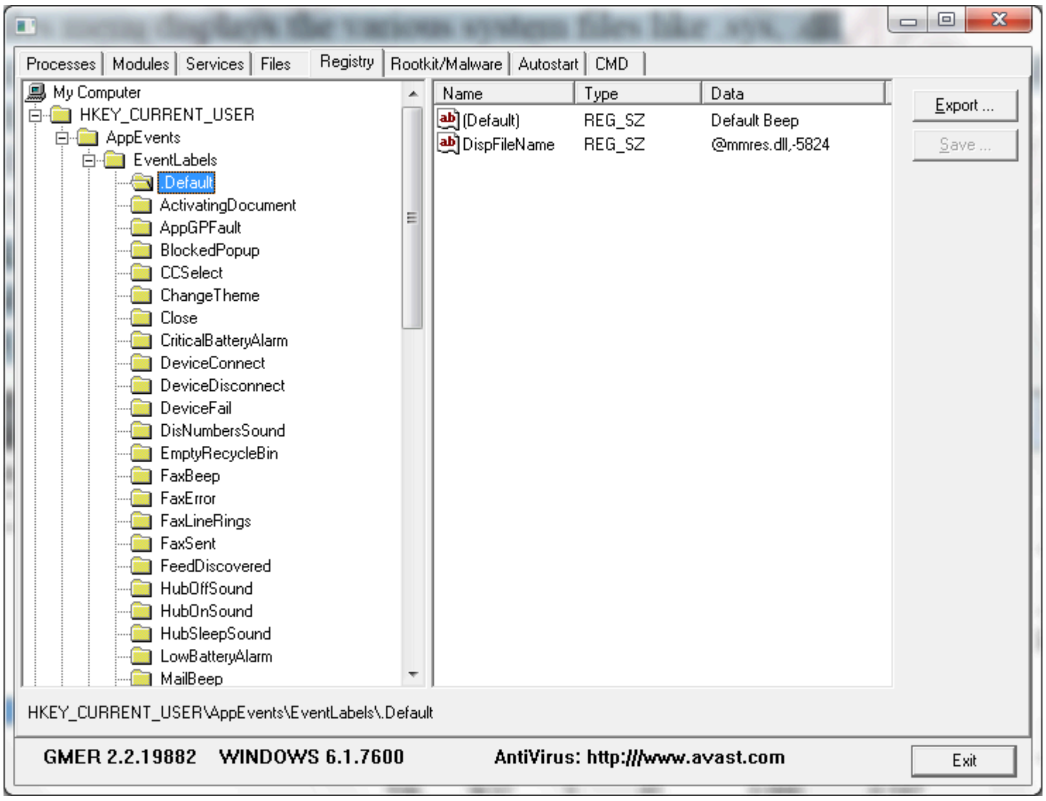


```
My Computer
  C:
    PerfLogs
    Users
    $RECYCLE.BIN
    Boot
    Documents and Settings
    DRIVERS
    Intel
    kfsensor
    MSOCache
    Program Files
    ProgramData
    Recovery
    searchplugins
    Snort
    System Volume Information
    TempEI4
    TurboC++
    Windows
  D:
  E:
  F:
  G:
  H:
```

```
$RECYCLE.BIN                IO.SYS
Boot                        lan.log
Documents and Settings      MSDOS.SYS
DRIVERS                     pagefile.sys
Intel                       prefs.js
kfsensor                    realtek.log
MSOCache                    RHDSetup.log
PerfLogs
Program Files
ProgramData
Recovery
searchplugins
Snort
System Volume Information
TempEI4
TurboC++
Users
Windows
autoexec.bat
bootmgr
BOOTSECT.BAK
config.sys
END
hiberfil.sys
```

C:

GMER 2.2.19882   WINDOWS 6.1.7600   AntiVirus: http://www.avast.com

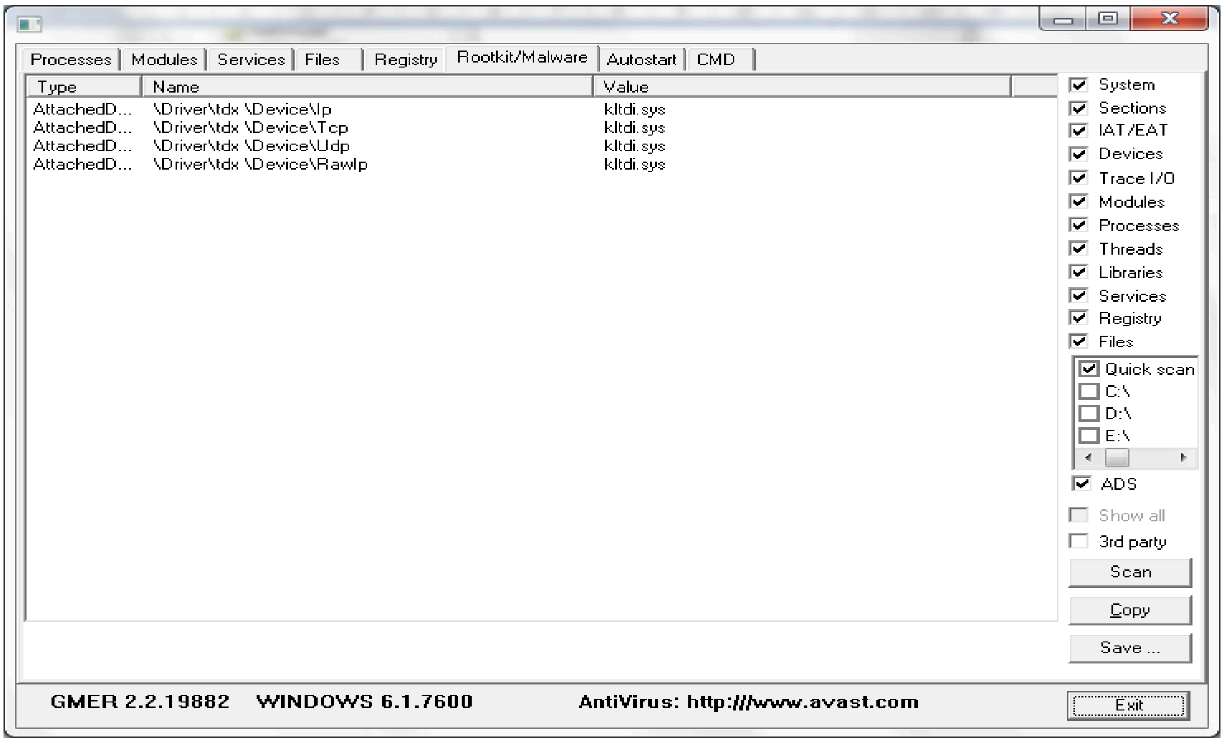


```
My Computer
  HKEY_CURRENT_USER
    AppEvents
      EventLabels
        Default
        ActivatingDocument
        AppGPFault
        BlockedPopup
        CCSelect
        ChangeTheme
        Close
        CriticalBatteryAlarm
        DeviceConnect
        DeviceDisconnect
        DeviceFail
        DisNumbersSound
        EmptyRecycleBin
        FaxBeep
        FaxError
        FaxLineRings
        FaxSent
        FeedDiscovered
        HubOffSound
        HubOnSound
        HubSleepSound
        LowBatteryAlarm
        MailBeep
```

Name	Type	Data
(Default)	REG_SZ	Default Beep
DispFileName	REG_SZ	@mmres.dll,-5824

HKEY_CURRENT_USER\AppEvents\EventLabels\.Default

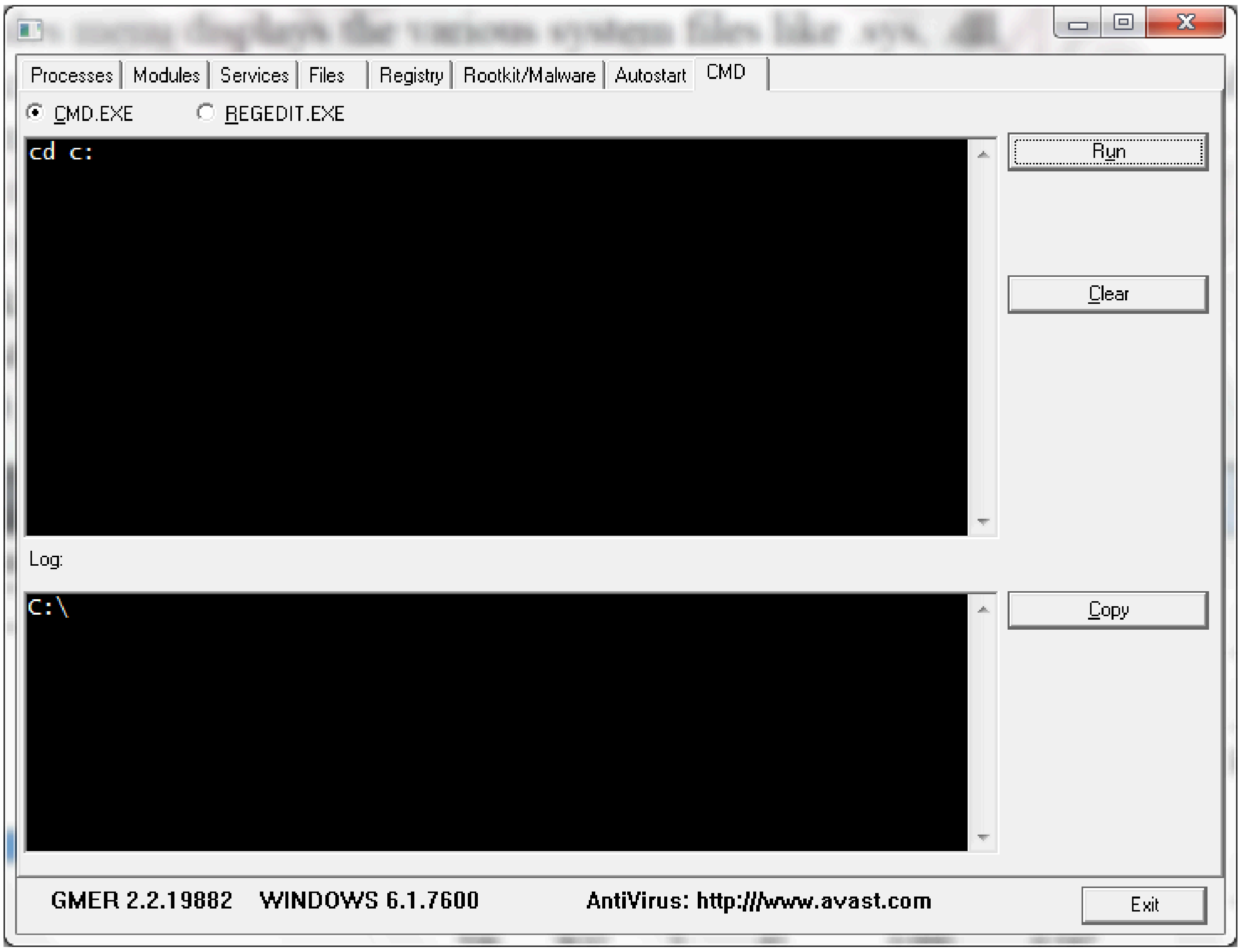
GMER 2.2.19882   WINDOWS 6.1.7600   AntiVirus: http://www.avast.com



```
Type         Name                          Value
AttachedD... \Driver\tdx \Device\Ip        kltdi.sys
AttachedD... \Driver\tdx \Device\Tcp       kltdi.sys
AttachedD... \Driver\tdx \Device\Udp       kltdi.sys
AttachedD... \Driver\tdx \Device\RawIp     kltdi.sys
```

[✓] System
[✓] Sections
[✓] IAT/EAT
[✓] Devices
[✓] Trace I/O
[✓] Modules
[✓] Processes
[✓] Threads
[✓] Libraries
[✓] Services
[✓] Registry
[✓] Files
[✓] Quick scan
[ ] C:\
[ ] D:\
[ ] E:\
[✓] ADS
[ ] Show all
[ ] 3rd party

GMER 2.2.19882 WINDOWS 6.1.7600 AntiVirus: http://www.avast.com



(● CMD.EXE) ○ REGEDIT.EXE

```
cd c:
```

Log:

```
C:\
```

GMER 2.2.19882 WINDOWS 6.1.7600 AntiVirus: http://www.avast.com

RESULT

Thus Rootkit was installed and various options were studied successfully.

Ex.No:13 PERFORM WIRELESS AUDIT ON AN ACCESS POINT OR A ROUTER AND DECRYPT WEP AND WPA USING NET STUMBLER

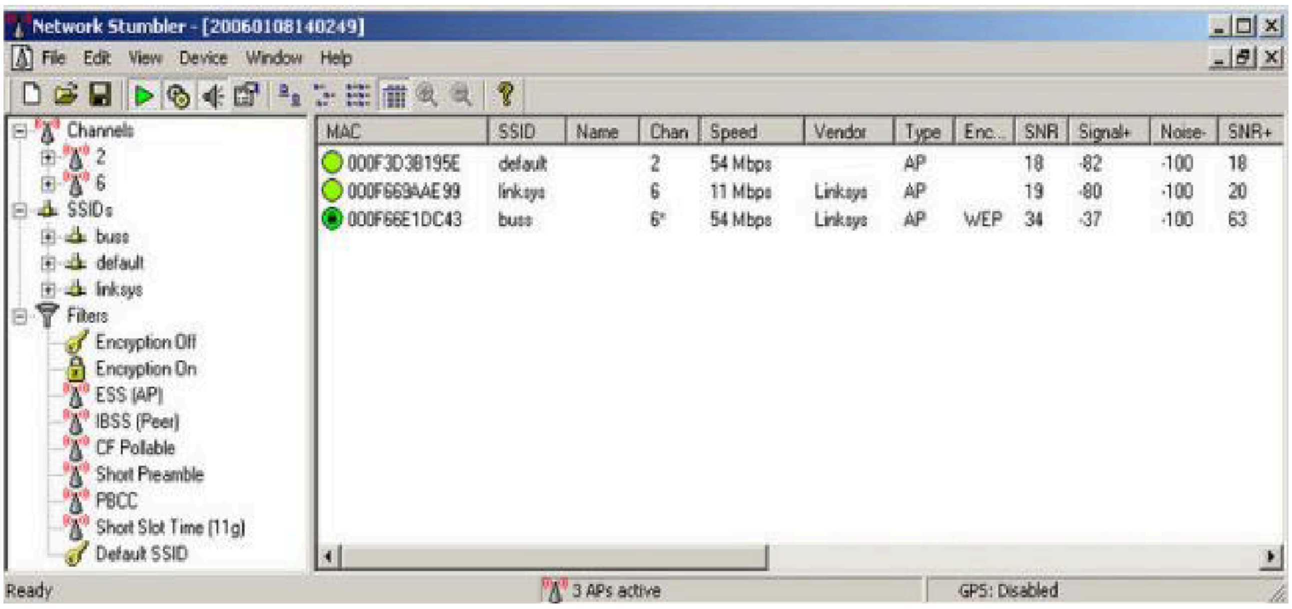
AIM

To perform wireless audit on an access point or a router and decrypt WEP and WPA using Net Stumbler.

PROCEDURE

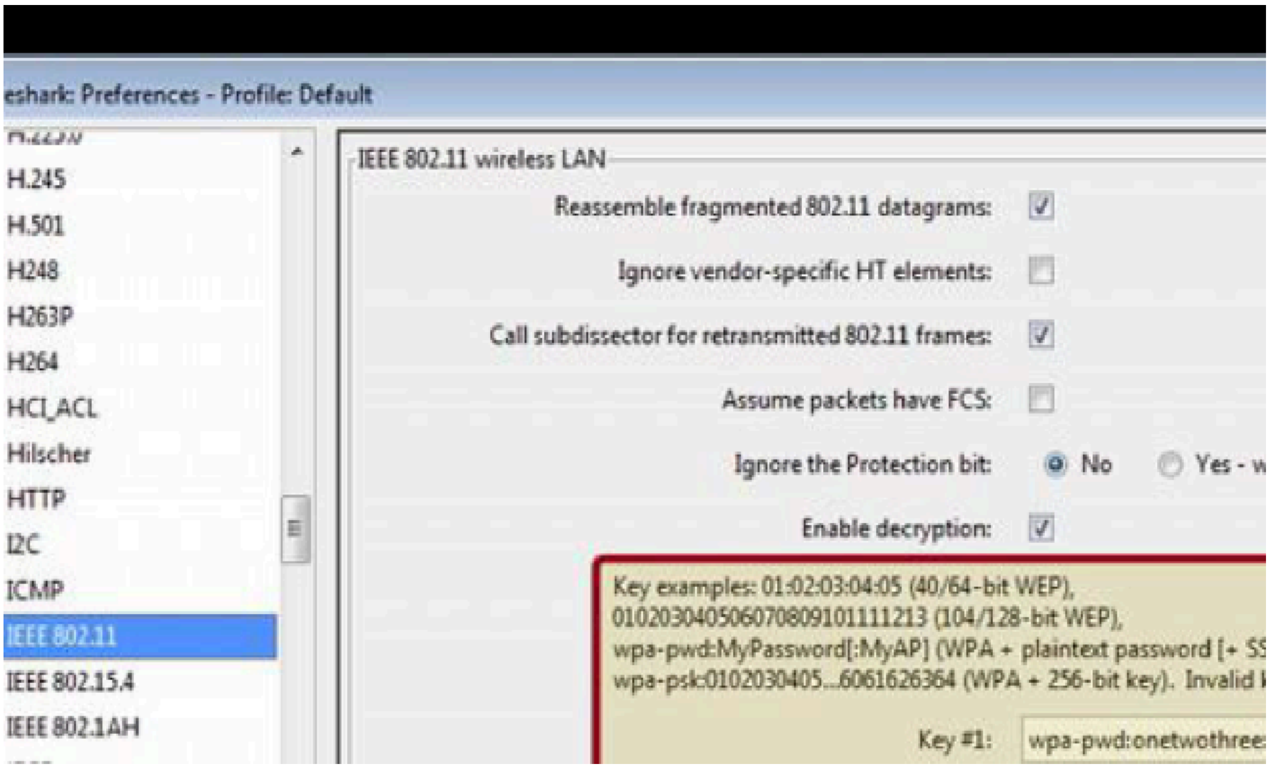
NetStumbler (Network Stumbler) is one of the Wi-Fi hacking tool which only compatible with windows, this tool also a freeware. With this program, we can search for wireless network which open and infiltrate the network. Its having some compatibility and network adapter issues.

Step 1: Download and install Netstumbler
Step 2: It is highly recommended that your PC should have wireless network card in order to access wireless router.
Step 3: Now Run Netstumbler in record mode and configure wireless card.
Step 4: There are several indicators regarding the strength of the signal, such as GREEN indicates Strong, YELLOW and other color indicates a weaker signal, RED indicates a very weak and GREY indicates a signal loss.
Step 5: Lock symbol with GREEN bubble indicates the Access point has encryption enabled.
Step 6: MAC assigned to Wireless Access Point is displayed on right hand pane.
Step 7: The next coloumn displays the Access points Service Set Identifier[SSID] which is useful to crack the password.
Step 8: To decrypt use WireShark tool by selecting Edit ☾ preferences ☾ IEEE 802.11.
Step 9: Enter the WEP keys as a string of hexadecimal numbers as A1B2C3D4E5.
Step 10: Stop the tool.



MAC	SSID	Name	Chan	Speed	Vendor	Type	Enc...	SNR	Signal+	Noise-	SNR+
000F3D3B195E	default		2	54 Mbps		AP		18	-82	-100	18
000F669AAE99	linksys		6	11 Mbps	Linksys	AP		19	-80	-100	20
000F66E1DC43	buss		6*	54 Mbps	Linksys	AP	WEP	34	-37	-100	63

Ready 3 APs active GPS: Disabled


eshark: Preferences - Profile: Default

H.245
H.501
H248
H263P
H264
HCI_ACL
Hilscher
HTTP
I2C
ICMP
IEEE 802.11
IEEE 802.15.4
IEEE 802.1AH

IEEE 802.11 wireless LAN

Reassemble fragmented 802.11 datagrams: ✓

Ignore vendor-specific HT elements: ☐

Call subdissector for retransmitted 802.11 frames: ✓

Assume packets have FCS: ☐

Ignore the Protection bit: [✓ No] Yes - w

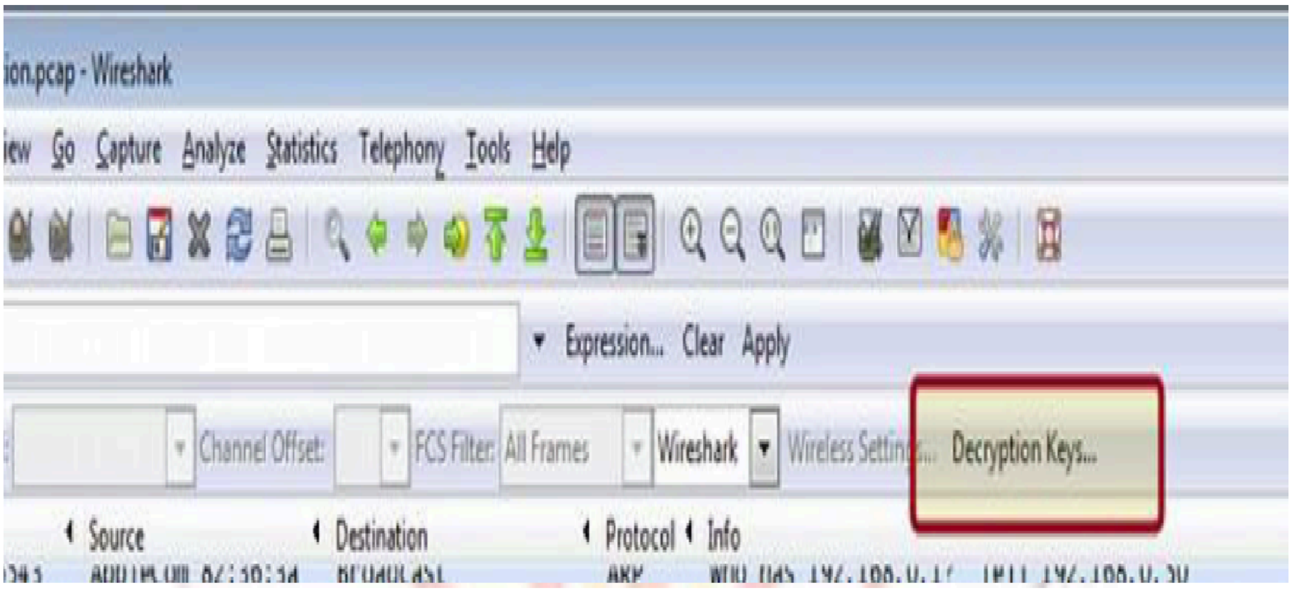
Enable decryption: ✓

Key examples: 01:02:03:04:05 (40/64-bit WEP),
0102030405060708091011111213 (104/128-bit WEP),
wpa-pwd:MyPassword[:MyAP] (WPA + plaintext password [+ SS
wpa-psk:0102030405...6061626364 (WPA + 256-bit key). Invalid k

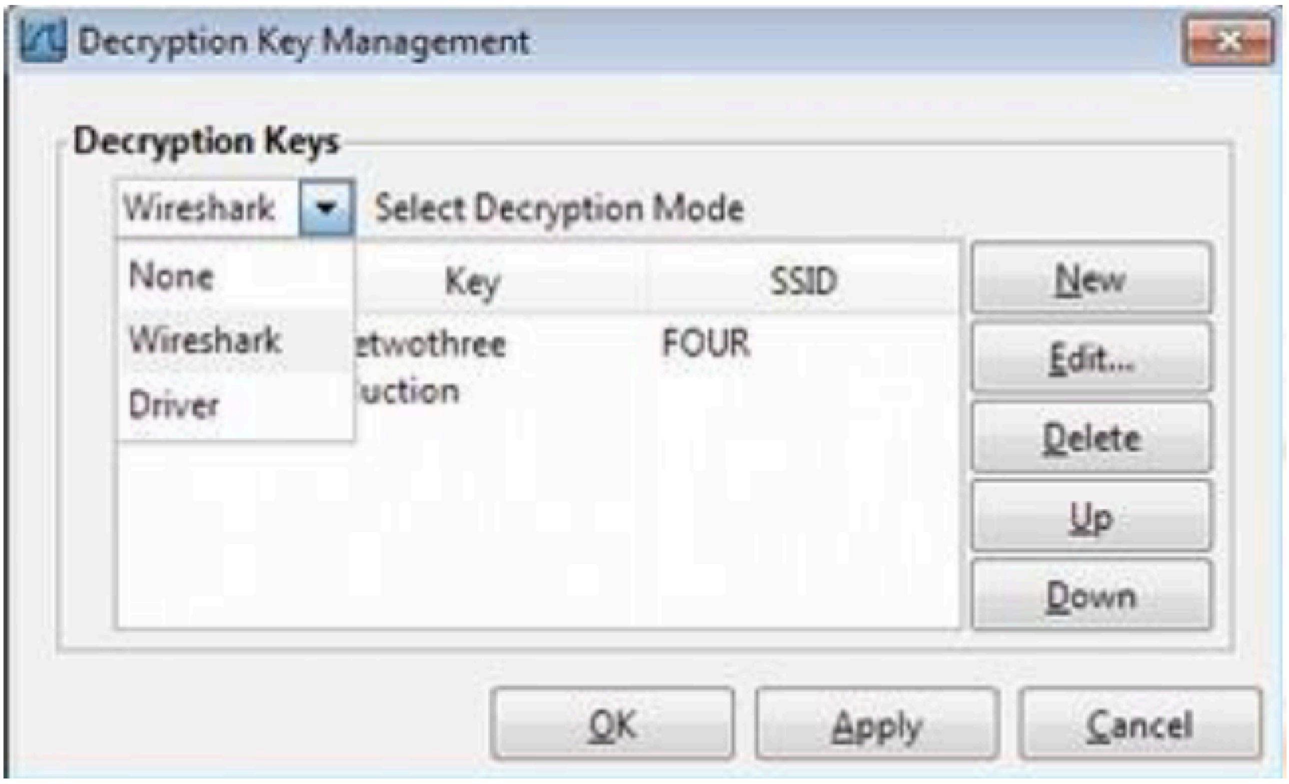
Key #1: wpa-pwd:onetwothree

Adding Keys: Wireless Toolbar

If you are using the Windows version of Wireshark and you have an AirPcap adapter you can add decryption keys using the wireless toolbar. If the toolbar isn't visible, you can show it by selecting View->Wireless Toolbar. Click on the Decryption Keys... button on the toolbar:



This will open the decryption key managment window. As shown in the window you can select between three decryption modes: None, Wireshark, and Driver:



etwothree FOUR
uction

RESULT

Thus wireless audit on an access point or a router was performed and decrypted WEP and WPA using Net Stumbler done successfully.

Ex.No: 14 CREATION OF DIGITAL SIGNATURE, SECURE DATA STORAGE, SECURE DATA TRANSMISSION USING GNUPG

AIM

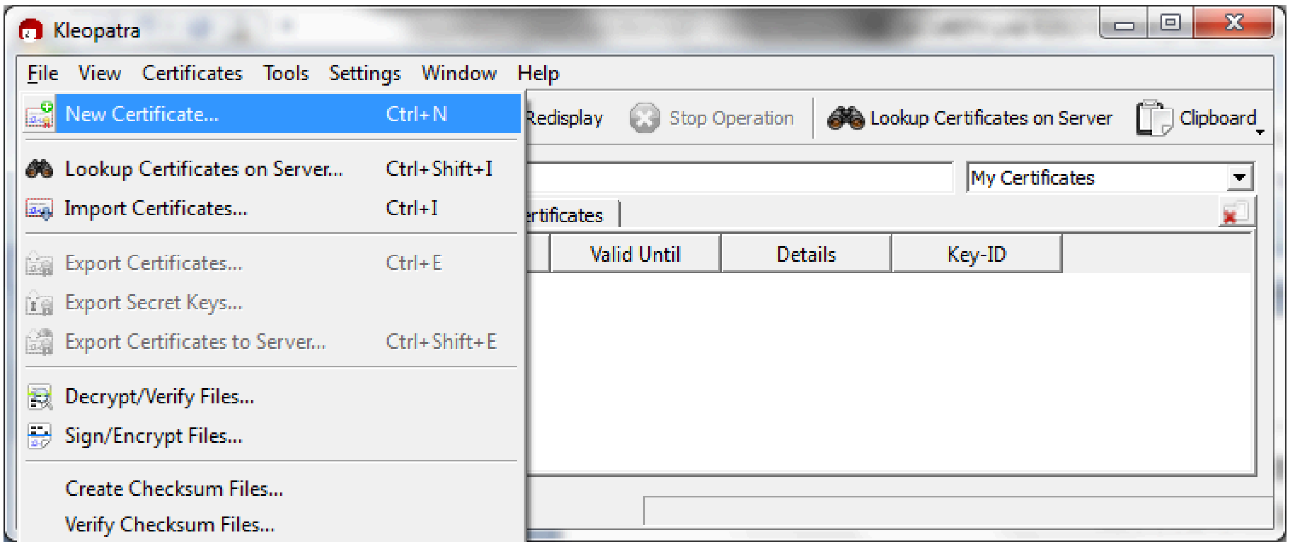
To create Digital Signature, secure Data Storage & transmission using GnuPG.

PROCEDURE

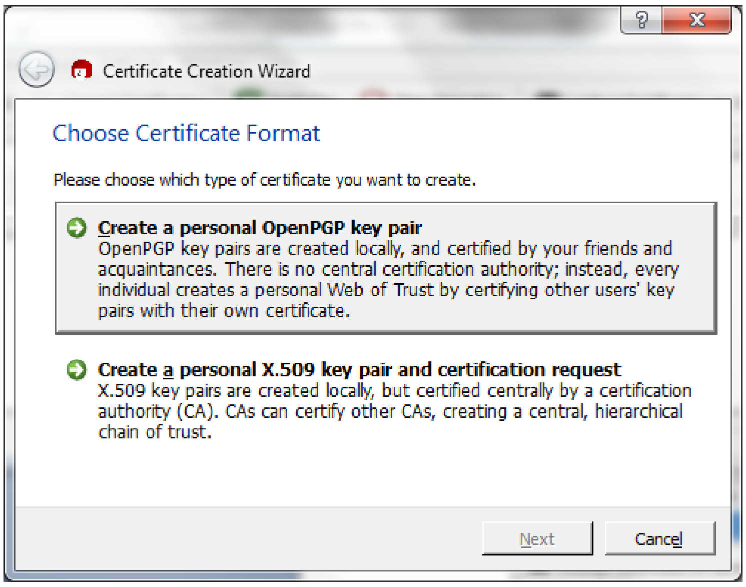
GENERATING KEYPAIR

Step 1: Open up Kleopatra.

Step 2: Go to 'File', then 'New Certificate…'


Kleopatra
File View Certificates Tools Settings Window Help
New Certificate... Ctrl+N
Lookup Certificates on Server... Ctrl+Shift+I
Import Certificates... Ctrl+I
Export Certificates... Ctrl+E
Export Secret Keys...
Export Certificates to Server... Ctrl+Shift+E
Decrypt/Verify Files...
Sign/Encrypt Files...
Create Checksum Files...
Verify Checksum Files...
Redisplay
Stop Operation
Lookup Certificates on Server
Clipboard
My Certificates
rtificates
Valid Until
Details
Key-ID

Step 3: The Certificate Creation Wizard should pop up, click on 'Create a personal OpenPGP key pair'



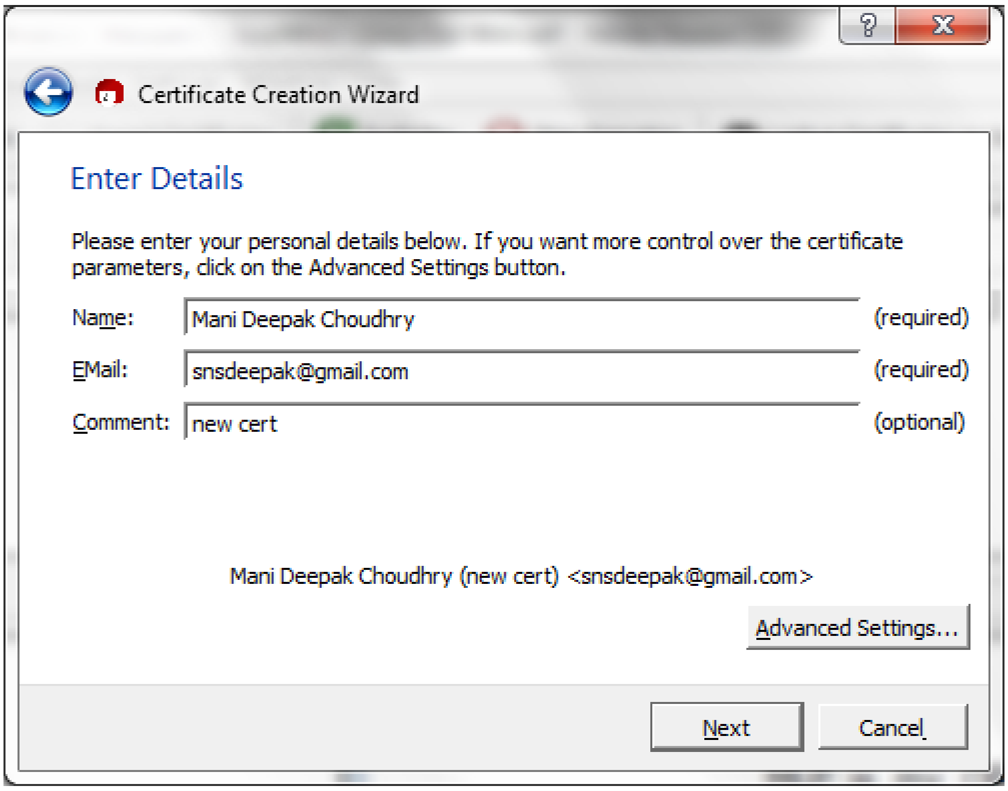
Choose Certificate Format

Please choose which type of certificate you want to create.

Create a personal OpenPGP key pair
OpenPGP key pairs are created locally, and certified by your friends and acquaintances. There is no central certification authority; instead, every individual creates a personal Web of Trust by certifying other users' key pairs with their own certificate.

Create a personal X.509 key pair and certification request
X.509 key pairs are created locally, but certified centrally by a certification authority (CA). CAs can certify other CAs, creating a central, hierarchical chain of trust.

Step 4: Now you'll enter your details. Use your marketplace username as 'Name', and fill out the rest with whatever you want. You don't need to use a real email. Check the picture for an example on how it should look.



Enter Details

Please enter your personal details below. If you want more control over the certificate parameters, click on the Advanced Settings button.

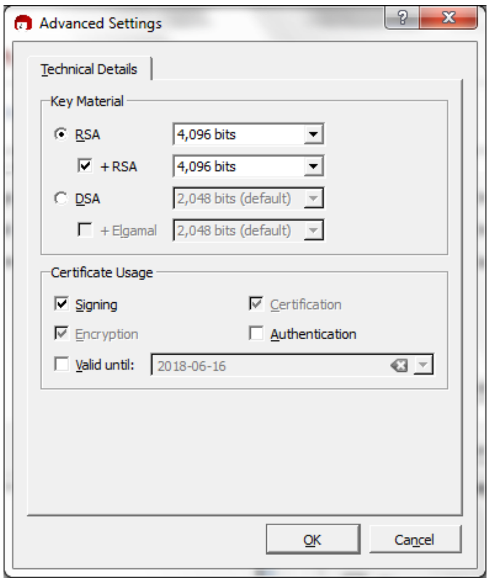
Name: Mani Deepak Choudhry (required)

EMail: snsdeepak@gmail.com (required)

Comment: new cert (optional)

Mani Deepak Choudhry (new cert) <snsdeepak@gmail.com>

Step 5: Click 'Advanced Settings…', and another window should appear. Under 'Key Material', make sure 'RSA' is checked. In the drop down menu beside it, and select '4,096 bits'. Check the picture to confirm you have everything set correctly, then click 'Ok'



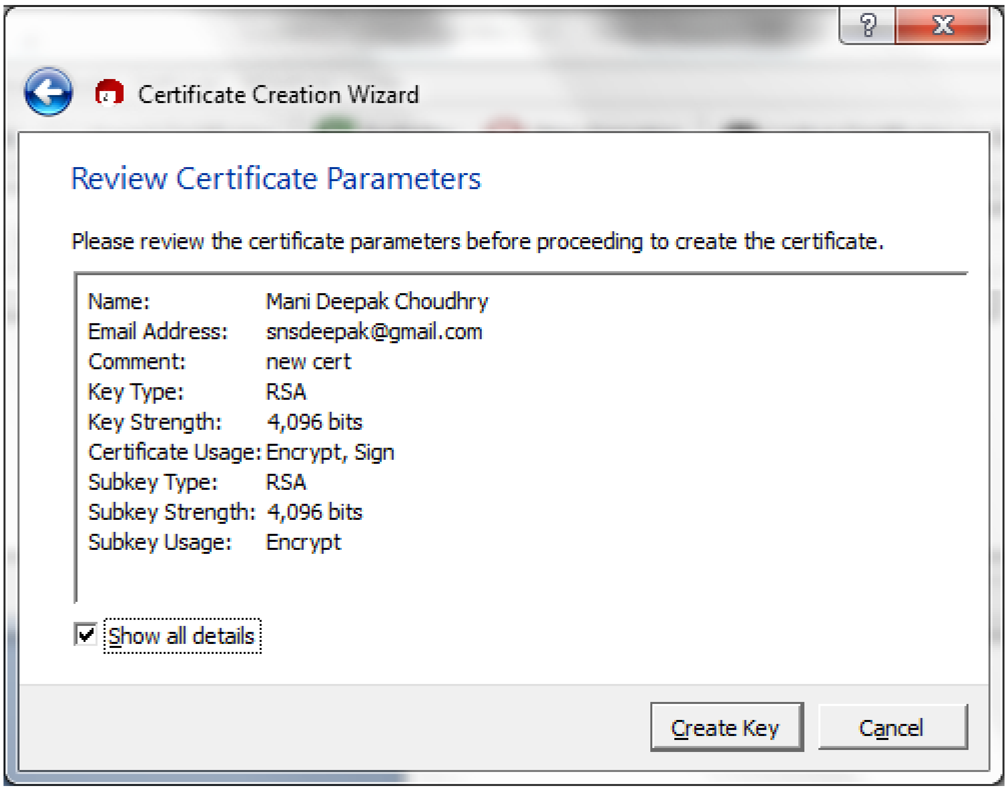
Key Material

- (●) RSA: 4,096 bits
- [✓] + RSA: 4,096 bits
- ( ) DSA: 2,048 bits (default)
- [ ] + Elgamal: 2,048 bits (default)

Certificate Usage

- [✓] Signing
- [✓] Certification
- [✓] Encryption
- [ ] Authentication
- [ ] Valid until: 2018-06-16

Step 6: Confirm you filled out all of your info correctly, then click 'Create Key'

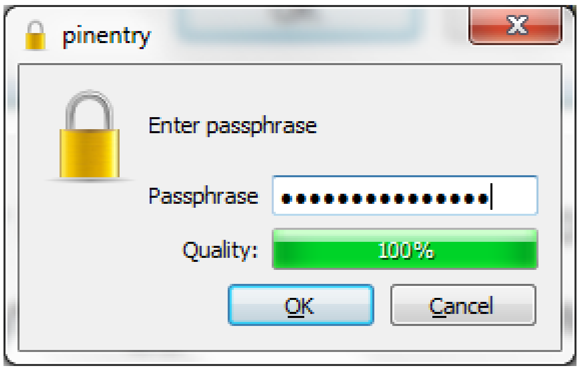

Certificate Creation Wizard

Review Certificate Parameters

Please review the certificate parameters before proceeding to create the certificate.

Name:	Mani Deepak Choudhry
Email Address:	snsdeepak@gmail.com
Comment:	new cert
Key Type:	RSA
Key Strength:	4,096 bits
Certificate Usage:	Encrypt, Sign
Subkey Type:	RSA
Subkey Strength:	4,096 bits
Subkey Usage:	Encrypt

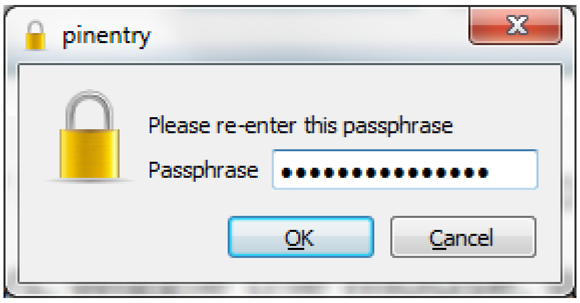
Step 7: Another window will pop up asking to enter a passphrase. Do so, then click 'Ok'


pinentry

Enter passphrase

Passphrase: ●●●●●●●●●●●●●●●

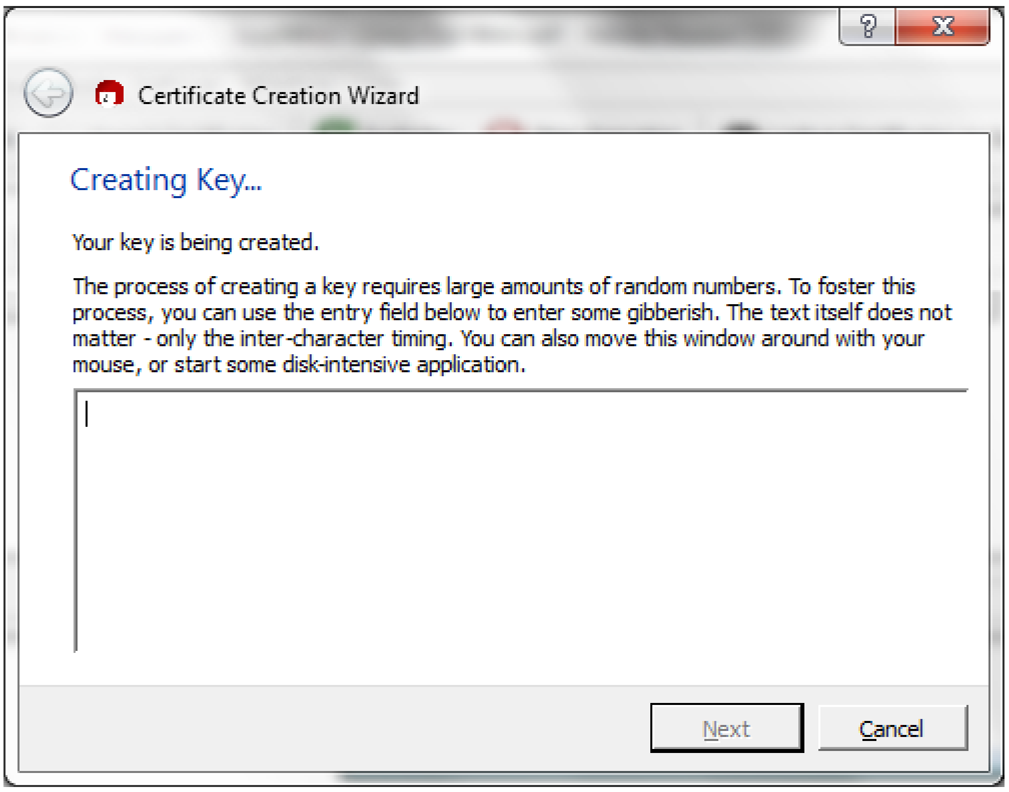
Quality: 100%


pinentry

Please re-enter this passphrase

Passphrase: ●●●●●●●●●●●●●●●

Step 8: It will now generate your key. It will need you to do random things to create entropy. Mash keys, wiggle the mouse, watch porn, download torrents, whatever

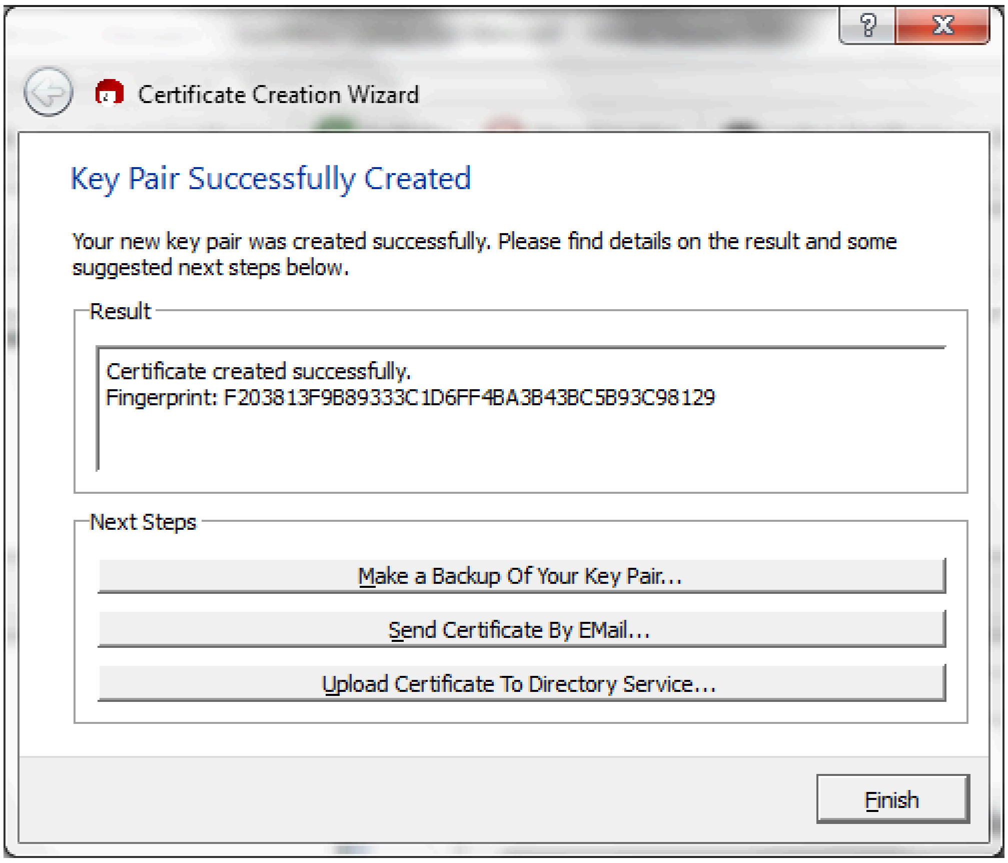


Creating Key...

Your key is being created.

The process of creating a key requires large amounts of random numbers. To foster this process, you can use the entry field below to enter some gibberish. The text itself does not matter - only the inter-character timing. You can also move this window around with your mouse, or start some disk-intensive application.

Step 9: Your key is now created. Go ahead and click 'Finish'


Certificate Creation Wizard

Key Pair Successfully Created

Your new key pair was created successfully. Please find details on the result and some suggested next steps below.

Result

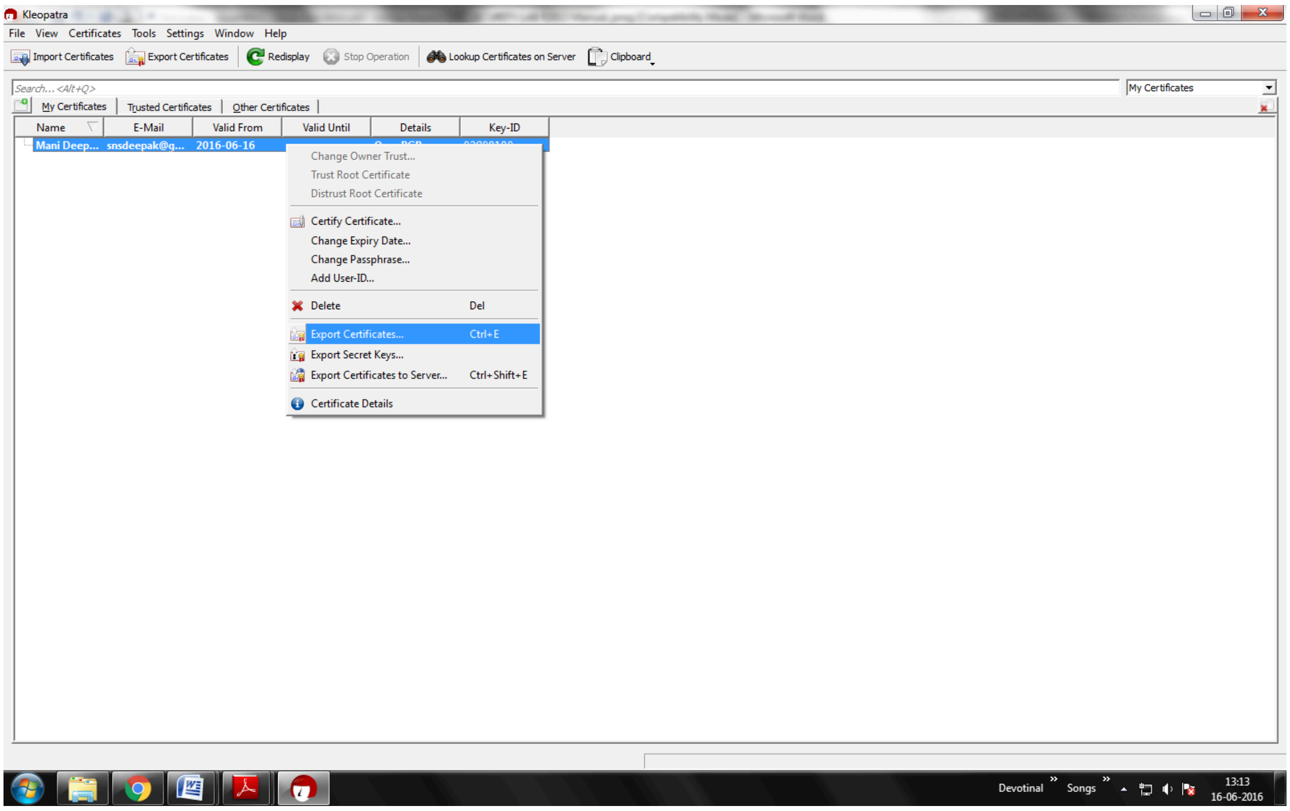
Certificate created successfully.
Fingerprint: F203813F9B89333C1D6FF4BA3B43BC5B93C98129

Next Steps

Make a Backup Of Your Key Pair...
Send Certificate By EMail...
Upload Certificate To Directory Service...
Finish

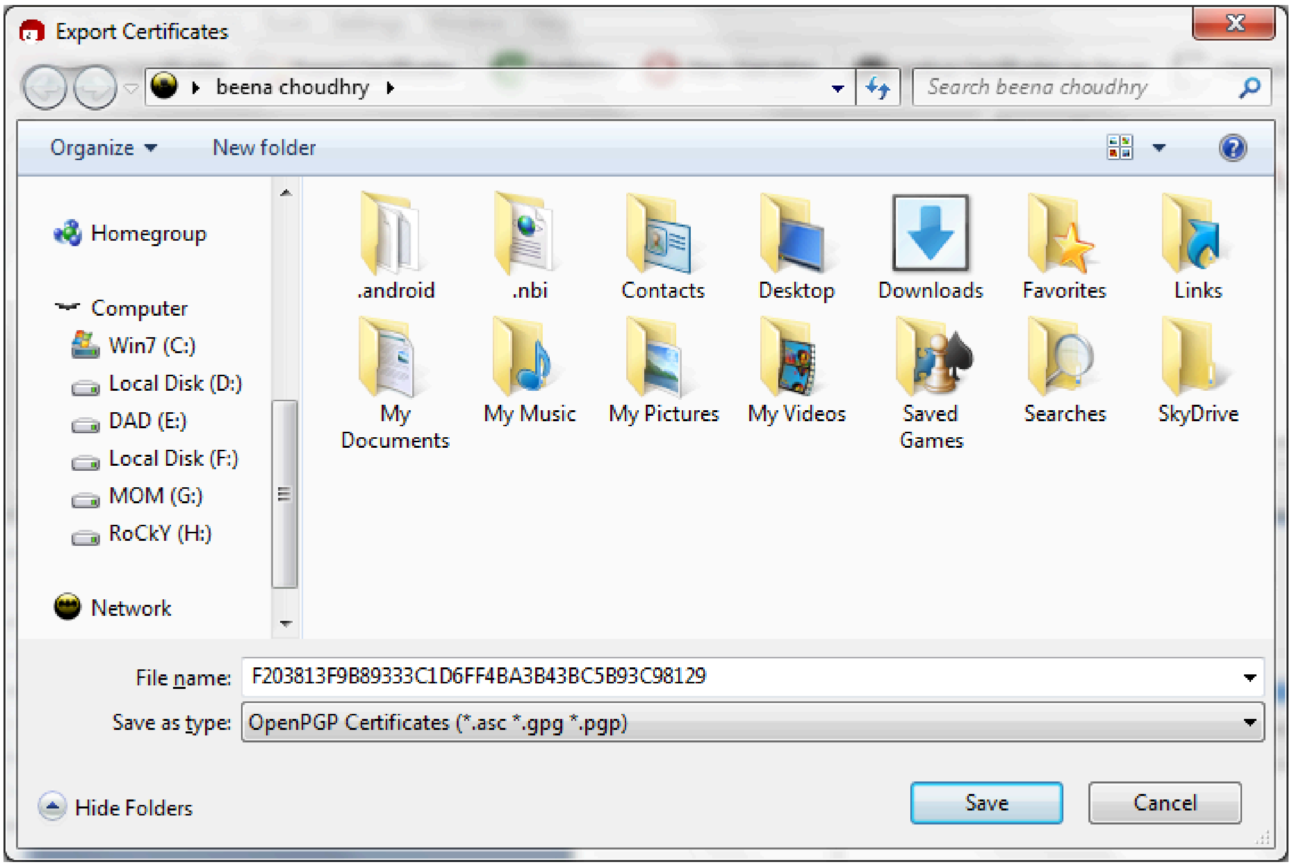
OBTAINING YOUR PUBLIC KEY

Step 1: Right click on your key, then click 'Export Certificates...'



Name	E-Mail	Valid From	Valid Until	Details	Key-ID
Mani Deep...	snsdeepak@g...	2016-06-16			

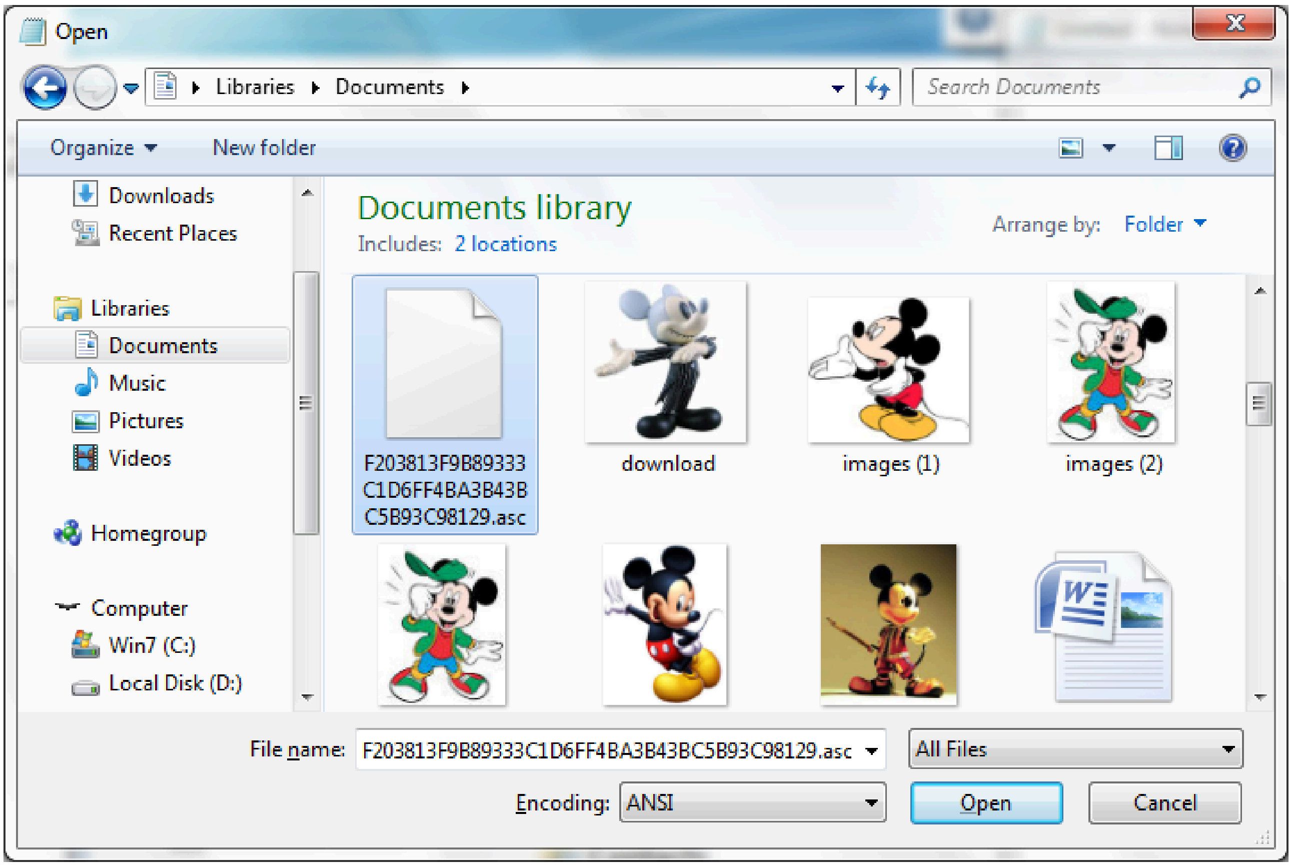
Step 2: Browse where you want to save, give it a name, then click 'Save'



File name: F203813F9B89333C1D6FF4BA3B43BC5B93C98129

Save as type: OpenPGP Certificates (*.asc *.gpg *.pgp)

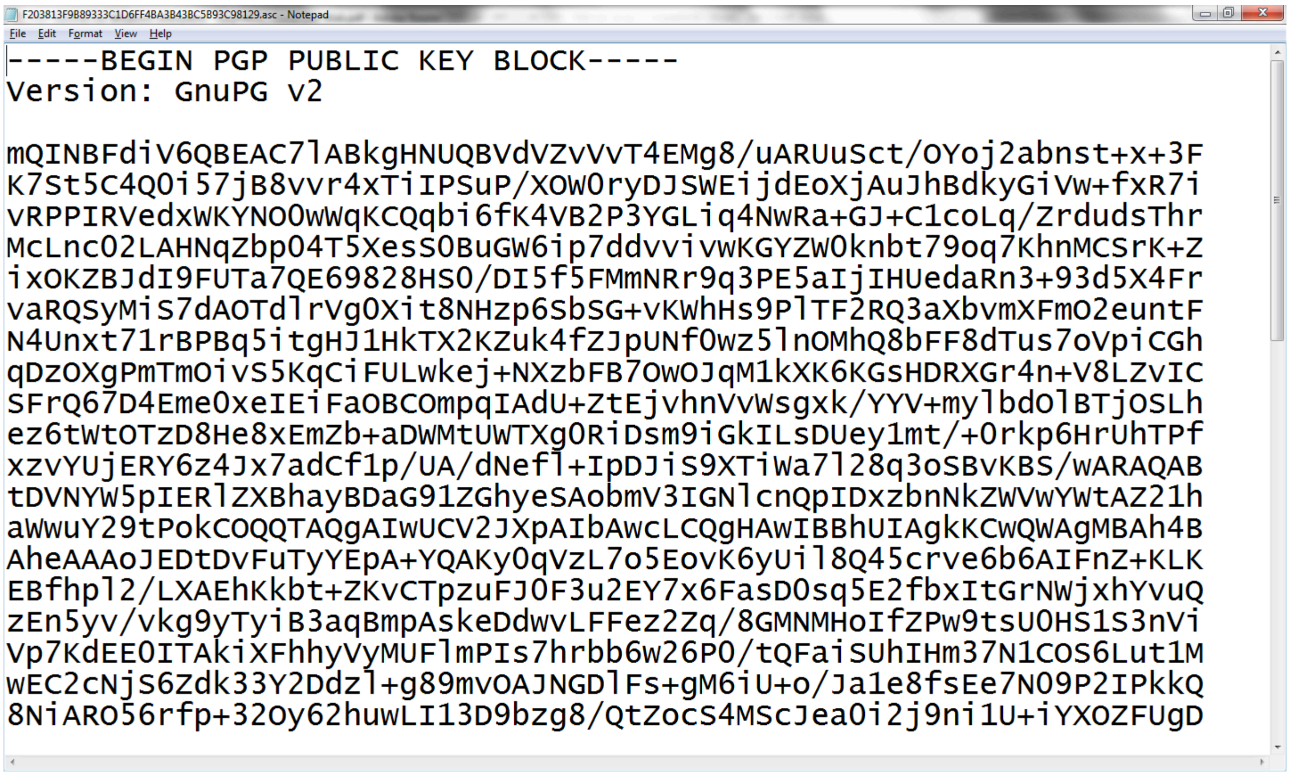
Step 3: Open your favourite text editor, browse to where the file is saved. You may have to select 'All files' from the dropdown menu. Click the file you saved, then open



File name: F203813F9B89333C1D6FF4BA3B43BC5B93C98129.asc

All Files

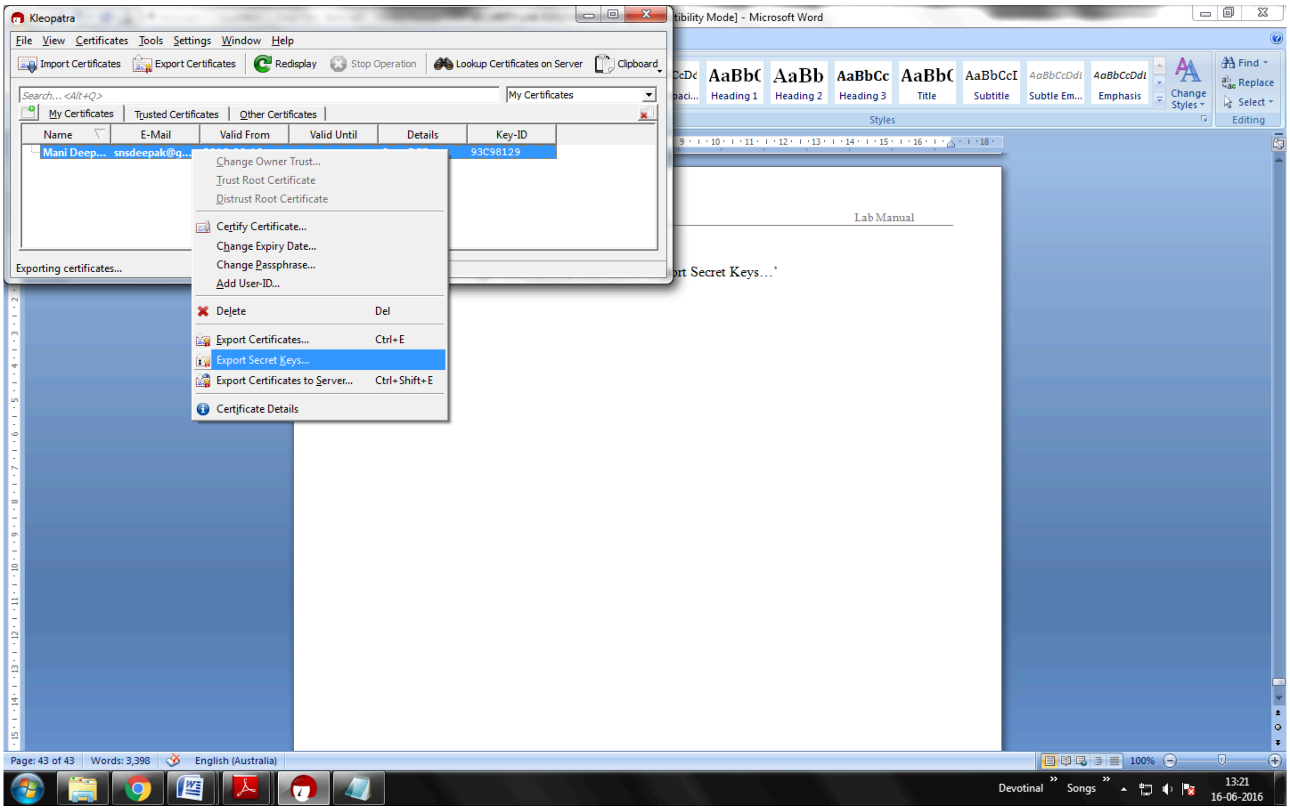
Encoding: ANSI


F203813F9B89333C1D6FF4BA3B43BC5B93C98129.asc - Notepad
File Edit Format View Help

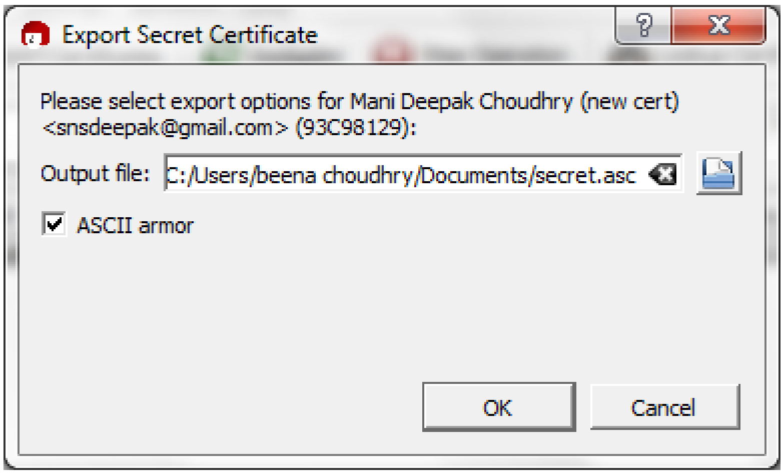
```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2

mQINBFdiV6QBEAC7lABkgHNUQBVdVZvVvT4EMg8/uARUuSct/OYoj2abnst+x+3F
K7St5C4Q0i57jB8vvr4xTiIPSuP/XOW0ryDJSWEijdEoXjAuJhBdkyGiVw+fxR7i
vRPPIRVedxWKYNO0wWqKCQqbi6fK4VB2P3YGLiq4NwRa+GJ+C1coLq/ZrdudsThr
McLnc02LAHNqZbp04T5XesS0BuGW6ip7ddvvivwKGYZW0knbt79oq7KhnMCSrK+Z
ixOKZBJdI9FUTa7QE69828HS0/DI5f5FMmNRr9q3PE5aIjIHUedaRn3+93d5X4Fr
vaRQSyMiS7dAOTdlrVg0Xit8NHzp6SbSG+vKWhHs9PlTF2RQ3aXbvmXFmO2euntF
N4Unxt71rBPBq5itgHJ1HkTX2KZuk4fZJpUNf0wz5lnOMhQ8bFF8dTus7oVpiCGh
qDzOXgPmTmOivS5KqCiFULwkej+NXzbFB7OwOJqM1kXK6KGsHDRXGr4n+V8LZvIC
SFrQ67D4EmeOxeIEiFaOBCOmpqIAdU+ZtEjvhnVvWsgxk/YYV+mylbdOlBTjOSLh
ez6tWtOTzD8He8xEmZb+aDWMtUWTXg0RiDsm9iGkILsDUey1mt/+0rkp6HrUhTPf
xzvYUjERY6z4Jx7adCf1p/UA/dNefl+IpDJiS9XTiWa7l28q3oSBvKBS/wARAQAB
tDVNYW5pIERlZXBhayBDaG91ZGhyeSAobmV3IGNlcnQpIDxzbnNkZWVwYWtAZ21h
aWwuY29tPokCOQQTAQgAIwUCV2JXpAIbAwcLCQgHAwIBBhUIAgkKCwQWAgMBAh4B
AheAAAoJEDtDvFuTyYEpA+YQAKy0qVzL7o5EovK6yUil8Q45crve6b6AIFnZ+KLK
EBfhpl2/LXAEhKkbt+ZKvCTpzuFJOF3u2EY7x6FasD0sq5E2fbxItGrNWjxhYvuQ
zEn5yv/vkg9yTyiB3aqBmpAskeDdwvLFFez2Zq/8GMNMHoIfZPw9tsUOHS1S3nVi
Vp7KdEEOITAkiXFhhyVyMUFlmPIs7hrbb6w26P0/tQFaiSUhIHm37N1COS6Lut1M
wEC2cNjS6Zdk33Y2Ddzl+g89mvOAJNGDlFs+gM6iU+o/Ja1e8fsEe7NO9P2IPkkQ
8NiARO56rfp+32Oy62huwLI13D9bzg8/QtZocS4MScJea0i2j9ni1U+iYXOZFUgD
```

OBTAINING PRIVATE KEY

Step 1: Right click on your key, select 'Export Secret Keys…'



Step 2: Select where you want it saved, give it a name, check ‘ASCII armor’, and click ‘Ok’

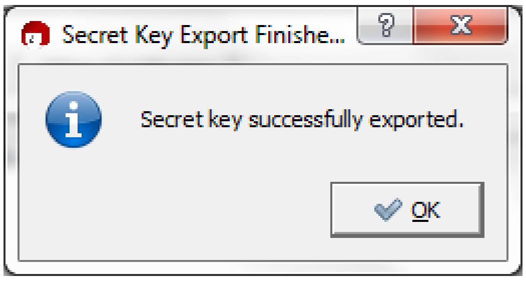


Please select export options for Mani Deepak Choudhry (new cert) <snsdeepak@gmail.com> (93C98129):

Output file: C:/Users/beena choudhry/Documents/secret.asc

☑ ASCII armor

Step 3: You now have your private key

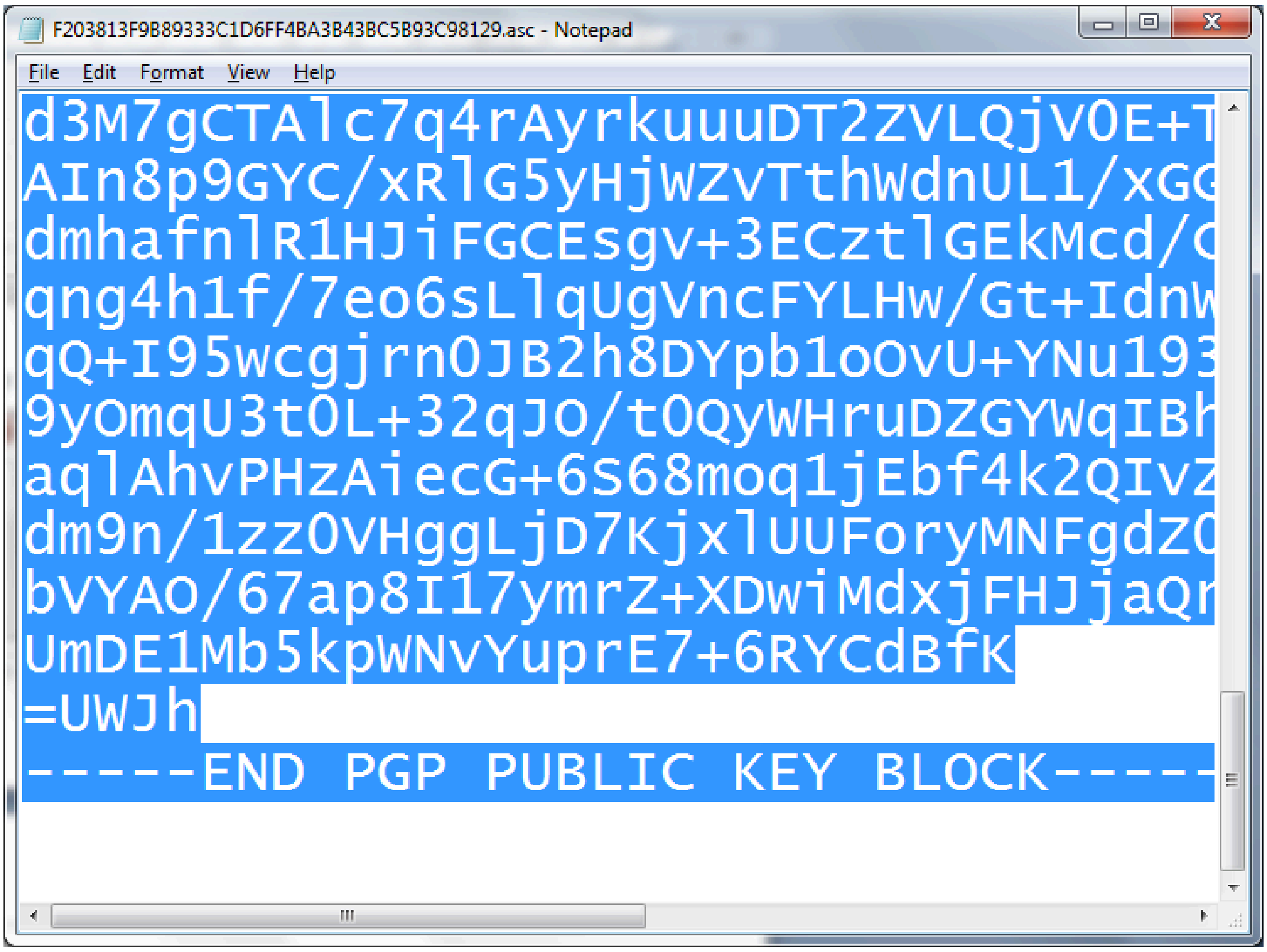


Secret key successfully exported.

IMPORTING A PUBLIC KEY

Step 1: Find a public key you want to import.

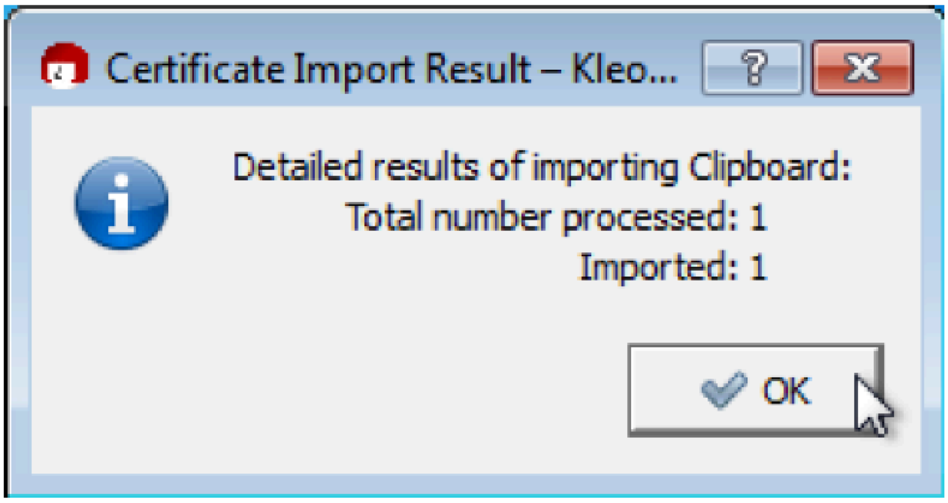
Step 2: Copy everything from ‘——BEGIN PGP PUBLIC KEY BLOCK——‘ to ‘—— END PGP PUBLIC KEY BLOCK—‘


F203813F9B89333C1D6FF4BA3B43BC5B93C98129.asc - Notepad

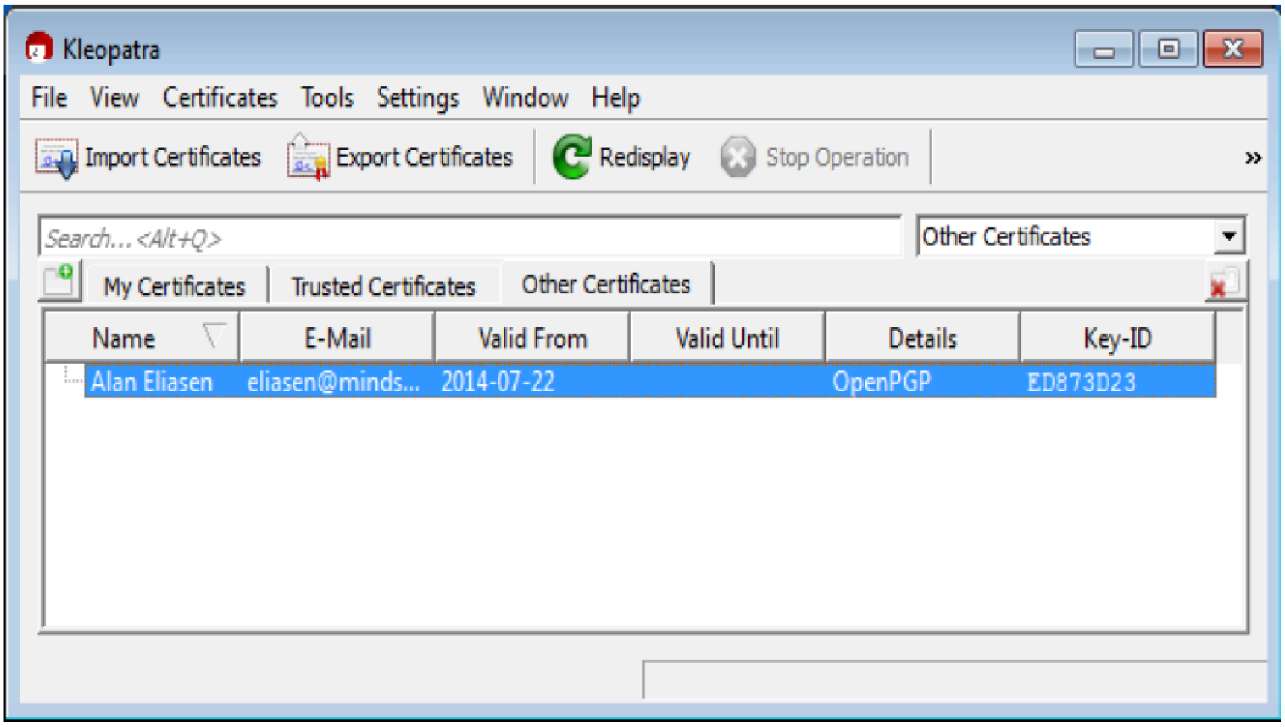
```
d3M7gCTAlc7q4rAyrkuuuDT2ZVLQjVOE+T
AIn8p9GYC/xRlG5yHjWZvTthWdnUL1/xGG
dmhafnlR1HJiFGCEsgv+3ECztlGEkMcd/C
qng4h1f/7eo6sLlqUgVncFYLHw/Gt+IdnW
qQ+I95wcgjrn0JB2h8DYpb1oOvU+YNu193
9yOmqU3t0L+32qJO/tOQyWHruDZGYWqIBh
aqlAhvPHzAiecG+6S68moq1jEbf4k2QIvZ
dm9n/1zz0VHggLjD7KjxlUUForyMNFgdZ0
bVYAO/67ap8I17ymrZ+XDwiMdxjFHJjaQr
UmDE1Mb5kpWNvYuprE7+6RYCdBfK
=UWJh
-----END PGP PUBLIC KEY BLOCK-----
```

Step 3: In your task bar, right click on the Kleopatra icon, go to 'Clipboard', then click 'Certificate Import'

Step 4: If it worked, you should see a window pop up, click 'Ok'.


Certificate Import Result – Kleo...

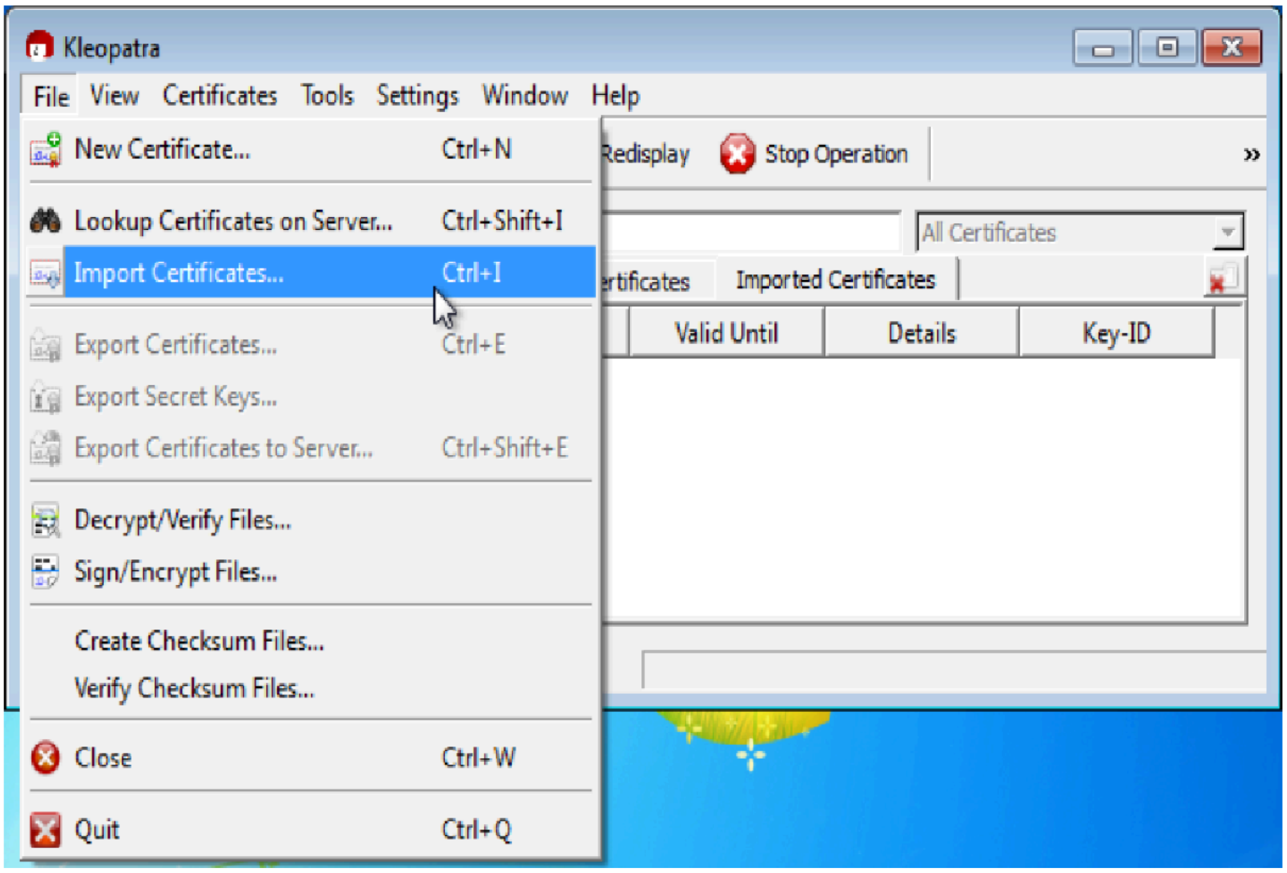
Detailed results of importing Clipboard:
Total number processed: 1
Imported: 1



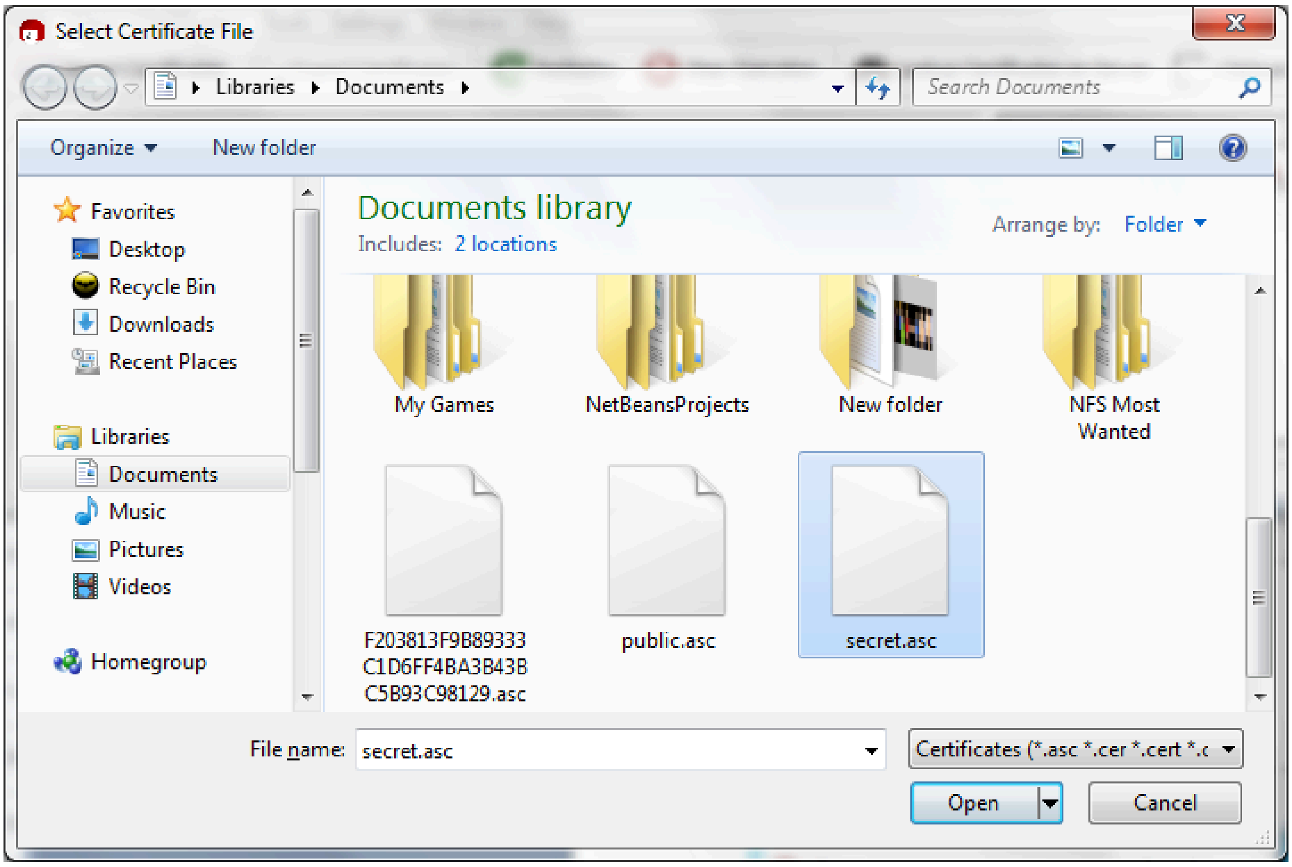
Name	E-Mail	Valid From	Valid Until	Details	Key-ID
Alan Eliasen	eliasen@minds...	2014-07-22		OpenPGP	ED873D23

IMPORTING YOUR PRIVATE KEY

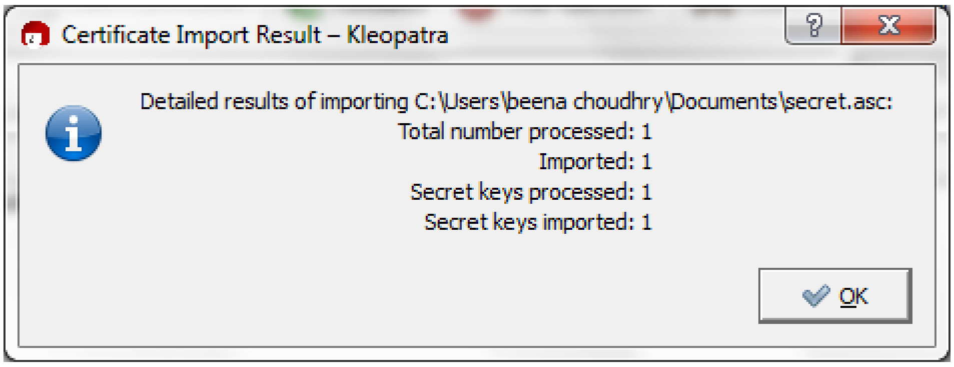
Step 1: Go to ‘File’, then click ‘Import Certificates…’



Step 2: Browse to where your private key is, select it, then click ‘Open’

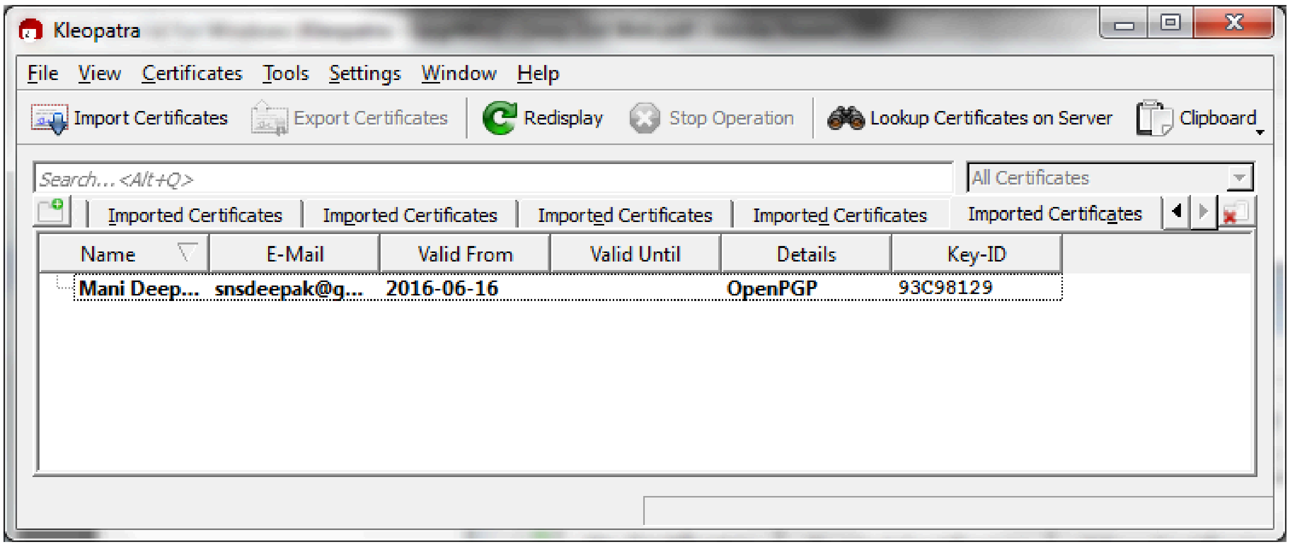


Step 3: It will import your private key, and pop up a window to confirm. Click ‘Ok’



Detailed results of importing C:\Users\beena choudhry\Documents\secret.asc:
Total number processed: 1
Imported: 1
Secret keys processed: 1
Secret keys imported: 1

Step 4: You should now see your key information under the 'My Certificates' tab

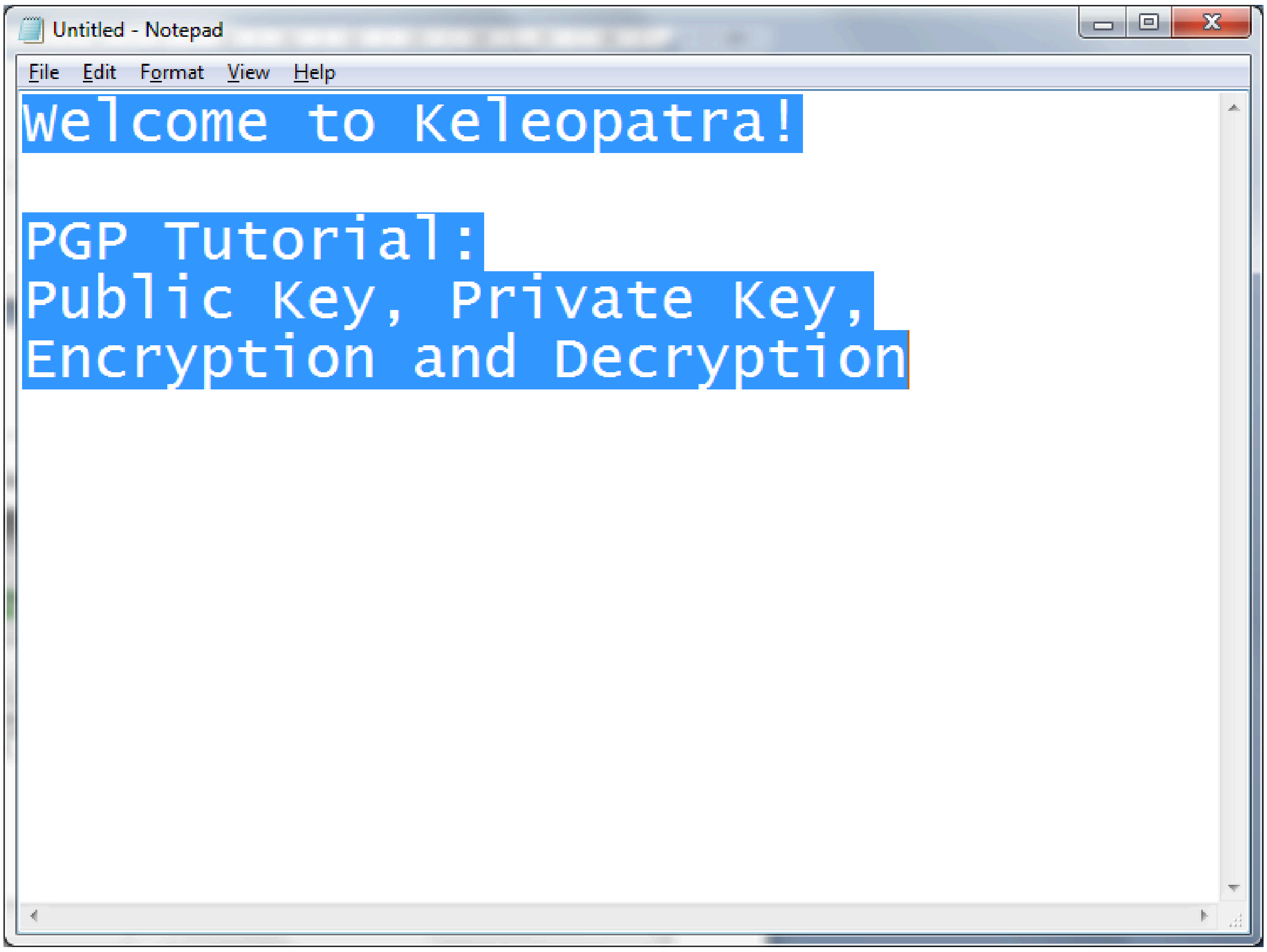

Kleopatra
File View Certificates Tools Settings Window Help
Import Certificates Export Certificates Redisplay Stop Operation Lookup Certificates on Server Clipboard
Search... <Alt+Q>
All Certificates
Imported Certificates Imported Certificates Imported Certificates Imported Certificates Imported Certificates

Name	E-Mail	Valid From	Valid Until	Details	Key-ID
Mani Deep...	snsdeepak@g...	2016-06-16		OpenPGP	93C98129

ENCRYPTING A MESSAGE

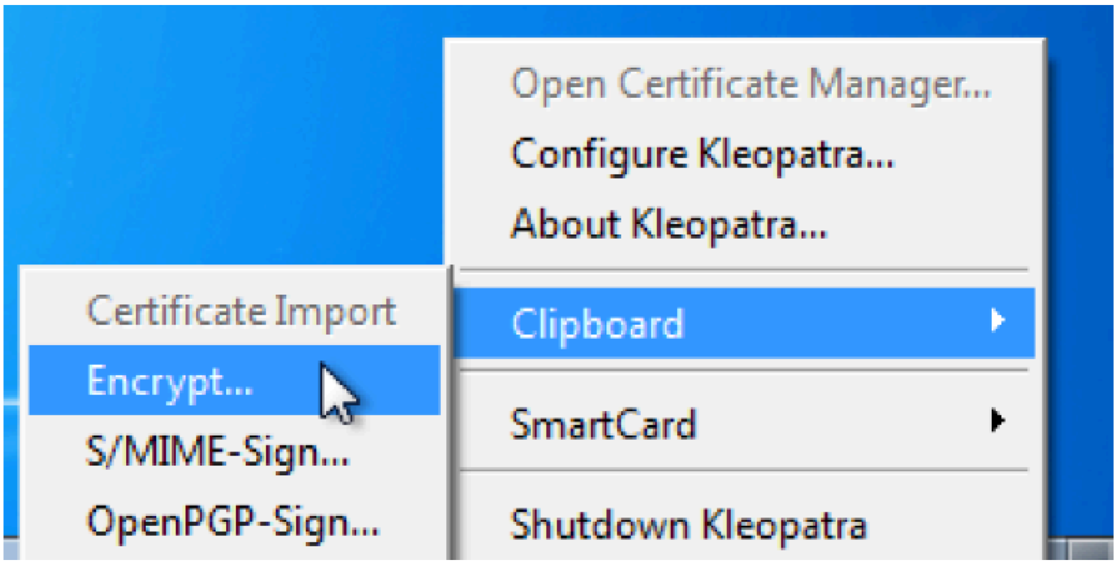
Step 1: Open up your text editor of choice.

Step 2: Type out your message, select it all, and copy it.

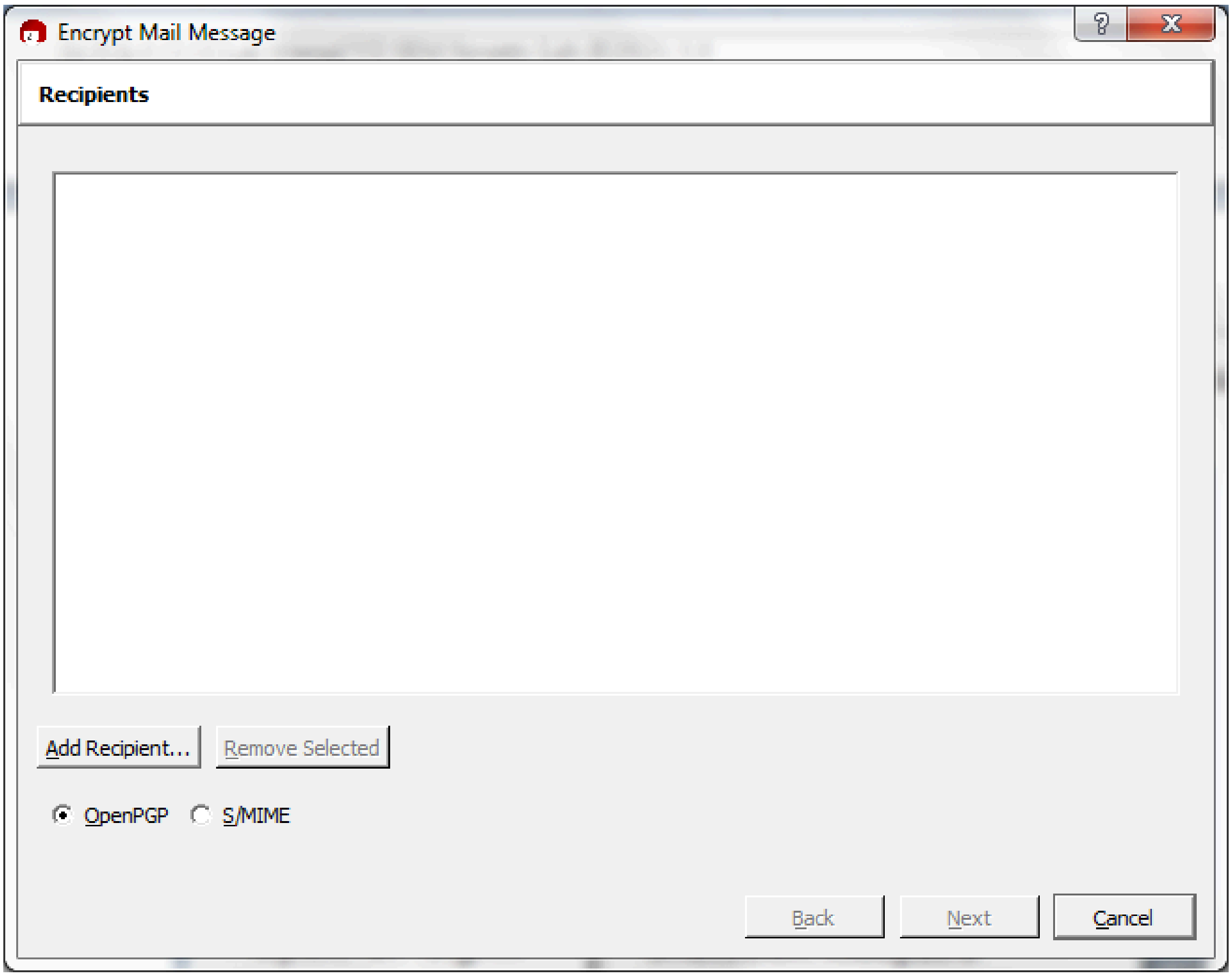


```
Welcome to Keleopatra!

PGP Tutorial:
Public Key, Private Key,
Encryption and Decryption
```

Step 3: In your task bar, right click on the Kleopatra icon, go to 'Clipboard', then click 'Encrypt…'

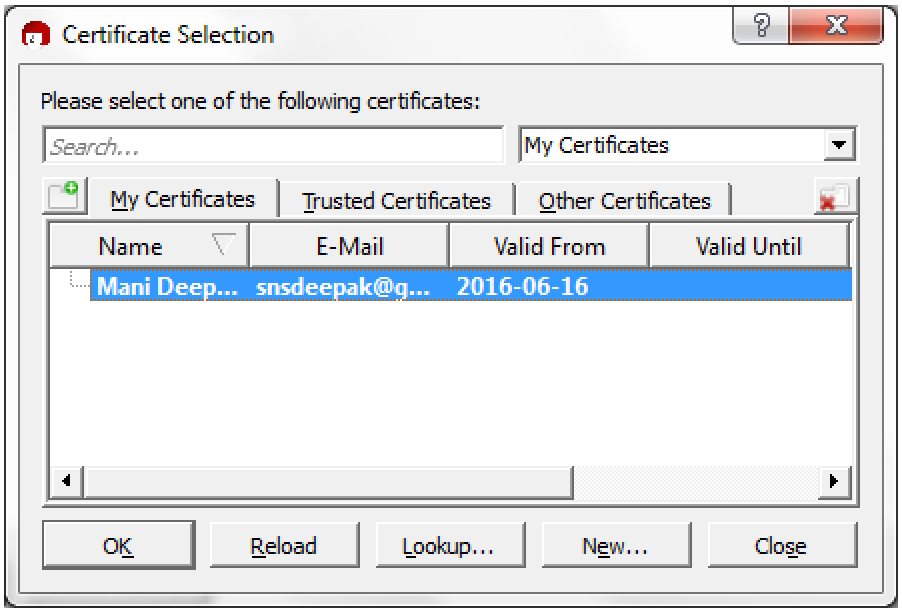


Step 4: window will open. Click 'Add Recipient…'



● OpenPGP ○ S/MIME

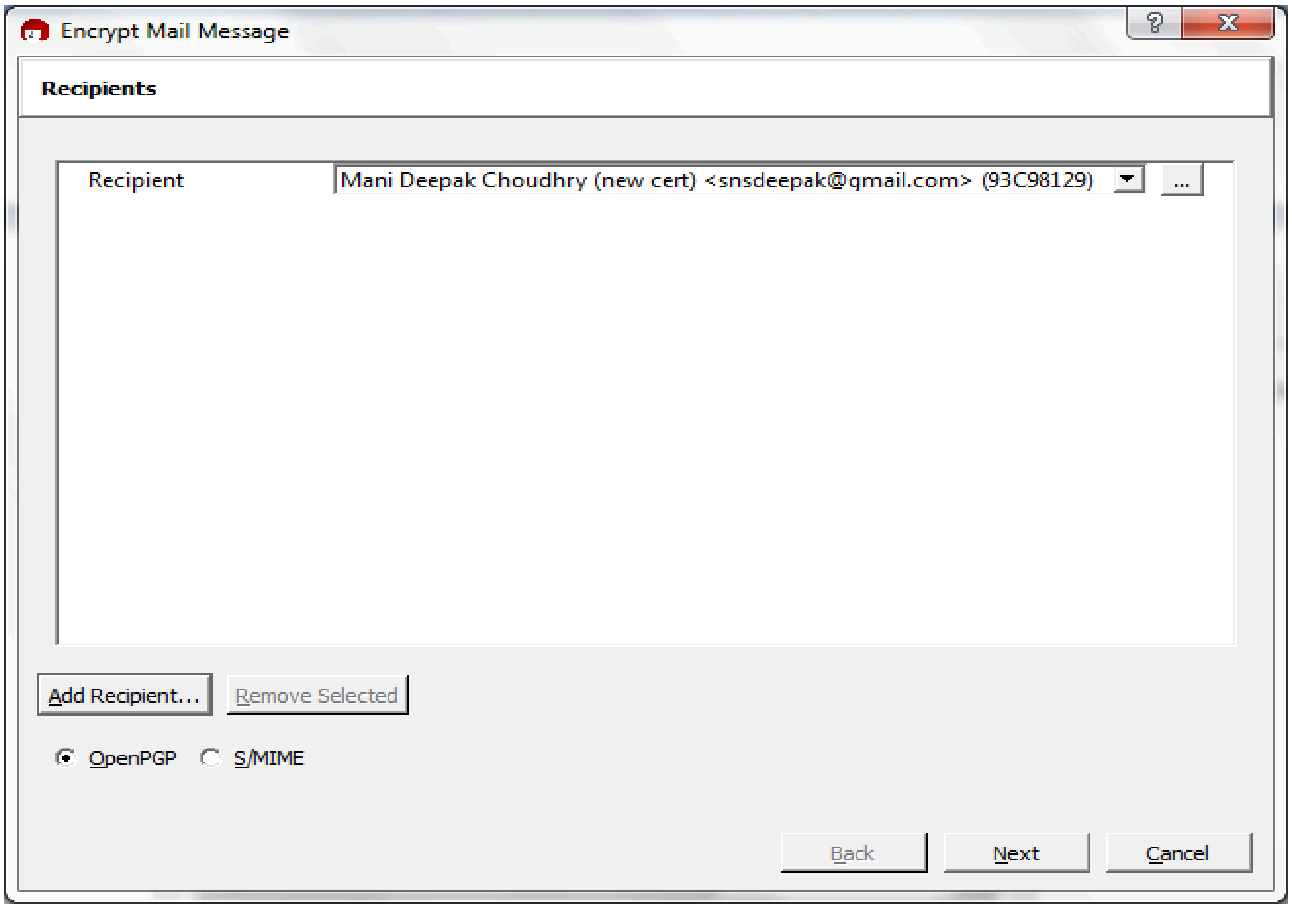
Step 5: Another window will appear. Click the 'Other Certificates' tab, then select who you want to send your message to, then click 'Ok'.



Please select one of the following certificates:

Name	E-Mail	Valid From	Valid Until
Mani Deep...	snsdeepak@g...	2016-06-16	

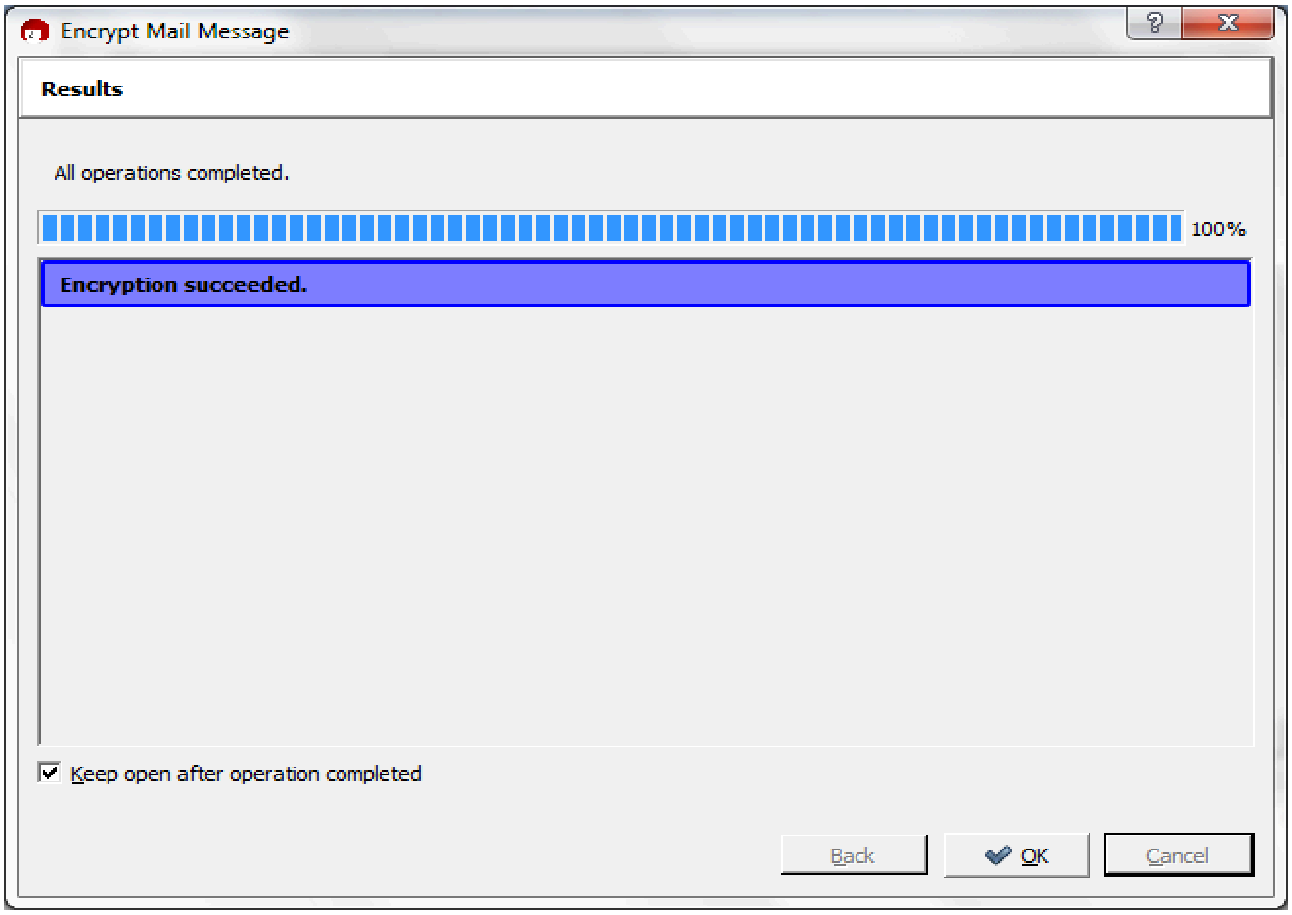
Step 6: You should be back at the previous window with the recipient listed. Click 'Next'



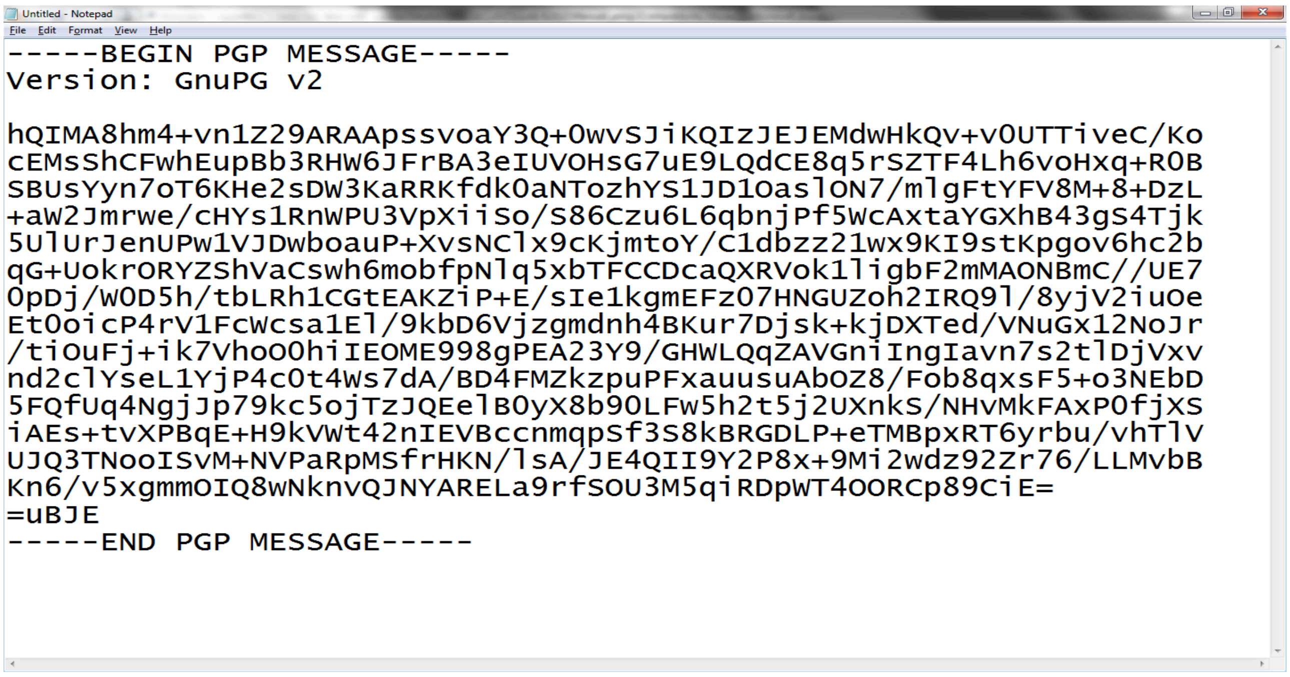
Recipient: Mani Deepak Choudhry (new cert) <snsdeepak@gmail.com> (93C98129)

[✓ OpenPGP] S/MIME

Step 7: If all went well, you should see this window. Click 'Ok'



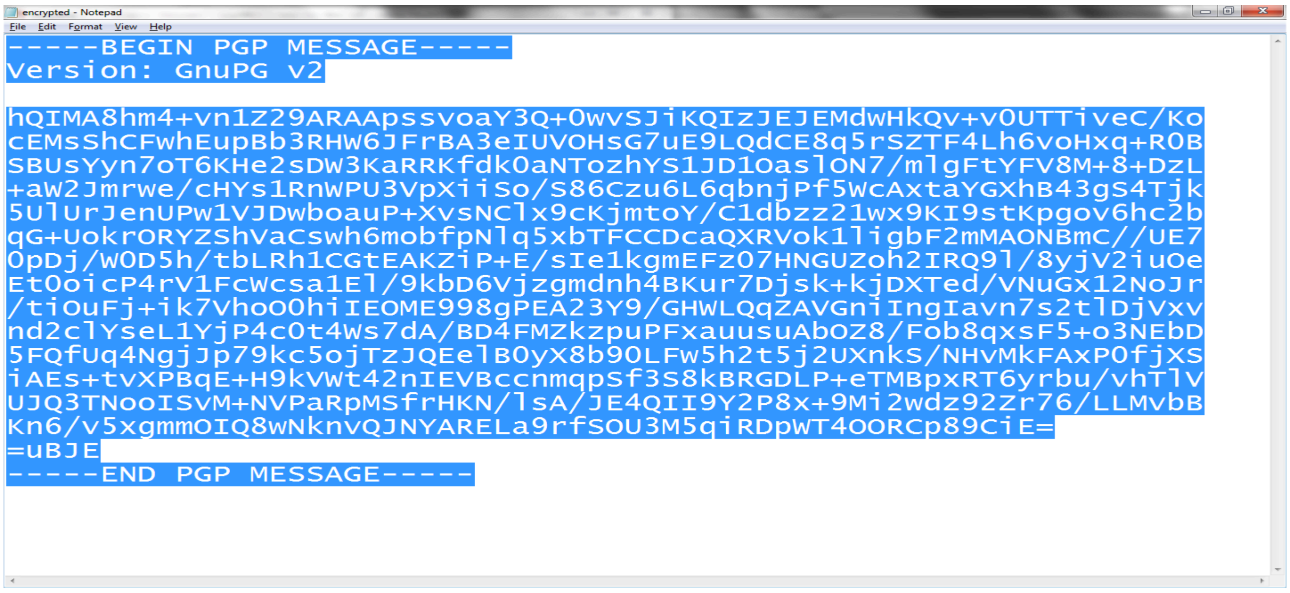
Step 8: Your encrypted message will be in your clipboard, all you need to do is paste it into the message box and send



```
-----BEGIN PGP MESSAGE-----
Version: GnuPG v2

hQIMA8hm4+vn1Z29ARAApssvoaY3Q+0wvSJiKQIzJEJEMdwHkQv+v0UTTiveC/Ko
cEMsShCFwhEupBb3RHW6JFrBA3eIUVOHsG7uE9LQdCE8q5rSZTF4Lh6voHxq+R0B
SBUsYyn7oT6KHe2sDW3KaRRKfdk0aNTozhYS1JD1Oas1ON7/m1gFtYFV8M+8+DzL
+aW2Jmrwe/cHYs1RnWPU3VpXiiSo/S86Czu6L6qbnjPf5WcAxtaYGXhB43gS4Tjk
5UlUrJenUPw1VJDwboauP+XvsNClx9cKjmtoY/C1dbzz21wx9KI9stKpgov6hc2b
qG+UokrORYZShVaCswh6mobfpNlq5xbTFCCDcaQXRVok1ligbF2mMAONBmC//UE7
0pDj/W0D5h/tbLRh1CGtEAKZiP+E/sIe1kgmEFz07HNGUZoh2IRQ9l/8yjV2iuOe
Et0oicP4rV1FcWcsa1El/9kbD6Vjzgmdnh4BKur7Djsk+kjDXTed/VNuGx12NoJr
/tiOuFj+ik7VhoO0hiIEOME998gPEA23Y9/GHWLQqZAVGniIngIavn7s2tlDjVxv
nd2clYseL1YjP4c0t4Ws7dA/BD4FMZkzpuPFxauusAbOZ8/Fob8qxsF5+o3NEbD
5FQfUq4NgjJp79kc5ojTzJQEelB0yX8b90LFw5h2t5j2UXnkS/NHvMkFAxP0fjXS
iAEs+tvXPBqE+H9kVWt42nIEVBccnmqpSf3S8kBRGDLP+eTMBpxRT6yrbu/vhTlV
UJQ3TNooISvM+NVPaRpMSfrHKN/lsA/JE4QII9Y2P8x+9Mi2wdz92Zr76/LLMvbB
Kn6/v5xgmmOIQ8wNknvQJNYAREla9rfSOU3M5qiRDpWT4OORCp89CiE=
=uBJE
-----END PGP MESSAGE-----
```

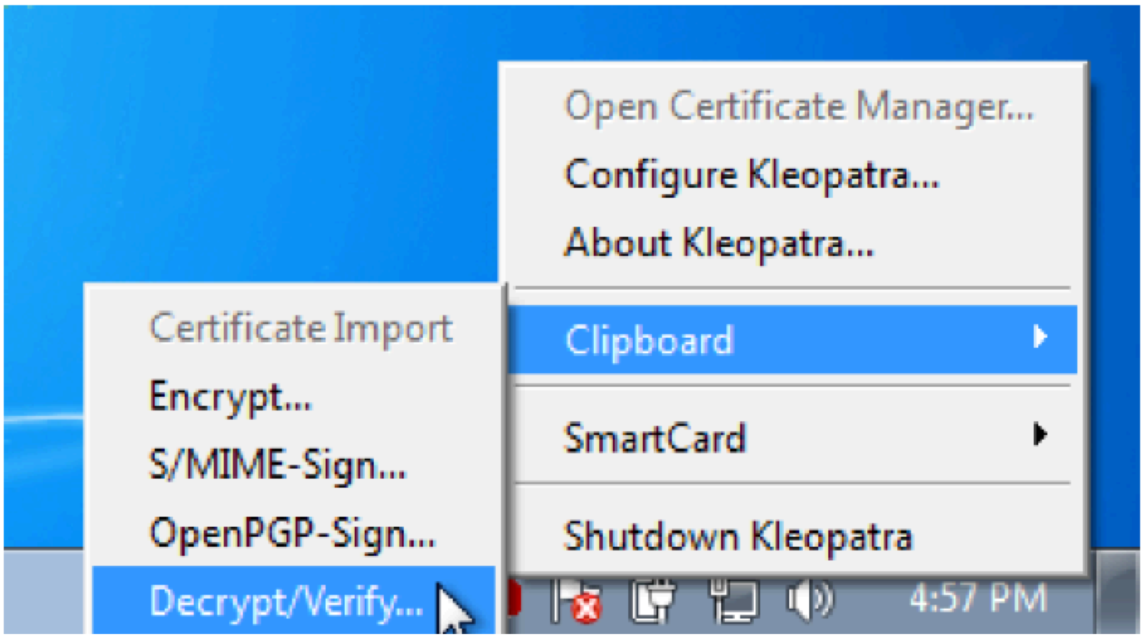
DECRYPTING A MESSAGE

Step 1: Copy everything that was sent.

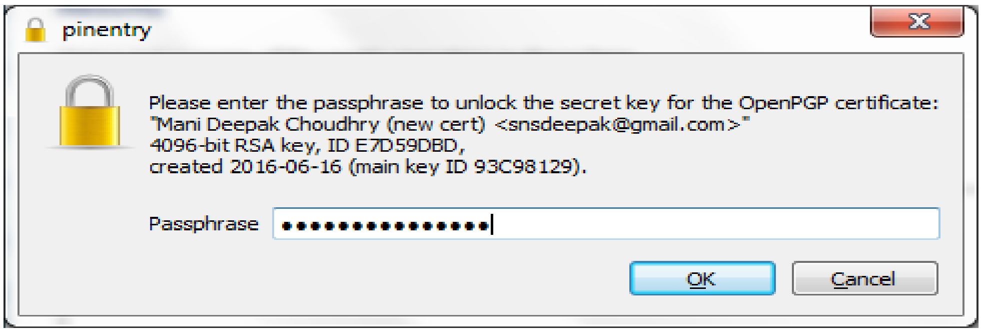


```
-----BEGIN PGP MESSAGE-----
Version: GnuPG v2

hQIMA8hm4+vn1Z29ARAApssvoaY3Q+0wvSJiKQIzJEJEMdwHkQv+vOUTTiveC/Ko
cEMsShCFwhEupBb3RHW6JFrBA3eIUVOHsG7uE9LQdCE8q5rSZTF4Lh6voHxq+R0B
SBUsYyn7oT6KHe2sDW3KaRRKfdk0aNTozhYS1JD1Oas1ON7/m1gFtYFV8M+8+DzL
+aW2Jmrwe/cHYs1RnWPU3VpXiiSo/S86Czu6L6qbnjPf5WcAxtaYGXhB43gS4Tjk
5UlUrJenUPw1VJDwboauP+XvsNClx9cKjmtoY/C1dbzz21wx9KI9stKpgov6hc2b
qG+UokrORYZShVaCswh6mobfpNlq5xbTFCCDcaQXRVok1ligbF2mMAONBmC//UE7
0pDj/W0D5h/tbLRh1CGtEAKZiP+E/sIe1kgmEFz07HNGUZoh2IRQ9l/8yjV2iuOe
Et0oicP4rV1FcWcsa1El/9kbD6Vjzgmdnh4BKur7Djsk+kjDXTed/VNuGx12NoJr
/tiOuFj+ik7VhoO0hiIEOME998gPEA23Y9/GHWLQqZAVGniIngIavn7s2tlDjVxv
nd2clYseL1YjP4c0t4Ws7dA/BD4FMZkzpuPFxauusAbOZ8/Fob8qxsF5+o3NEbD
5FQfUq4NgjJp79kc5ojTzJQEelB0yX8b90LFw5h2t5j2UXnkS/NHvMkFAxP0fjXS
iAEs+tvXPBqE+H9kVWt42nIEVBccnmqpSf3S8kBRGDLP+eTMBpxRT6yrbu/vhTlV
UJQ3TNooISvM+NVPaRpMSfrHKN/lsA/JE4QII9Y2P8x+9Mi2wdz92Zr76/LLMvbB
Kn6/v5xgmmOIQ8wNknvQJNYARELa9rfSOU3M5qiRDpWT4OORCp89CiE=
=uBJE
-----END PGP MESSAGE-----
```

Step 2: In your task bar, right click on the Kleopatra icon, go to 'Clipboard', then click 'Decrypt/Verify…'

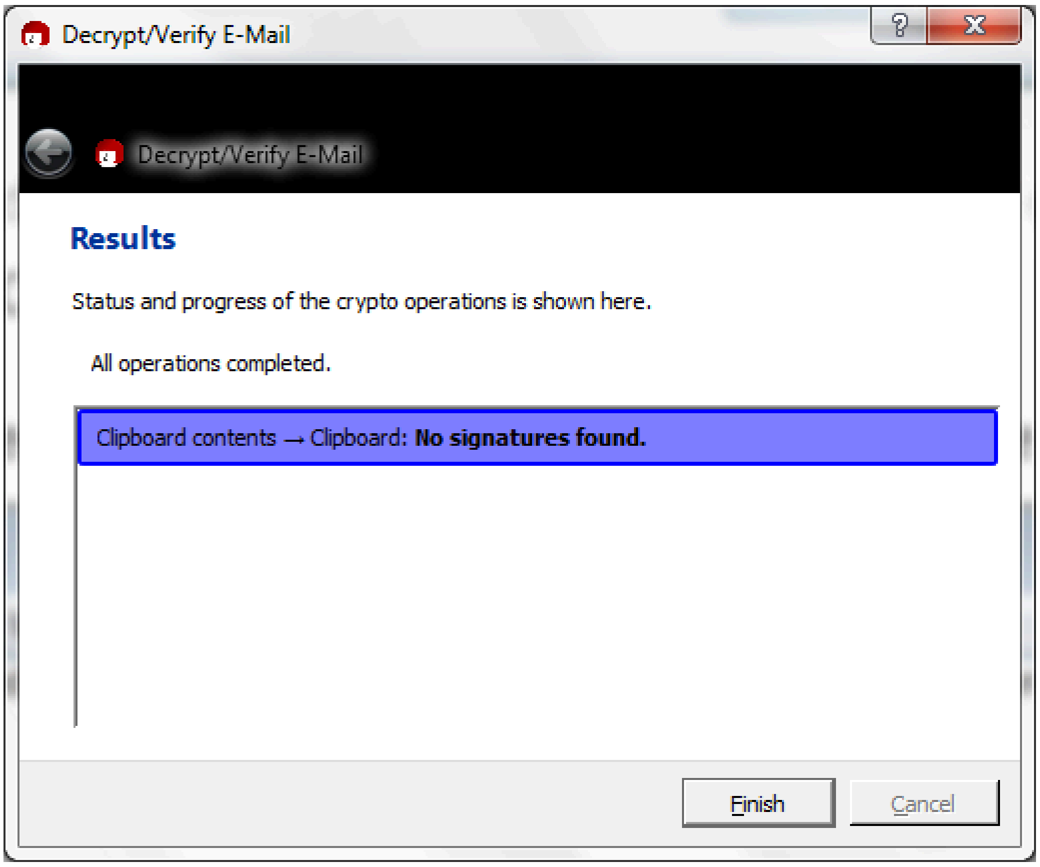


Step 3: A window will pop up asking for your passphrase, enter that then click 'Ok'.



Please enter the passphrase to unlock the secret key for the OpenPGP certificate:
"Mani Deepak Choudhry (new cert) <snsdeepak@gmail.com>"
4096-bit RSA key, ID E7D59DBD,
created 2016-06-16 (main key ID 93C98129).

Passphrase ••••••••••••••••

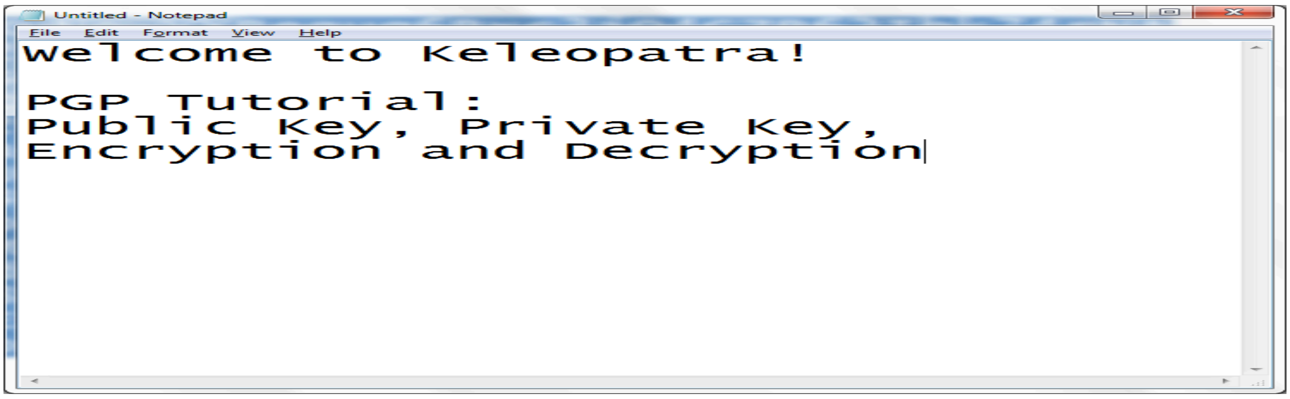


Results

Status and progress of the crypto operations is shown here.

All operations completed.

Clipboard contents → Clipboard: **No signatures found.**



```
Welcome to Keleopatra!

PGP Tutorial:
Public Key, Private Key,
Encryption and Decryption
```

RESULT

Thus creation of Digital Signature, secure data stirage and transmission was done using Kleopatra Tool using GnuPG was done and output is verified successfully.

Ex.No: 15 SETUP A HONEY POT AND MONITOR THE HONEYPOT ON NETWORK USING KF SENSOR

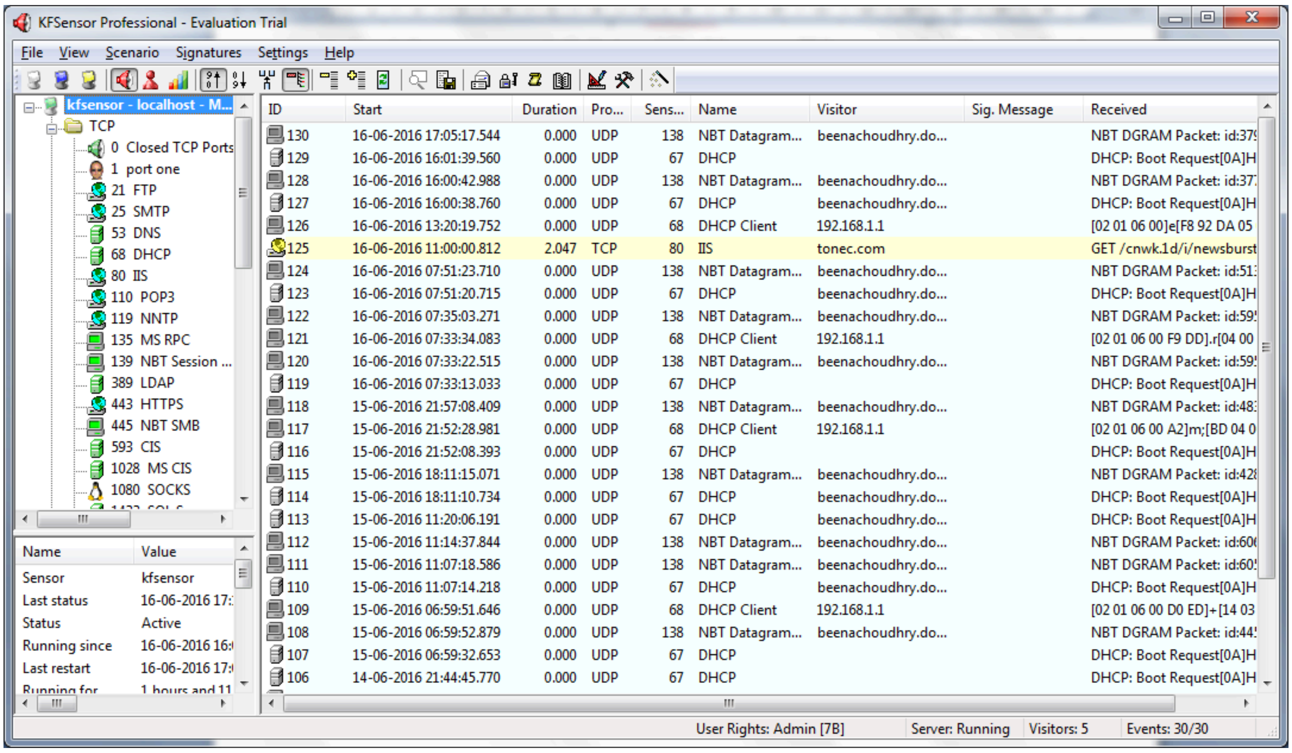
AIM

To setup a honey pot and monitor the honey pot on network using KF Sensor.

PROCEDURE

Honey Pot is a device placed on Computer Network specifically designed to capture malicious network traffic.

KF Sensor is the tool to setup as honeypot when KF Sensor is running it places a siren icon in the windows system tray in the bottom right of the screen. If there are no alerts then green icon is displayed.

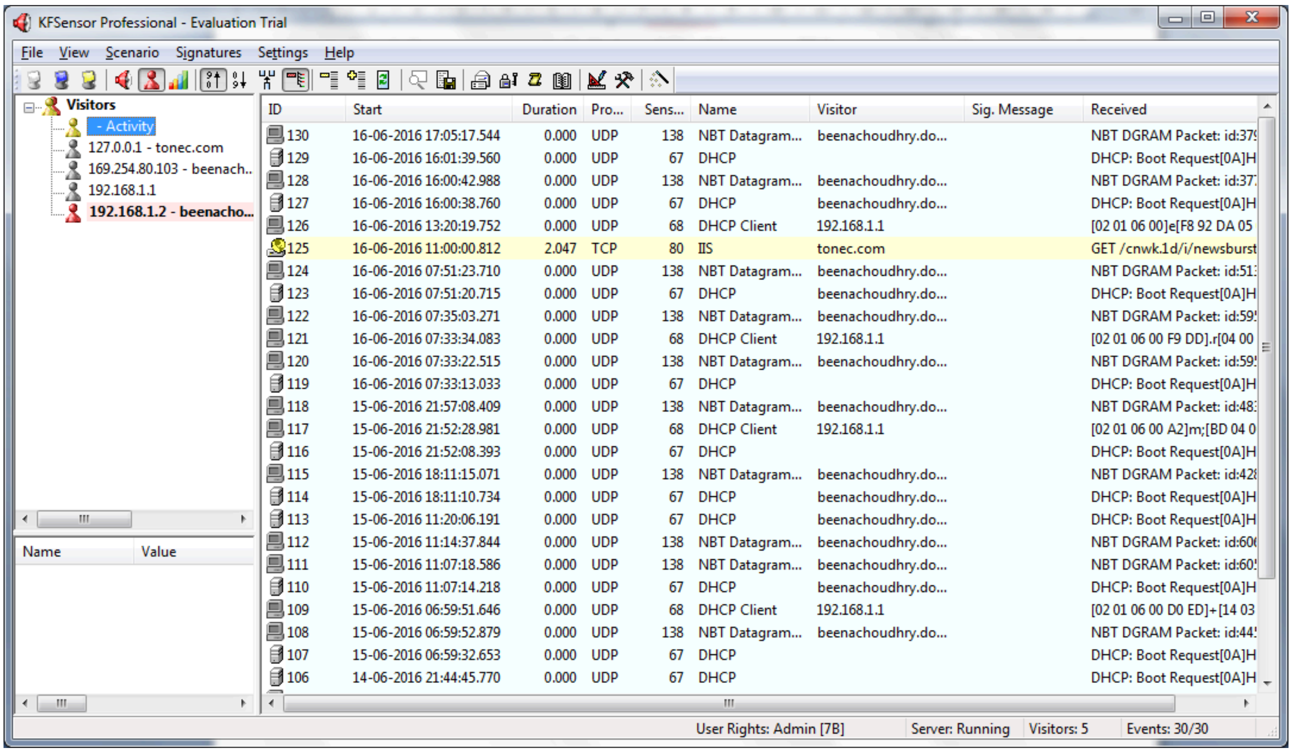

KFSensor Professional - Evaluation Trial
File View Scenario Signatures Settings Help

kfsensor - localhost - M...
- TCP
  - 0 Closed TCP Ports
  - 1 port one
  - 21 FTP
  - 25 SMTP
  - 53 DNS
  - 68 DHCP
  - 80 IIS
  - 110 POP3
  - 119 NNTP
  - 135 MS RPC
  - 139 NBT Session ...
  - 389 LDAP
  - 443 HTTPS
  - 445 NBT SMB
  - 593 CIS
  - 1028 MS CIS
  - 1080 SOCKS

Name	Value
Sensor	kfsensor
Last status	16-06-2016 17:
Status	Active
Running since	16-06-2016 16:
Last restart	16-06-2016 17:
Running for	1 hours and 11

ID	Start	Duration	Pro...	Sens...	Name	Visitor	Sig. Message	Received
130	16-06-2016 17:05:17.544	0.000	UDP	138	NBT Datagram...	beenachoudhry.do...		NBT DGRAM Packet: id:37
129	16-06-2016 16:01:39.560	0.000	UDP	67	DHCP			DHCP: Boot Request[0A]H
128	16-06-2016 16:00:42.988	0.000	UDP	138	NBT Datagram...	beenachoudhry.do...		NBT DGRAM Packet: id:37
127	16-06-2016 16:00:38.760	0.000	UDP	67	DHCP	beenachoudhry.do...		DHCP: Boot Request[0A]H
126	16-06-2016 13:20:19.752	0.000	UDP	68	DHCP Client	192.168.1.1		[02 01 06 00]e[F8 92 DA 05
125	16-06-2016 11:00:00.812	2.047	TCP	80	IIS	tonec.com		GET /cnwk.1d/i/newsburst
124	16-06-2016 07:51:23.710	0.000	UDP	138	NBT Datagram...	beenachoudhry.do...		NBT DGRAM Packet: id:51
123	16-06-2016 07:51:20.715	0.000	UDP	67	DHCP	beenachoudhry.do...		DHCP: Boot Request[0A]H
122	16-06-2016 07:35:03.271	0.000	UDP	138	NBT Datagram...	beenachoudhry.do...		NBT DGRAM Packet: id:59
121	16-06-2016 07:33:34.083	0.000	UDP	68	DHCP Client	192.168.1.1		[02 01 06 00 F9 DD].r[04 00
120	16-06-2016 07:33:22.515	0.000	UDP	138	NBT Datagram...	beenachoudhry.do...		NBT DGRAM Packet: id:59
119	16-06-2016 07:33:13.033	0.000	UDP	67	DHCP			DHCP: Boot Request[0A]H
118	15-06-2016 21:57:08.409	0.000	UDP	138	NBT Datagram...	beenachoudhry.do...		NBT DGRAM Packet: id:48
117	15-06-2016 21:52:28.981	0.000	UDP	68	DHCP Client	192.168.1.1		[02 01 06 00 A2]m;[BD 04 0
116	15-06-2016 21:52:08.393	0.000	UDP	67	DHCP			DHCP: Boot Request[0A]H
115	15-06-2016 18:11:15.071	0.000	UDP	138	NBT Datagram...	beenachoudhry.do...		NBT DGRAM Packet: id:42
114	15-06-2016 18:11:10.734	0.000	UDP	67	DHCP	beenachoudhry.do...		DHCP: Boot Request[0A]H
113	15-06-2016 11:20:06.191	0.000	UDP	67	DHCP	beenachoudhry.do...		DHCP: Boot Request[0A]H
112	15-06-2016 11:14:37.844	0.000	UDP	138	NBT Datagram...	beenachoudhry.do...		NBT DGRAM Packet: id:60
111	15-06-2016 11:07:18.586	0.000	UDP	138	NBT Datagram...	beenachoudhry.do...		NBT DGRAM Packet: id:60
110	15-06-2016 11:07:14.218	0.000	UDP	67	DHCP	beenachoudhry.do...		DHCP: Boot Request[0A]H
109	15-06-2016 06:59:51.646	0.000	UDP	68	DHCP Client	192.168.1.1		[02 01 06 00 D0 ED]+[14 03
108	15-06-2016 06:59:52.879	0.000	UDP	138	NBT Datagram...	beenachoudhry.do...		NBT DGRAM Packet: id:44
107	15-06-2016 06:59:32.653	0.000	UDP	67	DHCP			DHCP: Boot Request[0A]H
106	14-06-2016 21:44:45.770	0.000	UDP	67	DHCP			DHCP: Boot Request[0A]H

User Rights: Admin [7B] Server: Running Visitors: 5 Events: 30/30



KFSensor Professional - Evaluation Trial
File View Scenario Signatures Settings Help
Visitors
- Activity
127.0.0.1 - tonec.com
169.254.80.103 - beenach..
192.168.1.1
192.168.1.2 - beenacho...
Name Value

ID	Start	Duration	Pro...	Sens...	Name	Visitor	Sig. Message	Received
130	16-06-2016 17:05:17.544	0.000	UDP	138	NBT Datagram...	beenachoudhry.do...		NBT DGRAM Packet: id:37
129	16-06-2016 16:01:39.560	0.000	UDP	67	DHCP			DHCP: Boot Request[0A]H
128	16-06-2016 16:00:42.988	0.000	UDP	138	NBT Datagram...	beenachoudhry.do...		NBT DGRAM Packet: id:37
127	16-06-2016 16:00:38.760	0.000	UDP	67	DHCP	beenachoudhry.do...		DHCP: Boot Request[0A]H
126	16-06-2016 13:20:19.752	0.000	UDP	68	DHCP Client	192.168.1.1		[02 01 06 00]e[F8 92 DA 05
125	16-06-2016 11:00:00.812	2.047	TCP	80	IIS	tonec.com		GET /cnwk.1d/i/newsburst
124	16-06-2016 07:51:23.710	0.000	UDP	138	NBT Datagram...	beenachoudhry.do...		NBT DGRAM Packet: id:51
123	16-06-2016 07:51:20.715	0.000	UDP	67	DHCP	beenachoudhry.do...		DHCP: Boot Request[0A]H
122	16-06-2016 07:35:03.271	0.000	UDP	138	NBT Datagram...	beenachoudhry.do...		NBT DGRAM Packet: id:59
121	16-06-2016 07:33:34.083	0.000	UDP	68	DHCP Client	192.168.1.1		[02 01 06 00 F9 DD].r[04 00
120	16-06-2016 07:33:22.515	0.000	UDP	138	NBT Datagram...	beenachoudhry.do...		NBT DGRAM Packet: id:59
119	16-06-2016 07:33:13.033	0.000	UDP	67	DHCP			DHCP: Boot Request[0A]H
118	15-06-2016 21:57:08.409	0.000	UDP	138	NBT Datagram...	beenachoudhry.do...		NBT DGRAM Packet: id:48
117	15-06-2016 21:52:28.981	0.000	UDP	68	DHCP Client	192.168.1.1		[02 01 06 00 A2]m;[BD 04 0
116	15-06-2016 21:52:08.393	0.000	UDP	67	DHCP			DHCP: Boot Request[0A]H
115	15-06-2016 18:11:15.071	0.000	UDP	138	NBT Datagram...	beenachoudhry.do...		NBT DGRAM Packet: id:42
114	15-06-2016 18:11:10.734	0.000	UDP	67	DHCP	beenachoudhry.do...		DHCP: Boot Request[0A]H
113	15-06-2016 11:20:06.191	0.000	UDP	67	DHCP	beenachoudhry.do...		DHCP: Boot Request[0A]H
112	15-06-2016 11:14:37.844	0.000	UDP	138	NBT Datagram...	beenachoudhry.do...		NBT DGRAM Packet: id:60
111	15-06-2016 11:07:18.586	0.000	UDP	138	NBT Datagram...	beenachoudhry.do...		NBT DGRAM Packet: id:60
110	15-06-2016 11:07:14.218	0.000	UDP	67	DHCP	beenachoudhry.do...		DHCP: Boot Request[0A]H
109	15-06-2016 06:59:51.646	0.000	UDP	68	DHCP Client	192.168.1.1		[02 01 06 00 D0 ED]+[14 03
108	15-06-2016 06:59:52.879	0.000	UDP	138	NBT Datagram...	beenachoudhry.do...		NBT DGRAM Packet: id:44
107	15-06-2016 06:59:32.653	0.000	UDP	67	DHCP			DHCP: Boot Request[0A]H
106	14-06-2016 21:44:45.770	0.000	UDP	67	DHCP			DHCP: Boot Request[0A]H

User Rights: Admin [7B] Server: Running Visitors: 5 Events: 30/30

RESULT

Thus honey pot was set up and monitored on network using KF Sensor done successfully.